

SPECIFICATION

TCC79XX

**High Performance and Low-Power Processor
For Digital Media Applications**

Rev. 1.03

Jan 23, 2009

Telechips

DISCLAIMER

All information and data contained in this material are without any commitment, are not to be considered as an offer for conclusion of a contract, nor shall they be construed as to create any liability. Any new issue of this datasheet invalidates previous issues. Product availability and delivery are exclusively subject to our respective order confirmation form; the same applies to orders based on development samples delivered. By this publication, Telechips, Inc. does not assume responsibility for patent infringements or other rights of third parties that may result from its use. Further, Telechips, Inc, reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of Telechips, Inc.

This product is designed for general purpose, and accordingly customer be responsible for all or any of intellectual property licenses required for actual application. Telechips does not provide any indemnification for any intellectual properties owned by third party.

Telechips can not ensure that this application is the proper and sufficient one for any other purposes but the one explicitly expressed herein. Telechips is not responsible for any special, indirect, incidental or consequential damage or loss whatsoever resulting from the use of this application for other purposes.

COPYRIGHT STATEMENT

Copyright in the material provided by Telechips is owned by Telechips unless otherwise noted.

For reproduction or use of Telechips' copyright material, permission should be sought from Telechips. That permission, if given, will be subject to conditions that Telechips' name should be included and interest in the material should be acknowledged when the material is reproduced or quoted, either in whole or in part. You must not copy, adapt, publish, distribute or commercialize any contents contained in the material in any manner without the written permission of Telechips. Trade marks used in Telechips' copyright material are the property of Telechips.

Important Notice

This product may include technology owned by Microsoft Corporation and in this case it cannot be used or distributed without a license from Microsoft Licensing, GP.

For customers who use licensed Codec ICs and/or licensed codec firmware of mp3:

"Supply of this product does not convey a license nor imply any right to distribute content created with this product in revenue-generating broadcast systems (terrestrial, Satellite, cable and/or other distribution channels), streaming applications(via internet, intranets and/or other networks), other content distribution systems(pay-audio or audio-on-demand applications and the like) or on physical media(compact discs, digital versatile discs, semiconductor chips, hard drives, memory cards and the like). An independent license for such use is required. For details, please visit <http://mp3licensing.com>".

For customers who use other firmware of mp3:

"Supply of this product does not convey a license under the relevant intellectual property of Thomson and/or Fraunhofer Gesellschaft nor imply any right to use this product in any finished end user or ready-to-use final product. An independent license for such use is required. For details, please visit <http://mp3licensing.com>"

For customers who use Digital Wave DRA solution:

"Supply of this implementation of DRA technology does not convey a license nor imply any right to this implementation in any finished end-user or ready-to-use terminal product. An independent license for such use is required."

Revision History

Revision	Date	Description
1.03	Jan 23, 2009	<ul style="list-style-type: none"> ● CKC - Page 23-17: The description about "Enter Halt Mode" is added. ● Introduction - Page 1-2: "Operating Modes" are modified. ● UART -Page 22-1: IrDA is removed. ● CKC - Page 23-4: Addresses of MCLKCTRL and SCLKCTRL Register are corrected. ● RTC - Page 28-12: PWDN description for PMWKUP is corrected.
1.02	Dec 17, 2008	<ul style="list-style-type: none"> ● Port Configuration - Reset value is corrected. <ul style="list-style-type: none"> - Page 33-3: AINCFG, Page 33-18: CCKO, Page 33-17:DAO, Page 33-15:PORTCFG9 Page 33-16:PORTCFG10 ● APPENDIX A <ul style="list-style-type: none"> - PLL Table is updated. (FPLL=118MHz, 295.2MHz, 354MHz) ● Changed Parts <ul style="list-style-type: none"> - LCD, CKC, I2C, Port Configuration, Boot Procedure - Refer to Change Log
1.01	Oct 8, 2008	<ul style="list-style-type: none"> ● Boot Procedure <ul style="list-style-type: none"> - Page 35-2: Data bit-width selection is corrected when booting from NOR. - Page 35-19: UART boot procedure is corrected.
1.00	Jul 9, 2008	<ul style="list-style-type: none"> ● Changed parts <ul style="list-style-type: none"> - LCD Interface, NTSC/PAL Encoder, Memory-to-Memory Scaler, USB Controller, NFC, Camera Interface, Graphic Engine, EHI, GPSB, CKC, Vectored Interrupt Controller, Timer/Counter, I2C, RTC, External Memory Controller, Port Multiplexer & GPIO, Boot Procedure - Refer to Change Log
0.20	Jun 23, 2008	<ul style="list-style-type: none"> ● Initial Release
0.10	Jan 10, 2008	

Table of Contents

DISCLAIMER	1
COPYRIGHT STATEMENT	1
REVISION HISTORY	3
TABLE OF CONTENTS	5
TABLE OF FIGURES	9
TABLE OF TABLES	13
1 INTRODUCTION	1-1
1.1 FEATURE	1-1
1.2 BLOCK DIAGRAM.....	1-4
1.3 PIN DESCRIPTIONS	1-5
2 ADDRESS AND REGISTER MAP	2-1
2.1 ADDRESS MAP.....	2-1
3 BUS ARCHITECTURE	3-1
3.1 MAIN BUS ARCHITECTURE.....	3-1
3.2 VIDEO HARDWARE BUS ARCHITECTURE.....	3-2
3.3 I/O BUS ARCHITECTURE.....	3-3
4 ARM926EJ-S MAIN PROCESSOR	4-1
4.1 OVERVIEW.....	4-1
4.2 FUNCTIONAL DESCRIPTION	4-2
4.3 COPROCESSOR CP15.....	4-6
5 LCD INTERFACE	5-1
5.1 OVERVIEW.....	5-1
5.2 RAW IMAGE SOURCE PROCESSING	5-3
5.3 DISPLAY INTERFACE.....	5-9
5.4 REGISTER DESCRIPTION	5-18
6 LCD SYSTEM INTERFACE	6-1
6.1 OVERVIEW.....	6-1
6.2 OPERATION	6-1
6.3 REGISTER DESCRIPTIONS	6-4
7 NTSC / PAL ENCODER COMPOSITE OUTPUT	7-1
7.1 OVERVIEWS	7-1
7.2 FEATURES	7-1
7.3 SELECTING OPERATION MODE.....	7-2
7.4 INPUT TIMING	7-2
7.5 VIDEO STANDARD SELECTION.....	7-5
7.6 BASIC VIDEO ADJUSTMENTS	7-5
7.7 PROGRAMMABLE BANDWIDTH.....	7-5
7.8 ANALOG VIDEO OUTPUT CONFIGURATION.....	7-6
7.9 REGISTERS	7-7
7.10 COPY GENERATION MANAGEMENT SYSTEMS.....	7-21
7.11 10-BIT DAC.....	7-23

8 MEMORY TO MEMORY SCALER	8-1
8.1 OVERVIEW.....	8-1
8.2 OPERATION	8-2
8.3 REGISTERS.....	8-4
9 USB CONTROLLER (HOST1.1, DEVICE2.0)	9-1
9.1 OVERVIEW.....	9-1
9.2 VBUS VALID AND END INTERRUPT.....	9-2
9.3 REGISTER DESCRIPTION FOR USB DEVICE CONTROLLER	9-3
9.4 REGISTER DESCRIPTION FOR USB Host CONTROLLER.....	9-18
9.5 REGISTER DESCRIPTION FOR UTMII (USB PHY).....	9-20
10 IDE.....	10-1
10.1 OVERVIEW.....	10-1
10.2 REGISTER DESCRIPTION.....	10-2
11 DMA CONTROLLER	11-1
11.1 OVERVIEW.....	11-1
11.2 REGISTER DESCRIPTION.....	11-2
12 MEMORY STICK HOST CONTROLLER	12-1
12.1 OVERVIEW.....	12-1
12.2 REGISTER DESCRIPTIONS	12-2
12.3 OPERATION & TIMING DIAGRAM	12-8
13 SD/SDIO/MMC/CE-ATA HOST CONTROLLER	13-1
13.1 OVERVIEW.....	13-1
13.2 BLOCK DIAGRAM	13-1
13.3 REGISTER DESCRIPTION.....	13-2
13.4 TIMING DIAGRAM.....	13-32
14 NFC	14-1
14.1 FUNCTION DESCRIPTION	14-1
14.2 REGISTER DESCRIPTION.....	14-1
15 ECC	15-1
15.1 FUNCTIONAL DESCRIPTION.....	15-1
15.2 REGISTER DESCRIPTION.....	15-2
16 CAMERA INTERFACE	16-1
16.1 OVERVIEW.....	16-1
16.2 OPERATION	16-2
16.3 CAMERA REGISTER DESCRIPTIONS.....	16-8
16.4 EFFECTER REGISTER DESCRIPTIONS	16-30
16.5 SCALER REGISTER DESCRIPTIONS	16-35
17 GRAPHIC ENGINE	17-1
17.1 OVERVIEW.....	17-1
17.2 FEATURES	17-3
17.3 REGISTER DESCRIPTION.....	17-5
17.4 GRAPHIC ENGINE OPERATION.....	17-26
18 EHI.....	18-1
18.1 OVERVIEW.....	18-1
18.2 REGISTER DESCRIPTION.....	18-2

18.3	OPERATION	18-14
19	DAI & CDIF (AUDIO & CD INTERFACE).....	19-1
19.1	DAI (DIGITAL AUDIO INTERFACE)	19-1
19.2	CDIF (CD MEDIA INTERFACE)	19-8
20	GPSB (GENERAL PURPOSE SERIAL BUS)	20-1
20.1	FUNCTIONAL DESCRIPTION	20-1
20.2	FEATURE	20-2
20.3	REGISTER DESCRIPTION	20-3
20.4	GPSB TIMING DIAGRAM	20-15
20.5	MPEG2-TS INTERFACE	20-17
21	SPDIF TRANSMITTER.....	21-1
21.1	OVERVIEW.....	21-1
21.2	REGISTER DESCRIPTIONS	21-2
21.3	OPERATION & TIMING DIAGRAM.....	21-8
22	UART	22-1
22.1	OVERVIEW.....	22-1
22.2	REGISTER DESCRIPTION	22-3
22.3	UART CONTROLLER REGISTER MAP	22-4
22.4	PORT MUX REGISTER.....	22-17
23	CKC.....	23-1
23.1	OVERVIEW.....	23-1
23.2	REGISTER DESCRIPTIONS	23-3
23.3	OPERATION & TIMING DIAGRAM.....	23-15
24	VECTORED INTERRUPT CONTROLLER	24-1
24.1	OVERVIEW.....	24-1
24.2	PRIORITY INTERRUPT CONTROLLER	24-3
24.3	VECTORED INTERRUPT CONTROLLER.....	24-9
25	TIMER / COUNTER	25-1
25.1	OVERVIEW.....	25-1
25.2	REGISTER DESCRIPTION	25-2
26	I2C CONTROLLER.....	26-1
26.1	FUNCTIONAL DESCRIPTION	26-1
26.2	I2C SLAVE FUNCTION	26-1
26.3	RELATED BLOCKS	26-2
26.4	REGISTER DESCRIPTION	26-3
27	ADC INTERFACE	27-1
27.1	OVERVIEW.....	27-1
27.2	ADC CONTROLLER REGISTER DESCRIPTION	27-2
28	RTC (REAL-TIME CLOCK).....	28-1
28.1	OVERVIEW.....	28-1
28.2	FUNCTION DESCRIPTION	28-2
28.3	RTC OPERATION	28-3
28.4	PROGRAMMER'S MODEL	28-6
29	EXTERNAL MEMORY CONTROLLER.....	29-1

29.1	OVERVIEW.....	29-1
29.2	SDRAM CONTROLLER	29-2
29.3	MISCELLANEOUS CONFIGURATION	29-7
29.4	EXTERNAL MEMORY CONTROLLER	29-9
29.5	SUB-REGISTERS OF NAND TYPE MEMORY	29-12
29.6	MISCELLANEOUS CONTROL REGISTER	29-14
30	SRAM INTERFACE	30-1
30.1	OVERVIEW.....	30-1
30.2	REGISTERS.....	30-2
31	MAILBOX.....	31-1
31.1	OVERVIEW.....	31-1
31.2	REGISTER DESCRIPTIONS	31-2
31.3	OPERATION & TIMING DIAGRAM	31-6
32	MISCELLANEOUS CONTROLLER	32-1
32.1	I/O Bus Controller	32-2
32.2	SYSTEM CONTROLLER.....	32-4
32.3	VIRTUAL MMU TABLE	32-7
33	PORT MULTIPLEXER & GPIO.....	33-1
33.1	OVERVIEW.....	33-1
33.2	REGISTER DESCRIPTION.....	33-2
34	JTAG DEBUG INTERFACE.....	34-1
34.1	INTERNAL JTAG CONNECTIONS	34-1
34.2	EXTERNAL JTAG CONNECTIONS.....	34-3
34.3	DEBUGGING WITH OPENICE32 & AIJI SPIDER	34-4
35	BOOT PROCEDURE	35-1
35.1	POWER UP/DOWN SEQUENCE FOR CORE AND I/O POWER.....	35-1
35.2	BOOT MODE	35-2
35.3	OVERALL PROCEDURE.....	35-4
35.4	EHI Boot (BM[2:0] = 000B)	35-5
35.5	USB Boot (BM == 011B).....	35-8
35.6	EXTERNAL NOR Boot (BM == 110B).....	35-10
35.7	NFC NAND Boot (BM == 111B).....	35-11
35.8	I2C MASTER Boot – EEPROM Boot (BM == 001B).....	35-14
35.9	SERIAL FLASH Boot.....	35-15
35.10	SPI SLAVE Boot	35-17
35.11	UART Boot	35-18
35.12	DUAL CRC CHECKING	35-20
APPENDIX A.....		I
CHANGE LOG		I

Table of Figures

FIGURE 1.1 TCC79XX FUNCTIONAL BLOCK DIAGRAM	1-4
FIGURE 3.1 THE MAIN BUS ARCHITECTURE	3-1
FIGURE 3.2 THE VIDEO HARDWARE BUS ARCHITECTURE	3-2
FIGURE 3.3 THE I/O HARDWARE BUS ARCHITECTURE	3-3
FIGURE 4.1 THE TCC79XX CORE ARCHITECTURE	4-1
FIGURE 4.2 LITTLE-ENDIAN ADDRESSES OF BYTES-WORDS	4-2
FIGURE 5.1 LCD CONTROLLER BLOCK DIAGRAM	5-1
FIGURE 5.2 OVERALL IMAGE DATA FLOW	5-2
FIGURE 5.3 VIRTUAL DISPLAY IN THE LCDC	5-3
FIGURE 5.4 SUPPORTED PIXEL DATA FORMAT (BPP BITS OF LINC REGISTER)	5-4
FIGURE 5.5 RGB2RGB888 CONVERSION	5-5
FIGURE 5.6 RAW IMAGE OFFSET	5-6
FIGURE 5.7 LCD COLOR LOOKUP TABLE	5-7
FIGURE 5.8 PIP IN THE LCDC	5-9
FIGURE 5.9 CONNECTION OF LCDC OUTPUT SIGNALS	5-9
FIGURE 5.10 STN MODE TIMING DIAGRAM	5-12
FIGURE 5.11 TFT MODE TIMING DIAGRAM	5-13
FIGURE 5.12 NTSC INTERLACE MODE TIMING DIAGRAM	5-15
FIGURE 5.13 PAL INTERLACE MODE TIMING DIAGRAM	5-15
FIGURE 5.14 CCIR656 EMBEDDED SYNC. INFORMATION	5-16
FIGURE 5.15 CCIR656 FORMAT DIAGRAM	5-17
FIGURE 5.16 OUTPUT PIXEL DATA FORMAT	5-22
FIGURE 5.17 CLOCK CONTROL OF LCD INTERFACE	5-24
FIGURE 5.18 BIT PADDING	5-34
FIGURE 5.19 SUPPORTED PIXEL DATA FORMAT (BPP, YCbCr, AND BR OF LINC REGISTER)	5-36
FIGURE 5.20 VERTICAL SYNC DETECTOR IN PIP MODE	5-51
FIGURE 6.1 LCD SYSTEM INTERFACE BLOCK DIAGRAM	6-1
FIGURE 6.2 RELATIONSHIP BETWEEN LCSN/LXA AND ADDRESS	6-1
FIGURE 6.3 WRITING / READING OPERATION THROUGH ON-CHIP CPU	6-2
FIGURE 6.4 EXAMPLE OF LCDC OUTPUT SIGNALS FOR LCDSI	6-3
FIGURE 6.5 LCDSI OUTPUT PIXEL DATA ORGANIZATION ACCORDING TO FMT	6-6
FIGURE 6.6 TIMING CONFIGURATION OF LCDSI OUTPUT SIGNALS FOR THE ON-CHIP CPU ACCESS	6-9
FIGURE 6.7 TIMING CONFIGURATION OF LCDSI OUTPUT SIGNALS FOR THE LCDC	6-10
FIGURE 7.1 DIGITAL INPUT TIMING (ITU-R BT.656 8BIT PARALLEL INPUT)	7-3
FIGURE 7.2 DIGITAL INPUT TIMING (ITU-R BT.601 4:2:2 16BIT PARALLEL INPUT)	7-3
FIGURE 7.3 EXAMPLE OF INPUT TIMING	7-4
FIGURE 7.4 LUMA BANDWIDTH	7-6
FIGURE 7.5 CHROMA BANDWIDTH	7-6
FIGURE 7.6 COPY GENERATION MANAGEMENT SYSTEMS	7-21
FIGURE 7.7 EXAMPLE: CRC CALCULATOR FOR CGMS	7-22
FIGURE 7.8 10-BIT DAC CONNECTION	7-23
FIGURE 8.1 SCALER BLOCK DIAGRAM	8-1
FIGURE 8.2 MEMORY TO MEMORY SCALING OPERATION	8-2
FIGURE 8.3 STORING THE RESULT IMAGE	8-3
FIGURE 9.1 USB CONTROLLER BLOCK DIAGRAM	9-1
FIGURE 9.2 USB HOST CONTROLLER BLOCK DIAGRAM	9-18
FIGURE 10.1 IDE CONTROLLER BLOCK DIAGRAM	10-1
FIGURE 10.2 PIO INTERFACE TIMING DIAGRAMS	10-4
FIGURE 10.3 HSTROBE AND MAX3 COUNTER TIMING DIAGRAMS	10-6
FIGURE 11.1 GDMA CONTROLLER BLOCK DIAGRAM	11-1

FIGURE 11.2 RELATION BETWEEN HOP AND BURST TRANSFERS (IF BURST SIZE IS 4.).....	11-8
FIGURE 11.3 THE EXAMPLE OF VARIOUS TYPES OF TRANSFER	11-9
FIGURE 11.4 DATA TRANSFER WHEN CHANNEL0 AND CHANNEL1 ARE ENABLED	11-14
FIGURE 12.1 MEMORY STICK HOST CONTROLLER BLOCK DIAGRAM.....	12-1
FIGURE 12.2 COMMUNICATION EXAMPLE	12-8
FIGURE 12.3 PARALLEL MODE SETTING PROCEDURE	12-9
FIGURE 12.4 THE TIMING DIAGRAM OF SERIAL INTERFACE READ PROTOCOL.....	12-11
FIGURE 12.5 THE TIMING DIAGRAM OF SERIAL INTERFACE WRITE PROTOCOL	12-12
FIGURE 12.6 THE TIMING DIAGRAM OF SERIAL INTERFACE INT GENERATION.....	12-12
FIGURE 12.7 THE TIMING DIAGRAM OF PARALLEL INTERFACE READ PROTOCOL.....	12-13
FIGURE 12.8 THE TIMING DIAGRAM OF PARALLEL INTERFACE WRITE PROTOCOL	12-14
FIGURE 12.9 THE TIMING DIAGRAM OF PARALLEL INTERFACE INT GENERATION.....	12-14
FIGURE 13.1 SD/SDIO/MMC/CE-ATA BLOCK DIAGRAM	13-1
FIGURE 13.2 SDIO/SD – WRITE INTERRUPT CYCLE TIMING	13-32
FIGURE 13.3 SDIO/SD – READ INTERRUPT CYCLE TIMING.....	13-32
FIGURE 13.4 CMD60 READ TRANSACTION	13-32
FIGURE 13.5 CMD60 WRITE TRANSACTION	13-33
FIGURE 14.1 NAND FLASH CONTROLLER BLOCK DIAGRAM.....	14-1
FIGURE 14.2 EXAMPLE OF ADDRESS/COMMAND WRITING OPERATION	14-4
FIGURE 14.3 TIMING DIAGRAM OF READ/WRITE ENANLE SIGNAL	14-8
FIGURE 14.4 EXAMPLE OF PAGE PROGRAM	14-12
FIGURE 14.5 EXAMPLE OF PAGE READ	14-13
FIGURE 15.1 ECC BLOCK DIAGRAM	15-1
FIGURE 15.2 SLC ECC ENCODING SEQUENCE.....	15-14
FIGURE 15.3 SLC ECC DECODING SEQUENCE.....	15-15
FIGURE 15.4 MLC ECC4 ENCODING SEQUENCE	15-16
FIGURE 16.1 CIF BLOCK DIAGRAM.....	16-2
FIGURE 16.2 INPUT DATA FORMAT FROM THE EXTERNAL CAMERA MODULE	16-3
FIGURE 16.3 INPUT IMAGE SIZE FROM THE EXTERNAL CAMERA MODULE	16-3
FIGURE 16.4 ACCEPTING FRAME DATA FROM THE EXTERNAL CAMERA MODULE	16-4
FIGURE 16.5 OUTPUT IMAGE OF THE EFFECTER AND INPUT IMAGE OF THE CIF SCALER.....	16-5
FIGURE 16.6 THE GEOMETRIC PROPERTY OF OVERLAY IMAGE	16-6
FIGURE 16.7 IMAGE DATA TO BE STORED TO THE MEMORY	16-6
FIGURE 16.8 TWO MODES TO STORE THE IMAGE DATA.....	16-7
FIGURE 16.9 CCIR-656 FORMAT DIAGRAM	16-11
FIGURE 16.10 INPUT IMAGE WINDOWING	16-13
FIGURE 16.11 OVERLAY IMAGE WINDOWING	16-24
FIGURE 17.1 GRAPHIC ENGINE BLOCK DIAGRAM (FCH2 IS NOT AVAILABLE IN THE TCC79XX).....	17-2
FIGURE 17.2 SUPPORTED DATA FORMAT OF GRAPHIC ENGINE.....	17-4
FIGURE 17.3 EXAMPLE: INPUT SOURCE IMAGE DATA TO RGB888 CONVERSION	17-8
FIGURE 17.4 SOURCE & DESTINATION MIRROR / ROTATE OPERATION	17-26
FIGURE 17.5 SOURCE YUV TO RGB CONVERSION	17-27
FIGURE 17.6 ARITHMETIC OPERATION	17-28
FIGURE 17.7 ROP / ALPHA-BLENDING BLOCK DIAGRAM	17-29
FIGURE 17.8 ROP AND ALPHA-BLENDING OPERATION EXAMPLE.....	17-29
FIGURE 17.9 RGB TO YUV FORMAT CONVERTER	17-31
FIGURE 18.1 EXAMPLE OF EHAM USAGE	18-7
FIGURE 18.2 INTERRUPT REQUEST USING EHCFG.WRIRQ AND CSIRQ	18-11
FIGURE 18.3 EXAMPLE OF WRITING / READING OPERATION (80 INTERFACE, 16BITS).....	18-14
FIGURE 18.4 EXAMPLE OF WRITING / READING OPERATION (68 INTERFACE, 16BITS)	18-15
FIGURE 18.5 ACCESS TO THE ON-CHIP SYSTEM BUS	18-16
FIGURE 18.6 LOCK TRANSFER VS NORMAL TRANSFER (EHRWCS.BSIZE=3)	18-17
FIGURE 18.7 EXAMPLE OF WRITING TO THE ON-CHIP SYSTEM BUS (80 INTERFACE, 8BITS)	18-18
FIGURE 18.8 EXAMPLE OF READING FROM THE ON-CHIP SYSTEM BUS (68 INTERFACE, 8BITS).....	18-18

FIGURE 18.9 PSEUDO CODE FOR GETTING AND RELEASING A SEMAPHORE	18-19
FIGURE 19.1 DAI BLOCK DIAGRAM	19-1
FIGURE 19.2 DAI BUS TIMING DIAGRAM	19-7
FIGURE 19.3 CDIF BLOCK DIAGRAM	19-10
FIGURE 19.4 CDIF BUS TIMING DIAGRAM	19-11
FIGURE 20.1 GPSB INTERFACE BLOCK DIAGRAM	20-1
FIGURE 20.2 OVERALL GPSB BLOCK DIAGRAM	20-1
FIGURE 20.3 PACKET STRUCTURE	20-10
FIGURE 20.4 TX/RX ADDRESSING MODES	20-12
FIGURE 20.5 SPI TIMING 0	20-15
FIGURE 20.6 SPI TIMING 1	20-15
FIGURE 20.7 SPI TIMING 2	20-15
FIGURE 20.8 SPI TIMING 3	20-15
FIGURE 20.9 SSP TIMING	20-16
FIGURE 20.10 BITSTREAM OF MPEG2-TS	20-17
FIGURE 20.11 MPEG2-TS TIMING INFORMATION	20-17
FIGURE 22.1 UART BLOCK DIAGRAM	22-1
FIGURE 23.1 CKC BLOCK DIAGRAM	23-1
FIGURE 23.2 HARDWARE BLOCK GROUP	23-9
FIGURE 23.3 CLOCK CHANGE TIMING DIAGRAM	23-15
FIGURE 23.4 PERIPHERAL CLOCK CONFIGURATION PROCEDURE	23-16
FIGURE 23.5 TIMING DIAGRAM TO ENTER AND EXIT HALT MODE	23-17
FIGURE 23.6 TIMING DIAGRAM OF ENTER AND EXIT POWER-DOWN MODE	23-18
FIGURE 24.1 BLOCK DIAGRAM OF VECTORED INTERRUPT CONTROLLER	24-1
FIGURE 25.1 16-BIT AND 20BIT TIMER/COUNTER BLOCK DIAGRAM	25-1
FIGURE 25.2 TIMING DIAGRAM OF TIMER/COUNTER	25-4
FIGURE 25.3 32-BIT COUNTER BLOCK DIAGRAM	25-7
FIGURE 26.1 I2C BLOCK DIAGRAM	26-1
FIGURE 26.2 I2C SLAVE BLOCK DIAGRAM	26-2
FIGURE 27.1 ADC CONTROLLER BLOCK DIAGRAM	27-1
FIGURE 28.1 RTC BLOCK DIAGRAM	28-1
FIGURE 28.2 THE RTC TIME SETTING SEQUENCE	28-3
FIGURE 28.3 RTC ALARM TIME SETTING SEQUENCE	28-4
FIGURE 28.4 PEND CLEAR SEQUENCE	28-4
FIGURE 28.5 RTC OPERATION PROCESS FLOW CHART	28-5
FIGURE 28.6 EXAMPLE OF CRYSTAL OSCILLATOR CIRCUIT CONNECTION	28-6
FIGURE 29.1 MEMORY CONTROLLER BLOCK DIAGRAM	29-1
FIGURE 29.2 SDRAM CYCLE DIAGRAM	29-3
FIGURE 29.3 BASIC TIMING DIAGRAM FOR EXTERNAL MEMORIES	29-11
FIGURE 30.1 RELATIONSHIP BETWEEN THE INTERNAL SYSTEM ADDRESS AND SRAMIF CS/XA	30-1
FIGURE 30.2 TIMING CONFIGURATION FOR SRAMIF OUTPUT SIGNALS	30-3
FIGURE 30.3 INTERFACE SIGNALS ACCORDING TO BW[1:0] BITS	30-4
FIGURE 31.1 MAILBOX HARDWARE BLOCK DIAGRAM	31-1
FIGURE 31.2 FLOW-CHART FOR "SENDMESSAGE"	31-6
FIGURE 31.3 FLOW-CHART FOR "RECVMESSAGE"	31-7
FIGURE 33.1 I/O FUNCTION ACCORDING TO BP BIT	33-20
FIGURE 34.1 INTERNAL JTAG CONNECTION WHEN SDO1 AND SDO0 ARE '0'	34-1
FIGURE 34.2 INTERNAL JTAG CONNECTION WHEN SDO1 = 0 AND SDO0 = 1	34-2
FIGURE 34.3 INTERNAL JTAG CONNECTION WHEN SDO1 AND SDO0 ARE '1'	34-2
FIGURE 34.4 JTAG INTERFACE CIRCUIT DIAGRAM	34-3
FIGURE 34.5 CONNECTION BETWEEN HOST PC AND OPENICE32 AND TARGET SYSTEM	34-4
FIGURE 34.6 PIN CONFIGURATION OF 20 AND 14 PIN CONNECTOR	34-5
FIGURE 34.7 20-14 ADAPTER BOARD CONNECTOR	34-5
FIGURE 34.8 20-14 ADAPTER BOARD	34-5

FIGURE 35.1 POWER UP/DOWN SEQUENCE FOR CORE AND I/O POWER	35-1
FIGURE 35.2 RESET SEQUENCE	35-3
FIGURE 35.3 OVERALL FLOWCHART OF BOOT CODE	35-4
FIGURE 35.4 INITCFG BIT FIELD	35-5
FIGURE 35.5 BIP DATA STRUCTURE	35-5
FIGURE 35.6 EXTERNAL HOST UPLOADS PROGRAM WHEN BIPEN = 1	35-6
FIGURE 35.7 EHI Boot PROCEDURE	35-7
FIGURE 35.8 USB BOOT PROCEDURE	35-9
FIGURE 35.9 EXTERNAL NOR BOOT PROCEDURE	35-10
FIGURE 35.10 NAND BOOT PROCEDURE	35-12
FIGURE 35.11 BASIC STRUCTURE AND BOOT FLOW OF NAND BOOT	35-13
FIGURE 35.12 I2C Boot PROCEDURE	35-14
FIGURE 35.13 DATA STRUCTURE FOR HEADER INFORMATION STORED IN SERIAL FLASH	35-15
FIGURE 35.14 PSEUDO CODE FOR SERIAL FLASH BOOTING PROCEDURE	35-16
FIGURE 35.15 SPI SLAVE Boot PROCEDURE	35-17
FIGURE 35.16 UART BOOT PROCEDURE	35-19

Table of Tables

TABLE 4.1 EXCEPTION VECTOR AND PRIORITIES	4-4
TABLE 4.2 EXCEPTION ENTRY/EXIT	4-5
TABLE 5.1 STN LCD PALETTE ADDRESS	5-11
TABLE 5.2 STN LCD DITHERING PATTERN REGISTER MAP	5-11
TABLE 5.3 MONOCHROME STN LCD (4BITS, 1BPP) EXAMPLE	5-12
TABLE 5.4 TFT LCD (RGB565) EXAMPLE	5-14
TABLE 5.5 LCDC REGISTER MAP (BASE ADDRESS = 0xF0000000)	5-18
TABLE 6.1 LCDSI REGISTER MAP(0xF0000000).....	6-4
TABLE 7.1 SETTING FOR OUTPUT FORMATS	7-5
TABLE 7.2 SUMMARY OF DAC VOLTAGE AND CODES	7-6
TABLE 8.1 SCALER REGISTERS (BASE ADDRESS = 0xF0070000).....	8-4
TABLE 9.1 USB REGISTER MAP (BASE ADDRESS = 0xF0010000)	9-3
TABLE 9.2 USB HOST REGISTER MAP (BASE ADDRESS = 0xF0020000)	9-19
TABLE 10.1 SPEED OF DATA TRANSFER (BYTE PER SECOND)	10-1
TABLE 10.2 IDE REGISTERS (BASE = 0xF0030000).....	10-2
TABLE 11.1 GENERAL DMA CONTROLLER REGISTER MAP	11-2
TABLE 11.2 BASE ADDRESS OF ALL GDMA CHANNELS	11-3
TABLE 12.1 MSHC REGISTER MAP (BASE ADDRESS = 0xF0051000)	12-2
TABLE 12.2 PORTCFG REGISTER MAP (BASE ADDRESS = 0xF0051080)	12-2
TABLE 13.1 SLOT0/1 REGISTER MAP (BASE ADDRESS = 0xF0090000/0xF0090100).....	13-2
TABLE 13.2 CHANNEL CONTROL REGISTER MAP (BASE ADDRESS = 0xF0090800)	13-3
TABLE 13.3 COMMAND FORMAT	13-6
TABLE 13.4 RESPONSE REGISTER	13-9
TABLE 13.5 RELATION BETWEEN TRANSFER COMPLETE AND DATA TIMEOUT ERROR	13-19
TABLE 13.6 RELATION BETWEEN COMMAND COMPLETE AND COMMAND TIMEOUT ERROR	13-19
TABLE 13.7 RELATION BETWEEN AUTO CMD12 CRC ERROR AND AUTO CMD12 TIMEOUT ERROR.....	13-24
TABLE 14.1 NAND FLASH CONTROLLER REGISTER MAP (BASE ADDRESS = 0xF00B0000).....	14-1
TABLE 14.2 PAGE SIZE OF NAND FLASH	14-3
TABLE 15.1 ECC REGISTER MAP (BASE ADDRESS = 0XF005B000).....	15-2
TABLE 16.1 CIF REGISTER MAP (BASE ADDRESS = 0xF0060000)	16-8
TABLE 16.2 EFFECT REGISTER MAP (BASE ADDRESS = 0xF0060100)	16-30
TABLE 16.3 SCALER REGISTER MAP (BASE ADDRESS = 0xF0060200)	16-35
TABLE 17.1 GRAPHIC ENGINE REGISTER MAP (BASE ADDRESS = 0xF6000000).....	17-5
TABLE 18.1 EHI EXTERNAL INTERFACE PIN	18-1
TABLE 18.2 EHI REGISTER MAP.....	18-2
TABLE 18.3 ACCESS TO THE EHI	18-14
TABLE 19.1 DAI REGISTER MAP (BASE ADDRESS = 0XF0059000).....	19-3
TABLE 19.2 CDIF REGISTER MAP (BASE ADDRESS = 0XF0059080)	19-8
TABLE 20.1 GPSB REGISTER MAP (BASE ADDRESS = 0XF0057000).....	20-3
TABLE 21.1 SPDIF REGISTER MAP (BASE ADDRESS = 0xF005C000).....	21-1
TABLE 22.1 UART REGISTER MAP	22-3
TABLE 22.2 UART PORT MUX REGISTER	22-3
TABLE 23.1 CKC REGISTER MAP (BASE ADDRESS = 0XF3000000)	23-2
TABLE 24.1 PRIORITY INTERRUPT CONTROLLER REGISTER MAP (BASE ADDRESS = 0XF3001000)	24-1
TABLE 24.2 VECTORED INTERRUPT CONTROLLER REGISTER MAP (BASE ADDRESS = 0XF3001080).....	24-2
TABLE 25.1 TIMER/COUNTER REGISTER MAP (BASE ADDRESS = 0XF3003000).....	25-2
TABLE 25.2 TC32 COUNT MODE	25-7
TABLE 26.1 I2C REGISTER MAP (BASE ADDRESS = 0XF0052000)	26-3
TABLE 27.1 ADC CONTROLLER REGISTER MAP (BASE ADDRESS = 0XF3004000)	27-2
TABLE 28.1 RECOMMENDED OSCILLATOR CIRCUIT CONSTANTS	28-5

TABLE 28.2 RTC REGISTER MAP (BASE ADDRESS = 0xF3002000).....	28-6
TABLE 28.3 RTC CONTROL REGISTER (RTCCON)	28-7
TABLE 28.4 RTC INTERRUPT CONTROL REGISTER (INTCON)	28-8
TABLE 28.5 RTC ALARM CONTROL REGISTER (RTCALM)	28-8
TABLE 28.6 ALARM SECOND DATA REGISTER (ALMSEC).....	28-8
TABLE 28.7 ALARM MINUTE DATA REGISTER (ALMMIN).....	28-9
TABLE 28.8 ALARM HOUR DATA REGISTER (ALMHOUR)	28-9
TABLE 28.9 ALARM DATE DATA REGISTER (ALMDATE).....	28-9
TABLE 28.10 ALARM DAY OF WEEK DATA REGISTER (ALMDAY).....	28-9
TABLE 28.11 ALARM MONTH DATA REGISTER (ALMMON).....	28-10
TABLE 28.12 ALARM YEAR DATA REGISTER (ALMYEAR)	28-10
TABLE 28.13 BCD SECOND DATA REGISTER (BCDSEC)	28-10
TABLE 28.14 BCD MINUTE DATA REGISTER (BCDMIN)	28-10
TABLE 28.15 BCD HOUR DATA REGISTER (BCDHOUR)	28-11
TABLE 28.16 BCD DATE DATA REGISTER (BCDDATE)	28-11
TABLE 28.17 BCD DAY OF WEEK DATA REGISTER (BCDDAY)	28-11
TABLE 28.18 BCD MONTH DATA REGISTER (BCDMON)	28-11
TABLE 28.19 BCD YEAR DATA REGISTER (BCDYEAR)	28-12
TABLE 28.20 RTC INTERRUPT MODE REGISTER (RTCIM).....	28-12
TABLE 28.21 RTC INTERRUPT PENDING REGISTER (RTCPEND)	28-12
TABLE 29.1 CHANNEL 0 MEMORY CONTROLLER REGISTER MAP (BASE ADDRESS = 0xF1000000).....	29-2
TABLE 29.2 NAND FLASH REGISTER MAP (BASE ADDRESS = N * 0x10000000)	29-2
TABLE 29.3 PAGE SIZE OF NAND FLASH.....	29-12
TABLE 30.1 SRAMIF REGISTER MAP (0xF0080000)	30-2
TABLE 31.1 MAILBOX REGISTER MAP (BASE ADDRESS = 0xF4000000).....	31-1
TABLE 32.1 IOBC REGISTER MAP (BASE ADDRESS = 0xF0050000)	32-1
TABLE 32.2 SYSC REGISTER MAP (BASE ADDRESS = 0xF3005000)	32-1
TABLE 32.3 VMT REGISTER MAP (BASE ADDRESS = 0XF7000000).....	32-1
TABLE 33.1 GPIO REGISTER MAP (BASE ADDRESS = 0xF005A000).....	33-2
TABLE 35.1: CONFIGURATION VALUE	35-2
TABLE 35.2 SUPPORTED NAND FLASH TYPES.....	35-12

1 INTRODUCTION

The TCC79XX is the system LSI for digital multimedia applications based on ARM926EJ-S, ARM's proprietary 32-bit RISC CPU core. It can decode and encode various types of audio/video standards with software and dedicated hardware accelerators – MP3 / AAC / MPEG4-AAC / MPEG4-BSAC, JPEG / MPEG1 / MPEG2 / MPEG4-SP/ASP / H.264 and other types of audio/video standard.

The on-chip high speed USB2.0 device controller enables the data transmission between a personal computer and storage device such as NAND flash, HDD, CD, SD, MMC and Memory Stick etc, which can be controlled by the TCC79XX.

1.1 FEATURE

- 32 Bits ARM926EJ-S RISC CPU Core for System and Audio Processing
 - 16KB/16KB Instruction and Data Caches
 - MMU Supported
 - Java Acceleration Supported
 - 4KB Instruction TCM(Tightly Coupled Memory for Exception Handler)
 - 8KB Data TCM (Tightly Coupled Memory for Fast Data Transfer H/W)
 - JTAG interface for code debugging
- MEMORY ORGANIZATION
 - Boot ROM of 8Kbytes for various boot procedure (HPI BOOT, NAND, USB) and security
 - Internal SRAM of 64KB for program shared memory with hardware
 - Internal SRAM of 32KB for Video Processor
 - 8KB DTCM interface for communication with software and hardware
 - Memory controller for various memories including PROM, NOR type Flash, SRAM, and SDRAM etc.
- VIDEO HARDWARE ACCELERATOR
 - MPEG1/MPEG2/MPEG4/Divx/Xvid/MPEG4-ASP/AVC
 - Supported Video Decoder
 - ◆ MPEG1, MPEG2
 - ◆ MPEG4-SP
 - ◆ MPEG4-ASP
 - ◆ MPEG4-AVC (H.264)
 - ◆ DIVX
 - ◆ H.263
 - ◆ WMV9
 - Supported Video Encoder
 - ◆ MPEG4-SP
 - ◆ H.263
 - ◆ Bitstream Accelerator
 - Hardwired VLD for MPEG4 Decoding
 - ◆ Bitstream DMA for Software Bit Manipulation
 - Hardwired VLD for MPEG4 Decoding
 - Hardwired CAVLD for MPEG4-AVC (H.264)
 - Various Types of Motion Interpolator (Frame Reconstructor)
 - ◆ MPEG4-SP : Half-pel and Quarter-pel
 - ◆ MPEG4-AVC : 6 Tap Interpolator
 - ◆ WMV9 : Bi-linear & Bi-cubic Interpolator
 - SIMD(Single Instruction/Multiple Data) Architectured DSP
 - ◆ 8x8 DCT and IDCT for JPEG, MPEG
 - ◆ 4x4 Integer Transform for MPEG4-AVC
 - ◆ 8x8, 8x4, 4x8 Integer Transform for WMV9
 - ◆ Overlapped Transform for WMV9
 - ◆ Quantizer/Inverse Quantizer
 - Motion Estimator

- ◆ 5x5 Points Half-pel SAD Calculator
- ◆ Full-Search Motion Estimator Controller
- ◆ Fast Search Motion Estimator Controller
- 4KB Video Cache
- Well-Organized Pipelined Architecture Controller
- VIDEO I/O
 - LCD Interface
 - ◆ Progressive or Interlaced Digital Video Output
 - ◆ Support of STN LCD
 - ◆ Support of CCIR-601/656
 - ◆ Support of TFT LCD
 - RGB Interface : Up to 24-bit data width
 - CPU Interface : Up to 18-bit data width
 - ◆ 16-Level Image Overlay & Chroma-Keying, OSD
 - ◆ Color Space Converter
 - ◆ Three 256-Level Color Look-Up Table
 - ◆ Dual-LCD Controller (CPU Interface only)
 - Camera Interface
 - ◆ Support of CCIR-601/656 Format
 - ◆ Image Effector
 - ◆ Hardware Scaler
 - ◆ 4-Level Image Overlay, Chroma-Keying & Scaler for Preview
 - ◆ Various Input Format : YCbCr 4:2:2/4:2:0, RGB 4:2:2, 4:2:0, Bayer RGB
 - ◆ Color Space Dispatcher
 - ◆ Up-to 5M Pixels with Scaling
 - NTSC/PAL composite output
 - ◆ NTSC-M/4.43, PAL-B/D/G/H/I/M/N/Combination N
 - ◆ 10-bit DAC
- AUDIO & STORAGE INTERFACES
 - AUDIO I/O
 - ◆ I2S Master & Slave Interface
 - ◆ S/PDIF TX Interface
 - HIGH SPEED INTERFACE
 - ◆ High speed USB2.0 device / USB1.1 host
 - ◆ Host port interface via 80/68 compatible – 8 / 16 bits
 - STORAGE I/F
 - ◆ UDMA33/66, PIO, IDE Interface for HDD
 - ◆ Various Memory Card Interfaces for NAND / SD / MMC / Memory Stick etc.
 - MISC. INTERFACE
 - ◆ 4-Channel UART for serial host interface
 - 2-channel without hardware flow control
 - 2-channel with hardware flow control
 - ◆ Configurable 2-Channel I2C
 - 2-Channel Masters
 - 1 Channel Master and 1 Channel I2C Slave
 - ◆ 4-Channel GPSB with TS interface for supporting various serial interfaces
- ON-CHIP PERIPHERALS
 - Four 16bit timer with PWM output/counters and one 32bit timer
 - 9 Channel DMA for transferring bulk data
 - ◆ Including Hardware DMA Request for Each Channel
 - 8 channel General purpose 10-bit ADC
 - Operating Modes
 - ◆ Normal Operation
 - ◆ Deep Power Down : RTC Only works
 - ◆ LCD Bypass Mode : HPI passes through LCD ports
 - ◆ Power Down : All the oscillator disabled

- RTC : Power-Down Mode & Auto-wakeup
- 2 Channel Clock Outputs Controlled by Clock Generator
- PROCESS & ELECTRICAL CHARACTERISTICS
 - 65nm CMOS process

1.2 Block Diagram

The following figure shows the block diagram of the TCC79XX.

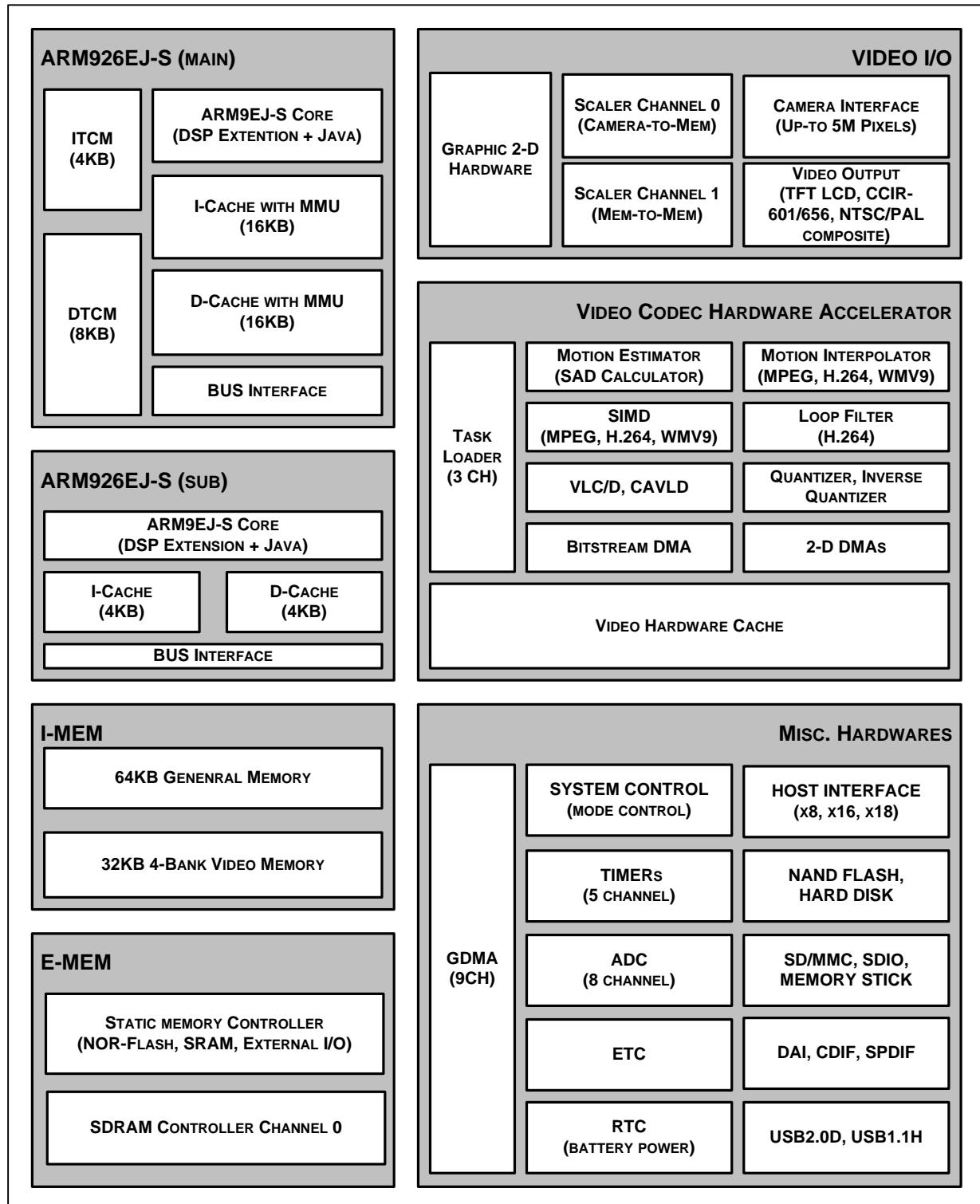


Figure 1.1 TCC79XX Functional Block Diagram

1.3 Pin Descriptions

The TCC79XX is a CMOS device. Floating level on input signals cause unstable device operation and abnormal current consumption. Pull-up or pull-down resistors should be used appropriately for input or bidirectional pins.

Refer to CHIP SPEC for more information.

2 ADDRESS AND REGISTER MAP

2.1 Address Map

The TCC79XX has fixed address maps for on-chip resources and off-chip resources. The following table represents overall address space of system.

Address Space	Region	Device Name
0x00000000 – 0x0FFFFFFF	R.0	<p>This region is remapped to as follows.</p> <ol style="list-style-type: none"> 1) If Remap is 000b, On-chip memory for region R.1 2) If Remap is 001b, Off-chip SDRAM for region R.2 3) If Remap is 011b, External Memory CS3 for region R.7 4) If Remap is 111b, On-chip boot-ROM for region R.E 5) In any other case, this region is not remapped to any region.
0x00000000 – 0x00000FFF		Special Region for Instruction TCM
0x10000000 – 0x1000FFFF	R.1	Assigned to on-chip 64kB memory Region
0x10010000 – 0x10017FFF		Assigned to on-chip 32kB Video Memory Region
0x40000000 – 0x4FFFFFFF	R.4	Assigned to external chip select CSN_CS0
0x50000000 – 0x5FFFFFFF	R.5	Reserved
0x60000000 – 0x6FFFFFFF	R.6	Reserved
0x70000000 – 0x7FFFFFFF	R.7	Assigned to external chip select CSN_NOR
0xA0000000 – 0xA0001FFF	R.A	Assigned to DTCM memory region
0xE0000000 – 0xEFFFFFFF	R.E	Assigned to internal boot ROM
0xF0000000 – 0xFFFFFFFF	R.F	Assigned to on-chip peripherals

The address space (0x00000000 ~ 0x0FFFFFFF) is initially allocated to internal or external NOR flash memory for booting procedure, and a special flag exists in system controller unit for remapping this space to other type of memories. Refer to the description of system controller for detailed operation.

The TCC79XX has one chip select for SDRAM, and two chip selects for other type of static memories. Their address space is fixed on the base flag of the configuration registers for each chip selector which is read-only.

The TCC79XX has various peripherals for specific interface controllers or on-chip hardware components. These peripherals can be configured appropriately by its own registers that can be accessed through specially allocated address. These address maps are represented in the following table. In case of memory controller, its space is separated for preventing illegal accessing.

Refer to corresponding sections for detail information of each peripheral.

Base Address	Peripherals
0xF0000000	LCD Controller
0xF0010000	USB2.0 Device Controller
0xF0020000	USB1.1 Host Controller
0xF0030000	HDD(IDE) Controller
0xF0040000	Central DMA Controller for Channel 0,1,2
0xF0040100	Central DMA Controller for Channel 3,4,5
0xF0040200	Central DMA Controller for Channel 6,7,8
0xF0050000	I/O Bus Control and Configuration Registers
0xF0051000	Memory Stick Interface Controller
0xF0052000	I2C Interface Controller
0xF0055000	UART Channel 0, 1, 2, 3
0xF0057000	GPSB channel 0, 1, 2, 3
0xF0059000	DAI Interface Controller
0xF0059080	CDIF Interface Controller
0xF005A000	GPIO Controller
0xF005B000	ECC Calculator
0xF005C000	SPDIF TX Interface Controller
0xF0060000	Camera Interface
0xF0070000	Memory-to-Memory Scaler
0xF0080000	SRAM-like master interface
0xF0090000	SD/MMC Interface Controller
0xF00A0000	External Host Interface Controller Chip Selector 1 (HPCSN_L)
0xF00B0000	NAND Flash Controller
0xF1000000	External Memory Controller Including SDRAM
0xF2000000	Video Hardware Accelerator
0xF3000000	System Manage & Clock Controller
0xF3001000	Priority Vectored Interrupt Controller
0xF3002000	RTC(Real-Time Clock)
0xF3003000	Timer Counters
0xF3004000	8 Channel / 10 Bits ADC
0xF3005000	System Control and Configuration Registers
0xF3006000	Protection Hardware (Special Hardware)
0xF4000000	Mail Box between Main and Sub Processors
0xF5000000	External Host Interface Controller Chip Selector 0 (HPCSN)
0xF6000000	Graphic Engine
0xF7000000	Virtual MMU

3 BUS ARCHITECTURE

3.1 Main Bus Architecture

The Figure 3.1 shows the main bus architecture.

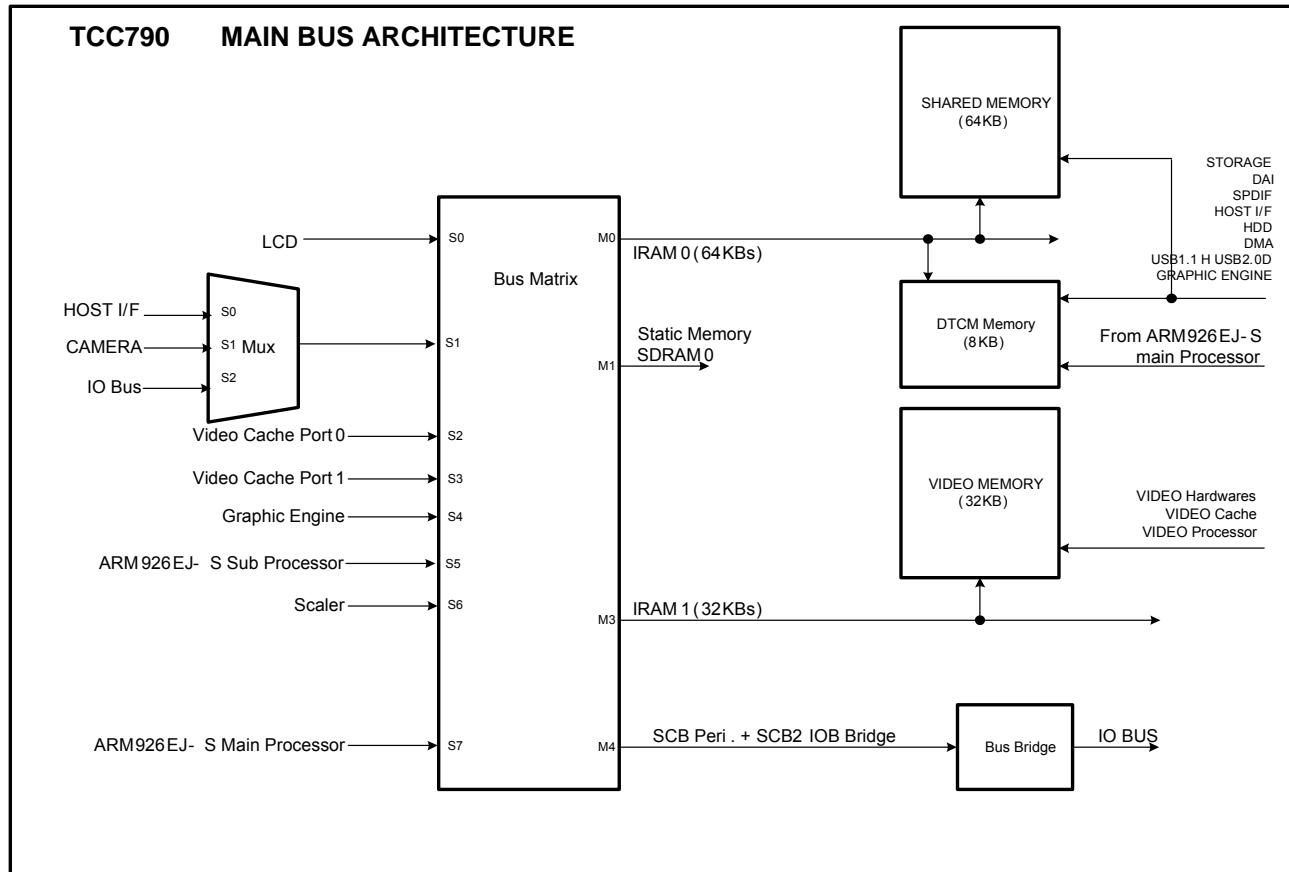


Figure 3.1 The Main Bus Architecture

The main bus matrix has the 8 master ports and 4 slave ports. The highest priority master port is the LCD interface.

The master ports named “Video Cache Port 0” and “Video Cache Port 1” are from the video cache controller.

The memory resources of TCC79XX are divided into 4 groups. The first group (M0) is for internal SRAM with 64KBs shared memory and 8KBs DTCM, which can be accessed directly by various hardwares, for example, USB2.0 device, HDD controller, central DMA controller and etc. When the dedicated hardware accesses the corresponding shared memory, the other hardware including ARM926EJ-S main processor can access other resources simultaneously, which increases the system performance.

The second resource (M1) is for external static memory or SDRAM with shared address and data bus. And the last resource (M4) is for accessing the various on-chip peripherals.

The third resource (M3) is for video memory with 32KBs size, which is divided into 4 shared memory groups. It can be accessed by ARM926EJ-S main processor, ARM926EJ-S sub processor, and various video hardwares.

3.2 Video Hardware Bus Architecture

The Figure 3.2 shows the video hardware bus architecture. The video hardware is composed of various video hardwares and storage devices. The various hardwares are composed of “BIT-STREAM DMA”, “MOTION ESTIMATOR”, “VIDEO DMA”, “VIDEO HARDWARES”, and “SIMD VECTOR PROCESSING ENGINE”. The storage devices are the “VIDEO CACHE” and “VIDEO MEMORY”.

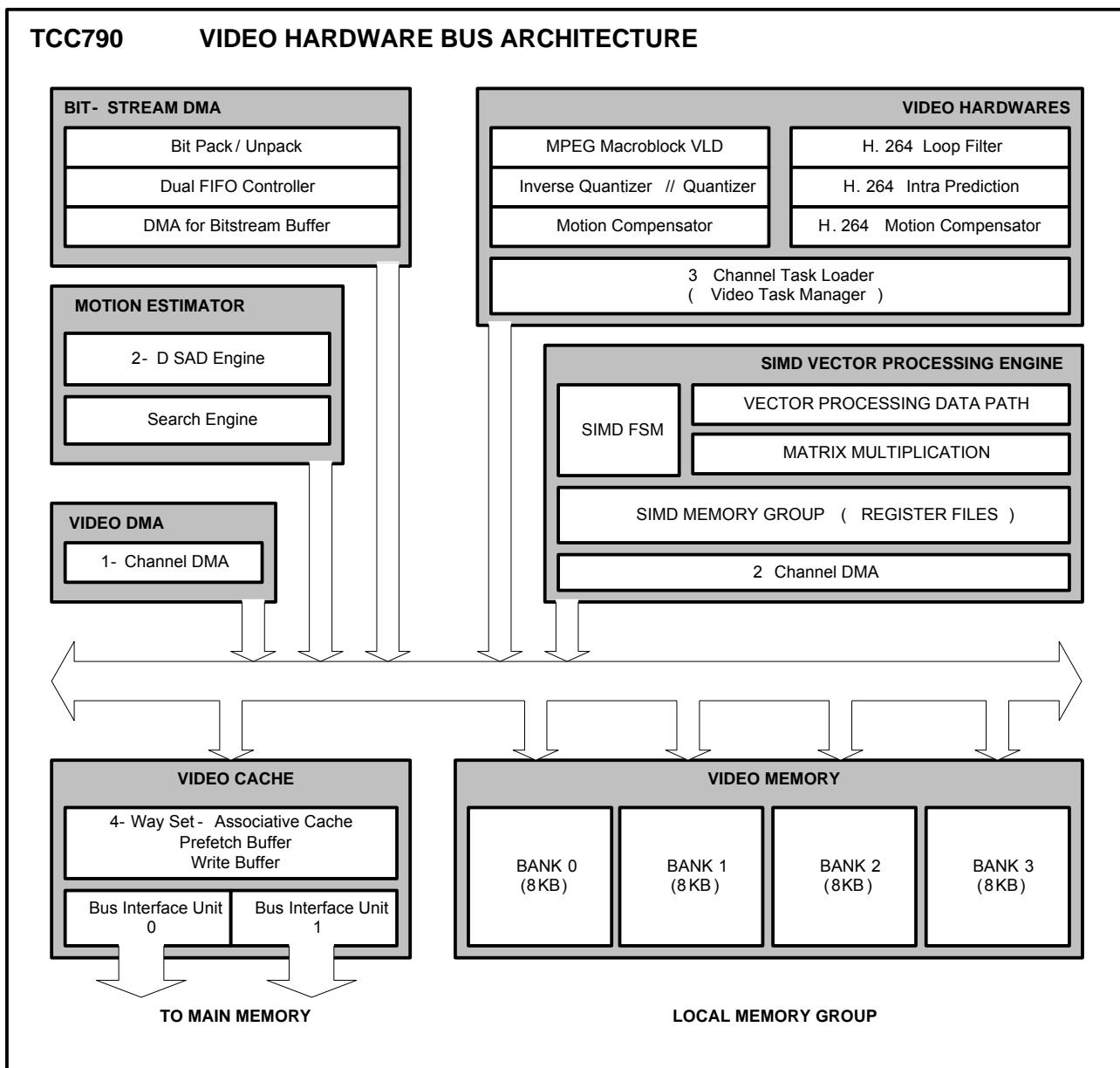


Figure 3.2 The Video Hardware Bus Architecture

The “VIDEO CACHE” is for caching the frame data. This can access the main memory directly, external SDRAM or internal SRAM. This has two bus interface units, one of which is for accessing cacheable region and other for non-cacheable region.

The “VIDEO MEMORY” has the 32KBs on-chip SRAM. The SRAM in “VIDEO MEMORY” is split into 4 banks and the 4 banks can be accessed in parallel by the various hardware.

3.3 I/O Bus Architecture

Figure 3.3 shows the I/O bus architecture.

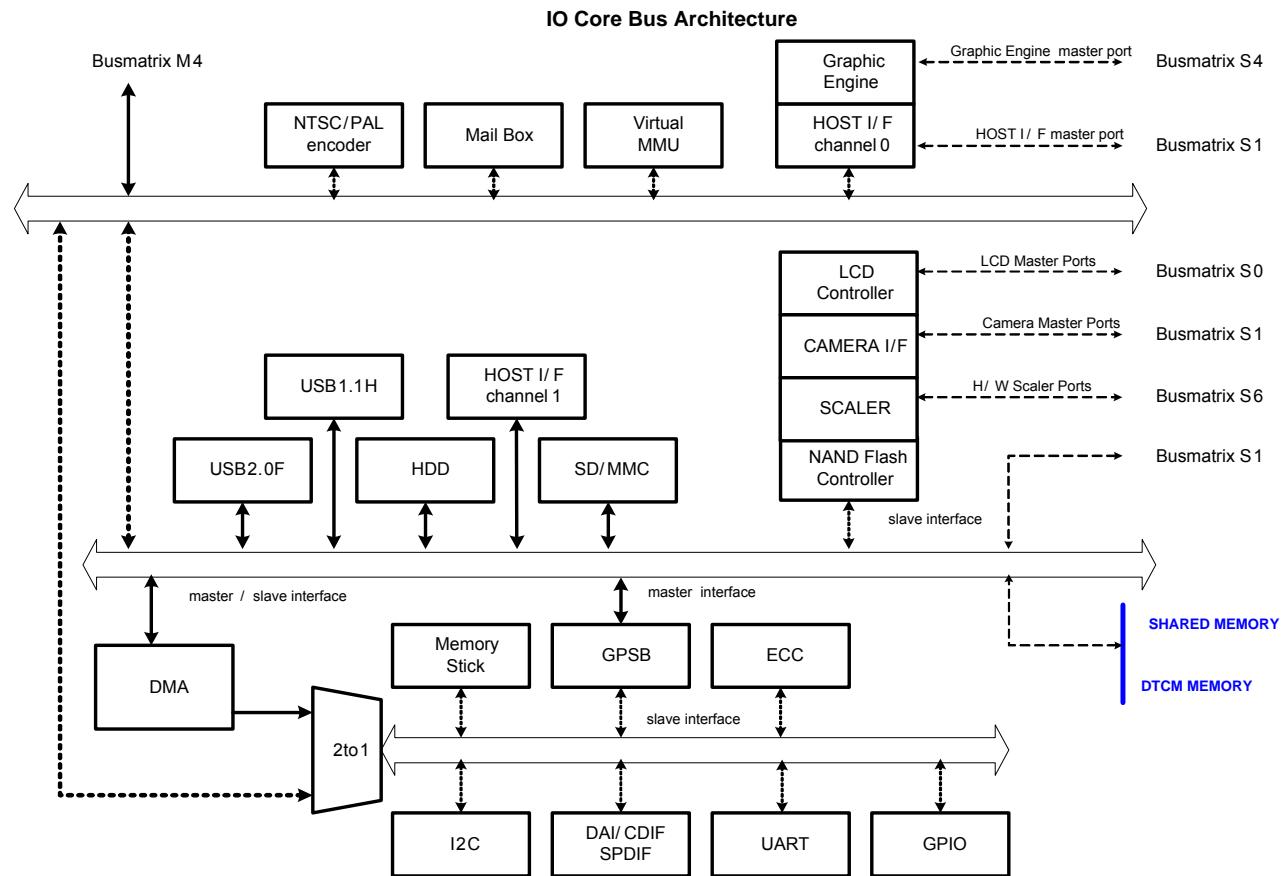


Figure 3.3 The I/O Hardware Bus Architecture

4 ARM926EJ-S MAIN PROCESSOR

4.1 Overview

The TCC79XX has adopted the ARM926EJ-S main processor core for controlling system and processing various kinds of digital signals. It has a Harvard architecture with separate 16Kbyte data and 16Kbytes instruction caches, each with 8-word of line length.

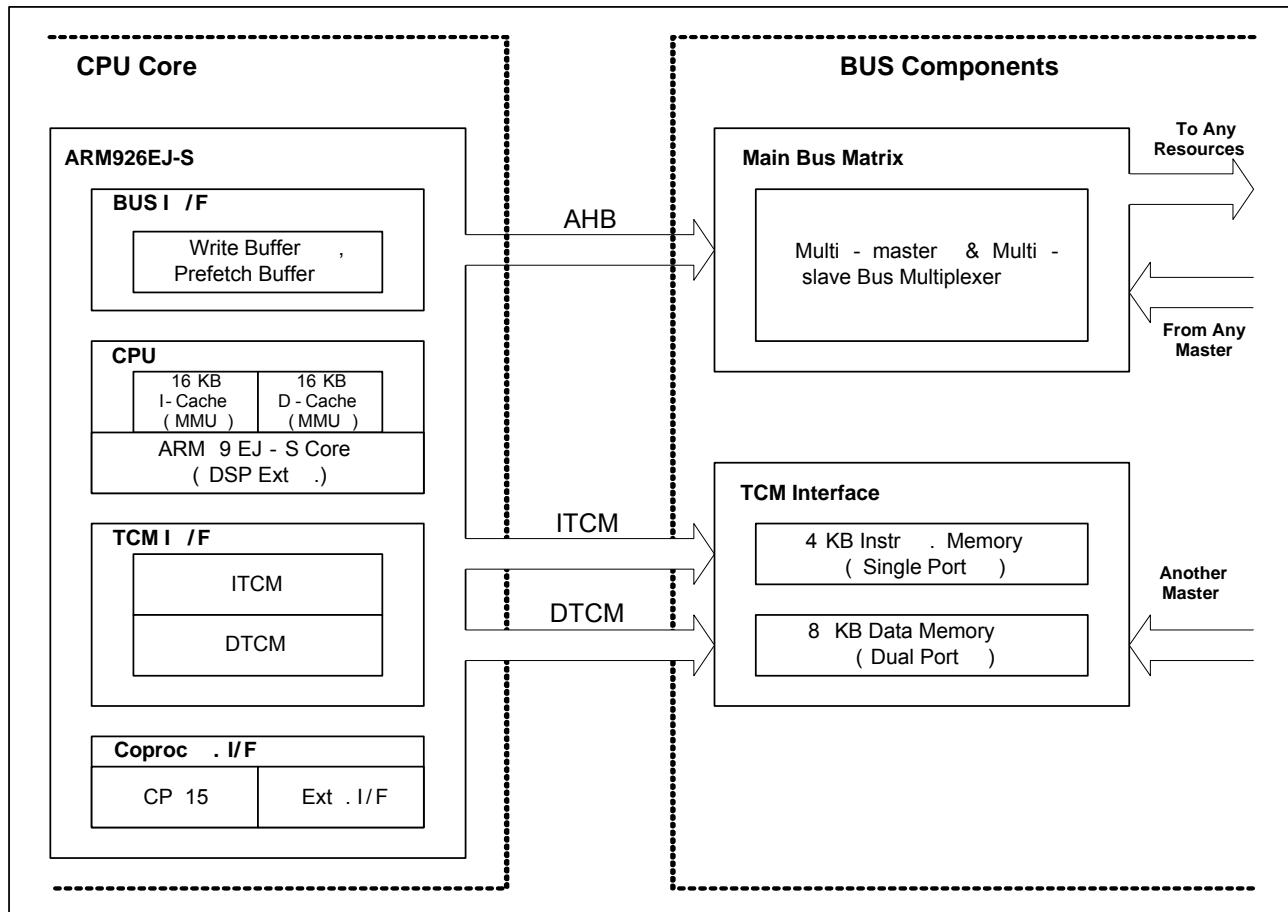


Figure 4.1 The TCC79XX Core Architecture

The ARM926EJ-S CPU core has the 3 interfaces and core features. The BUS I/F makes it possible to read from or write to the various resources. The CPU is composed of ARM9E-S core and instruction/data cache with 16KB. The cache fill or replace transfer is fulfilled through the bus interface unit. The line size of each cache is 8 words, so the burst size of the cache transfer including prefetching the instruction or data is 8 words burst.

The ARM926EJ-S main processor has the special interface different from previous ARM core. The TCM interface is very useful interface to access the external memory fast. The operating frequency of TCM interface is same as CPU core frequency, which means that if you access the TCM memory, the access speed is same as cache operation with 100% hit ratio. In this system, two memory resources are connected to TCM interface. The ITCM memory with 4KB is connected to ITCM interface of the core. The DTCM memory with 8KB is connected to DTCM interface of the core. The ITCM memory is organized by single-port SRAM and the DTCM memory by dual-port SRAM. The ITCM is very useful for exception handler or any other codes called very often such as the basis of OS kernel functions. Also, if you lay up any type of codes which require very high-performance on the ITCM, you can get the higher performance of the corre-

sponding codes.

The DTCM memory is also very useful resource. When you access the DTCM memory, the access speed is same as data cache with 100% hit ratio. And the DTCM has the dual-port memory structure. The CPU can access the resource without any interference of another master hardware. This feature makes it possible for you to communicate with any master hardware with highest access speed.

In this chapter, we describe the key functions to make the good use of the ARM926EJ-S core and TCM such as coprocessor instruction, methods for controlling the cache and TCM. If you want to have the detailed information for the ARM926EJ-S core, refer to the ARM TRM (Technical Reference Manual) r0p5 version.

4.2 Functional Description

4.2.1 Memory Formats

The ARM926EJ-S views memory as a linear collection of bytes numbered upwards from 0. Bytes 0 to 3 hold the first word. In the same way, bytes 4 to 7 hold the second word and so on. ARM926EJ-S itself can treat words in memory as being stored either in big-endian or little-endian, but in the ARM926EJ-S, there is only little-endian supported by the memory controller. The following figure illustrates the structure of little-endian type of memories.

Higher Address	31	24 23	16 15	8 7	0	Word Value
		0x99	0x66	0xAA	0x55	8 0x9966AA55
		0xFE	0xDC	0xBA	0x98	4 0xFEDCBA98
		0x76	0x54	0x32	0x10	0 0x76543210

Higher Address Lower Address Higher Address Lower Address

Figure 4.2 Little-Endian Addresses of Bytes-Words

4.2.2 Operating Modes

ARM926EJ-S supports seven modes of operation:

- USER (usr) The normal ARM program execution mode
- FIQ (fiq) Designed to support a data transfer or channel process
- IRQ (irq) Used for general purpose interrupt handling
- Supervisor (svc)P Protected mode for the operating system
- Abort (abt) Entered after a data or instruction prefetch abort
- System (sys) A privileged user mode for the operating system
- Undefined (und) Entered when an undefined instruction is executed

Switching between these modes may be made under software control, or may be brought about by interrupts or exception processing. Most application programs will execute in User mode. The non-user mode known as privileged modes are entered in order to service interrupts or exceptions, or to access protected resources.

4.2.3 Instruction Set Features

ARM Instruction Set

The ARM instruction set contains the following six main instruction types.

- Branch instructions

- Data processing instructions
- Status register transfer instructions
- Load/store instructions
- Cache/Protection instructions
- Exception generation instructions

Most data processing instructions update the four condition code flags (N, Z, C, V) in the current program status register (CPSR).

4.2.4 Addressing Modes

Load/Store Instructions

Load/store instructions provide three basic types of addressing modes. The addressing mode is formed from two parts; the base register and the offset.

Offset addressing mode The memory address consists of the offset added to or subtracted from the base register contents

Preindex addressing mode The memory address consists of the offset added to or subtracted from the base register contents. This new address is then overwritten to the base register.

Postindex addressing mode The memory address is the base register contents. The hardware then updates the base register by adding or subtracting the offset.

The offset takes one of three formats which are “immediate”, “register”, and “scaled register”. Immediate offset is unsigned number that can be added to or subtracted from the base register. The register offset is a general-purpose register that can be to or subtracted from the base register. The scaled register offset is a general-purpose register shifted by an immediate value.

Multiple Load/Store Instructions

Multiple load/store instructions load/store two or more general-purpose registers from/to memory. These instructions offer four addressing modes.

Increment After (IA) The base register contents are incremented after the transfer.

Increment Before (IB) The base register contents are incremented before the transfer.

Decrement After (DA) The base register contents are decremented after the transfer.

Decrement Before (DB) The base register contents are decremented before the transfer.

4.2.5 Exceptions

Exceptions arise whenever the normal flow of a program has to be halted temporarily, for example to service an interrupt from a peripheral. Before an exception can be handled, the current processor state must be preserved so that the original program can resume when the handler routine has finished.

It is possible for several exceptions to arise at the same time, and if this happens, they are dealt with in a fixed priority. This priority is illustrated in Table 4.1.

Table 4.1 Exception Vector and Priorities

Address	Exception	Mode in Entry	Priority
0x00000000	Reset	Supervisor	1
0x00000004	Undefined Instruction	Undefined	6
0x00000008	Software Interrupt	Supervisor	6
0x0000000C	Abort (Prefetch)	Abort	5
0x00000010	Abort (Data)	Abort	2
0x00000014	Reserved	Reserved	-
0x00000018	IRQ	IRQ	4
0x0000001C	FIQ	FIQ	3

- The higher priority has the lower number.

Actions on Entering an Exception

When handling an ARM exception the ARM926EJ-S core

- (1) Preserves the address of the next instruction in the link register of appropriate mode. If the exception has been entered from ARM state, then the address of the next instruction is copied into the link register (current PC + 4, or +8 depending on the exception types). If the exception has been entered from THUMB state, then the value written into the link register is the current PC offset by a value such that the program resumes from the correct place on return from the exception. The exception handler does not need to determine the state when entering an exception. For example, in case of SWI exception, MOVS PC, R14_svc will always return to the next instruction regardless of whether the SWI was executed in ARM or THUMB state
- (2) Copies the CPSR into the SPSR of appropriate mode
- (3) Forces the CPSR mode bits to a value that represents the mode of exception
- (4) Forces the PC to fetch the next instruction from the relevant exception vector

It may also set the interrupt disable flags to prevent otherwise unmanageable nesting of exceptions. If the processor is in THUMB state when an exception occurs, it will automatically switch into ARM state when the PC is loaded with the exception vector address.

Actions on Leaving an Exception

When an exception has completed, the exception handler must move the link register, minus an offset to the PC. The offset will vary depending on the type of exception.

If the S bits set rd=r15 (MOVS r15, r?), the ARM926EJ-S copies the SPSR back to the CPSR.

An explicit switch back to THUMB state is never needed, since restoring to the CPSR from the SPSR automatically sets the T bit to the value it held immediately prior to the exception.

Exception Entry/Exit Summary

The following table summarizes the PC value preserved in the relevant R14 on exception entry, and the recommended instruction for exiting the exception handler.

Table 4.2 Exception Entry/Exit

Entry	Return Instruction	Previous State		Notes
		ARM	THUMB	
SWI	MOVS PC, R14_svc	PC + 4	PC + 2	Where the PC is the address of the SWI or undefined instruction.
UNDEF	MOVS PC, R14_und	PC + 4	PC + 2	
FIQ	SUBS PC, R14_fiq, #4	PC + 4	PC + 4	Where the PC is the address of the instruction that was not executed because the FIQ or IRQ took priority.
IRQ	SUBS PC, R14_irq, #4	PC + 4	PC + 4	
PABT	SUBS PC, R14_abt, #4	PC + 4	PC + 4	Where the PC is the address of instruction that had the Prefetch Abort.
DABT	SUBS PC, R14_abt, #8	PC + 8	PC + 8	Where the PC is the address of the Load or Store instruction that generated the Data Abort.
RESET	-			The value saved in R14_svc upon reset is unpredictable.

FIQ

The FIQ (Fast Interrupt Request) exception is designed to support a data transfer or channel process, and in ARM state, it has sufficient private registers to remove the need for saving registers (thus minimizing the overhead of context switching).

FIQ is generated by nFIQ signal that is controlled by internal or external interrupts defined as FIQ type. Irrespective of whether the exception was entered from ARM state or THUMB state, a FIQ handler should leave the interrupt by executing

SUBS PC, R14, #4

FIQ may be disabled by setting the CPSR's F flag (this is not possible in USER mode). If the F flag is clear, ARM926EJ-S checks for a LOW level on the nFIQ signal at the end of every instruction.

IRQ

The IRQ (Interrupt Request) exception is a normal interrupt caused by a LOW level on the nIRQ signal that is controlled by internal or external interrupt defined as IRQ type. IRQ has lower priority than FIQ and is masked out when a FIQ sequence is entered. It may be disabled at any time by setting the I bit in the CPSR, though this can only be done in a privileged (none-User) mode.

Irrespective of whether the exception was entered from ARM state or THUMB state, an IRQ handler should return from the interrupt by executing

SUBS PC, R14, #4

Abort

An abort indicates that the current memory access cannot be completed. ARM926EJ-S checks for the abort condition during memory access cycles.

There are two types of abort:

- Prefetch abort: occurs during an instruction prefetch
- Data abort: occurs during a data access

If a prefetch abort occurs, the prefetched instruction is marked as invalid, but the ex-

ception will not be taken until the instruction reaches the head of the pipeline. If the instruction is not executed, the abort does not take place.

If a data abort occurs, the action taken depends on the instruction type:

The swap instruction (SWP) is aborted as if it had not been executed.

Block data transfer instructions (LDM, STM) complete. If write-back is set, the base is updated. If the instruction would have overwritten the base with data, the overwriting is prevented. All register overwriting is prevented after an abort is indicated, which means in particular that R15 is preserved in an aborted LDM instruction.

After fixing the reason for the abort, the handler should execute the following irrespective of the state (ARM or THUMB):

SUBS PC, R14, #4	; for a prefetch abort
SUBS PC, R14, #8	; for a data abort

This restores both the PC and the CPSR, and retries the aborted instruction.

Software Interrupt

The software interrupt instruction (SWI) is used for entering Supervisor mode, usually to request a particular supervisor function. A SWI handler should return by executing the following irrespective of the state (ARM or THUMB):

MOV PC, R14

This restores the PC and CPSR, and returns to the instruction following the SWI instruction.

Undefined Instruction

When ARM926EJ-S comes across an instruction that it cannot handle, it takes the undefined instruction trap. This mechanism may be used to extend either the THUMB or ARM instruction set by software emulation.

After emulating the failed instruction, the trap handler should execute the following instruction irrespective of the state (ARM or THUMB):

MOVS PC, R14

This restores the CPSR and returns to the instruction following the undefined instruction.

4.3 Coprocessor CP15

Refer to ARM's technical reference manual.

5 LCD INTERFACE

5.1 Overview

The LCD interface (LCDC) is used to send out image data from system memory to LCD (RGB interface type) or NTSC/PAL encoder by properly formatting the raw image data stored in the memory. The LCDC provides all the necessary control signals to interface directly to mono STN, color STN, TFT panels, and NTSC/PAL encoders.

The features of the LCDC are as follows.

- supports Thin Film Transistor(TFT) color displays with 8-bit, 16-bit, 18-bit, 24-bit interface
- supports STN displays with 4 or 8-bit interface
- 1, 2 or 4 bits per pixel(bpp) displays for mono STN
- 8(332) /16 bpp(444dummy) color displays for color STN
- 16, 18, 24 bpp true-color non-palettized color displays for color TFT
- resolution programmable up to 1024 * 1024
- programmable timing for different display panels
- NTSC/PAL digital video encoder interface (CCIR601/656 interface)
- Supports color lookup table
- Supports the overlay and alpha blending (2 overlay and 1 original image)
- Supports the image up/down scaling (x2, x3, x4, x8)
- Supports the Picture-In-Picture using the picture from the external device

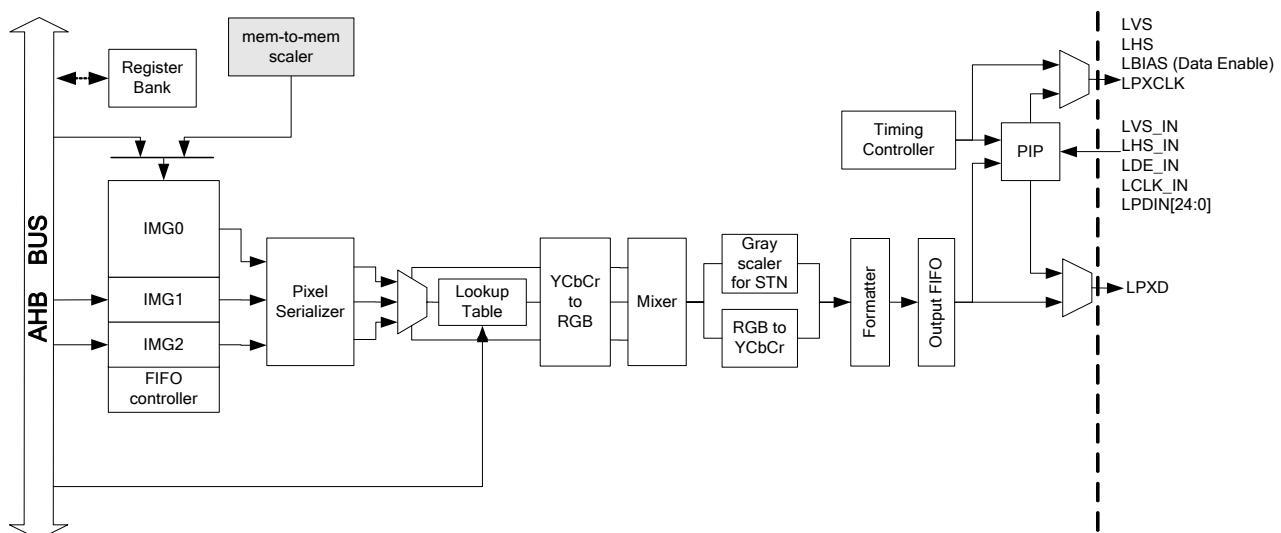


Figure 5.1 LCD controller Block Diagram

The following key parameters can be programmed:

- horizontal front and back porch
- horizontal synchronization pulse width
- number of panel clocks per line
- vertical front and back porch
- vertical synchronization pulse width
- vertical synchronization pulse delay for STN mode
- number of lines per panel
- signal polarity
- panel clock frequency
- AC panel bias

- bits-per-pixel
- display type, STN mono/color or TFT
- STN 4 or 8-bit interface mode
- NTSC/PAL, Interlace/Non-interlace mode
- Overlay/alpha blending mode
- Image up/down scale ratio

The raw image sources stored in frame buffer are transferred to the LCDC's input FIFO, on a demand basis, using the AMBA AHB master interface.

The LCDC starts the DMA data transfer after it has been initialized and enabled. The DMA automatically performs burst word (32-bit) transfers, filling the empty entries of the input FIFO. The data in the FIFO are fetched into one entry at a time, and each 32-bit data is unpacked into appropriate pixel data formats according to the raw image data format information. The frame buffer is in an off-chip memory area used to supply enough encoded pixel values to fill the entire screen one or more times. The pixel data buffer contains one encoded pixel values for each of the pixels present on the screen. The number of pixel data values depends on the size of the screen. The LCDC can fetch up to three raw image sources simultaneously and mixes them for making one frame image.

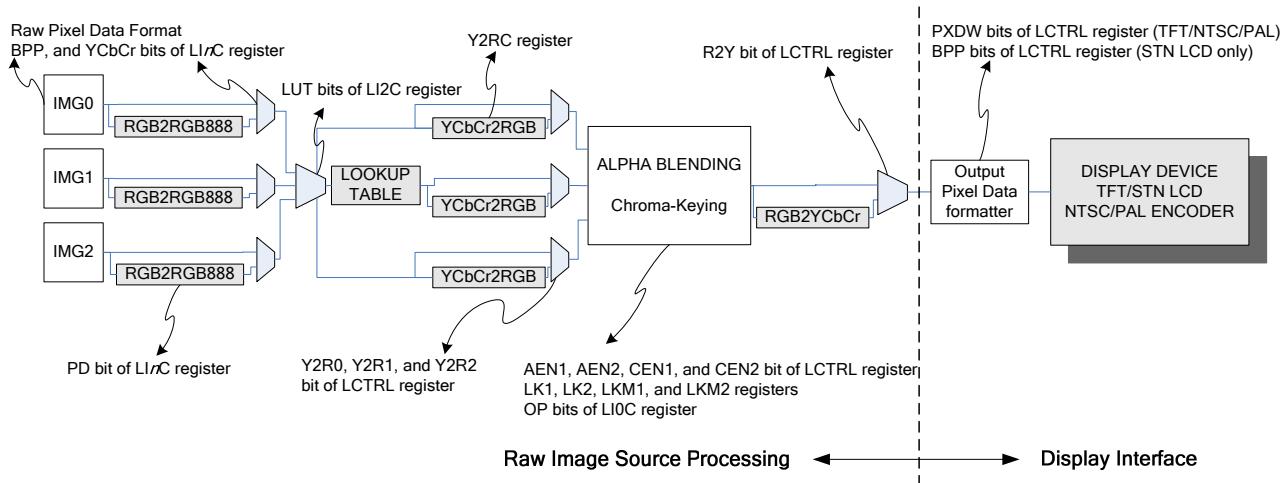


Figure 5.2 Overall Image Data Flow

Figure 5.2 shows the overall image data flow in the LCDC.

5.2 Raw Image Source Processing

Specifying Raw Image Sources

For processing the raw image sources, the LCDC defines a virtual display. It is shown in Figure 5.3. Display size, which consists of “Display Width” and “Display Height”, is determined by LDS register. Actually, it becomes the active display size of the connected LCD or NTSC/PAL encoder.

The LCDC can have up to 3 raw image sources, which are named as IMG0, IMG1, and IMG2. IEN0, IEN1, and IEN2 bit of LCTRL register are used for enabling the corresponding raw image source. It can display simultaneously up to 3 layers, which are named as LAYER0, LAYER1, and LAYER2. LAYER2 is the top layer. They are allocated for IMG0, IMG1, and IMG2 respectively.

The area which is not overlapped by layers is filled with the background color. It is determined by the LBCLBC register

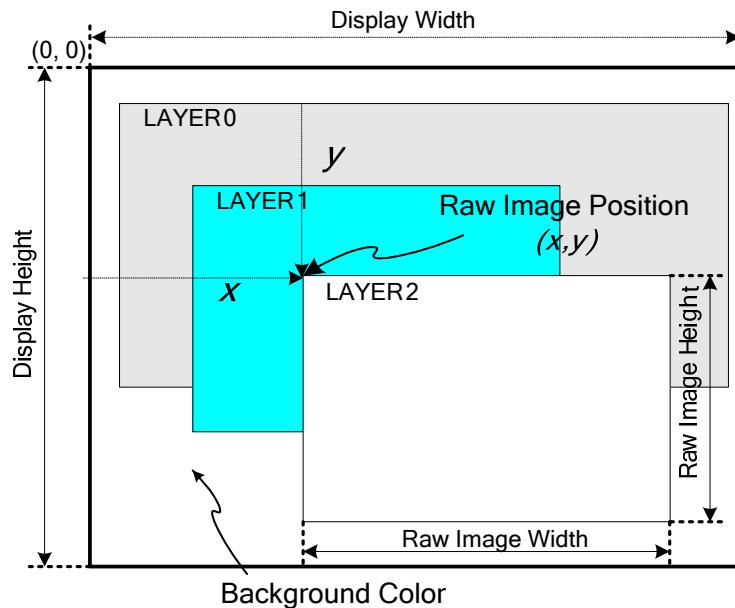


Figure 5.3 Virtual Display in the LCDC

The properties of a layer are determined by those of the corresponding raw image source; size, pixel data format, base address, position in the virtual display, etc.

1BPP	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	p31	p30	p29	p28	p27	p26	p25	p24	p23	p22	p21	p20	p19	p18	p17	p16
	2BPP	p15	p14		p13	p12		p11	p10		p9		p8			
	4BPP		p7		p6		p5		p4							
1BPP	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	p15	p14	p13	p12	p11	p10	p9	p8	p7	p6	p5	p4	p3	p2	p1	p0
	2BPP	p7	p6		p5	p4		p3	p2		p1		p0			
	4BPP		p3		p2		p1		p0							
BR=0																
1BPP	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	p24	p25	p26	p27	p28	p29	p30	p31	p16	p17	p18	p19	p20	p21	p22	p23
	2BPP	p12	p13		p14	p15		p8		p9	p10					
	4BPP		p6		p7		p4		p5							
1BPP	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	p8	p9	p10	p11	p12	p13	p14	p15	p0	p1	p2	p3	p4	p5	p6	p7
	2BPP	p4	p5		p6	p7		p0		p1	p2		p2		p3	
	4BPP		p2		p3		p0		p1							
BR=1																
RGB332	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R1[2:0]			G1[2:0]			B1[1:0]			R0[2:0]			G0[2:0]			B0[1:0]
RGB444	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	alpha[3:0]				R[3:0]				G[3:0]				B[3:0]			
RGB565	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R[4:0]					G[5:0]					B[4:0]					
RGB555	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	R[4:0]				G[4:0]				B[4:0]						
RGB666	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IGNORED															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R[3:0]				G[5:0]				B[5:0]							
RGB888	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IGNORED				alpha*[3:0]				R[7:0]							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	G[7:0]								B[7:0]							
Sequential YCbCr4:2:2	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Cr0[7:0]								Y1[7:0]							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cb0[7:0]								Y0[7:0]							
.....																
YCbCr4:2:2 YCbCr4:2:0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cb1[7:0]								Cb0[7:0]							
							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cr1[7:0]								Cr0[7:0]								

Figure 5.4 Supported Pixel Data Format (BPP bits of LinC register)

The pixel-data formats supported in the LCDC are shown in Figure 5.4. They are determined by the LI_{nC} registers ($n=0, 1, \text{ or } 2$). $LI0C$, $LI1C$, and $LI2C$ are for $IMG0$, $IMG1$, and $IMG2$ respectively. 1bpp , 2bpp , and 4bpp have no color information and can be arranged in big-endian pixel order as well as little-endian pixel order. Others have color information and should be arranged in little-endian pixel order only.

IMG1 and IMG2 do not support YCbCr4:2:0 and YCbCr4:2:2, but support “sequential YCbCr4:2:2”. If the pixel data format of a raw image source is YCbCr4:2:0 or YCbCr4:2:2, it needs three base addresses, which are specified by the LInBA0, LInBA1, and LInBA2 register. Otherwise, it needs one base address, which is specified by the LInBA0 register. Thus, when the pixel data format of IMG0 is YCbCr4:2:0 or YCbCr4:2:2, the base address of Y, Cb, and Cr image should be set to LIOBA0, LIOBA1, and LIOBA2 respectively. But, because IMG1 and IMG2 do not support YCbCr4:2:0 and YCbCr4:2:2, there are only L11BA0 and L12BA0 register.

Though a raw image source is not RGB888 format in RGB color space, it is internally converted to RGB888 as appending additional bits. When this conversion is occurred, the PD bits of LInC determine whether the additional bits are filled with zero or MSB. The difference between them is shown in Figure 5.5.

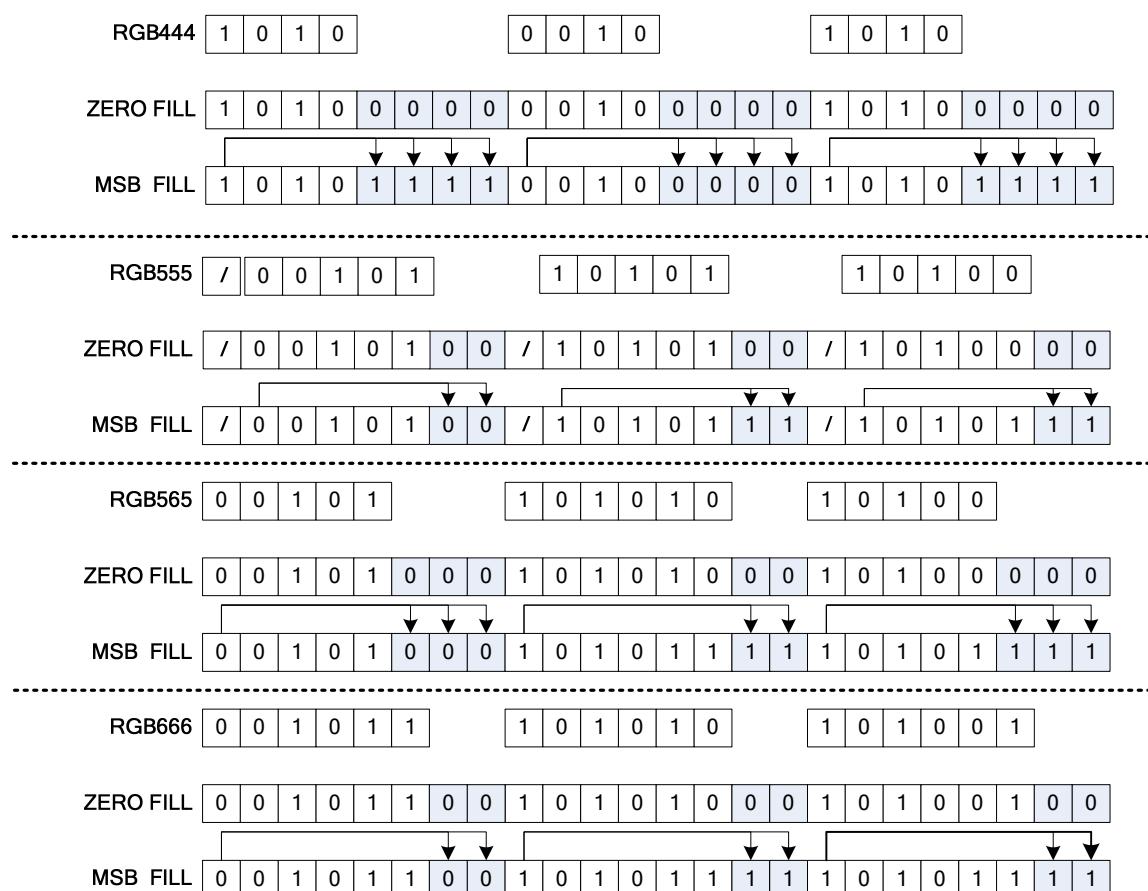


Figure 5.5 RGB2RGB888 conversion

The origin of the virtual display is the top-left corner ((0, 0) in Figure 5.3) and “Raw Image Position” is the location of the top-left corner of a raw image within the virtual display. It is determined by the LInP registers. “Raw Image Size”, which consists of “Raw Image Width” and “Raw Image Height”, is determined by the LInS registers.

And the LCDC has a simple scaler for each raw image source. It is controlled by LInSC registers. But, to get the high-quality image, SCALER should be used instead of it. Refer to “Memory To Memory Scaler” on page 7-1 for more information about the image scaling.

Usually, one frame of a raw image source is in the continuous address space. But, when “Raw Image Width” is 32-bit aligned pixel data in the frame buffer, “32bits * n memory space” can be added to the end of pixel line and this space is not displayed. n should be 0 or natural number. This “32bits * n memory space” is called the offset and it is controlled by LInO registers. Figure 5.6 shows how to use the offset.

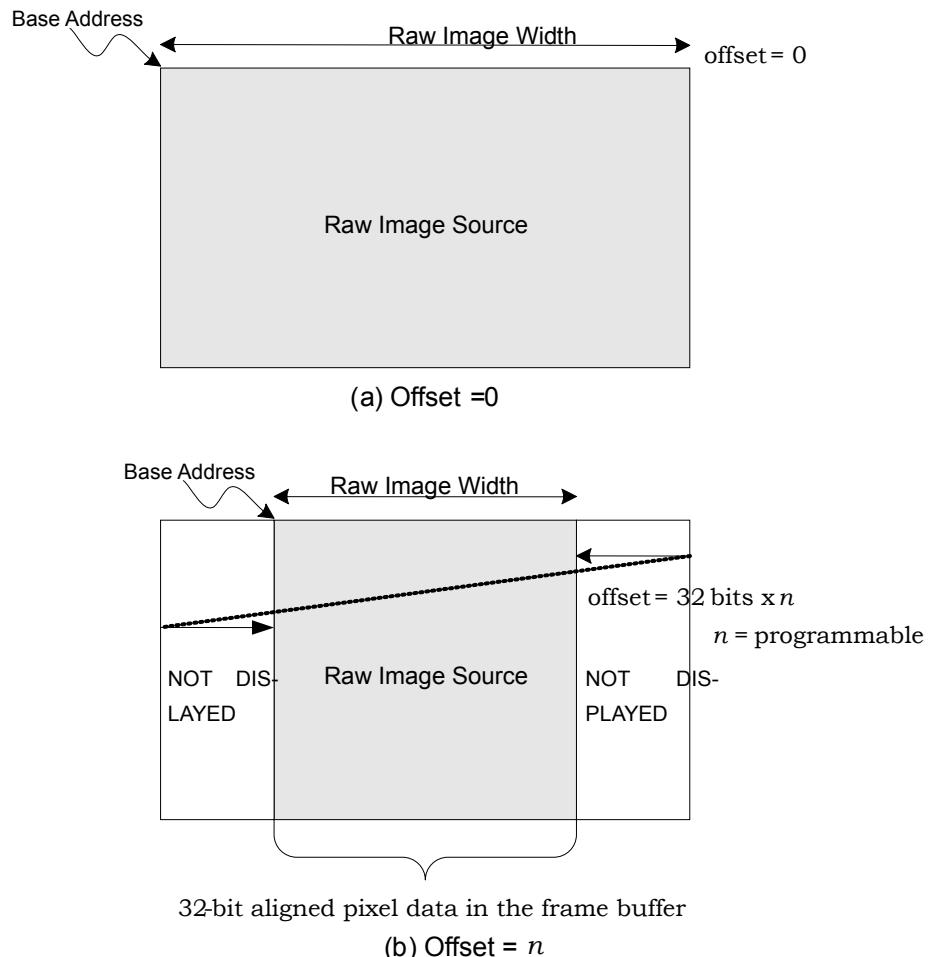


Figure 5.6 Raw Image Offset

YCbCr2RGB conversion

The LCDC has YCbCr to RGB color space converters for raw image sources. They can be enabled by Y2R0, Y2R1, and Y2R2 bit of LCTRL register; they are for IMG0, IMG1, and IMG2 respectively. Each converter can be one of 4 types; it is controlled by Y2R0M, Y2R1M, and Y2R2M bits of Y2RC register and they are for IMG0, IMG1, and IMG2 respectively. Notice that all raw image sources should be in the same color space, which is RGB space or YCbCr space, before doing alpha-blending and chroma-keying operation. Therefore, if raw image sources are in the different color space, these color space converter should be used.

Look-Up Table

The LCDC has the “24 bits x 256 entries” look-up table and it can be connected to one of IMG0, IMG1, and IMG2. Inputs to this table are RGB888 or YCbCr444 format as shown Figure 5.2. Each input pixel data addresses the table entry and the new pixel

value is the value in the corresponding entry. Its entry consists of three 8-bit color values. The overall structure of Look-Up Table is shown in Figure 5.7. This table is controlled by LUT bits of the LI2C register.

The table entries should be initialized before the LCDC uses it. Their address space is from 0xF0000C00 to 0xF000FFC and 32-bit write-only region.

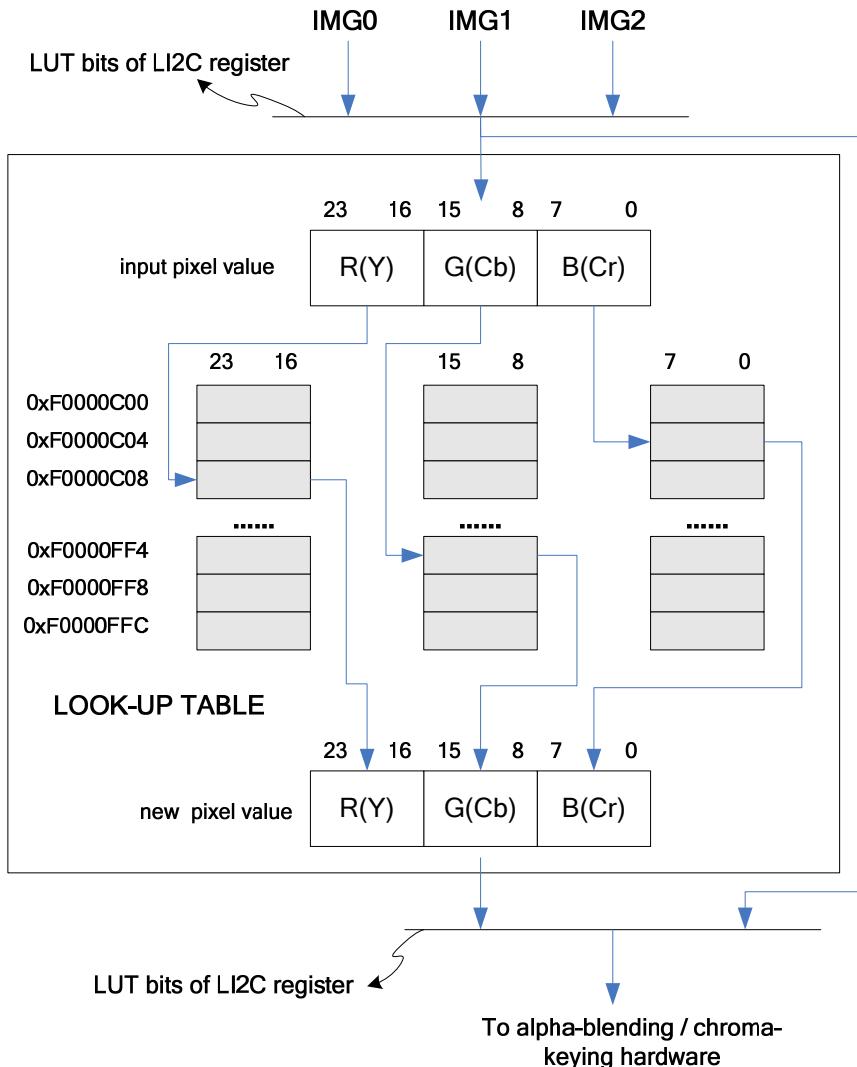


Figure 5.7 LCD Color Lookup Table

Alpha-blending

To process the overlapped region between layers, the LCDC supports the alpha-blending and the chroma-keying. When the alpha-blending and the chroma-keying are disabled, the top layer image is only shown in the overlapped region. Thus, LAYER1 is only shown in the overlapped region between the LAYER0 and the LAYER1.

The alpha-blending combines the overlay image (top layer) with the main image (bottom layer) according to the alpha value. The alpha value is specified by alpha bits in the pixel data of the overlay image and the alpha registers. The AEN2 bit of LCTRL register determines whether the LAYER2 enables the alpha-blending and the AEN1 bit determines whether LAYER1 enables the alpha-blending. Whether the alpha bits in the pixel data or alpha-registers are used as the alpha-value depends on the value of the SA1 bit of the LK1 register for the LAYER1 and the SA2 bit of the LK2 register for the LAYER2.

If the overlay image consists of the pixel data without the alpha value, you should use alpha-registers as the alpha-value (SA2 = 1 and/or SA1 = 1). The LAYER2 uses the A20 bits of LK2 register as the alpha-value and LAYER1 uses the A10 bits of LK2 register.

Chroma-Keying

Usually, a raw image source is the rectangular shape. Chroma-keying is used for clipping the arbitrary shape from a raw image source. In other words, chroma-keying disables the overlay process for selected pixels. They are identified by control registers, which are LK1, LK2. The LK1 is for the LAYER1 and the LK2 is for the LAYER2.

The overlay disabled pixel becomes the transparent pixel; the pixel of background layer is shown in the display.

For LAYER1, whether a pixel is overlayed or not depends on

$$\begin{aligned} OverlayDisabledPixel = & ((Pixel(RorY) \& LKM1.MKR1) == (LK1.KR1 \& LKM1.MKR1)) \& \\ & ((Pixel(GorU) \& LKM1.MKG1) == (LK1.KG1 \& LKM1.MKG1)) \& \\ & ((Pixel(BorV) \& LKM1.MKB1) == (LK1.KB1 \& LKM1.MKB1)) \end{aligned}$$

$$\begin{aligned} OverlayEnabledPixel = & ((Pixel(RorY) \& LKM1.MKR1) != (LK1.KR1 \& LKM1.MKR1)) \mid \\ & ((Pixel(GorU) \& LKM1.MKG1) != (LK1.KG1 \& LKM1.MKG1)) \mid \\ & ((Pixel(BorV) \& LKM1.MKB1) != (LK1.KB1 \& LKM1.MKB1)) \end{aligned}$$

For LAYER2, whether a pixel is overlayed or not depends on

$$\begin{aligned} OverlayDisabledPixel = & ((Pixel(RorY) \& LKM2.MKR1) == (LK2.KR1 \& LKM2.MKR1)) \& \\ & ((Pixel(GorU) \& LKM2.MKG1) == (LK2.KG1 \& LKM2.MKG1)) \& \\ & ((Pixel(BorV) \& LKM2.MKB1) == (LK2.KB1 \& LKM2.MKB1)) \end{aligned}$$

$$\begin{aligned} OverlayEnabledPixel = & ((Pixel(RorY) \& LKM2.MKR1) != (LK2.KR1 \& LKM2.MKR1)) \mid \\ & ((Pixel(GorU) \& LKM2.MKG1) != (LK2.KG1 \& LKM2.MKG1)) \mid \\ & ((Pixel(BorV) \& LKM2.MKB1) != (LK2.KB1 \& LKM2.MKB1)) \end{aligned}$$

RGB2YCbCr Conversion

When raw image sources are processed in RGB color space during alpha-blending and chroma-keying operation, the output image of them is also in RGB color space. This image can be converted to YCbCr color space as using the R2Y bit of LCTRL register for sending it out to the internal or the external NTSC/PAL encoder.

Picture-In-Picture

The LCDC displays a picture which is generated in the on-chip and a picture from the external display controller simultaneously.

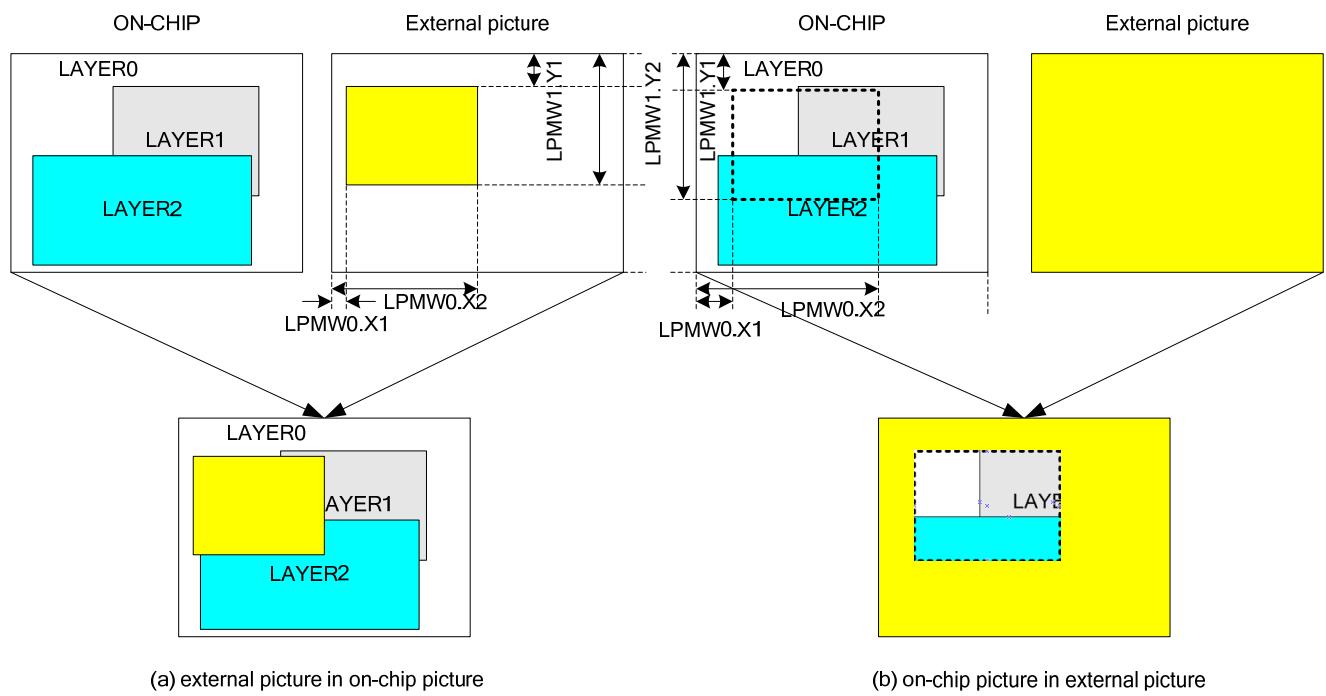
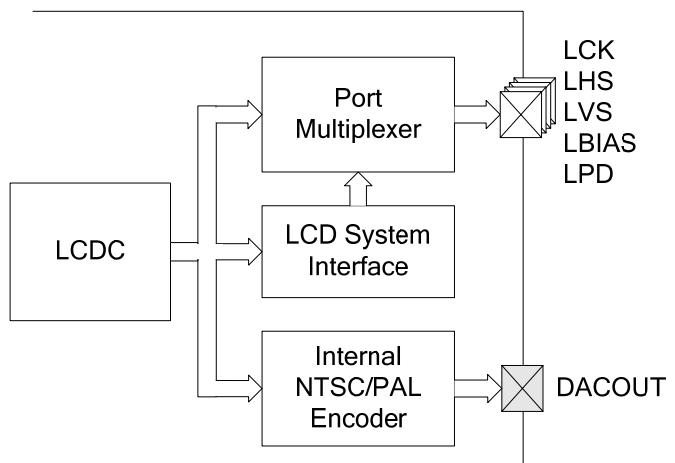
**Figure 5.8 PIP in the LCDC**

Figure 5.8 shows how to specify pictures. The on-chip picture becomes the background (Figure 5.8 (a)) or the foreground (Figure 5.8 (b)). Notice that the active display size of the on-chip LCDC should be the same as the active display size of the external display controller.

5.3 Display Interface

The LCDC drives VSYNC, HSYNC, PXCLK, ACBIAS (or Data Enable) and PCDATA signal. When the LCDC is enabled, these signals are broadcasted to LCD System Interface (Refer to 6. LCD system interface), Internal NTSC/PAL encoder (Refer to 7. NTSC / PAL Encoder Composite Output), and Port Multiplexer (Refer to 33. Port Multiplexer & GPIO). Therefore, the corresponding function block should process these signals. For example, when the LCDC is used for the internal NTSC/PAL encoder, the Port Multiplexer may need to change the function of LVS, LHS, LCK, LBIAS, and LPD pins for preventing NTSC/PAL encoder related signals from being output to them (e.g., the function of these pins can be set to GPIO).

**Figure 5.9 Connection of LCDC output signals**

5.3.1 STN-LCD

The LCDC generates VSYNC, HSYNC, PXCLK, ACBIAS, and PXDATA signals for STN LCD driver.

The output data format for STN LCD is determined by the interface data width and the color depth of STN LCD. The interface data width is determined by PXDW bits of LCTRL register. 4bits and 8bits are only supported for STN LCD. The color depth is BPP bits of LCTRL register and it is not related to a raw image pixel data format.

The timing diagram for STN mode is shown in Figure 5.10. VSYNC and HSYNC pulse are controlled by the configurations of the LPC field of LHTIME and FLC field of LVTIME1 and LVTIME2. Each field is related to the LCD size and display mode.

In 1bpp, 2bpp, 4bpp:

$$LPC = (HorizontalDisplaySize) / (PixelDataWidth) - 1$$

In 8pp and 16bpp (RGB):

$$LPC = (3 \times HorizontalDisplaySize) / (PixelDataWidth) - 1$$

Pixel data width is determined by PXDW of LCTRL register. In the case of STN LCD mode, it must be 4 or 8-bit width.

ACBIAS signal is used by an LCD driver in the STN LCD module to alternate the polarity of the row and column voltage used to turn the pixel on and off. It is controlled by the ACDIV field of LCLKDIV register:

$$f_{ACBIAS} = f_{HSYNC} / (2 \times ACDIV)$$

PXCLK frequency is determined by the CLKDIV field of LCLKDIV register as follows. The minimum value of CLKDIV is 3 in STN mode.

$$f_{PXCLK} = f_{LCLK} / (2 \times CLKDIV) \quad (1)$$

VSYNC frequency is related to the field of FPW, LSWC, LEWC, LPC, and FLC as well as HCLK and PXCLK.

$$f_{VSYNC} = f_{PXCLK} / [(LPW+1) + (LPC+1) + (LSWC+1) + (LEWC+1) \times (FLC+1) + (FPW+1)]$$

Therefore, if FR is the required refresh rate, fPXCLK_REQ, which is the required PXCLK, is as following.

$$f_{PXCLK_REQ} = FR \times [(LPW+1) + (LPC+1) + (LSWC+1) + (LEWC+1) \times (FLC+1) + (FPW+1)] \quad (2)$$

The LCDC contains look-up registers for STN LCD with RGB332 or 2BPP to support palletized color. They are LLUTR, LLUTG, and LLUTB. When RGB332 STN LCD is used, LLUTR register allows any 8 red levels to be selected from 16 possible red levels. LLUTG register is for green color and LLUTB register is for blue color. But, when 2BPP STN LCD is used, LLUTG register is only used. Refer to Table 5.1. When STN LCD type is not RGB332 or 2BPP, look-up registers are not used. STN LCD type is determined by BPP bits of LCTRL register.

Table 5.1 STN LCD Palette Address

Pixel Value	Red Palette	Green Palette	Blue Palette
0	LLUTR[3:0]	LLUTG[3:0]	LLUTG[3:0]
1	LLUTR[7:4]	LLUTG[7:4]	LLUTG[7:4]
2	LLUTR[11:8]	LLUTG[11:8]	LLUTG[11:8]
3	LLUTR[15:12]	LLUTG[15:12]	LLUTG[15:12]
4	LLUTR[19:16]	LLUTG[19:16]	N/A
5	LLUTR[23:20]	LLUTG[23:20]	N/A
6	LLUTR[27:24]	LLUTG[27:24]	N/A
7	LLUTR[31:28]	LLUTG[31:28]	N/A

The LCDC contains the dithering pattern registers for STN LCD: a 48-bit modulo 7 dithering pattern register (LDP7L and LDP7H), a 32-bit modulo 5 dithering pattern register (LDP5), a 16-bit modulo 4 dithering pattern register (LDP4), and a 16-bit modulo 3(LDP3) dithering pattern register. These dithering pattern registers can contain the programmable pre-dithered pattern values for each duty cycle ratio.

The LDP7H and LDP7L contain 5 pre-dithered patterns for 1/7, 3/7, 4/7, 5/7, and 6/7 duty cycle rate. Each field of LDP7H and LDP7L is 7-bit long. The LDP5 has 4 pre-dithered pattern fields for 1/5, 2/5, 3/5, and 4/5 duty cycle rate. Each field of LDP5 is 5-bit long. The LDP4 has 3 pre-dithered pattern fields for 1/4, 1/2(=2/4), and 3/4 duty cycle rate, and each field is 4-bit long. Likewise, the LDP3 has 2 fields for 1/3 and 2/3 duty cycle rate with 3-bit length.

Note that the pre-dithered data for 1 and 0 is not defined in the dithering pattern register, because these values are implemented with VDD and VSS condition. The pre-dithered value is pixel value or palletized color value according to STN LCD type. Therefore, if BPP bits of LCTRL register represent RGB332 or 2BPP, pre-dithered value is palletized color. Otherwise, it is pixel value.

Table 5.2 STN LCD Dithering Pattern Register map

Pre-dithered value	Dithering register	Duty cycle ratio
0	all 0s	0
1	DP1_7	1/7
2	DP1_5	1/5
3	DP1_4	1/4
4	DP1_3	1/3
5	DP2_5	2/5
6	DP3_7	3/7
7	DP2_4	1/2
8	DP4_7	4/7
9	DP3_5	3/5
10	DP2_3	2/3
11	DP5_7	5/7
12	DP3_4	3/4
13	DP4_5	4/5
14	DP6_7	6/7
15	all 1s	1

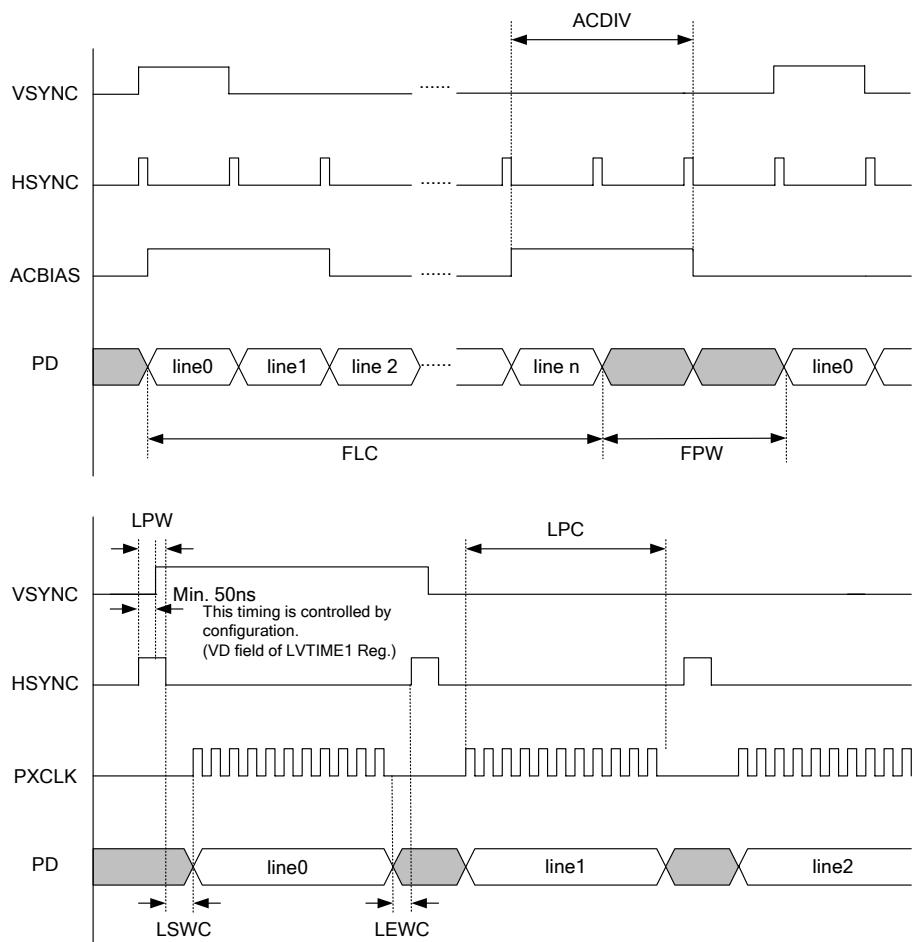


Figure 5.10 STN Mode Timing Diagram

EXAMPLE)

For a monochrome STN LCD, 4-bit interface panel, 4 pixels are captured by the panel in every panel clock cycle. Table 5.3 gives the major registers to be programmed for supporting 4-bit interface STN LCD. LCLK and Refresh rate are examples only. And LSWC, LEWC, LPW, and FPW are STN LCD panel dependent.

LCLK = 20 MHz, Refresh rate = 60 Hz
 PXDW* = 0 (4bits), BPP* = 2 (4bpp) , DP* = 0 (one pixel data per one pixel cycle)
 NI = 1, TV* = 0, TFT* = 0, STN* = 1
 LSWC* = LEWC* = LPW* = FPW* = 1 (STN LCD dependent)

Table 5.3 Monochrome STN LCD (4bits, 1BPP) example

Width (pixel)	Height (pixel)	LPC*	FLC*	DHSIZE*	DVSIZE*	f_{PXCLK_REQ}	CLKDIV*	f_{PXCLK}^{***}
160	160	39	159	160	160	0.393	25	0.4
160	200	39	199	160	200	0.491	20	0.5
320	200	79	199	320	200	0.973	10	1

*) control registers to be programmed. **) Refer to expression (2). ***) Refer to expression (1).

5.3.2 TFT-LCD

The LCDC supports 16, 18, 24 bpp true-color non-palleted color displays for color TFT LCD. It generates the control signals for LCD driver such as, VSYNC, HSYNC, PXCLK, PXDEN (ACBIAS) and PXDATA. The timing diagram of TFT mode is shown in Figure 5.11

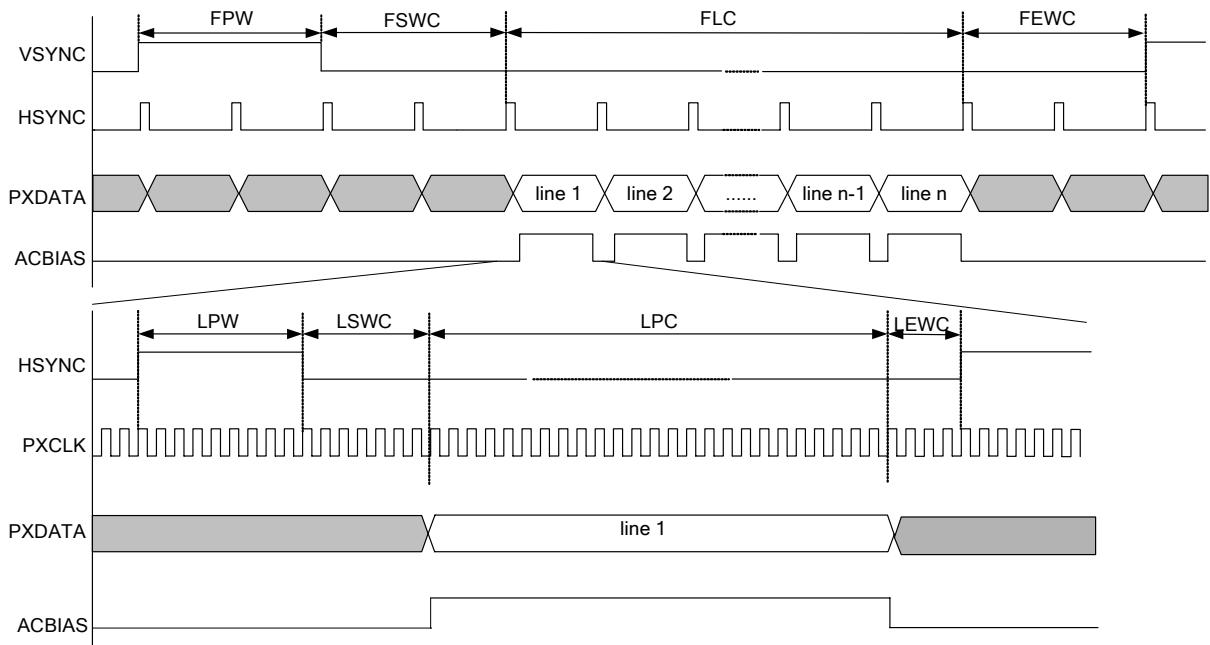


Figure 5.11 TFT Mode Timing Diagram

The VSYNC and HSYNC frequency is controlled by the LPC and FLC field.

$$LPC = (HorizontalDisplaySize) - 1$$

$$FLC = (VerticalDisplaySize) - 1$$

And PXCLK frequency is determined by the CLKDIV value.

$$f_{PXCLK} = f_{LCLK} / (2 \times CLKDIV) \quad (3)$$

The frequency of VSYNC signal is the frame rate. So the frame rate can be calculated as follows:

$$f_{Hsync} = \frac{f_{PXCLK}}{\{(LPW+1)+(LPC+1)+(LSWC+1)+(LEWC+1)\}}$$

$$f_{Vsync} = \frac{f_{Hsync}}{\{(FPW+1)+(FPC+1)+(FSWC+1)+(FEWC+1)\}}$$

Therefore, if FR is the required refresh rate in TFT mode, f_{PXCLK_REQ} , which is the required PXCLK, is as following.

$$f_{PXCLK_REQ} = FR \times \{(FPW + 1) + (FPC + 1) + (FSWC + 1) + (FEWC + 1)\} \times \{(LPW + 1) + (LPC + 1) + (LSWC + 1) + (LEWC + 1)\}$$

(4)

Example)

For TFT LCD(RGB565), if LCLK = 80MHz and Refresh rate = 60Hz,

PXDW* = 0x4, YCbCr* = 0, BPP* = 0x4, DP* = 0, NI* = 1, TV* = 0, TFT* = 1, STN* = 0
LSWC* = LEWC* = LPW* = 3 (TFT LCD dependent)
FSWC* = FEWC* = FPW* = 1 (TFT LCD dependent)

Table 5.4 TFT LCD (RGB565) example

Width (pixel)	Height (pixel)	LPC*	FLC*	DHSIZE*	DVSIZE*	f_{PXCLK_REQ} **	CLKDIV*	f_{PXCLK} ***
176	220	175	219	176	220	2.55	15	2.67
240	320	239	319	240	320	4.93	8	5
640	480	639	479	640	480	19.01	2	20

.) control registers to be programmed. **) Refer to expression (4). ***) Refer to expression (3).

5.3.3 NTSC/PAL Interface

The LCD can generate the control signals for 8-bit or 16-bit NTSC/PAL encoder. The supporting mode is CCIR601/656 interlace/non-interlace.

For NTSC/PAL interface, TV bit of LCTRL register must be set. Registers used in this mode are similar to those in TFT mode except for LVTIME1 and LVTIME2 registers; LVTIME1 is for odd field and LVTIME2 is for Even field. And these registers value is not based on HSYNC, but based on half of HSYNC. For example, if FPW of LVTIME1 is 3, pulse width of VSYNC on odd field is not 4 HSYNC cycles, but 2 HSYNC cycles. And if FPW of LVTIME1 is 4, it is 2.5 HSYNC cycles.

Interlace/Non-interlace mode can be configured by NI bit of LCTRL register. Figure 5.12 and Figure 5.13 show the timing diagram of NTSC and PAL interlace mode. In non-interlace mode, odd field sync signals are repeated instead.

The CCIR656 mode can be configured by 656 bit of LCTRL register. This mode uses 8 interface signals which have SYNC information as well as 8-bit pixel data. Thus, the additional sync signals are not needed. The output pixel clock must be 27 MHz. Figure 5.14 shows the embedded sync information in the CCIR656 mode.

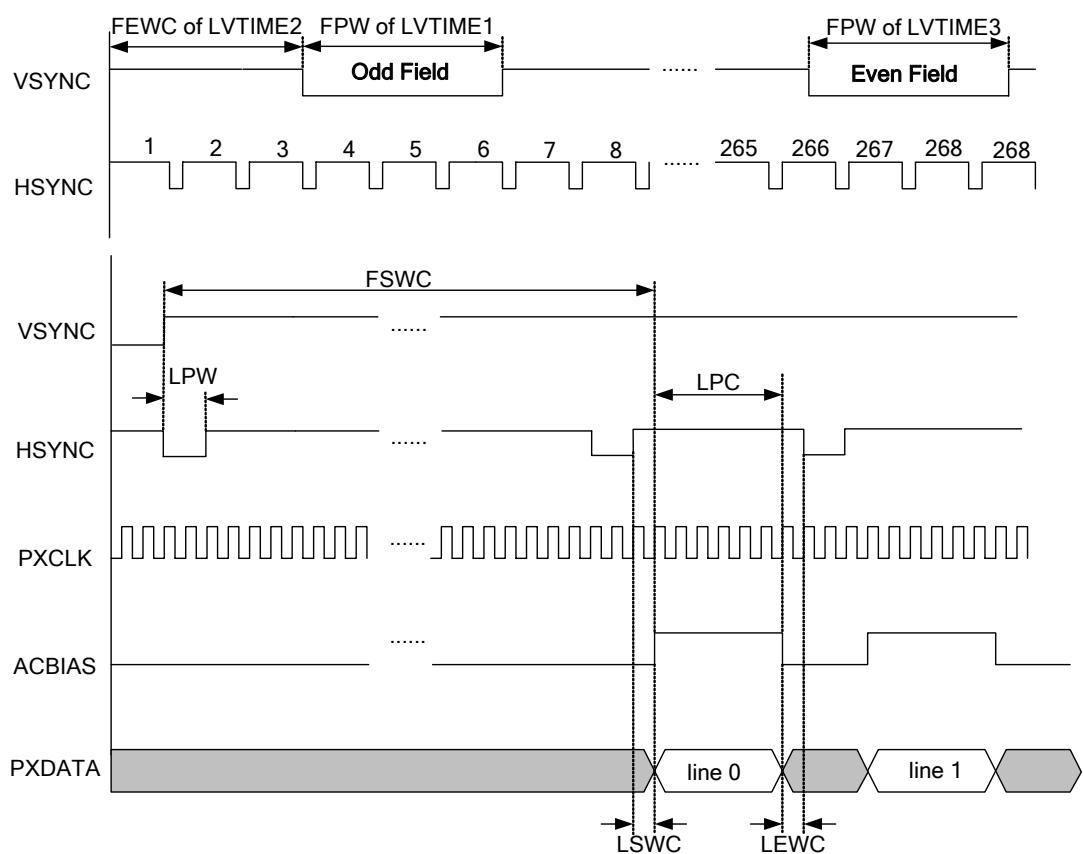


Figure 5.12 NTSC Interlace Mode Timing Diagram

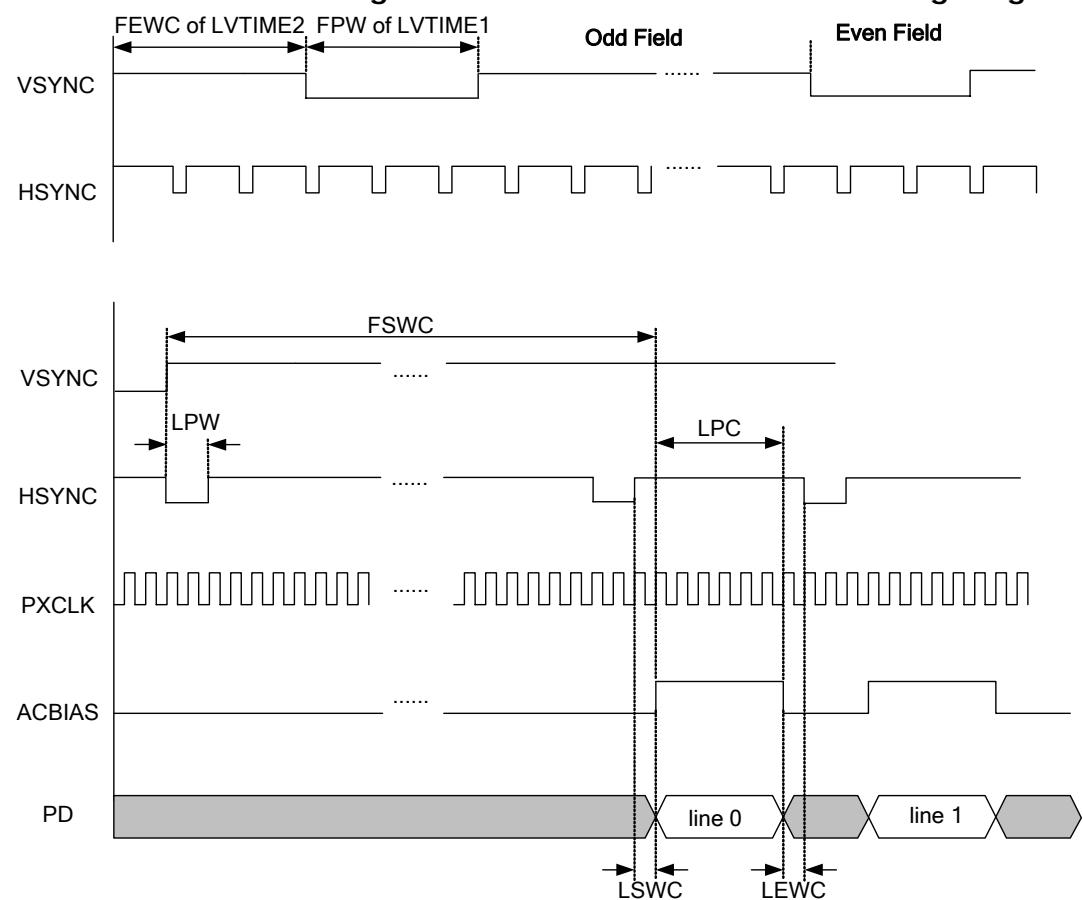


Figure 5.13 PAL Interlace Mode Timing Diagram

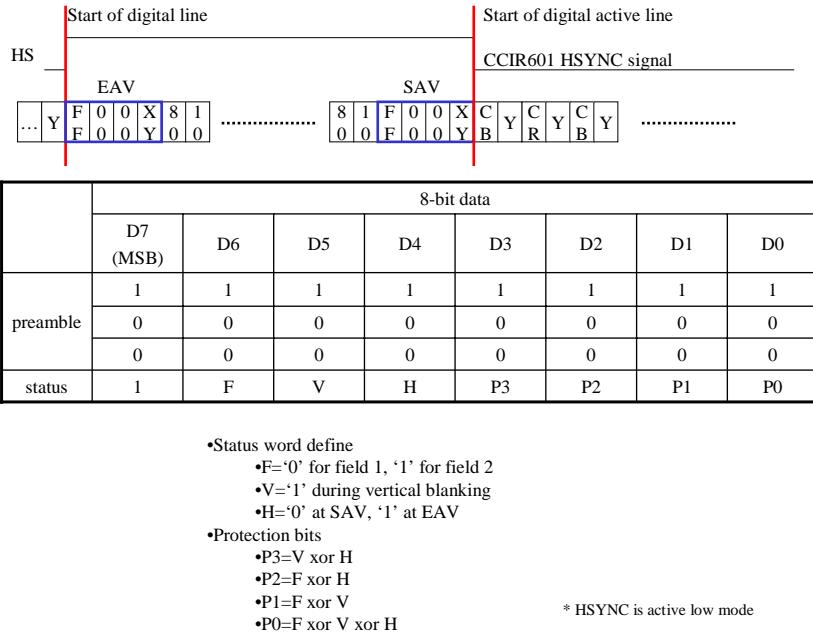
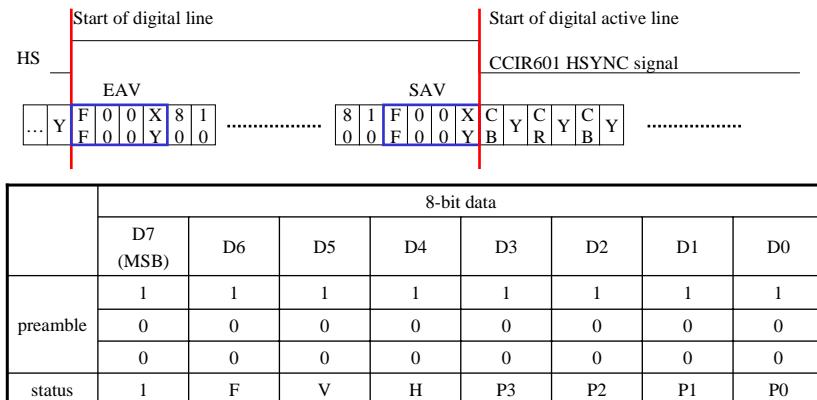


Figure 5.14 CCIR656 embedded sync. information

Figure 5.15 shows the standard timing diagram of CCIR.656 format.

In NTSC mode, one line consists of 1716 pixel clock cycles. The horizontal blanking area occupies 276 pixel clock cycles, and 8 pixel clock cycles within the horizontal blanking are SAV and EAV code. Active area occupies 1440 pixel clock cycles. One frame is composed of 525 lines and is divided into two fields, odd and even.

In PAL mode, one line consists of 1728 pixel clock cycles. The horizontal blanking area occupies 286 pixel clock cycles, and 8 pixel clock cycles within the horizontal blanking are SAV and EAV code. Active area occupies 1440 pixel clock cycles. One frame is composed of 625 lines and is divided into two fields, odd and even.



- Status word define
 - F='0' for field 1, '1' for field 2
 - V='1' during vertical blanking
 - H='0' at SAV, '1' at EAV
- Protection bits
 - P3=V xor H
 - P2=F xor H
 - P1=F xor V
 - P0=F xor V xor H

* HSYNC is active low mode

Figure 5.15 CCIR656 format diagram

5.4 Register Description

Table 5.5 LCDC Register Map (Base Address = 0xF0000000)

Name	Address	Type	Reset	Description
LCTRL	0x00	R/W	0x00000000	LCD Control Register
LBC	0x04	R/W	0x00000000	LCD Background Color Register
LCLKDIV	0x08	R/W	0x00000000	LCD Clock Divider Register
LHTIME1	0x0C	R/W	0x00000000	LCD Horizontal Timing Register 1
LHTIME2	0x10	R/W	0x00000000	LCD Horizontal Timing Register 2
LVTIME1	0x14	R/W	0x00000000	LCD Vertical Timing Register 1
LVTIME2	0x18	R/W	0x00000000	LCD Vertical Timing Register 2
LVTIME3	0x1C	R/W	0x00000000	LCD Vertical Timing Register 3
LVTIME4	0x20	R/W	0x00000000	LCD Vertical Timing Register 4
LLUTR	0x24	R/W	0x00000000	LCD Lookup Register for Red
LLUTG	0x28	R/W	0x00000000	LCD Lookup Register for Green
LLUTB	0x2C	R/W	0x00000000	LCD Lookup Register for Blue
LDP7L	0x30	R/W	0x4d2b3401	LCD Modulo 7 Dithering Pattern (low)
LDP7H	0x34	R/W	0x0000003f	LCD Modulo 7 Dithering Pattern (high)
LDP5	0x38	R/W	0x1d0b0610	LCD Modulo 5 Dithering Pattern Register
LDP4	0x3C	R/W	0x00000768	LCD Modulo 4 Dithering Pattern Register
LDP3	0x40	R/W	0x00000034	LCD 3-bit Dithering Pattern Register
LCP1	0x44	R/W	0x000000ff	LCD Clipping Register1
LCP2	0x48	R/W	0x000000ff	LCD Clipping Register2
LK1	0x4C	R/W	0x00000000	LCD Keying Register 1
LK2	0x50	R/W	0x00000000	LCD Keying Register 2
LKM1	0x54	R/W	0x00000000	LCD Keying Mask Register 1
LKM2	0x58	R/W	0x00000000	LCD Keying Mask Register 2
LDS	0x5C	R/W	0x00000000	LCD Display Size Register
LSTATUS	0x60	R/CLR	0x00000000	LCD Status Register
LIM	0x64	R/W	0x0000001f	LCD Interrupt Register.
LI0C	0x68	R/W	0x00000000	LCD Image 0 Control Register
LI0P	0x6C	R/W	0x00000000	LCD Image 0 Position Register
LI0S	0x70	R/W	0x00000000	LCD Image 0 Size Register
LI0BA0	0x74	R/W	0x00000000	LCD Image 0 Base Address 0 Register.
LI0CA	0x78	R/W	0x00000000	LCD Image 0 Current Address Register.
LI0BA1	0x7C	R/W	0x00000000	LCD Image 0 Base Address 1 Register
LI0BA2	0x80	R/W	0x00000000	LCD Image 0 Base Address 2 Register
LI0O	0x84	R/W	0x00000000	LCD Image 0 Offset Register
LI0SR	0x88	R/W	0x00000000	LCD Image 0 Scale ratio
LI1C	0x8C	R/W	0x00000000	LCD Image 1 Control Register
LI1P	0x90	R/W	0x00000000	LCD Image 1 Position Register
LI1S	0x94	R/W	0x00000000	LCD Image 1 Size Register
LI1BA0	0x98	R/W	0x00000000	LCD Image 1 Base Address 0 Register.
LI1CA	0x9C	R/W	0x00000000	LCD Image 1 Current Address Register.
-	0xA0		0x00000000	Reserved
-	0xA4		0x00000000	Reserved
LI1O	0xA8	R/W	0x00000000	LCD Image 1 Offset Register
LI1SR	0xAC	R/W	0x00000000	LCD Image 1 Scale ratio-
LI2C	0xB0	R/W	0x00000000	LCD Image 2 Control Register
LI2P	0xB4	R/W	0x00000000	LCD Image 2 Position Register
LI2S	0xB8	R/W	0x00000000	LCD Image 2 Size Register

Name	Address	Type	Reset	Description
LI2BA0	0xBC	R/W	0x00000000	LCD Image 2 Base Address 0 Register.
LI2CA	0xC0	R/W	0x00000000	LCD Image 2 Current Address Register.
-	0xC4		0x00000000	Reserved
-	0xC8		0x00000000	Reserved
LI2O	0xCC	R/W	0x00000000	LCD Image 2 Offset Register
LI2SR	0xD0	R/W	0x00000000	LCD Image 2 Scale ratio
DLCTRL	0xD4	W	0x00000000	Dual LCD Control Register
-	0xD8		0x00000000	Reserved
DLCSA0	0xDC	W	0x00000000	Dual LCD Configuration Start Address 0
DLCSA1	0xE0	W	0x00000000	Dual LCD Configuration Start Address 1
-	0xE4	R	0x00000000	Reserved
Y2RC	0xE8	R/W	0x00000000	YCbCr2RGB Control Register
S2LC	0xEC	R/W	0x00000000	Scaler to LCDC Control Register
LPMCTRL	0xF0	R/W	0x00000000	LCD PIP Control Register.
LPMW0	0xF4	R/W	0x00000000	LCD PIP Window Register 0
LPMW1	0xF8	R/W	0x00000000	LCD PIP Window Register 1
LPMH	0xFC	R/W	0x00000000	LCD PIP Height Register
LPMC	0x100	R/W	0x00000000	LCD PIP Chroma-keying Register
LPMMC	0x104	R/W	0x00000000	LCD PIP Chroma-keying Mask Register
LCDLUT	0xC00-0xFFFF	W	-	LCD Lookup Table.

LCTRL (LCD Control Registers)

0xF0000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Y2R2	AEN2	CEN2	Y2R1	AEN1	CEN1	Y2R0	656	CL	BPP<2:0>		PXDW<3:0>				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	IV	IH	IP	CLEN	R2Y	DP	NI	TV	TFT	STN	MSEL	IEN2	IEN1	IENO	LEN

Y2R2 [31] YCbCr to RGB Converter IMG2	
0	Disable
1	Enable
AEN2 [30] Alpha Blending Enable Bit for LAYER2	
0	Disable
1	Enable
CEN2 [29] Chroma Key Enable Bit for LAYER2	
0	Disable
1	Enable
Y2R1 [28] YCbCr to RGB Conversion Enable Bit for IMG1	
0	Disable
1	Enable
AEN1 [27] Alpha Blending Enable Bit for LAYER1	
0	Disable
1	Enable
CEN1 [26] Chroma Key Enable Bit for LAYER1	
0	Disable
1	Enable
Y2R0 [25] YCbCr to RGB Conversion Enable Bit for IMG0	
0	Disable
1	Enable
656 [24] CCIR 656 Mode	
0	Disable
1	Enable
CL [23] Pixel Clock Mask	
0	IT SHOULD BE 0 when the LCDC is used for inputs of LCDSI or PIP mode is enabled.
1	IT SHOULD BE 1 when the LCDC is used for LCD RGB interface or inputs of NTSC/PAL encoder.

BPP [22:20]		Bit Per Pixel for STN-LCD
0	1bpp	
1	2bpp	
2	4bpp	
3	RGB332	
4	RGB444	
5-7	Reserved	

BPP is only for STN LCD.

PXDW [19:16]		Pixel Data Width for Output
0	4 bits (It is only used for STN LCD)	
1	8 bits (It is only used for STN LCD)	
2	8 bits (stripe type: R8-G8-B8)	
3	16bits (RGB565)	
4	16bits (RGB555)	
5	18bits (RGB666)	
6	8 bits (Cb-Y-Cr-Y)	
7	8 bits (Cr-Y-Cb-Y)	
8	16 bits (YCb - YCr)	
9	16 bits (YCr - YCb)	
10	8 bits (delta type: odd line R8-G8-B8, even line G8-B8-R8)	
11	8 bits (delta type: odd line G8-B8-R8, even line R8-G8-B8)	
12	24 bits mode (stripe type: RGB888)	
13	8 bits (RGB-dummy mode: R8-G8-B8-DUMMY)	

Refer to Figure 5.16 on page 5-22 for more information about PXDW.

ID [15]		Inverted Data Enable (ACBIAS pin)
0	Active high	
1	Active low	
IV [14]		Inverted Vertical Sync
0	active high	
1	active low	
IH [13]		Inverted Horizontal Sync
0	active high	
1	active low	
IP [12]		Inverted Pixel Clock
0	Pixel data is driven on the rising edge of pixel clock.	
1	Pixel data is driven on the falling edge of pixel clock.	
CLEN[11]		Clipping Enable
0	Disable	
1	Enable	
R2Y[10]		RGB to YCbCr Converter Enable for OUTPUT
0	Disable	
1	Output pixel data is converted to YCbCr format.	

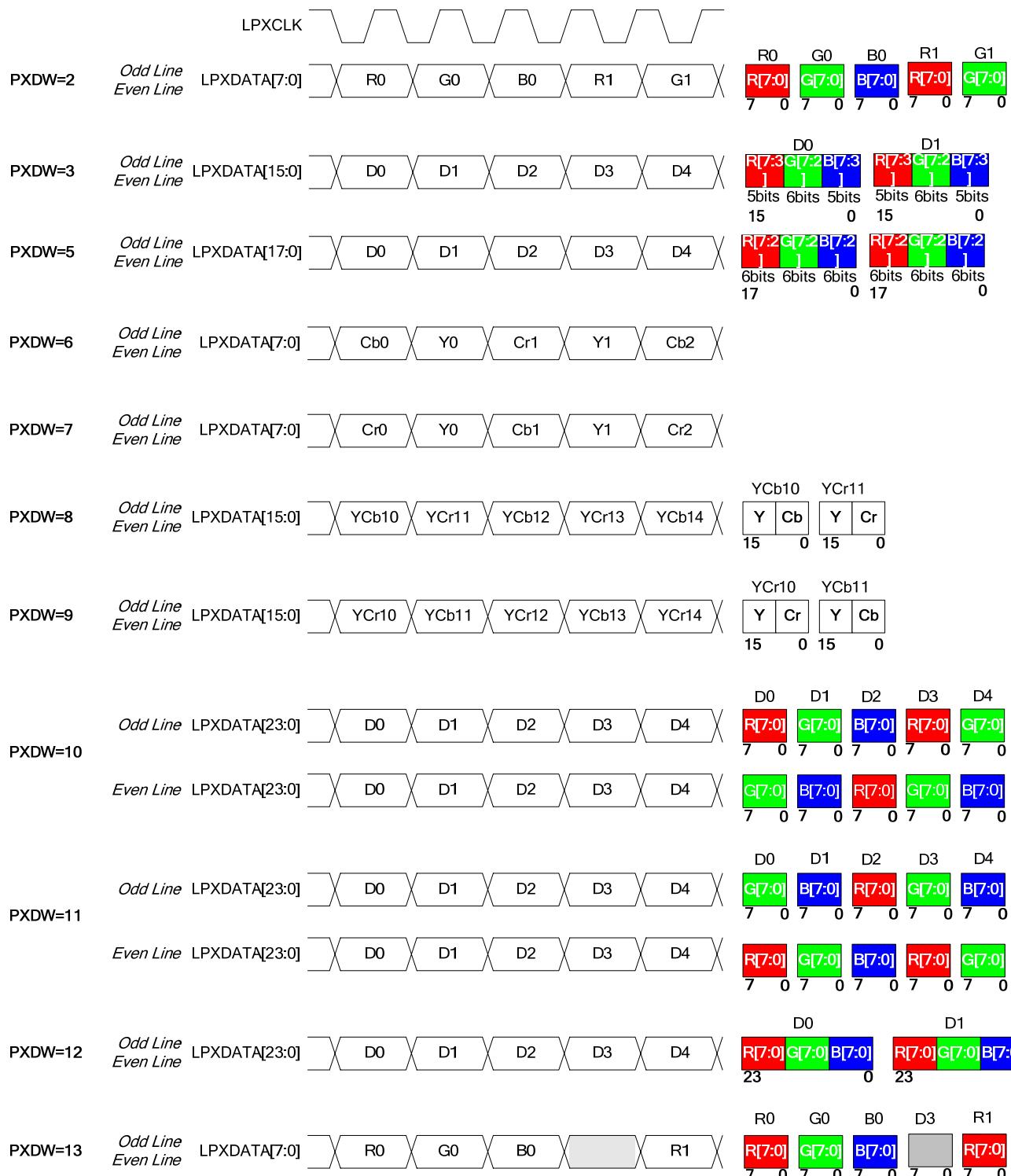


Figure 5.16 Output Pixel Data Format

DP[9]			Double Pixel Data
0			One pixel data per 1 PXCLK cycle is output
1			One pixel data per 2 PXCLK cycle is output.
NI[8]			Non-interlace
0			Interlace mode
0			Odd field timing control : LVTME1, LVTME2 Even field timing control : LVTIME3, LVTIME4
1			Non-interlace mode (progressive mode)
TV[7]			Non-interlace
0			STN-LCD mode
0			TFT-LCD mode
1			TV mode Values of LVTIME _n registers are based on HSYNC cycle/2.
MSEL[4]			Master Select for IMG0
0			IMG0 comes from the internal or the external memory.
1			IMG0 comes from the memory to memory scaler.
IEN2[3]			IMG2 Enable bit
0			Disable
1			Enable
IEN1[2]			IMG1 Enable bit
0			Disable
1			Enable
IEN0[1]			IMG0 Enable bit
0			Disable
1			Enable
LEN[0]			LCD Controller Enable
0			Disable
1			Enable

LBC (LCD Background Color)																0xF0000004			
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																BG2<7:0>			
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																BG1<7:0>			
BG1<7:0>																BG0<7:0>			

FIELD	Description
BG2 [23:16]	Background color 2 (Y/B)
BG1 [15:8]	Background color 1 (Cb/G)
BG0 [7:0]	Background color 0 (Cr/R)

LCLKDIV (LCD Clock Divider Register)																0xF0000008			
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																ACDIV<7:0>			
CS 0																ACDIV<7:0>			
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																PXCLKDIV<7:0>			

FIELD	Description
CS [31]	LCLK source selector
ACDIV [23:16]	0 : The clock source of LCLK is CKC (Clock Controller) 1 : The clock source of LCLK is HCLK (system bus clock)
LCLKDIV [15:8]	AC bias clock divisor (STN only) $LCLK = HCLK / (LCLKDIV+1)$ (LCLKDIV=0~255)
PXCLKDIV [7:0]	Pixel clock divider. Note that programming CLKDIV less than 3 is illegal for STN LCD. $PXCLK = LCLK / (2^{PXCLKDIV})$ (PXCLKDIV=0~255)

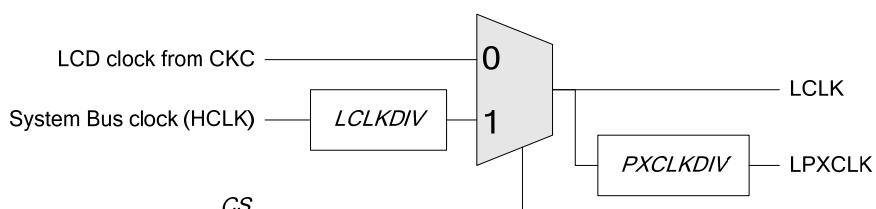


Figure 5.17 Clock control of LCD interface

LHTIME1 (LCD Horizontal Timing Register 1)

0xF000000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															LPW<8:0>
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															LPC<10:0>

FIELD	Description
LPW [24:16]	Line pulse width Line pulse count is the number of pixel clock cycles in each line minus 1 on the screen.
LPC [10:0]	TFT/NTSC(16bit)/PAL(16bit) : active horizontal pixel – 1 Color STN : (3 * Horizontal display size / pixel width) Mono STN : (Horizontal display size / pixel width) - 1

LHTIME2 (LCD Horizontal Timing Register 2)

0xF0000010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															LSWC<8:0>
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															LEWC<8:0>

FIELD	Description
LSWC [24:16]	Line start wait clock is the number of dummy pixel clock cycles minus 1 to be inserted from the start of each horizontal line of pixels.
LEWC [8:0]	Line end wait clock is the number of dummy pixel clock cycles minus 1 to be inserted before the end of each horizontal line of pixels

LVTIME1 (LCD Vertical Timing Register 1) 0xF0000014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VDB<4:0>					0	VDF<3:0>				FPW<5:0>					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								FLC<10:0>							

FIELD	Description
VDB[31:27]	Back porch VSYNC delay Delay cycle is -10 to 10cycle delay by PXLCLK. When TV mode, VDB value is equal to VDF. Ex) if VD=5, VSYNC delay is 5 cycle delay by HSYNC.
VDF[25:22]	Front porch of VSYNC delay Delay cycle is 0 to 10 cycle delay by PXLCLK. Ex) if VD=5, VSYNC delay is 5 cycle delay by HSYNC.
FPW [21:16]	TFT/TV : Frame pulse width is the pulse width of frame clock (VSYNC). STN : N/A
FLC [10:0]	Frame line count is the number of lines in each frame on the screen.

Refer to Figure 5.10, Figure 5.11, and Figure 5.12.

LVTIME2 (LCD Vertical Timing Register 2) 0xF0000018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								FSWC<8:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								FEWC<8:0>							

FIELD	Description
FSWC [24:16]	TFT/TV: Frame start wait cycle is the number of lines to be inserted at the end of each frame. STN : FSWC is N/A. If FSWC[0] is set, VSYNC signal starts on negative falling edge of HSYNC.
FEWC [8:0]	TFT/TV: Frame end wait cycle is the number of lines to be inserted at the beginning of each frame. STN extra dummy lines between the end and beginning of frame.

LVTIME3 (LCD Vertical Timing Register 3) 0xF000001C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								FPW<5:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								FLC<10:0>							

LVTIME4 (LCD Vertical Timing Register 4) 0xF0000020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								FSWC<8:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								FEWC<8:0>							

If NI of LCTRL is 0, LVTIME3 and LVTIME4 is for even field. Otherwise, LVTIME3 and LVTIME4 must be equal to LVTIME1 and LVTIME2, respectively.

LLUTR (LCD Lookup Register for RED)

0xF0000024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LLUTR<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LLUTR<15:0>															

This register is used for supporting palletized color STN-LCD. It is divided into 8 nibbles. The passive color mode uses a lookup table register, which allows any 8 red levels to be selected out of the 16 possible red levels. The most significant 3-bit of 8-bit encoded pixel addresses 8 red palette locations. Note that LLUTR register is only used in STN-LCD mode.

LLUTG (LCD Lookup Register for GREEN)

0xF0000028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LLUTG<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LLUTG<15:0>r															

This register is used for supporting palletized color STN-LCD. It is divided into 8 nibbles. The passive color mode uses a lookup table register, which allows any 8 green levels to be selected out of the 16 possible green levels. The most significant 3-bit of 8-bit encoded pixel addresses 8 green palette locations. Note that LLUTG register is only used in STN-LCD mode.

LLUTB (LCD Lookup Register for BLUE)

0xF000002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LLUTB<15:0>															

This register is used for supporting palletized color STN-LCD. It is divided into 4 nibbles. The passive color mode uses a lookup table register, which allows any 4 blue levels to be selected out of the 16 possible blue levels. The most significant 2-bit of 8-bit encoded pixel addresses 4 blue palette locations. Note that LLUTB register is only used in STN-LCD mode.

LDP7L (LCD Dithering Pattern Register)

0xF0000030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								DP5_7<6:0>	0						DP4_7<6:0>
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								DP3_7<6:0>	0						DP1_7<6:0>

LDP7H (LCD Dithering Pattern Register)

0xF0000034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0									0						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0									DP6_7<6:0>						

LDP5 (LCD Dithering Pattern Register)

0xF0000038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0									DP4_5<4:0>	0					DP3_5<4:0>
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								DP2_5<4:0>	0						DP1_5<4:0>

LDP4 (LCD Dithering Pattern Register)

0xF000003C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0									0						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								DP3_4<3:0>			DP2_4<3:0>				DP1_4<3:0>

LDP3 (LCD Dithering Pattern Register)

0xF0000040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0									0						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0									DP2_3<2:0>						DP1_3<2:0>

These dithering pattern registers are only used by STN LCD. It is recommended that reset values are used.

Pre-dithered value	Dithering register	Duty cycle ratio
0	all 0s	0
1	DP1_7	1/7
2	DP1_5	1/5
3	DP1_4	1/4
4	DP1_3	1/3
5	DP2_5	2/5
6	DP3_7	3/7
7	DP2_4	1/2
8	DP4_7	4/7
9	DP3_5	3/5
10	DP2_3	2/3
11	DP5_7	5/7
12	DP3_4	3/4
13	DP4_5	4/5
14	DP6_7	6/7
15	all 1s	1

LCP1 (LCD Clipping Register 1)

0xF0000044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLP2L								CLP2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLP1L								CLP1							

FIELD	Description
CLP2L[31:24]	Clipping U/G below this value. (standard value is 0)
CLP2 [23:16]	Clipping U/G upper this value. (standard value is 255)
CLP1L [15:8]	Clipping Y/R below this value. (standard value is 0)
CLP1 [7:0]	Clipping Y/R upper this value. (standard value is 255)

LCP2 (LCD Clipping Register 2)

0xF0000048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLP3L								CL3Y							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

FIELD	Description
CLP3L [15:8]	Clipping V/B below this value. (standard value is 0)
CLP3 [7:0]	Clipping V/B upper this value. (standard value is 255)

LCP1 and LCP2 may be used not to interpret pixel data as embedded sync signal in the CCIR-656 interface.

LK1 (LCD Keying Register)

0xF000004C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SA1					A10<3:0>						KR1<7:0>				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				KG1<7:0>						KB1<7:0>					

BIT	NAME	DESCRIPTION
31	SA1	0: A10[3:0] bits and A11[3:0] are used as alpha value for LAYER1 1: Alpha bits in the pixel are used as alpha value for LAYER1. These bits are used as alpha value for LAYER1 when SA1 bit is set to 0.
27-24	A10[3:0]	output pixel = main pixel *(1- 0.0625 * A10) + overlay pixel* (0.0625*A10) Therefore , it means (100 - (A10 x 6.25))% transparency (e.g., A10= 0 : 100%, A10=8 : 50%, A10=15 : 6.25% transparency)
23-16	KR1[7:0]	chroma-key value in R(Y) channel for LAYER1
15-8	KG1[7:0]	chroma-key value in G(U) channel for LAYER1
7-0	KB1[7:0]	chroma-key value in B(V) channel for LAYER1

LK2 (LCD Keying Register)

0xF0000050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SA2					A20<3:0>						KR2<7:0>				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			KG2<7:0>							KB2<7:0>					

BIT	NAME	DESCRIPTION
31	SA2	0: A20[3:0] bits and A21[3:0] are used as alpha value for LAYER2 1: Alpha bits in the pixel are used as alpha value for LAYER2. These bits are used as alpha value for LAYER2 when SA2 bit is set to 0.
27-24	A20[3:0]	output pixel = main pixel *(1- 0.0625 * A20) + overlay pixel* (0.0625*A20) Therefore , it means (100 - (A20 x 6.25))% transparency (e.g., A20= 0 : 100%, A20=8 : 50%, A20=15 : 6.25% transparency)
23-16	KR2[7:0]	chroma-key value in R(Y) channel for LAYER2
15-8	KG2[7:0]	chroma-key value in G(U) channel for LAYER2
7-0	KB2[7:0]	chroma-key value in B(V) channel for LAYER2

Refer to "Chroma-Keying" on page 5-8 for more information.

LKM1 (LCD Keying Mask Register)

0xF0000054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					A11<3:0>										MKR1<7:0>
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					MKG1<7:0>										MKB1<7:0>

BIT	NAME	DESCRIPTION
27-24	A11[3:0]	These bits should be the same as value of "A10".
23-16	MKR1[7:0]	Masking chroma-key value in R(Y) channel for LAYER1
15-8	MKG1[7:0]	Masking chroma-key value in G(U) channel for LAYER1
7-0	MKB1[7:0]	Masking chroma-key value in B(V) channel for LAYER1

LKM2 (LCD Keying Mask Register)

0xF0000058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					A21<3:0>										MKR2<7:0>
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					MKG2<7:0>										MKB2<7:0>

BIT	NAME	DESCRIPTION
27-24	A21[3:0]	These bits should be the same as value of "A20."
23-16	MKR2[7:0]	Masking chroma-key value in R(Y) channel for LAYER2
15-8	MKG2[7:0]	Masking chroma-key value in G(U) channel for LAYER2
7-0	MKB2[7:0]	Masking chroma-key value in B(V) channel for LAYER2

Refer to "Chroma-Keying" on page 5-8 for more information.

LDS (LCD Display Size Register)

0xF000005C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VSIZE<11:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSIZE<11:0>															

FIELD

Description

VSIZE [27:16] Horizontal size : number of active pixel in a line

HSIZE [11:0] Vertical size : number of active lines

LSTATUS (LCD Status Register)

0xF0000060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

VS VS [15] Monitoring vertical sync.

FIELD

Description

VS[15] It is the same as LVSYNC logic level.

ITY [12] "1" SHOULD BE WRITTEN TO THIS BIT.

ICR [8] When writing 1 to this bit, the LCDC interrupt is cleared.

BY [6] Busy signal

BY [6] When LCDC is operating, this bit is set to 1. If LEN is disabled, BY will be 0 after current frame has been displayed.

EF [5] Even-field (read only)

EF [5] 0 : Odd field or frame
1 : Even field or frame

DD [4] Disable Done (Read/Clear)

DD [4] If LEN is disabled, DD will be 1 after current frame has been displayed.
If MDD bit of LIM register is 0, it can become LCD interrupt source.

DD [4] Register update (Read/Clear)

RU [3] It indicates that all registers programmed are applied to current frame data. If MRU bit of LIM register is 0, it can become LCD interrupt source.

RU [3] FIFO underrun (Read/Clear)

FU [0] It indicates that FIFO underrun has been occurred. In this case, LCLK frequency must be lower. If MFU bit of LIM register is 0, it can become LCD interrupt source.

LIM (LCD Interrupt Masking Register)

0xF0000064

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MDD MRU 0 MFU

FIELD	Description
MDD [4]	Masking “disable done interrupt”. 0: DD bit can become LCD interrupt source.
MRU [3]	Masking “register update interrupt” 0: RU bit can become LCD interrupt source.
MFU [0]	Masking “FIFO underrun interrupt”. 0: FU bit can become LCD interrupt source.

LI0C (LCD IMG0 Control Register)

0xF0000068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD	OP							BR	YCbCr					BPP	

PD [15]	Bit padding
0	ZERO padding
1	MSB padding

Although raw image source is not RGB888 format, it is internally converted to RGB888 as appending additional bits. When this conversion is occurred, these padding bits are determined by PD[15] bit.

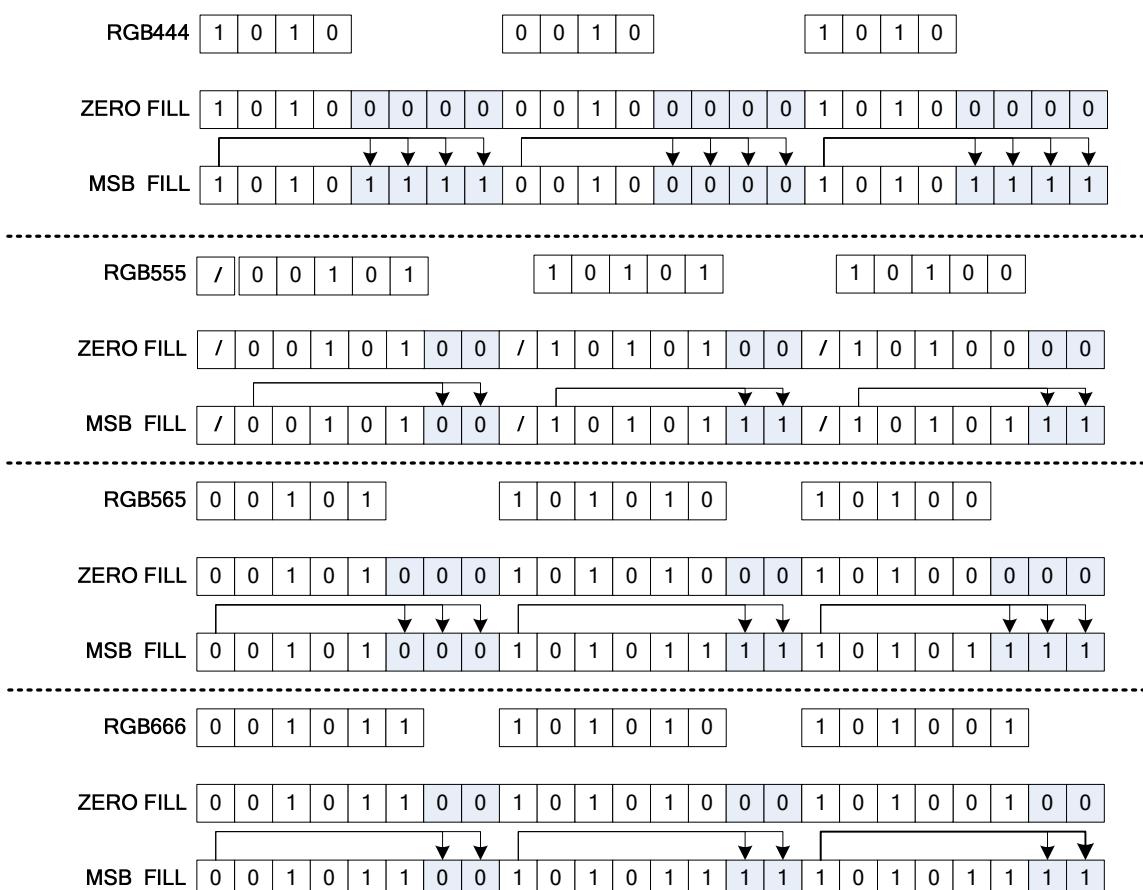


Figure 5.18 Bit Padding

OP[13:12] should be set to 5. Its reset value is 5.

BR[7]	Bit Reverse
0	Little-endian pixel data
1	Big-endian pixel data

BR[7] is only used when BPP is 1, 2, or 4 bpp

YCbCr[6]	YCbCr[5]	YCbCr[4]	YCbCr
0	0	1	YCbCr 4:2:0
0	1	1	YCbCr 4:2:2
1	1	1	sequential YCbCr 4:2:2

YCbCr[6:4] is valid only if the BPP[3:0] is 3.

BPP [3:0]	Bit Per Pixel for Raw Image Source
0	1bpp
1	2bpp
2	4bpp
	RGB332 or YCbCr
3	If the YCbCr[4] is 0, IMG0 is RGB332 format in RGB color space. If the YCbCr[4] is 1, BPP[3:0] should be 3 and IMG0 is in YCbCr color space.
4	RGB444
5	RGB565
6	RGB555
7	RGB888
8	RGB666 (18bpp)

1BPP	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																													
	p31 p30 p29 p28 p27 p26 p25 p24 p23 p22 p21 p20 p19 p18 p17 p16																													
2BPP	p15 p14 p13 p12 p11 p10 p9 p8 p7 p6 p5 p4 p3 p2 p1 p0																													
	p15 p14 p13 p12 p11 p10 p9 p8 p7 p6 p5 p4 p3 p2 p1 p0																													
4BPP	p7 p6 p5 p4 p3 p2 p1 p0																													
	p7 p6 p5 p4 p3 p2 p1 p0																													
BR=0																														
1BPP	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																													
	p24 p25 p26 p27 p28 p29 p30 p31 p16 p17 p18 p19 p20 p21 p22 p23																													
2BPP	p12 p13 p14 p15 p8 p9 p10 p11 p12 p13 p14 p15 p16 p17 p18 p19																													
	p12 p13 p14 p15 p8 p9 p10 p11 p12 p13 p14 p15 p16 p17 p18 p19																													
4BPP	p6 p7 p4 p3 p2 p1 p0																													
	p6 p7 p4 p3 p2 p1 p0																													
BR=1																														
RGB332																														
R1[2:0] G1[2:0] B1[1:0] R0[2:0] G0[2:0] B0[1:0]																														
RGB444																														
alpha[3:0] R[3:0] G[3:0] B[3:0]																														
RGB565																														
R[4:0] G[5:0] B[4:0]																														
RGB555																														
I R[4:0] G[4:0] B[4:0]																														
RGB666																														
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																														
IGNORED R[5:4]																														
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																														
R[3:0] G[5:0] B[5:0]																														
RGB888																														
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																														
IGNORED alpha*[3:0] R[7:0]																														
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																														
G[7:0] B[7:0]																														
Sequential																														
YCbCr4:2:2																														
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																														
Cr0[7:0] Y1[7:0]																														
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																														
Cb0[7:0] Y0[7:0]																														
.....																														
YCbCr4:2:2																														
YCbCr4:2:0																														
.....																														
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																														
Cr1[7:0] Cr0[7:0]																														

Figure 5.19 Supported Pixel Data Format (BPP, YCbCr, and BR of LinC register)

LI0P (LCD IMG0 Position Register)

0xF000006C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMG_Y<10:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMG_X<10:0>															

FIELD

Description

IMG_Y [26:16] Y position to display.

IMG_X [10:0] X position to display.

LI0S (LCD IMG0 Size Register)

0xF0000070

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HEIGHT<11:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WIDTH<11:0>															

FIELD

Description

HEIGHT [27:16] IMG0 height

WIDTH [11:0] IMG0 width

If the part of the raw image source with HEIGHT and WIDTH exists outside of the virtual display because of IMG_X and/or IMG_Y, “Offset register” should be used and “Size register” should be recalculated for the raw image source to be located in the virtual display.

LI0BA0 (LCD IMG0 Base Address0)

0xF0000074

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I0_BASE0<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I0_BASE0<15:2>															

FIELD

Description

I0_BASE0 [31:2]	IMG0 base address If IMG0 is YCbCr data, it is Y base address.
-----------------	---

LI0CA (LCD IMG0 Current Address)

0xF0000078

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I0_CUR<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I0_CUR<15:0>															

FIELD

Description

I0_CUR [31:0]	IMG0 current address.
---------------	-----------------------

LI0BA1 (LCD IMG0 Base Address1)

0xF000007C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I0_BASE1<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I0_BASE1<15:2>															

FIELD

Description

I0_BASE1 [31:2]	IMG0 base address If IMG0 is YCbCr data, it is U base address. Otherwise, it is not used.
-----------------	---

LI0BA2 (LCD IMG0 Base Address2)

0xF0000080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I0_BASE2<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I0_BASE2<15:2>															

FIELD

Description

I0_BASE2 [31:2]	IMG0 base address If IMG0 is YCbCr data, it is V base address. Otherwise, it is not used.
-----------------	---

LI0O (LCD IMG0 Offset)

0xF0000084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I0_OFS1<26:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I0_OFS0<10:0>															

FIELD	Description
	IMG0 offset.
I0_OFS1[26:16]	If IMG0 is in YCbCr color space, it is offset for Cb and Cr image. Otherwise, it is not used.
	IMG0 offset.
I0_OFS0 [10:0]	If IMG0 is in YCbCr color space, it is offset for Y image. Otherwise, it is offset for RGB image.

To use “Offset”, notice that there are restrictions about the corresponding raw image source. Refer to Figure 5.6 and descriptions on page 5-6.

LI0SC (LCD IMG0 Scale)

0xF0000088

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Y_SCALE				X_SCALE			

SCALE[3]	[2]	[1]	[0]	Description
0	0	0	0	Non-Scalable
0	0	0	1	Downscale by 2
0	0	1	0	Downscale by 3
0	0	1	1	Downscale by 4
0	1	1	1	Downscale by 8
1	0	0	1	Upscale by 2
1	0	1	0	Upscale by 3
1	0	1	1	Upscale by 4
1	1	1	1	Upscale by 8

LI1C (LCD IMG1 Control Register)

0xF000008C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD											BR	YCbCr	BPP		

PD [15]	Bit padding
0	ZERO Padding
1	MSB Padding

Though raw image source is not RGB888 format, it is internally converted to RGB888 as appending additional bits. When this conversion is occurred, these padding bits are determined by PD[15] bit. Refer to Figure 5.18.

BR [7]	Bit Reverse
0	Little-endian pixel data
1	Big-endian pixel data

BR[7] is only used when BPP is 1, 2, or 4 bpp

YCbCr[6]	YCbCr[5]	YCbCr[4]	YCbCr
"Sequential YCbCr 4:2:2"			
1	1	1	When YCbCr color space is used, IMG1 only supports "Sequential YCbCr 4:2:2".
When YCbCr[4] is set to 1, BPP[3:0] SHOULD BE 3.			
3			

BPP [3:0]	Bit Per Pixel
0	1bpp
1	2bpp
2	4bpp
3	RGB332
3	When YCbCr[4] is set to 1, BPP[3:0] SHOULD BE 3.
4	RGB444
5	RGB565
6	RGB555
7	RGB888
8	RGB666

Refer to Figure 5.19 on page 5-36 about BR, BPP[3:0], and YCbCr[6:4] bits.

LI1P (LCD IMG1 Position Register)

0xF0000090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMG_Y<10:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMG_X<10:0>															

FIELD	Description
IMG_Y [26:16]	Y position to display.
IMG_X [10:0]	X position to display.

LI1S (LCD IMG1 Size Register)

0xF0000094

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HEIGHT<11:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WIDTH<11:0>															

FIELD	Description
HEIGHT [27:16]	IMG1 height
WIDTH [11:0]	IMG1 width

If the part of the raw image source with HEIGHT and WIDTH exists outside of the virtual display because of IMG_X and/or IMG_Y, “Offset register” should be used and “Size register” should be reassigned for the raw image source to be located in the virtual display.

LI1BA0 (LCD IMG1 Base Address0) 0xF0000098

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I1_BASE0<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I1_BASE0<15:2>															
0															

FIELD	Description
I1_BASE0	IMG1 base address
[31:2]	If IMG1 is YCbCr data, it is Y base address.

LI1CA (LCD IMG1 Current Address) 0xF000009C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I1_CUR<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I1_CUR<15:0>															

FIELD	Description
I1_CUR [31:0]	IMG1 current address.

LI1O (LCD IMG1 Offset)

0xF00000A8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I1_OFS0<10:0>															

FIELD	Description
	IMG1 offset.
I1_OFS0 [10:0]	If IMG1 is in YCbCr color space, it is offset for Y image. Otherwise, it is offset for RGB image.

To use “Offset”, notice that there are restrictions about the corresponding raw image source. Refer to Figure 5.6 and descriptions on page 5-6.

LI1SC (LCD IMG1 Scale)

0xF00000AC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Y_SCALE					X_SCALE		
0															

SCALE[3]	[2]	[1]	[0]	Description
0	0	0	0	Non-Scalable
0	0	0	1	Downscale by 2
0	0	1	0	Downscale by 3
0	0	1	1	Downscale by 4
0	1	1	1	Downscale by 8
1	0	0	1	Upscale by 2
1	0	1	0	Upscale by 3
1	0	1	1	Upscale by 4
1	1	1	1	Upscale by 8

LI2C (LCD IMG2 Control Register)

0xF00000B0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD					LUT			BR		YCbCr			BPP		

PD[15]	Bit padding
0	ZERO Padding
1	MSB Padding enable

Although raw image source is not RGB888 format, it is internally converted to RGB888 as appending additional bits. When this conversion is occurred, these padding bits are determined by PD[15] bit. Refer to Figure 5.18.

LUT [10:9]	Use Lookup Table
0	Not used
1	IMG0 uses the color lookup table.
2	IMG1 uses the color lookup table.
3	IMG2 uses the color lookup table.

Refer to “Look-Up Table” on page 5-6. The lookup table cannot be used for 2 or more raw image sources at the same time.

BR[7]	Bit Reverse
0	Little-endian pixel data
1	Big-endian pixel data

BR[7] is only used when BPP is 1, 2, or 4 bpp

YCbCr[6]	YCbCr[5]	YCbCr[4]	YCbCr
“Sequential YCbCr 4:2:2”			
1	1	1	When YCbCr color space is used, IMG2 only supports “Sequential YCbCr 4:2:2”. In this case, BPP[3:0] SHOULD BE 3.

BPP [3:0]	Bit Per Pixel
0	1bpp
1	2bpp
2	4bpp
3	RGB332 When YCbCr[4] is set to 1, BPP[3:0] SHOULD BE 3.
4	RGB444
5	RGB565
6	RGB555
7	RGB888
8	RGB666

Refer to Figure 5.4 Supported Pixel Data Format about BR, BPP[3:0], YCbCr[6:4] bits

LI2P (LCD IMG2 Position Register)

0xF00000B4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMG_Y<10:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMG_X<10:0>															

FIELD	Description
IMG_Y [26:16]	Y position to display.
IMG_X [10:0]	X position to display.

LI2S (LCD IMG2 Size Register)

0xF00000B8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HEIGHT<11:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WIDTH<11:0>															

FIELD	Description
HEIGHT [27:16]	IMG2 height
WIDTH [11:0]	IMG2 width

If the part of the raw image source with HEIGHT and WIDTH exists outside of the virtual display because of IMG_X and/or IMG_Y, “Offset register” should be used and “Size register” should be reassigned for the raw image source to be located in the virtual display.

LI2BA0 (LCD IMG2 Base Address0)**0xF00000BC**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2_BASE0<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2_BASE0<15:2>															
0															

FIELD**Description**

I2_BASE0 [31:2]	IMG2 base address If IMG2 is YCbCr data, it is Y base address.
-----------------	---

LI2CA (LCD IMG2 Current Address)**0xF00000C0**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2_CUR<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2_CUR<15:0>															

FIELD**Description**

I2_CUR [31:0]	IMG2 current address.
---------------	-----------------------

LI2O (LCD IMG2 Offset)

0xF00000CC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2_OFS0<10:0>															

FIELD	Description
I2_OFS0 [10:0]	IMG2 offset. Address offset in Y channel of FIFO (FIFO0).

To use “Offset”, notice that there are restrictions about the corresponding raw image source. Refer to Figure 5.6 and descriptions on page 5-6.

LI2SC (LCD IMG2 Scale)

0xF00000D0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y_SCALE										X_SCALE					

SCALE[3]	[2]	[1]	[0]	Description
0	0	0	0	Non-Scalable
0	0	0	1	Downscale by 2
0	0	1	0	Downscale by 3
0	0	1	1	Downscale by 4
0	1	1	1	Downscale by 8
1	0	0	1	Upscale by 2
1	0	1	0	Upscale by 3
1	0	1	1	Upscale by 4
1	1	1	1	Upscale by 8

DLCTRL (Dual LCD Control)

0xF00000D4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DRE [4] Dual LCD Register Update Enable

0	Disable
1	Enable

DLE [0] Dual LCD Operation Enable

0	Disable
1	Enable

DLCSA1 (Dual LCD Configuration Start Address 1)

0xF00000DC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DLCSA1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

FIELD Description

DLCSA1 [31:2] Configuration start address for LCD1.

DLCSA0 (Dual LCD Configuration Start Address 0)

0xF00000E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DLCSA0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

FIELD Description

DLCSA0 [31:2] Configuration start address for LCD0.

Y2RC (Y2R Configuration Register)

0xF00000E8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

When the YCbCr to RGB color space converter for the raw image source is enabled, its type is determined by Y2RC register. Y2RM0, Y2RM1, and Y2RM2 are for IMG0, IMG1, and IMG2 respectively.

Y2RMn[1:0]	YUV to RGB conversion equations
0	<p>The basic equations to convert between 8-bit digital RGB data width a 16-235 nominal range.</p> <p>This equation is used SDTV set (CCIR601 format)</p> $R = Y + 1.371x(V - 128)$ $G = Y - 0.336x(U - 128) - 0.698x(V - 128)$ $B = Y + 1.732x(U - 128)$
1	<p>The RGB data has a range of 0-255, as is commonly found in computer systems, the following equations may be more convenient to use.</p> <p>The equation is used SD computer system.</p> $R = 1.164x(Y - 16) + 1.596x(V - 128)$ $G = 1.164x(Y - 16) - 0.391x(U - 128) - 0.813x(V - 128)$ $B = 1.164x(Y - 16) + 2.018x(U - 128)$
2	<p>The basic equations to convert between 8-bit digital RGB data width a 16-235 nominal range.</p> <p>This equation is used HDTV set (CCIR709 format)</p> $R = Y + 1.540x(V - 128)$ $G = Y - 0.183x(U - 128) - 0.459x(V - 128)$ $B = Y + 1.816x(U - 128)$
3	<p>The RGB data has a range of 0-255, as is commonly found in computer systems, the following equations may be more convenient to use.</p> <p>The equation is used HD computer system.</p> $R = 1.164x(Y - 16) + 1.793x(V - 128)$ $G = 1.164x(Y - 16) - 0.213x(U - 128) - 0.534x(V - 128)$ $B = 1.164x(Y - 16) + 2.115x(U - 128)$

Figure 5.13 YCbCr to RGB converter type

S2LC (Scaler To LCD Control Register)

0xF00000EC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

S2L

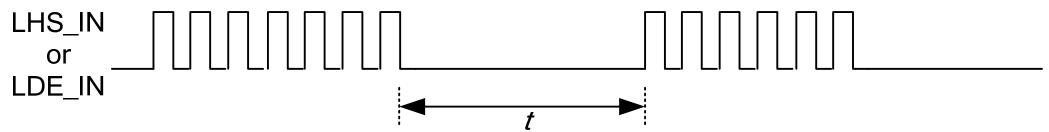
When MSEL of LCTRL is equal to 1, IMG0 is from the memory-to-memory scaler. If S2L is set to 0 when MSEL of LCTRL is 1, the LCDC can only accept this raw image source. If S2L is set to 1 when MSEL of LCTRL is 1, the LCDC can accept IMG1 and IMG2 from the memory as well as IMG0 from the memory-to-memory scaler. In this case, alpha-blending and chroma-keying among the raw image sources are available.

LPMCTRL (LCD PIP Mode Control)

0xF00000F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DC[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								PL	PS	FL	DS	DEP	VP	ENS	EN

FIELD	Description
Trigger Level of Vertical Sync Detector.	
DC [31:16]	If active horizontal sync is deasserted during over (DC x LCK_IN) cycles, the LCDC recognizes that vertical sync is asserted. It is only available when DS is 1.



If $t > DC * LCK_IN$ period,
it is recognized that vertical sync is asserted.

Figure 5.20 Vertical sync detector in PIP mode

PL [7]	PIP picture select
0	external picture in on-chip picture
1	The on-chip picture becomes the background.
0	on-chip picture in external picture
1	The external picture becomes the background.
PS [6]	PIP display mode
0	Chroma-key method
1	Windowing method
FL [5]	FIFO Flush
0	Normal Operation
1	FIFO is flushed.
DS [4]	Detect select
0	Vertical sync is inputted through LVS_IN
1	Vertical sync is detected by LHS_IN (ENS=1) or LDE_IN (ENS=0) using DC. Refer to DC field.
DEP [3]	Data Enable Polarity
0	LDE_IN (ENS=0) or LHS_IN (ENS=1) is "active high" signal.
1	LDE_IN (ENS=0) or LHS_IN (ENS=1) is "active low" signal.
VP [2]	VS polarity
0	LVS_IN is "active high" signal.
1	LVS_IN is "active low" signal.

ENS [1]		Data Enable Select
0		LDE_IN is used for active horizontal sync.
1		LHS_IN is used for active horizontal sync.
EN [0]		PIP Enable
0		Disable
1		Enable

LPMW0 (LCD PIP Mode Windowing 0) 0xF00000F4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
X1[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X2[15:0]															

LPMW1 (LCD PIP Mode Windowing 1) 0xF00000F8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Y1[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y2[15:0]															

LPMH (LCD PIP Mode Height) 0xF00000FC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HEIGHT[15:0]															

FIELD	Description
HEIGHT[15:0]	The number of LDE_IN cycles or LHS_IN cycles while sync is deasserted – 1.

Refer to Figure 5.8 on page 5-9 and the following figure.

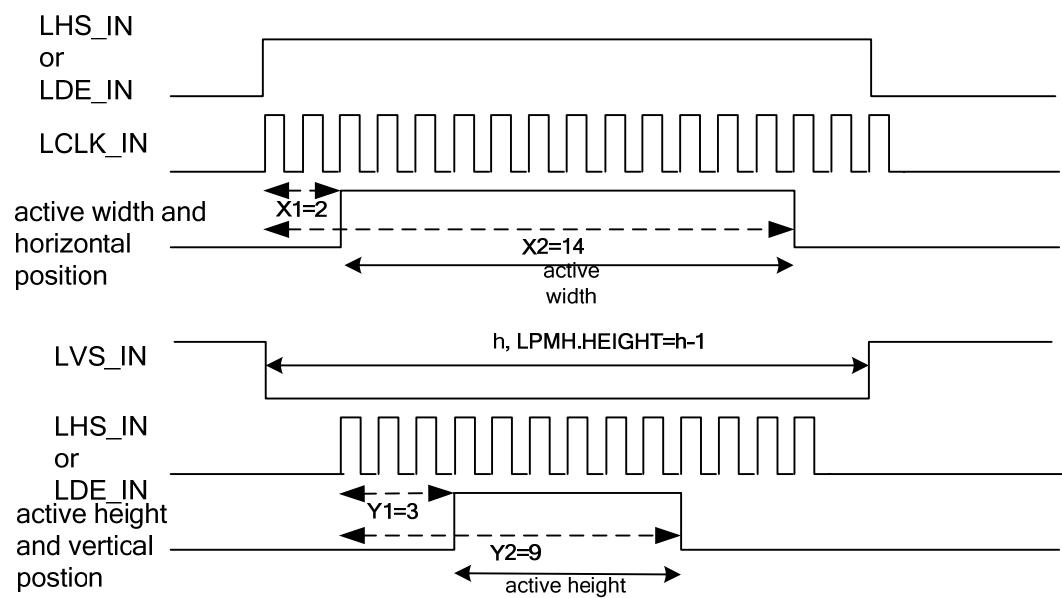


Figure 5.14 PIP position

LPMC (LCD PIP Mode Chroma-key)

0xF0000100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[23:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

LPMMC (LCD PIP Mode Mask Chroma-key)

0xF0000104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MKEY[23:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MKEY[15:0]															

The PIP chroma-keying is similar to the LCDC chroma-keying which is described on page 5-8.

6 LCD SYSTEM INTERFACE

6.1 Overview

The LCD system interface (LCDSI) is used to send out the image data from the system memory to the LCD module which has 68/80-system interface. The LCDSI can be accessed by both LCDC and the on-chip CPU. The setup time, hold time, and pulse width of the interface signals are programmable. It supports up to 18-bit data width.

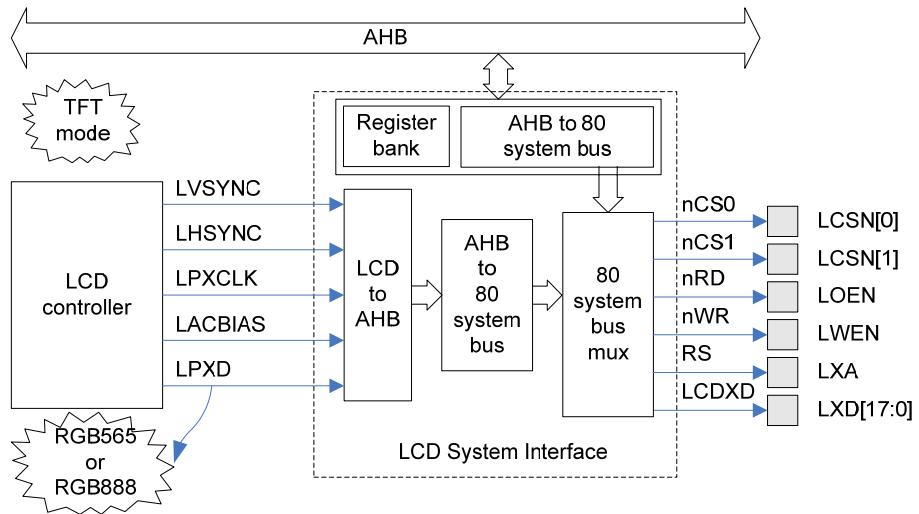


Figure 6.1 LCD system interface block diagram

6.2 Operation

Because the TCC79XX supports various GPIO modes, ports related to LCDSI must be configured before it is used. And LCDSI configuration determines whether it is connected to LCDC or on-chip CPU bus. Figure 6.2 shows its relationship simply. The following describes how to configure LCDSI for connecting to LCDC and on-chip CPU bus.

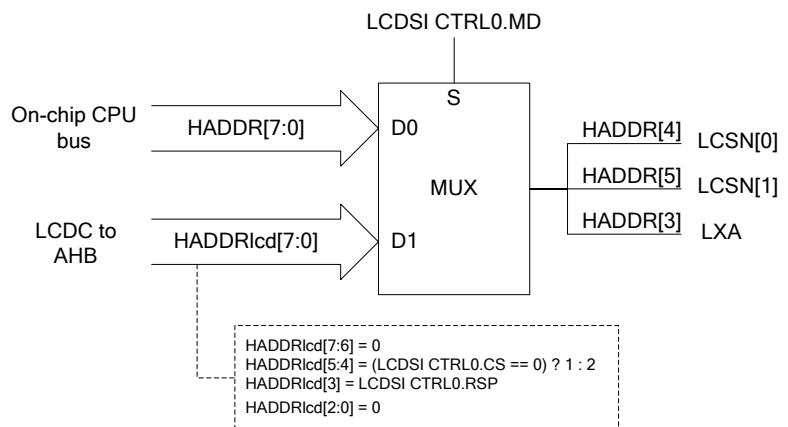


Figure 6.2 Relationship between LCSN/LXA and address

6.2.1 GPIO configuration for LCDSI

The ports which correspond to the LCDSI signals should be configured to the LCDSI mode. Refer to Port Multiplexer & GPIO on page 33-1.

6.2.2 Reading/Writing operation through the on-chip CPU

The LCDSI allows the on-chip CPU to read from and write to an external LCD module, which has 68/80-system interface. To access the device which is connected to LCSN[0] or LCSN[1], the on-chip CPU must set LCDSI CTRL0.IM to 0. After that, if the on-chip CPU accesses LCDSI CS0RS0 register, then reading or writing operations are generated on the device connected to LCSN[0]. While these operations are executed, LXA is low (Figure 6.3 (a)). If the on-chip CPU accesses LCDSI CS0RS1 register, LXA is high (Figure 6.3 (b)). Similarly, to access the device connected to LCSN[1], the on-chip CPU must access LCDSI CS1RS0 or LCDSI CS1RS1 register. Notice that LXA signal is irrelevant to LCDSI CTRL0.RSP when LCDSI CTRL0.IM = 0.

Timing and data width configuration about the LCDSI signals can be programmed via LCDSI CTRL1-4 registers. Refer to page 6-8 for more information about the LCDSI CTRL1-4 registers.

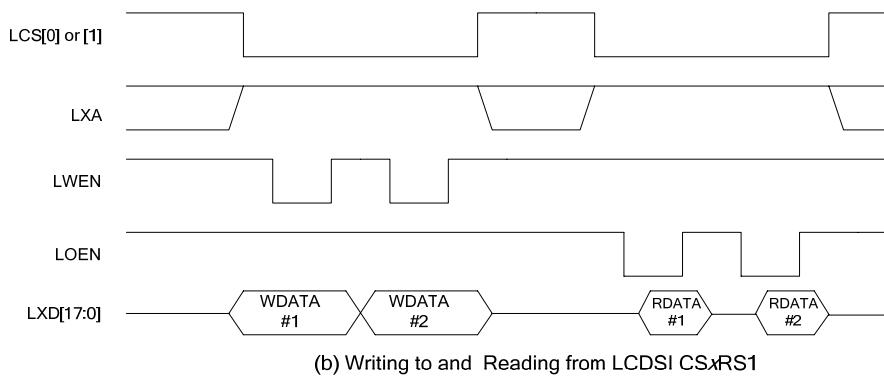
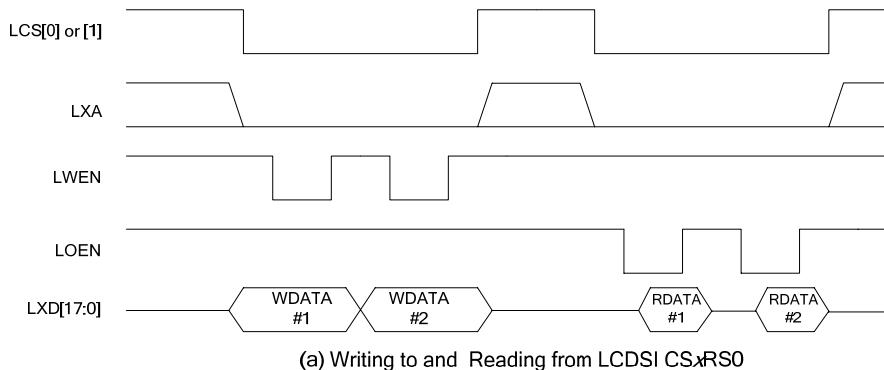


Figure 6.3 Writing / Reading operation through on-chip CPU

6.2.3 Writing operation through LCD Controller

For converting LCDC control signals to 68/80-system interface signals, the LCDC must be configured to TFT mode. And LPXD must be RGB565 or RGB888 and driven at negative edge of LPXCLK. LACBIAS and LVSYNC polarity is also configured by LCDSI CTRL0.IA and LCD CTRL0.IVS and the values must be same with those of LCDC polarity registers. Refer to LCDC register set for more information.

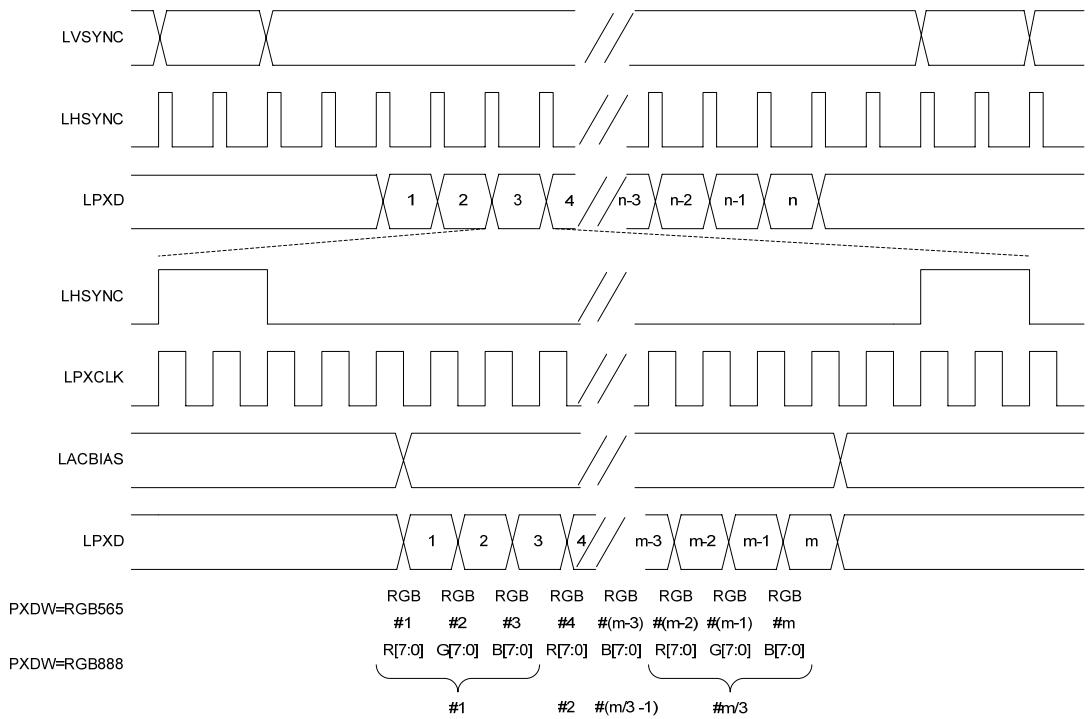


Figure 6.4 Example of LCDC output signals for LCDSI

To enter this mode, LCDSI CTRL0.IM must be set to 1. If this register field is set, LCDSI CTRL0.CS, CTRL0.RSP, CTRL0.FMT, CTRL0.IVS, and CTRL0.IA registers are available. These register fields specify nCS, RS, and output pixel data format during operation and polarity of LVSYNC and LACBIAS signal. For example, if an LCD module is connected to nCS0 and requires RS signal to be low, LCDSI CTRL0.CS and LCDSI CTRL0.RSP must be set to 0.

And LCDSI output signals can be adjusted by programming LCDSI CTRL1[31:16] register. Notice that these registers can be accessed when CTRL0.IM = 0. Therefore, timing parameters need to be set before CTRL0.IM is set to 1. LCDXD is dependent on LCDC LPXD, LCDSI CTRL0.FMT register and LCDSI CTRL1,2,3,4.WBW register. Refer to Figure about all of LCDXD formats which are supported by LCDSI.

The following is the procedure that one frame data of LCDC is sent to LCD module through LCDSI.

- (1) Set LCDSI CTRL0.IM=0.
- (2) Set LCDSI CTRL0 register except CTRL0.IM.
- (3) Set the timing parameters for LCD module via LCDSI CTRL1-4 registers.
- (4) Set LCDC register for one frame data.
- (5) Set LCDSI CTRL0.IM=1.
- (6) Enable LCDC and Disable it sequentially.

If LCDC is not disabled at 6, LCDC output data are sent to LCD module through LCDSI continuously.

6.3 Register Descriptions

All control registers for LCDSI are listed in Table 6.1.

Table 6.1 LCDSI Register map(0xF0000000)

Name	Address	Type	Reset	Description
LCDSI CTRL0	0x400	R/W	0x00000000	Control register for LCDSI
LCDSI CTRL1 ¹	0x800	R/W	0xA0229011	Control register for LCSN[0] when LXA=0 (for core access)
LCDSI CTRL2 ¹	0x804	R/W	0xA0429021	Control register for LCSN[0] when LXA=1 (for core access)
LCDSI CTRL3 ¹	0x808	R/W	0xA0129009	Control register for LCSN[1] when LXA=0 (for core access)
LCDSI CTRL4 ¹	0x80C	R/W	0xA0229011	Control register for LCSN[1] when LXA=1 (for core access)
LCDSI CS0RS0 ¹	0x810	R/W	-	If this register is read or written, reading or writing operations are generated on LCSN[0] while LXA = 0.
LCDSI CS0RS1 ¹	0x818	R/W	-	If this register is read or written, reading or writing operations are generated on LCSN[0] while LXA = 1.
LCDSI CS1RS0 ¹	0x820	R/W	-	If this register is read or written, reading or writing operations are generated on LCSN[1] while LXA = 0.
LCDSI CS1RS1 ¹	0x828	R/W	-	If this register is read or written, reading or writing operations are generated on LCSN[1] while LXA = 1.
LCDSI CTRL5	0x830	R/W	0xA0229011	Control register for LCSN[0] when LXA=0 (for lcd access)
LCDSI CTRL6	0x834	R/W	0xA0429021	Control register for LCSN[0] when LXA=1 (for lcd access)
LCDSI CTRL7	0x838	R/W	0xA0129009	Control register for LCSN[1] when LXA=0 (for lcd access)
LCDSI CTRL8	0x83C	R/W	0xA0229011	Control register for LCSN[1] when LXA=1 (for lcd access)

¹ These registers are only accessed by on-chip CPU when LCDSI CTRL0.IM is 0.

LCDSI CTRL0

0xF0000400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IA	IVS				0			CS	RSP			FMT		OM	IM

IM		Input Mode
0		It converts on-chip CPU signals to 68/80-system interface signals.
1		It converts LCDC output signals to 68/80-system interface signals. Before this bit is set, LCDC must be configured with TFT RGB565 or RGB888 mode. If TFT RGB565 mode is used, FMT[0] bit must be set to 0.
OM		Output Mode
0		LCDSI output signals are 80-system interface type.
1		LCDSI output signals are 68-system interface type. LOEN : R/W signal LWEN: Enable signal
FMT[3:0]		Pixel Data Format
0000		LCDC pixel data output : D1[15:0](RGB565) LCDSI pixel data output(8bits) : D1[7:0], D1[15:8] LCDSI CTRL1-4.WBW must be 1.
0110		LCDC pixel data output : D1[15:0](RGB565) LCDSI pixel data output (16bits) D1[15:0] LCDSI CTRL1-4.WBW must be 0.
0001		LCDC pixel data output : D1[7:0], D2[7:0], D3[7:0] (RGB888) LCDSI pixel data output (8bits): D1[7:0], D2[7:0], D3[7:0] LCDSI CTRL1-4.WBW must be 1.
0011		LCDC pixel data output : D1[7:0], D2[7:0], D3[7:0] (RGB888) LCDSI pixel data output (9bits) : {D1[7:2], D2[7:5]}, {D2[4:2], D3[7:2]} LCDSI CTRL1-4.WBW must be 0.
0101		LCDC pixel data output : D1[7:0], D2[7:0], D3[7:0] (RGB888) LCDSI pixel data output (16bits): D1[7:6], {D1[5:2], D2[7:2], D3[7:2]} LCDSI CTRL1-4.WBW must be 0.
0111		LCDC pixel data output : D1[7:0], D2[7:0], D3[7:0] (RGB888) LCDSI pixel data output (16bits): {D1[7:3], D2[7:2], D3[7:3]} LCDSI CTRL1-4.WBW must be 0.
1000		LCDC pixel data output : D1[17:0] (RGB666) LCDSI pixel data output (16bits): {D1[17:9]}, {D1[8:0]} LCDSI CTRL1-4.WBW must be 0.
1010		LCDC pixel data output : D1[17:0] (RGB666) LCDSI pixel data output (16bits): {D1[17:2]}, {D1[1:0]} LCDSI CTRL1-4.WBW must be 0.
1100		LCDC pixel data output : D1[17:0] (RGB666) LCDSI pixel data output (186bits): {D1[17:0]} LCDSI CTRL1-4.WBW must be 0

FMT bits are used when IM bit is set to 1.

LCD SYSTEM INTERFACE

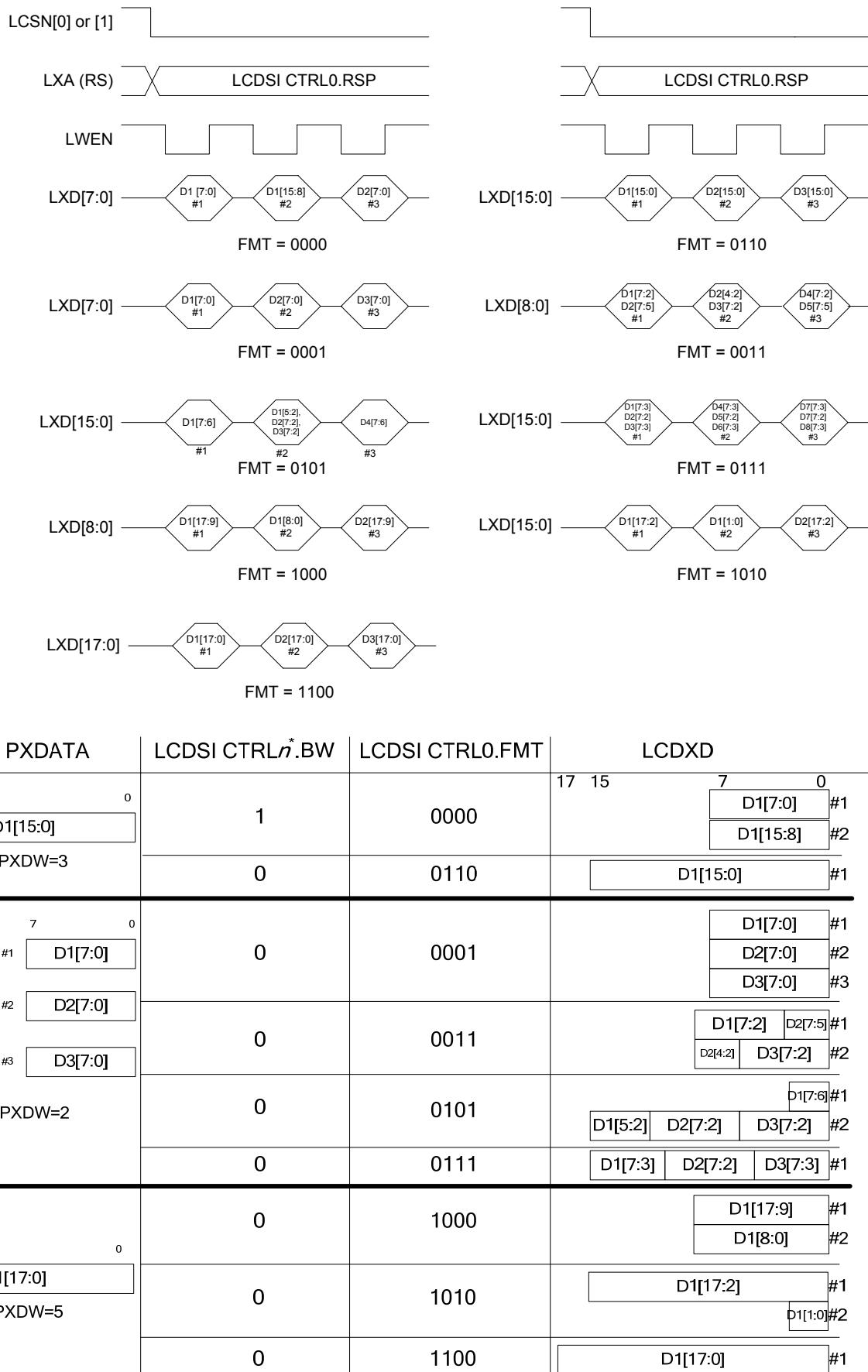
*) $n = 5, 6, 7, \text{ or } 8$

Figure 6.5 LCDSI output pixel data organization according to FMT

RSP	RS Polarity
0	If IM bit is set to 1, LXA (RS) is low while LCSN[0] or LSN[1] is asserted. Otherwise, it is not applicable.
1	If IM bit is set to 1, LXA (RS) is low while LCSN[0] or LSN[1] is asserted. Otherwise, it is not applicable.
CS	Chip select
0	If IM bit is set to 1, LCSN[0] is used. Otherwise, it is not applicable.
1	If IM bit is set to 1, LCSN[1] is used. Otherwise, it is not applicable.
IA	Inverse ACBIAS
0	LACBIAS(Data Enable) signal of LCDC is active high.
1	LACBIAS(Data Enable) signal of LCDC is active low.
IVS	Inverse VSYNC
0	LVSYNC signal of LCDC is active high.
1	LVSYNC signal of LCDCL is active low.

FMT, RSP, CS, IA and IVS are only used when IM bit is set to 1.

LCDSI CTRL1 – 4

0xF0000800, 0xF0000804, 0xF0000808, 0xF000080C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BW[1]	W ² STP				WPW								WHLD		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BW[0]	R ³ STP				RPW								RHLD		

These registers are only available when the LCDSI is accessed by the on-chip CPU, which means that LCTRL0.IM is set to 0.

BW[1:0]	Bus Width
0	LCDSI CSnRSm registers become 8-bit memory mapped I/O devices. Therefore, when the on-chip CPU accesses them with a 32-bit data operation, LCDSI generates 4 8-bit data operation.
1	LCDSI CSnRSm registers become 16-bit memory mapped I/O devices. Therefore, when the on-chip CPU accesses them with a 32-bit data operation, LCDSI generates 2 16-bit data operation.
2,3	LCDSI CSnRSm registers become 32-bit memory mapped I/O devices. Therefore, when the on-chip CPU accesses them with a 32-bit data operation, LCDSI generates 2 32-bit data operation. But, upper 14bits are truncated because LXD is 18bits.
STP	Setup Time
N	N cycles are issued between the falling edge of LCSN and the falling edge of LWEN (Writing operation) or LOEN(Reading operation).
PW	Pulse Width
N	(N+1) cycles are issued between the falling edge of LWEN (or LOEN) and the rising edge of LWEN (or LOEN).
HLD	Hold Time
N	N cycles are issued between the rising edge of LWEN (or LOEN) and the rising edge of LCSN.

² Prefix W means writing operation.

³ Prefix R means reading operation.

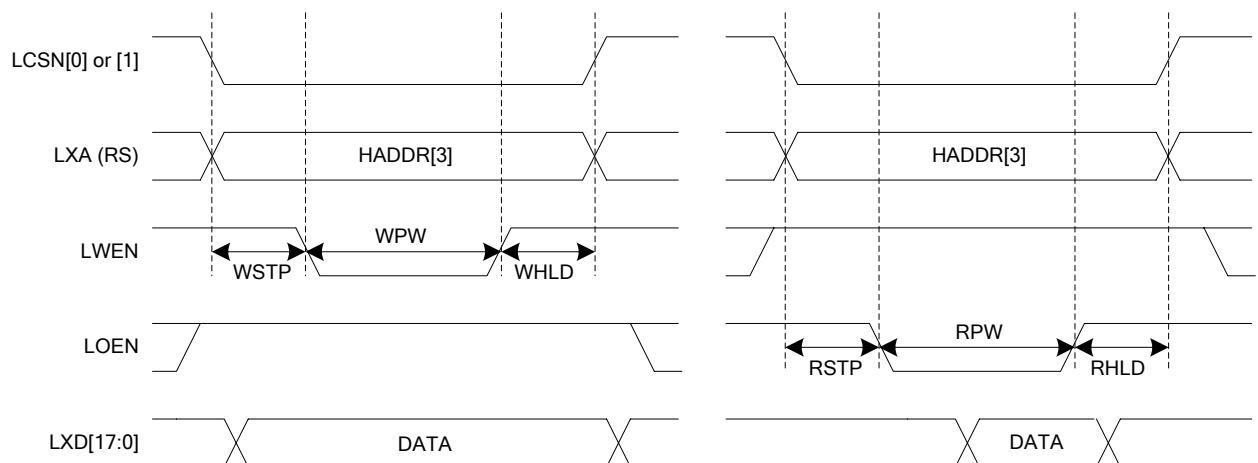


Figure 6.6 Timing configuration of LCDSI output signals for the on-chip CPU access

LCDSI CTRL5 – 7

0xF0000830, 0xF0000834, 0xF0000838, 0xF000083C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BW	WSTP				WPW				WHLD						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved

These registers are only available when the input of LCDSI is the output of LCDC, which means that LCTRL0.IM is set to 1. In this case, writing operations are only available. Figure 6.2 shows the timing information of control signals and related registers.

BW	Bus Width
0 or 1	BW and LCTRL0.FMT determine LXD format. Refer to Figure 6.5 on page 6-6.
WSTP	Setup Time
N	N cycles are issued between the falling edge of LCSN and the falling edge of LWEN (Writing operation).
WPW	Pulse Width
N	(N+1) cycles are issued between the falling edge of LWEN and the rising edge of LWEN.
WHLD	Hold Time
N	N cycles are issued between the rising edge of LWEN and the rising edge of LCSN.

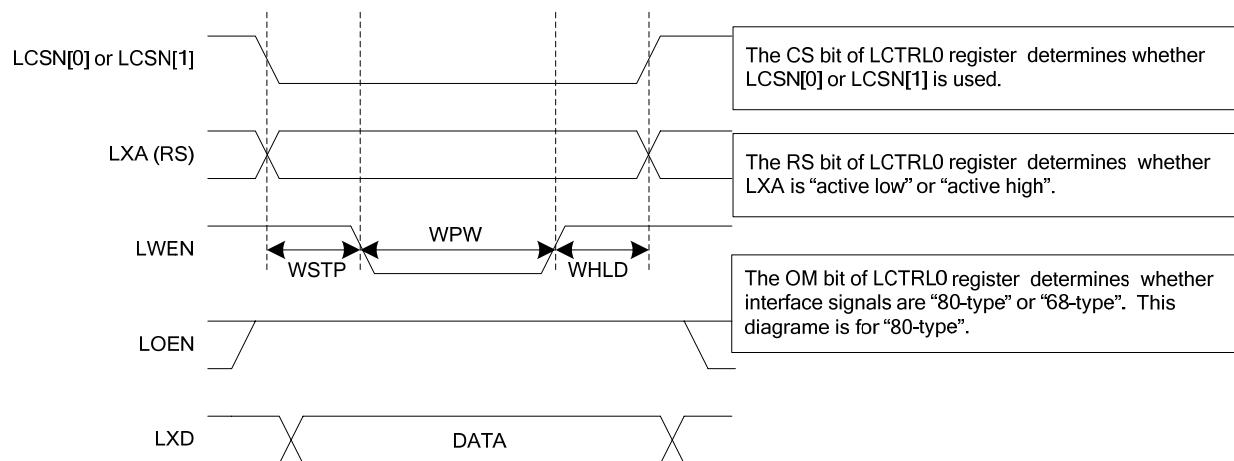


Figure 6.7 Timing configuration of LCDSI output signals for the LCDC

7 NTSC / PAL ENCODER COMPOSITE OUTPUT

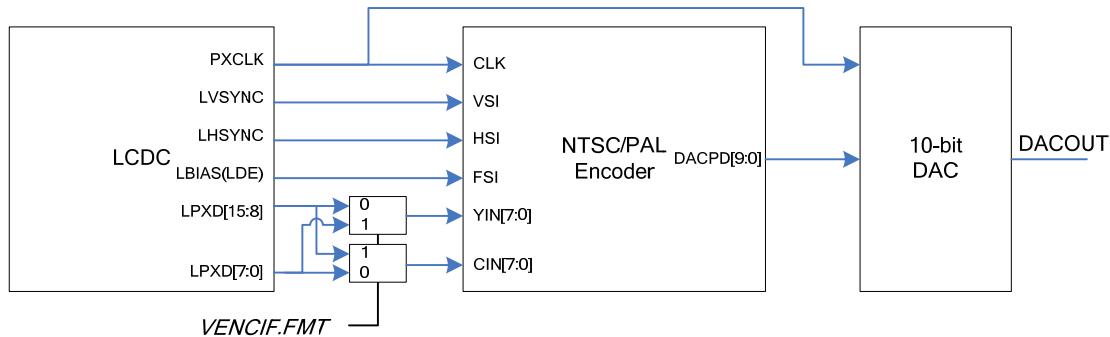
7.1 Overviews

The NTSC/PAL encoder meets all requirements of the ITU -R BT.470-3 specifications. This is designed to support all standards and variations of the NTSC and PAL encoding systems. This includes cross standards pseudo PAL and pseudo NTSC. This allows independent control of field rate, chroma subcarrier and the chroma encoding algorithm. Both the luma and chroma bandwidths can be varied to optimize for various data input conditions. Because input signals are generated by "LCD Interface", LCD can not be displayed while NTSC/PAL composite output is enabled.

7.2 Features

- Encoding at square pixel or CCIR601 rates including flexible support for multiple clock frequencies
- Programmable Luma and Chroma bandwidth
- Programmable Saturation, Hue, Contrast and Brightness
- Supports all NTSC and PAL formats
- NTSC-M/4.43, PAL-B/D/G/H/I/M/N/Combination N
- 8 bit or 16bit YUV input
- ITU-R BT.601 4:2:2 16-bit Parallel Input
- ITU-R BT.656 Parallel Input Format
- Controlled entry and exit of active video pixel and line
- Composite outputs.
- Outputs 10 bit to DAC
 - Maximum conversion rate : 27MSPS
 - Output Load Resistance : 37.5 ohm
 - Analog Output Range : 0.0 – 1.3V(Typical)

7.3 Selecting Operation Mode



To use the NTSC/PAL encoder, the LCDC should be configured as NTSC/PAL interface and output timing should be configured as the corresponding input timing of the NTSC/PAL encoder.

The NTSC/PAL encoder can operate at square pixel or 601 sampling rates for 50Hz or 60Hz standards. This is controlled using the **PIXSEL** and **IFMT** bits. Selecting the operating mode changes various internal timing locations to ensure that the final analog output signal meets appropriate standards. The operation frequencies and related information are described below.

DESCRIPTION	SYMBOL	601 Sampling Clock		Square Pixel Clock	
		60Hz Field	50Hz Field	60Hz Field	50Hz Field
DAC Operation FREQ	F_{CK27}	27MHz	27MHz	24.5454MHz	29.5MHz
Pixel Data Rate	F_{CK13}	13.5MHz	13.5MHz	12.2727MHz	14.75MHz
Number of Clocks per Line	T_{HS}	1716	1728	1560	1888
Number of CK13(CK27) clocks for active video (Encoded pixels)	T_{AV}	720 (1440)	720 (1440)	640 (1280)	768 (1536)
Number of Lines per Field	L_{VS}	262.5	312.5	262.5	312.5
Default numbers of CK13 clocks from HS active edge to first input data	T_{FPX}	244	262	238	304

7.4 Input Timing

The 3 signals HSI, VSI and FSI are used to define the horizontal, vertical and field reference points. The **ISYNC[2:0]** configuration register is used to select timing configurations. Once a timing mode is selected, each input has a polarity control (**HSIP**, **VSIP** and **FSIP**). These are used to define the “active” edge of the input sync. Finally, by programming the **HOFFSET** and **VOFFSET** bits, the desired pixel and line location for reset at the “active” edge is selected.

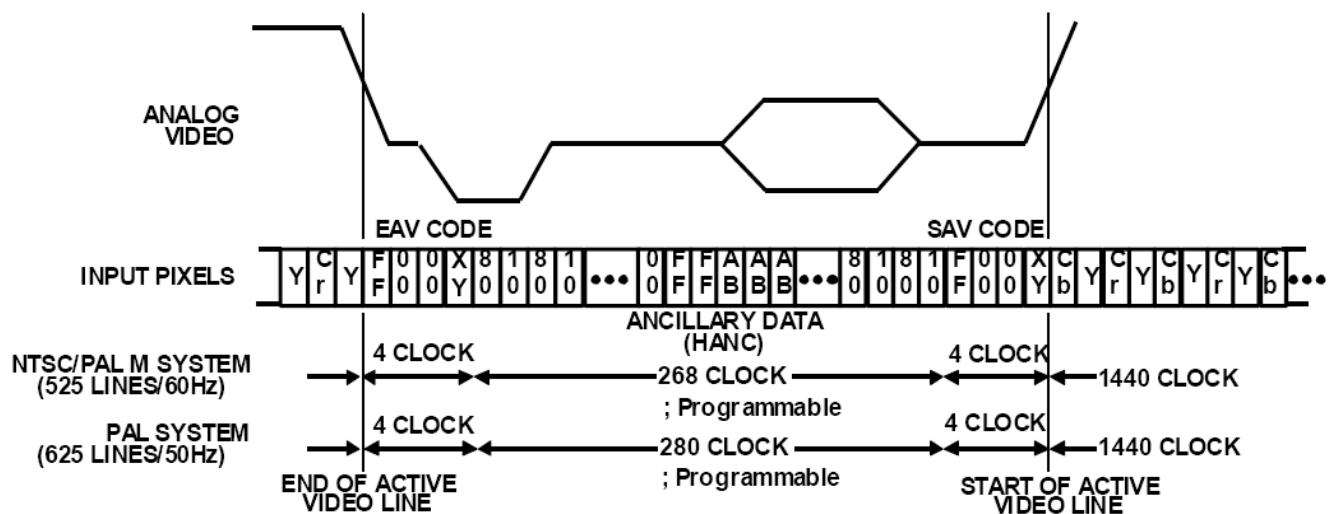


Figure 7.1 Digital Input Timing (ITU-R BT.656 8bit parallel Input)

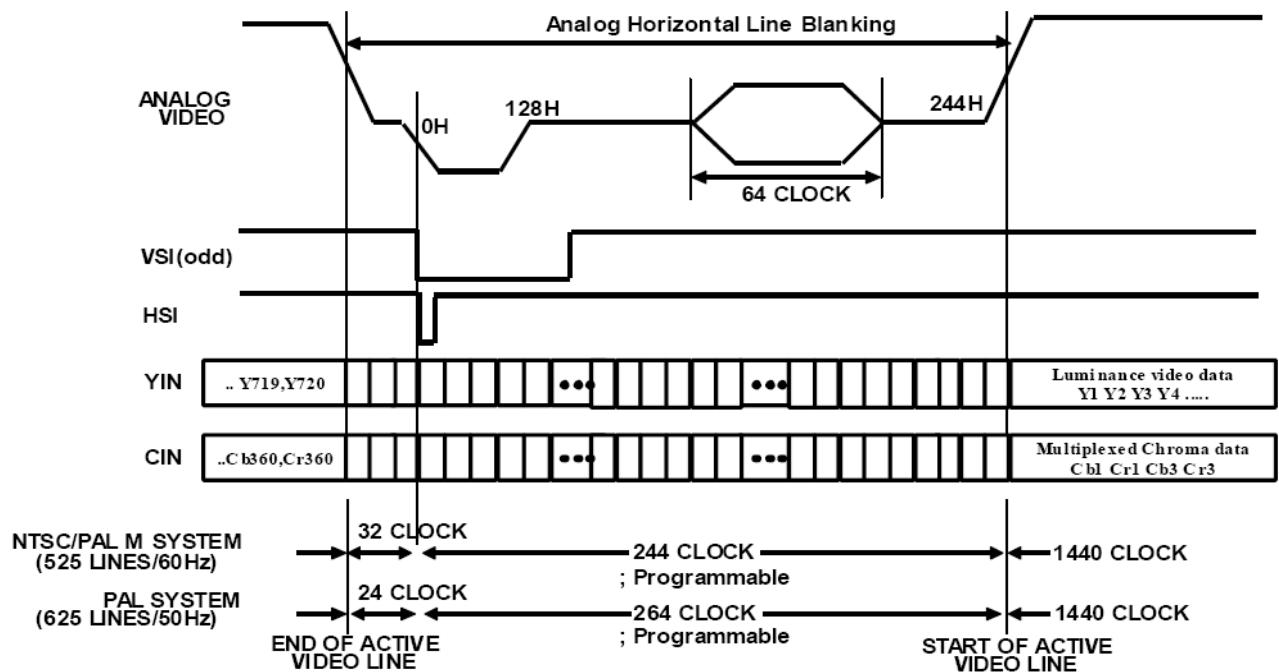


Figure 7.2 Digital Input Timing (ITU-R BT.601 4:2:2 16bit Parallel Input)

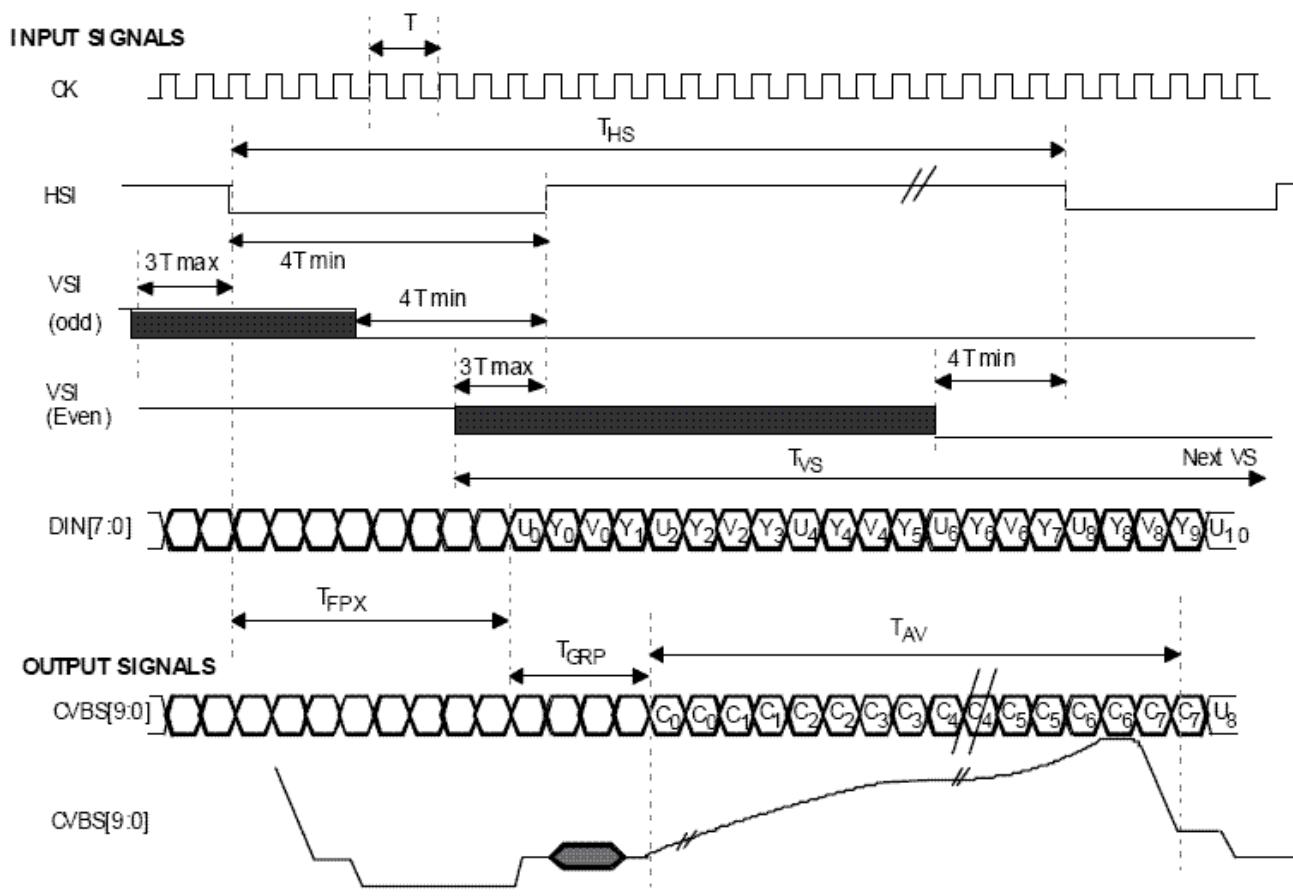


Figure 7.3 Example of Input Timing

7.5 Video Standard Selection

The NTSC/PAL encoder supports all NTSC and PAL standards throughout the world. The encoder supports independent control of the Chroma modulation frequency, selection of phase alternating Line encoded chroma, and field frequency. In addition, for non SCH locked standards, the Chroma can be allowed to free run using the **FDRST**. If the **FSCADJ** register is set to any value other than '0', the **SCH** relationship will not be able to be maintained. For these setting it is also recommended that the **FDRST** bit be set to free run mode.

Table 7.1 Setting for Output Formats

Requested output Format			Required Encoder Settings				
Format	Field Rate	FSC	IFMT	FSCSEL	PHALT	FDRST	PED
NTSC M	59.94Hz (525)	3.5795454	0	0	0	1	1/0
NTSC N	50 Hz (625)	3.5795454	1	0	0	0	1/0
PAL M	59.952 (525)	3.57561189	0	2	1	0	
PAL N	50 Hz (625)	3.58205625	1	3	1	0	
PAL B/G/H/I	50 Hz (625)	4.43361875	1	1	1	1	0
Pseudo PAL	60 Hz (525)	4.43361875	0	1	1	0	
Pseudo NTSC	50 Hz (625)	3.5795454	1	0	0	0	
NTSC 4.43	60Hz (525)	4.43361875	0	1	0	0	

7.6 Basic Video Adjustments

The standard video adjustments for Chroma and Luma are included. Chroma controls include Saturation and Hue (**SAT HUE**). Luma adjustments include Contrast and Brightness (**CONTBRIGHT**). An additional **SCH** control is included that changes the Chroma subcarrier phase relative to the horizontal sync. Note that this feature operates only when **FDRST** = 0.

7.7 Programmable Bandwidth

The data bandwidth for the Luma and chroma paths is shown in the following frequency plots. The YBW control allows 3 different settings to optimize the output bandwidth to the receiver. The same applies to chroma bandwidth using the CBW control.

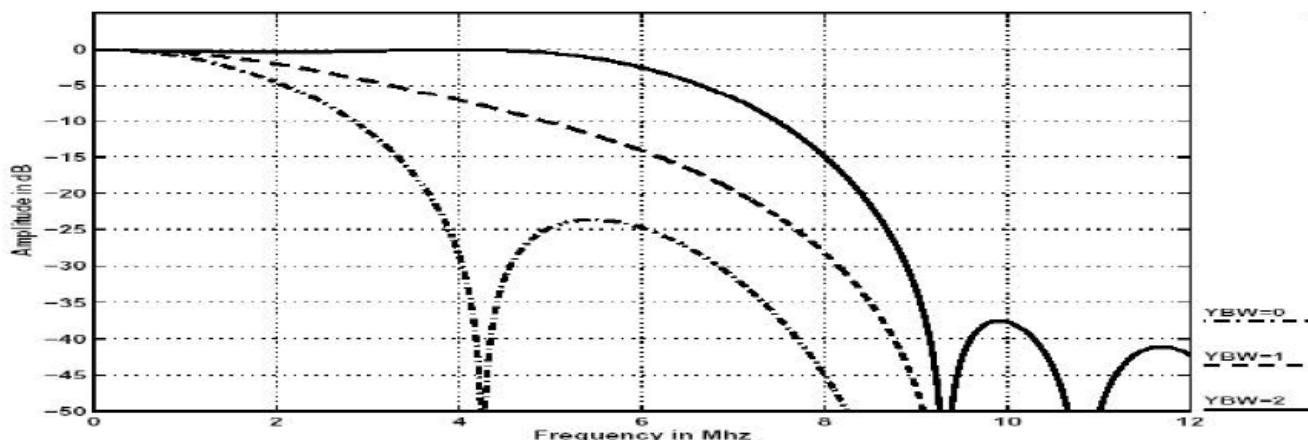


Figure 7.4 Luma Bandwidth

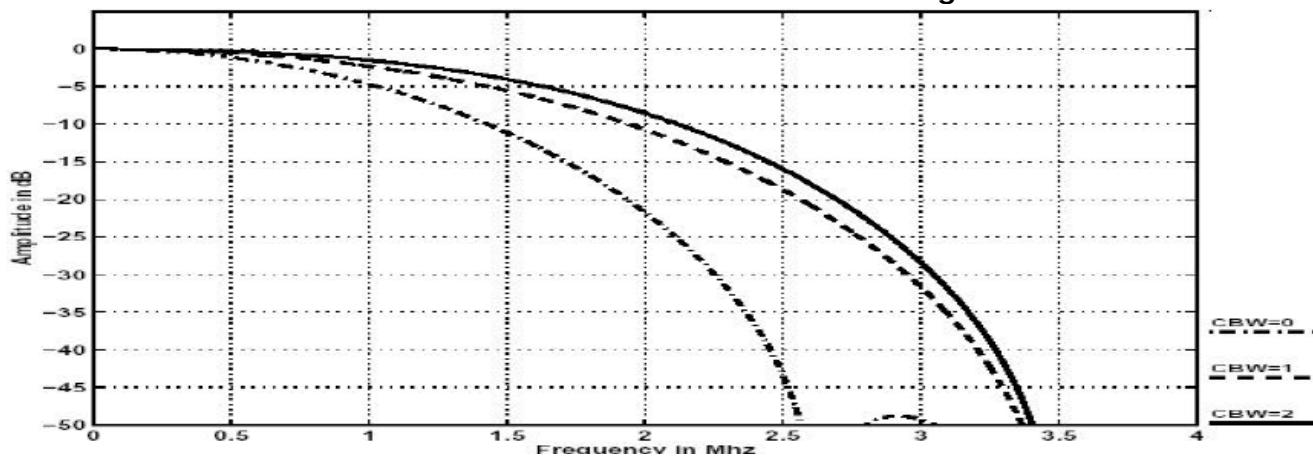


Figure 7.5 Chroma Bandwidth

7.8 Analog Video Output Configuration

The TCC79XX supports composite video. Refer to output configuration register DACSEL for further details. The DAC output levels and the associated digital codes are summarized below. DAC voltage assumes the standard 140IRE = 1v. Numbers are shown for NTSC type video with a pedestal.

Table 7.2 Summary of DAC voltage and Codes

Signal Level	CVBS / LUMA Value	IRE Value	DAC Voltage
Max output	1023	137.2	1.282v
100% White	810	100	1.015v
Black	282	7.37	353mv
Sync	12	-40	15mv
White - Blank	570	100	714mv delta
White - sync	798	140	1v delta
Color burst	228	40	285mv delta

7.9 Registers

STATA

0xF9000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVID										FIELD					

BIT	NAME	R/W	RESET	DESCRIPTION
7-5	REVID	R	0	Current Revision Identification Number The current video field
2-0	FIELD	R	-	0-3 or 4-7 for NTSC fields 1-4 0-7 for PAL fields 1-8

ECMDA

0xF9000004

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								PWDENC	FDRST	FSCCELL	PED	PIXEL	IFMT	PHALT	

BIT	NAME	R/W	RESET	DESCRIPTION
				Power Down Mode
7	PWDENC	R/W	0	0: Normal Operation 1 Power down mode for the entire digital logic circuits of encoder digital core.
				SCH Chroma-Luma locking control
6	FDRST	R/W	0	0: Constant relationship between color burst and horizontal sync maintained for appropriate video standards 1: Chroma is free running as compared to horizontal sync
				Modulation frequency of chroma output
5-4	FSCSEL	R/W	0	0 Color subcarrier frequency = 3.57954545 MHz for NTSC 1 Color subcarrier frequency = 4.43361875 MHz for PAL - B,D,G,H,I,N 2 Color subcarrier frequency = 3.57561149 MHz for PAL - M 3 Color subcarrier frequency = 3.58205625 MHz for PAL - combination N
				Define input pedestal format
3	PED	R/W	0	0 Video output has no pedestal*. 1 Video output has a pedestal.
				Select Pixel sampling rate.
2	PIXEL	R/W	0	0 Input data is at 601 rates. * 1 Input data is at square pixel rates.
				Format of Output Data
1	IFMT	R/W	0	0 525 Lines 1 625 Lines
				Phase Alternate control for PAL encoded chroma signal output
0	PHALT	R/W	0	0 NTSC encoded color 1 PAL encoded color

ECMDB

0xF9000008

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										YBIBLK	CBW	YBW			

BIT	NAME	R/W	RESET	DESCRIPTION
VBI Blanking control				
4	VBIBLK	R/W	0	0 Input data is passed through for non vbi processing lines 1 Video data is forced to being black level for vertical non vbi processed lines.
Chroma Bandwidth control				
3-2	CBW	R/W	0	0 Low bandwidth 1 Medium bandwidth 2 High bandwidth 3 Not used, default to low bandwidth
Luma Bandwidth control				
1-0	YBW	R/W	0	0 Low bandwidth 1 Medium bandwidth 2 High bandwidth 3 Not used, default to low bandwidth

GLK

0xF900000C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										XT24	GLKEN	GLKE			

BIT	NAME	R/W	RESET	DESCRIPTION
24MHz Crystal input when high				
4	XT24	R/W	0	0 27MHz Crystal input 1 24MHz Crystal input
Genlock reset control				
3	GLKEN	R/W	0	0 Genlock reset disable 1 Genlock reset enable
Chroma Fsc Generation frequency control				
2-1	GLKE	R/W	0	0,1 Chroma Fsc is generated from internal constants based on current user setting. 2 Fsc is adjusted based on external RTCO input 3 Fsc tracks non standard Encoder clock (CLKI) programmed frequencies.
Genlock mode PAL ID Polarity control				
0	GLKPL	R/W	0	0 PAL bit input polarity is active high 1 PAL bit input polarity is active low
Phase Alternate control for PAL encoded chroma signal output				
0	PHALT	R/W	0	0 NTSC encoded color 1 PAL encoded color

SCH

0xF9000010

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCH															

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	SCH	R/W	0	Programs the Color burst phase relative to the sync tip. '0' is the nominal value. The 8 bit control covers the entire 360 range as a 2's compliment number.

HUE

0xF9000014

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HUE															

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	HUE	R/W	0	Programs the Active video Color burst phase relative to color burst. The 8 bit control covers the entire 360 range as a 2's compliment number.

SAT

0xF9000018

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAT															

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	SAT	R/W	0	Controls the Active video Chroma gain relative to the color burst gain. Value is 2's compliment with '0' the nominal value.

CONT

0xF900001C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONT															

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	CONT	R/W	0	Controls Luma gain. Value is 2's compliment with '0' the nominal value.

BRIGHT

0xF9000020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCH															

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	BRIGHT	R/W	0	Controls Luma offset. Value is 2's compliment with '0' the nominal value.

FSC_ADJM

0xF9000024

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSC_ADJM															

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	FSC_ADJM	R/W	0	FSCADJ[15:8]

FSC_ADJL

0xF9000028

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSCADJ[7:0]															

BIT	NAME	R/W	RESET	DESCRIPTION
7-0	FSC_ADJL	R/W	0	FSCADJ[7:0]

FSCADJ[7:0] allows the pixel clock to be varied up to +/- 200ppm of its nominal value. This allows dot crawl adjustment. This 16 bit signal is multiplied by 4 and added to the internal chroma frequency constant.

ECMDC

0xF900002C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CSMDE	CSMD	RGBSYNC					

BIT	NAME	R/W	RESET	DESCRIPTION
Composite sync CSYN Pin tri-state control				
7	CSMDE	R/W	0	0 Composite sync mode not enable - Pin is tri-stated 1 Composite sync mode enable - Output reflexes mode defined by CSMD
Composite sync CSYN Pin output control				
6-5	CSMD	R/W	0	0 Composite Sync 1 Keyed clamp 2 Keyed pulse 3 N/A
Enable RGBSYNC when output is configured for analog EGB modes				
4-3	RGBSYNC	R/W	0	0 No sync on the RGB output signals 01 Sync on the RGB output signals 10 Sync on the G output signal

DACSEL

0xF9000040

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACSEL															

BIT	NAME	R/W	RESET	DESCRIPTION
7-4	-	R/W	0	SHOULD BE ZERO Data type output selection for DAC.
3-0	DACSEL	R/W	0	0: DAC digital output is disabled, Output is code 0. 1: Data output in CVBS format – Composite Video others : N/A

DACP0

0xF9000050

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD															

BIT	NAME	R/W	RESET	DESCRIPTION
7-1	-	R/W	0	SHOULD BE ZERO
0	PD	R/W	0	0 DAC Normal Operation 1 DAC Power down

ICNTL

0xF9000080

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	FSIP	VSIP	HSIP	HSVSP	VSMD	ISYNC
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	------	------	-------	------	-------

BIT	NAME	R/W	RESET	DESCRIPTION
7	FSIP	R/W	0	Field polarity. It controls the polarity of any F or Field control as input or output. 0: Odd field (fields 1,3,5,7) active low. 1: Odd field active high.
6	VSIP	R/W	0	Vertical field polarity 0: Vertical sync active low. 1: Vertical sync active high.
5	HSIP	R/W	0	Horizontal sync polarity 0: Horizontal sync active low. 1: Horizontal sync active high.
4	HSVSP	R/W	0	Horizontal and vertical sync latch enable 0: Enable to latch on the falling edge of VS and HS . 1: Enable to latch on the rising edge of VS and HS.
3	VSMD	R/W	0	Vertical Sync Output format 0: Odd Field VS and FS outputs aligned to video line Start. Even Field VS and FS output aligned to video line midpoint. 1: Odd Field VS and FS outputs aligned to video line Start. Even Field VS and FS outputs aligned to video line Start.
2-0	ISYNC	R/W	0	Timing configuration for Horizontal, Vertical and Field inputs. 0, 1, 3, and 7: The LCDC does not support this mode. 2: Alignment input format from pins HSI and VSI. F --- Field timing from the latched value of HSI and VSI rising or falling edge. V --- Vertical timing from the VSI pin rising or falling edge. H --- Horizontal timing from the HSI pin rising or falling edge. 4: Alignment input format from pin VSI Only. Note that since there is no field information in this mode, the software counter reset is required for proper field identification. F --- Field timing from a software reset only. V --- Vertical timing from the VSI pin rising or falling edge. H --- Horizontal timing from the VSI pin rising or falling edge. 5: Alignment is from embedded EAV,SAV codes (line by line). F --- Field timing from the SAV/EAV 'F' bit. V --- Vertical timing from the SAV/EAV 'V' bit. H --- Horizontal timing from the SAV/EAV 'H' bit. 6: Alignment is from embedded EAV,SAV codes (frame by frame). F --- Field timing from the SAV/EAV 'F' bit.

V --- Vertical timing from the SAV/EAV 'F' bit.

H --- Horizontal timing from the SAV/EAV 'F' bit.

HVOFFST

0xF9000084

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSEL				VOFFST				HOFFSET[10:8]							

BIT	NAME	R/W	RESET	DESCRIPTION
Digital input format select.				
7-6	INSEL	R/W	0	0 Input format is 16 bit YUV 4:2:2 data Sampled at 27MHz. 1 Input format is 16 bit YUV 4:2:2 data Sampled at 13.5MHz . 2 Input format is 8 bit YUV 4:2:2 data 27MHz data Samples at 13.5MHz. 3 Input is from TESTI port, same format as mode 2. (TEST MODE)
Vertical offset bit 8. Refer to VOFFSET register.				
3	VOFFST	R/W	0	VOFFSET[8:0] determines Vertical alignment point for the VSI input. Programmed value is the "line number" that pixel counter is reset to when a VSI transition occurs in input.
Horizontal offset bit 8 ~ 10. Refer to HOFFSET register.				
2-0	HOFFST	R/W	0	HOFFSET[10:0] determines Horizontal alignment point for the HSI input. Programmed value is the "pixel number" that pixel counter is reset to when a HIS transition occurs in input.

HOFFST

0xF9000088

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HOFFSET[7:0]															

Refer to the HVOFFST register description.

VOFFST

0xF900008C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VOFFSET[7:0]															

Refer to the HVOFFST register description.

HSVSO

0xF9000090

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															

HSOE

0xF9000094

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															

HSOB

0xF9000098

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															

VSOB

0xF900009C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															

VSOE

0xF90000A0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															

VENCON

0xF9000800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															

BIT	NAME	RESET	DESCRIPTION
31-1			RESERVED
0	EN	0	It determines whether output signals of the LCDC are connected to the NTSC/PAL encoder. 0: disable 1: Enable (connection)

VENCIF

0xF9000804

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MV FMT															

BIT	NAME	RESET	DESCRIPTION
31-2			RESERVED
1	MV	0	RESERVED
0	FMT	0	It determines data bus connection type between LCDC and NTSC/PAL encoder. 0: LCDC PXDATA[7:0] is connected to NTSC/PAL CIN[7:0] LCDC PXDATA[15:8] is connected to NTSC/PAL YIN[7:0] 1: LCDC PXDATA[7:0] is connected to NTSC/PAL YIN[7:0] LCDC PXDATA[15:8] is connected to NTSC/PAL CIN[7:0]

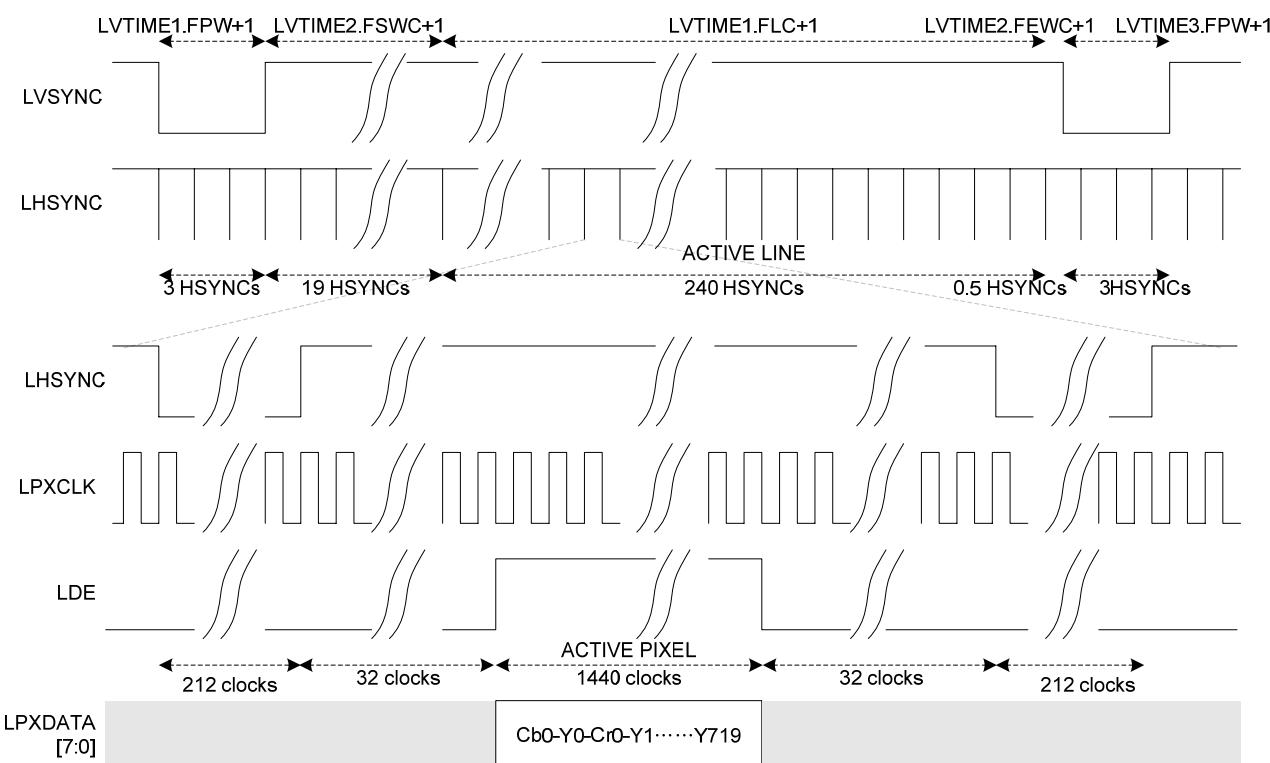
Example of NTSC/PAL interface

We will strongly recommend you to use the values in the tables.

NTSC-M interface

The following tables show how to configure the LCDC and the NTSC/PAL encoder for NTSC. PXCLK of the LCDC should be configured as 27MHz.

LCDC register configuration for NTSC-M	
LDS.HSIZE = 720	LCTRL.TV = 1
LDS.VSIZE = 480	LCTRL.NI = 0
LHTIME1.LPC = 720 * 2 - 1	LCTRL.DP = 0
LHTIME1.LPW = 212 - 1	LCTRL.PXDW = 6
LHTIME2.LSWC = 32 - 1	LCTRL.IH = 1
LHTIME2.LEWC = 32 - 1	LCTRL.IV = 1
LVTIME1.FLC = 480 - 1	LCTRL IP = 1
LVTIME1.FPW = 6 - 1	
LVTIME2.FSWC = 38 - 1	
LVTIME2.FEWC = 1 - 1	
LVTIME3.FLC = 480 - 1	
LVTIME3.FPW = 6 - 1;	
LVTIME4.FSWC = 37 - 1	
LVTIME4.FEWC = 2 - 1	



NTSC/PAL encoder register configuration for NTSC-M

DACPD.PD = 0 (not power down)	HOFFST.HOFFSET = 0
ECMDA.PWDENC = 0	VOFFST.VOFFSET =1
ECMDA.FDRST = 1	
ECMDA.PHALT = 0	HVOFFST.HOFFSET = 0
ECMDA.IFMT = 0	HVOFFST.VOFFSET = 0
ECMDA.PIXSEL = 0	HOFFSET.INSEL = 2
ECMDA.PED = 1	
ECMDA.FSCSEL = 0	
ECMDB.VBIBLK = 1	
DACSEL.DACSEL = 1 (composite output)	
ICNTL.ISYNC = 2	
ICNTL.HSIP = 0	
ICNTL.VSIP = 1	
ICNTL.FSIP = 1	
VENCIF.FMT = 1	
VENCON.EN = 1	

PAL-B/G/H/I interface

The following tables show how to configure the LCDC and the NTSC/PAL encoder for PAL-B/G/H/I. PXCLK of the LCDC should be configured as 27MHz.

LCDC register configuration for PAL B/G/H/I	
LDS.HSIZE = 720	LCTRL.TV = 1
LDS.VSIZE = 576	LCTRL.NI = 0
	LCTRL.DP = 0
LHTIME1.LPC = 720 * 2 - 1	LCTRL.PXDW = 6
LHTIME1.LPW = 280 - 1	LCTRL.IH = 1
	LCTRL.IV = 1
LHTIME2.LSWC = 2 - 1	LCTRL IP = 1
LHTIME2.LEWC = 6 - 1	
LVTIME1.FLC = 576 - 1	
LVTIME1.FPW = 0	
LVTIME2.FSWC = 43 - 1	
LVTIME2.FEWC = 5 - 1	
LVTIME3.FLC = 576 - 1	
LVTIME3.FPW = 0;	
LVTIME4.FSWC = 44 - 1	
LVTIME4.FEWC = 4 - 1	

NTSC/PAL encoder register configuration for PAL B/G/H/I

DACP.D.PD = 0 (not power down) HOFFST.HOFFSET = 0

ECMDA.PWDENC = 0 VOFFST.VOFFSET =1

ECMDA.FDRST = 1

ECMDA.PHALT = 1

HVOFFST.HOFFSET = 0

ECMDA.IFMT = 1

HVOFFST.VOFFSET = 0

ECMDA.PIXSEL = 0

HVOFFST.INSEL = 2

ECMDA.PED = 0

ECMDA.FSCSEL = 1

ECMDB.VBIBLK = 1

DACSEL.DACSEL = 1 (composite output)

ICNTL.ISYNC = 2

ICNTL.HSIP = 0

ICNTL.VSIP = 1

ICNTL.FSIP = 1

VENCIF.FMT = 1

VENCON.EN = 1

7.10 Copy Generation Management Systems

Copy Generation Management Systems (CGMS) is supported for 525/60 for Japan. Line 20 and 283 are used to transmit the CGMS data when the CGMS function is enabled. The CGMS consists of a 2 bit start code and a 20 bit data. The 20 bits data, including 6 bits of Cyclic Redundancy Check, are loaded into the CGMS register via the host interface prior to being transmitted. The data are transmitted at a rate of 447.443 kHz per bit with the amplitude ranging from 0 IRE to 70 IRE.



Figure 7.6 Copy Generation Management Systems

VSTAT

0xF90000C0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															CGRDY

BIT	NAME	RW	RESET	DESCRIPTION
Others				RESERVED
				Copy Generation Management System Status (READ ONLY)
1	CGRDY	R	0	0: CGMS is able to set. 1: CGMS is set already.

VCTRL

0xF90000C4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								0	0	0	0	0	CGOE	CGEE	0

BIT	NAME	RW	RESET	DESCRIPTION
Others				Should Be Zero
				Copy Generation Management System enable odd field.
2	CGOE	R/W	0	0: CGMS is not enabled. 1: CGMS is enabled.
				Copy Generation Management System enable even field.
1	CGEE	R/W	0	0: CGMS is not enabled. 1: CGMS is enabled.

CGMSA

0xF90000E0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										CGMS6	CGMS5	CGMS4	CGMS3	CGMS2	CGMS1

CGMSB

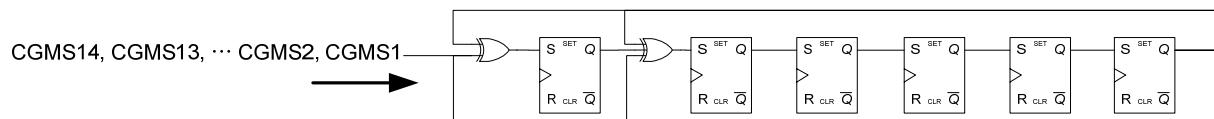
0xF90000E4															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									CGMS14	CGMS13	CGMS12	CGMS11	CGMS10	CGMS9	CGMS8

CGMS8	CGMS7	DESCRIPTION
0	0	Copying permitted without restriction.
0	1	One copy permitted.
1	0	No more copies.(One copy has already been made.)
1	1	No copying permitted.

CGMSC

0xF90000E8															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										CGMS20	CGMS19	CGMS18	CGMS17	CGMS16	CGMS15

CGMSA, CGMSB, and CGMSC registers are 20bits CGMS data. Especially, CGMSC register is for the CRC for CGMSA and CGMSB and is manually set according to CGMSA and CGMSB value. Figure 7.7 shows example of the CRC calculator for CGMS. For example, when CGMSA=0x00 and CGMSB=0x03, CGMSC (CRC) should be set to 0x3A.



The CRC used is $X^6 + X + 1$, all preset to "1".

Figure 7.7 Example: CRC calculator for CGMS

7.11 10-Bit DAC

10-bit DAC is used for composite video signal output.

7.11.1 Features

- Resolution : 10-bit
- Maximum conversion rate : 27MSPS
- BGR (Internal / External)
- Power down mode
- Output load resistance : 37.5Ω
- Analog output range : $0.0 \sim 1.3V$ (Typical)
- Power supply : 3.0V single

7.11.2 Functional Description

The DAC is initially power on state. To make the DAC enter the power down mode, the PD bit of DACPD register in NTSC/PAL encoder should be set to 1

The DAC uses reference current to decide the 1LSB current size by dividing the reference current by 31 times. To adjust the full current output, you must decide the "R(IREF)" resistor value(connected to IREF pin) and "VREF" voltage value(connected to VREF pin). Its voltage output can be obtained by connecting R_{LOAD} (connected to DACOUT pin).

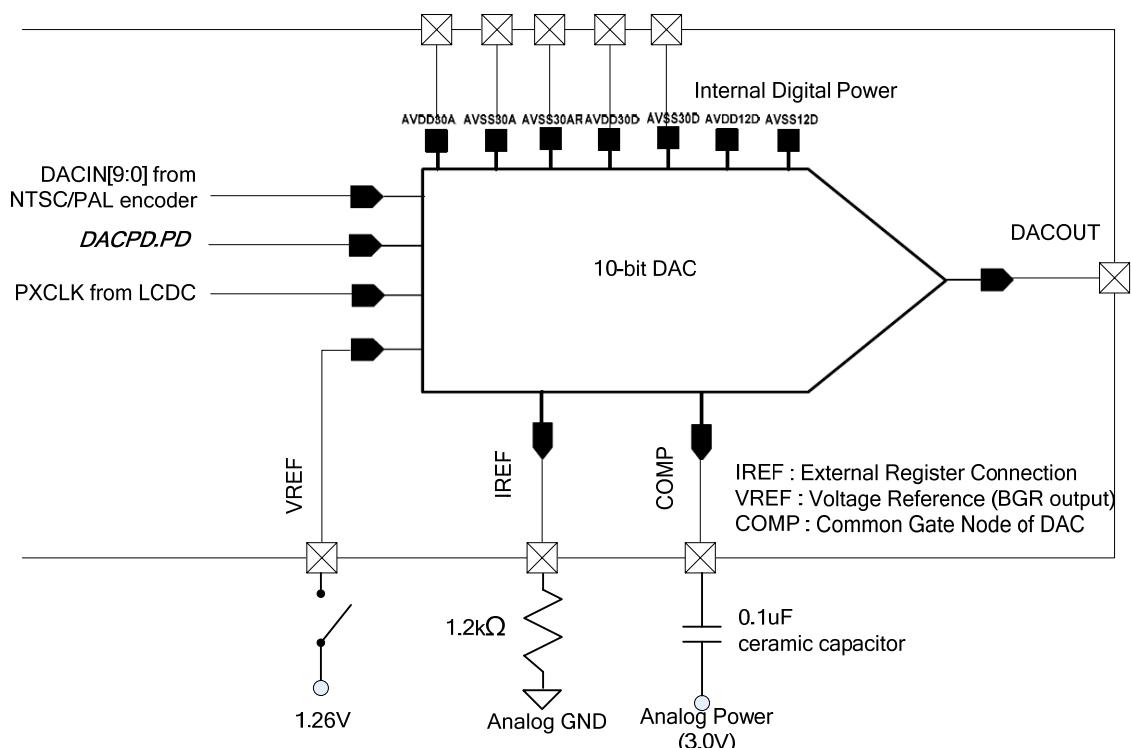


Figure 7.8 10-bit DAC Connection

The voltage output of DAC are decided by $R(IREF)$, R_{LOAD} , and $V(VREF)$.

$$VO = \frac{V(VREF)}{R(IREF) \times 31} \sum_{i=0}^9 (2^i \times DACIN[i]) * R_{LOAD}$$

For example, when the corresponding powers are applied to DAC power pins, VREF might be about 1.26V. If $1.2k\Omega$ is connected to IREF pin and DACIN bits are all 1s and R_{LOAD} is 37.5Ω , DACOUT is about 1.3V. in case $R(IREF)$ is $1.5k\Omega$, DACOUT is about 1.04V. If you cannot change resister value connected to IREF pin, you can apply the corresponding voltage to VREF pin to adjust VO voltage.

8 MEMORY TO MEMORY SCALER

8.1 Overview

The block diagram of the Memory to Memory Scaler (MSC) is shown in the following figure. The hardware reads the source image and resizes it and finally writes the scaled image to the destination region. Specially, the hardware can interface to the LCD controller directly in progressive display output mode.

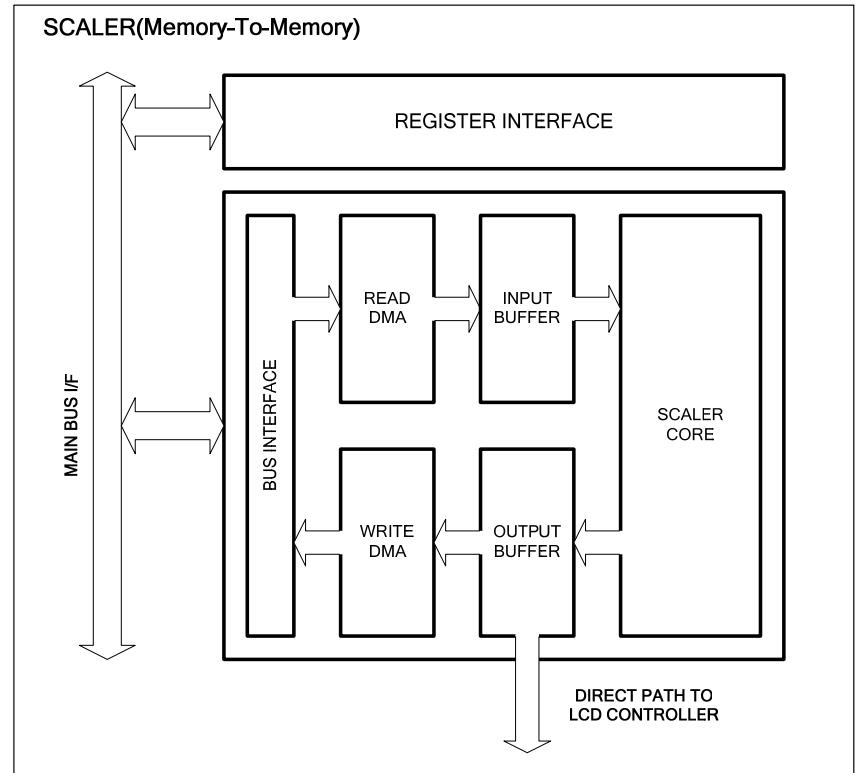


Figure 8.1 Scaler Block Diagram

8.2 Operation

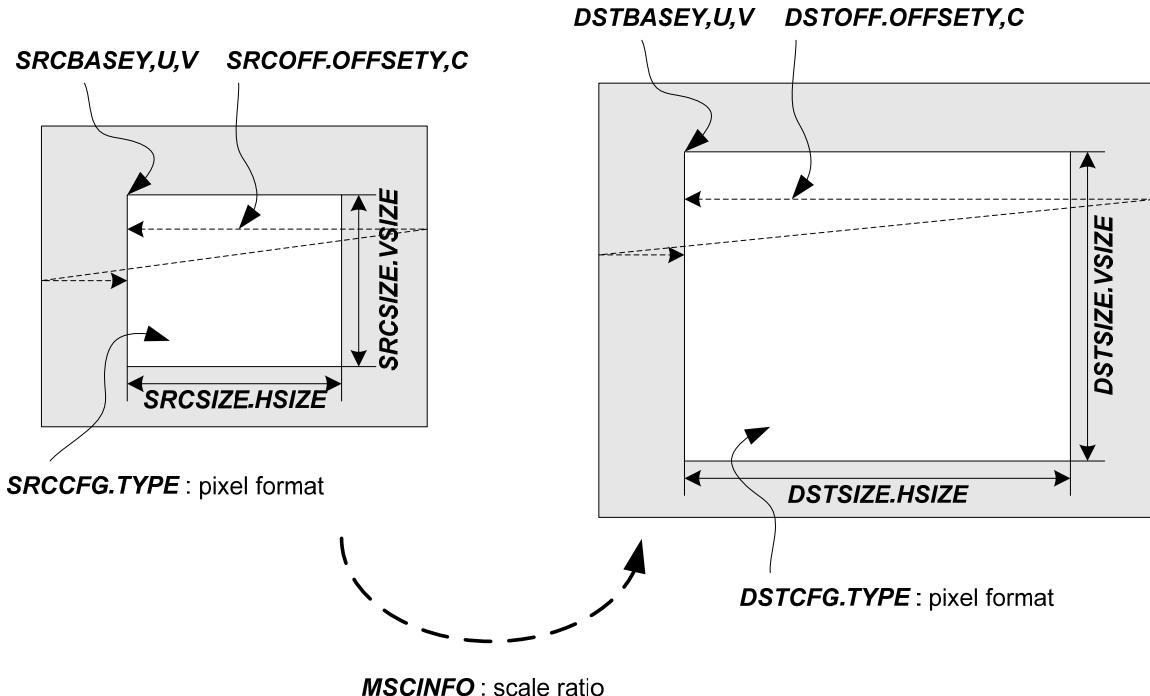


Figure 8.2 Memory to Memory Scaling Operation

The MSC reads the source image in the memory and resizes it and then stores the result image to the specified memory space or send it to the LCDC. Figure 8.2 shows that memory-to-memory scaling operation and the corresponding registers.

The MSC has two modes to store the result image to the memory. One is the frame mode. The other is the rolling mode. The frame mode needs as much memory space as the whole result image size (Figure 8.3(a)). But, the rolling mode needs as much memory space as region which is specified by the rolling lines (Figure 8.3(b)).

In case of rolling mode, there are two methods the MSC informs the on-chip CPU the current status. One is to use the rolling line trigger level; the other is to use the middle line trigger level. Whenever the MSC stores one line to be scaled, the current rolling line counter (CRCNT register) increases by 1. When this counter reaches the rolling line trigger level (ROLLCNT of DSTRMCNT register), it is reset to 0 and addresses which specifies the location to store the result image return to destination base addresses (DSTBASEY, DSTBASEU, and DSTBASEV) and the MSC can issue the interrupt request. The MSC can also inform the on-chip CPU how many lines are stored from the destination base address. When the current rolling line counter reaches the middle line trigger level (MIDCNT of DSTRMCNT register), the MSC can issue the interrupt request. But, in this case, the rolling line counter is not reset to 0 and address does not return to the destination base address.

Additionally, the MSC can make scaling operation stopped temporarily. The middle line trigger level and the rolling line trigger level are used as the indicator. Whenever the current rolling line counter reaches middle line and/or the rolling line trigger level, the MSC pauses in the operation. To resume the operation, RLS bit of MSCCTR register is set to 1. To enable this function, RGSM and MGSM of MSCCTR register should be set to 1. RGSM is for the rolling line trigger level and MGSM is for the middle line trigger level.

To enable the rolling mode, REN bit of MSCCTR register should be set to 1. The rolling

line trigger level is determined by ROLLCNT of DSTRMCNT register. To enable the middle line trigger level, MEN and REN bit of MSCCTR register should be set to 1 and the middle line trigger level is determined by MIDCNT of DSTRMCNT register. And to enable the corresponding interrupt, RIQREN and MIRQEN should be set to 1. They are for the rolling line trigger level interrupt and the middle line trigger level interrupt, respectively.

Finally, when the whole result image is stored to the memory completely, MSCSTR.BUSY bit is cleared to 0 and MSCSTR.RDY bit is set to 1. And these can be used as an interrupt request source.

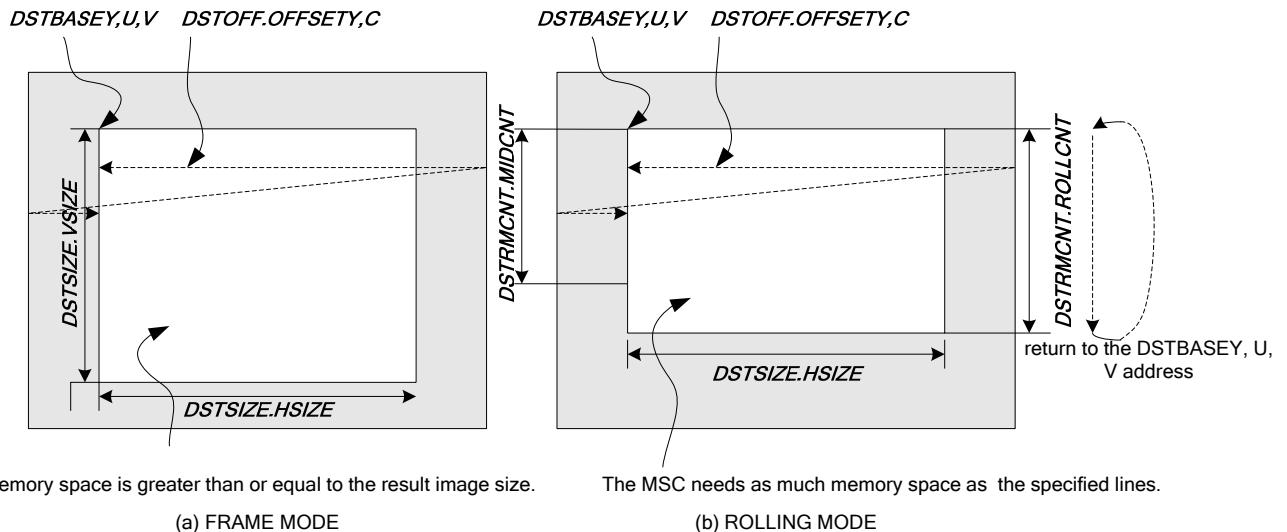


Figure 8.3 Storing the result image

8.3 Registers

Table 8.1 Scaler Registers (Base Address = 0xF0070000)

Name	Addr	Type	Reset	Description
SRCBASEY	0x000	R/W	0x00000000	Scaler source base address for Y
SRCBASEU	0x004	R/W	0x00000000	Scaler source base address for U (Cb)
SRCBASEV	0x008	R/W	0x00000000	Scaler source base address for V (Cr)
SRCSIZE	0x00c	R/W	0x00000000	Source image size register
SRCOFF	0x010	R/W	0x00000000	Source image line offset register
SRCCFG	0x014	R/W	0x00000000	Source image configuration register
DSTBASEY	0x020	R/W	0x00000000	Scaler destination base address for Y
DSTBASEU	0x024	R/W	0x00000000	Scaler destination base address for U (Cb)
DSTBASEV	0x028	R/W	0x00000000	Scaler destination base address for V (Cr)
DST_SIZE	0x02c	R/W	0x00000000	Destination image size register
DSTOFF	0x030	R/W	0x00000000	Destination image line offset register
DSTCFG	0x034	R/W	0x00000000	Destination image configuration register
MSCINF	0x040	R/W	0x00000000	Scaling information register
MSCCTR	0x044	R/W	0x00000000	Scaler control register
MSCSTR	0x048	R/W	0x00000000	Scaler status register
DSTSILINE	0x050	R/W	0x00000000	Destination Set Line
DSTCLINE	0x054	R	0x00000000	Destination Current Line

SRC Image Y Base Address (SRCBASEY)

0xF0070000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCBASEY[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCBASEY[15:2]															

Field	Name	RW	Reset	Description
				Source Base Address
31 ~ 2	SRCBASEY	RW	-	Y for 4:2:0 or 4:2:2 Separate Mode
				Y for 4:2:2 Sequential Mode

SRC Image U Base Address (SRCBASEU)

0xF0070004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCBASEU[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCBASEU[15:2]															

Field	Name	RW	Reset	Description
				Source Base Address for U
31 ~ 2	SRCBASEU	RW	-	SRCBASEU[31:28] and SRCBASEY[31:28] should be the same value.

SRC Image V Base Address (SRCBASEV)

0xF0070008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCBASEV [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCBASEV[15:2]															

Field	Name	RW	Reset	Description
				Source Base Address for V
31 ~ 2	SRCBASEV	RW	-	SRCBASEV[31:28] and SRCBASEY[31:28] should be the same value.

SRC Image Size (SRCSIZE)

0xF007000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Field	Name	RW	Reset	Description
27 - 16	VSIZE	RW	0	Vertical Image Size by pixel unit
11-0	HSIZE	RW	0	Horizontal Image Size by pixel unit

SRC Image Offset (SRCOFF)

0xF0070010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Field	Name	RW	Reset	Description
27 ~ 16	OFFSETC	RW	0x0	Chrominance Address Offset
11 ~ 0	OFFSETY	RW	0x0	Luminance Address Offset

SRC Image Configuration (SRCCFG)

0xF0070014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TYPE [2:0]	Description
7 :RGB454	
6: RGB444	
5: RGB555	
4: RGB565	
n	
3 : YUV420 separate	
2 : YUV422 separate	
1 : YUV422 sequential mode 1	
0 : YUV422 sequential mode 0	

31	0		
Cr0	Y1	Cb0	Y0

YCbCr4:2:2 Sequential Mode 0

Y1	Cr0	Y0	Cb0
----	-----	----	-----

YCbCr4:2:2 Sequential Mode 1

DST Image Y Base Address (DSTBASEY)

0xF0070020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DSTBASEY[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTBASEY[15:2]															

Field	Name	RW	Reset	Description
				Destination Base Address
31 ~ 2	DSTBASEY	RW	-	Y for 4:2:0 or 4:2:2 Separate Mode
				Y for 4:2:2 Sequential Mode

DST Image U Base Address (DSTBASEU)

0xF0070024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DSTBASEU[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTBASEU[15:2]															

Field	Name	RW	Reset	Description
				Destination Base Address for U
31 ~ 2	DSTBASEU	RW	-	DSTBASEU[31:28] and DSTBASEY[31:28] should be the same value.

DST Image V Base Address (DSTBASEV)

0xF0070028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DSTBASEV[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTBASEV[15:2]															

Field	Name	RW	Reset	Description
				Destination Base Address for V
31 ~ 2	DSTBASEV	RW	-	DSTBASEV[31:28] and DSTBASEY[31:28] should be the same value.

DST Image Size (DSTSIZEx)

0xF007002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Field	Name	RW	Reset	Description
27 - 16	VSIZE	RW	0	Vertical Image Size by pixel unit
11-0	HSIZE	RW	0	Horizontal Image Size by pixel unit

DST Image Offset (DSTOTFF)

0xF0070030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Field	Name	RW	Reset	Description
27 ~ 16	OFFSETC	RW	0x0	Chrominance Address Offset
11 ~ 0	OFFSETY	RW	0x0	Luminance Address Offset

DST Image Configuration (DSTCFG)

0xF0070034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Field	Name	RW	Reset	Description
Destination Type Register				
1 ~ 0	TYPE	RW	0x0	7 :RGB454 6: RGB444 5: RGB555 4: RGB565 3 : YUV420 separate 2 : YUV422 separate 1 : YUV422 sequential mode 1 0 : YUV422 sequential mode 0
* Byte Sequence is same as SRCCFG				
Destination Type Register				
4	PATH	RW	0x0	'0' : Memory The scaled image is written to the destination memory. '1' : LCD Channel 0 Direct Path The scaled image is written to the LCD for IMGO.
6	RDY	RW	0x0	Access Wait Control Register Valid for PATH being '1' '0' : Wait for "WAITCNT+1" cycles for LCD Access '1' : Wait until Output FIFO is not empty
10 ~ 8	WAITCNT	RW	0x0	Wait Cycle Count for RDY being '1'
Chrominance Writing Mode Register Defined for 4:2:0 Separate Mode				
11	COP	RW	0x0	'0' : Y0→U0→V0→Y1→Y2→U1 ... '1' : Y0→U0→Y1→V0→Y2→U1 ...

MSC Information (MSCINFO)

0xF0070040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Field	Name	RW	Reset	Description
29 ~ 16	VRATIO	RW	0x0	Vertical scale ratio VRATIO = 256 * SRCSIZE.VSIZE/DSTSIZE.VSIZE
13 ~ 0	HRATIO	RW	0x0	Horizontal scale ratio HRATIO = 256 * SRCSIZE.HSIZE/DSTSIZE.HSIZE

MSC Control (MSCCTR)

0xF0070044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CKG	TST							REN	MEN			RLS	-	RGSM	MGSM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							RST	RIRQE N	MIRQE N	CON	BP		BUSY	RDY	EN

Field	Name	RW	Reset	Description
0	EN	RW	0x0	Enable Register 0b : Operation Disabled 1b : Operation Enabled
1	RDY	RW	0x0	1b : Ready Interrupt Enable
2	BUSY	RW	0x0	1b : Busy Interrupt Enable
				Bypass Enable Signal (Test Purpose)
4	BP	R/W	0x0	0b : Not-bypassed (Normal Operation) 1b : Bypassed * This should be "ZERO"
				Continuous Mode
5	CON	R/W	0x0	0b : single mode In single mode, immediately after EN bit is asserted manually, EN bit is cleared automatically for single frame scaling operation.
				1b : continuous frame mode In continuous mode, EN bit is not cleared automatically. Therefore, until EN bit is cleared manually, scaling operation is continued.
6	MIRQEN	RW	0x0	Enable an interrupt using the middle line counter 0b : disable 1b : enable
7	RIRQEN	RW	0x0	Enable an interrupt using the rolling line counter 0b : disable 1b : enable
				Internal state machine Reset Signal
8	RST	RW	0x0	0b : Not Reset 1b : Reset
				Enable Stop Mode using the middle line counter
16	MGSM	RW	0x0	0b : disable 1b : enable (MEN and REN should be also set to 1)
				Enable Stop Mode using the rolling line counter
17	RGSM	RW	0x0	0b : disable 1b : enable (REN should be also set to 1)
19	RLS	RW	0x0	Release Stop Mode

					When scaling operation is stopped by “rolling line counter” or “middle line counter”, it is resumed by writing 1 to this bit.
					This bit is automatically cleared.
					Enable the middle line counter
22	MEN	RW	0x0	0b : disable 1b : enable	
					Enable the rolling mode
23	REN	RW	0x0	0b : disable (frame mode) 1b : enable (rolling mode)	
30	TST	RW	0x0	Should be zero for test purpose	
					Clock Gating Enable Signal
31	CKG	RW	0x0	0b : Gating Enable 1b : Gating Disabled * This should be “ZERO”	

MSC Status (MSCSTR)

0xF0070048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				-				IR	IM	IBUSY	IRDY			BUSY	RDY

Field	Name	RW	Reset	Description
Ready Status Register				
0	RDY	R	0x0	0b : Not-Ready 1b : Ready
Busy Status Register				
1	BUSY	R	0x0	0b : Not-Busy Status 1b : Busy Status
4	IRDY	RW	0x0	Ready Interrupt Flag By writing '1', it would be cleared.
5	IBUSY	RW	0x0	Busy Interrupt Flag By writing '1', it would be cleared.
6	IR	RW	0	Rolling line interrupt Flag By writing '1', it would be cleared.
7	IM	RW	0	Middle line interrupt Flag By writing '1', it would be cleared.
Ready Status Signals (Test Purpose)				
18 ~ 16	STS	R	0x0	[16] : Output Port Ready Status [17] : Input Port Ready Status [18] : Scaler Interface Ready Status

DST Rolling/Middle Line Count (DSTRMCNT)															0xF0070050				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	ROLLCNT [11:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	MIDCNT[11:0]			
-															-				

Current Rolling Count (CRCNT)															0xF0070054				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	C_RCNT[11:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	-			

C_ROLL_CNT	Description
	Number of lines that are stored the result image from the base address.

9 USB CONTROLLER (HOST1.1, DEVICE2.0)

9.1 Overview

The TCC79XX supports a fully compliant to USB 2.0 specification, high-speed (480 Mbps) functions and suspend/resume signaling. The USB function controller has an endpoint EP0 for control, two in/output endpoints EP1/EP2 for bulk data transaction and a EP3 for interrupt data transaction. The endpoint EP0 has a single 64 byte FIFO; Max packet size is 64 bytes. The endpoint EP1 has a dual 1024 bytes FIFO; Max packet size is 512 bytes. The EP2 has a dual 1024 bytes FIFO; Max packet size is 512 bytes. And the EP3 has a dual 128 bytes FIFO; Max packet size is 64 bytes, respectively.

Interrupt (Status) and Interrupt Enable registers are broken down into 2 banks: Endpoint Interrupts, USB Interrupts. The MAXP, ENDPOINT INTERRUPT and ENDPOINT INTERRUPT ENABLE registers are used regardless of the direction of the endpoint. The associated CSR registers correspond to the direction of endpoint.

The TCC79XX also supports 1 port of USB host interface that has the following features.

- OHCI Rev. 1.0 compliant
- USB Rev. 1.1 compatible
- 1 down stream port

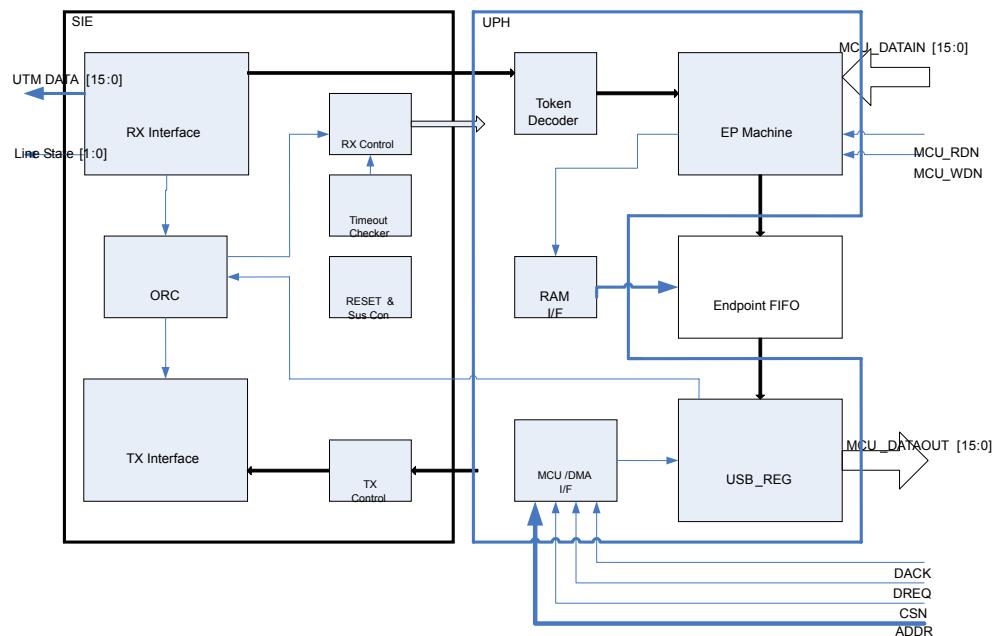


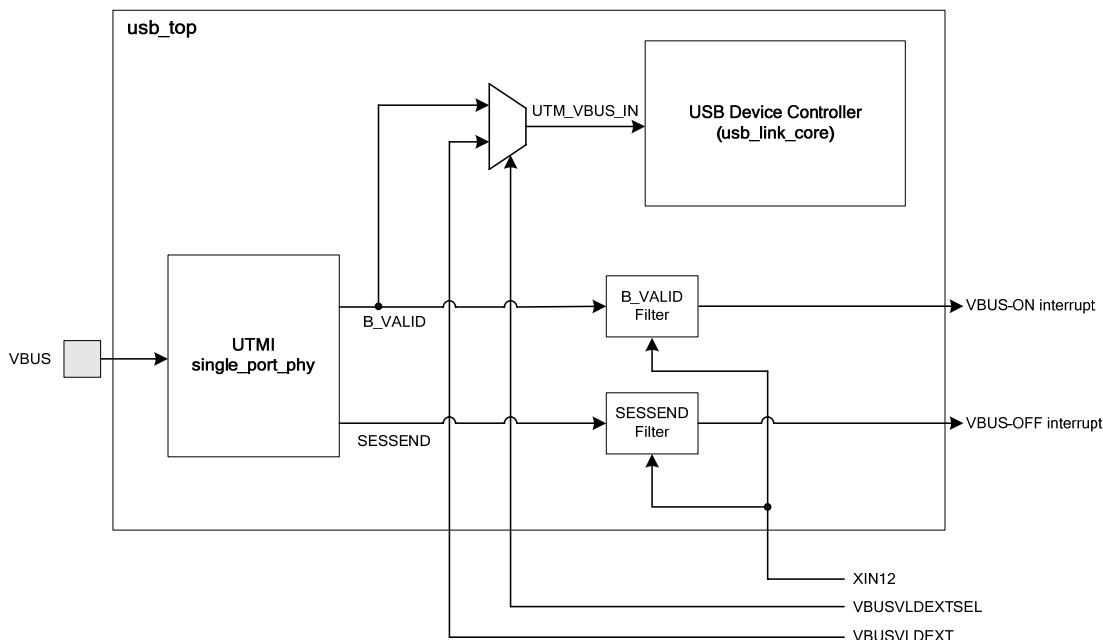
Figure 9.1 USB Controller Block Diagram

9.2 VBUS Valid and End Interrupt

VBUS is the USB power supply pin. An off-chip charge pump must provide power to VBUS pin.

VBUS inputted to UTMI can be set by inputting TCC79XX module. VBUS entering into USB device controller can be selected through multiplexer. The original UTM_VBUS data can be authorized. If it is set to 1, VBUSVLDEXT value can be transmitted to VBUS data of USB device controller. VBUSVLDEXT and VBUSVLDEXTSEL can be set through GPIO in TEST mode or USB Configuration Registration.

There are two interrupts relating to VBUS in USB controller. These are interrupts which indicate ON and OFF of VBUS and they use B_VALID signal and SESSEND signal of UTMI output. B_VALID signal is set to 1 when VBUS is 1 and SESSEND signal is set to 1 when VBUS is 0. Therefore, when VBUS value changes, one of two interrupts definitely occurs. In addition, the filter places before interrupt signal is sent to CKC block in order to remove the glitch of signals. This filter uses XIN12 signal generated in CKC and passes signals when the signals maintain over minimum 400ns value. The important point is OTGDIABLE signal should be 0 in order to use VBUS_OFF interrupt since SESSEND signal does not work when OTGDIABLE signal is 1. The initial value of OTGDIABLE is 1.



9.3 Register Description for USB Device Controller

Table 9.1 USB Register Map (Base Address = 0xF0010000)

Non-Indexed Registers			Indexed Registers		
Name	Abbr.	Addr.	Name	Abbr.	Addr.
Index Register	IR	0x00	Endpoints Status Register	ESR	0x2C
Endpoint Interrupt Register	EIR	0x04	Endpoint Control Register	ECR	0x30
Endpoint Interrupt Enable Register	EIER	0x08	Byte Read Count Register	BRCR	0x34
Reserved			Byte Write Count Register	BWCR	0x38
Reserved			Max Packet Register	MPR	0x3C
Endpoint Direction Register	EDR	0x14			
Test Register	RT	0x18			
System Status Register	SSR	0x1C			
System Control Register	SCR	0x20			
EP0 Status Register	EP0SR	0x24			
EP0 Control Register	EP0CR	0x28			
System Control Register 2	SCR2	0x58			
EP0 Buffer Register	EP0BUF	0x60			
EP1 Buffer Register	EP1BUF	0x64			
EP2 Buffer Register	EP2 BUF	0x68			
EP3 Buffer Register	EP3 BUF	0x6C			
PHYLINK Interface Control Register	PLICR	0xA0			
PHY Control Register	PCR	0xA4			

Index Register**0xF0010000**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														IDX	

IDX[1:0]	Endpoint
n	Endpoint

The index register is used for indexing a specific endpoint. In most cases setting the index register value should precede any other operation.

Endpoint Interrupt Flag Register**0xF0010004**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved														EP3	EP2	EP1	EP0

Operating in MCU mode the endpoint interrupt register lets the MCU knows what endpoint generates the interrupt. Clearing the bits can be accomplished by writing '1' to the bit position where the interrupt is detected.

EP3 [3]	Type	EP3 Interrupt Flag
1	R	Indicates that the USB EP3 interrupt has been generated
	W	Clear the EP3 interrupt flag.

EP2 [2]	Type	EP2 Interrupt Flag
1	R	Indicates that the USB EP2 interrupt has been generated
	W	Clear the EP2 interrupt flag.

EP1 [1]	Type	EP1 Interrupt Flag
1	R	Indicates that the USB EP1 interrupt has been generated
	W	Clear the EP1 interrupt flag.

EP0 [0]	Type	EP0 Interrupt Flag
1	R	Indicates that the USB EP0 interrupt has been generated
	W	Clear the EP0 interrupt flag.

Endpoint Interrupt Enable Register

0xF0010008

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												EP3	EP2	EP1	EP0

EP3 [3]	Type	EP3 Interrupt Control
1	R/W	Enable EP3 interrupt
0		Disable EP3 interrupt

EP2 [2]	Type	EP2 Interrupt Control
1	R/W	Enable EP2 interrupt
0		Disable EP2 interrupt

EP1 [1]	Type	EP1 Interrupt Control
1	R/W	Enable EP1 interrupt
0		Disable EP1 interrupt

EP0 [0]	Type	EP0 Interrupt Control
1	R/W	Enable EP0 interrupt
0		Disable EP0 interrupt

Endpoint Direction Register

0xF0010014

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												EP3D	EP2D	EP1D	EP0D

EP3D[3]	Type	Endpoint 3 Direction Selection
1	R/W	Tx Endpoint
0		Rx Endpoint
EP2D[2]	Type	Endpoint 2 Direction Selection
1	R/W	Tx Endpoint
0		Rx Endpoint
EP1D[1]	Type	Endpoint 1 Direction Selection
1	R/W	Tx Endpoint
0		Rx Endpoint
EP0D[0]	Type	Endpoint 0 Direction Selection
1	R/W	Tx Endpoint
0		Rx Endpoint

System Status Register

0xF001001C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BAERR	TMERR	BSERR	TCERR	DCERR	EOERR	VBOFF	VBON	TBM	DP	DM	HSP	SDE	HFRM	HFSUS P	HFRES

This register reports operational status of the core, especially about error status and power saving mode status. Except the line status, every status bits in the System status Register could be an interrupt sources. When the register is read after an interrupt due to certain system status changes, MCU should write back 1 to the corresponding bits to clear it.

BAERR[15]	Type	Byte Align Error
1	R/W	If error interrupt enable bit of SCR register is set to 1, BAERR is set to 1 when byte alignment error is detected
0		
TMERR[14]	Type	Timeout Error
1	R/W	If error interrupt enable bit of SCR register is set to 1, TMERR is set to 1 when timeout error is detected
0		
BSERR[13]	Type	Bit Stuff Error
1	R/W	If error interrupt enable bit of SCR register is set to 1, BSERR is set to 1 when bit stuff error is detected.
0		
TCERR[12]	Type	Token CRC Error
1	R/W	If error interrupt enable bit of SCR register is set to 1, TCERR is set to 1 when CRC error in token packet is detected.
0		
DCERR[11]	Type	Data CRC Error
1	R/W	If error interrupt enable bit of SCR register is set to 1, DCERR is set to 1 when CRC error in data packet is detected
0		
EOERR[10]	Type	EB OVERRUN Error
1	R/W	If error interrupt enable bit of SCR register is set to 1, EOERR is set to 1 when EB overrun error in transceiver is detected
0		

VBOFF[9]	Type	VBUS OFF
1	R/W	If vbus off interrupt enable bit of SCR register is set to 1, VBUSOFF is set to 1 when VBUS is low
0		
VBON[8]	Type	VBUS ON
1	R/W	If vbus off interrupt enable bit of SCR register is set to 1, VBUSOFF is set to 1 when VBUS is high
0		
TBM[7]	Type	Toggle Bit Mismatch
1	R/W	If error interrupt enable bit of SCR register is set to 1, TBM is set to 1 when Toggle mismatch is detected
0		
DP[6]	Type	DP Data Line State
1	R/W	DP informs the status of D+ line
0		
DM[5]	Type	DM Data Line State
1	R/W	DM informs the status of D- line
0		
HSP[4]	Type	Host Speed
1	R/W	High speed
0		Full speed
SDE[3]	Type	Speed Detection End
1	R/W	SDE is set by the core when the HS Detect Handshake process is ended.
0		
HFRM[2]	Type	Host Forced Resume
1	R/W	HFRM is set by the core in suspend state when Host sends resume signaling.
0		
HFSUSP[1]	Type	Host Forced Suspend
1	R/W	HFSUSP is set by the core when the SUSPEND signaling from host is detected.
0		
HFRES[0]	Type	Host Forced Reset
1	R/W	HFRES is set by the core when the RESET signaling from host is detected.
0		

System Control Register

0xF0010020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTZIEN		DIEN	VBOFE	VBOE	RWDE	EIE	BIS	SPDEN	RRDE	IPS		MFRM	HSSPE	HRESE

FIELD	TYPE	DESCRIPTION
DTZIEN[14]	R/W	DMA Total Counter Zero Int. Enable 1: Enable 0: Disable
DIEN[12]	R/W	DUAL Interrupt Enable 1: Enable 0: Disable
VBOFE[11]	R/W	VBUS OFF Enable 1: Enable 0: Disable
VBOE[10]	R/W	VBUS ON Enable 1: Enable 0: Disable
RWDE[9]	R/W	Reverse Write Data Enable 1: High byte data is first sent to Host 0: Low byte data is first sent to Host
EIE[8]	R/W	Error Interrupt Enable This bit must be set to 1 to enable error interrupt
BIS[7]	R/W	BUS Interface Select The MCU bus width is selected by BIS. When set to 0, bus width is 8bit, when set to 1, bus width is set to 16bit.
SPDEN[6]	R/W	Speed Detect End Interrupt Enable When set to 1, Speed detection interrupt is generated
RRFE[5]	R/W	Reverse Read Data Enable 1: First received byte is loaded in High byte field. 0: First received byte is loaded in Low byte field.
		Interrupt Polarity Select
IPS[4]	R/W	The signal polarity of the interrupt from the core is changed through IPS. When set to 0, the interrupt is considered to be active in low state. When set to 1, the interrupt is considered to be active in high
		Resume by MCU
MFRM[2]	R/W	If this bit is set, the suspended core generates a resume signal. This bit is set when MCU write 1. this bit is cleared when MCU write 0.
		Suspend Enable
HSSPE[1]	R/W	1: Enable 0: Disable
		Reset Enable
HRESE[0]	R/W	1: Enable 0: Disable

EP0 Status Register (NON_INDEXED)

0xF0010024

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								LWO		SHT		TST		RSR	

This register stores status information of the Endpoint 0. This status information is set automatically by the core when corresponding conditions are met. After reading the bits, MCU should write 1 to clear them.

LWO[6]	Type	Last Word Odd
1	R/W	LWO informs that the last word of a packet in FIFO has an invalid upper byte. The bit is cleared automatically after the MCU reads it from the FIFO.
0		
SHT[4]	Type	Stall Handshake Transmitted
1	R/W	SHT informs that STALL handshake due to stall condition is sent to Host. This bit is an interrupt source. This bit is cleared when MCU write 1.
0		
TST[1]	Type	Tx successfully transmitted
1	R/W	TST is set by core after core sends TX data to Host and receives ACK successfully. TST is one of the interrupt sources.
0		
RSR[0]	Type	Rx successfully received
1	R/W	RSR is set by core after core receives error free packet from Host and sent ACK back to Host successfully. RSR is one of the interrupt sources.
0		

EP0 Control Register

0xF0010028

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												ESS	TZLS		

EP0 control register is used for the control of endpoint 0. Controls such as enabling ep0 related interrupts and toggle controls can be handled by EP0 control register.

EES[1]	Type	Endpoint Stall Set
1	R/W	ESS is set by MCU when it intends to send STALL hand shake to Host.
0		This bit is cleared when the MCU writes 0 on it ESS is needed to be set 0 after MCU writes 1 on it.
TZLS[0]	Type	Tx Zero Length Set
1	R/W	TZLS is set by MCU when it intends to send Tx zero length data to Host. This bit is cleared when the MCU write 0 on it.
0		

Endpoint0 Buffer Register

0xF0010060

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP0 BUF															

Endpoint1 Buffer Register

0xF0010064

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP1 BUF															

Endpoint2 Buffer Register

0xF0010068

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP2 BUF															

Endpoint3 Buffer Register

0xF001006C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP3 BUF															

The buffer register is used to hold data for TX/RX transfer.

PHYLINK Interface Control Register (PLICR)

0xF00100A0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLC						LPC									

The interface timing between PHY and Link can be controlled using this register.

Bit	Name	Initial	R/W	Description
31-12	-	0	R	Reserved
11-8	PLC	0010b	R/W	Link to PHY interface Control. The region of PLC is from 4'b0000 to 4'b1111.
7-4	LPC	0100b	R/W	Link to PHY interface Control. The region of LPC is from 4'b0000 to 4'b1111.
3-0	-	0	R	Reserved

PHY Control Register (PCR)

0xF00100A4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								URSTC	SIDC	OPMC	TMSC	XCRC	SUSPC	PCE	

PHY Control Inputs can be controlled using this register.

Bit	Name	Initial	R/W	Description
31-12	-	0	R	Reserved
7	URSTC	0	R/W	UTMI_RESET control (PHY Software Reset) 0: UTMI_RESET is 0 (Reset enable). 1: UTMI_RESET is 1
6	SIDC	0	R/W	SIDDQ control 0: SIDDQ is 0. 1: SIDDQ is 1. SIDC is connected to SIDDQ of USB PHY only when PCE bit is 1.
5-4	OPMC	0	R/W	OPMODE control 00: Normal 01: Non-Driving 10: Disable bit stuffing and NRZI encoding 11: Reserved OPMC is connected to OPMODE of USB PHY only when UPCR0.DPPD is 0. Refer to UPCR0 register on page 9-20.
3	TMSC	0	R/W	TERMSEL control 0: TERMSEL is 0. 1: TERMSEL is 1. TMSC is connected to TERMSEL of USB PHY only when UPCR0.DPPD is 0. Refer to UPCR0 register on page 9-20.
2	XCRC	0	R/W	XCVRSEL control 0: XCVRSEL is 0. (HS Transceiver) 1: XCVRSEL is 1. (FS Transceiver) XCRC is connected to XCVRSEL of USB PHY only when UPCR0.DPPD is 0. Refer to UPCR0 register on page 9-20.
1	SUSPC	0	R/W	SUSPENDMD control 0: SUSPENDMD is 0. 1: SUSPENDMD is 1.
0	PCE	0	R/W	PHY Control Enable 0: Control Disable. 1: Control Enable.

Endpoint Status Register (Indexed)

0xF001002C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FUDR	FOVF			FPID	OSD	DTCZ	SPT	DOM	FFS	FSC	LWO		PSIF	TPS	RPS

The endpoint status register reports current status of an endpoint (except EP0) to the MCU.

FUDR[15]		Type	FIFO Underflow
1		R/W	FUDR is only used for ISO mode. FUDR is set when FIFO is empty and Host sends IN token.
0			This bit is cleared when the MCU writes 1.
FOVF[14]		Type	FIFO Overflow
1		R/W	FOVF is only used for ISO mode FOFV is set when FIFO is full and Host sends OUT data. This bit is cleared when the MCU write 1.
0			
FPID[11]		Type	First OUT Packet Interrupt Disable
1		R/W	First out packet interrupt disable in out DMA operation. First received OUT packet generates interrupt if this bit is disabled and DEN in DMA control register is enabled
0			
OSD[10]		Type	OUT Start DMA
1		R/W	OSD is set when First OUT packet is received after Registers related DMA operation are set.
0			
DTCZ[9]		Type	DMA Total Count Zero
1		R/W	DTCZ is set when DMA total counter reach to 0
0			This bit is cleared when the MCU writes 1 on it
SPT[8]		Type	DMA Total Count Zero
1		R/W	DTCZ is set when DMA total counter reach to 0
0			This bit is cleared when the MCU writes 1 on it
DOM[7]		Type	Dual Operation Mode
1		R	DOM is set when the max packet size of corresponding endpoint is equal to a half FIFO size.
0			Endpoint 0 does not support dual mode
FFS[6]		Type	FIFOFlushed
1		R/W	FFS informs that FIFO is flushed. This bit is an interrupt source. This bit is cleared when the MCU clears FLUSH bit in Endpoint Control Register.
0			
FSC[5]		Type	Functional Stall Condition
1		R/W	FSC informs that STALL handshake due to functional stall condition is sent to Host. This bit is set when endpoint stall set bit is set by the MCU. This bit is cleared when the MCU writes 1 on it
0			

LWO[4]	Type	Last Word Odd
1	R	LWO informs that the lower byte of last word is only valid. This bit is automatically cleared after the MCU reads packet data received Host.
0		
PSIF[3:2]	Type	Packet Status In FIFO
		00: No packet in FIFO. 01: One packet in FIFO. 10: Two packets in FIFO. 11: Invalid value.
TPS[1]	Type	Tx Packet Success
1	R/W	TPS is used for Single or Dual transfer mode TPS is activated when one packet data in FIFO was successfully transferred to Host and received ACK from Host. This bit should be cleared by writing 1 on it after being read by MCU.
0		
RPS[0]	Type	Rx Packet Success
1	R	RPS is used for Single or Dual transfer mode RPS is activated when FIFO has a packet data to receive. RPS is automatically cleared when MCU reads all packet from FIFO. MCU can identify the packet size through BYTE READ COUNT REGISTER
0		

Endpoint Control Register (INDEXED)

0xF0010030

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserve		INHLD	OUTHD		TNPMF	IME	DUEN	FLUSH	TTE		TTS		CDP	ESS	TZLS

The endpoint control register is useful for controlling an endpoint both in normal operation and test case. Putting an endpoint in specific operation mode can be accomplished through the endpoint control register.

INHLD[12]	Type	IN Packet HOLD
1	R/W	The USB sends NAK handshake to Host regardless of IN fifo status.
0		The USB can send IN data to Host according to IN fifo status(normal operation)
OUTHD[11]	Type	IN Packet HOLD
1	R/W	The USB does not accept OUT data from Host
0		The USB can accept OUT data from Host according to OUT fifo status(normal operation)
TNPMF[10:9]	Type	Transaction Number / Micro Frame
		TNPMF is useful for ISO transfer.
	R/W	00: Invalid value. 01: 1 transaction per micro frame 10: 2 transaction per micro frame 11: 3 transaction per micro frame
IME[8]	Type	ISO Mode Endpoint
1	R/W	ISO mode
0		Bulk(interrupt) mode
DUEN[7]	Type	Dual FIFO mode Enable
1	R/W	Dual Enable
0		Dual disable(Single mode)
FLUSH[6]	Type	FIFO Flush
1	R/W	FIFO is flushed when this bit is set to 1. The bit is automatically cleared after MCU write 1.
0		
TTE[5]	Type	TX Toggle Enable
1	R/W	The MCU can force TX data toggle bit with TTE. The bit is useful for test. The TX data toggle bit changes automatically in normal operation.
0		0 : disable / 1: enable

TTS[4:3]	Type	TX Toggle Select
		TTS is used for test. This is valid when TX Toggle Enable(TTE) is set 00 : DATA PID 0. R/W 01 : DATA PID 1. 10 : DATA PID 2.(Only in ISO mode) 11 : DATA PID 3.(Only in ISO mode)
CDP[2]	Type	Clear Data PID
		In RX Mode When this bit is set to 1, data toggle bit in core to be compared with the data PID of received packet is reset to 0. this bit is automatically cleared R/W after MCU writes 1. In TX Mode TX data PID to be transmitted to host is reset to 0 when this bit is set to 1. this bit is automatically cleared after MCU write 1.
ESS[1]	Type	Endpoint Stall Set
1	R/W	ESS is set by the MCU when the MCU intend to send STALL hand- shake to Host.
0		
TZLS[5]	Type	TX Zero Length Set
1	R/W	This bit is used for Test. TZLS is set by the MCU when the MCU intend to send zero length TX data to Host this bit is cleared when the MCU write 0 in it.
0		

Byte Read Count Register (INDEXED)

0xF0010034

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						RDCNT									

FIFO READ Count[9:0] RDCNT is read only The BRCD informs the amount of received data from host.

RDCNT informs the amount of data in word (16bit) unit. Through the LWO bit of EP0SP, the MCU can determine valid byte in last data word.

In 8 bit Interface, RDCNT keeps the byte size of received data.

Byte Write Count Register (INDEXED)

0xF0010038

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						WRTCNT									

The byte write count register keeps the byte count value of a TX packet from MCU. The counter value will be used to determine the end of TX packet.

Through BWCR, the MCU must load the byte counts of a TX data packet to the core. The core uses this count value to determine the end of packet. The count value to this register must be less than MAXP.

MAX Packet Register (INDEXED)

0xF001003C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						MAXP									

The max packet size of each endpoint is determined by programming the MAX packet Register.

The max packet size of each endpoint is determined by MAX packet register. The range of max packet is from 0 to 512 bytes

Note. This USB2.0 device has 4 FIFO memories. In addition, the size of EP0, EP1, EP2 and EP3 are 64bytes, 1024bytes, 1024bytes, and 128bytes respectively.

9.4 Register Description for USB Host Controller

The USB host controller complies with OHCI Rev 1.0. Refer to the specification of OHCI (Open Host Controller Interface) Rev 1.0 for more detailed information.

The following figure illustrates a block diagram of USB host controller.

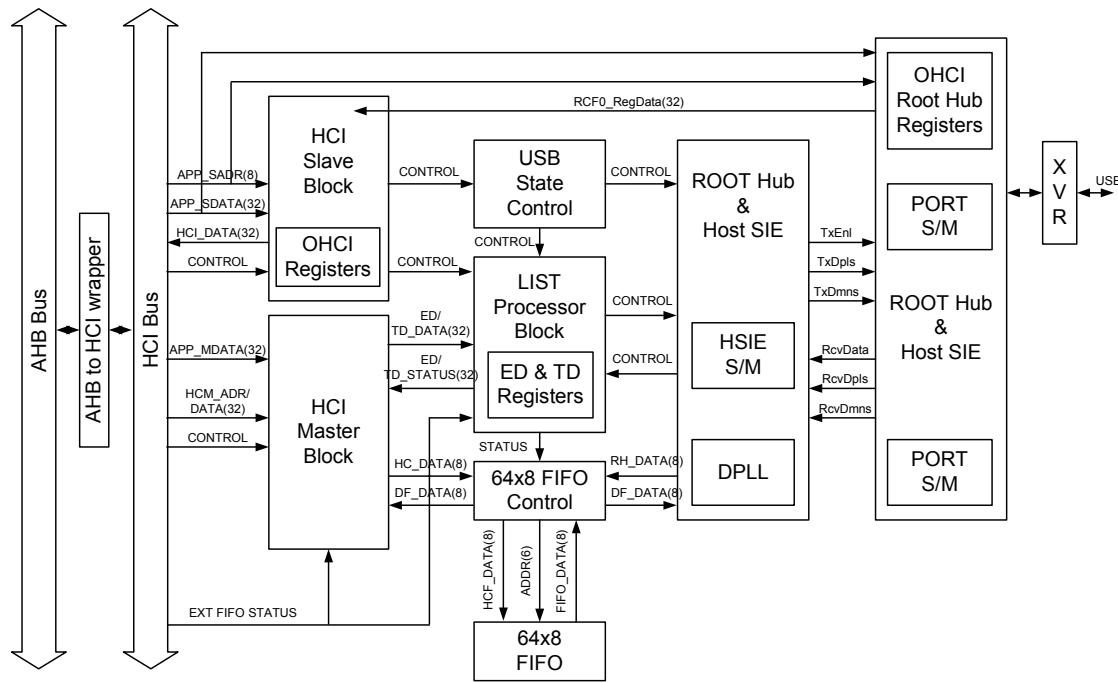


Figure 9.2 USB Host Controller Block Diagram

The following table in next page describes the address mapping of OHCI Rev1.0 registers.

Table 9.2 USB Host Register Map (Base Address = 0xF0020000)

Name	Address	Type	Reset	Description
HcRevision	0x00	R	0x00000010	
HcControl	0x04	R/W	0x00000000	
HcCommandStatus	0x08	R	0x00000000	
HcInterruptStatus	0x0C	R	0x00000000	Control and status registers
HcInterruptEnable	0x10	R/W	0x00000000	
HcInterruptDisable	0x14	W	0x00000000	
HcHCCA	0x18	R/W	0x00000000	
HcPeriodCurrentED	0x1C	R	0x00000000	
HcControlHeadED	0x20	R/W	0x00000000	
HcControlCurrentED	0x24	R/W	0x00000000	Memory pointer registers
HcBulkHeadED	0x28	R/W	0x00000000	
HcBulkCurrentED	0x2C	R/W	0x00000000	
HcDoneHead	0x30	R	0x00000000	
HcRmInterval	0x34	R/W	0x00002EDF	
HcFmRemaining	0x38	R/W	0x00000000	
HcFmNumber	0x3C	R/W	0x00000000	Frame counter registers
HcPeriodStart	0x40	R/W	0x00000000	
HcLSThreshold	0x44	R/W	0x00000628	
HcRhDescriptorA	0x48	R/W	0x02001202	
HcRhDescriptorB	0x4C	R/W	0x00000000	
HcRhStatus	0x50	R/W	0x00000000	Root hub registers
HcRhPortStatus1	0x54	R/W	0x00000100	
HcRhPortStatus2	0x58	R/W	0x00000100	

9.5 Register Description for UTMI (USB PHY)

USB PHY Configuration Register0 (UPCR0) 0xF00100C8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR	CM	RCS	RCD	SDI	FO	VBDS	DMPD	DPPD	TBSH	TBS	VBD	LBE			

BIT	Name	RW	Reset	Description
-----	------	----	-------	-------------

Per-Port Reset

When asserted, this customer-specific signal resets the corresponding port's transmit and receive logic without disabling the clocks within the USB 2.0 nanoPHY.

0: The transmit and receive FSMs are operational, and the line_state logic becomes sequential after 11 PHYCLOCK cycles.

1: The transmit and receive finite state machines are reset, and the line_state logic combinationally reflects the state of the single-ended receivers.

14 PR R/W 0

Asserting PORTRESET does not override any USB 2.0 nanoPHY inputs that normally control the USB state, nor does it cause any transient, illegal USB states.

Within 100 ns of asserting PORTRESET, the controller must set the inputs that control the USB to values that cause a safe state. A safe state for Host and Device modes is defined as follows:

- Host mode: Non-driving (OPMODE = 2'b01) with the 15-kilohm pull-down resistors enabled (DPPULLDOWN and DMPULLDOWN = 1'b1)
 - Device mode: Non-driving with the 1.5-kilohm pull-up resistor disabled (OPMODE = 2'b01)
- For more information

COMMONONNN

This signal controls the power-down signals in the XO, Bias, and PLL blocks when the USB 2.0 nanoPHY is suspended.

13 CM R/W 1

0: The XO, Bias, and PLL Blocks remains powered down in suspend mode.

1: The XO, Bias, and PLL Blocks are powered down in suspend mode.

REFCLKSEL

This signal selects the reference clock source for the PLL block.

12-11	RCS	R/W	1	RCS	Reference Clock Select for PLL Block
				11	The PLL uses CLKCORE as reference
				10	The PLL uses CLKCORE as reference
				01	The XO block uses an external clock supplied on the XO

00 The XO block uses the clock from a crystal

REFCLKDIV

This signal selects the USB PHY reference clock

				RCD	Reference Clock Frequency Select
10-9	RCD	R/W	0	11	Reserved
				10	48 MHz
				01	24 MHz
				00	12 MHz

SIDDQ

This test signal enables you to perform IDDQ testing by powering down all analog blocks.

				SID	IDDQ Test Enable
8	SDI	R/W	1	1	The analog blocks are powered down
				0	The analog blocks are not powered down.

FSXCVROWNER

This controller signal enables the UTMI or serial interface.

				FO	UTMI/Serial Interface Select
7	FO	R/W	0	1	The TXENABLEN, FSDATAEXT, and FSSE0EXT inputs drive USB 2.0 nanoPHY data output onto the D+ and D- lines. Data that the USB 2.0 nanoPHY receives from the D+ and D- lines appears on the FSVMINUS and FSVPLUS outputs.
				0	Data on the D+ and D- lines is transmitted and received through the UTMI.

VBUSVLDEXTSEL

This controller signal enables the UTMI or serial interface. This signal selects the VBUSVLDEXT input or the internal Session Valid comparator to indicate when the VBUS signal on the USB cable is valid

6 VBDS R/W 0

				VBDS	External VBUS Valid Select
				1	The VBUSVLDEXT input is used
				0	The internal Session Valid comparator is used

DMPULLDOWN

This controller signal enables or disables the pull-down resistance on the D-line.

5 DMPD R/W 1

When an A/B device is acting as a host (Downstream-facing port), DPPULLDOWN and DMPULLDOWN are enabled. UTMI+/OTG-compliant controllers are not allowed to toggle DPPULLDOWN and DMPULLDOWN during normal operation. UPCR2.OPMODE =

2'b01(non-driving) has precedence over the DPPULLDOWN and DMPULLDOWN controls.

DMPD		D- Pull-Down Resistor Enable
-------------	--	-------------------------------------

- | | |
|---|---|
| 1 | The pull-down resistance on D- is enabled. |
| 0 | The pull-down resistance on D- is disabled. |

DPPULLDOWN

This controller signal enables or disables the pull-down resistance on the D+line.

4 DPPD R/W 1 When an A/B device is acting as a host (downstream-facing port), DPPULLDOWN and DMPULLDOWN are enabled. UTMI+/OTG-compliant controllers are not allowed to toggle DPPULLDOWN and DMPULLDOWN during normal operation.

DPPD		D+ Pull-Down Resistor Enable
-------------	--	-------------------------------------

- | | |
|---|--|
| 1 | The pull-down resistance on D+ is enabled |
| 0 | The pull-down resistance on D+ is disabled |

TXBITSTUFFENH

This controller signal enables or disables bits stuffing on DATAIN[7:0] when OPMODE[1:0] =2'b11

3 TBSH R/W 0

TBSH		High-Byte Transmit Bit-Stuffing Enable
-------------	--	---

- | | |
|---|---------------------------|
| 1 | Bit stuffing is enabled |
| 0 | Bit stuffing is disabled. |

TXBITSTUFFEN

This controller signal enables or disables bits stuffing on DATAIN[7:0] when OPMODE[1:0] =2'b11

2 TBS R/W 0

TBS		Low-Byte Transmit Bit-Stuffing Enable
------------	--	--

- | | |
|---|--------------------------|
| 1 | Bit stuffing is enabled |
| 0 | Bit stuffing is disabled |

VBUSVLD

External VBUS Valid Indicator

This signal is valid in device mode only when the VBUSVLDEXTSEL signal is high. VBUSVLDEXTSEL indicates whether the VBUS signal on the USB cable is valid. In addition, VBUSVLDEXTSEL enables the pull-up resistor on the D+ line.

1 VBD R/W 0

In Host mode, this input is not used and can be tied to 1'b0.

VBD		External VBUS Valid Indicator
------------	--	--------------------------------------

- | | |
|---|---|
| 1 | The VBUS signal is valid, and the pull-up resistor on D+ is enabled |
| 0 | The VBUS signal is not valid, and the pull-up resistor on D+ is disabled. |

0 LBE R/W 0 LOOPBACKENB

This controller signal places the USB 2.0 nanoPHY in Loopback mode, which enables the receive and transmit logic concurrently. Loopback mode is for test purposes only and cannot be used for normal operation.

LBE	Loopback Test Enable
1	During data transmission, the receive logic is enabled.
0	During data transmission, the receive logic is disabled.

USB PHY Configuration Register1 (UPCR1)

0xF00100CC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXFSLST				SQRXT				OTGT				CDT			

TXFSLST	[15:12]	FS/LS Pull-Up Resistance Adjustment (TXFSLSTUNE)
1111	-2.5%	
0111	Design default (default)	
0011	+2.5%	
0001	+5%	
0000	+7.5%	
Others	Reserved	

This bus adjusts the low- and full-speed pull-up resistance, based on nominal power, voltage, and temperature.

SQRXT	[10:8]	Squelch Threshold Tune (SQRXTUNE)
111	-20%	
110	-15%	
101	-10%	
100	-5%	
011	Design default (default)	
010	+5%	
001	+10%	
000	+15%	
Others	Reserved	

This bus adjusts the voltage level for the threshold used to detect valid high-speed data.

OTGT	[6:4]	VBUS Valid Threshold Adjustment (OTGTUNE)
111	+9%	
110	+6%	
101	+3%	
100	Design default (default)	
011	-3%	
010	-6%	
001	-7%	
000	-12%	
Others	Reserved	

This bus adjusts the voltage level for the VBUS valid threshold.

CDT	[2:0]	Disconnect Threshold Adjustment (COMPDISTUNE)
111	+6%	
110	+4.5%	
101	+3%	
100	+1.5%	
011	Design default (default)	
010	-1.5%	
001	-3%	
000	-4.5%	

This bus adjusts the voltage level for the threshold used to detect a disconnect event at the host.

USB PHY Configuration Register2 (UPCR2)

0xF00100D0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TM	[14]	USB Termination Select (TERMSEL)
1	Full Speed termination is enabled (default).	
0	High Speed termination is enabled.	

This controller signal selects FS or HS termination. If UPCR0.DPPD is set to 0, UTMI is controlled by PCR.TMSC on page 9-12.

Four PHYCLOCK cycles are required for internal synchronous reset generation, and an additional six cycles are required to enable HS terminations in the digital core. Therefore, the controller must not transmit a high-speed packet within 10 PHYCLOCK cycles after switching TERMSEL to HS termination.

XCVRSEL	[13:12]	Tranceiver Select
11	Sends an LS packet on an FS bus or receives an Ls packet	
10	LS transceiver	
01	FS transceiver (default)	
00	HS transceiver	

This controller bus selects the HS, FS, or LS transceiver. If UPCR0.DPPD is set to 0, UTMI is controlled by PCR.XCRC on page 9-12.

The voltage regulator in the HS Transmitter requires 1.5 μ s to power up. Therefore, the controller must not transmit a high speed packet within 1.6 μ s after switching XCVRSEL to HS Transceiver (from any other setting).

OPMODE	[10:9]	UTMI Operational Mode
11	Normal operation without SYNC or EOP generation. If the XCVRSEL bus is not set to 2'b00 when OPMODE[1:0] is set to 2'b11, USB 2.0 nanoPHY behavior is undefined.	
10	Disable bit stuffing and NRZI encoding	
01	Non driving	
00	Normal(default)	

This controller bus selects the UTMI operational mode. If UPCR0.DPPD is set to 0, UTMI is controlled by PCR.OPMC on page 9-12.

TXVRT	[8:5]	HS DC Voltage Level Adjustment (TXVREFTUNE)
1111	+8.75%	
1110	+7.5%	
1101	+6.25%	
1100	+5%	
1011	+3.75%	
1010	+2.5%	
1001	+1.25%	
1000	Design default (default)	
0111	-1.25%	

0110	-2.5%
0101	-3.75%
0100	-5%
0011	-6.25%
0010	-7.5%
0001	-8.75%
0000	-10%

This bus adjusts the voltage to which the high speed DC level is tuned.

TXRT	[3:2] HS Transmitter Rise/Fall Time Adjustment(TXRISETUNE)
01	-8%
00	Design default (default)
Others	Reserved

This bus adjusts the rise/fall times the high speed waveform.

TP	[0] HS transmitter Pre-Emphasis Enable (TXPREEMPHASISTUNE)
1	The HS Transmitter pre-emphasis is enabled
0	The HS transmitter pre-emphasis is disabled. (default)

This signal enables or disables the pre-emphasis for a J-K or K-J state transition in HS mode.

USB PHY Configuration Register3 (UPCR3)

0xF00100D4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SRST	[15]	USB PHY Software Reset
	1	USB PHY is on normal state
SRST	0	USB PHY is on reset state
OTG	[12]	OTG Disable
	1	The OTG block is powered down.
OTG	0	The OTG block is not powered down

This controller signal powers down the OTG block, including the VBUS Valid comparator. If the application does not use OTG functionality, you can set this input high to save power.

DRV	[10]	DRV VBUS
	1	The VBUS Valid comparator is enabled.
DRV	0	The VBUS Valid comparator is disabled.

This controller signal enables or disables the VBUS Valid comparator. This signal drives 5 V on VBUS through an external charge pump. When OTGDISABLE is set to 1'b0 and DRV VBUS is asserted, the band gap circuitry and VBUS Valid comparator are powered, even in Suspend mode.

CHR	[9]	VBUS Input Charge Enable
	1	Enables charging VBUS through the SRP pull-up resistance (CHRGVBUS must be asserted for a minimum of 30 ms)
CHR	0	Disables charging VBUS through the SRP pull-up resistance (default)

This controller signal enables or disables charging the VBUS input through the SRP pull-up resistance.

DCH	[8]	VBUS Input Discharge Enable
1		Enables discharging VBUS through the SRP pull-down resistance (DISCHRGVBUS must be asserted for a minimum of 50 ms).
0		Disables discharging VBUS through the SRP pull-down resistance. (default)

This controller signal enables or disables discharging the VBUS input through the SRP pull-down resistance.

TXHSXVT	[3:2]	Transmitter High Speed Crossover Adjustment
3		The crossover voltage is increased by 15mV
2		The crossover voltage is increased by 30mV
1		Default setting
0		Reserved

TXHSXVT adjusts the voltage at which the DP and DM signals cross while transmitting in HS mode

ABE	[0]	TX Jitter Adjustment
1		TX Jitter Fix enable (default)
0		TX Jitter Fix disable

10 IDE**10.1 Overview**

IDE controller supports PIO mode0,1,2,3,4 and UDMA mode0,1,2,3,4,5. IDE controller interfaces with ATA/ATAP compliant devices. In UDMA (Ultra DMA) mode, CPU can receive or transmit data using internal trigger level interrupt but for the better performance, use Internal DMA (IDMA). Internal DMA is one of AHB master. When FIFO Pointers reaches trigger level, and IDE (Internal DMA Enable) bit is set, Internal DMA receive data from source address or transmit the data to target address. When IDE bit is set, CPU can't access data of internal FIFO. When operation is ended, check UDMAINT register. UDMAINT register includes important information (e.g., word aligned, FIFO error, valid data remained in FIFO etc).

The simple block diagram of IDE Controller is as followings.

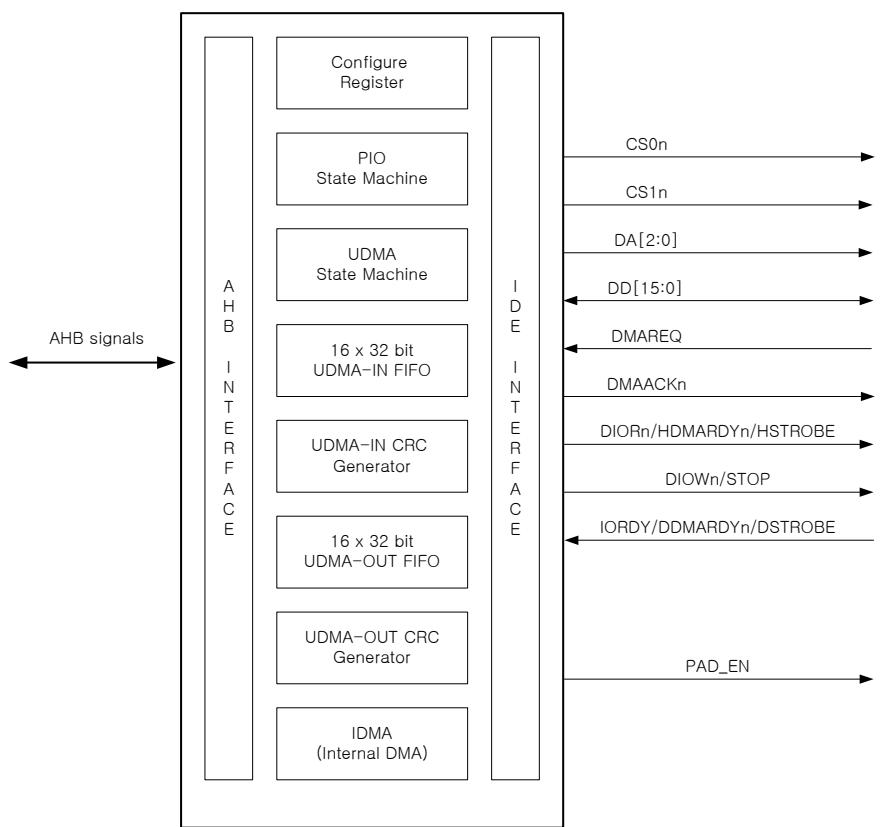


Figure 10.1 IDE Controller Block Diagram

The Maximum theoretical bandwidth of the IDE Interface is shown in Table 10.1.

Table 10.1 Speed of Data Transfer (Byte per Second)

MODE	SPEED	MODE	SPEED
PIO MODE0	3.3 MBps	UDMA MODE 0	16.67 MBps
PIO MODE 1	5.22 MBps	UDMA MODE 1	25 MBps
PIO MODE 2	8.33MBps	UDMA MODE 2	33.33 MBps
PIO MODE 3	11.11MBps	UDMA MODE 3	44.44 MBps
PIO MODE 4	16.67MBps	UDMA MODE 4	66.67 MBps
		UDMA MODE 5	100 MBps

10.2 Register Description

The overall registers are shown in Table 10.2.

Table 10.2 IDE Registers (Base = 0xF0030000)

Name	Address	Type	Reset	Description
CS0n	0x00~ 0x1F	R/W	-	PIO CS0n Access Register
CS1n	0x20 ~0x3F	R/W	-	PIO CS1n Access Register
PIOCTRL	0x40	R/W	0x00600000	PIO Mode Control Register
UDMACTRL	0x44	R/W	0x00000000	UDMA Mode Control Register
IDMACTRL	0x48	R/W	0x00000000	IDMA Control Register
IDMASA	0x4C	R/W	0x00000000	IDMA Source Address Register
IDMASP	0x50	R/W	0x00000000	IDMA Source Parameter Register
IDMACSA	0x54	R	0x00000000	IDMA Current Source Address Register
IDMADA	0x58	R/W	0x00000000	IDMA Destination Address Register
IDMADP	0x5C	R/W	0x00000000	IDMA Destination Parameter Register
IDMACDA	0x60	R	0x00000000	IDMA Current Destination Address Register
IDEINT	0x64	R/W	0x0000_0000	IDE Interrupt Register
UDMATCNT	0x68	R/W	0x00FF_FFFF	UDMA Transfer Counter Register
UDMAIN	0x6C	R	-	UDMA-IN Access Register
UDMAOUT	0x70	W	-	UDMA-OUT Access register
UDMACRC	0x74	R	0x0000_4ABA	UDMA CRC Register
UDMACTCNT	0x78	R	0x00FF_FFFF	UDMA Current Transfer Counter Register

PIO CS0n Access Register (CS0n)

0xF0030000~0xF003001F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DD0[15:0]															

Bit	Name	R/W	Reset	Description
31:16			0	Reserved
15:0	DD0	R/W	0	16-bit access data for CS0n

DD0 is access data to read or to be written for CS0n in PIO MODE.

For example, to write 0x1234(data value) to PIO address 0x6 using CS0n, CPU or DMA target address(HADDR[6:0]) must be 0x18 and DD0 must be 0x1234, which means HADDR[4:2] equals DA[2:0] and DD0 equals DD[15:0]. To read data of PIO address 0x7, CPU or DMA target address (HADDR[6:0]) must be 0x1C, which means HADDR[4:2] equals DA[2:0] and DD0 is read data of PIO address 0x7.

PIO CS1n Access Register (CS1n)

0xF0030020 ~ 0xF003003F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DD1[15:0]															

DD1[15:0]

PIO Data for CS0n

DD1 16bit access data for CS1n

DD1 is access data to read or to be written for CS1n in PIO MODE.

To write 0x1234 (data value) to PIO address 0x6 using CS1n, CPU or DMA target address(HADDR[6:0]) must be (0x18 + 0x20) and DD1 must be 0x1234, which means HADDR[4:2] equals DA[2:0] and DD1 equals DD[15:0]. To read data of PIO address 0x7, CPU or DMA target address (HADDR[6:0]) must be (0x1C + 0x20), which means HADDR[4:2] equals DA[4:2] and DD1 is read data of PIO address 0x7.

PIO Mode Control Register (PIOCTRL)

0xF0030040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									SYNC	MD					STP
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STP PW HLD RDY															

SYNC[22:21]

Sync Bit Register for IORDY/DDMARDYn

00	Bypass
01	1 SYNC
10	2 SYNC

These bits are needed to synchronize from HDD clock domain to AHB clock domain for IORDY (and DDMARDYn) signal. The default value is 2 sync.

MD[20]

MODE SEL

0	PIO MODE
1	UDMA(IN/OUT) MODE

To Operate PIO mode, clear this bit and to operate UDMA mode, set this bit.

STP[16:13]

Number Of Cycle for Setup Time

STP (STP value + 1) AHB cycles are issued

These bits define the time requirement from the active CSn to the active DIORn (or DIOWn) in PIO MODE. For further details, see Figure 10.2.

PW[12:7]

Number Of Cycle for Pulse Width

PW (PW value + 1) AHB cycles are issued

These bits define the time requirement for active duration of the DIORn (or DIOWn) . For further details, See Figure 10.2.

HLD[6:1]

Number Of Cycle for HOLD Time

HLD (HLD value + 1) AHB cycles are issued

These bits define the time requirement from the negation of the DIORn (or DIOWn) to the negation of the CSn. For further details, see Figure 10.2.

IORDY[0]	IORDY Enable
0	PW cycles is irrelative of IORDY
1	PW cycles are extended by IORDY

When this bit is set, PW Cycles are extended by IORDY. For further details, see Figure 10.2

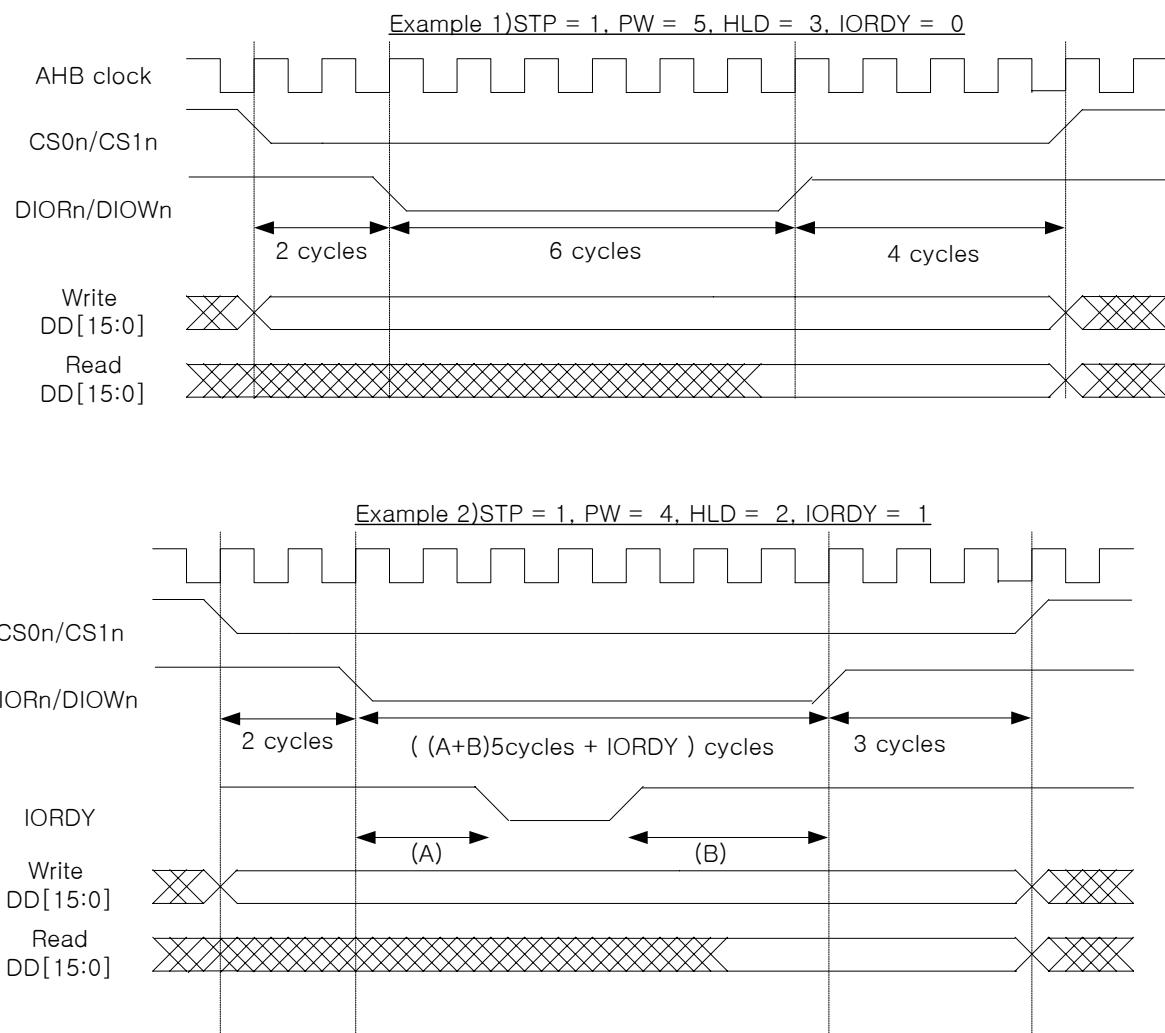


Figure 10.2 PIO Interface Timing Diagrams

UDMA Mode Control Register (UDMACTRL)

0xF0030044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		HRTL[1:0]		OTL[2:0]		ITL[1:0]						FDV[6:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AHM		MAX3[1:0]	0	SNH[2:0]			0		HTB	TS	MS	UHT	UIO	UEN	

HRTL[28:27]		HDMARDYn Trigger Level	
00	4		
01	8		
10	12		
11	14		

In UDMA-IN mode, before UDMA-IN FIFO is full, Host must negate HDMARDYn signal. User must control these bits to prevent UDMA-IN FIFO from being full. These bits are only relative to UDMA-IN mode. If the number of valid data in UDMA-IN FIFO is more than this trigger level, HDMARDYn signal is automatically negated. This trigger level must be assigned the value which is equal to or greater than ITL (Input Trigger Level).

OTL[27:25]		Output Trigger Level	
000	Single Transfer (1read/ 1write)		
001	4 (vcnt <= 12)		
010	8 (vcnt <= 8)		
011	12 (vcnt <=4)		
110	16 (empty)		

In UDMA-OUT mode, the output trigger level (OTL) is relative to Transfer Size (TS[0])bit. IDMA writes data to UDMA-OUT FIFO when it's possible to write as many data as the number of trigger level.

* For the better performance, use 32bit Transfer Mode (TS == 1)

ITL[24:23]		Input Trigger Level	
00	Single Transfer		
01	4 (32bit data) (vcnt >= 4)		
10	8 (32bit data) (vcnt >=8)		
11	12 (32bit data) (vcnt >= 12)		

Note. UDMA-IN mode, the Input Trigger Level is only relative to 32bit data.

IDMA reads from the UDMA-IN FIFO whenever the number of valid data is over than input trigger level.

* For the better performance, use 32bit Transfer Mode (TS == 1)

FDV[22:16]		Frequency Divided Value	
FDV	Frequency Divided Value		

In case UDMA-IN mode, FDV value is needed to control timing parameters which is relative to MAX3 register.

In case UDMA-OUT mode, FDV value controls HSTROBE frequency. Don't set this value to zero in UDMA-OUT mode. Figure 10.3 shows the meaning of these bits.

AHM[15]	Adjust Hold margin
0	The hold margin of the data to be transferred has one AHB clock period.
1	The hold margin of the data to be transferred has two AHB clock period

When this bit is clear, the data to be transferred changes after one rising AHB clock is occurred. (Data hold margin is one clock)

In UDMA-OUT mode, when this bit is set to 1 and the values of FDV register are over than 2, the data to be transferred changes after two rising AHB clock is occurred. (Data hold margin is two clocks)

MAX3[13:12]	MAX3 Cycles (wait number of FDV cycles)
MAX3	(MAX3 + 1) FDV cycles are issued before host generate STOP signal in UDMA-IN MODE.

In UDMA-IN mode, before terminating operation, Host shall receive zero, one, two or three additional data word after negating DMARDYn. Therefore, delay is needed to operate correctly.

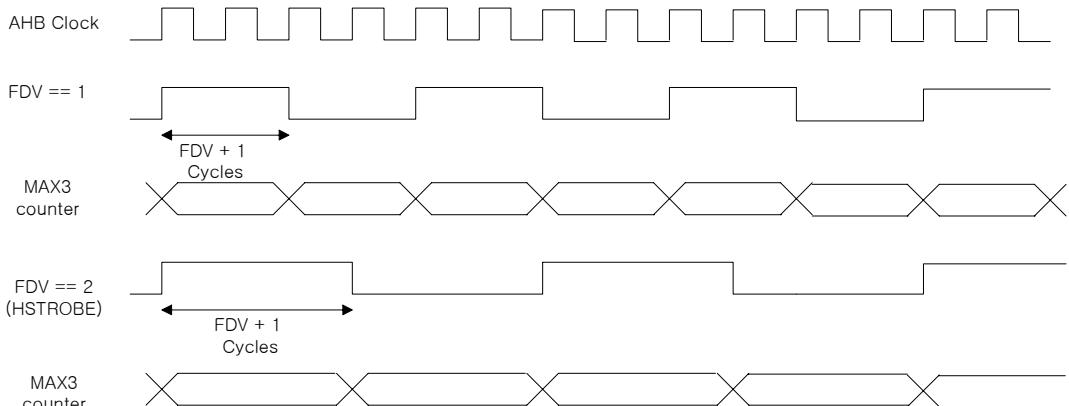


Figure 10.3 HSTROBE and Max3 Counter Timing Diagrams

SNH[10:8]	Setup and Hold time
SNH	Each TACK, TENV, TMLI is extended for (SNH+1) AHB cycles Tss is extended for (2*SNH+1) AHB cycles

There are the timing requirements in each mode. For example, in case UDMA mode4, TACK: min 20ns, TENV: min 20ns and max 55ns, TMLI : min 20ns, Tss : min 50ns is needed.

HTB[5]	Host Terminate By Transfer Counter
1	HOST terminates operation when CTCNT reaches zero.

When this bit is enabled and all data which is specified by UDMATCNT register are transferred completely, host automatically terminates all operation and then sets OPS bit to 1. If OPE is enabled, host generates interrupt to CPU.

When this bit is not enabled and CTCNT and UEN are enabled, Host wait until Device terminates operation.

TS[4]	Transfer Size
0	16 bit Transfer Mode (in Internal Bus)
1	32 bit Transfer Mode (in Internal Bus)

MS [3]	Multiple Section Enable Bit
1	To Support Multiple Section Toggle '1' is recommended

This bit should be set to one for normal operation. Don't clear this bit.

HT [2]	Host Termination
HT	Host Termination

Host can terminate UDMA operation normally by force using this bit. When this bit is set, IDE interface signals are negated or active by standard interface protocol.

UIO [1]	UDMA IO Configure
0	UDMA-IN MODE
1	UDMA-OUT MODE

When this bit is cleared, IDE Controller will receive the data from Device and then will send to CPU or General DMA. When this bit is set. IDE Controller will receive from CPU or general DMA and then will send the data to Devices.

UEN [0]	UDMA Enable Bit
1	UDMA Enable Bit

This bit is UDMA enable bit. When Operation is over, this bit is automatically cleared.

When this bit is clear by CPU, internal state goes to idle state and IDE interface signals are negated abnormally.

IDMA Control Register (IDMACTRL) **0xF0030048**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

LOC [3]**Locked Transfer**

1 DMA transfer executed with lock transfer

Lock field controls the LOCK signal (refer to AHB specification). When the LOCK is set to 1, the DMA transfer doesn't be bothered by other AHB masters like LCD controller, ARM etc. This field is only meaningful in case of non-burst type transfers.

CON [2]**Continuous Transfer**

0 DMA transfer begins from CSAR / CDAR address.

DMA transfer begins from CSAR / CDAR address.

1 It must be used after the former transfer has been executed, so that CSAR and CDAR contain a meaningful value.

REP [1]**Internal DMA Repeat**

0 When UDMA operation end, IDE bit is automatically disabled.

1 When UDMA operation end, IDE bit is not disabled.

IDE [0]**Internal DMA Enable**

1 Internal DMA (IDMA) enable bit.

IDMA Source Address Register (IDMASA)

0xF003004C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR[15:0]															

This register contains the start address of source memory block for IDMA transfer. The transfer begins reading data from this address. This register is only relative of UDMA-OUT mode.

IDMA Source Parameter Register (IDMASP)															0xF0030050
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMASK[23:8]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMASK[7:0]								SINC[7:0]							

SMASK [31:8]	Source Address Mask Register
0	non-masked
1	Masked so that source address bit doesn't be changed during DMA transfer

Each bit field controls the dedicated bit of source address field. That is, if SMASK[23] is set to 1, the 28th bit of source address is masked, and if SMASK[22] is set to 1, the 27th bit of source address is masked, and so on. If a bit is masked, a corresponding bit of address bus is not changed during DMA transfer. This function can be used to generate circular buffer address.

SINC [7:0]	Source Address Increment Register
sinc	Source address is added by amount of sinc at every write cycles. sinc is represented as 2's complement, so if SINC[7] is 1, the source address is decremented.

The addresses of DMA transfer are 32bit wide, but the upper 4bit of them are not affected during DMA transfer. If the source or destination address reaches its maximum address space like 0x7FFFFFFF or 0x2FFFFFFF, the next transfer is starting from 0x70000000 or 0x20000000 not from 0x80000000 or 0x30000000.

IDMA Current Source Address Register (IDMACSA)																0xF0030054			
31																			
CSAR[31:16]																			
15																			
CSAR[15:0]																			

This register contains the current source address of DMA transfer. It represents that the current transfer read data from this address. This is read only register.

IDMA Destination Address Register (IDMADA)																0xF0030058	
31																	
15																DAR[31:16]	
15																DAR[15:0]	

This register contains the start address of destination memory block for DMA transfer.

This register is only relative of UDMA-IN mode.

IDMA Destination Parameter Register (IDMADP)																0xF003005C
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	DMASK[23:8]
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	DMASK[7:0]
DINC[7:0]																DINC[7:0]

DMASK [23:8]	Destination Address Mask Register
0	non-masked
1	Masked so that destination address bit doesn't be changed during DMA transfer

Each bit field controls the corresponding bit of source address field. That is, if DMASK[23] is set to 1, the 28th bit of source address is masked, and if DMASK[22] is set to 1, the 27th bit of source address is masked, and so on. If a bit is masked, a corresponding bit of address bus is not changed during DMA transfer. This function can be used to generate circular buffer address.

DINC [7:0]	Destination Address Increment Register
dinc	Destination address is added by amount of dinc at every write cycles. dinc is represented as 2's complement, so if DINC[7] is 1, the destination address is decremented.

The addresses of DMA transfer are 32bit wide, but the upper 4bit of them are not affected during DMA transfer. If the source or destination address reaches its maximum address space like 0x7FFFFFFF or 0x2FFFFFFF, the next transfer is starting from 0x70000000 or 0x20000000 not from 0x80000000 or 0x30000000.

IDMA Current Destination Address Register (IDMACDA)																0xF0030060			
31																CDAR[31:16]			
15																CDAR[15:0]			

This register contains current destination address of DMA transfer. It represents that the current transfer write data to this address. This is read only register.

IDEINT Register

0xF0030064

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UB	NWA	DT	0		VEE[4:0]					0		VOE[4:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			URS	FES	TRS	OPS				0		URE	FEE	TRE	OPE

UB[31]	Unit Busy
0	State of UDMA is IDLE.
1	UDMA is BUSY.

This bit is set while UDMA is running operation.

NWA[30]	Not Word Aligned
0	Word (32 bits) aligned.
1	Not Word (32 bits) aligned.

This bit is for 32bit transfer mode(TS == 1).

This bit is set when UDMA -IN/OUT Operation is ended, remain FIFO data is not word aligned. When TS= = 1(32bit transfer mode) and this bit is set, CPU or DMA have to read last 16bit data.

DT[28]	Device Terminate
0	Device did not terminate operation.
1	Device terminated operation.

This bit is set when Device terminates operation before host terminates operation.

VEE[26:22]	Valid EVEN Entry
VEE	- Number of Valid data in Even FIFO when UDMA-IN mode
	- Number of Valid data in UDMA-OUT FIFO when UDMA-OUT mode

These bits specify for UDMA-IN mode. When operation is ended, these bits represent how many valid data is remain in even FIFO . Data needs to be read after internal DMA transfer is ended, because data may be remained in the receive FIFO when the receive FIFO trigger condition is not satisfied. Width of even FIFO is 16bit.

In case UDMA-OUT mode, these bits represent Number of data that are not transferred in UDMA-OUT FIFO.

VOE[26:22]	Valid ODD Entry
VOE	Number of Valid data in Odd FIFO When UDMA-IN mode

These bits specify for UDMA-IN mode. When operation is ended, these bits represent how many valid data is remain in odd FIFO. Data needs to be read after internal DMA transfer is ended, because data may be remained in the receive FIFO when the receive FIFO trigger condition is not satisfied .Width of odd FIFO is 16bit.

Cf. 32bit data of UDMA-IN FIFO is composed of one odd FIFO 16bit data and one even FIFO 16bit data.

For example, If VEE is equal to 0x6 and VOE is equal to 0x5 when operation ends, CPU or DMA has to read (5 x 32 bit + 1 x 16 bit) data.

URS[11]	UDMA Request Status
0	UDMA Request from the Device is not active or automatically cleared.
1	UDMA Request from the Device is active

This bit is set when DMARQ from the Device is active high. This bit is automatically cleared when UEN bit is set.

FES[10]	FIFO Error Status
0	Not FIFO Error
1	FIFO Error

This bit is set when FIFO Error occurs. In this case, Host have to terminate operation using HT bit, and check Frequency and timing parameter, and then try transfer again. This bit is cleared when this bit is written to 1.

TRS[9]	Trigger Level Request Status
0	Trigger Level Request is not active or automatically cleared.
1	Trigger Level Request is active

When this bit is set, CPU can read or write data. This bit is never set when IDE (Internal DMA Enable bit) is set. This bit is cleared when the number of valid data in In(Out)-FIFO is less(more) than Trigger level.

OPS[8]	Operation End Status
0	Host or Device does not terminate operation yet.
1	Host or Device terminates operation.

This bit is set when the operation is ended and cleared when this bit is written to 1.

URE[3]	UDMA Request Interrupt Enable
0	UDMA Request Interrupt is disabled.
1	UDMA Request Interrupt is enabled.

FEE[2]	FIFO Error Interrupt Enable
0	FIFO Error Interrupt is disabled.
1	FIFO Error Interrupt is enabled.

TRE[1]	Trigger Level Interrupt Enable
0	Trigger Level Interrupt is disabled.
1	Trigger Level Request Interrupt is enabled.

OPE[0]	Operation End Interrupt Enable
0	Operation End Interrupt is disabled in UDMA Mode.
1	Operation End Interrupt is enabled in UDMA Mode.

UDMATCNT Register

0xF0030068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDMATCNT[23:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

UDMATCNT[15:0]

These bits define how many data need to be transfer By host. The host will transfer 2*UDMATCNT bytes in UDMA-OUT mode and 4*UDMATCNT bytes in UDMA-IN mode.

UDMAIN Register (Read Only)																0xF003006C			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	UDMAIN[31:16]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	UDMAIN[15:0]			

This register is Read Access port to read data (received from the Device) in IDE FIFO. When TS is cleared (Transfer size: 16bits), UDMAIN[31:16] will be zero and UDMAIN[15:0] is valid data.

UDMAOUT Register (Write Only)

0xF0030070

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDMAOUT[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDMAOUT[15:0]															

This register is Write Access port to write data(to be send to Device) in IDE FIFO. When TS is cleared (Transfer size: 16bits), only UDMAOUT[15:0] is valid data.

UDMACRC Register (Read Only)																0xF0030074
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	UDMACRC[15:0]

When operation is ended, it's possible to check difference between Device CRC value and Host CRC value. If each CRC value is different, the operation was ended incorrectly.

UDMACTCNT Register

0xF0030078

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTCNT[23:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTCNT[15:0]															

CTCNT[23:0]

Current TCNT

CTCNT Current Transfer Counter.

At the beginning of Transfer, The CTCNT is updated by TCNT value.

When UDMA is Enabled, CTCNT will be decremented according to TS (transfer Size). CPU can check how many data was transferred by checking these bits. When TCE bit is set and all number of data specified by UDMATCNT was transferred (CTCNT reaches zero), Host will terminate operation.

11 DMA CONTROLLER

11.1 Overview

The TCC79XX has three 3-channel general DMA (GDMA) controllers for data transfer. The block diagram of GDMA controller is in the following figure.

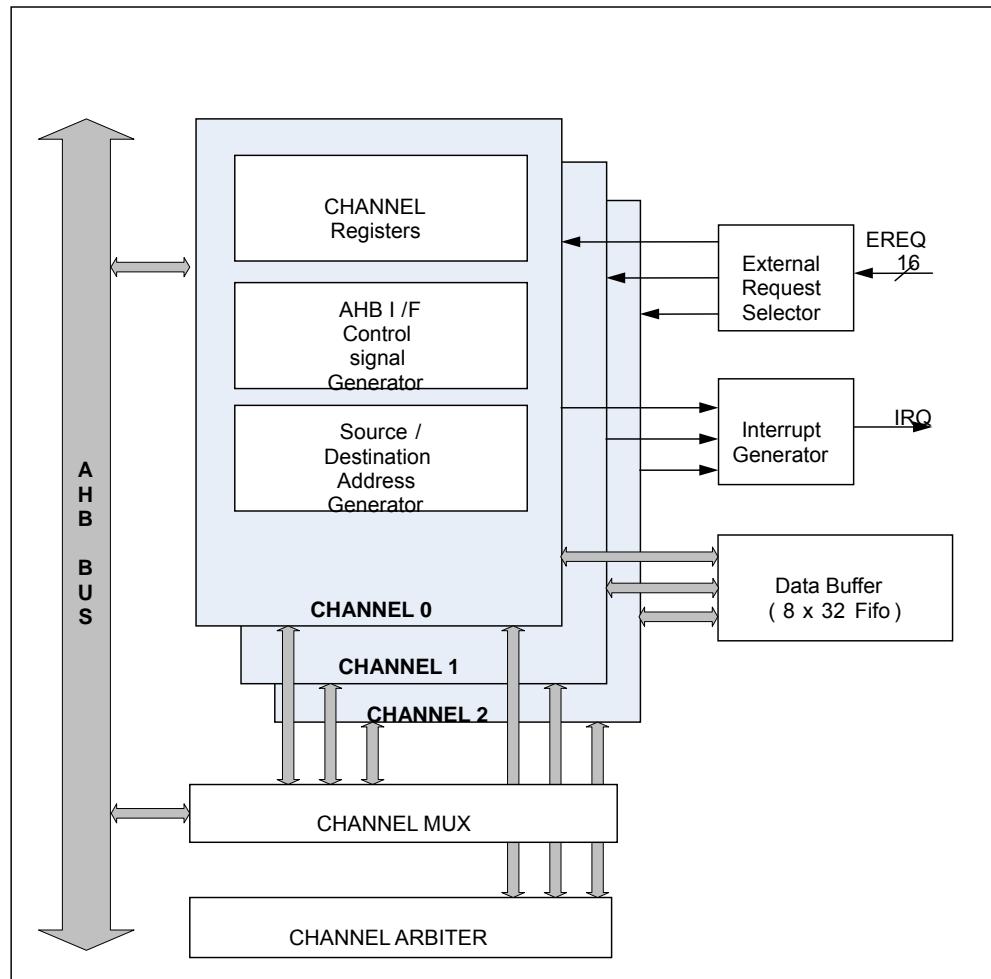


Figure 11.1 GDMA Controller Block Diagram

11.2 Register Description

Table 11.1 General DMA Controller Register Map

	Name	offset	Type	Reset	Description
C	<u>ST_SADR0</u>	0x00	R/W	0x00000000	Start Address of Source Block
H	<u>SPARAM0</u>	0x04	R/W	0x00000000	Parameter of Source Block
A	<u>C_SADR0</u>	0x0C	R	0x00000000	Current Address of Source Block
N	<u>ST_DADR0</u>	0x10	R/W	0x00000000	Start Address of Destination Block
N	<u>DPARAM0</u>	0x14	R/W	0x00000000	Parameter of Destination Block
E	<u>C_DADR0</u>	0x1C	R	0x00000000	Current Address of Destination Block
L	<u>HCOUNT0</u>	0x20	R/W	0x00000000	Initial and Current Hop count
	<u>CHCTRL0</u>	0x24	R/W	0x00000000	Channel Control Register
0	<u>RPTCTRL0</u>	0x28	R/W	0x00000000	Repeat Control Register
	<u>EXTREQ0</u>	0x2C	R/W	0x00000000	External DMA Request Register
C	<u>ST_SADR1</u>	0x30	R/W	0x00000000	Start Address of Source Block
H	<u>SPARAM1</u>	0x34	R/W	0x00000000	Parameter of Source Block
A	<u>C_SADR1</u>	0x3C	R	0x00000000	Current Address of Source Block
N	<u>ST_DADR1</u>	0x40	R/W	0x00000000	Start Address of Destination Block
N	<u>DPARAM1</u>	0x44	R/W	0x00000000	Parameter of Destination Block
E	<u>C_DADR1</u>	0x4C	R	0x00000000	Current Address of Destination Block
L	<u>HCOUNT1</u>	0x50	R/W	0x00000000	Initial and Current Hop count
	<u>CHCTRL1</u>	0x54	R/W	0x00000000	Channel Control Register
1	<u>RPTCTRL1</u>	0x58	R/W	0x00000000	Repeat Control Register
	<u>EXTREQ1</u>	0x5C	R/W	0x00000000	External DMA Request Register
C	<u>ST_SADR2</u>	0x60	R/W	0x00000000	Start Address of Source Block
H	<u>SPARAM2</u>	0x64/0x68	R/W	0x00000000	Parameter of Source Block
A	<u>C_SADR2</u>	0x6C	R	0x00000000	Current Address of Source Block
N	<u>ST_DADR2</u>	0x70	R/W	0x00000000	Start Address of Destination Block
N	<u>DPARAM2</u>	0x74/0x78	R/W	0x00000000	Parameter of Destination Block
E	<u>C_DADR2</u>	0x7C	R	0x00000000	Current Address of Destination Block
L	<u>HCOUNT2</u>	0x80	R/W	0x00000000	Initial and Current Hop count
	<u>CHCTRL2</u>	0x84	R/W	0x00000000	Channel Control Register
2	<u>RPTCTRL2</u>	0x88	R/W	0x00000000	Repeat Control Register
	<u>EXTREQ2</u>	0x8C	R/W	0x00000000	External DMA Request Register
	<u>CHCONFIG</u>	0x90	R/W	0x00000000	Channel Configuration Register

The GDMA registers are listed in Table 11.1. The TCC79XX has 3 GDMA, which are GDMA0, GDMA1, and GDMA2. And their base addresses are 0xF0040000, 0xF0040100, and 0xF0040200 respectively. One GDMA has 3 channels, which are Channel 0, Channel 1 and Channel 2. Channel offset is 0x00, 0x30, and, 0x60 as shown in Table 11.1 and Table 11.2.

Table 11.2 BASE Address of All GDMA Channels

GDMA NAME	GDMA BASE	CHANNEL #	BASE Address
GDMA0	0xF0040000	0	0xF0040000
		1	0xF0040030
		2	0xF0040060
GDMA1	0xF0040100	0	0xF0040100
		1	0xF0040130
		2	0xF0040160
GDMA2	0xF0040200	0	0xF0040200
		1	0xF0040230
		2	0xF0040260

Start Source Address Register (ST_SADR)																BASE ⁴ +0x00
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	ST_SADR[31:16]
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	ST_SADR[15:0]

This register contains the start address of source memory block for DMA transfer. The transfer begins reading data from this address.

Source Block Parameter Register (SPARAM)																BASE + 0x04
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	SMASK[23:8]
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SMASK[7:0]
																SINC[7:0]

SMASK[23:0]	[31:8]	Source Address Mask Register
0	non-masked	
1	Masked so that source address bit doesn't be changed during DMA transfer	

Each bit field controls the dedicated bit of source address field. That is, if SMASK[23] is set to 1, the 28th bit of source address is masked, and if SMASK[22] is set to 1, the 27th bit of source address is masked, and so on. If a bit is masked, a corresponding bit of address bus is not changed during DMA transfer. This function can be used to generate circular buffer address.

SINC[7:0]	[7:0]	Source Address Increment Register
sinc	Source address is added by amount of sinc at every write cycles. sinc is represented as 2's complement, so if SINC[7] is 1, the source address is decremented.	

The addresses of DMA transfer are 32bit wide, but the upper 4bit of them are not affected during DMA transfer. If the source or destination address reaches its maximum address space like 0x7FFFFFFF or 0x2FFFFFFF, the next transfer is starting from 0x70000000 or 0x20000000 not from 0x80000000 or 0x30000000.

Current Source Address Register (C_SADR)																BASE + 0x0C
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	C_SADR[31:16]
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	C_SADR[15:0]

This register contains the current source address of DMA transfer. It represents that the current transfer read data from this address. This is read only register.

⁴ BASE = 0xF0040nm0 (n = 0, 1, or 2 m= 0, 3, or 6). Refer to “BASE Address” column of “Table 11.2”.

Start Destination Address Register (ST_DADR)

BASE⁵ + 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST_DADR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST_DADR[15:0]															

This register contains the start address of destination memory block for DMA transfer.

Destination Block Parameter Register (DPARAM)

BASE + 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMASK[23:8]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMASK[7:0]								DINC[7:0]							

DMASK[28:0] [31:8] Destination Address Mask Register

0	non-masked
---	------------

1	Masked so that destination address bit doesn't be changed during DMA transfer
---	---

Each bit field controls the corresponding bit of source address field. That is, if DMASK[23] is set to 1, the 28th bit of source address is masked, and if DMASK[22] is set to 1, the 27th bit of source address is masked, and so on. If a bit is masked, a corresponding bit of address bus is not changed during DMA transfer. This function can be used to generate circular buffer address.

DINC[7:0] [7:0] Destination Address Increment Register

Destination address is added by amount of dinc at every write cycles.

dinc	dinc is represented as 2's complement, so if DINC[7] is 1, the destination address is decremented.
------	--

The addresses of DMA transfer have 32bit wide, but the upper 4bit of them are not affected during DMA transfer. If the source or destination address reaches its maximum address space like 0x7FFFFFFF or 0x2FFFFFFF, the next transfer is starting from 0x70000000 or 0x20000000 not from 0x80000000 or 0x30000000.

Current Destination Address Register (C_DADR)

BASE + 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C_DADR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C_DADR[15:0]															

This register contains current destination address of DMA transfer. It represents that the current transfer write data to this address. This is read only register.

⁵ BASE = 0xF0040nm0 ($n = 0, 1$, or 2 $m = 0, 3$, or 6). Refer to “BASE Address” column of “Table 11.2”.

HOP Count Register (HCOUNT)**BASE⁶ + 0x20**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C_HCOUN[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST_HCOUN[15:0]															

C_HCNT[15:0]	[31:16]	Current Hop Count
cn	Represent cn number of Hop transfer remains	
ST_HCNT[15:0]	[15:0]	Start Hop Count
sn	DMA transfers data by amount of sn Hop transfers	

At the beginning of transfer, the C_HCNT is updated by ST_HCNT register. At the end of every hop transfer, this is decremented by 1 until it reaches to zero. When this reaches to zero, the DMA finishes its transfer and may or may not generate its interrupt according to IEN flag of CHCTRL register.

⁶ BASE = 0xF0040nm0 (n = 0, 1, or 2 m= 0, 3, or 6). Refer to “BASE Address” column of “Table 11.2”.

Channel Control Register (CHCTRL)

BASE⁷ + 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CONT	[15]	Issue Continuous Transfer
0	DMA transfer begins from ST_SADR / ST_DADR address	
1	DMA transfer begins from C_SADR / C_DADR address	
	It must be used after the former transfer has been executed, so that C_SADR and C_DADR contain a meaningful value.	
DTM	[14]	Differential Transfer Mode
0	Differential Transfer Mode Disable	
	Differential Transfer Mode Enable	
	for WSIZE = 10 and BSIZE = 11: 32 bit-to- 16bit transfer	
1	4 Read(Word Unit) / 8 Write(Half-Word Unit) for WSIZE = 11 and BSIZE = 11 : 16 bit-to- 32bit transfer. 8 Read(Half-Word Unit) / 4 Write(Word Unit)	
SYNC	[13]	Hardware Request Synchronization
0	Do not Synchronize Hardware Request.	
1	Synchronize Hardware Request.	
HRD	[12]	Hardware Request Direction
0	ACK/EOT signals are issued when DMA-Read Operation.	
1	ACK/EOT signals are issued When DMA-Write Operation.	
LOCK	[11]	Issue Locked Transfer
1	DMA transfer executed with lock transfer	

Lock field controls the LOCK signal (refer to AHB specification). When the LOCK is set to 1, the DMA transfer is not bothered by other AHB masters like LCD controller, ARM etc. This field is only meaningful in case of non-burst type transfers.

BST	[10]	BURST Transfer
0	DMA transfer executed with arbitration.	
1	DMA transfer executed with no arbitration. (burst operation)	

Arbitration means that at the end of every HOP transfer, the AHB bus is released from DMA channel so other master can occupy the bus when that master has requested the bus.

Burst means that once the DMA request occurs, all of transfers are executed without further DMA requests.

⁷ BASE = 0xF0040nm0 (n = 0, 1, or 2 m= 0, 3, or 6). Refer to “BASE Address” column of “Table 11.2”.

TYPE[1:0]	[9:8]	Transfer Type
00	SINGLE	transfer with edge-triggered detection
11	SINGLE	transfer with level-sensitive detection
01	HW	transfer
10	SW	transfer

In SINGLE Type, after one Hop data transferring DMA checks External DMA Request (DREQ) and then if its bit is active, DMA transfers next hop data. DREQ is detected level-sensitive or edge-triggered by SINGLE transfer TYPE.

The 1 Hop of transfer means 1 burst of read followed by 1 burst of write. 1 burst means 1, 2 or 4 consecutive read or write-cycles defined by BSIZE field of CHCTRL register. The Figure 11.2 illustrates the relation among the above transfers.

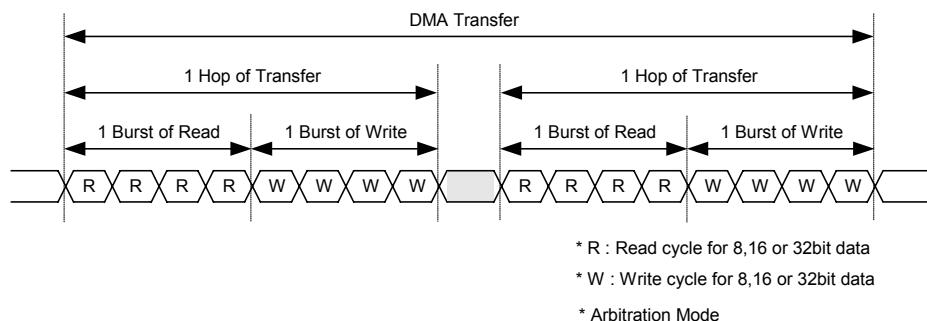


Figure 11.2 Relation between Hop and Burst Transfers (If burst size is 4.)

Hardware type transfer means that the DMA transfer is triggered by external or internal hardware blocks selected by DMASEL field in EXTREQn register.

Software type transfer means that the DMA transfer is triggered by EN bit of CHCTRL register. When this is set to 1, transfer request signal is generated internally and then the transfer begins immediately.

Hardware demand type transfer (HW_DEMAND) means that once the DMA request occurs, DMA checks request signal each hope transfer, and if request signal is set, DMA transfer one hope's data. After transferring all hope's data, DMA operation will be finished.

Figure 11.3 is the example of various types of transfer.

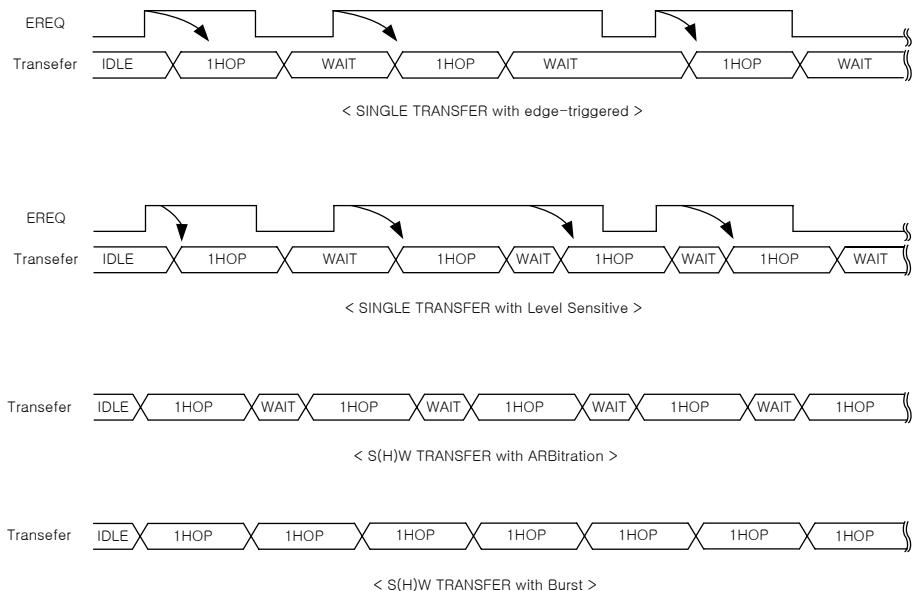


Figure 11.3 The Example Of Various Types of Transfer.

BSIZE[1:0]	[7:6]	Burst Size
0	1	1 Burst transfer consists of 1 read or write cycle.
1	2	1 Burst transfer consists of 2 read or write cycles
2	4	1 Burst transfer consists of 4 read or write cycles
3	8	1 Burst transfer consists of 8 read or write cycles
WSIZE[1:0]	[5:4]	Word Size
0	8	Each cycle read or write 8bit data
1	16	Each cycle read or write 16bit data
2, 3	32	Each cycle read or write 32bit data
FLAG	[3]	DMA Done Flag
1		Represents that all hop of transfers are fulfilled. When writing 1 to this bit, it is cleared to 0

It is not automatically cleared by starting another transfer, so before starting any other DMA transfer, user must clear this flag to 0 for checking DMA status correctly.

IEN	[2]	Interrupt Enable
1		At the same time the FLAG goes to 1, DMA interrupt request is generated.

To generate IRQ or FIQ interrupt, the DMA flag of IEN register in the interrupt controller must be set to 1 ahead.

REP	[1]	Repeat Mode Control
0		After all of hop transfer has executed, the DMA channel is disabled
1		The DMA channel remains enabled. When another DMA request has occurred, the DMA channel start transfer data again with the same manner (type, address, increment, mask) as the latest transfer of that channel.

EN	[0]	DMA Channel Enable
		DMA channel is terminated and disabled. It does not affect the HCOUNT register, so if the current hop counter is not zero when channel is disabled, it is possible that the transfer illegally starts right after channel is re-enabled. Make sure that HCOUNT is zero not to continue transfer after channel is re-enabled.
1		DMA channel is enabled. If software type transfer is selected, this bit generates DMA request directly, or if hardware type transfer is used, the selected interrupt request flag generate DMA request.

Repeat Control Register (RPTCTRL)

BASE⁸ + 0x28

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRI	EOT				0							RPTCNT[23:16]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RPTCNT[15:0]

DRI	[31]	Disable Repeat Interrupt
0		DMA Interrupt is occurred when the end of each Repeated DMA operation.
1		DMA Interrupt occur is occurred when the last DMA Repeated DMA operation

This bit is meaningful when Repeat Mode is enabled.

EOT	[30]	EOT
0		EOT signal is occurred when the end of each Repeated DMA operation in HW(including Single) transfer Mode.
1		EOT Signal is occurred when the last Repeated DMA operation in HW(including Single) transfer Mode.

This bit is meaningful when Repeat Mode is enabled.

RPTCNT[23:0]	[23:0]	Repeat Count
0		DMA transfer data endlessly.
None zero		DMA transfer the number of (N + 1) * HCOUNT data

This bit is meaningful when Repeat Mode is enabled. When this bit is cleared in repeat mode, DMA will run endlessly. To exit endless repeat mode, clear EN bit of DMACTRL or disable Repeat Mode. It's possible to circular transfer using repeat count.

⁸ BASE = 0xF0040nm0 (n = 0, 1, or 2 m= 0, 3, or 6). Refer to "BASE Address" column of "Table 11.2".

External DMA Request Register (EXTREQn)

BASE⁹ + 0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMASEL[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMASEL[15:0]															

DMASEL	Connected Hardware
[0]	GPSB Channel 0 TX
[1]	GPSB Channel 1 TX
[2]	GPSB Channel 2 TX
[3]	GPSB Channel 3 TX
[4]	GPSB Channel 0 RX
[5]	GPSB Channel 1 RX
[6]	GPSB Channel 2 RX
[7]	GPSB Channel 3 RX
[8]	UART Channel 2 Transmitter
[9]	UART Channel 2 Receiver
[10]	UART Channel 3 Transmitter
[11]	UART Channel 3 Receiver
[12]	ECC
[15:13]	-
[16]	-
[17]	Memory Stick
[18]	NAND Flash Controller
[19]	I2C Channel 0
[20]	SPDIF Packet(Audio) Data
[21]	SPDIF User Data
[22]	CD I/F
[23]	DAI Transmitter
[24]	DAI Receiver
[25]	I2C Channel 1
[26]	UART Channel 0 Transmitter
[27]	UART Channel 0 Receiver
[28]	-
[29]	UART Channel 1 Transmitter
[30]	UART Channel 1 Receiver
[31]	-

⁹ BASE = 0xF0040nm0 ($n = 0, 1$, or 2 $m = 0, 3$, or 6). Refer to “BASE Address” column of “Table 11.2”.

Channel Configuration Register (CHCONFIG)

GBASE¹⁰ +0x90

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					0				IS2	IS1	IS0	0	MIS2	MIS1	MIS0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				0	SWP2	SWP1	SWP0	0		PRI[2:0]		0			FIX

IS2	[22]	Channel 2 Alternate interrupt status
0	No interrupt in the corresponding channel	
1	The corresponding channel interrupt is occurred	

Without regard to Interrupt enable bit(IEN) of the channel2, this bit indicates the channel2 interrupt status.

This bit is automatically cleared when FLAG bit of the corresponding channel is cleared. This bit is read only.

IS1	[21]	Channel 1 Alternate interrupt status
0	No interrupt in channel 1	
1	Channel1 Interrupt is occurred	

Except for channel difference, this bit is the same as IS2 bit.

ISO	[20]	Channel 0 Alternate interrupt status
0	No interrupt in channel 0	
1	Channel1 Interrupt is occurred	

Except for channel difference, this bit is the same as IS2 bit.

MIS2	[18]	Channel2 Masked Interrupt Status
0	Masked interrupt is not occurred in channel 2	
1	Channel2 Masked Interrupt is occurred	

This bit is set when the channel2 interrupt occurs and interrupt enable bit (IEN) of channel2 is set. This bit is automatically cleared when FLAG bit of the channel2 is cleared. This bit is read only.

MIS1	[17]	Channel1 Masked Interrupt Status
0	Masked interrupt is not occurred in channel 1	
1	Channel1 Masked Interrupt is occurred	

Except for channel difference, this bit is the same as MIS2 bit.

MISO	[16]	Channel0 Masked Interrupt Status
0	Masked interrupt is not occurred in channel 0	
1	Channel0 Masked Interrupt is occurred	

Except for channel difference, this bit is the same as MIS1 bit.

¹⁰ GBASE = 0xF0040n00 (n = 0, 1, or 2). Refer to “GDMA BASE” column of “Table 11.2”.

SWP2	[10]	Channel2 SWAP Enable bit
0	Do not swap Data.	
1	Swap Channel Data.	

When this bit is set, data to be written to destination address will be swapped.

For example, the 32bit source data which consists of 4bytes {D3, D2, D1, D0} will be stored {D0, D1, D2, D3} in destination address. The 16bit source data which consists of 2bytes {D1, D0} will be stored {D0,D1} in destination address.

SWP1	[9]	Channel1 SWAP Enable bit
0	Do not Swap Channel1 Data.	
1	Swap Channel1 Data.	

Except for channel difference, the function controlled by this bit is the same as its SWP2 bit.

SWP0	[8]	Channel0 SWAP Enable bit
0	Do not Swap Channel0 Data.	
1	Swap Channel0 Data.	

Except for channel difference, the function controlled by this bit is the same as its SWP2 bit.

PRI[2:0]	[5:4]	Channel Priority
000	CH0 > CH1 > CH2	
001	CH0 > CH2 > CH1	
010	CH1 > CH0 > CH2	
011	CH1 > CH2 > CH0	
100	CH2 > CH1 > CH0	
101	CH2 > CH0 > CH1	

PRI bits is meaningful when fix[0] bit is enabled.

FIX	[0]	Fixed Priority Operation
0	Round-Robin (Cyclic) Mode.	
1	Fixed Priority Mode.	

In round-robin mode, each channel is enabled one by one every one hop transferring.

In Fixed mode, according to PRI bit, the highest channel is serviced first and lower priority channel is serviced after higher priority channel operation is finished. See Figure 11.4 for more information.

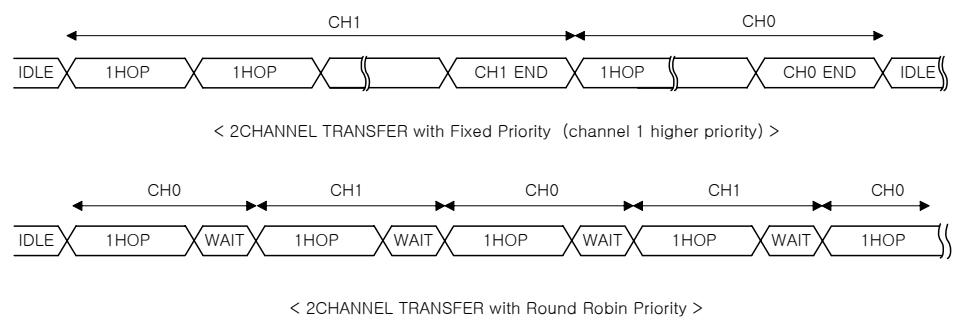


Figure 11.4 Data transfer when Channel0 and Channel1 are enabled

12 MEMORY STICK HOST CONTROLLER

12.1 Overview

This is a host controller with 32-bit CPU interface that supports Memory Stick Ver. 1.x and Memory Stick PRO. This conforms to "Memory Stick Standard Format Specifications ver. 1.4-00" and "Memory Stick Standard Memory Stick PRO Format Specifications ver. 1.00-01"

The memory stick host controller has the following features.

- Data transmit/receive FIFO (64 bits x 4)
- Hardware CRC Generator
- Memory Stick Serial Clock (Serial : 20MHz-MAX, Parallel : 40MHz-MAX)
- Burst Transfer via On-Chip DMA Controller

The communication protocol with the Memory Stick is started by writing from CPU to the command register. When the protocol finishes, the CPU is notified that the protocol is ended by an interrupt request.

When the protocol is started and enters the data transfer state, data is requested by issuing a DMA transfer request or an interrupt request to the CPU.

The RDY time-out time when the handshake state (read protocol:BS2, write protocol:BS3) is established in communication with the Memory Stick can be designated as the number of Memory Stick transfer clocks. When a time out occurs, the CPU is notified that the protocol has ended due to a time out error by an interrupt request.

CRC off can be set as a test mode. When CRC off is set, CRC is not added to the data transmitted to the Memory Stick.

The controller has 4 registers named as COMMAND, DATA, STATUS, SYSTEM register. The COMMAND register is for starting protocol communication between the controller and device. The DATA register is for transmit/received data with little-endian format. The STATUS register has the various flags for indicating the controller status, the transmit/receive FIFO status, controller interrupt status, etc. Finally SYSTEM register has the bit-field for controlling the controller such as DMA transfer mode, CRC-off mode, interrupt enable mode, serial/parallel mode, etc.

The Figure 12.1 shows the block diagram of memory stick host controller.

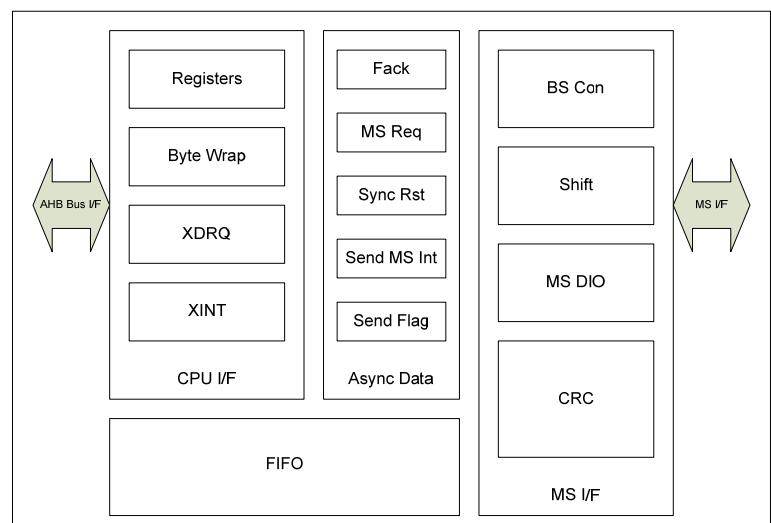


Figure 12.1 Memory Stick Host Controller Block Diagram

12.2 Register Descriptions

Table 12.1 MSHC Register Map (Base Address = 0xF0051000)

Name	Address	Type	Reset	Description
COMMAND	0x00	R/W	0x00000000	Command Register
DATA	0x04	R/W	Unknown	Data Register
STATUS	0x08	R/W	0x00001020	Status Register
SYSTEM	0x0C	R/W	0x00004455	System Control Register

Table 12.2 PORTCFG Register Map (Base Address = 0xF0051080)

Name	Address	Type	Reset	Description
PORTCFG	0x00	R/W	0x00000000	Port Configuration Register

COMMAND Register

0xF0051000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPC		0	DSL	DSZ											

TPC [15:12]	Transfer Protocol Code	
0x2	MS_RD_LDATA	Read Long Data Command
0x3	MS_RD_SDATA	Read Short Data Command
0x4	MS_RD_REG	Read Register Command
0x7	MS_GET_INT	Get Interrupt Command
0xD	MS_WR_LDATA	Write Long Data Command
0xC	MS_WR_SDATA	Write Short Data Command
0xB	MS_WR_REG	Write Register Command
0x8	MS_SET_RW_REG_ADRS	Set R/W Register Address Command.
0xE	MS_SET_CMD	Set Command
0x9	MS_EX_SET_CMD	Exit Set Command

DSL[10]	Data Select Bit (Should be '0')	
0	Internal FIFO – Only Supported	
1	External Memory – Not-Supported	

DSZ[9:0]	Transmit/Receive Data Size	
0 ~ 1023	Number of Transmit/Receive Data Size When '0', the size is 1024 bytes	

When the command register is written, the communication protocol with the Memory Stick starts and data transmit/receive is performed.

The data transfer direction with the Memory Stick is determined from TPC[3]. When TPC[3] = 0, the read protocol is performed, and when TPC[3] = 1, the write protocol is performed. When the TPC transfer direction differs from the direction specified by system register bit FDIR, the TPC[3] value is reflected to system register bit FDIR when the protocol starts.

When the protocol starts, RDY changes to '0' to indicate that protocol execution is underway. RDY = 0 is held during communication with the Memory Stick, and when communication ends an interrupt is generated as RDY = 1.

DATA Register

0xF0051004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BYTE 3								BYTE 2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYTE 1								BYTE 0							

[31:0]

DATA Register

Support of Word Access Only

Byte 0 should be transmitted/received first

STATUS Register

0xF0051008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DRQ	MSINT	RDY	0		CRC	TOE	0		EMP	FUL	CED	ERR	BRQ	CNK

BIT	NAME	R/W	RESET	Description
31-15				RESERVED
14	DRQ	R	0	DMA Request Status 0: Not-requested 1: DMA Requested
13	MSINT	R	0	Interrupt Status 0: Not-Activated 1: Interrupt Activated
12				Device Ready Status 0: Now-Busy State 1: Command Receive Enabled or Protocol Ended
11-10				RESERVED
9	CRC	R	1	CRC Error Status 0: No CRC Error 1: CRC Error Occurred
8	TOE	R	0	Time-Out Error 0: No Time-out Error 1: Time-out Error Occurred
7-6				RESERVED
5	EMP	R	1	FIFO Empty Status 0: Not-Empty : Data Contained 1: FIFO Empty
4	FUL	R	0	FIFO Full Status 0: Not-Full 1: FIFO Full
3				Command End Status – Valid for Parallel Mode 0: Not-End 1: Command Ended
2				Error Status – Valid for Parallel Mode 0: Not-Error 1: Error Occurred ; Cleared by writing command register
1				Data Buffer Request – Valid for Parallel Mode 0: Not- Requested 1: Data Buffer Requested : Cleared by writing command register
0				Command Not Acknowledge – Valid for Parallel Mode 0: Acknowledged 1: Not Acknowledge : Cleared by writing command register

SYSTEM Register

0xF005100C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RST	SRAC	INTEN	NCRC	ICLR	MSIEN	FCLR	FDIR	DAM	DRM	DRSL	REI	REQ	BSY		

BIT	NAME	R/W	RESET	Sync. Reset
15	RST	RW	0	0: Not-Reset 1: Reset : Automatically cleared
14	SRAC	RW	1	Serial Mode Enable 0 for Parallel Mode, 1 for Serial Mode
13	INTEN	RW	0	Interrupt Request Output Enable 0 for Disabled, 1 for Enabled
				No CRC
12	NCRC	RW	0	0: CRC Output Enabled 1: CRC Output Disabled
11	ICLR	RW	0	Interrupt Clear Register 0 for Not-Clear, 1 for Interrupt Clear : Automatically Cleared
10	MSIEN	RW	1	Device Interrupt Request 0 for Disabled, 1 for Enabled
9	FCLR	RW	0	FIFO Clear 0 for Not Clear, 1 for FIFO Clear : Automatically Cleared
8	FDIR	RW	0	FIFO Direction 0 for FIFO Receive Mode, 1 for Transmit Mode
7	DAM	RW	0	DMA Access Mode (Should be '0') 0 for Dual Address Mode, 1 for Single Address Mode
				DMA Request Mode (Recommended to '0')
6	DRM	RW	1	0 for Level Mode, 1 for Edge Mode * In Edge Mode, the transfer size from DMA should be 1 word.
				Interrupt Mode During DMA Request
5	DRSL	RW	0	0: Do not generate interrupt when a data request occurs. → In DMA mode. 1: Generate interrupt when a data request occurs → In software mode.
				Rising Edge Input
4	REI	RW	1	0: Input data sampled by falling edge 1: Input data sampled by rising edge
				Rising Edge Output
3	REO	RW	0	0: Output data generated at falling edge 1: Output data generated at rising edge
				Busy Count
2-0	BSY	RW	5	initial value 5

PORTCFG Register

0xF0051080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCKIS	BIG	BS													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PORT

BIT	NAME	R/W	RESET	Sync. Reset
31	SCKIS	RW	0	Memory Stick Clock Selection 0: Internally Generated Clock 1: loop-back clock through PAD
30	BIG	RW	1	Big Endian Data Format 0: Little Endian Format 1: Big Endian Format
29-28	BS	RW	0	Bus Size Selection 0: 16bits interface 1: 32bits interface 2~3: unpredicted
27-4			0	Read as 0
3-0	PORT	RW	0	It specifies the port number to be used by the memory stick host controller. (0~5)

Refer to "33.2.1 Port Configuration" on page 33-4.

12.3 Operation & Timing Diagram

12.3.1 Operation During Reset

An internal reset (initialization of the internal registers and operating sequence) is performed by setting software reset in CKC or RST in this controller. However, when the host controller is reset during communication with the Memory Stick, the resulting bus state may differ from the Memory Stick. Therefore, when reset is performed during communication, power-on-reset sequence needs to be applied to the Memory Stick.

12.3.2 Communication with the Memory Stick

An example of communication with the Memory Stick is shown below – Figure 12.2

This example shows the case when TPC SET_CMD transmit is executed.

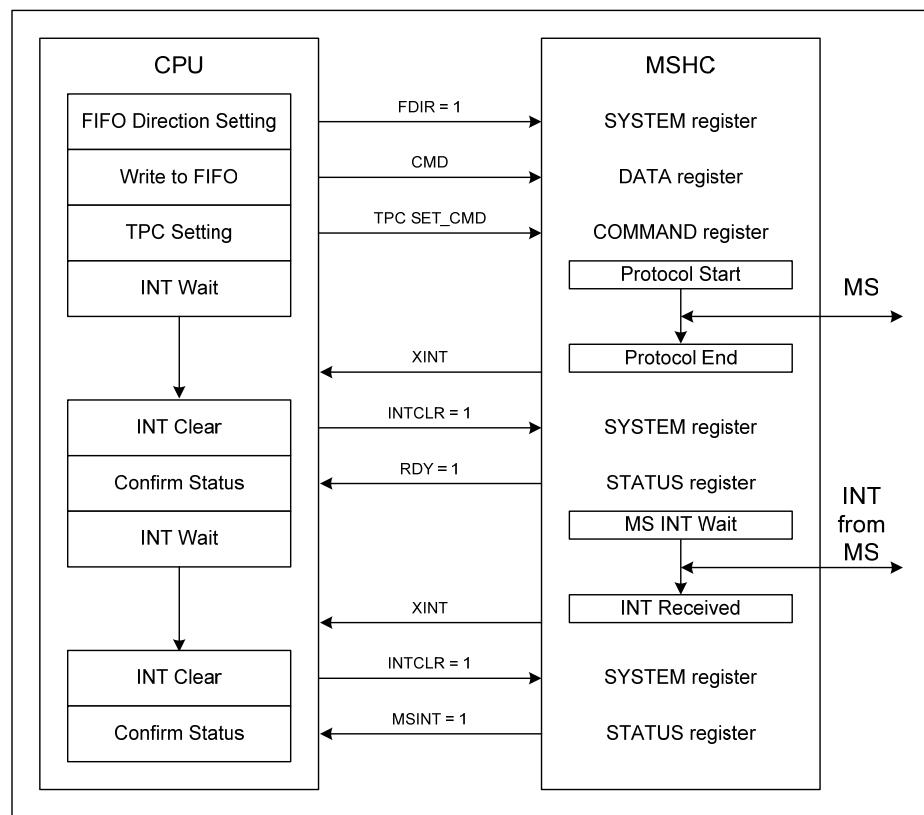


Figure 12.2 Communication Example

- (1) Set system register bit FDIR = 1 to set the FIFO direction to CPU → MS.
- (2) Write the command data to the FIFO.
- (3) Write the TPC and the data transfer size to the command register and start the protocol.
- (4) The CPU waits for the protocol end interrupt.
- (5) After the protocol ends, an interrupt is output from the host controller.
- (6) The CPU receives this interrupt, then sets system register bit INTCLR = 1 to clear the interrupt.
- (7) Read the status register to confirm that communication with the Memory Stick ended normally.
- (8) The CPU waits for the INT interrupt from the Memory Stick. After INT generation from the Memory Stick, an interrupt is output from the host controller. The CPU receives this

interrupt, then sets system register bit INTCLR = 1 to clear the interrupt.
 (9) Read the status register to confirm INT generation from the Memory Stick.

12.3.3 Parallel Interface Mode Setting Procedure

The Figure 12.3 shows the parallel interface mode setting procedure

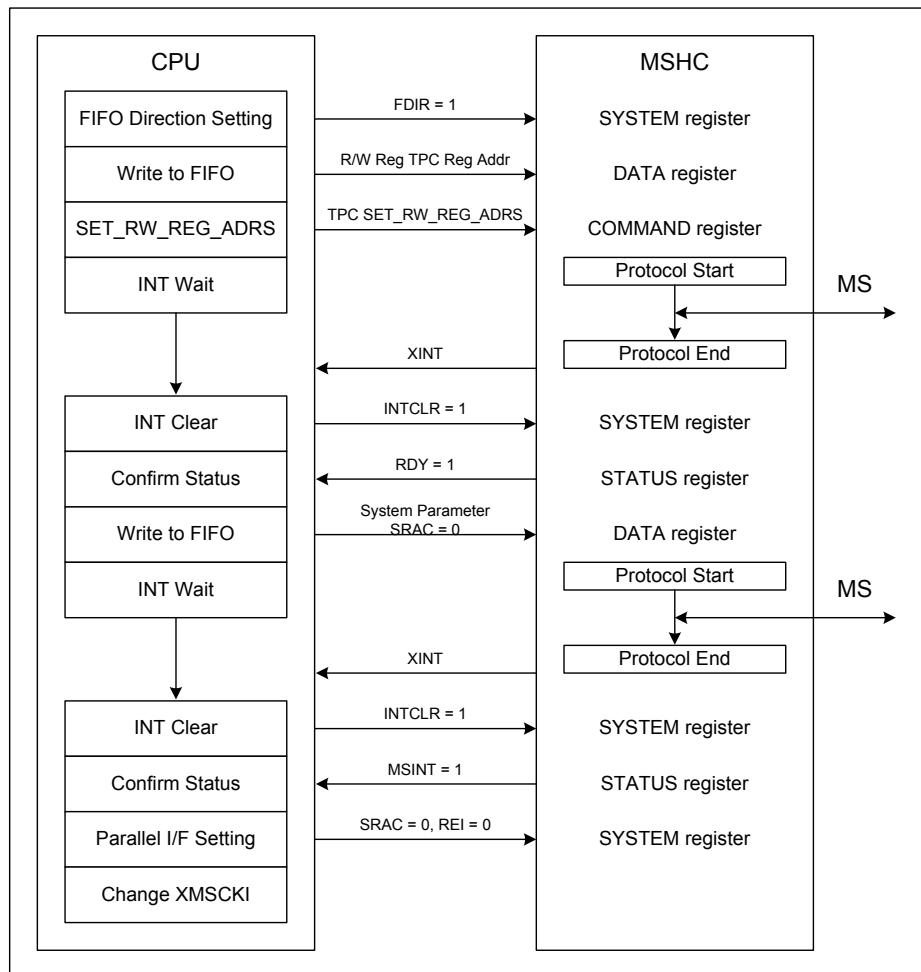


Figure 12.3 Parallel Mode Setting Procedure

- (1) Identify the memory stick media and confirm that is a Memory Stick PRO.
- (2) Set the Memory Stick to parallel interface mode.
- (3) Write SRAC = 0 and REI = 0 to the system register of the host controller and set the host controller to parallel interface mode.

When using a high-speed SCLK, change XMSCKI using an external clock generator, etc. while communication is not being performed with the Memory Stick.

Note that communication with the Memory Stick is not possible if the host controller is set to parallel interface mode first.

12.3.4 Interrupt Request & Clear

When system register bit INTEN = 1, interrupt request output is enabled. When system register bit INTEN = 0, XIN = 1 is held.

- (1) End of Communication with the Memory Stick. An interrupt request is generated when status register bit RDY changes from '0' to '1'.
- (2) Data Transfer Request. An interrupt request is generated when status register bit DRQ changes from '0' to '1' while system register bit DRSL = 1.
- (3) INT Received from the Memory Stick. An interrupt request is generated when status register bit MSINT changes from '0' to '1' while system register bit MSIEN = 1.
- (4) Clear Interrupt Status. Interrupt requests can be cleared by setting system register bit INTCLR = 1. However, when an interrupt request is generated by a data transfer request(DRQ) while system register bit DRSL is set to '1', the interrupt request can also be cleared by writing or reading the FIFO

When using a high-speed SCLK, change XMSCKI using an external clock generator, etc. while communication is not being performed with the Memory Stick.

12.3.5 Serial Interface Read Protocol

The Figure 12.4 shows the timing diagram for serial interface read protocol.

XSCKO output stops high during the BS0 period when communication is not performed with the Memory Stick. Communication with the Memory Stick is started by write to the command register. (Timing 1)

TPC is transmitted during the BS1 period. The host controller waits for RDY signal from the Memory Stick during the BS2 period. Data is received from the Memory Stick during the BS3 period. However, when the data cannot be transferred to the CPU in time, XSCKO output stops high and data receive waits until the FIFO is not full. Also, the BS3 period is one cycle longer than the length by the Specifications, but this is not a problem for the communication protocol. High is input for the extended portion data.

After communication with the Memory Stick ends, the CPU is notified of communication end. (Timing 7) In order to secure the margin of a bus change, at the timing to change the data transmission from the Host Controller, XSCKO is fixed to High and SDIO is made into Hi-Z for 1 * SCLK cycle. (Timing 2)

At the timing to change the output from the Memory Stick, the BS1 period is extended for 1 * SCLK cycle (timing 3).

When system register bit INTEN = 1, interrupt request output is enabled. When system register bit INTEN = 0, XIN = 1 is held.

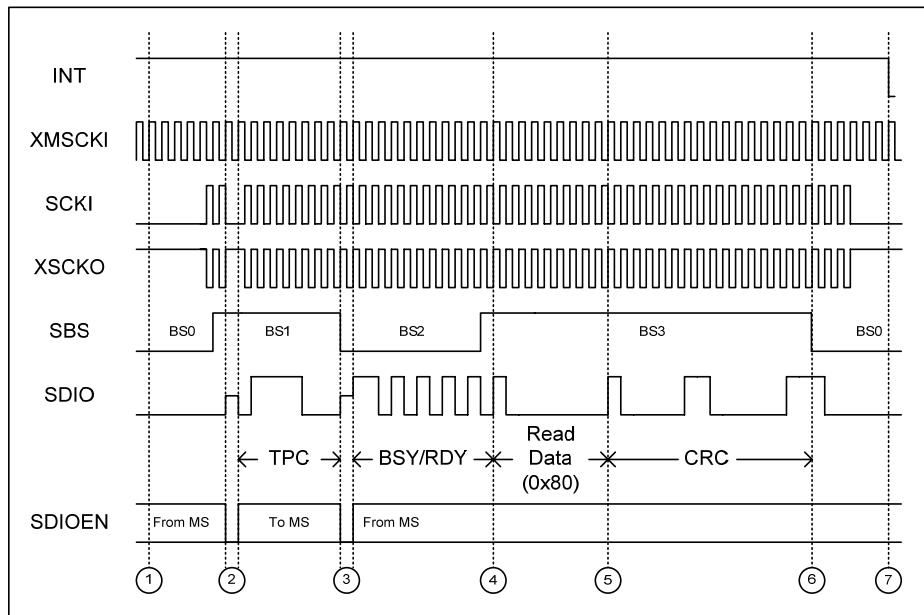


Figure 12.4 The Timing Diagram of Serial Interface Read Protocol

12.3.6 Serial Interface Write Protocol

The Figure 12.5 shows the timing diagram of serial interface write protocol.

XSCKO output stops high during the BS0 period when communication is not performed with the Memory Stick. Communication with the Memory Stick is started by write to the command register. (Timing 1)

TPC is transmitted during the BS1 period. Data is transmitted to the Memory Stick during the BS2 period. However, when the data cannot be transferred from the CPU in time, XSCKO output stops high and data transmit waits until data is written in the FIFO. CRC is added to the end of the data. The host controller waits for RDY signal from the Memory Stick during the BS3 period.

After communication with the Memory Stick ends, the CPU is notified of communication end. (Timing 7) In order to secure the margin of a bus change, at the timing to change the data transmission from the Host Controller, SCKO is fixed to High and SDIO is made into Hi-Z during 1 * SCLK cycle. (timing 2)

At the timing to change the output from the Memory Stick, the BS2 period is extended for 1 * SCLK cycle (Timing 5). And, BS1 period as well as a Read Protocol is extended for 1 * SCLK cycle.

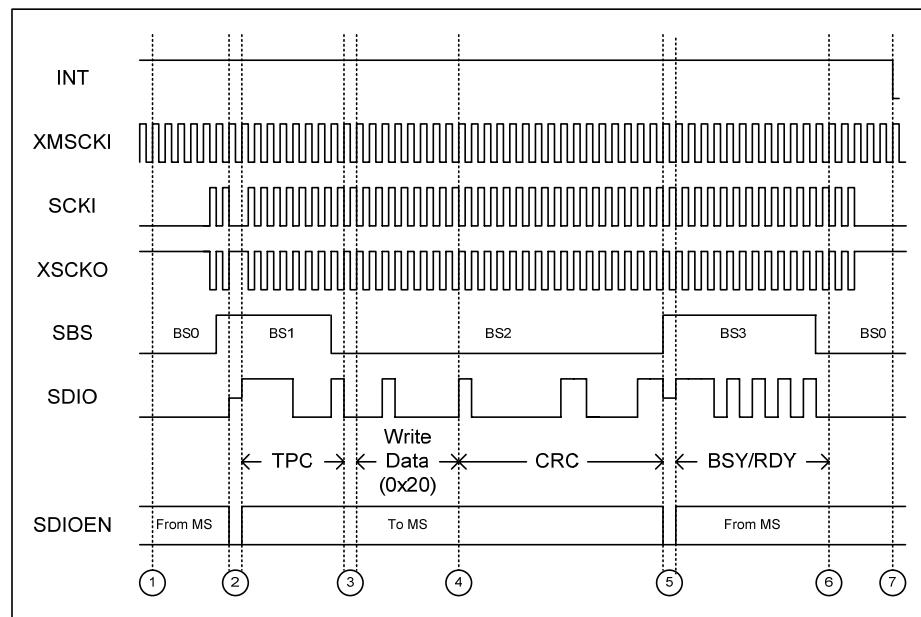


Figure 12.5 The Timing Diagram of Serial Interface Write Protocol

12.3.7 Serial Interface INT Generation

The Figure 12.6 shows the timing diagram of serial interface INT generation.

When an INT is generated from the Memory Stick during the BS0 period (Timing 5), the host controller notifies the CPU of INT generation from the Memory Stick. (Timing 6)

When an INT is generated from the Memory Stick during generation of the communication end interrupt, the communication end interrupt is first cleared, then the host controller notifies the CPU of INT generation from the Memory Stick. In serial interface mode, status register bit MSINT = 1.

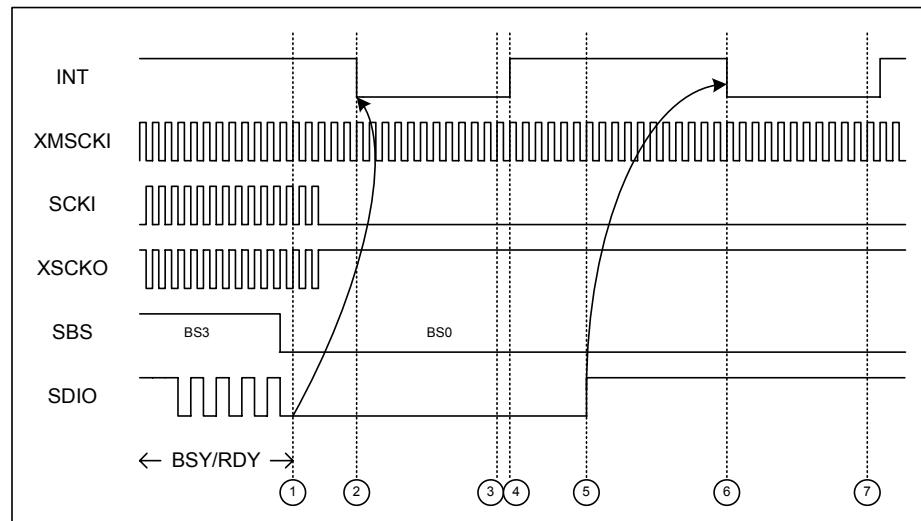


Figure 12.6 The Timing Diagram of Serial Interface INT Generation

12.3.8 Parallel Interface Read Protocol

The Figure 12.7 shows the timing diagram of parallel interface read protocol.

XMSCKI output stops high during the BS0 period when communication is not performed with the Memory Stick. Communication with the Memory Stick is started by write to the command register. (Timing 1)

TPC is transmitted during the BS1 period. The host controller waits for RDY signal from the Memory Stick during the BS2 period. Data is received from the Memory Stick during the BS3 period. However, when the data cannot be transferred to the CPU in time, XSCKO output stops high and data receive waits until the FIFO is not full.

After communication with the Memory Stick ends, the CPU is notified of communication end. (Timing 7)

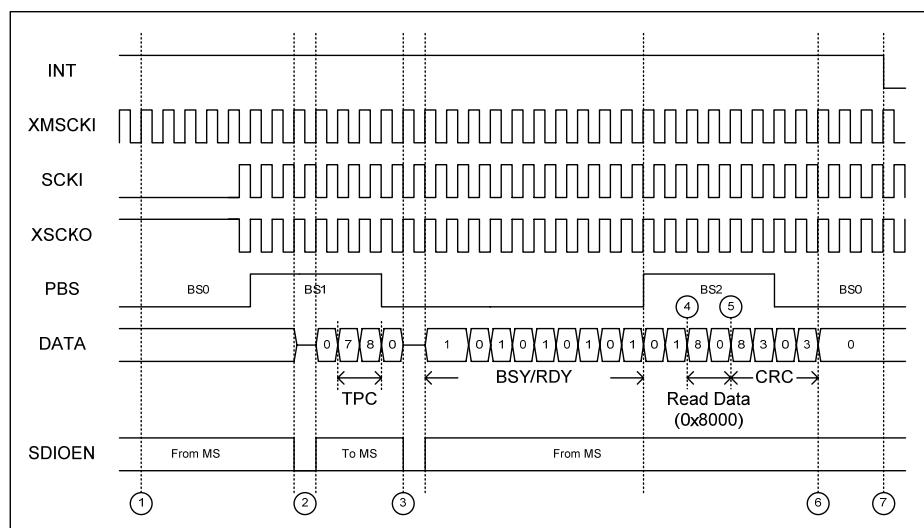


Figure 12.7 The Timing Diagram of Parallel Interface Read Protocol

12.3.9 Parallel Interface Write Protocol

The Figure 12.8 shows the timing diagram of parallel interface write protocol.

XMSCKI output stops high during the BS0 period when communication is not performed with the Memory Stick. Communication with the Memory Stick is started by write to the command register. (Timing 1)

TPC is transmitted during the BS1 period. Data is transmitted to the Memory Stick during the BS2 period. However, when the data cannot be transferred from the CPU in time, XSCKO output stops high and data transmit waits until data is written in the FIFO. CRC is added to the end of the data. The host controller waits for RDY signal from the Memory Stick during the BS3 period.

After communication with the Memory Stick ends, the CPU is notified of communication end. (Timing 7)

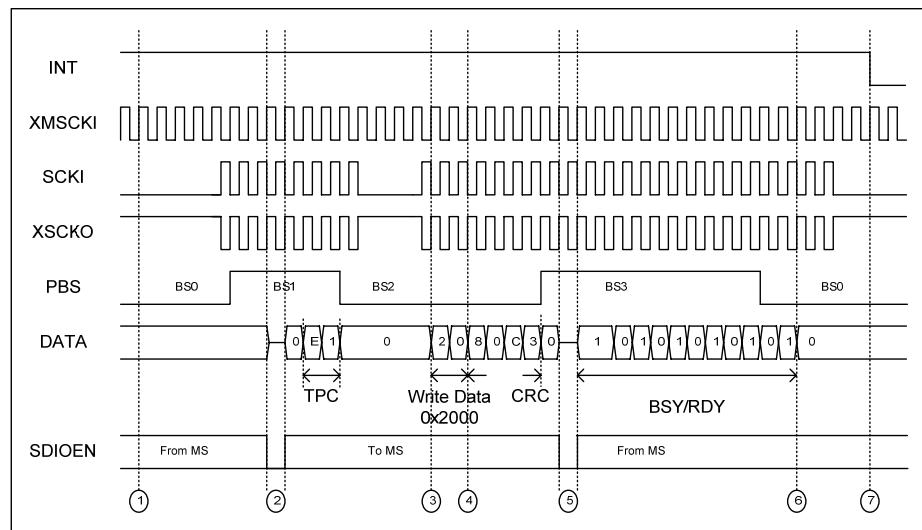


Figure 12.8 The Timing Diagram of Parallel Interface Write Protocol

12.3.10 Parallel Interface INT Generation

The Figure 12.9 shows the timing diagram of parallel interface INT generation.

When an INT is generated from the Memory Stick during the BS0 period (timing 5), the host controller notifies the CPU of INT generation from the Memory Stick. (Timing 6). When an INT is generated from the Memory Stick during generation of the communication end interrupt, the communication end interrupt is first cleared, then the host controller notifies the CPU of INT generation from the Memory Stick.

In parallel interface mode, status register bit MSINT = 1, and the INT value is reflected to status register bits CED, ERR, BRQ, and CNK.

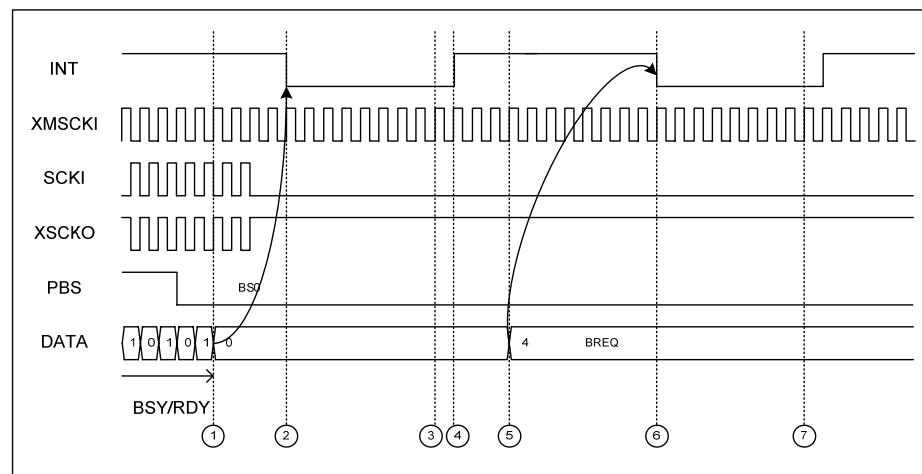


Figure 12.9 The Timing Diagram of Parallel Interface INT Generation

13 SD/SDIO/MMC/CE-ATA HOST CONTROLLER

13.1 Overview

The TCC79XX complies with following versions of specification concerned.

- SD Host Controller Specification Version 2.0
- SDIO Card Specification Version 1.2
- MMC Specification Version 4.2, MMC Plus and MMC Mobile
- CE-ATA Digital Protocol Revision 1.1

The TCC79XX has two slots and separate configuration registers. So those can be assigned to separate host controllers at the same time. For more effective data transaction, two 1K dual port FIFO are used. You may access those using inherent SDMA and ADMA. No DMA access is also available.

Refer to the SD Host Controller Standard Specification V2.00 of SD Association to know more specific explanation.

13.2 Block Diagram

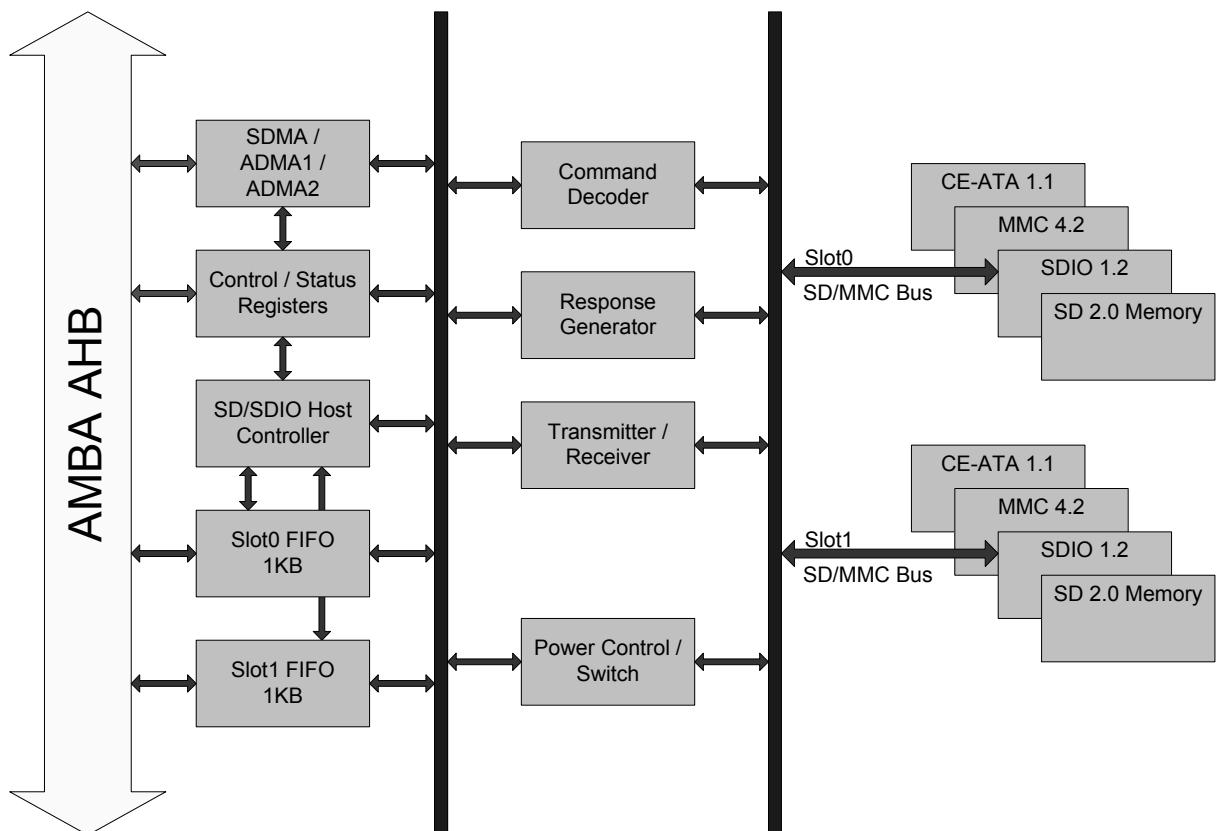


Figure 13.1 SD/SDIO/MMC/CE-ATA Block Diagram

13.3 Register Description

Table 13.1 Slot0/1 Register Map (Base Address = 0xF0090000/0xF0090100)

Name	Address	Type	Reset	Description
SDMA	0x000	R/W	0x0000	SDMA System Address
BSIZE	0x004	R/W	0x0000	Block Size
BCNT	0x006	R/W	0x0000	Block Count
ARG	0x008	R/W	0x0000	Argument
TMODE	0x00C	R/W	0x0000	Transfer Mode
CMD	0x00E	R/W	0x0000	Command
RESP0	0x010	R	0x0000	Response0
RESP1	0x012	R	0x0000	Response1
RESP2	0x014	R	0x0000	Response2
RESP3	0x016	R	0x0000	Response3
RESP4	0x018	R	0x0000	Response4
RESP5	0x01A	R	0x0000	Response5
RESP6	0x01C	R	0x0000	Response6
RESP7	0x01E	R	0x0000	Response7
DATAL	0x020	R/W	-	Buffer Data Port(Low)
DATAH	0x022	R/W	-	Buffer Data Port(High)
STATEL	0x024	R	0x0000	Present State(Low)
STATEH	0x026	R	0x0000	Present State(High)
CONTL	0x028	R/W	0x0000	Power Control / Host Control
CONTH	0x02A	R/W	0x0000	Wakeup Control / Block Gap Control
CLK	0x02C	R/W	0x0000	Clock Control
TIME	0x02E	R/W	0x0000	Software Reset / Timeout Control
STSL	0x030	R	0x0000	Normal Interrupt Status(Low)
STSH	0x032	R	0x0000	Normal Interrupt Status(High)
STSENL	0x034	R/W	0x0000	Normal Interrupt Status Enable(Low)
STSENH	0x036	R/W	0x0000	Normal Interrupt Status Enable(High)
INTENL	0x038	R/W	0x0000	Normal Interrupt Signal Enable(Low)
INTENH	0x03A	R/W	0x0000	Normal Interrupt Signal Enable(High)
CMD12ERR	0x03C	R	0x0000	Auto CMD12 Error Status
CAPL	0x040	R	0x30B0	Capabilities(Low)
CAPH	0x042	R	0x69EF	Capabilities(High)
CURL	0x048	R	0x0001	Maximum Current Capabilities(Low)
CURH	0x04A	R	0x0000	Maximum Current Capabilities(High)
FORCEL	0x050	W	0x0000	Force event for AutoCmd12 Error
FORCEH	0x052	W	0x0000	Force event for Error Interrupt Status
ADMAERR	0x054	R/W	0x0000	ADMA Error Status
ADDR0	0x058	R/W	0x0000	ADMA Address[15:0]
ADDR1	0x05A	R/W	0x0000	ADMA Address[31:16]
ADDR2	0x05C	R/W	0x0000	ADMA Address[47:32]
ADDR3	0x05E	R/W	0x0000	ADMA Address[63:48]
SLOT	0x0FC	R	0x0000	Slot Interrupt Status
VERSION	0x0FE	R	0x0002	Host Controller Version

The address map of registers is arranged by the half word(16bits) to specify those functions. But The TCC79XX accesses data by the word(32bits). Those are only for slot0. If you access slot1 registers, 0x100 have to be added to address of slot0 registers.

Table 13.2 Channel Control Register Map (Base Address = 0xF0090800)

Name	Address	Type	Reset	Description
SDPORTCTRL	0x00	R/W	0x0000	SD/MMC port control register

The TCC79XX has 8 external ports for SD/MMC interface. Each slot can be connected to the one of these ports.

SDPORTCTRL**0xF0090800**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CD1	CD0		WP1	WP0											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOT1															

BIT	NAME	R/W	RESET	DESCRIPTION
31	CD1	R/W	0	Card Detection for SLOT 1 This bit is connected to card detection signal of SLOT1
30	CD0	R/W	0	Card Detection for SLOT 0 This bit is connected to card detection signal of SLOT0
28	WP1	R/W	0	Write Protect for SLOT 1 This bit is connected to write protect signal of SLOT1
27	WP0	R/W	0	Write Protect for SLOT 0 This bit is connected to write protect signal of SLOT0
7-4	SLOT1	R/W	0	These bits specify what ports are used for SLOT1. 0-7: Port Number
3-0	SLOT0	R/W	0	These bits specify what ports are used for SLOT0. 0-7: Port Number
Others	-	R	0	Read as 0

SDMA System Address

0xF0090n¹¹00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDMA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDMA[15:0]															

SDMA [31:0]

SDMA System Address

System memory address for a DMA transfer

N

This contains system memory address for a SDMA transfer. This should be initialized before starting a SDMA transaction. After SDMA has stopped, the next system address of the next contiguous data position can be read from this register. The SDMA transfer waits at the every boundary specified by the Host SDMA Buffer Boundary in the Block Size register.

¹¹ n = slot number 0 or 1.

Block Count and Size0xF0090n¹²04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCNT[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSIZE[12]	SDMABUF[2:0]														BSIZE[11:0]

BCNT [31:16] Block Count for Current Transfer

This is enabled when Block Count Enable in the Transfer Mode register is set to 1 and is valid only for multiple block transfers.

N	0000h - Stop Count
	0001h - 1 block
	0002h - 2 blocks

	FFFFh - 65535 blocks

BSIZE [15], [11:0] Transfer Block Size

The block size for block data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53. Values ranging from 1 up to the maximum buffer size can be set

N	0000h - No Data Transfer
	0001h - 1 Byte
	0002h - 2 Bytes
	0003h - 3 Bytes
	0004h - 4 Bytes

	01FFh - 511 Bytes
	0200h - 512 Bytes

	0800h - 2048 Bytes
	1000h - 4096 Bytes

SDMABUF [14:12] Host SDMA Buffer Size

These bits specify the size of contiguous buffer in the system memory.

The SDMA transfer shall wait at the every boundary specified by these fields and the Host Controller generates the DMA Interrupt to request the Host Driver to update the SDMA System Address register. At the end of transfer, the Host Controller may issue or may not issue DMA Interrupt. In particular, DMA Interrupt shall not be issued after Transfer Complete Interrupt is issued.

N	000b - 4KB(Detects A11 Carry out)
	001b - 8KB(Detects A12 Carry out)
	010b - 16KB(Detects A13 Carry out)
	011b - 32KB(Detects A14 Carry out)
	100b - 64KB(Detects A15 Carry out)
	101b - 128KB(Detects A16 Carry out)
	110b - 256KB(Detects A17 Carry out)
	111b - 512KB(Detects A18 Carry out)

¹² n = slot number 0 or 1.

Argument

0xF0090n¹³08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARG[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARG[15:0]															

ARG	[31:0]	Command Argument
N	This is specified as bit39-8 of Command-Format. See the Table 13.3.	

Table 13.3 Command Format

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'1'	x	x	x	'1'
Description	start bit	transmission bit	response index	card status	CRC7	end bit

Command and Transfer Mode

0xF0090n 0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMDINDEX															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTYPE															

BIT	NAME	R/W	RESET	DESCRIPTION
Command Index				
29-24	CMDINDEX	R/W	0	This bit shall be set to the command number (CMD0-63, ACMD0-63).
Command Type				
There are three types of special commands.				
- Abort(11b) CMD12, CMD52 for writing 'I/O Abort' in CCCR				
23-22	CTYPE	R/W	0	- Resume(10b) CMD52 for writing "Function Select" in CCCR
- Suspend(00b) CMD52 for writing "Bus Suspend" in CCCR				
- Normal(00b) all other commands				
Data Present Select				
21	DATSEL	R/W	0	1: This bit is set to 1 to indicate that data is present and shall be transferred using the DAT line. 0: No Data Present
Command Index Check Enable				
20	CICHK	R/W	0	1: If this bit is set to 1, the HC shall check the index field in the response to see if it has the same value as the command index.

¹³ n = slot number 0 or 1.

				0: Disable
Command CRC Check Enable				
19	CRCHK	R/W	0	1: If this bit is set to 1, the HC shall check the CRC field in the response. 0: Disable
Response Type				
17:16	RTYPE	R/W	0	Response Type Select 00 - No Response 01 - Response length 136 (R2) 10 - Response length 48 (R3, R4, R1, R5, R7) 11 - Response length 48 check Busy after response (R1b, R5b)
SPI Mode				
7	SPI	R/W	0	1: SPI Mode 0: SD Mode
CMD Completion Enable for CE-ATA				
6	ATACMD	R/W	0	Device will send command completion signal for CE-ATA Device will not send command completion signal for CE-ATA
Multi/Single Block Select				
5	MS	R/W	0	This bit is set when issuing multiple-block transfer commands using DAT line. For any other commands, this bit shall be set to 0. 1: Multiple Block 0: Single Block
Data Transfer Direction Select				
4	DIR	R/W	0	1: Read (Card to Host) 0: Write (Host to Card)
Auto CMD12 Enable				
2	m	R/W	0	1: When this bit is set to 1, the HC shall issue CMD12 automatically when last block transfer is completed. 0: Disable
Block Count Enable				
1	BCNTEN	R/W	0	1: enable the Block count register, which is only relevant for multiple block transfers. 0: Disable
DMA Enable				
0	DMAEN	R/W	0	DMA can be enabled only if DMA Support bit in the Capabilities register is set.

1: A DMA operation shall begin when the HD writes to the upper byte of Command register (0x00F).
0: No data transfer or Non-DMA data transfer

Response1/0

0xF0090n¹⁴10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESPONSE1[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESPONSE0[15:0]															

Response3/2

0xF0090n 14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESPONSE3[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESPONSE2[15:0]															

Response5/4

0xF0090n 18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESPONSE5[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESPONSE4[15:0]															

Response7/6

0xF0090n 1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESPONSE7[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESPONSE6[15:0]															

RESPONSE [127:0]	Command Response
N	command responses from the SD Bus

Table 13.4 Response Register

Kind of Response	Meaning of Response	Response Field	Response Register
R1, R1b(normal response)	Card Status	R[39:8]	RESPONSE[31:0]
R1b(Auto CMD12 response)	Card Status for Auto CMD12	R[39:8]	RESPONSE[127:96]
R2(CID, CSD register)	CID or CSD reg. incl.	R[127:8]	RESPONSE[119:0]
R3(OCR register)	OCR register for memory	R[39:8]	RESPONSE[31:0]
R4(OCR register)	OCR register for I/O etc	R[39:8]	RESPONSE[31:0]
R5, R5b	SDIO response	R[39:8]	RESPONSE[31:0]
R6(Published RCA response)	New published RCA[31:16] etc	R[39:8]	RESPONSE[31:0]

¹⁴ n = slot number 0 or 1.

Buffer Data Port

0xF0090n¹⁵20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

DATA [31:0]

Buffer Data

1 The Host Controller Buffer can be accessed through this 32-bit Data Port Register.

Present State

0xF0090n 24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			DAT[7:4]			CMD	DAT[3:0]			SDWP	SDCD	CDST	CDIN		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			RDEN	WREN	RDACT	WRACT	Reserved					DATACT	NODAT	NOCMD	

DAT [28:25], [23:20]

Data Line Signal Level

This status is used to check DAT[7:0] line level to recover from errors, and for debugging.

CMD [24]

CMD Line Signal Level

This status is used to check CMD line level to recover from errors, and for debugging.

SDWP [19]

Write Protect Switch Pin Level

- 1 This bit reflects the SDWP# pin. Write enabled
0 Write protected

SDCD [18]

Card Detect Pin Level

- 1 This bit reflects the inverse value of the SDCD# pin. Card present.
0 No Card present.

CDST [17]

Card State Stable

- 1 If this bit is set to 1, it means the Card Detect Pin Level is stable. No Card or Inserted.
0 Reset of Debouncing.

CDIN [16]

Card Inserted

- 1 This bit indicates whether a card has been inserted.
0 Reset or Debouncing or No Card.

RDEN [11]

Buffer Read Enable

- 1 This status is used for non-DMA read transfers. If this bit is 1, readable data exists in the buffer.
0 Read Disable

¹⁵ n = slot number 0 or 1.

WREN	[10]	Buffer Write Enable
1		This status is used for non-DMA read transfers. If this bit is 1, data can be written to the buffer.
0		Write Disable
RDACT	[9]	Read Transfer Active
1		This status is used for detecting completion of a read transfer.
0		No valid data
WRACT	[8]	Write Transfer Active
1		This status indicates a write transfer is active.
0		No valid data
DATACT	[2]	DAT Line Active
1		This bit indicates whether one of the DAT line on SD bus is in use.
0		DAT line inactive.
NODAT	[1]	Command Inhibit(DAT)
1		This status bit is generated if either the DAT Line Active or the Read transfer Active is set to 1. It cannot issue command which uses the DAT line
0		It can issue command which uses the DAT line
NOCMD	[0]	Command Inhibit(CMD)
1		indicates the CMD line is in use.
0		indicates the CMD line is not in use and the HC can issue a SD command using the CMD line.

Wakeup/Block Gap/Power/Host Control

0xF0090n¹⁶28

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					WKOUT	WGIN	WKINT					BGINT	RDWAI	CON-	BGSTO
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					VOLTSEL		POW	DET-SEL	DETCD	SD8	SELDMA		HS	SD4	LED

WKOUT [26]

Wakeup Enable On Card Removal

1	This bit enables wakeup event via Card Removal assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit.
0	Disable

WGIN [25]

Wakeup Enable On Card Insertion

1	This bit enables wakeup event via Card Insertion assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit.
0	Disable

WKINT [24]

Wakeup Enable On Card Interrupt

1	This bit enables wakeup event via Card Interrupt assertion in the Normal Interrupt Status register. This bit can be set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1
0	Disable

BGINT [19]

Interrupt At Block Gap

1	This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer.
0	Disable

RDWAIT [18]

Read Wait Control

1	If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using DAT[2] line.
0	Disable

CONREQ [17]

Continue Request

1	This bit is used to restart a transaction which was stopped using the Stop At Block Gap Request.
0	Stop is ignored

BGSTOP [16]

Stop At Block Gap Request

1	This bit is used to stop executing a transaction at the next block gap for non-DMA,SDMA and ADMA transfers. The Host Driver shall leave this bit set to 1 until the Transfer Complete is set to 1.
0	Keep transferring

¹⁶ n = slot number 0 or 1.

VOLTSEL [11:9]		SD Bus Voltage Select
By setting these bits, the HD selects the voltage level for the SD card. Only 3.3V is supported. <i>Others will be supported in the future.</i>		
N		111b - 3.3 V(TYP.) 110b - 3.0 V(TYP.) 101b - 1.8 V(TYP.) 100b - 000b – Reserved
POW [8]		SD Bus Power
1		Before setting this bit, the SD host driver shall set SD Bus Voltage Select. <i>This will be supported in the future.</i>
DETSEL [7]		Card Detect Signal Selection
1		The card detect test level is selected for test purpose.
0		SDCD# is selected for normal use
DETCD [6]		Card Detect Test Level
1		This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted,
0		No Card
SD8 [5]		SD 8bit Mode
1		8 bit mode is selected
0		8 bit mode is not selected
SELDMA [4:3]		DMA Select
One of supported DMA modes can be selected. Use of selected DMA is determined by DMA Enable of the Transfer Mode register.		
N		00 - SDMA is selected 01 - 32-bit Address ADMA1 is selected 10 - 32-bit Address ADMA2 is selected 11 - 64-bit Address ADMA2 is selected
HS [2]		High Speed Enable
1		If this bit is set to 1, the HC outputs CMD line and DAT lines at the rising edge of the SD clock (up to 50 MHz)
0		The Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock (up to 25MHz).
SD4 [1]		Data Transfer Width
1		4 bit mode
0		1 bit mode
LED [0]		LED Control
-		This bit is used to caution the user not to remove the card while the SD card is being accessed. <i>This will be supported in the future.</i>

Software Reset/Timeout/Clock Control 0xF0090n¹⁷2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					RSTDAT T	RSTCM D	RSTAL L	Reserved					TIMEOUT[3:0]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDCLKSEL[7:0]								Reserved					SCKEN	CLKRD Y	CLKEN

RSTDAT [26] Software Reset for DAT Line

Only part of data circuit is reset. The following registers and bits are cleared by this bit:

Buffer Data Port Register

- Buffer is cleared and Initialized.

Present State register

- Buffer read Enable
- Buffer write Enable
- Read Transfer Active
- Write Transfer Active
- 1 - DAT Line Active
- Command Inhibit (DAT)

Block Gap Control register

- Continue Request
- Stop At Block Gap Request

Normal Interrupt Status register

- Buffer Read Ready
- Buffer Write Ready
- Block Gap Event
- Transfer Complete

0 Keep working

RSTCMD [25] Software Reset For CMD Line

Only part of command circuit is reset. The following registers and bits are cleared by this bit:

Present State register

- 1 - Command Inhibit (CMD)

Normal Interrupt Status register

- Command Complete

0 Keep working

RSTALL [24] Software Reset For All

1 This reset affects the entire HC except for the card detection circuit. If this bit is set to 1, the host driver should issue reset command and reinitialize the SD card.

0 Keep working

¹⁷ n = slot number 0 or 1.

TIMEOUT	[19:16]	Data Timeout Counter Value
This value determines the interval by which DAT line time-outs are detected		
N	1111 - Reserved	
	1110 - TMCLK * 2^27	

	0001 - TMCLK * 2^14	
	0000 - TMCLK * 2^13	
SDCLKSEL	[15:8]	SDCLK Frequency Select
This register is used to select the frequency of the SDCLK pin.		
N	80h - base clock divided by 256	
	40h - base clock divided by 128	
	20h - base clock divided by 64	
	10h - base clock divided by 32	
	08h - base clock divided by 16	
	04h - base clock divided by 8	
	02h - base clock divided by 4	
	01h - base clock divided by 2	
	00h - base clock(10MHz-63MHz)	
According to the Physical Layer Specification, the maximum SD Clock frequency is 25 MHz in normal speed mode and 50MHz in high speed mode, and shall never exceed this limit.		
SCKEN	[2]	SD Clock Enable
1	SD Clock is enabled.	
0	The Host Controller shall stop SDCLK when writing this bit to 0. SDCLK Frequency Select can be changed when this bit is 0.	
CLKRDY	[1]	Internal Clock Stable
1	This bit is set to 1 when SD clock is stable after writing to Internal Clock Enable in this register to 1. The SD Host Driver shall wait to set SD Clock Enable until this bit is set to 1.	
0	Not ready	
CLKEN	[0]	Internal Clock Enable
1	Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, the Host Controller shall set Internal Clock Stable in this register to 1. This bit shall not affect card detection.	
0	Stop	

Normal Interrupt Status

0xF0090n¹⁸30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
				VENDOR[3:0]		Reserved	ADMA	ACMD12	CLIMIT	DATEN D	DATCR C	DAT-TIME	CINDEX	CMDEN D	CMDCR C	CMDTI ME
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ERR				Reserved			CDINT	CDOUT	CDIN	RDRDY	WRRDY	DMA	BLKGA P	TDONE	CDONE	

VENDOR [31:28]		Vendor Specific Error Status
N		Additional status bits can be defined in this register by the vendor.
ADMA [25]		ADMA Error
1		This bit is set when the Host Controller detects errors during ADMA based data transfer. In addition, the Host Controller generates this Interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor.
0		No Error
ACMD12 [24]		Auto CMD12 Error
1		This bit is set when detecting that one of the bits in Auto CMD12 Error Status register has changed from 0 to 1. This bit is set to 1, not only when the errors in Auto CMD12 occur but also when Auto CMD12 is not executed due to the previous command error..
0		No Error
CLIMIT [23]		Current Limit Error
1		Reading 1 means the HC is not supplying power to SD card due to some failure. <i>This will be supported in the future.</i>
0		No Error
DATEND [22]		Data End Bit Error
1		Occurs when detecting 0 at the end bit position of read data which uses the DAT line or the end bit position of the CRC status.
0		No Error
DATCRC [21]		Data CRC Error
1		Occurs when detecting CRC error when transferring read data which uses the DAT line or when detecting the Write CRC Status having a value of other than "010".
0		No Error
DATTIME [20]		Data Timeout Error
1		Occurs when detecting one of following timeout conditions. 1) Busy Timeout for R1b, R5b type. 2) Busy Timeout after Write CRC status 3) Write CRC status Timeout 4) Read Data Timeout
0		No Error

¹⁸ n = slot number 0 or 1.

CINDEX	[19]	Command Index Error
1	Occurs if a Command Index error occurs in the Command Response.	
0	No Error	
CMDEND	[18]	Command End Bit Error
1	Occurs when detecting that the end bit of a command response is 0.	
0	No Error	
CMDCRC	[17]	Command CRC Error
		Command CRC Error is generated in two cases.
1	1) If a response is returned and the Command Time-out Error is set to 0, this bit is set to 1 when detecting a CRC error in the command response 2) The HC detects a CMD line conflict by monitoring the CMD line when a command is issued.	
0	No Error	
CMDTIME	[16]	Command Timeout Error
		This bit is set only if the no response is returned within 64 SDCLK cycles from the end bit of the command. If the Host Controller detects a CMD line conflict, this bit shall be set without waiting for 64 SD clock cycles because the command will be aborted by the Host Controller.
1		
0	No Error	
ERR	[15]	Error Interrupt
1	If any of the bits in the Error Interrupt Status Register are set, then this bit is set. Therefore the HD can test for an error by checking this bit first.	
0	No Error	
CDINT	[8]	Card Interrupt
1	Card interrupt is generated. In 1-bit mode, the Host Controller shall detect the Card Interrupt without SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the Host System. It is necessary to define how to handle this delay.	
0	No card interrupt	
CDOOUT	[7]	Card Removal
1	This status is set if the Card Inserted in the Present State register changes from 1 to 0. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed	
0	Card State Stable or Debouncing	
CDIN	[6]	Card Insertion
1	This status is set if the Card Inserted in the Present State register changes from 0 to 1. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed.	
0	Card State Stable or Debouncing	

RDRDY	[5]	Buffer Read Ready
1		This status is set if the Buffer Read Enable changes from 0 to 1 and it is ready to read buffer.
0		Not ready to read buffer.
WRRDY	[4]	Buffer Write Ready
1		This status is set if the Buffer Write Enable changes from 0 to 1 and it is ready to read buffer.
0		Not ready to read buffer.
DMA	[3]	DMA Interrupt
1		This status is set if the Host Controller detects the Host DMA Buffer boundary during transfer. In case of ADMA, by setting Int field in the descriptor table, Host Controller generates this interrupt. This interrupt shall not be generated after the Transfer Complete.
0		No DMA Interrupt
BLKGAP	[2]	Block Gap Event
1		If the Stop At Block Gap Request in the Block Gap Control register is set, this bit is set when both a read / write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this bit is not set to 1.
		- Read Transaction: This bit is set at the falling edge of the DAT Line Active Status (When the transaction is stopped at SD Bus timing).
		- Write Transaction: This bit is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing).
0		No Block Gap Event

TDONE	[1]	Transfer Complete
		This bit is set when a read / write transaction is completed.
		<p>- Read Transaction:</p> <p>This bit is set at the falling edge of Read Transfer Active Status.</p> <p>This interrupt is generated in two cases. The first is when a data transfer is completed as specified by data length (After the last data has been read to the Host System). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request in the Block Gap Control register(After valid data has been read to the Host System).</p>
1		<p>- Write Transaction:</p> <p>This bit is set at the falling edge of the DAT Line Active Status. This interrupt is generated in two cases. The first is when the last data is written to the SD card as specified by data length and the busy signal released. The second is when data transfers are stopped at the block gap by setting Stop At Block Gap Request in the Block Gap Control register and data transfers completed. (After valid data is written to the SD card and the busy signal released).</p>
		<p>- Write Transaction:</p> <p>This bit is set when busy is de-asserted.</p>
0		Not complete

Table 13.5 Relation between Transfer Complete and Data Timeout Error

Transfer Complete	Data Timeout Error	Meaning of the status
0	0	Interrupted by another factor
0	1	Timeout occur during transfer
1	Don't Care	Command Execution complete

CDONE	[0]	Command Complete
1		This bit is set when get the end bit of the command response (Except Auto CMD12).
0		No Command Complete

Table 13.6 Relation between Command Complete and Command Timeout Error

Command Complete	Command Timeout Error	Meaning of the status
0	0	Interrupted by another factor
Don't Care	1	Response not received within 64 SDCLK cycles
1	0	Response received

Normal Interrupt Status Enable

0xF0090n¹⁹34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		VENDOR				ADMA	ACMD1 2	CLIMIT	DATEN D	DATCR C	DAT- TIME	CINDEX	CMDEN D	CMDCR C	CMDTI ME
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

VENDOR	[31:28]	Vendor Specific Error Status Enable
1	Enabled	
0	Masked	
ADMA	[25]	ADMA Error Status Enable
1	Enabled	
0	Masked	
ACMD12	[24]	Auto CMD12 Error Status Enable
1	Enabled	
0	Masked	
CLIMIT	[23]	Current Limit Error Status Enable
1	Enabled	
0	Masked	
DATEND	[22]	Data End Bit Error Status Enable
1	Enabled	
0	Masked	
DATCRC	[21]	Data CRC Error Status Enable
1	Enabled	
0	Masked	
DATTIME	[20]	Data Timeout Error Status Enable
1	Enabled	
0	Masked	
CINDEX	[19]	Command Index Error Status Enable
1	Enabled	
0	Masked	
CMDEND	[18]	Command End Bit Error Status Enable
1	Enabled. If this bit is set to 0, the Host Controller shall clear interrupt request to the System. The Card Interrupt detection is stopped when this bit is cleared and re-started when this bit is set to 1. The Host Driver should clear the Card Interrupt Status Enable before servicing the Card Interrupt and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.	
0	Masked	

¹⁹ n = slot number 0 or 1.

CMDCRC	[17]	Command CRC Error Status Enable
1	Enabled	
0	Masked	
CMDTIME	[16]	Command Timeout Error Status Enable
1	Enabled	
0	Masked	
CDINT	[8]	Card Interrupt Status Enable
1	Enabled	
0	Masked	
CDOUT	[7]	Card Removal Status Enable
1	Enabled	
0	Masked	
CDIN	[6]	Card Insertion Status Enable
1	Enabled	
0	Masked	
RDRDY	[5]	Buffer Read Ready Status Enable
1	Enabled	
0	Masked	
WRRDY	[4]	Buffer Write Ready Status Enable
1	Enabled	
0	Masked	
DMA	[3]	DMA Interrupt Status Enable
1	Enabled	
0	Masked	
BLKGAP	[2]	Block Gap Event Status Enable
1	Enabled	
0	Masked	
TDONE	[1]	Transfer Complete Status Enable
1	Enabled	
0	Masked	
CDONE	[0]	Command Complete Status Enable
1	Enabled	
0	Masked	

Normal Interrupt Signal Enable

0xF0090n²⁰38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						ADMA	ACMD1 2	CLIMIT	DATEN D	DATCR C	DAT- TIME	CINDEX	CMDEN D	CMDCR C	CMDTI ME
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR							CDINT	CDOUT	CDIN	RDRDY	WRRDY	DMA	BLKGA P	TDONE	CDONE

VENDOR [31:28]		Vendor Specific Error Signal Enable			
1 Enabled					
0 Masked					
ADMA [25]		ADMA Error Signal Enable			
1 Enabled					
0 Masked					
ACMD12 [24]		Auto CMD12 Error Signal Enable			
1 Enabled					
0 Masked					
CLIMIT [23]		Current Limit Error Signal Enable			
1 Enabled					
0 Masked					
DATEND [22]		Data End Bit Error Signal Enable			
1 Enabled					
0 Masked					
DATCRC [21]		Data CRC Error Signal Enable			
1 Enabled					
0 Masked					
DATTIME [20]		Data Timeout Error Signal Enable			
1 Enabled					
0 Masked					
CINDEX [19]		Command Index Error Signal Enable			
1 Enabled					
0 Masked					
CMDEND [18]		Command End Bit Error Signal Enable			
1 Enabled					
0 Masked					

²⁰ n = slot number 0 or 1.

CMDCRC	[17]	Command CRC Error Signal Enable
1	Enabled	
0	Masked	
CMDTIME	[16]	Command Timeout Error Signal Enable
1	Enabled	
0	Masked	
CDINT	[8]	Card Interrupt Signal Enable
1	Enabled	
0	Masked	
CDOUT	[7]	Card Removal Signal Enable
1	Enabled	
0	Masked	
CDIN	[6]	Card Insertion Signal Enable
1	Enabled	
0	Masked	
RDRDY	[5]	Buffer Read Ready Signal Enable
1	Enabled	
0	Masked	
WRRDY	[4]	Buffer Write Ready Signal Enable
1	Enabled	
0	Masked	
DMA	[3]	DMA Interrupt Signal Enable
1	Enabled	
0	Masked	
BLKGAP	[2]	Block Gap Event Signal Enable
1	Enabled	
0	Masked	
TDONE	[1]	Transfer Complete Signal Enable
1	Enabled	
0	Masked	
CDONE	[0]	Command Complete Signal Enable
1	Enabled	
0	Masked	

Auto CMD12 Error Status

0xF0090n²¹3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NOCMD INDEX ENDBIT CRC TIME-OUT NORUN

NOCMD	[7]	Command Not Issued
1	Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error (D04 - D01)	
0	No Error	
INDEX	[4]	Auto CMD12 Index Error
1	This bit is set if the Command Index error occurs in response to a command.	
0	No Error	
ENDBIT	[3]	Auto CMD12 End Bit Error
1	This bit is set when detecting that the end bit of command response is 0.	
0	No Error	
CRC	[2]	Auto CMD12 CRC Error
1	This bit is set when detecting a CRC error in the command response.	
0	No Error	
TIMEOUT	[1]	Auto CMD12 Timeout Error
1	This bit is set if the no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to 1, the other error status bits (D04-D02) are meaningless.	
0	No Error	
NORUN	[0]	Auto CMD12 Not Executed
1	Setting this bit to 1 means the HC cannot issue Auto CMD12 to stop memory multiple block transfer due to some error. If this bit is set to 1, other error status bits (D04-D01) are meaningless.	
0	Executed	

Table 13.7 Relation between Auto CMD12 CRC Error and Auto CMD12 Timeout Error

Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Kinds of Error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

²¹ n = slot number 0 or 1.

Present State

0xF0090n²²40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIBLK	SPI	BUS64	INT	V18	V30	V33	RE-SUME	SDMA	HS		ADMA2	EXT-BUS		MAXBLK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASECLK								TUNIT		TIMEOUTCLK					

SPIBLK [30]		SPI Block Mode
1	Supported	
0	Not supported	
SPI [29]		SPI Mode
1	Supported	
0	Not supported	
BUS64 [28]		64bit System Bus Support
1	Setting 1 to this bit indicates that the Host Controller supports 64-bit address descriptor mode and is connected to 64-bit address system bus.	
0	Does not support 64 bit system address	
INT [27]		Interrupt Mode
1	Supported	
0	Not supported	
V18 [26]		Voltage Support 1.8V
1	Supported	
0	Not supported	
V30 [25]		Voltage Support 3.0V
1	Supported	
0	Not supported	
V33 [24]		Voltage Support 3.3V
1	Supported	
0	Not supported	
RESUME [23]		Suspend / Resume Support
1	Supported	
0	Not supported	
SDMA [22]		SDMA Support
1	Supported	
0	Not supported	
HS [21]		High Speed Support
1	Supported	
0	Not supported	

²² n = slot number 0 or 1.

ADMA2 [19]		ADMA2 Supported
1	Supported	
0	Not supported	
EXTBUS [18]		Extended Media Bus Support
1	Supported	
0	Not supported	
MAXBLK [17:16]		Max Block Length
This value indicates the maximum block size that the Host Driver can read and write to the buffer in the Host Controller. The buffer shall transfer this block size without wait cycles. It is noted that transfer block length shall be always 512 bytes for SD Memory Cards regardless of this field.		
N		
00 - 512 byte		
01 - 1024 byte		
10 - 2048 byte		
11 - 4096 byte		
BASECLK [13:8]		Base Clock Frequency For SD Clock
This value indicates the base (maximum) clock frequency for the SD clock. Unit values are MHz. If the real frequency is 16.5MHz, the larger value shall be set 01 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value		
N		
Not 0 - 1 MHz to 63 MHz		
000000b - Get information via another method.		
TUNIT [7]		Timeout Clock Unit
This bit shows the unit of base clock frequency used to detect Data Timeout Error.		
1		
1	The unit is MHz.	
0	The unit is KHz.	
TIMEOUTCLK [5:0]		Timeout Clock Frequency
This bit shows the base clock frequency used to detect Data Timeout Error. The Timeout Clock Unit defines the unit of this field's value.		
N		
Timeout Clock Unit =0 [KHz] unit: 1KHz to 63KHz		
Timeout Clock Unit =1 [MHz] unit: 1MHz to 63MHz		
Not 0 - 1KHz to 63Khz or 1MHz to 63MHz		
000000b - Get Information via another method		

Maximum Current Capabilities																0xF0090n ²³ 48
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Reserved
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	MAXCURV18[7:0]
MAXCURV30[7:0]								MAXCURV33[7:0]								

MAXCURV18 [23:16]	Maximum Current for 1.8V
Maximum Current for 1.8V	
<i>This will be supported in the future.</i>	
MAXCURV30 [15:8]	Maximum Current for 3.0V
Maximum Current for 3.0V	
<i>This will be supported in the future.</i>	
MAXCURV33 [7:0]	Maximum Current for 3.3V
Maximum Current for 3.3V	
<i>This will be supported in the future.</i>	

²³ n = slot number 0 or 1.

Force Event for Error Interrupt / Auto CMD12 Error Status 0xF0090n²⁴50

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VENDOR[3:0]				Reserved		ADMA	ACMD12	CLIMIT	DATEN D	DATCR C	DAT-TIME	CINDEX	CMDEN D	CMDCR C	CMDTI ME
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								NOCMD	Reserved		INDEX	ENDBIT	CRC	TIME-OUT	NORUN

VENDOR	[31:28]	Force Event For Vendor Specific Error Status
1	Generate interrupt	
0	No interrupt	
ADMA	[25]	Force Event For ADMA Error
1	Generate interrupt	
0	No interrupt	
ACMD12	[24]	Force Event For Auto CMD12 Error
1	Generate interrupt	
0	No interrupt	
CLIMIT	[23]	Force Event For Current Limit Error
1	Generate interrupt	
0	No interrupt	
DATEND	[22]	Force Event For Data End Bit Error
1	Generate interrupt	
0	No interrupt	
DATCRC	[21]	Force Event For Data CRC Error
1	Generate interrupt	
0	No interrupt	
DATTIME	[20]	Force Event For Data Timeout Error
1	Generate interrupt	
0	No interrupt	
CINDEX	[19]	Force Event For Command Index Error
1	Generate interrupt	
0	No interrupt	
CMDEND	[18]	Force Event For Command End Bit Error
1	Generate interrupt	
0	No interrupt	
CMDCRC	[17]	Force Event For Command CRC Error
1	Generate interrupt	
0	No interrupt	

²⁴ n = slot number 0 or 1.

CMDTIME	[16]	Force Event For Command Timeout Error
1	Generate interrupt	
0	No interrupt	
NOCMD	[7]	Force Event For Command Not Issued
1	Generate interrupt	
0	No interrupt	
INDEX	[4]	Force Event For Auto CMD12 Index Error
1	Generate interrupt	
0	No interrupt	
ENDBIT	[3]	Force Event For Auto CMD12 End Bit Error
1	Generate interrupt	
0	No interrupt	
CRC	[2]	Force Event For Auto CMD12 CRC Error
1	Generate interrupt	
0	No interrupt	
TIMEOUT	[1]	Force Event For Auto CMD12 Timeout Error
1	Generate interrupt	
0	No interrupt	
NORUN	[0]	Force Event For Auto CMD12 Not Executed
1	Generate interrupt	
0	No interrupt	

ADMA Error Status

0xF0090*n*²⁵54

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															
												LEN	ERRSTATE		

LEN	[2]	ADMA Length Mismatch Error
1		<p>This error occurs in the following 2 cases.</p> <p>(1) While Block Count Enable being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length.</p> <p>(2) Total data length can not be divided by the block length.</p>
0		No Error
ERRSTATE	[1:0]	ADMA Error State
		<p>This field indicates the state of ADMA when error is occurred during ADMA data transfer. This field never indicates "10" because ADMA never stops in this state.</p>
N		<p>00 - ST_STOP (Stop DMA) Points next of the error descriptor</p> <p>01 - ST_FDS (Fetch Descriptor) Points the error descriptor</p> <p>10 - Never set this state (Not used)</p> <p>11 - ST_TFR (Transfer Data) Points the next of the error descriptor</p>

ADMA System Address0

0xF0090n58

ADMA System Address1

0xF0090n5C

ADDR [63:0]	ADMA System Address
N	<p>This register holds byte address of executing command of the Descriptor table. 32-bit Address Descriptor uses lower 32-bit of this register. At the start of ADMA, the Host Driver shall set start address of the Descriptor table. The ADMA increments this register address, which points to next line, when every fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register shall hold valid Descriptor address depending on the ADMA state. The Host Driver shall program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. ADMA2 ignores lower 2-bit of this register and assumes it to be 00b.</p>

Refer to the SD Host Controller Standard Specification V2.00 of SD Association to know how to make the descriptor table.

²⁵ n = slot number 0 or 1.

SPI Interrupt Support0xF0090n²⁶F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										SPIINT[7:0]					

SPIINT [7:0]**SPI Interrupt Support**

N This bit is set to indicate the assertion of interrupts in the SPI mode at any time, irrespective of the status of the card select (CS) line. If this bit is zero, then SDIO card can only assert the interrupt line in the SPI mode when the CS line is asserted.

Host Controller Version / Slot Interrupt Status

0xF0090n FC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								SLOTINT[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VENDOR[7:0]								SPEC[7:0]							

SLOTINT [23:16]**Interrupt Signal For Each Slot**

These status bit indicate the logical OR of Interrupt signal and Wakeup signal for each slot.

N Bit 00 - Slot 1
 Bit 01 - Slot 2
 Bit 02 - Slot 3

 Bit 07 - Slot 8

VENDOR [15:8]**Vendor Version Number**

N This status is reserved for the vendor version number. The HD should not use this status.

SPEC [7:0]**Specification Version Number**

This Status indicates the Host Controller Spec Version.

N 00 - SD Host Specification version 1.0
 01 - SD Host Specification version 2.00 including only the feature of the Test Register
 02 - SD Host Specification version 2.00 including the feature of the Test Register and ADMA
 others – Reserved

²⁶ n = slot number 0 or 1.

13.4 Timing Diagram

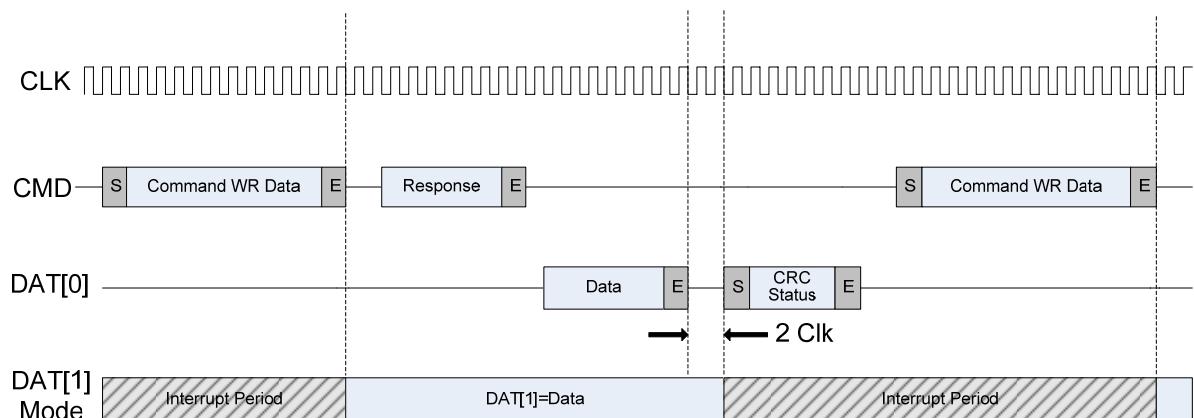


Figure 13.2 SDIO/SD – Write Interrupt Cycle Timing

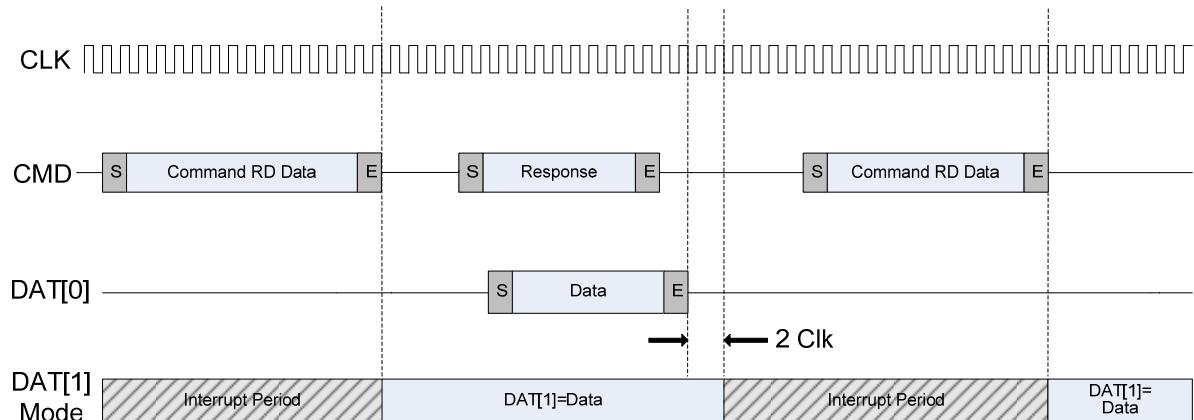


Figure 13.3 SDIO/SD – Read Interrupt Cycle Timing

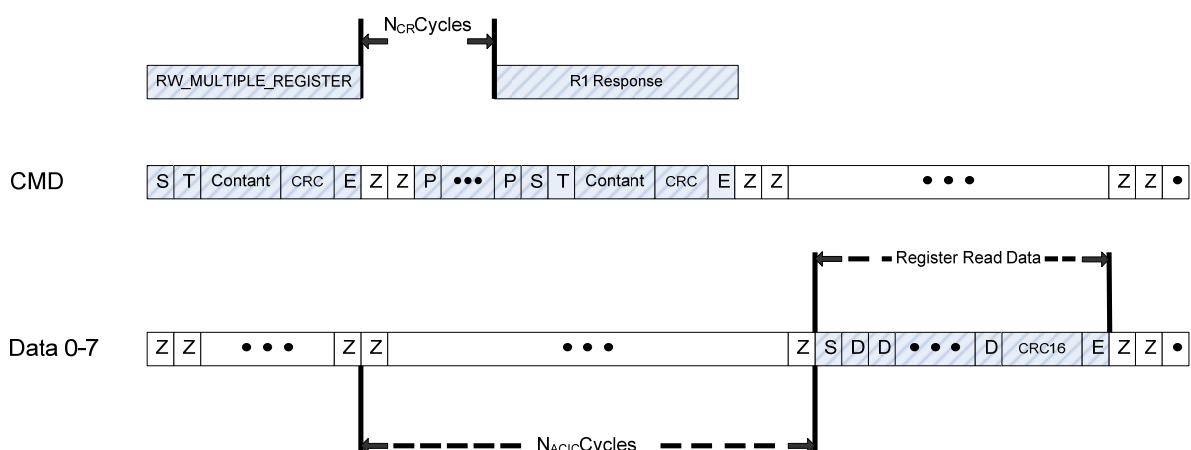


Figure 13.4 CMD60 Read Transaction

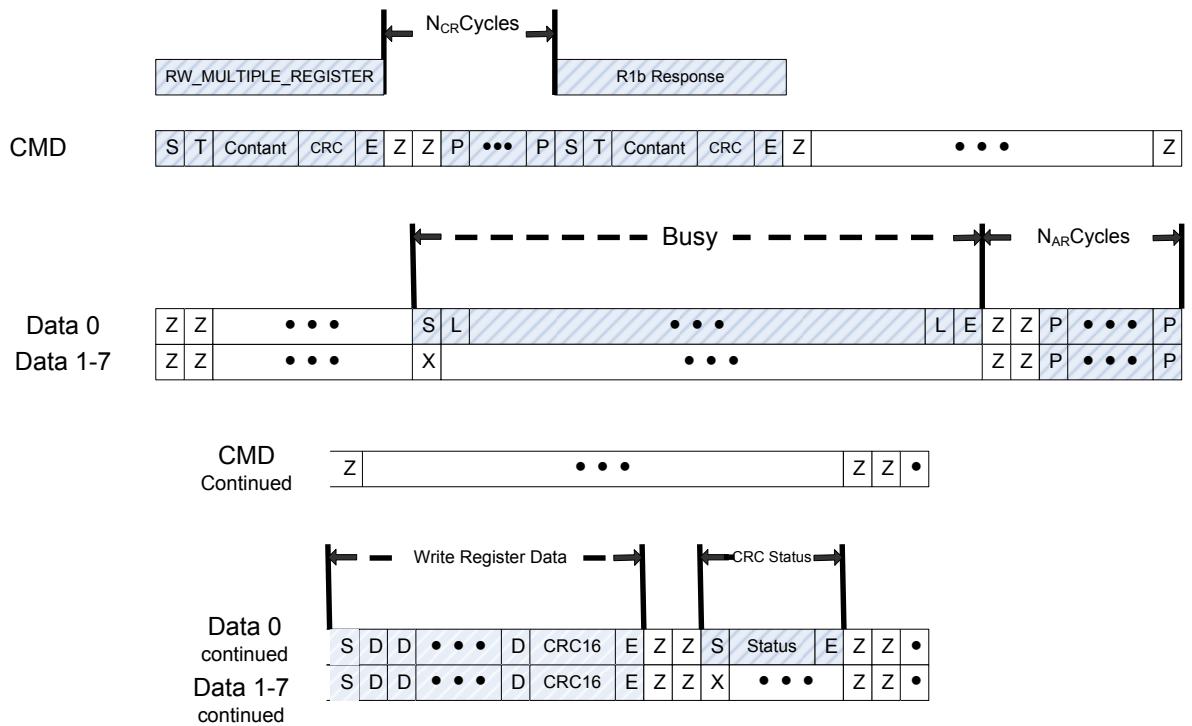


Figure 13.5 CMD60 Write Transaction

14 NFC

14.1 Function Description

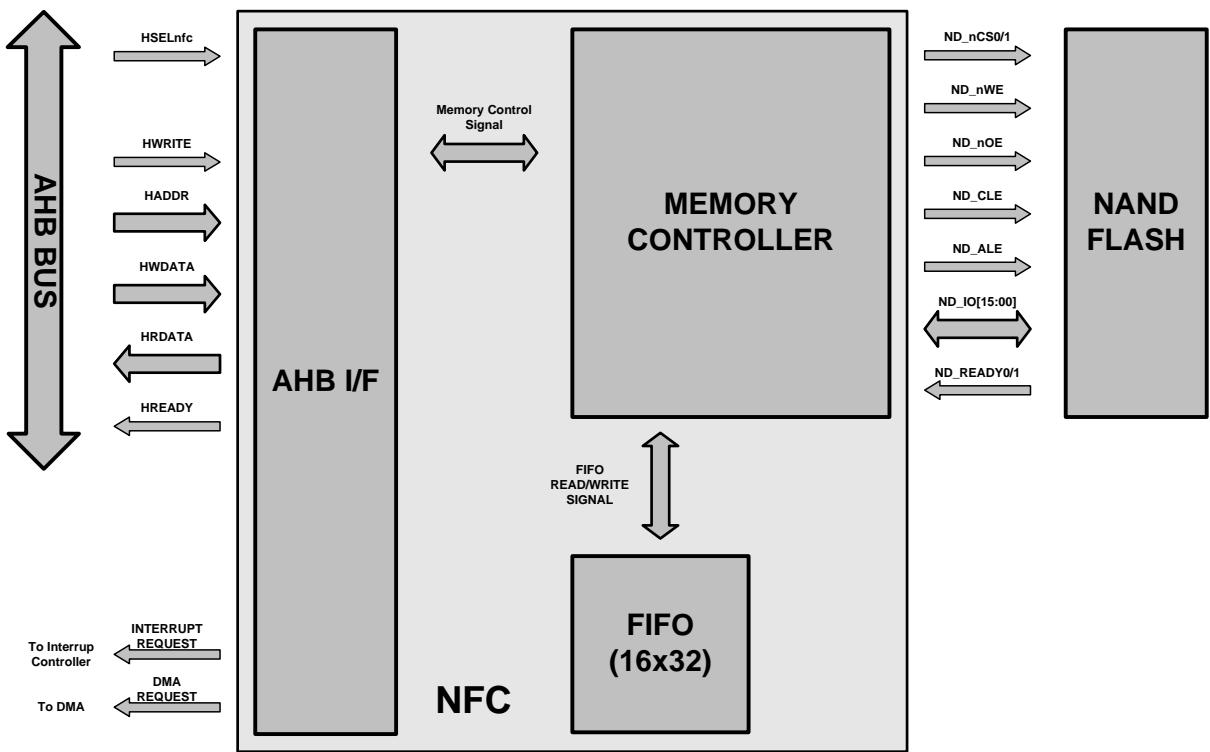


Figure 14.1 NAND Flash Controller Block Diagram

14.2 Register Description

Table 14.1 NAND Flash Controller Register Map (Base Address = 0xF00B0000)

Name	Address	Type	Reset	Description
NFC_CMD	0x00	W	-	NAND Flash Command Register
NFC_LADDR	0x04	W	-	NAND Flash Linear Address Register
NFC_BADDR	0x08	W	-	NAND Flash Block Address Register
NFC_SADDR	0x0C	W	-	NAND Flash Signal Address Register
NFC_WDATA	0x1x	R/W	Unknown	NAND Flash Word Data Register
NFC_LDATA	0x2x/3x	R/W	Unknown	NAND Flash Linear Data Register
NFC_SDATA	0x40	R/W	Unknown	NAND Flash Single Data Register
NFC_CTRL	0x50	R/W	0x03e08000	NAND Flash Control Register
NFC_PSTART	0x54	W	-	NAND Flash Program Start Register
NFC_RSTART	0x58	W	-	NAND Flash Read Start Register
NFC_DSIZE	0x5C	R/W	0x0000ffff	NAND Flash Data Size Register
NFC_IREQ	0x60	R/W	0x07000000	NAND Flash Interrupt Request Register
NFC_RST	0x64	W	-	NAND Flash Controller Reset Register
NFC_CTRL1	0x68	R/W	0x00000000	NAND Flash Control Register 1
NFC_MDATA	0x7x	R/W	Unknown	NAND Flash Multiple Data Register

Command Register (NFC_CMD)

0xF00B0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD1[7:0]								CMD0[7:0]							

For 16bit bus width of NAND flash, the NFC_CMD1 may be used as command register

For 8bit bus width of NAND flash, if MSK of NFC_CTRL is Low, the NFC_CMD1 must be Zero.

- NFC_CMD = 0xFFFF and MSK(NFC_CTRL[15]) = 1 : ND_IO[15:0] = 0x00FF
- NFC_CMD = 0xFFFF and MSK(NFC_CTRL[15]) = 0 : ND_IO[15:0] = 0xFFFF

The following values are an example commands for NAND flash of SAMSUNG. Refer to corresponding datasheet of NAND flash chip for more detailed list of commands.

(For 16bit bus width)	(for 8bit parallel configuration, 16bit bus width)
0x0000	0x0000 : Page Read Command
0x0080	0x8080 : Page Program Command
0x0060	0x6060 : Block Erase Command
0x0070	0x7070 : Status Read Command

Linear Address Register (NFC_LADDR)

0xF00B0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LADDR[15:0]															

By writing to this register, memory controller generates linear address shown in Table 19.3.

For NFC_LADDR = 0x12345600, CADDR (NFC_CTRL[14:12]) = 2 and PSIZE(NFC_CTRL[17:16]) = 1

- MSK(NFC_CTRL[15]) = 1 : ND_IO[15:0] = 0x0000 – 0x002B – 0x001A
- MSK(NFC_CTRL[15]) = 0 : ND_IO[15:0] = 0x0000 – 0x2B2B – 0x1A1A

Block Address Register (NFC_BADDR)

0xF00B0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BADDR[15:0]															

By writing to this register, memory controller generates Block address shown in Table 19.3.

For NFC_BADDR = 0x12345600, CADDR(NFC_CTRL[14:12]) = 2 and PSIZE(NFC_CTRL[17:16]) = 1

- MSK(NFC_CTRL[15]) = 1 : ND_IO[15:0] = 0x002B – 0x001A
- MSK(NFC_CTRL[15]) = 0 : ND_IO[15:0] = 0x2B2B – 0x1A1A

Single Address Cycle Register (NFC_SADDR)																0xF00B000C			
31																Reserved			
15																SADDR1[7:0]			
SADDR0[7:0]																SADDR0[7:0]			

For 16bit bus width of NAND flash, the NFC_SADDR1 may be used as single address register.

For 8bit bus width of NAND flash, if MSK of NFC_CTRL is Low, the NFC_SADDR1 must be Zero.

- NFC_SADDR = 0xFFFF and MSK(NFC_CTRL[15]) = 1 : ND_IO[15:0] = 0x00FF
- NFC_SADDR = 0xFFFF and MSK(NFC_CTRL[15]) = 0 : ND_IO[15:0] = 0xFFFF

When CPU writes to this register, one cycle of address cycle is generated.

(For 16bit bus width) (for 8bit parallel configuration, 16bit bus width)
0x0012 0x1212 : for 0x12 Single Address

Table 14.2 Page size of NAND Flash

# of Cycle (CADDR)	PSIZE = 0	PSIZE = 1	PSIZE = 2	PSIZE = 3	PSIZE > 3
1st	ADDR[7:0]	ADDR[7:0]	ADDR[7:0]	ADDR[7:0]	ADDR[7:0]
2nd	ADDR[16:9]	ADDR[16:9]	ADDR[10:8]	ADDR[11:8]	ADDR[12:8]
3rd	ADDR[24:17]	ADDR[24:17]	ADDR[18:11]	ADDR[19:12]	ADDR[20:13]
4th	ADDR[31:25]	ADDR[31:25]	ADDR[26:19]	ADDR[27:20]	ADDR[28:21]
5th	-	-	ADDR[31:27]	ADDR[31:28]	ADDR[31:29]

The Table 14.2 represents the relation between each cycle and address generation.

User must set this information appropriately to PSIZE and CADDR field of NFC_CTRL register ahead of accessing NAND data.

ADDR means address value that is written to NFC_LADDR or NFC_BADDR register. The shaded cycles represent Block address cycles. That is, NAND address cycles start from there when NFC_BADDR register is accessed.

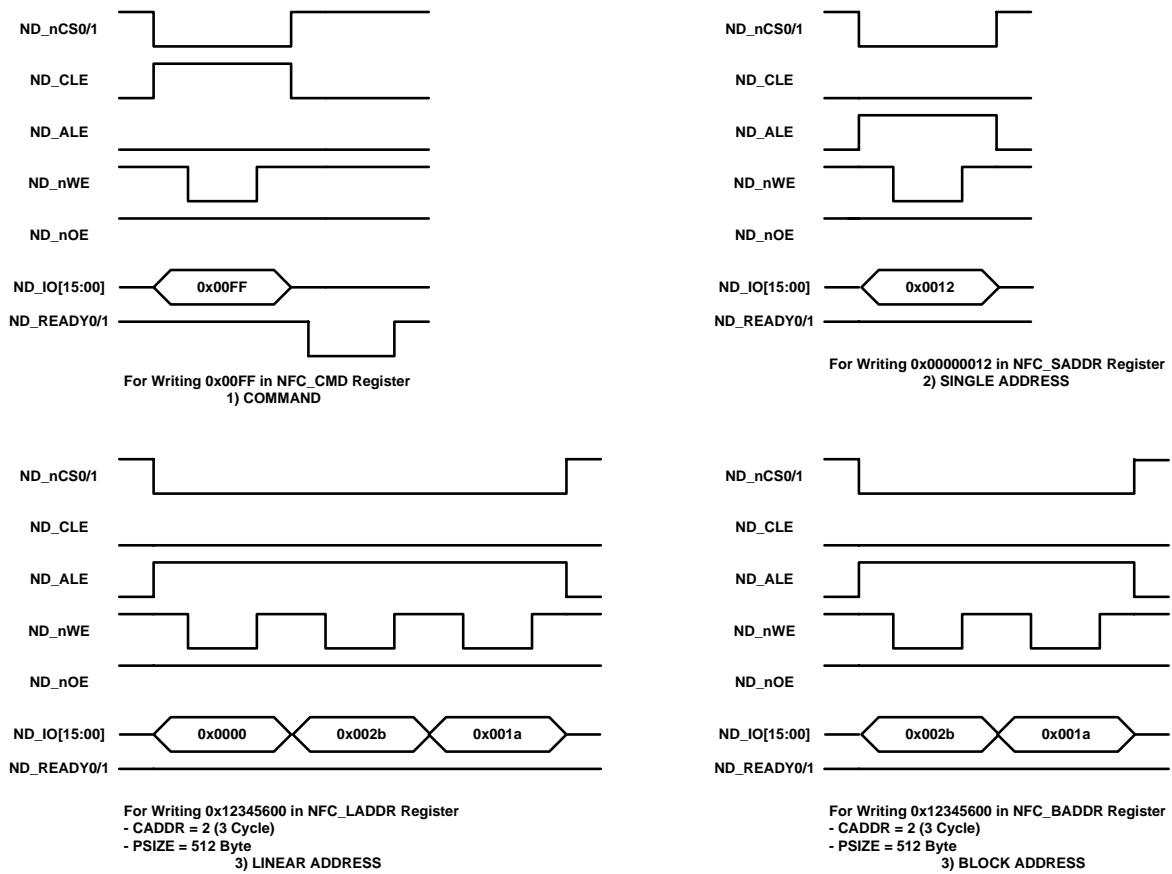


Figure 14.2 Example of Address/Command Writing Operation

Word Data Register (NFC_WDATA)

0xF00B001x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NFC_WDATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NFC_WDATA[15:0]															

This register is used for single word data transfer by CPU.

This Register is useful in reading and writing NAND Flash data of spare area.

Linear Data Register (NFC_LDATA)

0xF00B002x/3x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NFC_LDATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NFC_LDATA[15:0]															

This register is used for burst data transfer by DMA/CPU.

To start burst data transfer, User must write any value to NFC_PSTART or NFC_RSTART

Single Data Register (NFC_SDATA)

0xF00B0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NFC_SDATA[7:0]								NFC_SDATA[7:0]							

This register is used for reading and writing one 8bit or 16bit data according to NAND Flash Bus Size.

For 16bit bus width of NAND flash, the NFC_SDATA0/1 may be used as single data register, otherwise only NFC_SDATA0 is used as single data register.

This Register is useful in reading NAND ID/Status data.

Multiple Data Register (NFC_MDATA)

0xF00B007x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NFC_MDATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NFC_MDATA[15:0]															

This register is used for reading and writing multiple 8bit or 16bit data according to NAND Flash Bus Size.

NFC_CTRL1.DNUM[01:00] register defined the number of reading and writing.

NAND Flash Control Register (NFC_CTRL) **0xF00B0050**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IEN	IEN	IEN	DEN	FS	BW	CS3	CS2	CS1	CS0	RDY	BSIZE[1:0]		PSIZE[2:0]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK	CADDR[2:0]				bSTP[3:0]				bPW[3:0]				bHLD[3:0]		

RDY_IEN	[31]	NAND Flash Ready Interrupt
0	NAND Flash Ready Interrupt Disable	
1	NAND Flash Ready Interrupt Enable	
PROG_IEN	[30]	NAND Flash Program Interrupt
0	NAND Flash Program Interrupt Disable	
1	NAND Flash Program Interrupt Enable	
READ_IEN	[29]	NAND Flash Read Interrupt
0	NAND Flash Read Interrupt Disable	
1	NAND Flash Read Interrupt Enable	
DEN	[28]	NAND Flash DMA Request
0	NAND Flash DMA Request Disable	
1	NAND Flash DMA Request Enable	

Transfer type of CHCTRL register in DMA must be single transfer with level-sensitive detection.

FS	[27]	NAND Flash FIFO Status
0	FIFO status is Busy to write and read in FIFO	
1	FIFO status is Ready to write and read in FIFO	

For burst data transfer by ARM, user must check that this bit is high, ahead of access NFC_LDATA.

BW	[26]	NAND Flash Bus Width Select
0	Bus width = 8 bit	
1	Bus width = 16 bit	
CS3SEL	[25]	NAND Flash CS3 Selection
0	NAND Flash nCS3 is Enable	
1	NAND Flash nCS3 is Disable	
CS2SEL	[24]	NAND Flash CS2 Selection
0	NAND Flash nCS2 is Enable	
1	NAND Flash nCS2 is Disable	
CS1SEL	[23]	NAND Flash CS1 Selection
0	NAND Flash nCS1 is Enable	
1	NAND Flash nCS1 is Disable	

CS0SEL	[22]	NAND Flash CS0 Selection
0		NAND Flash nCS0 is Enable
1		NAND Flash nCS0 is Disable

RDY	[21]	NAND Flash Ready Flag
0		External NAND Flash is Busy
1		External NAND Flash is Ready

BSIZE[1:0]	[20:19]	Burst Size of NAND Controller
00		1Read/Write
01		2Read/Write
10		4Read/Write
11		8Read/Write

This register value must be same BURST SIZE of CHCTRL in DMA.

PSIZE[2:0]	[18:16]	Page Size of NAND Flash
000		1 Page = 256 Half-Word
001		1 Page = 512 Byte
010		1 Page = 1024 Half-Word
011		1 Page = 2048 Byte
1xx		1 Page = 4096 Byte

MSK	[15]	NAND Flash IO Mask Enable Bit
0		NAND Flash High Byte(ND_IO[15:8]) is not mask
1		NAND Flash High Byte(ND_IO[15:8]) is mask

CADDR[2:0]	[14:12]	Number of Address Cycles
N		The number of address command cycle for NAND type flash. (N+1) cycle is used for generating address cycle command.

bSTP[3:0]	[11:08]	Number of Base Cycle for Setup Time
N (= 0~15)		bSTP = N

Set-Up Cycle of NAND Flash (STP)

$$WSTP(\text{ Write Set-Up Cycle }) = bSTP(\text{ NFC_CTRL[11:8] }) + wSTP(\text{ NFC_CTRL1[11:8] })$$

$$RSTP(\text{ Read Set-Up Cycle }) = bSTP(\text{ NFC_CTRL[11:8] }) + rSTP(\text{ NFC_CTRL1[23:20] })$$

For Command and Address Cycle

- For WSTP = 0 : Set-Up Cycle of NAND Flash (STP) = 1
- For WSTP != 0 : Set-Up Cycle of NAND Flash (STP) = WSTP

For Single Data(NFC_SDATA) Write Cycle

- Set-Up Cycle of NAND Flash (STP) = WSTP

For Single Data(NFC_SDATA) Read Cycle

- Set-Up Cycle of NAND Flash (STP) = RSTP

For Word Data(NFC_WDATA) Write Cycle

- For WSTP = 0 : Set-Up Cycle of NAND Flash (STP) = 1
- For WSTP != 0 : Set-Up Cycle of NAND Flash (STP) = WSTP

For Word Data(NFC_WDATA) Read Cycle

- For RSTP = 0 : Set-Up Cycle of NAND Flash (STP) = 1
- For RSTP != 0 : Set-Up Cycle of NAND Flash (STP) = RSTP

For Linear Data(NFC_LDATA) Write Cycle

- Set-Up Cycle of NAND Flash (STP) = WSTP

For Linear Data(NFC_LDATA) Read Cycle

- Set-Up Cycle of NAND Flash (STP) = RSTP

bPW[3:0]	[07:04]	Number of Base Cycle for Pulse Width
N (= 0~15)	bPW = N	

Pulse-Width Cycle of NAND Flash (PW)

$$\begin{aligned} WPW(\text{ Write Pulse Width Cycle }) &= bPW(\text{ NFC_CTRL[7:4] }) + \\ wPW(\text{ NFC_CTRL1[7:4] }) \\ RPW(\text{ Read Pulse Width Cycle }) &= bPW(\text{ NFC_CTRL[7:4] }) + \\ rPW(\text{ NFC_CTRL1[19:16] }) \end{aligned}$$

For Command and Address Cycle

- Pulse Width Cycle of NAND Flash (PW) = WPW

For Single/Word/Linear Data(NFC_SDATA / NFC_WDATA / NFC_LDATA) Write Cycle

- Pulse Width Cycle of NAND Flash (PW) = WPW

For Single/Word/Linear Data(NFC_SDATA / NFC_WDATA / NFC_LDATA) Read Cycle

- Pulse Width Cycle of NAND Flash (PW) = RPW

bHLD[3:0]	[03:00]	Number of Base Cycle for Hold Time
N (= 0~15)	bHLD = N	

Hold Cycle of NAND Flash (HLD)

$$WHLD(\text{ Write Hold Cycle }) = bHLD(\text{ NFC_CTRL[3:0] }) + wHLD(\text{ NFC_CTRL1[3:0] })$$

$$RHLD(\text{ Read Hold Cycle }) = bHLD(\text{ NFC_CTRL[3:0] }) + rHLD(\text{ NFC_CTRL1[15:12] })$$

For Command and Address Cycle

- For WHLD = 0 : Hold Cycle of NAND Flash (HLD) = 1
- For WHLD != 0 : Hold Cycle of NAND Flash (HLD) = WHLD

For Single/Word/Linear Data(NFC_SDATA / NFC_WDATA / NFC_LDATA) Write Cycle

- For WHLD = 0 : Hold Cycle of NAND Flash (HLD) = 1
- For WHLD != 0 : Hold Cycle of NAND Flash (HLD) = WHLD

For Single/Word/Linear Data(NFC_SDATA / NFC_WDATA / NFC_LDATA) Read Cycle

- For RHLD = 0 : Hold Cycle of NAND Flash (HLD) = 1
- For RHLD != 0 : Hold Cycle of NAND Flash (HLD) = RHLD

The following figure displays the element cycle diagram for external memories.

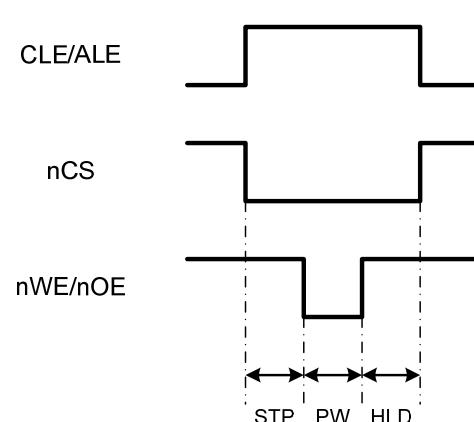


Figure 14.3 Timing Diagram of Read/Write Enable Signal

Program Start Register (NFC_PSTART)

0xF00B0054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Don't care															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Don't care															

When this register is written by any value, data transfer is started. That is, Programming operation to NAND flash is started.

For Burst data transfer by DMA/ARM, you must set EN flag of DMA_CHCTRL register first, and then write any value to NFC_PSTART ahead of accessing NFC_LDATA.

Read Start Register (NFC_RSTART)

0xF00B0058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Don't care															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Don't care															

When this register is written by any value, data transfer is started. That is, Reading operation to NAND flash is started.

For Burst data transfer by DMA/ARM, you must set EN flag of DMA_CHCTRL register first, and then write any value to NFC_RSTART ahead of accessing NFC_LDATA.

Data Size Register (NFC_DSIZE)

0xF00B005C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NFC_DSIZE[15:0]															

NFC_DSIZE	NAND Flash Data Size
N	N is transferred byte data size

For Samsung 258 Half-word/512 Byte small block, N is 512(byte).

For Samsung 1024 Half-Word/2048 Byte big block, N is 2048(byte).

NAND Flash Request Register (NFC_IREQ)

0xF00B0060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															

FLAG[2]	[6]	NAND Flash Ready Flag
1	Read	The Rising edge of Ready Signal is occurred.
1	Write	Ready Flag Clear

When this register is cleared, ready interrupt request also is cleared.

FLAG[1]	[5]	NAND Flash Program Flag
1	Read	Program data transfer is finished.
1	Write	Program Flag Clear

When this register is cleared, program interrupt request also is cleared.

FLAG[0]	[4]	NAND Flash Read Flag	
1	Read	Read	data transfer is finished.
1	Write	Read	Flag Clear

When this register is cleared, read interrupt request also is cleared.

IRQ[2]	[2]	NAND Flash Ready Interrupt Request	
1	Read	Ready	Interrupt is occurred.
1	Write	Ready	Interrupt Request Clear

When this register is cleared, ready flag also is cleared.

IRQ[1]	[1]	NAND Flash Program Interrupt Request	
1	Read	Program	Interrupt is finished.
1	Write	Program	Interrupt Request Clear

When this register is cleared, program flag also is cleared.

IRQ[0]	[0]	NAND Flash Read Interrupt Request	
1	Read	Read	Interrupt is finished.
1	Write	Read	Interrupt Request Clear

When this register is cleared, read flag also is cleared.

Controller Reset Register (NFC_RST)																0xF00B0064
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Don't care																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Don't care																

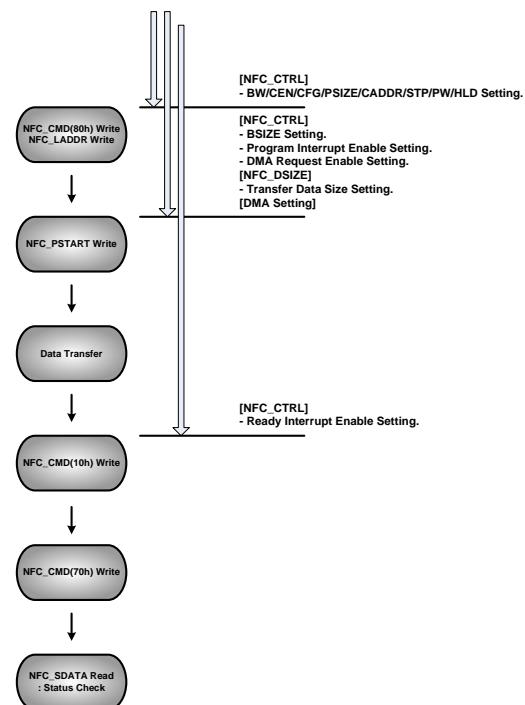
When this register is written by any value, Controller and Register is reset.

NAND Flash Control Register 1(NFC_CTRL1)

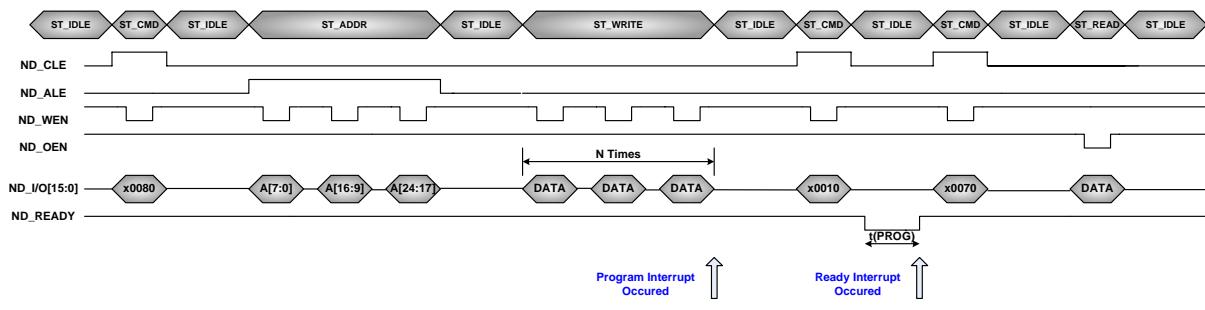
0xF00B0068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAK					IDL	DNUM[1:0]			rSTP[3:0]				rPW[3:0]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rHLD[3:0]				wSTP[3:0]				wPW[3:0]				wHLD[3:0]			

IDL	[26]	NFC IDLE State Flag
0		NFC Active State(Read Only)
1		NFC IDLE State (Read Only)
DNUM[01:00]	[25:24]	NFC MDATA Number
N		N = NFC MDATA Access Number
DACK	[31]	DMA Acknowledge Selection
0		DMA Acknowledge Disable For burst data transfer by ARM, this register is low.
1		DMA Acknowledge Enable
rSTP[3:0]	[23:20]	Number of Read Cycle for Setup Time
N (= 0~15)		rSTP = N
rPW[3:0]	[19:16]	Number of Read Cycle for Pulse Width
N (= 0~15)		rPW = N
rHLD[3:0]	[15:12]	Number of Read Cycle for Hold Time
N (= 0~15)		rHLD = N
wSTP[3:0]	[11:08]	Number of Write Cycle for Setup Time
N (= 0~15)		wSTP = N
wPW[3:0]	[07:04]	Number of Write Cycle for Pulse Width
N (= 0~15)		wPW = N
wHLD[3:0]	[03:00]	Number of Write Cycle for Hold Time
N (= 0~15)		wHLD = N



A) NAND FLASH Page Program Flow Chart



B) NAND FLASH Page Program Timing Diagram

Figure 14.4 Example of Page Program

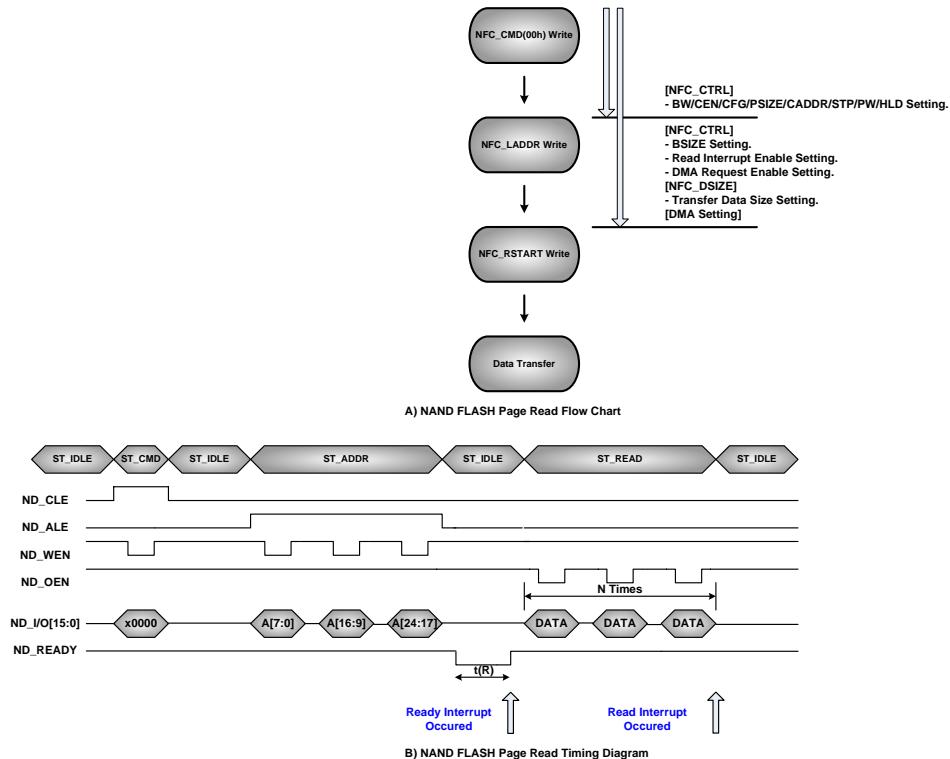


Figure 14.5 Example of Page Read

15 ECC

15.1 Functional Description

The ECC (Error Correction Code) is used to correct data error in storage device or various kind of communicating system. The TCC79XX has simple ECC generation and Error Correction Module. By enable ECC module, it consistently monitors internal bus activity and calculate ECC whenever there is read or write cycle from/to a predefined memory area. The area can be determined by special Register so this module can be used ECC calculation itself not only for specific storage device such as NAND flash.

The following figure represents block diagram including internal bus connection for ECC module.

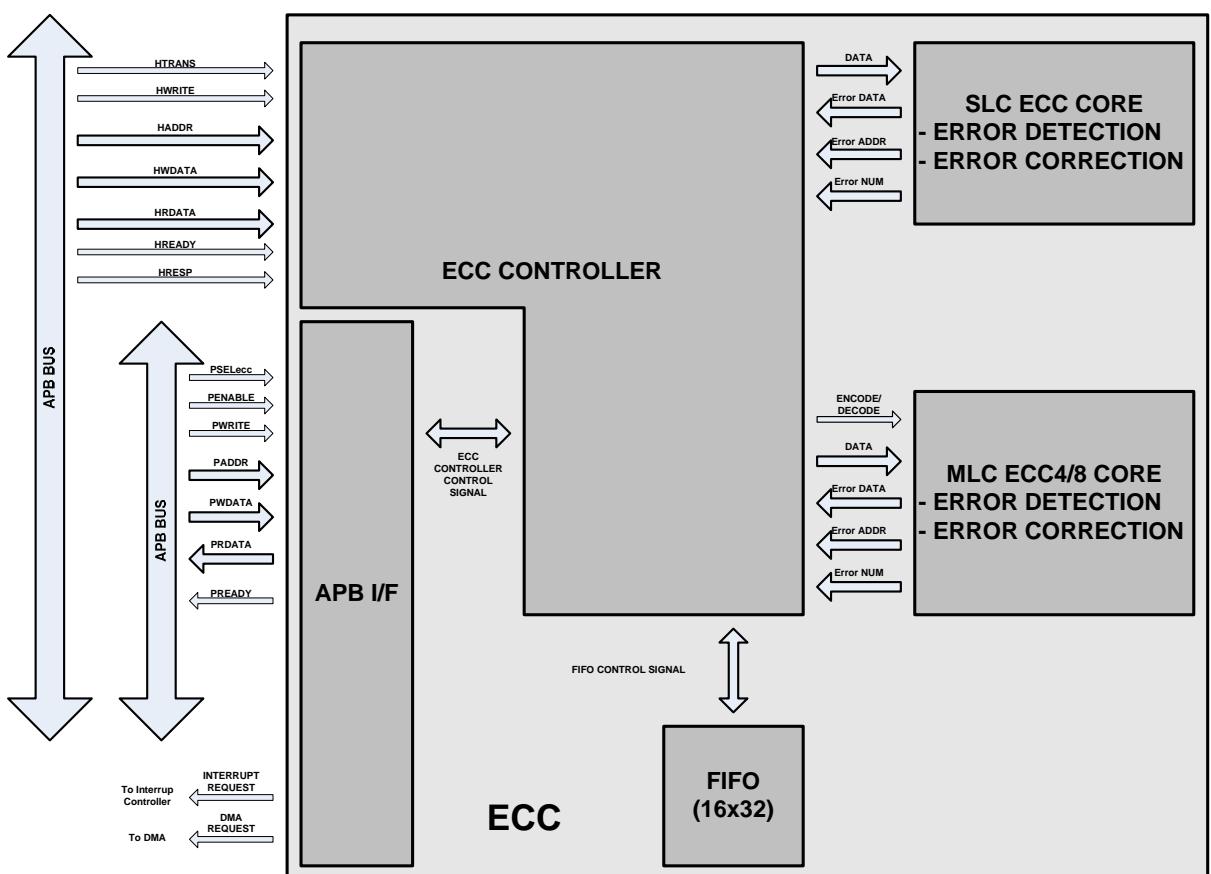


Figure 15.1 ECC Block Diagram

15.2 Register Description

Table 15.1 ECC Register Map (Base Address = 0xF005B000)

Name	Address	Type	Reset	Description
ECC_CTRL	0x00	R/W	0x00000000	ECC Control Register
ECC_BASE	0x04	R/W	0x00000000	Base Address for ECC Calculation
ECC_MASK	0x08	R/W	0x00000000	Address mask for ECC area.
ECC_CLEAR	0x0C	R/W		ECC Clear
SECC_0	0x10	R/W	0x00000000	1st SLC ECC Code Register
SECC_1	0x14	R/W	0x00000000	2nd SLC ECC Code Register
SECC_2	0x18	R/W	0x00000000	3rd SLC ECC Code Register
SECC_3	0x1C	R/W	0x00000000	4th SLC ECC Code Register
SECC_4	0x20	R/W	0x00000000	5th SLC ECC Code Register
SECC_5	0x24	R/W	0x00000000	6th SLC ECC Code Register
SECC_6	0x28	R/W	0x00000000	7th SLC ECC Code Register
SECC_7	0x2C	R/W	0x00000000	8th SLC ECC Code Register
SECC_8	0x30	R/W	0x00000000	9th SLC ECC Code Register
SECC_9	0x34	R/W	0x00000000	10th SLC ECC Code Register
SECC_10	0x38	R/W	0x00000000	11th SLC ECC Code Register
SECC_11	0x3C	R/W	0x00000000	12th SLC ECC Code Register
SECC_12	0x40	R/W	0x00000000	13th SLC ECC Code Register
SECC_13	0x44	R/W	0x00000000	14th SLC ECC Code Register
SECC_14	0x48	R/W	0x00000000	15th SLC ECC Code Register
SECC_15	0x4C	R/W	0x00000000	16th SLC ECC Code Register
MECC4_0	0x10	R/W	0x00000000	1st MLC ECC4 Code Register
MECC4_1	0x14	R/W	0x00000000	2nd MLC ECC4 Code Register
MECC4_2	0x18	R/W	0x00000000	3rd MLC ECC4 Code Register
MECC8_0	0x10	R/W	0x00000000	1st MLC ECC8 Code Register
MECC8_1	0x14	R/W	0x00000000	2nd MLC ECC8 Code Register
MECC8_2	0x18	R/W	0x00000000	3rd MLC ECC8 Code Register
MECC8_3	0x1C	R/W	0x00000000	4th MLC ECC8 Code Register
MECC8_4	0x20	R/W	0x00000000	5th MLC ECC8 Code Register
SECC_EADDR0	0x50	R	0x00000000	SLC ECC Error Address Register0
SECC_EADDR1	0x54	R	0x00000000	SLC ECC Error Address Register1
SECC_EADDR2	0x58	R	0x00000000	SLC ECC Error Address Register2
SECC_EADDR3	0x5C	R	0x00000000	SLC ECC Error Address Register3
SECC_EADDR4	0x60	R	0x00000000	SLC ECC Error Address Register4
SECC_EADDR5	0x64	R	0x00000000	SLC ECC Error Address Register5
SECC_EADDR6	0x68	R	0x00000000	SLC ECC Error Address Register6
SECC_EADDR7	0x6C	R	0x00000000	SLC ECC Error Address Register7
SECC_EADDR8	0x70	R	0x00000000	SLC ECC Error Address Register8
SECC_EADDR9	0x74	R	0x00000000	SLC ECC Error Address Register9
SECC_EADDR10	0x78	R	0x00000000	SLC ECC Error Address Register10
SECC_EADDR11	0x7C	R	0x00000000	SLC ECC Error Address Register11
SECC_EADDR12	0x80	R	0x00000000	SLC ECC Error Address Register12
SECC_EADDR13	0x84	R	0x00000000	SLC ECC Error Address Register13

SECC_EADDR14	0x88	R	0x00000000	SLC ECC Error Address Register14
SECC_EADDR15	0x8C	R	0x00000000	SLC ECC Error Address Register15
MECC4_EADDR0	0x50	R	0x00000000	MLC ECC Error Address Register0
MECC4_EADDR1	0x54	R	0x00000000	MLC ECC Error Address Register1
MECC4_EADDR2	0x58	R	0x00000000	MLC ECC Error Address Register2
MECC4_EADDR3	0x5C	R	0x00000000	MLC ECC Error Address Register3
MECC4_EDATA0	0x70	R	0x00000000	MLC ECC Error Data Register0
MECC4_EDATA1	0x74	R	0x00000000	MLC ECC Error Data Register1
MECC4_EDATA2	0x78	R	0x00000000	MLC ECC Error Data Register2
MECC4_EDATA3	0x7C	R	0x00000000	MLC ECC Error Data Register3
MECC8_EADDR0	0x50	R	0x00000000	MLC ECC8 Error Address Register0
MECC8_EADDR1	0x54	R	0x00000000	MLC ECC8 Error Address Register1
MECC8_EADDR2	0x58	R	0x00000000	MLC ECC8 Error Address Register2
MECC8_EADDR3	0x5C	R	0x00000000	MLC ECC8 Error Address Register3
MECC8_EADDR4	0x60	R	0x00000000	MLC ECC8 Error Address Register4
MECC8_EADDR5	0x64	R	0x00000000	MLC ECC8 Error Address Register5
MECC8_EADDR6	0x68	R	0x00000000	MLC ECC8 Error Address Register6
MECC8_EADDR7	0x6C	R	0x00000000	MLC ECC8 Error Address Register7
MECC8_EDATA0	0x70	R	0x00000000	MLC ECC8 Error Data Register0
MECC8_EDATA1	0x74	R	0x00000000	MLC ECC8 Error Data Register1
MECC8_EDATA2	0x78	R	0x00000000	MLC ECC8 Error Data Register2
MECC8_EDATA3	0x7C	R	0x00000000	MLC ECC8 Error Data Register3
MECC8_EDATA4	0x80	R	0x00000000	MLC ECC8 Error Data Register4
MECC8_EDATA5	0x84	R	0x00000000	MLC ECC8 Error Data Register5
MECC8_EDATA6	0x88	R	0x00000000	MLC ECC8 Error Data Register6
MECC8_EDATA7	0x8C	R	0x00000000	MLC ECC8 Error Data Register7
ERRNUM	0x90	R	0x00000000	ECC Error Number
ECC_IREQ	0x94	R/W	0x00000000	ECC Interrupt Control Register
ECC_FSMSTATE	0x98	R	0x00000001	ECC FSM State Register
ENCSEED	0xF0	W		Test Mode Register
ENCMASK	0xF4	W		Test Mode Register
ENCDATA	0xF8	R/W		Test Mode Register

ECC Control Register (ECC_CTRL)																0xF005B000
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
IEN	IEN	DEN	DRQ													DATASIZE[10:00]
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	ECCCNT[09:00]
																EN[2:0]

IEN_MECC8	[31]	MLC ECC8 Decoding Interrupt Enable
0		MLC ECC8 Decoding Interrupt Disable
1		MLC ECC8 Decoding Interrupt Enable

*) After MLC ECC8 Decoding interrupt, ERRNUM Register can be used to detect data error, and it is able to correct data error by ERRADDRx/ERRDATAx.

If IEN_MECC8 bit of ECC_CTRL Register is set and MLC ECC8 Decoding operation is finished, ECC interrupt is really generated and user can clear interrupt request Register by writing "1" into FLG field or IREQ field.

IEN_MECC4	[30]	MLC ECC4 Decoding Interrupt Enable
0		MLC ECC4 Decoding Interrupt Disable
1		MLC ECC4 Decoding Interrupt Enable

*) After MLC ECC4 Decoding interrupt, ERRNUM Register can be used to detect data error, and it is able to correct data error by ERRADDRx/ERRDATAx.

If IEN_MECC4 bit of ECC_CTRL Register is set and MLC ECC4 Decoding operation is finished, ECC interrupt is really generated and user can clear interrupt request Register by writing "1" into FLG field or IREQ field.

DEN	[29]	ECC DMA Request Enable
0		ECC DMA Request Disable
1		ECC DMA Request Enable

*) For ECC DMA Request Enable, DMA Transfer Type must be Single Transfer with Level Sensitive Detection and DMA Word Size must be 32 bit Data..

In case of MLC ECC4/8, if Transfer is not done through nand Flash, DEN should definitely be Enabled.

DATASIZE	[26:16]	ECC Data Size
N	Data Size = N	For SLC ECC : $0 < N \leq 1024$ (Word Size)
N	Data Size = N	For MLC ECC4 : $0 < N \leq 512 + 6$ (Byte Size)
N	Data Size = N	For MLC ECC8 : $512 \leq N \leq 512 + 16$ (Byte Size)

*) In case of MLC ECC4/8, Data Size is Byte Size.

In case of SLC ECC, Data Size is Word Size.

ECCNT	[13:04]	ECC Counter Value
N		ECC Counter Value (Read Only)

*) In case of MLC ECC4/8, Data Size is Byte Size.

In case of SLC ECC, Data Size is Word Size.

EN	[02:00]	ECC Enable Control
00x	ECC Disable	
010	SLC ECC Encoding Enable	
011	SLC ECC Decoding Enable	
100	MLC ECC4 Encoding Enable	
101	MLC ECC4 Decoding Enable	
110	MLC ECC8 Encoding Enable	
111	MLC ECC8 Decoding Enable	

ECC Base Address Register (ECC_BASE)

0xF005B004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECC_BASE[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_BASE[15:2]															

ECC Address Mask Register (ECC_MASK)

0xF005B008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_MASK[15:2]															

*)The ECC is calculated whenever the specified region of memory is accessed. The region for ECC calculating is determined by ECC_BASE & ECC_MASK Register. The real base address is determined by following formula.

$$\text{Real base address} = \text{ECC_BASE}(0xF005B004) \& \sim(\text{ECC_MASK}[27:2] << 2)$$

(The real base address is assumed to be word aligned, so the least 2 bits are always 0.)

The size of region is also determined by ECC_MASK Register. If ECC_MASK Register has N concatenated 0 from LSB, the region size is set to 2^N bytes.

ECC Clear Register (ECC_CLR)

0xF005B00C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Don't care															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Don't care															

*) Whenever this Register is written by any value, SLC / MLC4 / MLC8 Block are cleared.

SLC ECC Code Register (SECC_x)															0xF005B010 + x*4							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16							
				0					SLC_ECCn_0													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SLC_ECCn_1	SLC_ECCn_2					

*) These Registers contain ECC output for SLC. It calculates ECC of SSFDC standard, and can contain up to 8 blocks of data.

For each output Register, there are a total of 22 bits of parity data (6 bits for column parity and 16 bits for line parity) as follows:

P1, P1', P2, P2', P4, P4', P8, P8', P16, P16',, P1024, P1024'

The parity data that have been generated are stored as follows.

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SLC_ECC0_0	P64	P64'	P32	P32'	P16	P16'	P8	P8'
SLC_ECC0_1	P1024	P1024'	P512	P512'	P256	P256'	P128	P128'
SLC_ECC0_2	P4	P4'	P2	P2'	P1	P1'	1	1

*) To correct error for SLC ECC, writing original ECC data that read from nand flash to SLC_ECCx Register.

For SLC ECC

Address	512 Byte	...	2048 Byte	4096 Byte
0xF005B010	SECC[047:024]		SECC[191:168]	SECC[383:360]
0xF005B014	SECC[023:000]		SECC[167:144]	SECC[359:336]
0xF005B018			SECC[143:120]	SECC[335:312]
0xF005B01C			SECC[119:096]	SECC[311:288]
0xF005B020			SECC[095:072]	SECC[287:264]
0xF005B024			SECC[071:048]	SECC[263:240]
0xF005B028			SECC[047:024]	SECC[239:216]
0xF005B02C			SECC[023:000]	SECC[215:192]
0xF005B030				SECC[191:168]
0xF005B034				SECC[167:144]
0xF005B038				SECC[143:120]
0xF005B03C				SECC[119:096]
0xF005B040				SECC[095:072]
0xF005B044				SECC[071:048]
0xF005B048				SECC[047:024]
0xF005B04C				SECC[023:000]

MLC ECC4 Code Register (MECC4_0)

0xF005B010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MLC_ECC4[15:00]

MLC ECC4 Code Register (MECC4_1)

0xF005B014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MLC_ECC4[47:32]

MLC_ECC4[31:16]

MLC ECC4 Code Register (MECC4_2)

0xF005B018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MLC_ECC[79:64]

MLC_ECC[63:48]

For MLC ECC4

Address	MLC ECC4 Code
0xF005B010	MECC4[15:00]
0xF005B014	MECC4[47:16]
0xF005B018	MECC4[79:48]

MLC ECC8 Code Register (MECC8_0)

0xF005B010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MLC_ECC8[15:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MLC_ECC8[15:00]															

For MLC ECC8

Address	MLC ECC8 Code
0xF005B010	MECC8[031:000]
0xF005B014	MECC8[063:032]
0xF005B018	MECC8[095:064]
0xF005B01C	MECC8[127:096]
0xF005B020	MECC8[159:128]

MLC ECC8 Code Register (MECC8_1)

0xF005B014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MLC_ECC8[15:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MLC_ECC8[15:00]															

MLC ECC8 Code Register (MECC8_2)

0xF005B018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MLC_ECC8[15:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MLC_ECC8[15:00]															

MLC ECC8 Code Register (MECC8_3)

0xF005B01C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MLC_ECC8[15:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MLC_ECC8[15:00]															

MLC ECC8 Code Register (MECC8_4)

0xF005B020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MLC_ECC8[15:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MLC_ECC8[15:00]															

SLC ECC Error Address (SECC_EADDRx, x = 0,...,15)

0xF005B050 + x*4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SECC_EADDRx[14:0]

*) Byte Address = SECC_EADDRx[14:3], Error Bit Address[2:0].

MLC ECC4 Error Address Register (MECC4_EADDRx, x = 0,1,2)															0xF005B050 + x*4			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	MECC_EADDRx[09:00]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			

For MECC4_EADDRx[9:0] >= 8,

Error Address = Byte Data Size(ECC_CTRL.DATASIZE) + 8 – MECC4_EADDRx[9:0] – 1

For MECC4_EADDRx[9:0] < 8,

Error Address = ECC Code Area.

MLC ECC4 Error Data Register (MECC4_EDATAx, x = 0,1,2)															0xF005B070 + x*4			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	MECC_EDATAx[09:00]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			

Correction Data = Error Data[9:0] ^ MECC4_EDATAx.

MLC ECC8 Error Address Register (MECC8_EADDRx, x = 0,...,7)															0xF005B050 + x*4				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	MECC8_EADDRx[09:00]			

For MECC8_EADDRx[9:0] >= 16,

Error Address = Byte Data Size(ECC_CTRL.DATASIZE) + 16 – MECC4_EADDRx[9:0] – 1

For MECC8_EADDRx[9:0] < 16,

Error Address = ECC Code Area

MLC ECC8 Error Data Register (MECC8_EDATAx, x = 0,...,7)															0xF005B070 + x*4				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	MECC8_EDATAx[09:00]			

Correction Data = Error Data[9:0] ^ MECC8_EDATAx.

MLC ECC4 Error Number Register (ERRNUM)

0xF005B070

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERRNUM[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRNUM[15:0]															

ERRNUM	SLC ECC Status
[2*n:n]	DECODE
00	No Error
01	ECC Error
10	Correctable Error
11	Correction Impossible Error

(n = 0,..., 15)

ERRNUM	MLC ECC4 Status
[4:0]	DECODE
00000	No Error
00001	1 Symbol Error Occurred
00010	2 Symbol Error Occurred
00011	3 Symbol Error Occurred
00100	4 Symbol Error Occurred
00101	Error Symbol > 4
00110	
~ 11111	

ERRNUM	MLC ECC8 Status
[4:0]	ENCODE DECODE
00000	Encoding Normal Completion No Error
00001	Insufficient Data Error 1 Symbol Error Occurred
00010	Excessive Data Error 2 Symbol Error Occurred
00011	
00100	
00101	
00110	
00111	
01000	
01001	
~ 10000	
10001	Insufficient Data Error
10010	Excessive Data Error
10011	Error in Bit[9:8]
10100	Detected Error out of Memory Map
10101	
~ 11101	Error Symbol > 8
1111x	

ECC Interrupt Request Register (ECC_IREQ)

0xF005B094

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										SEF	SDF	M4EF	M4DF	M8EF	M8DF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													M4DI		M8DI

SEF	[21]	SLC ECC Encoding Flag Register
0	Read	SLC ECC Encoding is not Completed
1	Read	SLC ECC Encoding is Completed
1	Write	SLC ECC Encoding Flag Clear
SDF	[20]	SLC ECC Decoding Flag Register
0	Read	SLC ECC Decoding is not Completed
1	Read	SLC ECC Decoding is Completed
1	Write	SLC ECC Decoding Flag and Interrupt Request Clear
M4EF	[19]	MLC ECC4 Encoding Flag Register
0	Read	MLC ECC4 Encoding is not Completed
1	Read	MLC ECC4 Encoding is Completed
1	Write	MLC ECC4 Encoding Flag Clear
M4DF	[18]	MLC ECC4 Decoding Flag Register
0	Read	MLC ECC4 Decoding is not Completed
1	Read	MLC ECC4 Decoding is Completed
1	Write	MLC ECC4 Decoding Flag and Interrupt Request Clear
M8EF	[17]	MLC ECC8 Encoding Flag Register
0	Read	MLC ECC8 Encoding is not Completed
1	Read	MLC ECC8 Encoding is Completed
1	Write	MLC ECC8 Encoding Flag Clear
M8DF	[16]	MLC ECC8 Decoding Flag Register
0	Read	MLC ECC8 Decoding is not Completed
1	Read	MLC ECC8 Decoding is Completed
1	Write	MLC ECC8 Decoding Flag and Interrupt Request Clear
M4DI	[02]	MLC ECC4 Decoding Interrupt Request Register
0	Read	MLC ECC4 Decoding Interrupt Request is not Occurred
1	Read	MLC ECC4 Decoding Interrupt Request is Occurred
1	Write	MLC ECC4 Decoding Flag and Interrupt Request Clear
M8DI	[00]	MLC ECC8 Decoding Interrupt Request Register
0	Read	MLC ECC8 Decoding Interrupt Request is not Occurred.
1	Read	MLC ECC8 Decoding Interrupt Request is Occurred
1	Write	MLC ECC8 Decoding Flag and Interrupt Request Clear

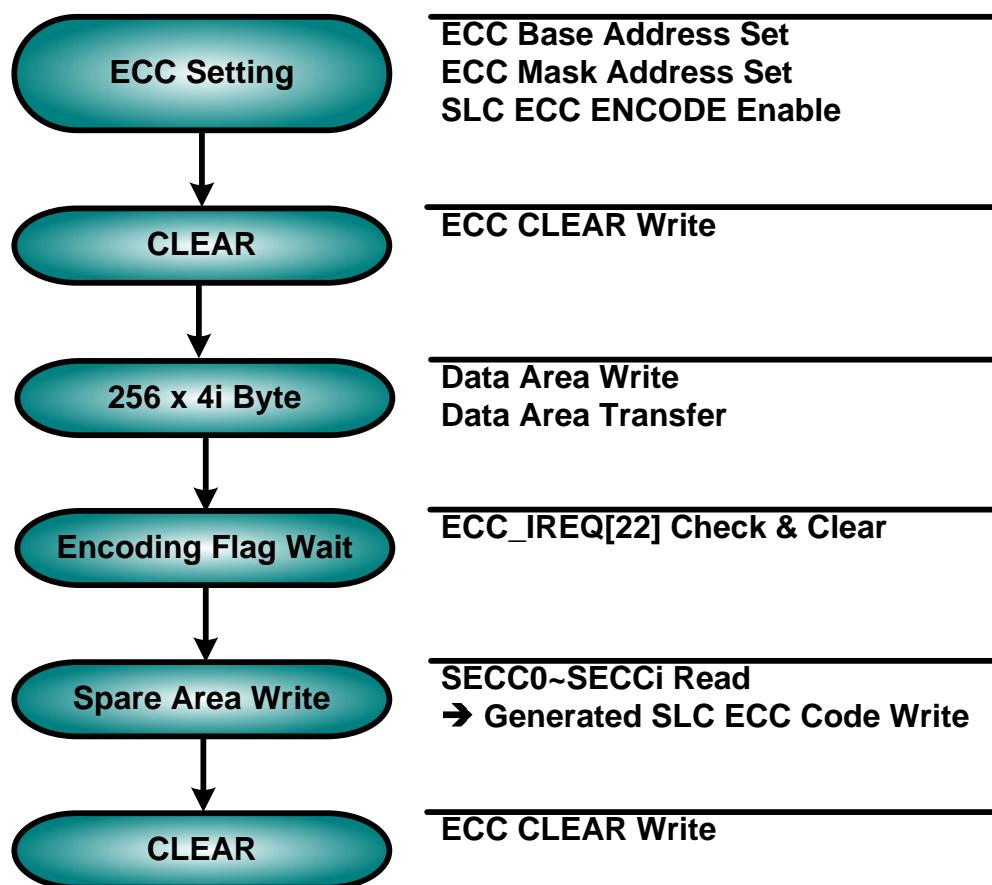


Figure 15.2 SLC ECC Encoding Sequence

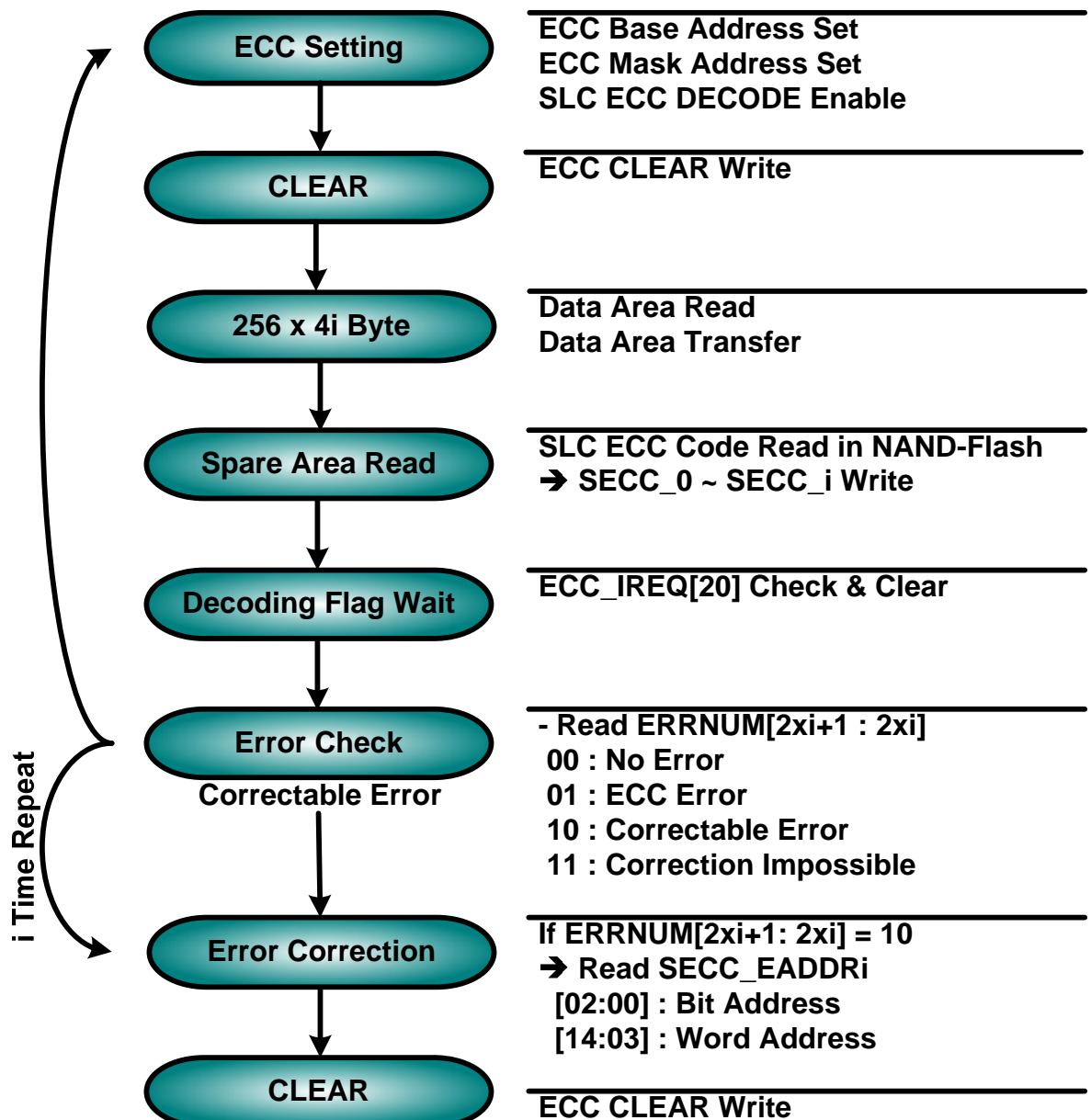


Figure 15.3 SLC ECC Decoding Sequence

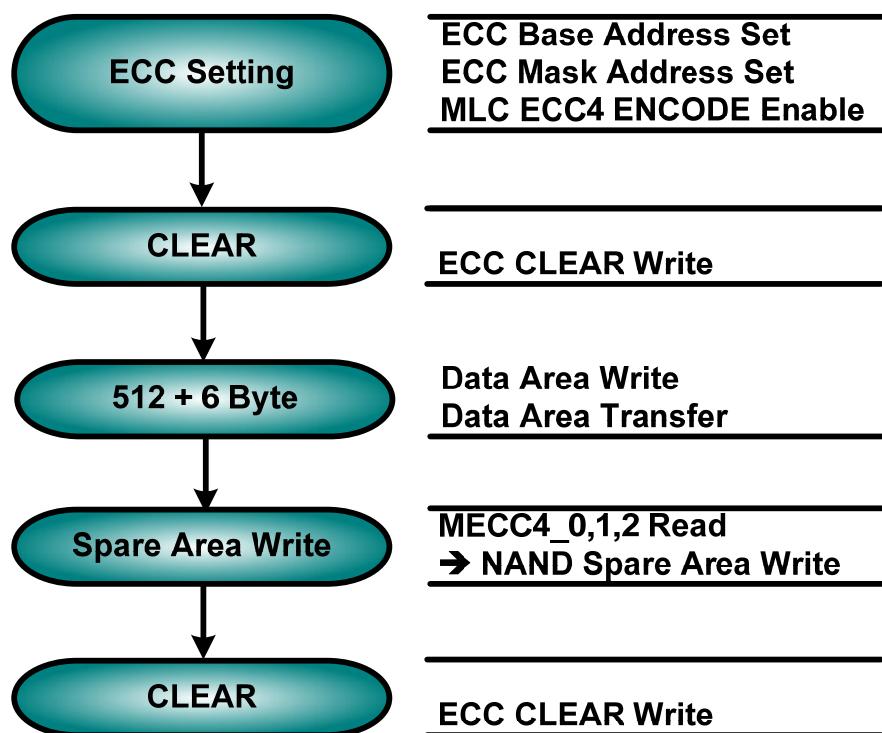


Figure 15.4 MLC ECC4 Encoding Sequence

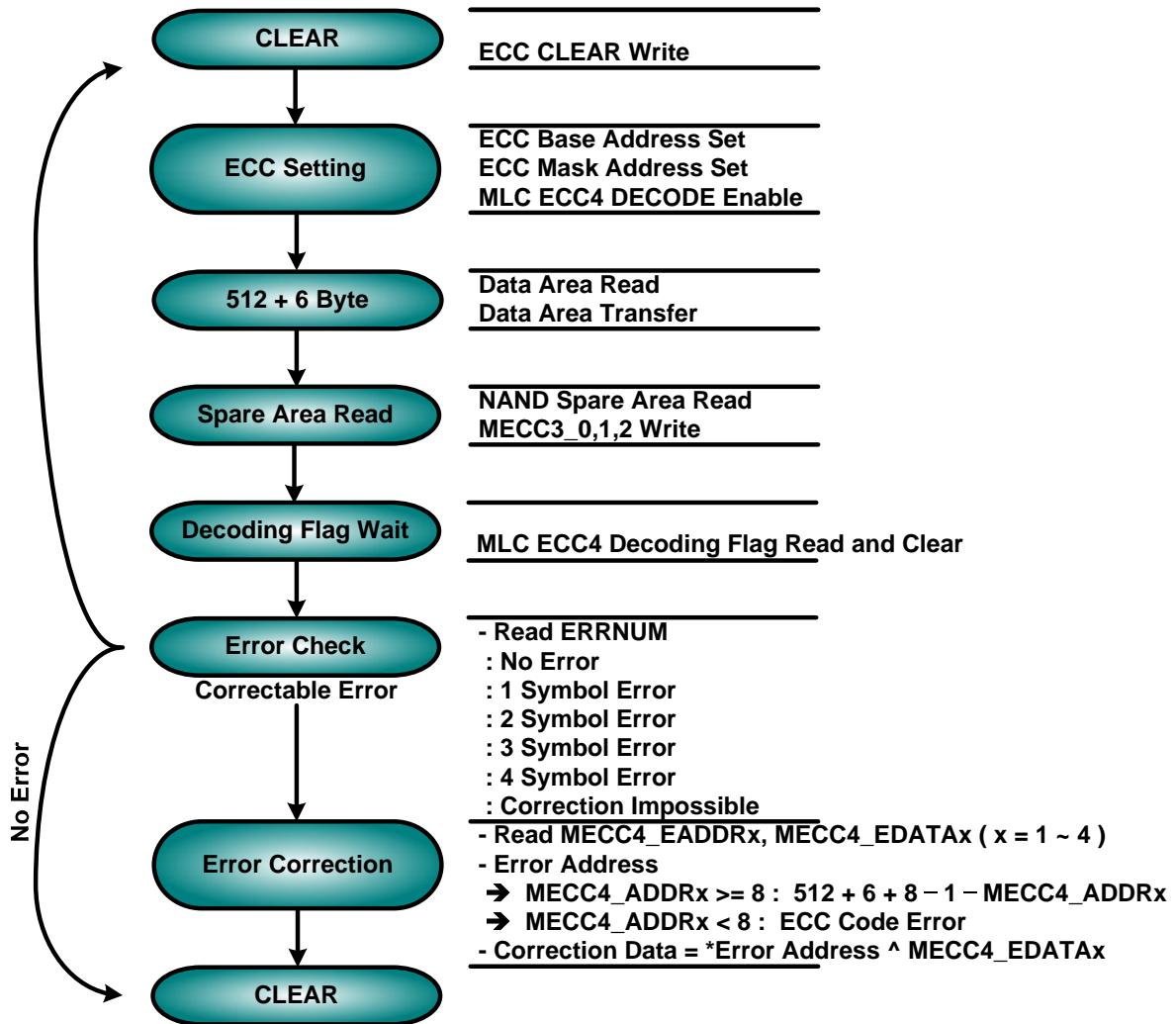


Figure 15.5 MLC ECC4 Decoding Sequence

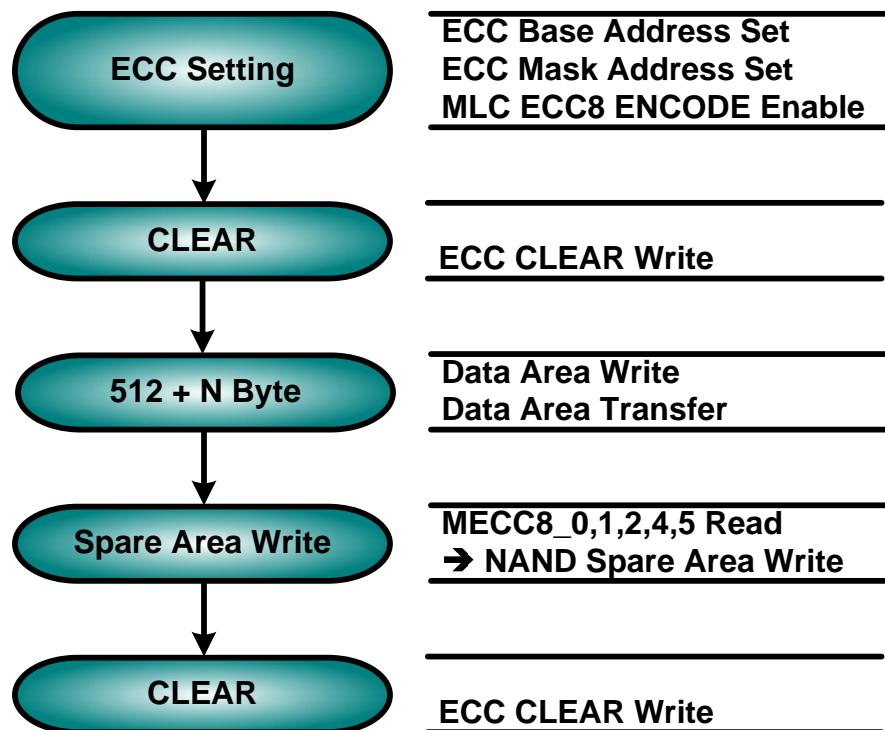


Figure 15.6 MLC ECC8 Encoding Sequence

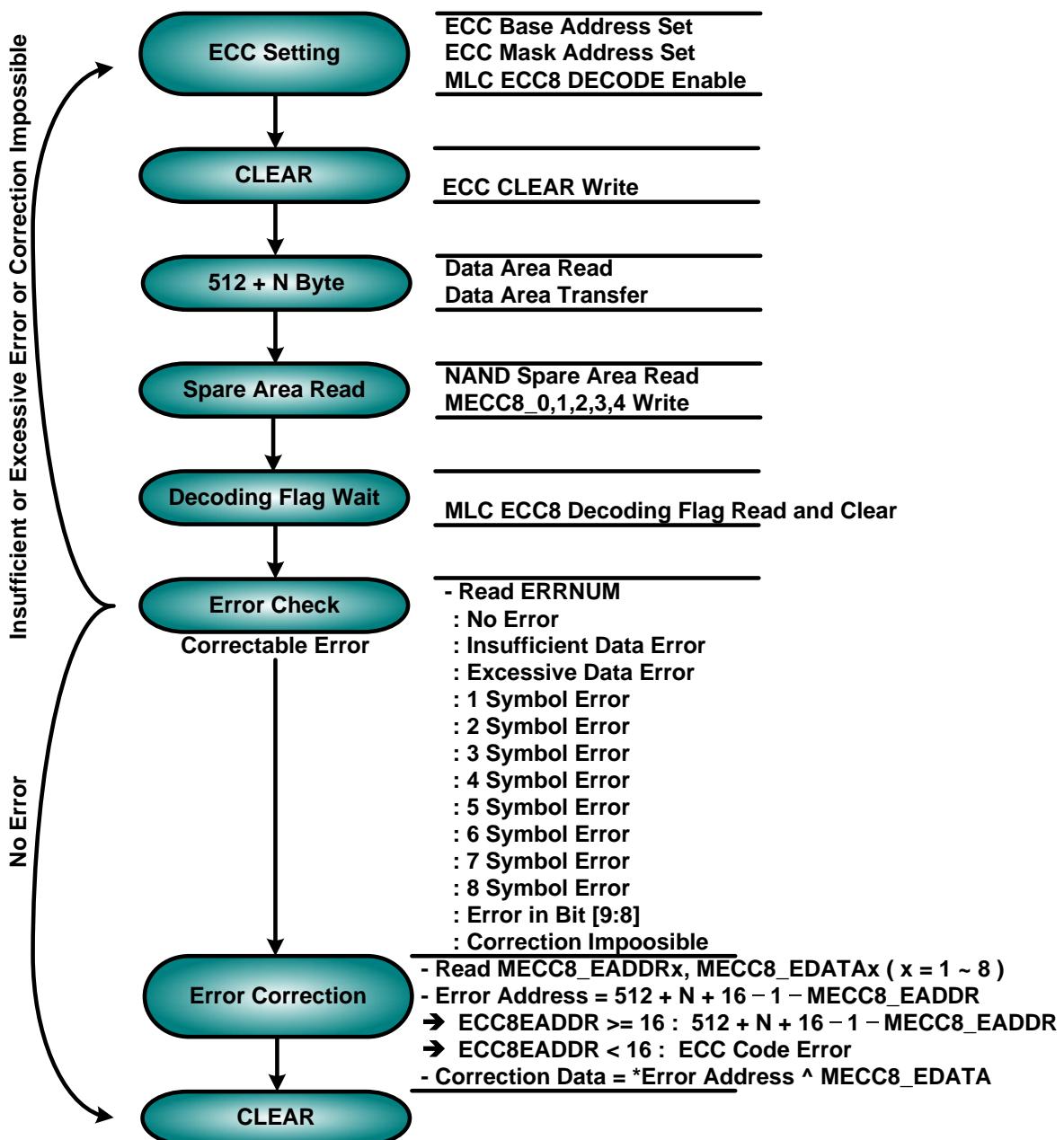


Figure 15.7 MLC ECC8 Decoding Sequence

16 CAMERA INTERFACE

16.1 Overview

The TCC79XX has the camera interface (CIF). Its features are as follows.

- INPUT FORMATS
 - CCIR 601/656, YUV4:2:2 (8bits)
 - RGB555, RGB565 (8 bits)
- OUTPUT FORMATS
 - YUV4:2:2, YUV4:2:0
 - Convert YUV422 to YUV420 mode
- Skip Frame Mode
- 4-level alpha blending/Chroma-Keying (1 overlay image)
- Image effector
 - Bias
 - Inversion of Y value
 - Strong C mode
 - Y clipping
 - Color filter
 - Sketch mode
 - Embossing
 - Gray
 - Sepia
- Image scaler (ratio : original * 256/target)
 - Upscale: 1 : 4
 - Downscale: 64 : 1
 - Each step size is 256 step
 - Max. Image: 1600 * 1200 at 8fps.

16.2 Operation

The block diagram of CIF is shown in Figure 16.1.

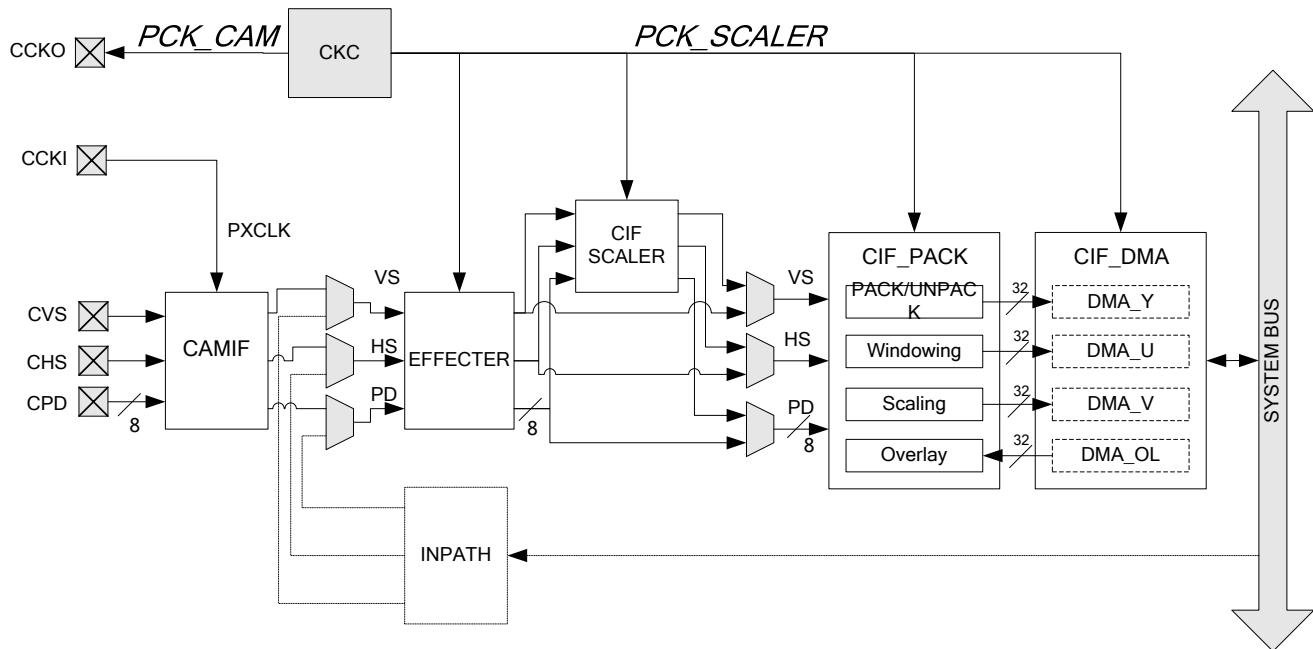


Figure 16.1 CIF Block Diagram

External Camera Module Interface (CAMIF)

The CIF uses CCIR-601/656-like protocol to interface with an external camera module. These signals are shown in the Figure 16.2.

The C656 bit of ICPCR1 register determines whether CPD[7:0] includes the embedded sync information or not. If the sync information is embedded in the pixel data stream, the additional sync signals, which are CVS and CHS, are not used. In this case, 656FCR1 and 656FCR2 register need to be configured. Otherwise, CVS and CHS are used for the vertical synchronous signal and the horizontal synchronous signal respectively. If the sync information is in the CVS and CHS, TV and VI bit of ICPCR1 register can be used to adjust the misaligned vertical (CVS) and horizontal sync (CHS).

When the format of input data is RGB565 or RGB555, it should be converted to YUV. Refer to CR2Y register on page 16-28.

PXCLK in Figure 16.1 is from a camera module. If a camera module needs the external clock input, CCKO can be used as input to one. Refer to PCK_CAM on page 23-10.

The POL bit of ICPCR1 register determines whether pixel data are latched at the rising edge or the falling edge of PXCLK. And CVS and CHS polarity are determined by the VSP and the HSP bit of ICPCR1 register respectively.

The geometric property of input image is determined by CEIS register (Figure 16.3).

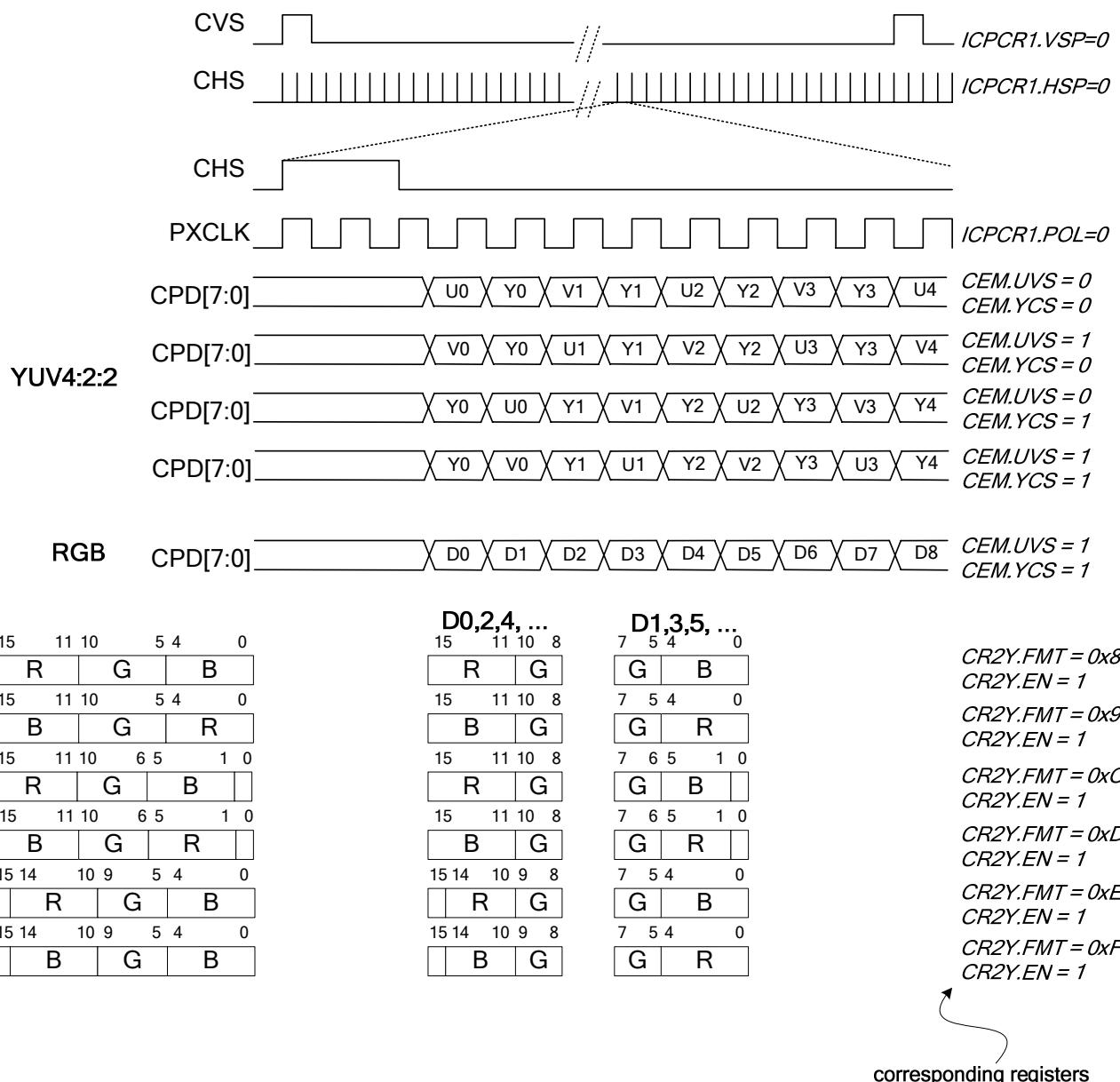


Figure 16.2 Input Data Format From the external camera module

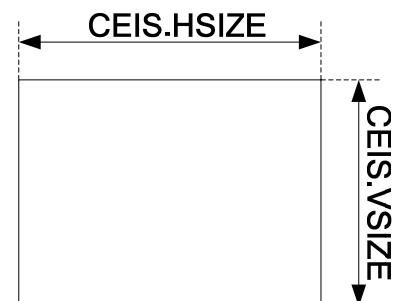


Figure 16.3 Input Image Size From the external camera module

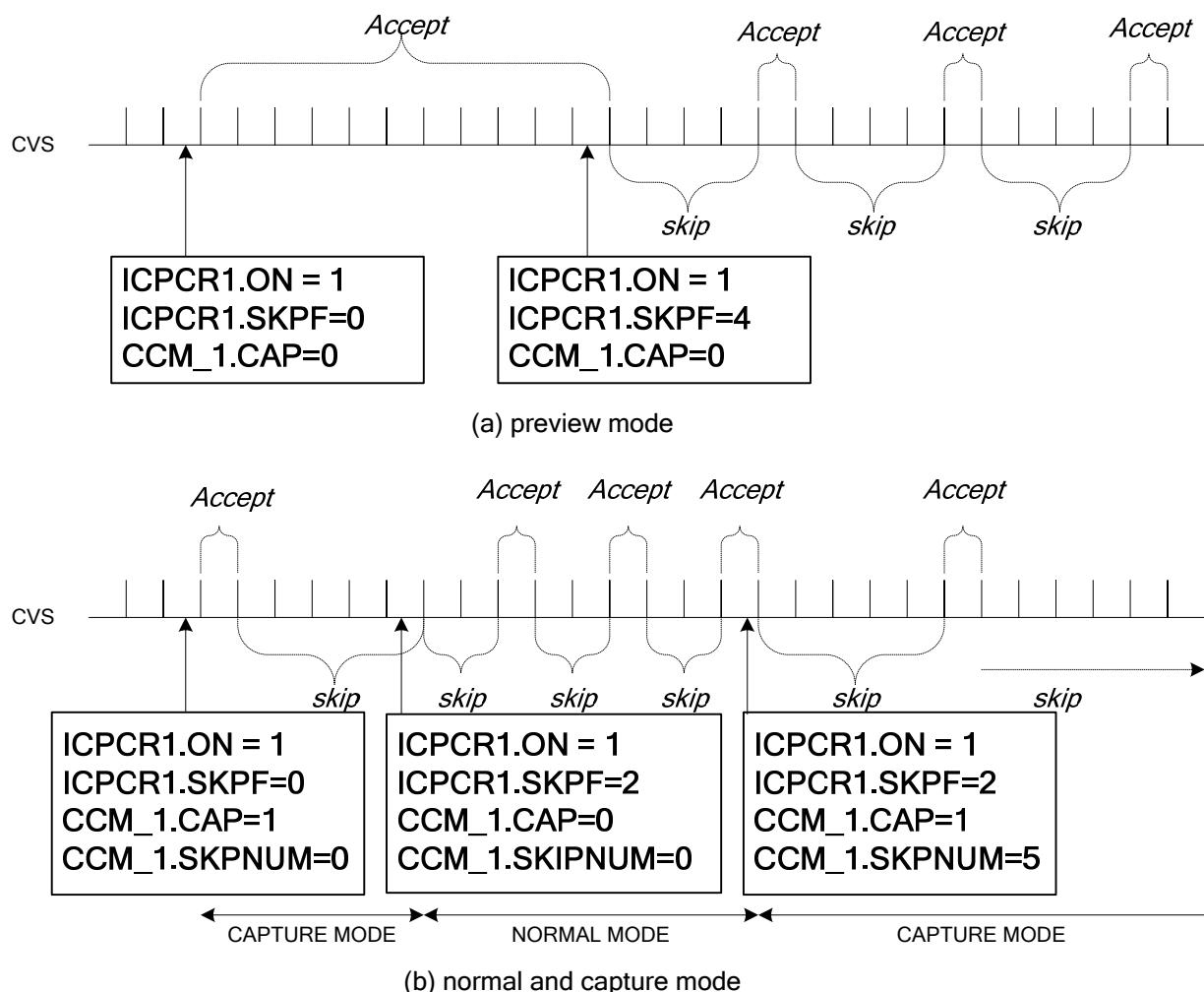


Figure 16.4 Accepting Frame Data from the external camera module

The CIF has two modes to accept one frame data from the external camera module. One is the preview mode. The other is the capture mode. The preview mode accepts the successive frame data from the external camera module. But, the capture mode accepts only one frame data.

Although the camera module sends the successive frame data to the CIF, the CIF can discard several frames before accepting a frame. They are called the skip frames. When the CIF is in the preview mode, the skip frames are determined by SKPF bits of ICPCR1 register. When the CIF is in the capture mode, the skip frames are determined by SKIPNUM bits of CCM_1 register. Refer to ICPCR1 register on page 16-9 and CCM_1 register on page 16-26. Figure 16.4 shows the difference between the preview mode and the capture mode.

The accepted frames are sent to the effector and their pixel format is YUV 4:2:2.

Effector

The effector supports YUV bias (YUV offset value), Inversion of Y value, strong C mode (x2 C value), Y clipping, color filter, sketch mode, embossing (positive and negative), gray, and sepia. Refer to the effector registers on page 16-31.

CIF Scaler

The TCC79XX provides the CIF scaler for scaling the image from the effector. The supported scaling ratio is from 1 : 4 (zoom up to 4 times) to 64 : 1 (zoom down to 64 times), and scaling step is 256. Maximum scaling image is 1600*1200*8 fps.

The output image of the effector becomes the CIF scaler input. Figure 16.5 shows relationship between them and how to specify them. Refer to CSSO and CSSS register on page 16-36. When the BPS bit of ICPCR1 register is set to 1, the effector output is not scaled by the CIF scaler.

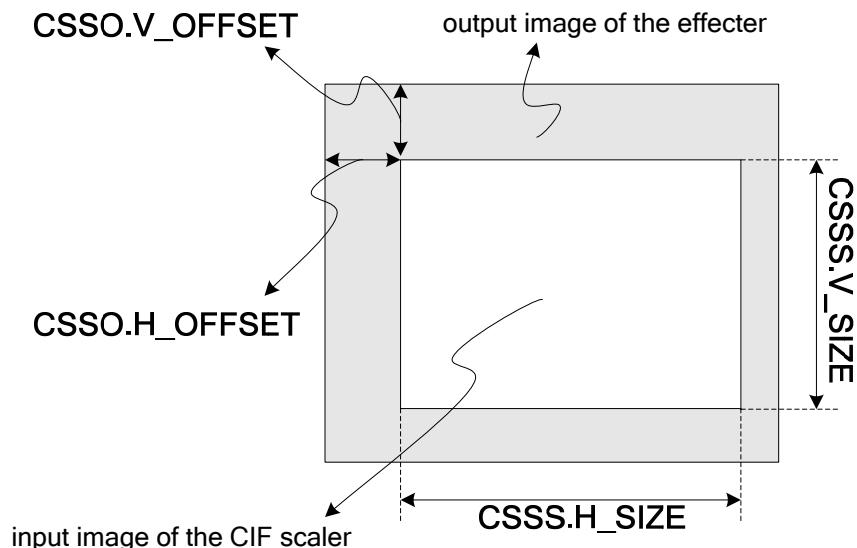


Figure 16.5 Output Image of the effector and Input Image of the CIF Scaler

Overlay

The CIF scaler output image (If the CIF scaler is disabled, it is equal to the effector output image) can be mixed with another image in the memory. This image in the memory is called the overlay image and the CIF scaler output image is called the background image. For this operation, the CIF has the alpha-blending and the chroma-keying function.

Figure 16.6 shows how to specify the geometric property of overlay image. Refer to OIS, OIW1, and OIW2 register on page 16-24. If the size of overlay image (not active overlay image) is equal to the size of background image, the OM bit of OCTRL1 should be set to 0 (full image overlay). Otherwise, it should be set to 1 (block image overlay).

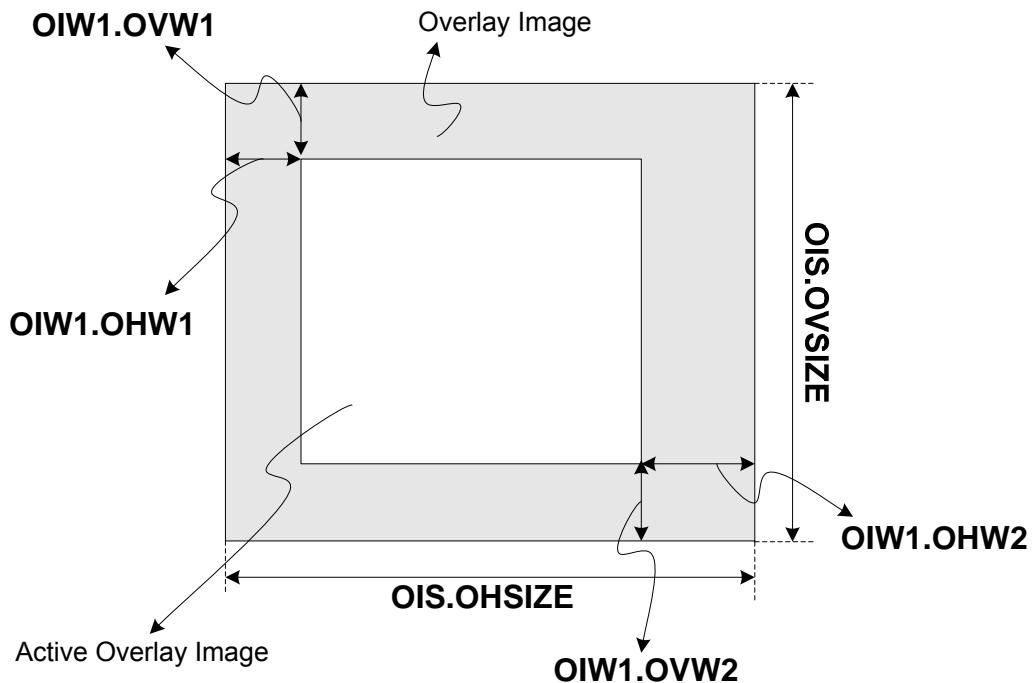


Figure 16.6 The Geometric Property of Overlay Image

Store to the memory

The image data which is from the camera module through effector and CIF scaler to overlay logic need to be stored to the memory. Figure 16.7 shows how to specify the image data to be stored. Refer to IIS, IIW1, and IIW2 register on page 16-13.

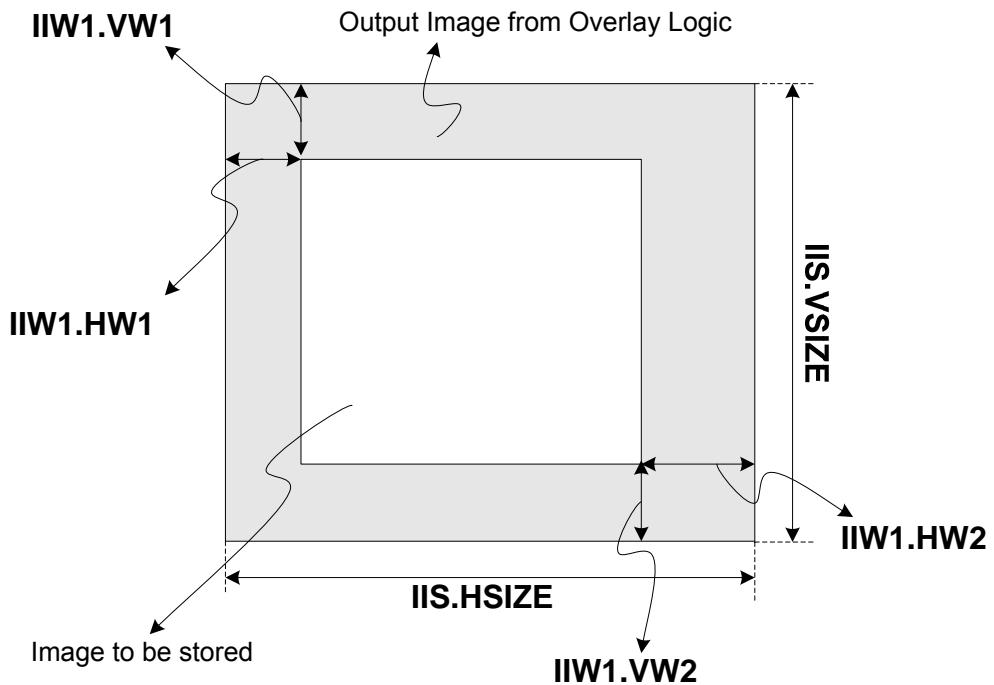


Figure 16.7 Image data to be stored to the memory

The image data from the overlay logic is YUV4:2:2. The image format to be stored can be YUV4:2:2SEQ0, YUV4:2:2, or YUV4:2:0. It is determined by the BP and the M420 bit of ICPCR1 register. Refer to the ICPCR1 register on page 16-9.

The CIF has two modes to store the image data. One is the frame mode and the other is the rolling mode. The frame mode needs the memory space to store a whole frame data. But, the rolling mode needs the memory space to store the part of a frame data. It is determined by the RLY, RLU, and RLV bits of CCM_1. Therefore, in the rolling mode, before the frame data are overwritten, they should be moved to another memory space.

CDCR2, CDCR3, and CDCR4 registers are the base addresses which the image data is stored to. They are for Y, U, and V image data respectively. Additionally, the end addresses are required in the rolling mode. They are determined by CDCR5, CDCR6, and CDCR7 register. Refer to these registers on page 16-14.

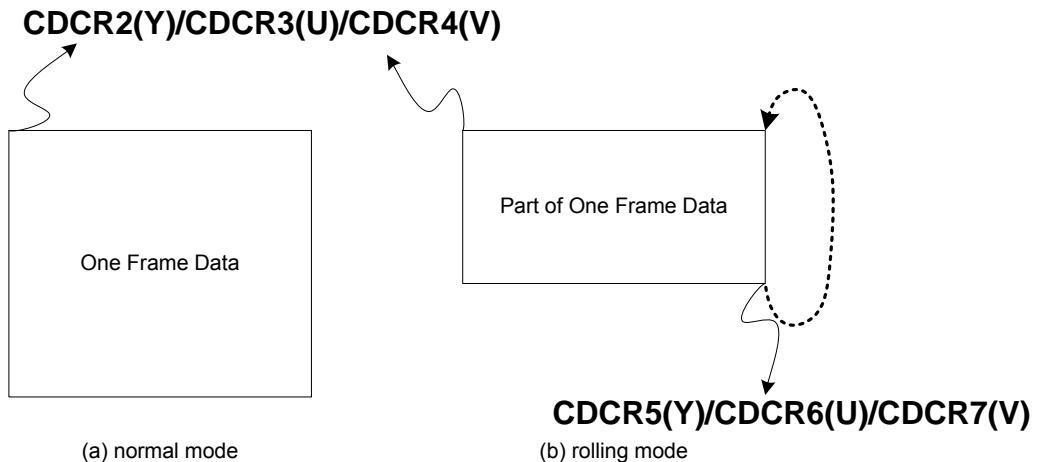


Figure 16.8 Two modes to store the image data

The CIF can inform the on-chip CPU its status during storing image data. When the CIF has stored a whole frame data to the memory, SOF of CIRQ register is set to 1. SCF and SOF of CIRQ register is the same function, but SCF is only available in the capture mode. In the rolling mode, ROLY, ROLU, and ROLV of CIRQ register represent whether the current address which is used to store the image data reaches the corresponding "end address" or not. ENS of CIRQ register is similar to ROLY, U, V but the current address is compared with CESR register. VIT of CIRQ register is set to 1 whenever the number of lines which the CIF receives data from the camera module and stores them to specified memory is equal to the multiple of 16*VCNT lines. If VCNT of CCM_2 register is set to 0, VIT bit is never set to 1. All these status bits can be the interrupt request source. Refer to CIRQ on page 16-18.

16.3 Camera Register Descriptions

Table 16.1 CIF Register Map (Base Address = 0xF0060000)

Name	Address	Type	Reset	Description
ICPCR1	0x00	W/R	0x00000000	Input Image Color/Pattern Configuration Register 1
656FCR1	0x04	W/R	0x06ff0000	CCIR656 Format Configuration Register 1
656FCR2	0x08	W/R	0x010b	CCIR656 Format Configuration Register 2
IIS	0x0C	W/R	0x00000000	Input Image Size
IIW1	0x10	W/R	0x00000000	Input Image Windowing 1
IIW2	0x14	W/R	0x00000000	Input Image Windowing 2
CDCR1	0x18	W/R	0x0003	DMA Configuration Register 1
CDCR2	0x1C	W/R	0x00000000	DMA Configuration Register 2
CDCR3	0x20	W/R	0x00000000	DMA Configuration Register 3
CDCR4	0x24	W/R	0x00000000	DMA Configuration Register 4
CDCR5	0x28	W/R	0x00000000	DMA Configuration Register 5
CDCR6	0x2C	W/R	0x00000000	DMA Configuration Register 6
CDCR7	0x30	W/R	0x00000000	DMA Configuration Register 7
FIFOSTATE	0x34	R	0x00000000	FIFO Status Register
CIRQ	0x38	W/R	0x00000000	Interrupt & Status register
OCTRL1	0x3C	W/R	0x37000000	Overlay Control 1
OCTRL2	0x40	W/R	0x00000000	Overlay Control 2
OCTRL3	0x44	W/R	0x00000000	Overlay Control 3
OCTRL4	0x48	W/R	0x00000000	Overlay Control 4
OIS	0x4C	W/R	0x00000000	Overlay Image Size
OIW1	0x50	W/R	0x00000000	Overlay Image Windowing 1
OIW2	0x54	W/R	0x00000000	Overlay Image Windowing 2
COBA	0x58	W/R	0x00000000	Overlay Base Address
CDS	0x5C	W/R	0x00000000	Camera Down Scaler
CCM1	0x60	W/R	0x00000000	Capture Mode Configuration 1
CCM2	0x64	W/R	0x00000000	Capture Mode Configuration 2
CESA	0x64	W/R	0x00000000	Point Encoding Start Address
CR2Y	0x6C	W/R	0x00000000	RGB2YUV Format converter Configuration
CCYA	0x70	R	-	Current Y Address
CCYU	0x74	R	-	Current U Address
CCYV	0x78	R	-	Current V Address
CCLC	0x7C	R		Current Line count
CCLC	0x7C	R		Current Line count

Input Image Color/Pattern Configuration Register 1 (ICPCR1)

0xF0060000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ON	-	TV	VI	0			BPS	0	POL	SKPF			M420		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BP	-	C656	CP	PF<1:0>	RGBM<1:0>	RGBBM<1:0>	CS<1:0>	0		BO	HSP	VSP			

ON [31]	On/Off on CIF
0	Disable CIF
1	Enable CIF
TV [29]	TV signal
0	CIF sync signal
1	TV sync signal
VI [28]	Vertical blank insert
0	Do not inset vertical blank
1	Insert vertical blank in 1 line
BPS [23]	Bypass Scaler
0	CIF scaler is used.
1	Bypass. CIF scaler is not used.
POL [21]	PXCLK Polarity
0	Rising edge
1	Falling edge.
SKPF [20:18]	Skip frame
0	Not-skipped
1 ~ 7	Number of frames to be skipped.
M420 [17:16]	YUV4:2:2 to YUV4:2:0
00	Not-Convert
10	Odd lines of U and V image are skipped.
11	Even lines of U and V image are skipped.
BP [15]	Bypass (Non-Separate)
Data format to be stored to memory	
0	If M420 bit is set to 0, the data format is YUV 4:2:2. Otherwise, the data format is YUV4:2:0.
1	YUV4:2:2SEQ0
C656 [13]	Convert 656 format
0	Disable
1	Enable
CP [12]	Color Pattern
0	It should be set to 0
PF [11:10]	Pattern Format
01	"01b" SHOULD BE WRITTEN TO THIS BIT.

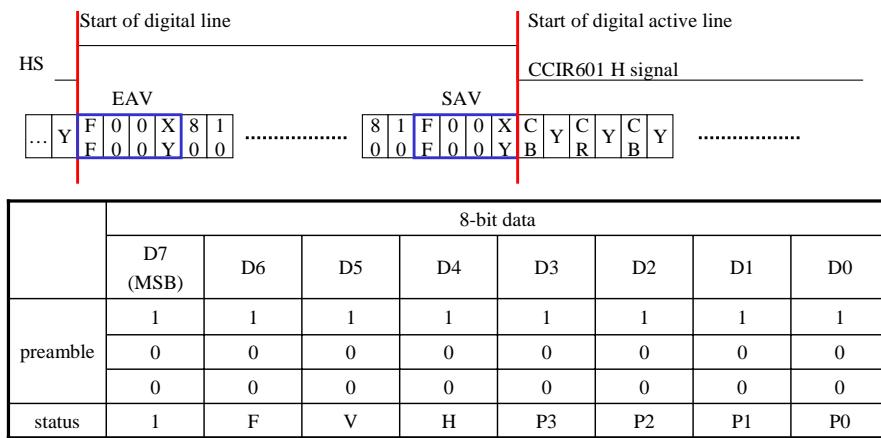
RGBM [9:8]	RGB Mode
00	It should be set to 0.
RGBBM [7:6]	RGB Bit Mode
00	It should be set to 0.
CS [5:4]	Color Sequence
00	It should be set to 0.
BO [2]	Bus Order
1	Switch the MSB/LSB 8bit bus. (Don't change)
HSP [1]	Horizontal Sync Polarity
0	Active low
1	Active high (default)
VSP [0]	Vertical Sync Polarity
0	Active low (default)
1	Active high Y

CCIR656 Format Configuration Register 1 (656FCR1)

0xF0060004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				PSL<1:0>				0				FPV<7:0>			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPV<7:0>				TPV<7:0>											

This register and next register (656FCR1 and 656FCR2) define the configuration of CCIR656.



- Status word define
 - F='0' for field 1, '1' for field 2 (interlace mode. If progressive, this value is '0')
 - V='1' during vertical blanking
 - H='0' at SAV, '1' at EAV
- Protection bits
 - P3=V xor H
 - P2=F xor H
 - P1=F xor V
 - P0=F xor V xor H

Figure 16.9 CCIR-656 Format Diagram

PSL [26:25]	Preamble and Status Location
00	The status word is located in the first byte of EAV & SAV
01	The status word is located in the second byte of EAV & SAV
10	The status word is located in the third byte of EAV & SAV
11	The status word is located in the forth byte of EAV & SAV

We must find the location of preamble and status for getting sync information. The total size of preamble and status is 4 bytes composed of 3 bytes of preamble and 1 byte of status. This register is used to find the location of status word.

FIELD	Description
FPV [23:16]	First preamble value Define the first preamble value. Default value is 0x00.
SPV [15:8]	Second preamble value Define the second preamble value. Default value is 0x00.
TPV [7:0]	Third preamble value Define the third preamble value. Default value is 0x00.

CCIR656 Format Configuration Register 2 (656FCR2) 0xF0060008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									HB<3:0>			0	VB<3:0>		

FIELD	Description
HB [8:5]	Horizontal blank In status word, it refers to the location of 'H' and H value at blanking. The MSB 3 bit means the location of 'H', the other bit means value at blanking. Default value is 0x09.
VB [3:0]	Vertical blank In status word, it refers to the location of 'V' and V value at blanking. The MSB 3 bit means the location of 'V', the other bit means value at blanking. Default value is 0x0B

Input Image Size (IIS)

0xF006000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HSIZE <15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VSIZE <15:0>															

FIELD	Description
HSIZE [31:16]	Horizontal size of input image
VSIZE [15:0]	Vertical size of input image

If the CIF scaler is used, this should be the same as CSDS register. Otherwise, this should be the same as CEIS register

Input Image Windowing1 (IIW1)

0xF0060010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HW1<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HW2<15:0>															

Input Image Windowing2 (IIW2)

0xF0060014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VW1<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VW2 <15:0>															

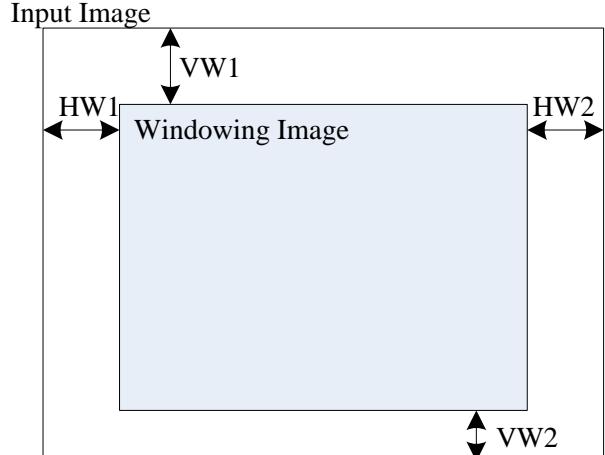


Figure 16.10 Input Image Windowing

Default value of HW1, HW2, VW1, and VW2 is 0.

CIF DMA Configuration Register 1 (CDCR1)

0xF0060018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															TM

Lock BS <1:0>

BS [1:0]

Preamble and Status Location

00	The DMA transfers the image data as 1 word to memory.
01	The DMA transfers the image data as 2 words to memory.
10	The DMA transfers the image data as 4 words to memory.
11	The DMA transfers the image data as 8 words to memory. (default)

LOCK [2]

Lock Transfer

0	Non-Lock (default)
1	Lock Transfer

TM [3]

Transfer Method

0	Burst Transfer (default)
1	INC Transfer

CIF DMA Configuration Register 2 (CDCR2)

0xF006001C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Input Image /Y(G) Channel Base Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Input Image / Y(G) Channel Base Address<15:0>															

FIELD

Description

CDCR2 [31:0] Input Image Base Address. / Y(G) channel base address

CIF DMA Configuration Register 3 (CDCR3)

0xF0060020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U(R) Channel Base Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U(R) Channel Base Address<15:0>															

FIELD

Description

CDCR3 [31:0] U(R) Channel Base Address.

CIF DMA Configuration Register 4 (CDCR4)

0xF0060024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
V(B) Channel Base Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V(B) Channel Base Address<15:0>															

FIELD

Description

CDCR4 [31:0] V(B) Channel Base Address

CIF DMA Configuration Register 5 (CDCR5)

0xF0060028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Input Image /Y(G) Channel End Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Input Image / Y(G) Channel End Address<15:0>															

FIELD	Description
CDCR5 [31:0]	Input Image End Address. / Y(G) channel end address This mode is operated, when rolling address Y is enabled.

CIF DMA Configuration Register 6 (CDCR6)

0xF006002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U(R) Channel End Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U(R) Channel End Address<15:0>															

FIELD	Description
CDCR6 [31:0]	U(R) Channel End Address This mode is operated, when rolling address U is enabled.

CIF DMA Configuration Register 7 (CDCR7)

0xF0060030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
V(B) Channel Base Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V(B) Channel Base Address<15:0>															

FIELD	Description
CDCR7 [31:0]	V(B) Channel End Address This mode is operated, when rolling address U is enabled.

FIFO States (FIFOSTATE)

0xF0060034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CLR	0	REO	REV
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	WEO	WEV	WEU	WEY	0	EO	EV	EU	EY	0	FO	FV	FU	FY	

CLR [21]		Clear FIFO states
1		When writing 1 to this bit, all status bits are cleared.
REO [19]		Overlay FIFO Read Error
0		Normal operation.
1		It represents that FIFO underrun is occurred.
REV [18]		Cr(B) Channel FIFO Read Error
0		Normal
1		It represents that FIFO underrun is occurred.
REU [17]		Cb(R) Channel FIFO Read Error
0		Normal
1		It represents that FIFO underrun is occurred.
REO [16]		Y(G) Channel FIFO Read Error
0		Normal
1		It represents that FIFO underrun is occurred.
WEO [13]		Overlay FIFO Write Error
0		Normal
1		It represents that FIFO overrun is occurred.
WEV [12]		Cr(B) Channel FIFO Write Error
0		Normal
1		It represents that FIFO overrun is occurred.
WEU [11]		Cb(R) Channel FIFO Write Error
0		Normal
1		It represents that FIFO overrun is occurred.
WEY [10]		Y(G) Channel FIFO Write Error
0		Normal
1		It represents that FIFO overrun is occurred.
EO [8]		Overlay FIFO Empty Signal
0		It represents that FIFO is not empty.
1		It represents that FIFO is empty.
EV [7]		V(B) Channel FIFO Empty Signal
0		It represents that FIFO is not empty.
1		It represents that FIFO is empty.

EU [6]	U(R) Channel FIFO Empty Signal
0	It represents that FIFO is not empty.
1	It represents that FIFO is empty.
EY [5]	Y(G) Channel FIFO Empty Signal
0	It represents that FIFO is not empty.
1	It represents that FIFO is empty.
FO [3]	Overlay FIFO Full Signal
0	It represents that FIFO is not full.
1	It represents that FIFO is full.
FV [2]	V(B) Channel FIFO Full Signal
0	It represents that FIFO is not full.
1	It represents that FIFO is full.
FU [1]	U(R) Channel FIFO Full Signal
0	It represents that FIFO is not full.
1	It represents that FIFO is full.
FY [0]	Y(G) Channel FIFO Full Signal
0	It represents that FIFO is not full.
1	It represents that FIFO is full.

CIF Interrupt Register (CIRQ)**0xF0060038**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IEN	URV	ITY	ICR	0	MVN	MVP	MVIT	MSE	MSF	MENS	MRLV	MRLU	MRLY	MSCF	MSOF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	VSS	0	VN	VP	VIT	SE	SF	ENS	ROLV	ROLU	ROLY	SCF	SOF

IEN [31]		Interrupt Enable
0		Interrupt disable
1		Interrupt enable
URV [30]		Update Register in VSYNC
0		All CIF registers are applied to CIF operation regardless of VSYNC.
1		All CIF registers are applied to CIF operation at rising edge of CVS signal.
ITY [29]		Interrupt Type.
1		"1" SHOULD BE WRITTEN TO THIS BIT.
ICR [28]		Interrupt Clear
1		When writing 1 to this bit, the CIF interrupt is cleared.
MVN [26]		Mask interrupt of VS negative edge
	Enable	
0		When VN bit is set to 1, the CIF interrupt request is issued.
1		Disable
MVP [25]		Mask interrupt of VS positive edge
	Enable	
0		When VP bit is set to 1, the CIF interrupt request is issued.
1		Disable
MVIT [24]		Mask interrupt of VCNT Interrupt
	Enable	
0		When VIT bit is set to 1, the CIF interrupt request is issued.
1		Disable
MSE [23]		Mask interrupt of Scaler Error
	Enable	
0		When SE bit is set to 1, the CIF interrupt request is issued.
1		Disable
MSF [22]		Mask interrupt of Scaler finish
	Enable	
0		When SF bit is set to 1, the CIF interrupt request is issued.
1		Disable

MENS [21]	Mask interrupt of Encoding start
	Enable
0	When ENS bit is set to 1, the CIF interrupt request is issued.
1	Disable
MRLV [20]	Mask interrupt of Rolling V address.
	Enable
0	When RLV bit is set to 1, the CIF interrupt request is issued.
1	Disable
MRLU [19]	Mask interrupt of Rolling U address.
	Enable
0	When RLU bit is set to 1, the CIF interrupt request is issued.
1	Disable
MRLY [18]	Mask interrupt of Rolling Y address.
	Enable
0	When RLY bit is set to 1, the CIF interrupt request is issued.
1	Disable
MSCF [17]	Mask interrupt of Capture frame.
	Enable
0	When SCF bit is set to 1, the CIF interrupt request is issued.
1	Disable
MSOF [16]	Mask interrupt of Stored one frame.
	Enable
0	When SOF bit is set to 1, the interrupt request is issued.
1	Disable
VSS [12]	Status of vertical sync.
0	Non- vertical sync blank area.
1	Vertical sync blank area.
VN [10]	VS negative.
0	-
1	It represents that CVS falling-edge is detected.
VP [9]	VS positive
0	-
1	It represents that CVS rising-edge is detected.

VIT [8]	VCNT Interrupt.
0	-
1	Refer to CCM_2 register on page 16-27.
SE [6]	Scaler Error.
0	-
1	It represents that CIF scaler has an error during scaling operation.
	When writing 1 to this bit, it is cleared.
SF [6]	Scaler Finish.
0	-
1	It represents that CIF scaler operation is completed.
	When writing 1 to this bit, it is cleared.
ENS [5]	Encoding start status.
0	-
1	Refer to CESA register on page 16-27
	When writing 1 to this bit, it is cleared.
ROLV [4]	Rolling V address status.
0	-
1	It represents that the current DMA V address reaches the DMA V rolling end address.
	When writing 1 to this bit, it is cleared.
ROLU [3]	Rolling U address status.
0	-
1	It represents that the current DMA U address reaches the DMA U rolling end address.
	When writing 1 to this bit, it is cleared.
ROLY [2]	Rolling Y address status.
0	-
1	It represents that the current DMA Y address reaches the DMA Y rolling end address.
	When writing 1 to this bit, it is cleared.

SCF [1]		Stored One frame for CAPTURE
0	-	It is only available when CAP bit of CCM_1 register is set to 1. Refer to CCM_1 register on page 16-26 about CAP bit.
1	-	It represents that the CIF has received one frame data from the camera module and stored them to the specified memory completely.
SOF [0]		Stored One frame
0	-	It represents that the CIF has received one frame data from the camera module and stored them to specified memory completely. It does not consider CAP bit.
1	-	

Overlay Control 1 (OCTRL1)

0xF006003C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			OCNT<4:0>				0				OM				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			OE	0	XR1	XR0	0	AP1<1:0>	AP0<1:0>	0	AEN	0	CEN		

OM[16]		Overlay Method
0	-	Full image overlay
1	-	Block image overlay

Full image overlay mode, overlay image size is equal to the input image size.

OE[12]		Overlay Enable
0	-	Disable
1	-	Enable
XR1[10]		XOR in AP1 is 3
0	-	XOR operation
1	-	100 %

When AP1 is 3 and CEN & AEN is 1, we select the 100% alpha value or XOR operation.

XR0 [9]		XOR in AP0 is 3
0	-	XOR operation
1	-	100 %

When AP0 is 3 and CEN & AEN is 1, we select the 100% alpha value or XOR operation

AP1 [7:6]		Alpha Value in alpha is 1
0	-	25 %
1	-	50 %
2	-	75 %
3	-	100 % or XOR operation (for XR value)

AP0 [5:4]		Alpha Value in alpha is 0
0	25 %	
1	50 %	
2	75 %	
3	100 % or XOR operation	

When RGB565 and AEN are set, alpha value depends on AP0 value.

AEN[2]		Alpha Enable
0	Disable	
1	Enable	

CEN[2]		Chroma key Enable
0	Disable	
1	Enable	

OCNT[29:24]		Overlay Count (FIFO)
n		

Overlay Control 2 (OCTRL2) 0xF0060040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved														CONV	RGB	MD

CONV[3]		Color Converter Enable
0	Disable	
1	Enable	

RGB[2:1]		RGB Mode
0	565RGB	
1	555RGB	
2	444RGB	
3	332RGB	

MD[0]		Color Mode
0	YUV color	
1	RGB color	

Overlay Control 3 – key value (OCTRL3)

0x F0060044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								KEYR<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYG<7:0>								KEYB<7:0>							

KEYR[23:16]	Chroma-key value R(U)
n	Chrome-key value in R(U) channel Default value is 0x00
KEYG[15:8]	Chroma-key value G(Y)
n	Chrome-key value in G(Y) channel. Default value is 0x00
KEYB[7:0]	Chroma-key value B(V)
n	Chrome-key value in B(V) channel. Default value is 0x00

Overlay Control 4 – mask key value (OCTRL4)

0x F0060048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								MKEYR<7:0>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MKEYG<7:0>								MKEYB<7:0>							

MKEYR[23:16]	Mask Chroma-key value R(U)
N	Chrome-key value in R(U) channel Default value is 0x00
MKEYG[15:8]	Mask Chroma-key value G(Y)
N	Chrome-key value in G(Y) channel. Default value is 0x00
MKEYB[7:0]	Mask Chroma-key value B(V)
n	Chrome-key value in B(V) channel. Default value is 0x00

Overlay Image Size (OIS)

0x F006004C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OHSIZE<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSIZE<15:0>															

FIELD

Description

OHSIZE [31:16]	Horizontal size of overlay image Default value is 0x0280. (decimal is 640)
OVSIZE [15:0]	Vertical size of overlay image Default value is 0x01E0 (decimal is 480)

Overlay Image Windowing 1 (OIW1)

0x F0060050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OHW1<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OHW2<15:0>															

Overlay Image Windowing 2 (OIW2)

0x F0060054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OVW1<15:0>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVW2 <15:0>															

Input Image

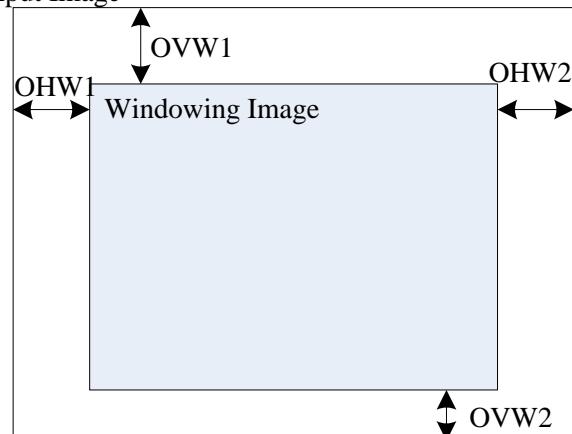


Figure 16.11 Overlay Image Windowing

CIF Overlay Base Address (COBA)

0x F0060058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Overlay Image Base Address<31:16>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Overlay Image Base Address<15:0>															

FIELD	Description
COBA [31:0]	Overlay Image Base Address. Default value is 0x20100000.

CIF Down Scaler (CDS)

0x F006005C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										SFH<1:0>		SFV<1:0>		0	

SFH [5:4]	Horizontal Scale Factor
0	1/1 down scale
1	1/2 down scale
2	1/4 down scale
3	1/8 down scale

SFV [3:2]	Vertical Scale Factor
0	1/1 down scale
1	1/2 down scale
2	1/4 down scale
3	1/8 down scale

SEN	Scale Enable
0	Disable
1	Enable

CIF Capture mode_1 (CCM_1)

0x F0060060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENCNUM				ROLNUMV				ROLNUMU				ROLNUMY			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CB	EIT	UES	SKIPNUM				RLV	RLU	RLY	CAP	

ENCNUM
[31:28]

Encode INT number (using CAP mode)

0~15 Refer to EIT bit.

ROLNUMV
[27:24]

Rolling Number in V (using CAP mode)

0~15 The number of times which V address is rolled while capture mode and rolling mode are enabled and one frame data is stored to the memory.

ROLNUMU
[23:20]

Rolling Number in U (using CAP mode)

0~15 The number of times which U address is rolled while capture mode and rolling mode are enabled and one frame data is stored to the memory.

ROLNUMY
[19:16]

Rolling Number in Y (using CAP mode)

0~15 The number of times which Y address is rolled while capture mode and rolling mode are enabled and one frame data is stored to the memory.

CB [10]

Capture Busy

0 -

1 It represents that frame data is storing to the memory during the capture mode.

EIT [9]

Encoding INT count

0 The ENS bit of CIRQ register is set to 1 when the current DMA Y address reaches the address to be specified in the CESA register. But, it can be occurred the only one time during storing one frame data.

1 The ENS bit of CIRQ register is set to 1 whenever the current DMA Y address reaches the address to be specified in the CESA register. ENCNUM represents the number of times which ENS bit is set to 1.

UES [8]

Using Encoding Start Address

0 The use of CESA register is disabled.

1 The use of CESA register is enabled. Refer to CESA register on page 16-27

SKIPNUM [7:4]

Skip frame number (using CAP mode)

0~15 It specifies the number of skip frames during capture mode.

RLV[3]

Rolling mode for V image

0 Disable (frame mode)

1 Enable

RLU[2]	Rolling Mode for U image
0	Disable (frame mode)
1	Enable
<hr/>	
RLY[1]	Rolling Mode for Y image
0	Disable (frame mode)
1	Enable
<hr/>	
CAP [0]	Capture mode
0	Preview mode
1	Capture mode

CIF CAPTURE MODE_2 (CCM_2)

0x F0060064

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								VCNT				VEN			

FIELD	Description (Using CAP mode)
VIT	VIT bit is set to 1 whenever the number of lines which the CIF receives data from the camera module and stores them to specified memory is equal to the multiple of 16*VCNT lines. If VCNT is set to 0, VIT bit is never set to 1.
VCNT[7:4]	VCNT can be 0 ~ 7.
	For example, when VCNT = 3, the VIT bit is set to 1 whenever a number of line to be stored is equal to a multiple of 48 line, which is 48, 96, 144, etc.

VEN [0]	VCNT enable (Using CAP mode)
0	Disable
1	Enable (VCNT is used for setting VIT bit)

CIF Encoding Start Address (CESA)

0x F0060068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Encoding Start Address [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Encoding Start Address [15:0]															

FIELD	Description
	Default value is 0x20100000.
CESA[31:0]	This is compared with the current DMA Y address. When the current DMA Y address reaches the address to be specified in the CESA register, the ENS bit of CIRQ is set to 1.
	When the EIT bit of CCM_1 register is equal to 0, it can be occurred the only one time during storing one frame data. This operation is enabled when UES bit of CCM_1 register is set to 1. Refer to CCM_1 register on page 16-26.

CIF R2Y Configuration (CR2Y)

0xF006006C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

FMT

EN

FMT [4:1]				Input format
0	0	0	0	16bit 565RGB (RGB sequence)
0	0	0	1	16bit 565RGB (BGR sequence)
0	1	0	0	16bit 555RGB (RGB-garbage)
0	1	0	1	16bit 555RGB (BGR-garbage)
0	1	1	0	16bit 555RGB (garbage-RGB)
0	1	1	1	16bit 555RGB (garbage-BGR)
1	0	0	0	8bit 565RGB (RGB sequence)
1	0	0	1	8bit 565RGB (BGR sequence)
1	1	0	0	8bit 555RGB (RGB-garbage)
1	1	0	1	8bit 555RGB (BGR-garbage)
1	1	1	0	8bit 555RGB (garbage-RGB)
1	1	1	1	8bit 555RGB (garbage-BGR)

EN [0]		R2Y enable
0	Disable	
1	Enable	

CIF Current Y Address (CCYA)

0xF0060070

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Current Y address [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Current U address [15:0]															

FIELD	Description
CCYA[31:0]	Current Y Address.

CIF Current U Address (CCUA)

0xF0060074

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Current U address [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Current U address [15:0]															

FIELD	Description
CCUA[31:0]	Current U Address

CIF Current V Address (CCVA)

0x F0060078

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Current V address [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Current V address [15:0]															

FIELD	Description
CCVA[31:0]	Current V Address

CIF Current Line Count (CCLC)

0x F006007C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Line count [15:0]															

FIELD	Description
LCNT[15:0]	Current Line Count

16.4 Effector Register Descriptions

Table 16.2 Effect Register Map (Base Address = 0xF0060100)

Name	Address	Type	Reset	Description
CEM	0x00	W/R	0x00000000	Effect mode register
CSUV	0x04	W/R	0x00000000	Sepia UV setting
CCS	0x08	W/R	0x00000000	Color selection register
CHFC	0x0C	W/R	0x00000000	H-filter coefficent0
CST	0x10	W/R	0x00000000	Sketch threshold register
CCT	0x14	W/R	0x00000000	Clamp threshold register
CBR	0x18	W/R	0x00000000	BIAS register
CEIS	0x1C	W/R	0x00000000	Image size register
-	0x40	W/R	0x00000000	Reserved
CISA1	0x44	W/R	0x00000000	Source address in Y channel
CISA2	0x48	W/R	0x00000000	Source address in U channel
CISA3	0x4C	W/R	0x00000000	Source address in V channel
CISS	0x50	W/R	0x00000000	Source image size
CISO	0x54	W/R	0x00000000	Source image offset
CIDS	0x58	W/R	0x00000000	Destination image size
CIS	0x5C	W/R	0x00000000	Target scale

CIF Effect Mode (CEM)

0x F0060100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UVS	VB	UB	YB	YCS	IVY	STC	YCL	CS	SKT	EMM	EMB	NEGA	GRAY	SEPI	NOR

UVS [15]		UV Swap
0		U-V-U-V sequence
1		V-U-V-U sequence
VB [14]		V Bias (V channel value offset)
0		Disable
1		Enable
UB [13]		U Bias (U channel value offset)
0		Disable
1		Enable
YB [12]		Y Bias (Y channel value offset)
0		Disable
1		Enable
YCS [11]		YC Swap
U(or V)-Y-V(or U)-Y sequence		
0		UV sequence is determined by UVS bit.
Y-U(or V)-Y-V(or U) sequence		
1		UV sequence is determined by UVS bit.
IVY[10]		Invert Y
0		Disable
1		Enable
STC[9]		Strong C
0		Disable
1		Enable
YCL[8]		Y Clamp (Y value clipping)
0		Disable
1		Enable
CS[7]		C Select (Color filter)
0		Disable
1		Enable (color filter)
SKT[6]		Sketch Enable
0		Disable
1		Enable

EMM[5]		Emboss Mode
0	Positive emboss	
1	Negative emboss	
EMB[4]		Emboss
0	Disable	
1	Enable	
NEGA[3]		Negative mode
0	Disable	
1	Enable	
GRAY[2]		Gray mode
0	Disable	
1	Enable	
SEPI[1]		Sepia mode
0	Disable	
1	Enable	
NOR[0]		Normal mode
0	Effect mode (effecter is enabled)	
1	Normal mode (effecter is disabled)	

CIF Sepia UV (CSUV)

0x F0060104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEPIA_U								SEPIA_V							

FIELD	Description
SEPIA_U[15:8]	U channel threshold value for sepia
SEPIA_V[7:0]	V channel threshold value for sepia

CIF Color Selection (CCS)

0x F0060108

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U start								U end							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V start								V end							

FIELD	Description
U_Start[31:24]	Color filter range start point of U channel
U_End[23:16]	Color filter range end point of V channel
V_Start[15:8]	Color filter range start point of U channel
V_End[7:0]	Color filter range end point of V channel

CIF H Filter Coeff. (CHFC)

0x F006010C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								Coeff 0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Coeff 1								Coeff 2							

FIELD	Description
Coeff0[23:16]	Horizontal filter coefficient0 for emboss or sketch.
Coeff1[15:8]	Horizontal filter coefficient1 for emboss or sketch.
Coeff2[7:0]	Horizontal filter coefficient2 for emboss or sketch

CIF Sketch Threshold. (CST)

0x F0060110

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sketch Threshold															

FIELD	Description
Sketch[7:0]	Sketch threshold

CIF Clamp Threshold. (CCT)

0x F0060114

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Clamp Threshold															

FIELD	Description
Clamp[7:0]	Clamp threshold

CIF Bias Register (CBR)

0x F0060118

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Y bias															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U bias															

FIELD	Description
Y_BIAS[23:16]	Y value offset
U_BIAS[15:8]	U value offset
V_BIAS[7:0]	V value offset

CIF Effect Image Size (CEIS)

0x F006011C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HSIZE															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VSIZE															

FIELD	Description
HSIZE[26:16]	Horizontal size of input image
VSIZE[10:0]	Vertical size of input image

16.5 Scaler Register Descriptions

Table 16.3 Scaler Register Map (Base Address = 0xF0060200)

Name	Address	Type	Reset	Description
CSC	0x00	W/R	0x00000000	Scaler configuration
CSSF	0x04	W/R	0x00000000	Scale factor
CSSO	0x08	W/R	0x00000000	Image offset
CSSS	0x0C	W/R	0x00000000	Source image size
CSDS	0x10	W/R	0x00000000	Destination image size

CIF Scaler CTRL (CSC) 0x F0060200															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

EN [0]		Scaler Enable
0	Disable	
1	Enable	

CIF Scaler SCALE Factor(CSSF) 0x F0060204															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		HSCALE													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		VSCALE													

FIELD	Description
HSCALE [29:16]	Horizontal scale factor
VSCALE [13:0]	Vertical scale factor

$$\text{HSCALE} = \text{SRC_HSIZE} * 256 / \text{DST_HSIZE}$$

$$\text{VSCALE} = \text{SRC_VSIZE} * 256 / \text{DST_VSIZE}$$

CIF Scaler Source Offset (CSSO) 0x F0060208															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		H_OFFSET													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		V_OFFSET													

FIELD	Description
H_OFFSET [27:16]	Horizontal offset
V_OFFSET [11:0]	Vertical offset

CIF Scaler Source Size (CSSS)

0x F006020C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			0												H_SIZE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			0												V_SIZE

FIELD	Description
H_SIZE [27:16]	Horizontal size in source image
V_SIZE [11:0]	Vertical size in source image

CIF Scaler DST_Size (CSDS)

0x F0060210

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			0												H_SIZE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			0												V_SIZE

FIELD	Description
H_SIZE [27:16]	Horizontal size in destination image
V_SIZE [11:0]	Vertical size in destination image

17 GRAPHIC ENGINE

17.1 Overview

Figure 17.1 shows the whole block diagram of Graphic Engine and its corresponding registers

As can be seen below, Graphic Engine supports Rotating / Mirroring image, bitwise ROP, Alpha-Blending, Arithmetic or Format Converting function. These functions can simultaneously work. Graphic engine has two different source channels and one destination channel. In other words, one output image can be created with two Input images by using an appropriate function of Graphic Engine. Moreover, specific local region operation is available since Graphic Engine supports Source Image Offset, Destination Image Offset and Window Offset.

[TOP Architecture]

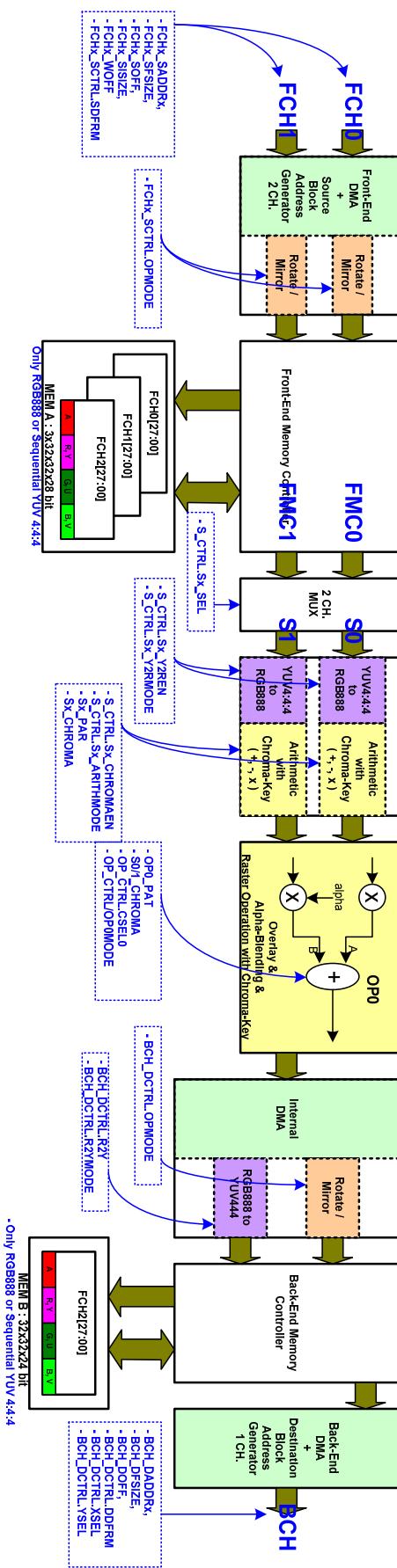


Figure 17.1 Graphic Engine Block Diagram (FCH2 is not available in the TCC79XX)

17.2 Features

The features of Graphic Engine are as follows

- Input / Output Image Format
 - RGB Format : RGB444, RGB454, RGB555, RGB565, RGB666, RGB888
 - Alpha-RGB Format : α RGB444, α RGB454, α RGB555, α RGB666, α RGB888
 - Sequential YUV : YUV4:4:4, YUV4:2:2
 - Separate YUV : YUV4:4:4, YUV4:4:0, YUV4:2:2, YUV4:2:0, YUV4:1:1, YUV4:1:0.
- Rotate/Mirror Operation
 - 2 Channel Source 90/180/270 Rotate with Window Offset and Source Offset
 - 1 Channel Destination 90/180/270 Rotate with Destination Offset
 - 2 Channel Source Vertical/Horizontal/Vertical & Horizontal Mirror with Window Offset and Source Offset
 - 1 Channel Destination Vertical/Horizontal/Vertical & Horizontal Mirror with Destination Offset
- Arithmetic Operation
 - 2 Channel Arithmetic Operation with Chroma-Key Function
 - Arithmetic Function : Bypass, Fill, Inv, Add, Sub, Multiply
- Raster Operation & Alpha-Blending & Overlay
 - 2 Channel to 1Channel ROP Operation with Chroma-Key & Window Offset
 - 2 Channel to 1Channel Alpha Blending with Chroma-Key & Window Offset
 - ROP Function : 16 Type
 - Alpha-Blending Operation by Fixed Alpha-Value
 - Alpha-Blending Operation by Alpha-RGB Input Image Format
- RGB to YUV Format Converting
 - 2 Independent Source RGB to YUV Format Converting
 - 4 RGB to YUV Format Converting Type
- YUV to RGB Format Converting
 - 1 Destination YUV to RGB Format Converting
 - 4 YUV to RGB Format Converting Type

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RGB444(12bit)												R1[3:0]	G1[3:0]	B1[3:0]				R0[3:0]	G0[3:0]	B0[3:0]														
Alpha-RGB444(16bit)												?1[3:0]	R1[3:0]	G1[3:0]	B1[3:0]	?0[3:0]		R0[3:0]	G0[3:0]	B0[3:0]														
RGB454(13bit)												R1[3:0]	G1[4:0]	B1[3:0]				R0[3:0]	G0[4:0]	B0[3:0]														
Alpha-RGB454(16bit)												?1[2:0]	R1[3:0]	G1[4:0]	B1[3:0]	?0[2:0]		R0[2:0]	G0[4:0]	B0[3:0]														
RGB555(15bit)												R1[4:0]	G1[4:0]	B1[4:0]				R0[4:0]	G0[4:0]	B0[4:0]														
Alpha-RGB555(16bit)												?1	R1[4:0]	G1[4:0]	B1[4:0]	?0	R0[4:0]	G0[4:0]	B0[4:0]															
RGB565(16bit)												R1[4:0]	G1[5:0]	B1[4:0]				R0[4:0]	G0[5:0]	B0[4:0]														
RGB666(18bit)																	R[5:0]	G[5:0]	B[5:0]															
Alpha-RGB666(22bit)																	?1[3:0]	R[5:0]	G[5:0]	B[5:0]														
RGB888(24bit)																	R[7:0]	G[7:0]	B[7:0]															
Alpha-RGB888(28bit)																	?1[3:0]	R[7:0]	G[7:0]	B[7:0]														
Sequential YUV 4:4:4																	Y[7:0]		U[7:0]		V[7:0]													
	31	..	24	23	..	16	15	..	8	7	..	0																						
Sequential YUV 4:2:2												V0[7:0]	Y1[7:0]	U0[7:0]	Y0[7:0]																			
Separate YUV 4:4:4 (Y)												Y3[7:0]	Y2[7:0]	Y1[7:0]	Y0[7:0]																			
Separate YUV 4:4:4 (U)												U3[7:0]	U2[7:0]	U1[7:0]	U0[7:0]																			
Separate YUV 4:4:4 (V)												V3[7:0]	V2[7:0]	V1[7:0]	V0[7:0]																			
Separate YUV 4:4:0 (Y)												Y3[7:0]	Y2[7:0]	Y1[7:0]	Y0[7:0]																			
Separate YUV 4:4:0 (U)												U3[7:0]	U2[7:0]	U1[7:0]	U0[7:0]																			
Separate YUV 4:4:0 (V)												V3[7:0]	V2[7:0]	V1[7:0]	V0[7:0]																			
Separate YUV 4:2:2 (Y)												Y3[7:0]	Y2[7:0]	Y1[7:0]	Y0[7:0]																			
Separate YUV 4:2:2 (U)												U6[7:0]	U4[7:0]	U2[7:0]	U0[7:0]																			
Separate YUV 4:2:2 (V)												V6[7:0]	V4[7:0]	V2[7:0]	V0[7:0]																			
Separate YUV 4:2:0 (Y)												Y3[7:0]	Y2[7:0]	Y1[7:0]	Y0[7:0]																			
Separate YUV 4:2:0 (U)												U6[7:0]	U4[7:0]	U2[7:0]	U0[7:0]																			
Separate YUV 4:2:0 (V)												V6[7:0]	V4[7:0]	V2[7:0]	V0[7:0]																			
Separate YUV 4:1:1 (Y)												Y3[7:0]	Y2[7:0]	Y1[7:0]	Y0[7:0]																			
Separate YUV 4:1:1 (U)												U12[7:0]	U8[7:0]	U4[7:0]	U0[7:0]																			
Separate YUV 4:1:1 (V)												V12[7:0]	V8[7:0]	V4[7:0]	V0[7:0]																			
Separate YUV 4:1:0 (Y)												Y12[7:0]	Y8[7:0]	Y4[7:0]	Y0[7:0]																			
Separate YUV 4:1:0 (U)												U12[7:0]	U8[7:0]	U4[7:0]	U0[7:0]																			
Separate YUV 4:1:0 (V)												V12[7:0]	V8[7:0]	V4[7:0]	V0[7:0]																			
YUV 4:4:4	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			
YUV 4:4:0	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○			
YUV 4:2:2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			
YUV 4:2:0	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○			
YUV 4:1:1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			
YUV 4:1:0	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○			

Figure 17.2 Supported Data Format of Graphic Engine

17.3 Register Description

Table 17.1 Graphic Engine Register Map (Base Address = 0xF6000000)

Name	Address	Type	Reset	Description
FCH0_SADDR0	0x00	R/W	0x00000000	Front-End Channel 0 Source Address 0
FCH0_SADDR1	0x04	R/W	0x00000000	Front-End Channel 0 Source Address 1
FCH0_SADDR2	0x08	R/W	0x00000000	Front-End Channel 0 Source Address 2
FCH0_SFSIZE	0x0C	R/W	0x00000000	Front-End Channel 0 Source Frame Pixel Size
FCH0_SOFF	0x10	R/W	0x00000000	Front-End Channel 0 Source Pixel Offset
FCH0_SISIZE	0x14	R/W	0x00000000	Front-End Channel 0 Source Image Pixel Size
FCH0_WOFF	0x18	R/W	0x00000000	Front-End Channel 0 Window Pixel Offset
FCH0_SCTRL	0x1C	R/W	0x00000000	Front-End Channel 0 Control
<hr/>				
FCH1_SADDR0	0x20	R/W	0x00000000	Front-End Channel 1 Source Address 0
FCH1_SADDR1	0x24	R/W	0x00000000	Front-End Channel 1 Source Address 1
FCH1_SADDR2	0x28	R/W	0x00000000	Front-End Channel 1 Source Address 2
FCH1_SFSIZE	0x2C	R/W	0x00000000	Front-End Channel 1 Source Frame Pixel Size
FCH1_SOFF	0x30	R/W	0x00000000	Front-End Channel 1 Source Pixel Offset
FCH1_SISIZE	0x34	R/W	0x00000000	Front-End Channel 1 Source Image Pixel Size
FCH1_WOFF	0x38	R/W	0x00000000	Front-End Channel 1 Window Pixel Offset
FCH1_SCTRL	0x3C	R/W	0x00000000	Front-End Channel 1 Control
<hr/>				
0x40-0x5F				
- Reserved				
S0_CHROMA	0x60	R/W	0x00000000	Source 0 Chroma-Key Parameter
S0_PAR	0x64	R/W	0x00000000	Source 0 Arithmetic Parameter
S1_CHROMA	0x68	R/W	0x00000000	Source 1 Chroma-Key Parameter
S1_PAR	0x6C	R/W	0x00000000	Source 1 Arithmetic Parameter
<hr/>				
0x70-0x77				
S_CTRL	0x78	R/W	0x00000000	Source Control Register
<hr/>				
- Reserved				
OP0_PAT	0x80	R/W	0x00000000	Source Operator 0 Pattern
<hr/>				
- Reserved				
OP_CTRL	0x88	R/W	0x00000000	Source Operation Control Register
<hr/>				
- Reserved				
BCH_DADDR0	0x90	R/W	0x00000000	Back-End Channel Destination Address 0
BCH_DADDR1	0x94	R/W	0x00000000	Back-End Channel Destination Address 1
BCH_DADDR2	0x98	R/W	0x00000000	Back-End Channel Destination Address 2
BCH_DFSIZE	0x9C	R/W	0x00000000	Back-End Channel Destination Frame Pixel Size
BCH_DOFF	0xA0	R/W	0x00000000	Back-End Channel Destination Pixel Offset
BCH_DCTRL	0xA4	R/W	0x00000000	Back-End Channel Control
<hr/>				
- 0xA8 – 0xAF Reserved				
GE_CTRL	0xB0	R/W	0x00000000	Graphic Engine Control
GE_IREQ	0xB4	R/W	0x00000000	Graphic Engine Interrupt Request

Front-End Channel 0 Source Address 0(FCH0_SADDR0)

0xF6000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR0[15:02]															

Reserved

Bit	Name	RW	Reset	Description
31-2	SADDR0	RW	-	It determines the Y image base address of channel 0 image when the data format of channel 0 is YUV format. But, in case the data format of channel 0 is a sequential YUV or an RGB, it determines the base address of the whole image.

The data format of FCH0 image is determined by the FCH0_SCTRL register on page 17-8.

Front-End Channel 0 Source Address 1 (FCH0_SADDR1)

0xF6000004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR1[15:02]															

Reserved

Bit	Name	RW	Reset	Description
31-2	SADDR1	RW	-	It determines the U image base address of front-end channel 0 (FCH0) when the data format of channel 0 is YUV format. But, in case the data format of FCH0 image is a sequential YUV or an RGB, it is not used.

Front-End Channel 0 Source Address 2 (FCH0_SADDR2)

0xF6000008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR2[15:02]															

Reserved

Bit	Name	RW	Reset	Description
31-2	SADDR1	RW	-	It determines the V image base address of FCH0 when the data format of FCH0 is YUV format. But, in case the data format of FCH0 image is a sequential YUV or an RGB, it is not used.

Front-End Channel 0 Source Frame Pixel Size (FCH0_SFSIZE)

0xF600000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

SFSIZE_Y[11:00]

SFSIZE_X[11:00]

SFSIZE_Y	[27:16]	Source Frame Pixel Size Y
N (0 ~ 4095)	height of FCH0 image by pixel units	
SFSIZE_X	[11:00]	Source Frame Pixel Size X
N (0 ~ 4095)	width of FCH0 image by pixel units	

Front-End Channel 0 Source Pixel Offset (FCH0_SOFF)

0xF6000010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				SOFF_Y[11:00]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SOFF_X[11:00]											

SOFF_Y	[27:16]	Source Pixel Offset Y
N (0 ~ 4095)		N =Y Axis Source Pixel Offset
SOFF_X	[11:00]	Source Pixel Offset X
N (0 ~ 4095)		N =X Axis Source Pixel Offset

Front-End Channel 0 Source Image Pixel Size (FCH0_SISIZE)

0xF6000014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				SISIZE_Y[11:00]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SISIZE_X[11:00]											

SISIZE_Y	[27:16]	Source Image Pixel Size Y
N (0 ~ 4095)		N =Y Axis Source Image Pixel Size
SISIZE_X	[11:00]	Source Image Pixel Size X
N (0 ~ 4095)		N =X Axis Source Image Pixel Size

Front-End Channel 0 Window Pixel Offset (FCH0_WOFF)

0xF6000018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				WOFF_Y[11:00]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WOFF_X[11:00]											

WOFF_Y	[27:16]	Window Pixel Offset Y
N (0 ~ 4095)		N =Y Axis Window Pixel Offset
WOFF_X	[11:00]	Window Pixel Offset X
N (0 ~ 4095)		N =X Axis Window Pixel Offset

When FCH0 Image is selected as Source 0 by SCTRL.S0SEL register, Window Offset should definitely be '0'.

Front-End Channel 0 Control (FCH0_SCTRL)

0xF600001C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				OPMODE[2:0]				0		ZF	SDFRM[04:00]				

OPMODE	[10:08]	Operation Mode
00x		Data Copy
010		Horizontal Mirror
011		Vertical Mirror
100		Vertical & Horizontal Mirror
101		90 Degree Rotate
110		180 Degree Rotate
111		270 Degree Rotate

ZF	[05]	Zero Fill
0		Zero Fill Mode Enable
1		MSB Fill Mode Enable

The above applies only when Data Format is RGB Format, not RGB888 / Alpha-RGB888.

RGB444	1 0 1 0	0 0 1 0	1 0 1 0
alpha RGB444			
ZERO FILL	1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0		
MSB FILL	1 0 1 0 1 1 1 1 0 0 1 0 0 0 0 0 1 0 1 0 1 0 1 1 1 1		

RGB454	0 0 1 0	1 0 1 0 1	1 0 1 0
alpha RGB454			
ZERO FILL	0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0		
MSB FILL	0 0 1 0 0 0 0 0 1 0 1 0 1 1 1 1 0 1 0 1 0 1 1 1 1		

RGB555	0 0 1 0 1	1 0 1 0 1	1 0 1 0 0
alpha RGB555			
ZERO FILL	0 0 1 0 1 0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0		
MSB FILL	0 0 1 0 1 0 0 0 1 0 1 0 1 1 1 1 0 1 0 1 0 1 1 1 1		

RGB666	0 0 1 0 1	1 0 1 0 1 0	1 0 1 0 0
alpha RGB666			
ZERO FILL	0 0 1 0 1 0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0		
MSB FILL	0 0 1 0 1 0 0 0 1 0 1 0 1 1 1 1 0 1 0 1 0 1 1 1 1		

RGB666	0 0 1 0 1 1	1 0 1 0 1 0	1 0 1 0 0 1
alpha RGB666			
ZERO FILL	0 0 1 0 1 1 0 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 0 1 0 0		
MSB FILL	0 0 1 0 1 1 0 0 1 0 1 0 1 1 1 1 0 1 0 1 0 1 1 1 1		

Figure 17.3 Example: Input Source Image Data to RGB888 Conversion

SDFRM	[04:00]	Source Data Format
00000		Source Data Format = Separate YUV 4:4:4
00001		Source Data Format = Separate YUV 4:4:0
00010		Source Data Format = Separate YUV 4:2:2
00011		Source Data Format = Separate YUV 4:2:0
00100		Source Data Format = Separate YUV 4:1:1
00101		Source Data Format = Separate YUV 4:1:0
00110		Source Data Format = Sequential YUV 4:4:4
00111		Source Data Format = Sequential YUV 4:2:2
01000		Source Data Format = RGB444
01001		Source Data Format = Alpha-RGB444
01010		Source Data Format = RGB454
01011		Source Data Format = Alpha-RGB454
01100		Source Data Format = RGB555
01101		Source Data Format = Alpha-RGB555
0111x		Source Data Format = RGB565
10000		Source Data Format = RGB666
10001		Source Data Format = Alpha-RGB666
10010		Source Data Format = RGB888
else		Source Data Format = Alpha-RGB888

Front-End Channel 1 Source Address 0 (FCH1_SADDR0)

0xF6000020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR0[15:02]															

Source Image Format is Separate YUV X:X:X, the value of the address is Source Address of Y region.

Source Image Format is not Separate YUV X:X:X, the value of the address is Source Address.

Front-End Channel 1 Source Address 1 (FCH1_SADDR1)

0xF6000024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR1[15:02]															

In case Source Image Format is Separate YUV X:X:X, the value of the address is Source Address of U region.

In case Source Image Format is not Separate YUV X:X:X, the value of the address is not used.

Front-End Channel 1 Source Address 2 (FCH1_SADDR2)

0xF6000028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR2[15:02]															

In case Source Image Format is Separate YUV X:X:X, the value of the address is Source Address of V region.

In case Source Image Format is not Separate YUV X:X:X, the value of the address is not used.

Front-End Channel 1 Source Frame Pixel Size (FCH1_SFSIZE)

0xF600002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SFSIZE_Y[11:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFSIZE_X[11:00]															

SFSIZE_Y [27:16] Source Frame Pixel Size Y

N (0 ~ 4095) N =Y Axis Source Frame Pixel Size

SFSIZE_X [11:00] Source Frame Pixel Size X

N (0 ~ 4095) N =X Axis Source Frame Pixel Size

Front-End Channel 1 Source Pixel Offset (FCH1_SOFF)

0xF6000030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SOFF_Y[11:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOFF_X[11:00]															

SOFF_Y	[27:16]	Source Pixel Offset Y
--------	---------	-----------------------

N (0 ~ 4095) N =Y Axis Source Pixel Offset

SOFF_X	[11:00]	Source Pixel Offset X
--------	---------	-----------------------

N (0 ~ 4095) N =X Axis Source Pixel Offset

Front-End Channel 1 Source Image Pixel Size (FCH1_SISIZE)

0xF6000034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SISIZE_Y[11:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SISIZE_X[11:00]															

SISIZE_Y	[27:16]	Source Image Pixel Size Y
----------	---------	---------------------------

N (0 ~ 4095) N =Y Axis Source Image Pixel Size

SISIZE_X	[11:00]	Source Image Pixel Size X
----------	---------	---------------------------

N (0 ~ 4095) N =X Axis Source Image Pixel Size

Front-End Channel 1 Window Pixel Offset (FCH1_WOFF)

0xF6000038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WOFF_Y[11:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WOFF_X[11:00]															

WOFF_Y	[27:16]	Window Pixel Offset Y
--------	---------	-----------------------

N (0 ~ 4095) N =Y Axis Window Pixel Offset

WOFF_X	[11:00]	Window Pixel Offset X
--------	---------	-----------------------

N (0 ~ 4095) N =X Axis Window Pixel Offset

When FCH1 Image is selected as Source 0 by SCTRL.S0SEL register, Window Offset should definitely be '0'.

Front-End Channel 1 Control (FCH1_SCTRL) 0xF600003C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCH1_OPMODE[2:0]								0	ZF	FCH1_SDfrm[04:00]					

OPMODE	[10:08]	Operation Mode
00x		Data Copy
010		Horizontal Mirror
011		Vertical Mirror
100		Vertical & Horizontal Mirror
101		90 Degree Rotate
110		180 Degree Rotate
111		270 Degree Rotate

ZF	[05]	Zero Fill
0		Zero Fill Mode Enable
1		MSB Fill Mode Enable

The above applies only when Data Format is RGB Format, not RGB888 / Alpha-RGB888. Refer to Figure 17.3.

SDfrm	[04:00]	Source Data Format
00000		Source Data Format = Separate YUV 4:4:4
00001		Source Data Format = Separate YUV 4:4:0
00010		Source Data Format = Separate YUV 4:2:2
00011		Source Data Format = Separate YUV 4:2:0
00100		Source Data Format = Separate YUV 4:1:1
00101		Source Data Format = Separate YUV 4:1:0
00110		Source Data Format = Sequential YUV 4:4:4
00111		Source Data Format = Sequential YUV 4:2:2
01000		Source Data Format = RGB444
01001		Source Data Format = Alpha-RGB444
01010		Source Data Format = RGB454
01011		Source Data Format = Alpha-RGB454
01100		Source Data Format = RGB555
01101		Source Data Format = Alpha-RGB555
0111x		Source Data Format = RGB565
10000		Source Data Format = RGB666
10001		Source Data Format = Alpha-RGB666
10010		Source Data Format = RGB888
else		Source Data Format = Alpha-RGB888

Source 0 Chroma-Key Parameter (S0_CHROMA)

0xF6000060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHROMA_RY[07:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHROMA_GU[07:00]								CHROMA_BV[07:00]							

CHROMA_RY	[23:16]	Chroma-Key Value
n	R or Y Chroma-Key Value	
CHROMA_GU	[15:08]	Chroma-Key Value
n	G or U Chroma-Key Value	
CHROMA_BV	[07:00]	Chroma-Key Value
n	B or V Chroma-Key Value	

Chroma-Key Value should be set to Sequential YUV4:4:4 or RGB888 irrespective of Input Source Format.

Writing should be done in RGB888 Format when Input Format is RGBxxx. When Input Format is YUVxxx and YUVtoRGB Converter is disabled, YUV444 Format is used for Writing. If input format is YUVxxx and YUVtoRGB Converter is enabled, writing is done in RGB888 Format.

S0_CHROMA Register is effective only when S_CTRL.S0_CHROMAEN is enabled and OP_CTRL.C0/1_SEL Register is 2'b01.

Source 0 Arithmetic Parameter (S0_PAR)

0xF6000064

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PAR_RY[07:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAR_GU[07:00]								PAR_BV[07:00]							

PAR_RY	[23:16]	Parameter Value for Arithmetic Operation
n	R or Y Parameter Value for Arithmetic Operation.	
PAR_GU	[15:08]	Parameter Value for Arithmetic Operation
N	G or U Parameter Value for Arithmetic Operation.	
PAR_BV	[07:00]	Parameter Value for Arithmetic Operation
n	B or V Parameter Value for Arithmetic Operation.	

Parameter Value should be set to Sequential YUV4:4:4 or RGB888 irrespective of Input Source Format.

Writing should be done in RGB888 Format when Input Format is RGBxxx. When Input Format is YUVxxx and YUVtoRGB Converter is disabled, YUV444 Format is used for Writing. If input format is YUVxxx and YUVtoRGB Converter is enabled, writing is done in RGB888 Format.

S0_PAR Register is effective only when SCTRL.S0ARITHMODE is Fill, Add, Sub and Multiply.

Source 1 Chroma-Key Parameter (S1_CHROMA) 0xF6000068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHROMA_RY[07:00]								CHROMA_RY[07:00]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHROMA_GU[07:00]								CHROMA_BV[07:00]							

CHROMA_RY [23:16]	Chroma-Key Value
n	R or Y Chroma-Key Value
CHROMA_GU [15:08]	Chroma-Key Value
n	G or U Chroma-Key Value
CHROMA_BV [07:00]	Chroma-Key Value
n	B or V Chroma-Key Value

Chroma-Key Value should be set to Sequential YUV4:4:4 or RGB888 irrespective of Input Source Format.

Writing should be done in RGB888 Format when Input Format is RGBxxx. When Input Format is YUVxxx and YUVtoRGB Converter is disabled, YUV444 Format is used for Writing. If input format is YUVxxx and YUVtoRGB Converter is enabled, writing is done in RGB888 Format.

S1_CHROMA Register is effective only when S_CTRL.S1_CHROMAEN is enabled and OP_CTRL.C0/1_SEL Register is 2'b10.

Source 1 Arithmetic Parameter (S1_PAR) 0xF600006C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PAR_RY[07:00]								PAR_RY[07:00]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAR_GU[07:00]								PAR_BV[07:00]							

PAR_RY [23:16]	Parameter Value for Arithmetic Operation
n	R or Y Parameter Value for Arithmetic Operation.
PAR_GU [15:08]	Parameter Value for Arithmetic Operation
N	G or U Parameter Value for Arithmetic Operation.
PAR_BV [07:00]	Parameter Value for Arithmetic Operation
n	B or V Parameter Value for Arithmetic Operation.

Parameter Value should be set to Sequential YUV4:4:4 or RGB888 irrespective of Input Source Format.

Writing should be done in RGB888 Format when Input Format is RGBxxx. When Input Format is YUVxxx and YUVtoRGB Converter is disabled, YUV444 Format is used for Writing. If input format is YUVxxx and YUVtoRGB Converter is enabled, writing is done in RGB888 Format.

S1_PAR Register is effective only when SCTRL.S1ARITHMODE is Fill, Add, Sub and Multiply.

Source Control (S_CTRL)

0xF6000078

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							S1ARITHMODE[2:0]			S0ARITHMODE[2:0]			SBZ ²⁷	S0/1_Y2REN	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			S0/1_Y2RMODE[01:00]				SBZ	S0/1_CHROMA EN		SBZ		S1SEL[01:00]		S0SEL[01:00]	

S1_ARITHMODE [24:22]		Source 1 Arithmetic Mode		
00x		Bypass Mode		
010		Fill Mode		
011		Inverter Mode		
100		Add Mode		
101		Substrate Type A Mode		
110		Substrate Type B Mode		
111		Multiplier Mode		

S0_ARITHMODE [21:19]		Source 0 Arithmetic Mode		
00x		Bypass Mode		
010		Fill Mode		
011		Inverter Mode		
100		Add Mode		
101		Substrate Type A Mode		
110		Substrate Type B Mode		
111		Multiplier Mode		

Arithmetic Fnc.	Equation	A	B
BYPASS	$Y = A$	Source Image	Do not Used
FILL	$Y = B$	Source Image	S0/1_PAR
INV	$Y = 255 - A$	Source Image	Do not Used
ADD	$Y = A + B$	Source Image	S0/1_PAR
SUBA	$Y = A - B$	Source Image	S0/1_PAR
SUBB	$Y = B - A$	Source Image	S0/1_PAR
	$Y = A \times B$		
MUL	Mantissa = B[07:06] Fraction = B[05:00]	Source Image	S0/1_PAR

Since Arithmetic Operation is done after YUVtoRGB Converting Operation, operation is working in RGB888 or YUV4:4:4 Format irrespective of Input Source Format. Therefore, writing should be done in RGB888 or YUV4:4:4 format.

If Chroma-Key Value and Selected Source Data are matched when Chroma-Key is enabled, BYPASS Operation is carried out.

²⁷ Should Be Zero

S1_Y2REN	[17]	Source 1 YUV to RGB Converter Enable
0	YUV to RGB Converter Disable	
1	YUV to RGB Converter Enable	

To enable YUV to RGB Converter, Data Format of Source 1 decided by SCTL.S1_SEL[02:00] should be YUVxxx.

S0_Y2REN	[16]	Source 0 YUV to RGB Converter Enable
0	YUV to RGB Converter Disable	
1	YUV to RGB Converter Enable	

To enable YUV to RGB Converter, Data Format of Source 0 decided by SCTL.S0_SEL[02:00] should be YUVxxx.

S1_Y2RMODE	[12:11]	Source 1 YUVtoRGB Convert Type
00	YUVtoRGB Converter Type0	
01	YUVtoRGB Converter Type1	
10	YUVtoRGB Converter Type2	
11	YUVtoRGB Converter Type3	

S0_Y2RMODE	[10:09]	Source 0 YUVtoRGB Convert Type
00	YUVtoRGB Converter Type0	
01	YUVtoRGB Converter Type1	
10	YUVtoRGB Converter Type2	
11	YUVtoRGB Converter Type3	

YUV4:4:4 to RGB888 Converter Type

- YUVtoRGB Format Converter Type 0
 $R = Y + 1.371 (V - 128)$
 $G = Y - 0.336 (U - 128) - 0.698 (V - 128)$
 $B = Y + 1.732 (U - 128)$
- YUVtoRGB Format Converter Type 1
 $R = 1.164 (Y - 16) + 1.159 (V - 128)$
 $G = 1.164 (Y - 16) - 0.391 (U - 128) - 0.813 (V - 128)$
 $B = 1.164 (Y - 16) + 2.018 (U - 128)$
- YUVtoRGB Format Converter Type 2
 $R = Y + 1.540 (V - 128)$
 $G = Y - 0.183 (U - 128) - 0.459 (V - 128)$
 $B = Y + 1.816 (U - 128)$
- YUVtoRGB Format Converter Type 3
 $R = 1.164 (Y - 16) + 1.793 (V - 128)$
 $G = 1.164 (Y - 16) - 0.213 (U - 128) - 0.534 (V - 128)$
 $B = 1.164 (Y - 16) + 2.115 (U - 128)$

S1_CHROMAEN	[7]	Source 1 Chroma-Key Enable for Arithmetic
0	Chroma-Key Disable for Arithmetic	
1	Chroma-Key Enable for Arithmetic	
S0_CHROMAEN	[6]	Source 0 Chroma-Key Enable for Arithmetic
0	Chroma-Key Disable for Arithmetic	
1	Chroma-Key Enable for Arithmetic	
S1_SEL	[03:02]	Source 1 Selection
00	Source 1 Disable	
01	Source 1 = Front-End Channel 0	
10	Source 1 = Front-End Channel 1	
11	Reserved	
S0_SEL	[01:00]	Source 0 Selection
00	Source 0 Disable	
01	Source 0 = Front-End Channel 0	
10	Source 0 = Front-End Channel 1	
11	Reserved	

Source 0 should select whatever image. In other words, if there is only one Input Source Image in either FCH0 or FCH1, the Input Source Image should be selected to Source 0.

If there are 2 Input Source Images in two of FCH0 and FCH1, the larger Input Source Image should be selected to S0 and the other should be selected to S1.

Source Operator 0 Pattern (OP0_PAT)															0xF6000080
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															PAT_RY[07:00]
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAT_GU[07:00]															PAT_BV[07:00]

ALPHA [27:24]	Source Operator 0 Alpha Value
n	Alpha Value for Alpha-Blending Operation of Source Operator 0.

OP0_PAR.ALPHA Register is effective only when OPCTRL.OP0MODE is Alpha-Blending Type 0.

PAT_RY [23:16]	Source Operator 0 Pattern Value
n	R or Y Pattern Value for Raster Operation of Source Operator 0

PAT_GU [15:08]	Source Operator 0 Pattern Value
n	G or U Pattern Value for Raster Operation of Source Operator 0

PAT_BV [07:00]	Source Operator 0 Pattern Value
n	B or V Pattern Value for Raster Operation of Source Operator 0

Source Operation Control (OP_CTRL)

0xF6000088

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											SBZ	SBZ	SBO ²⁸	SBO	SBZ
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CSEL0[1:0]

OP0_MODE[04:00]

OP1_MODE [20:16]		DESCRIPTION
00110		00110b should be written to OP1_MODE. (reset value= 00000b)
CSEL0 [06:05]		
00	Chroma-Key Operation Disable of Operator 0	
01	Chroma-Key Source = Source 0, Chroma-Key Value = S0_CHROMA	
10	Chroma-Key Source = Source 1, Chroma-Key Value = S1_CHROMA	
11	Reserved.	
OP0_MODE [04:00]		Operation 0 Mode
00000	Blackness	Result = 0
00001	Merged Copy	Result = P & A
00010	Merged Paint	Result = ~ A B
00011	Pattern Copy	Result = P
00100	Pattern Inverter	Result = P ^ B
00101	Pattern Paint	Result = (P ~ A) B
00110	Source Copy	Result = A
00111	Source Inverter	Result = A ^ B
01000	Source Paint	Result = A B
01001	Source AND	Result = A & B
01010	Source Erase	Result = A & ~ B
01011	Not Source Copy	Result = ~ A
01100	Not Source Erase	Result = ~ (A B)
01101	Destination Copy	Result = B
01110	Destination Inverter	Result = ~ B
01111	Whiteness	Result = 1
10xxx	Alpha-Blending Type 0	Alpha-Blending with Alpha-Value
11xxx	Alpha-Blending Type 1	Alpha-Blending with Alpha-Value of Alpha-RGB

Raster
Operation²⁸ Should Be One

Raster Operation & Alpha-Blending of Operator 0

Mode	Equation	A	B	P	ALPHA	CHROMA
Blackness	$Y = 0$	Source 0	Not Used.	Not Used.	Not Used.	S1_CHROMA
Merged Copy	$Y = P \& A$	Source 0	Not Used.	PAT0[23:00]	Not Used.	S1_CHROMA
Merged Paint	$Y = \sim A \mid B$	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Pattern Copy	$Y = P$	Source 0	Not Used.	PAT0[23:00]	Not Used.	S1_CHROMA
Pattern Inverter	$Y = P \wedge B$	Source 0	Source 1	PAT0[23:00]	Not Used.	S1_CHROMA
Pattern Paint	$Y = (P \mid \sim A) \mid B$	Source 0	Source 1	PAT0[23:00]	Not Used.	S1_CHROMA
Source Copy	$Y = A$	Source 0	Not Used.	Not Used.	Not Used.	S1_CHROMA
Source Inverter	$Y = A \wedge B$	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Source Paint	$Y = A \mid B$	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Source AND	$Y = A \& B$	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Source Erase	$Y = A \& \sim B$	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Not Source Copy	$Y = \sim A$	Source 0	Not Used.	Not Used.	Not Used.	S1_CHROMA
Not Source Erase	$Y = \sim (A \mid B)$	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Destination Copy	$Y = B$	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Destination Inverter	$Y = \sim B$	Source 0	Source 1	Not Used.	Not Used.	S1_CHROMA
Whiteness	$Y = 1$	Source 0	Not Used.	Not Used.	Not Used.	S1_CHROMA
Alpha-Blending Type 0	$Y = \text{ALPHA} * B + (1 - \text{ALPHA}) * A$	Source 0	Source 1	Not Used.	OP0_ALPHA	S1_CHROMA
Alpha-Blending Type 1	$Y = \text{ALPHA} * B + (1 - \text{ALPHA}) * A$	Source 0	Source 1	Not Used.	Alpha of Source 1	S1_CHROMA

In Alpha-Blending Type 0, OP0_ALPHA of OP_PAT0 Register is used as Alpha Value.

In Alpha-Blending Type 1, Alpha Value of Source 1 Data is used as Alpha Value. Therefore, in this case, Source Data Format should definitely be Alpha-RGB.

Back-End Destination Address 0 (BCH_DADDR0)

0xF6000090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCH_DADDR0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_DADDR0[15:02]															

Destination Image Format is Separate YUV X:X:X, the value of the address is Source Address of Y region.

Destination Image Format is not Separate YUV X:X:X, the value of the address is Source Address.

Back-End Destination Address 1 (BCH_DADDR1)

0xF6000094

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCH_DADDR1[31:16]															
9	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_DADDR1[15:02]															

In case Destination Image Format is Separate YUV X:X:X, the value of the address is Source Address of U region.

In case Destination Image Format is not Separate YUV X:X:X, the value of the address is not used.

Back-End Destination Address 2 (BCH_DADDR2)

0xF6000098

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCH_DADDR2[31:16]															
9	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_DADDR2[15:02]															

In case Destination Image Format is Separate YUV X:X:X, the value of the address is Source Address of V region.

In case Destination Image Format is not Separate YUV X:X:X, the value of the address is not used.

Back-End Channel Destination Frame Pixel Size (BCH_DFSIZE)

0xF600009C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DFSIZE_Y[11:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFSIZE_X[11:00]															

DFSIZE_Y [27:16] Destination Frame Pixel Size Y

N (0 ~ 4095) N = Y Axis Destination Frame Pixel Size

DFSIZE_X [11:00] Destination Frame Pixel Size X

N (0 ~ 4095) N = X Axis Destination Frame Pixel Size

Back-End Channel Destination Pixel Offset (BCH_DOFF) **0xF60000A0**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DOFF_Y[11:00]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOFF_X[11:00]															

DOFF_Y	[27:16]	Destination Pixel Offset Y
N (0 ~ 4095)	N =Y Axis Destination Pixel Offset	
DOFF_X	[11:00]	Destination Pixel Offset X
N (0 ~ 4095)	N =X Axis Destination Pixel Offset	

Destination Image Size & Frame Size & Offset Limitation

BCH_DCTRL.DDFRM	Destination Frame Size / Image Size / Offset Limitation
RGB454	No Limitation
RGB565	No Limitation
RGB666	No Limitation
RGB888	No Limitation
Separate YUV 4:4:4	No Limitation
Separate YUV 4:4:0	X axis = No Limitation, Y axis = multiples of 2
Separate YUV 4:2:2	X axis = multiples of 2, Y axis = No Limitation
Separate YUV 4:2:0	X axis = multiples of 2, Y axis = multiples of 2
Separate YUV 4:1:1	X axis = multiples of 4, Y axis = No Limitation
Separate YUV 4:1:0	X axis = multiples of 4, Y axis = multiples of 2
Sequential YUV 4:4:4	No Limitation
Sequential YUV 4:2:2	X axis = multiples of 2, Y axis = No Limitation

Destination Image Size is decided by both Source 0 size controlled by S0_CTRL and Back-End Destination Operation Mode controlled by BCH_CTRL.

Back-End Channel Control (BCH_DCTRL)

0xF60000A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													YSEL	XSEL[01:00]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R2Y	R2YMODE				OPMODE[02:00]				0				DDFRM[04:00]		

YSEL		YUV4:4:4 to YUVx:x:x Y Control
n		N = 0 or 1
XSEL		YUV4:4:4 to YUVx:x:x X Control
n		N = 0 ~ 3

Since inside Graphic Engine, only RGB888 and Sequential YUV4:4:4: exist as Data Format, if Destination Data Format is RGBxxx, RGB888 is converted to RGBxxx. If Destination Data Format is Sequential / Separate YUVxxx, Sequential YUV4:4:4 is converted to Sequential / Separate YUVxxx.

In this case, when Sequential YUV4:4:4 is converted to Sequential / Separate YUVxxx, BCH_CTRL.X/YSEL Register is reflected.

R2Y			Destination Format Converter Control
0			RGB to YUV Converter Disable
1			RGB to YUV Converter Enable
R2YMODE		[14:13]	RGBtoYUV Converter Type
0			RGBtoYUV Converter Type0
1			RGBtoYUV Converter Type1
2			RGBtoYUV Converter Type2
3			RGBtoYUV Converter Type3

RGB to YUV Converter Type

- RGBtoYUV Format Converter Type 0

$$Y = 0.299 R + 0.587 G + 0.114 B$$

$$U = -0.172 R - 0.339 G + 0.511 B + 128$$

$$V = 0.511 R - 0.428 G - 0.083 B + 128$$

- RGBtoYUV Format Converter Type 1

$$Y = 0.257 R + 0.504 G + 0.098 B + 16$$

$$U = -0.148 R - 0.291 G + 0.439 B + 128$$

$$V = 0.439 R - 0.368 G - 0.071 B + 128$$

- RGBtoYUV Format Converter Type 2

$$Y = 0.213 R + 0.715 G + 0.072 B$$

$$U = -0.117 R - 0.394 G + 0.511 B + 128$$

$$V = 0.511 R - 0.464 G - 0.047 B + 128$$

- RGBtoYUV Format Converter Type 3

$$Y = 0.183 R + 0.614 G + 0.062 B + 16$$

$$U = -0.101 R - 0.338 G + 0.439 B + 128$$

$$V = 0.439 R - 0.399 G - 0.040 B + 128$$

OPMODE	[10:08]	Operation Mode of Back-End DMA
00x	Data Copy	
010	Horizontal Mirror	
011	Vertical Mirror	
100	Vertical & Horizontal Mirror	
101	90 Degree Rotate	
110	180 Degree Rotate	
111	270 Degree Rotate	

DDFRM	[04:00]	Destination Data Format of Back-End DMA
00000		Destination Data Format = Separate YUV 4:4:4
00001		Destination Data Format = Separate YUV 4:4:0
00010		Destination Data Format = Separate YUV 4:2:2
00011		Destination Data Format = Separate YUV 4:2:0
00100		Destination Data Format = Separate YUV 4:1:1
00101		Destination Data Format = Separate YUV 4:1:0
00110		Destination Data Format = Sequential YUV 4:4:4
00111		Destination Data Format = Sequential YUV 4:2:2
01000		Destination Data Format = RGB444
01001		Destination Data Format = Alpha-RGB444
01010		Destination Data Format = RGB454
01011		Destination Data Format = Alpha-RGB454
01100		Destination Data Format = RGB555
01101		Destination Data Format = Alpha-RGB555
01110		Destination Data Format = RGB565
01111		Destination Data Format = RGB565
10000		Destination Data Format = RGB666
10001		Destination Data Format = Alpha-RGB666
10010		Destination Data Format = RGB888
else		Destination Data Format = Alpha-RGB888

Graphic Engine Control (GE_CTRL)

0xF60000B0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
															IEN	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
															EN[02:00]	

IEN	Graphic Engine Interrupt Enable
0	Interrupt Disable
1	Interrupt Enable

EN	[02:00]	Graphic Engine Enable
000	Graphic Engine Disable	
001	Graphic Engine Enable With Front-End Channel 0	
010	Graphic Engine Enable With Front-End Channel 1	
011	Reserved	
100	Graphic Engine Enable With Front-End Channel 0,1	
101	Reserved	
110	Reserved	
111	Reserved	

After all transfers complete, EN[02:00] Register is automatically cleared to 3'b000.

EN Register should be set last after all Control Registers are set.

Graphic Engine Interrupt Request Control (GE_IREQ)

0xF60000B4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
															FLG	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
															IRQ	

FLG	Graphic Engine Flag Bit
0 (Read)	Transfer of Graphic Engine is Not Completed.
0 (Write)	Flag Bit Clear
1	Transfer of Graphic Engine is Completed.

IRQ	Graphic Engine Interrupt Request
0 (Read)	Interrupt Request is not occurred.
0 (Write)	Interrupt Request is cleared.
1	Interrupt Request is occurred.

17.4 Graphic Engine Operation

17.4.1 Source Rotate / Mirror and Destination Rotate / Mirror Operation.

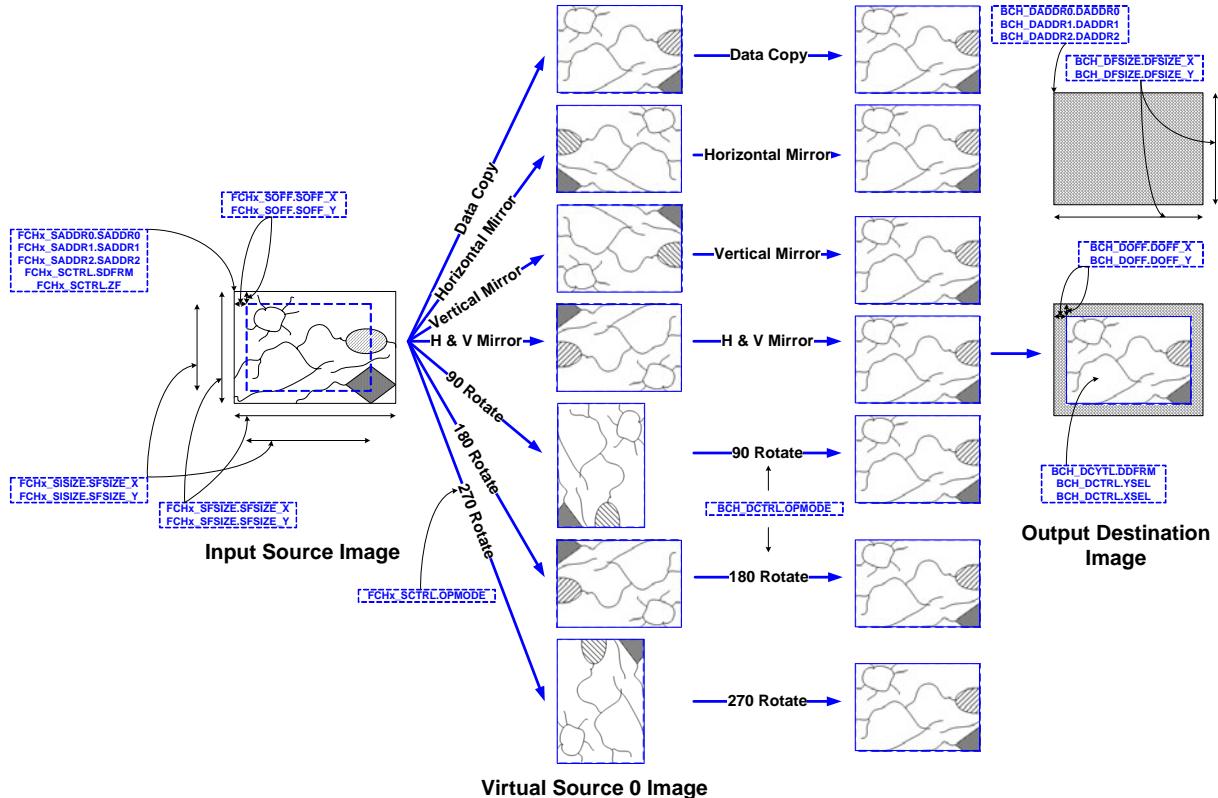


Figure 17.4 Source & Destination Mirror / Rotate Operation

Figure 17.4 illustrates one of functions supported by Graphic Engine, Rotate / Mirror function. The figure shows Internal Register Setting when there is one Input Source Image and Input Image Source is FCH1.

As explained before, if the number of an image is One, the image should be selected to Source 0. Window Offset of the image which is selected to Source 0 should definitely be '0'. In addition to this, GE_CTRL.EN Register should be set last.

- (1) FCH1_SADDR0 / 1 / 2 : The Input Source Image Start Address.
- (2) FCH1_SCTRL.SDFRM, FCH1_SCTRL.ZF : The Input Source Image Data Format
- (3) FCH1_SFSIZE : The Input Source Image Frame Size
- (4) FCH1_SISIZE : The Input Source Image Size
- (5) FCH1_SOFF : The Input Source Image Offset
- (6) FCH1_SCTRL.OPMODE : FCH1 Operation Mod
- (7) FCH1_WOFF : Should be Zero.
- (8) S_CTRL.S0SEL : Should be Front-End Channel 1
- (9) S_CTRL.S0_ARITHMOD : Should be Bypass Mode.
- (10) S_CTRL.S0_Y2REN, S0_CHROMAEN : Should be Disable.
- (11) OP_CTRL.C0SEL, C1SEL : Should be Disable.
- (12) OP_CTRL.OP0MODE, OP1MODE: Should be Source Copy
- (13) BCH_DCTRL.OPMODE : BCH Operation Mode.
- (14) BCH_DCTRL.Y2R : Should be Disable.

- (15) BCH_DADDR0 / 1 / 2 : The Output Destination Image Start Address.
- (16) BCH_DCTRL.DDFRM, DCH_DCTRL.X / YSEL : The Output Destination Image Data Format
- (17) BCH_DFSIZE : The Output Destination Image Frame Size
- (18) BCH_DOFF : The Output Destination Image Offset
- (19) GE_CTRL : GE_CTRL.EN = b'010

17.4.2 YUV to RGB Format Conversion

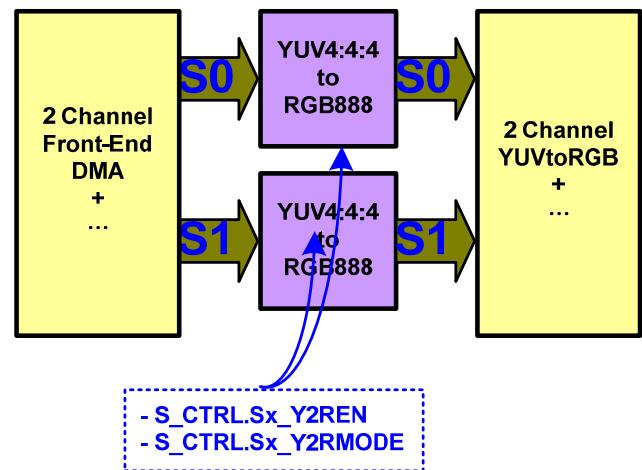


Figure 17.5 Source YUV to RGB Conversion

Figure 17.5 explains one of functions supported by Graphic Engine, 2 Channel YUV to RGB Format Conversion function. As can be seen, 2 Channel YUV Source Image of Graphic Engine can be converted to RGB888. S0 and S1 in Figure 17.5 are the images selected by S_CTRL.SxSEL Register.

Below shows 4 equations in YUV to RGB Conversion per each S_CTRL.Sx_Y2RMODE Register.

- YUVtoRGB Format Converter Type 0
 - $R = Y + 1.371 (V - 128)$
 - $G = Y - 0.336 (U - 128) - 0.698 (V - 128)$
 - $B = Y + 1.732 (U - 128)$
- YUVtoRGB Format Converter Type 1
 - $R = 1.164 (Y - 16) + 1.159 (V - 128)$
 - $G = 1.164 (Y - 16) - 0.391 (U - 128) - 0.813 (V - 128)$
 - $B = 1.164 (Y - 16) + 2.018 (U - 128)$
- YUVtoRGB Format Converter Type 2
 - $R = Y + 1.540 (V - 128)$
 - $G = Y - 0.183 (U - 128) - 0.459 (V - 128)$
 - $B = Y + 1.816 (U - 128)$
- YUVtoRGB Format Converter Type 3
 - $R = 1.164 (Y - 16) + 1.793 (V - 128)$
 - $G = 1.164 (Y - 16) - 0.213 (U - 128) - 0.534 (V - 128)$
 - $B = 1.164 (Y - 16) + 2.115 (U - 128)$

17.4.3 Arithmetic Operation

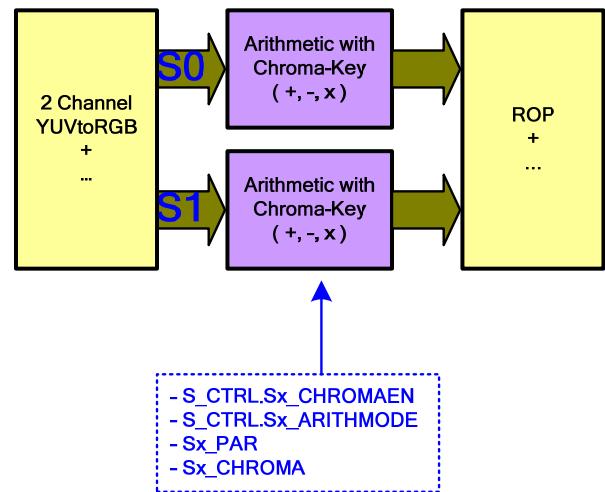


Figure 17.6 Arithmetic Operation

Figure 17.6 explains 2 Channel Arithmetic Operation, one of functions which Graphic Engine supports. As can be seen above, arithmetic operation for 2 Channel Image Source is available in Graphic Engine. In the figure, S0 and S1 are the results of performing YUV to RGB Conversion in the images selected by S_CTRL.SxSEL Register.

The following is about Arithmetic Operation per each S_CTRL.Sx_ARITHMODE Register.

A	B	Result	
		A = Sx_CHROMA	A != Sx_CHROMA
BYPASS	Source 0/1 Image	Do not Used	A Y = A
FILL	Do not Used	Sx_PAR	A Y = B
INV	Source 0/1 Image	Do not Used	A Y = 255 - A
ADD	Source 0/1 Image	Sx_PAR	A Y = A + B
SUBA	Source 0/1 Image	Sx_PAR	A Y = A - B
SUBB	Source 0/1 Image	Sx_PAR	A Y = B - A
			Y = A x B
MUL	Source 0/1 Image	Sx_PAR	A Mantissa = B[07:06] Fraction = B[05:00]

As can be seen above, Arithmetic Operation is performed after Source YUV to RGB Conversion.

Therefore, for Arithmetic Operation with Chroma-Key, the value after Source YUV to RGB Conversion should be set as Chroma-Key Value (Sx_CHROMA).

17.4.4 Alpha-Blending and ROP with Window Offset and Chroma-Key

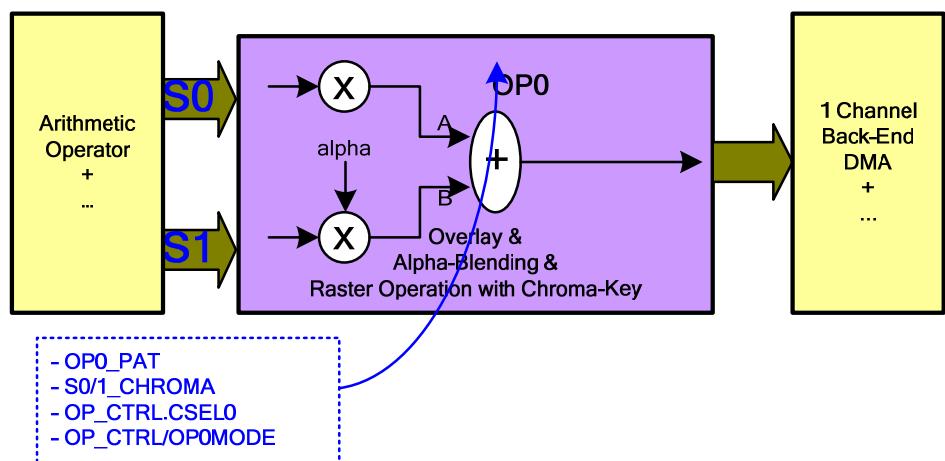


Figure 17.7 ROP / Alpha-Blending Block Diagram

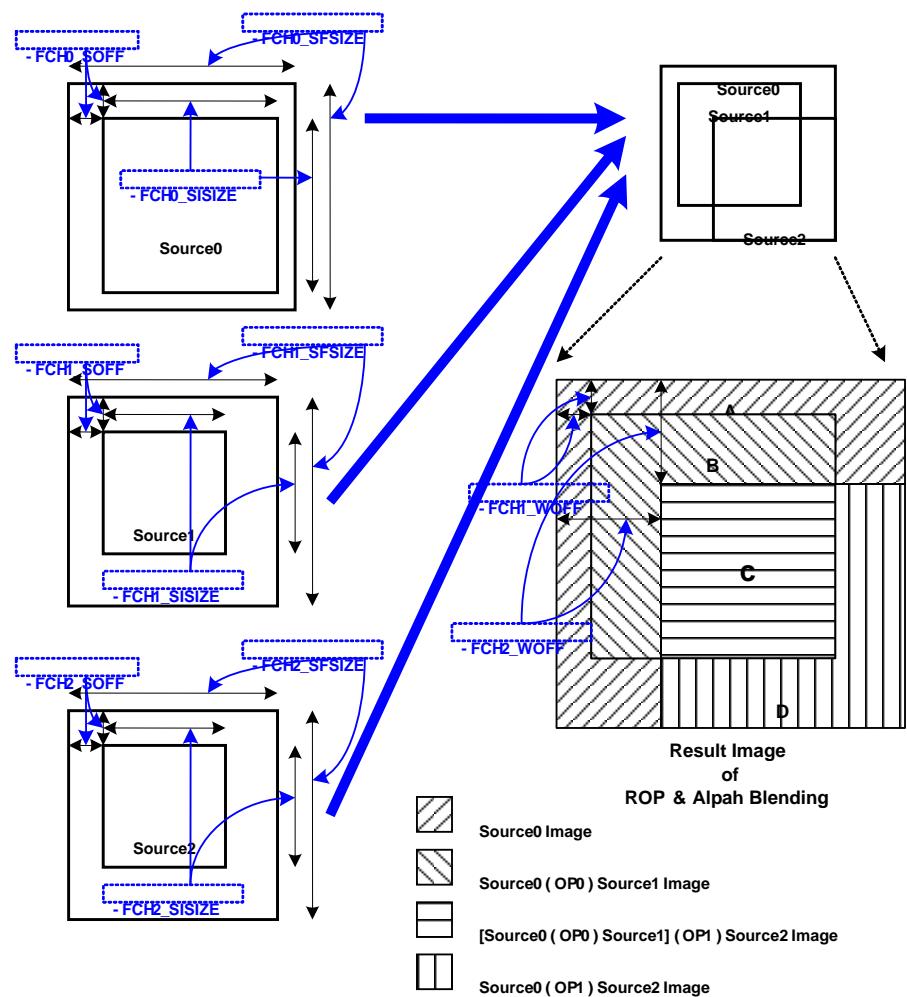


Figure 17.8 ROP and Alpha-Blending Operation Example
(FCH2 is not available in the TCC79XX)

This figure illustrates ROP, Alpha-Blending Operation. It assumes that FCH0 is Source 0 and FCH1, FCH2 are Source 1 and Source 2 respectively. As the figure explains, Source 0 Image should definitely include other Source Images. (Note that FCH2 is not available in the TCC79XX.)

In addition, in case of Source 0 Image, Window Offset should definitely be '0'. Other than these, there are no requirements for ROP and Alpha-Blending Operation.

Below shows the function of Operator 0. Note that Operator 1 should be "Src Copy" in the TCC79XX.

Result of Operator 0			
A = Source 0 selected by S_CTRL.S0SEL			
B = Source 1 selected by S_CTRL.S1SEL			
P = OP0_PAT[23:00]			
ALPHA0 = OP0_PAT[27:24] / 16			
ALPHA1 = Alpha-Value of Alpha-RGB Format Source 1 Data			
OPCTRL_CSEL0 01 10			
A = S0_CHROMA B = S1_CHROMA			
Blackness	Y = 0	B	A
Merged Copy	Y = P & A	B	A
Merged Paint	Y = ~A B	B	A
Pattern Copy	Y = P	B	A
Pattern Invert	Y = P ^ B	B	A
Pattern Paint	Y = (P ~A) B	B	A
Src Copy	Y = A	B	A
Src Inverter	Y = A ^ B	B	A
Srd Paint	Y = A B	B	A
Src AND	Y = A & B	B	A
Src Erase	Y = A & ~B	B	A
Not Src Copy	Y = ~A	B	A
Not Src Erase	Y = ~ (A B)	B	A
Dst Copy	Y = B	B	A
Dst Inverter	Y = ~ B	B	A
Whiteness	Y = 1	B	A
Alpha-Blending 0	Y = ALPHA0 * B + (1 - ALPHA0) * A	B	A
Alpha-Blending 1	Y = ALPHA1 * B + (1 - ALPHA1) * A	B	A

17.4.5 RGB to YUV Format Conversion

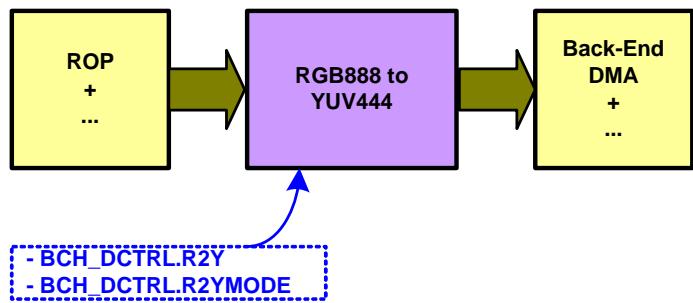


Figure 17.9 RGB to YUV Format Converter
(FCH2 is not available in the TCC79XX)

The figure is about 1 Channel RGB to YUV Format Conversion function provided by Graphic Engine.

As can be seen above, RGB to YUV Format Conversion is performed after ROP and Alpha-Blending Operation.

The following explains the equation of RGB to YUV Conversion per BCH_DCTRL.R2YMODE Register.

- RGBtoYUV Format Converter Type 0
 - $Y = 0.299 R + 0.587 G + 0.114 B$
 - $U = -0.172 R - 0.339 G + 0.511 B + 128$
 - $V = 0.511 R - 0.428 G - 0.083 B + 128$
- RGBtoYUV Format Converter Type 1
 - $Y = 0.257 R + 0.504 G + 0.098 B + 16$
 - $U = -0.148 R - 0.291 G + 0.439 B + 128$
 - $V = 0.439 R - 0.368 G - 0.071 B + 128$
- RGBtoYUV Format Converter Type 2
 - $Y = 0.213 R + 0.715 G + 0.072 B$
 - $U = -0.117 R - 0.394 G + 0.511 B + 128$
 - $V = 0.511 R - 0.464 G - 0.047 B + 128$
- RGBtoYUV Format Converter Type 3
 - $Y = 0.183 R + 0.614 G + 0.062 B + 16$
 - $U = -0.101 R - 0.338 G + 0.439 B + 128$
 - $V = 0.439 R - 0.399 G - 0.040 B + 128$

18 EHI**18.1 Overview**

This LSI has the external host interface (EHI) that allows the external host device to be connected to the on-chip system bus. The external host device can be directly connected to 68/80-series interfaces and access the memory area of this LSI. For software based data transfer, EHI can generate the internal interrupt of this LSI, and this LSI can also send interrupt request to the external host controller.

The features of EHI are as follows.

- 68/80 series interface with 8/16bits data can be supported.
- Burst transfer is supported and address can be incremented automatically.
- External host device can generate an internal interrupt of this LSI.
- Interrupt request can be sent to the external host by programming specific bits in EHI control register.
- Semaphore is supported for improving data transfer efficiency.

The interface signals are shown in Table 18.1. HPCSN_L and HPCSN should not be asserted at the same time.

Table 18.1 EHI External Interface Pin

PIN Name	I/O	Function
HPCSN_L	I	Chip select signal 1
HPCSN	I	Chip select signal 0
		Address signal. It is used for switching between normal access and EHIND/EHST register access.
HPXA[0]	I	Normal access (HPXA [0]= 0): EHI register indicated by EHIND register is accessed. EHIND/EHST access (HPXA[0] = 1) : Writing to EHIND register and reading from EHST register.
HPWRN	I	68-interface: Enable signal 80-interface: Write strobe signal
HPRDN	I	68-interface: Data reading or writing select signal 80-interface: Read strobe signal
HPXD[17:0]	B	Data bus
HPINTO	O	External host interrupt request signal / Ready signal for HPCSN
HPINTO1	O	External host interrupt request signal / Ready signal for HPCSN_L

18.2 Register Description

The EHI registers are shown in Table 18.2. Chip Select 0 (HPCSN) base address is 0xF5000000 and Chip Select 1 (HPCSN_L) base address is 0xF00A0000.

Table 18.2 EHI register map

Name	Offset	Int.	Ext.	Initial	Description
EHST	0x00	R/W	R/W	0x00000080	Status register
EHIINT *	0x04	R/W	R/W	0x00000000	Internal interrupt control register
EHEINT*	0x08	R/W	R/W	0x00000000	External interrupt control register
EHA	0x0C	R	R/W	0x00000000	Address register
EHAM	0x10	R/W	R	0x00000000	Address masking register
EHD	0x14	R/W	R/W	0x00000000	Data register
EHSEM*	0x18	R/W	R/W	0x00000000	Semaphore register
EHCFG*	0x1C	R/W	R/W	0x00000000	Configuration registers
EHIND	0x20	R	W	0x00000000	Index register
EHRWCS*	0x24	R	R/W	0x00000000	Read/Write Control/Status register

*) When the external host device writes to these registers, EHIND [1:0] bits are ignored.

EHI has two clock inputs; one is HCLK, which is for the on-chip system bus interface and the other is ECLK (EHI CLK) which is for interface with the external host.

Before the on-chip CPU accesses these registers, HCLK and ECLK should be enabled.

ECLK should be enabled for the external host to access these registers except for EHST register. Refer to 18.3.1Access to EHI registers for more information.

EHST (Status Register)

BASE²⁹ +0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			RFNE	WFNE	WFF	FUR	FOR	RDY							ST[6:0]

Bit	Name	Initial	Int.	Ext.	Description	
31-13	-	0	R	R	These bits are read as 0.	
12	RFNE	0	R	R	It represents that Read-FIFO is not empty. When it is equal to 0, Read-FIFO is empty.	
11	WFNE	0	R	R	It represents that the Write-FIFO is not empty. When it is equal to 0, Write-FIFO is empty.	
10	WFF	0	R	R	It represents that the Write-FIFO is full.	
9	FUR	0	R/C	R	It represents that Read-FIFO underrun is occurred. It is cleared when the on-chip CPU writes 1 to this bit.	
8	FOR	0	R/C	R	It represents that Write-FIFO overrun is occurred. It is cleared when the on-chip CPU writes 1 to this bit.	
7	RDY	1	R	R	It represents that an external host device can access the on-chip system bus.	
6-0	ST	0	R/W	R/W	It is read and written by an external host connected to EHI and by the on-chip CPU.	

The external host device has two methods for reading this register. One is that the external host device issues the read operation while HPXA is high. The other is that the external host sets EHIND to 0x00 or 0x01 (EHST offset) and issues the read operation while HPXA is low.

When using the 8-bit interface, EHST[7:0] can be read by the external host device while HPXA is high regardless of ECLK. EHST[15:8] is read by the external host device while HPXA is low and EHIND is set to 0x01. Therefore, when ECLK is disabled, the external host device can not read EHST[15:8].

When using the 16-bit interface, EHST[15:0] can be read by the external host device while HPXA is high. Therefore the external host device can read EHST[15:0] regardless of ECLK.

²⁹ BASE = 0xF5000000 is for HPCSN and 0xF00A0000 is for HPCSN_L.

EHIINT (Internal Interrupt Control Register)

BASE³⁰+0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														IIRQ_ST	IIRQ

Bit	Name	Initial	Int.	Ext.	Description
31-8	-	0	R	R	These bits are read as 0.
7-1	IIRQ_ST	0	R/W	R/W	It is only updated by software. It can specify the interrupt request number.
0	IIRQ	0	R/W	R/W	When it is equal to 1, EHI generates the interrupt request and it is sent to the on-chip interrupt controller. Note that it should be cleared manually. The external host device may need to check that the previous issued interrupt is processed by the on-chip CPU before issuing the new interrupt request. It is only updated by software.

EHIINT register is used to generate interrupts from the external host device. When the interrupt is generated, the on-chip CPU processes the interrupt service routine and should be clear EHIINT.IIRQ bit. Therefore, the external host can detect whether the requested interrupt is processed as it reads EHIINT register.

³⁰ BASE = 0xF5000000 is for HPCSN and 0xF00A0000 is for HPCSN_L.

EHEINT (External Interrupt Control Register)

BASE³¹ +0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														EIRQ_ST	

BIT	Name	Initial	Int.	Ext.	Descriptions
31-8	-	0	R	R	These bits are read as 0
7-1	EIRQ_ST	0	R/W	R/W	This specifies EIRQ interrupt source. The external host device can detect interrupt source via EIRQ_ST.
0	EIRQ	0	R/W	R/W	If HPINT is connected to the external interrupt input of the external host, this LSI can send interrupt request to the external host. When EHCFG.RDYE is equal to 0, EHEINT.EIRQ value is output through HPINT pin. Otherwise, this bit is not applicable.

EHEINT is used to issue interrupts to the external host via HPINT pin by the on-chip CPU. HPINT pin is used for interrupt request pin when EHCFG.RDYE is set to 0.

³¹ BASE = 0xF5000000 is for HPCSN and 0xF00A0000 is for HPCSN_L.

EHA (Address Register)**BASE³²+0x0C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EHA[15:2]															0

Bit	Name	Initial	Int.	Ext.	Description
31-2	-	0	R	R/W	The address for the on-chip system bus access of this LSI.
1-0	-	0	R	R	These bits are always 0.

EHA maps to the memory space of this LSI when the external host accesses the system bus of this LSI.

When the external host issues the incremental writing operation (EHRWCS.AI = 1 and EHRWCS.RW = 1), EHA increments automatically during data transfer and when it is completed (EHRWCS.RW = 0), EHA has the incremented address as the number of written data.

But, when the incremental reading operation is completed, EHA do not has the incremented address as the number of read data, but the prefetched address. Therefore, when the new incremental reading operation is issued, EHA should be updated.

EHA can be masked by EHAM register. Refer to the next page, EHAM register description, for more information.

³² BASE = 0xF5000000 is for HPCSN and 0xF00A0000 is for HPCSN_L.

EHAM (Address Masking Register)BASE³³ +0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHAM[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EHAM[15:2]															

Bit	Name	Initial	Int.	Ext.	Description
31-2	-	0	R/W	R	It masks EHA register. Therefore, the effective address on the on-chip system bus is EHA[31:2] & ~EHAM[31:2].
1-0	-	0	R	R	These bits are read as 0.

EHA can be masked by EHAM register and then the effective address is EHA[31:2] & ~EHAM[31:2]. When EHRWCS.AI is set to 1, the generated address of every transfers is (EHA[31:2] + 1) & ~EHAM[31:2].

For example, when EHA = 0x1000000, EHRWCS.AI=1, and EHAM = 0x0F00, EHA has the address between 0x10000000 and 0x100000FC.

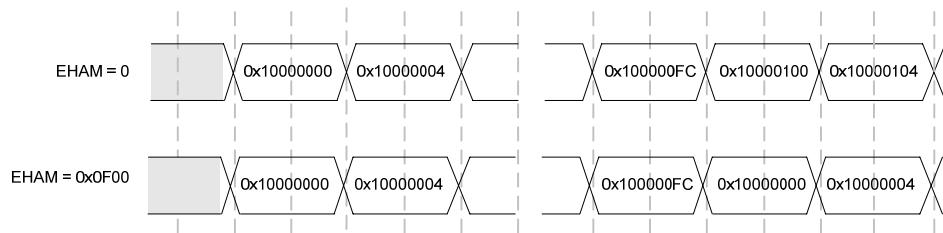


Figure 18.1 Example of EHAM usage

³³ BASE = 0xF5000000 is for HPCSN and 0xF00A0000 is for HPCSN_L.

EHD (Data Register)

BASE³⁴ +0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EHD[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EHD[15:0]															

Bit	Name	Initial	Int.	Ext.	Description
31-0	-	0	R/W	R/W	While EHRWCS.RW = 1 or 2, the external host device accesses the memory area of this LSI using EHA register. In this case, it is used for writing or reading data.

EHD is used to transfer data for system bus access mode.

³⁴ BASE = 0xF5000000 is for HPCSN and 0xF00A0000 is for HPCSN_L.

EHSEM (Semaphore Register)

BASE³⁵ +0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Name	Initial	Int.	Ext.	Description
					FLG = 00b Reading only: On-chip CPU, External host device
7-2	ST	0	-	-	FLG = 01b Reading/Writing: External host device Reading only: On-chip CPU
					FLG = 10b Reading/Writing: On-chip CPU Reading only: External host device
					If EHSEM is read and then FLG is 00b, this means EHSEM is not occupied by any master. If the external host device reads EHSEM which is not occupied by this LSI, then EHSEM.FLG becomes 01b. If the on-chip CPU reads EHSEM which is not occupied by the external host device, then EHSEM.FLG becomes 10b. If the on-chip CPU and the external host read EHSEM simultaneously, return value for external host is 00b and it for the on-chip CPU is 01b.
1-0	FLG	0	-	-	00b (NOT OCCUPIED) Reading only: On-chip CPU, External host device
					01b (EXTERNAL HOST) Reading/Writing: External host device Reading only: On-chip CPU
					10b (ON-CHIP CPU) Reading/Writing: On-chip CPU Reading only: External host device
					11b N/A

³⁵ BASE = 0xF5000000 is for HPCSN and 0xF00A0000 is for HPCSN_L.

EHCFG (Configuration Register)

BASE³⁶ +0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								WIRQ	CSIRQ		RDYP	RDYE	BW		MD

Bit	Name	Initial	Int.	Ext.	Description	
31-8	-	0	R	R	These bits are read as 0.	
7	WIRQ	0	R/W	R/W	Writing operation becomes the internal interrupt source. When HPCSN and HPWRN are asserted in 80-mode or HPCSN, HPWRN(E), and HPRDN(R/W) in 68-mode, the internal interrupt request is generated.	
					For preventing additional interrupts by write operation, this bit should be cleared manually.	
6	CSIRQ	0	R/W	R/W	HPCSN becomes the internal interrupt source. When HPCSN is asserted regardless of HPWRN and HPRDN, the internal interrupt request is generated.	
					For preventing additional interrupts by HPCSN, this bit should be cleared manually.	
5	-	0	R	R	This bit reads as 0.	
4	RDYP	0	R/W	R/W	When RDYE is equal to 1, EHST.RDY \oplus EHCFG.RDYP is outputted through HPINT. Therefore, it is only valid when RDYE is equal to 1. Refer to RDYE description.	
3	RDYE	0	R/W	R/W	0: HPINT is used for the external interrupt. EHEINT.EIRQ is outputted through HPINT. 1: HPINT is used for READY signal. In this case, RDYP is valid.	
2	BW	0	R/W	R/W	0 : 8-bit interface mode 1 : 16-bit interface mode	
					00b: 8-bit interface mode 01b: Reserved 10b: 16-bit interface mode	
2-1	BW	0	R/W	R/W	11b: 18-bit interface mode 18-bit data is written to and is read from lower 18bits of 32-bit word memory. Therefore, upper 14bits in 32-bit word memory cannot be used.	
0	MD	0	R/W	R/W	0 : 80 interface mode 1 : 68 interface mode	

EHCFG configures the overall EHI operation. The on-chip CPU should select the interface mode, 80-interface mode or 68-interface mode.

When the external device can not write to the EHI registers, because of clock configuration or interface timing problem, it can issues interrupts for the on-chip CPU using EHCFG.WIRQ or EHCFG.CSIRQ (Figure 18.2).

³⁶ BASE = 0xF5000000 is for HPCSN and 0xF00A0000 is for HPCSN_L.

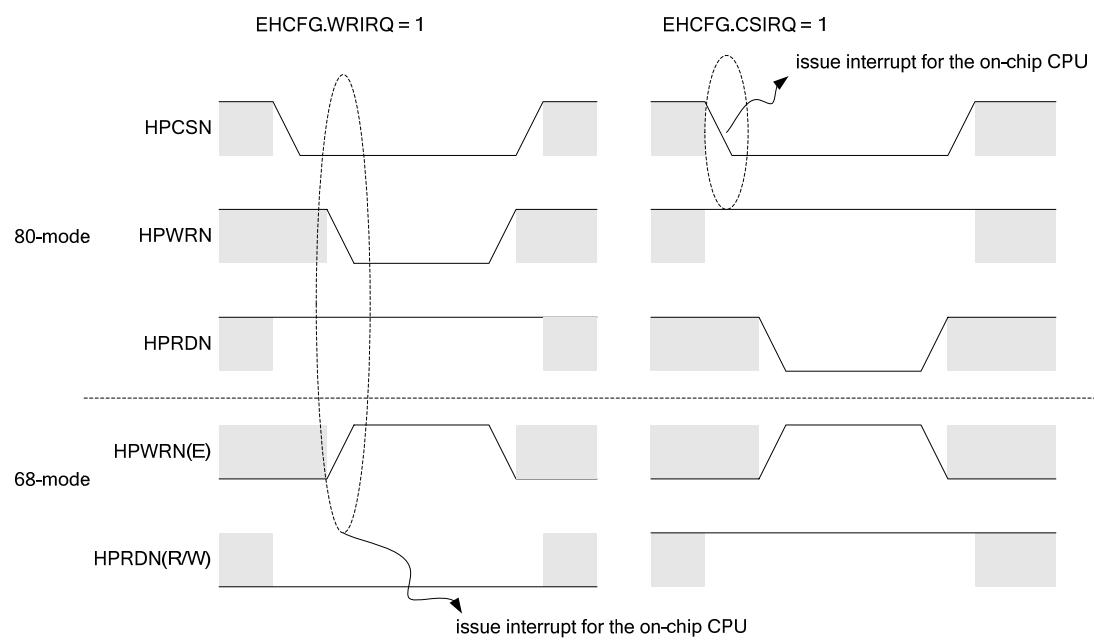


Figure 18.2 Interrupt request using EHCFG.WRIRQ and CSIRQ

EHIND (Index Register)

BASE³⁷ +0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EHIND[7:0]															

Bit	Name	Initial	Int.	Ext.	Description
31-8	-	0	R	R	These bits read as 0.
7-0	EHIND	0	R	W	<p>It selects the internal register for the external host. It can be only written by the external host device while HPXA is high. Refer to Table 18.2 for value corresponding to each register.</p> <p>0x00: EHST 0x04: EHIINT 0x08: EHEINT 0x0C: EHA 0x10: EHAM 0x14: EHD 0x18: EHSEM 0x1C: EHCFG 0x20: EHIND 0x24: EHRWCS</p> <p>When writing to EHIINT, EHEINT, EHSEM, EHCFG, and EHRWCS, EHIND[1:0] is ignored. For example, when writing to EHIINT[7:0] register, EHIND can be 4, 5, 6, or 7</p>
					EHIND is used to index other EHI registers and only written by the external host while HPXA is high.

³⁷ BASE = 0xF5000000 is for HPCSN and 0xF00A0000 is for HPCSN_L.

EHRWCS (Read/Write Control/Status Register)

BASE³⁸ +0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									AI	LK	RW				BSIZE

Bit	Name	Initial	Int.	Ext.	Description	
31-8	-	0	R	R	These bits are read as 0.	
7	AI	0	R	R/W	EHA is auto-incremented while EHRWCS.RW is 1 or 2.	
6	LK	0	R	R/W	Once the external host device is a master of the system bus of this LSI, bus-handover cannot be occurred until burst transfer as EHRWCS.BSIZE is completed.	
5-4	RW	0	R	R/W	0 : Access to EHI registers 1 : Writing to the system bus of this LSI. When BSIZE = 0(single transfer), it is cleared automatically. 2 : Reading from the system bus of this LSI. When BSIZE = 0, it is cleared automatically. 3 : Undefined	
3-0	BSIZE	0	R	R/W	It specifies how many words(1word = 32bits) will be transferred and EHRWCS.BSIZE + 1 words will be transferred. It should be programmed before EHRWCS.RW is set to 2 (Reading operation) or 1(Writing operation). For continuous burst transfer, EHST.RDY may need to be checked every burst size transfer.	

EHRWCS register is used to control the on-chip system bus access.

Before EHRWCS.RW is issued for reading or writing operation, other fields of this register and EHA should be configured. Refer to 18.3.2Access to the On-chip System Bus about reading and writing operation for the on-chip system bus.

³⁸ BASE = 0xF5000000 is for HPCSN and 0xF00A0000 is for HPCSN_L.

18.3 Operation

18.3.1 Access to EHI registers

Table 18.3 Access to the EHI

HCLK	ECLK	Reading From EHI register		Writing to EHI register		access to the on-chip system bus	
		on-chip	Ext.Host	on-chip	Ext.Host	Ext.Host	Ext.Host
X	X	X	△	X	X	X	X
X	O	X	O	X	O	X	
O	X	X	△	X	X	X	
O	O	O	O	O	O	O	O

△ = When 8bits interface is used, EHST[7:0] can be only read. When 16bits interface is used, EHST[15:0] can be only read.

ECLK and HCLK should be enabled for the on-chip CPU to access the EHI registers. And the external host can only access the EHI registers when ECLK is enabled except for EHST register. When ECLK is disabled, the external host can only read EHST[15:0] register in the 16-bit interface mode and EHST[7:0] register in the 8-bit interface mode. The relationship between clocks and accessibility is shown in Table 18.3.

Most of EHI registers are accessed by indirect addressing using EHIND register. To write to or read from an EHI register except for EHIND and EHST registers, EHIND register should be set to its offset value in advance and then HPXA should be low during reading or writing operation. If the external host intends to write data to EHIND, it should drive HPXA pin to high. When the external host device issues the reading operation while HPXA is high, it reads EHST register. Note that EHIND register is write-only. When the external host device intends to write to EHST register, it should also use EHIND register. The following shows how to write data (=0x12345678) to EHA register.

- (1) Write EHA[15:0] offset value(=0x0C) while HPXA = 1. EHIND register indicates EHA[15:0] register.
- (2) Write 0x5678 for EHA[15:0] while HPXA=0.
- (3) Write EHA[31:16] offset value(=0x0E) while HPXA = 1. EHIND register indicates EHA[31:16] register.
- (4) Write 0x1234 for EHA[31:16] while HPXA = 0.

Figure 18.3 and Figure 18.4 show that the external host device writes to and reads from EHA register.

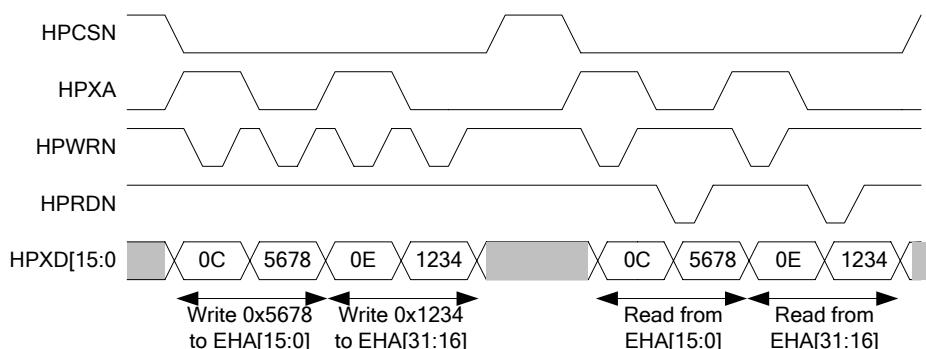


Figure 18.3 Example of writing / reading operation (80 interface, 16bits)

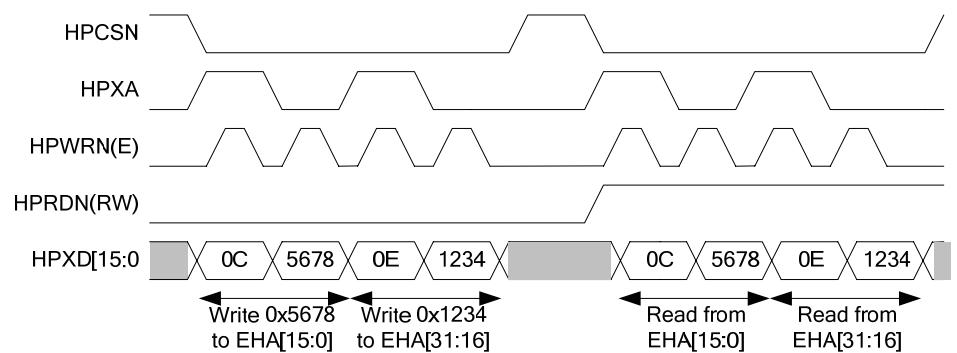


Figure 18.4 Example of writing / reading operation (68 interface, 16bits)

18.3.2 Access to the On-chip System Bus

The external host device that is connected to this LSI via EHI can access the on-chip system bus. This connection supports word-aligned (32-bits) burst transfer. When the external host device access the on-chip system bus, HCLK and ECLK should be enabled in advance.

The length of burst transfer is determined by EHRWCS.BSIZE. Once transfer is occurred, the length of burst cannot be modified until the requested transfer is completed. Therefore, EHRWCS.BSIZE, EHRWCS.AI, and EHRWCS.LK can be only modified while EHRWCS.RW = 0. If EHST.RDY is checked every burst transfer, the FIFO underrun or overrun cannot be occurred. But, if the burst transfers are intended to be issued without EHST.RDY check to improve data transfer efficiency, HCLK of this LSI should be fast enough not to underrun and overrun. FIFO underrun and overrun can be detected via EHST.FUR and EHST.FOR respectively.

Figure 18.5 shows how to program the external host device for access to the on-chip system bus.

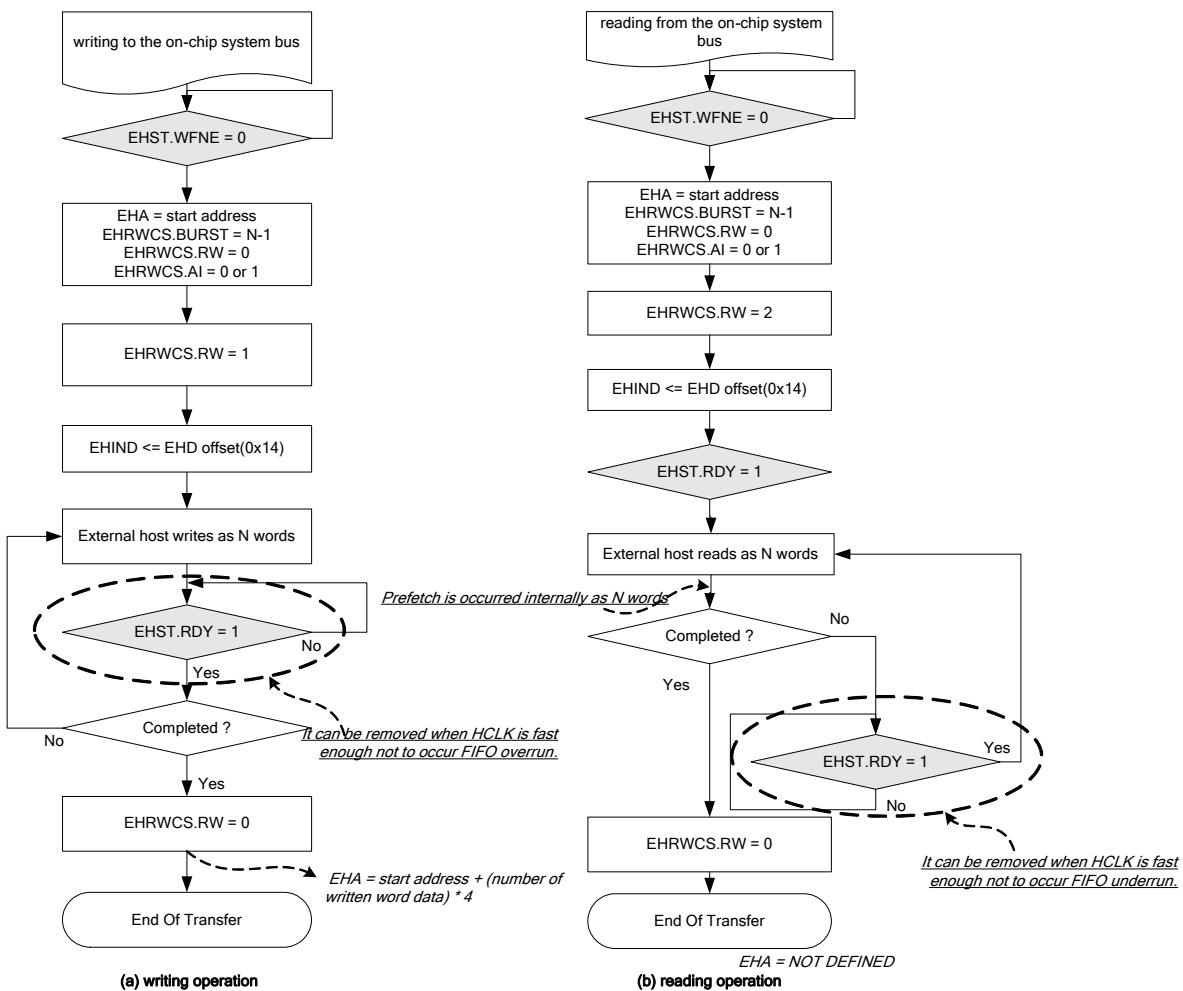


Figure 18.5 Access to the On-chip System Bus

Write-operation

The external host should configure writing operation before writing data; start address, address auto increment, lock or normal, and burst length.

The start address should be written to EHA register. When address auto increment is enable, address of the on-chip system bus is increment automatically every word transfer. This generated address is masked by EHAM. Therefore, address on the on-chip system bus is next EHA = (EHA[31:2] + 1) & ~EHAM[31:2]. When the writing operation is completed (EHRWCS.RW = 0), EHA has start address + number of written data.

EHRWCS.BSIZE specifies the maximum of burst length on the on-chip system. When the lock transfer mode is enabled, the bus handover is not occurred during burst transfer. But, when lock transfer mode is disabled and a higher priority master requests bus access, the bus handover is occurred and EHI burst transfer is terminated. And when the higher priority master looses access to bus, EHI burst transfer is continued. But, when the lock transfer mode is enabled, the overall system performance of the on-chip can be degraded.

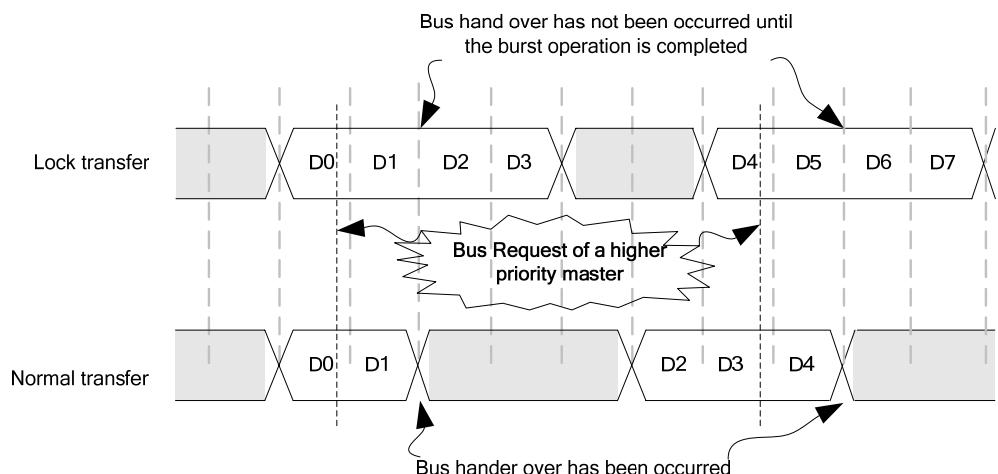


Figure 18.6 Lock transfer vs Normal transfer (EHRWCS.BSIZE=3)

After configuring writing operation, the external host enables writing operation as EHRWCS.RW is set to 1 and EHIND is set to EHD offset (0x14) sequentially. From now on, the external host can write data to the on-chip system bus while HPXA is low. For finishing the writing operation, the external host should set EHRWCS.RW to 0. After that, to confirm that all of written data arrived at destination area, the external host can check whether EHST.RDY is equal to 1.

The auto-increment writing operation which has that start address is 0x12345678, and the burst length is 8 is shown in Figure 18.7.

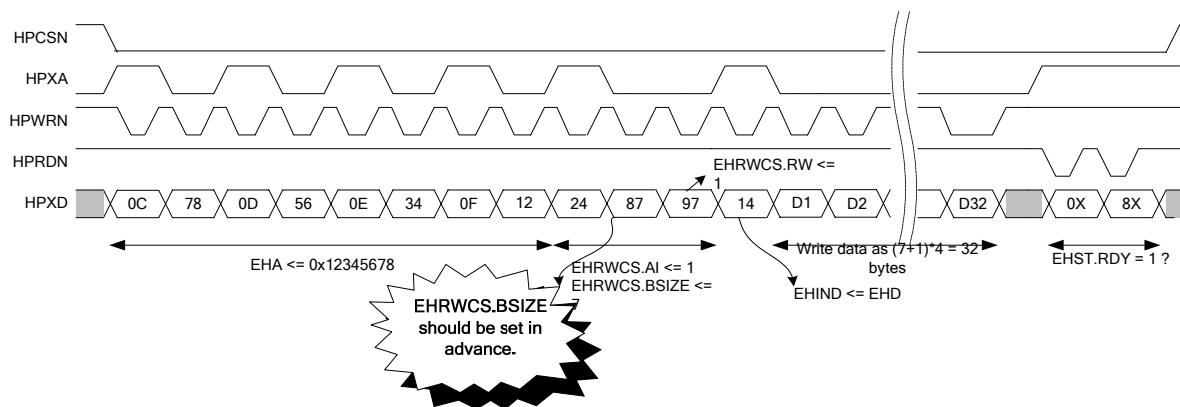


Figure 18.7 Example of Writing to the On-chip System Bus (80 interface, 8bits)

Read-operation

The configuration of reading operation is similar to that of the writing operation; start address, address auto increment, lock or normal, and burst length.

EHST.RDY may be checked every EHRWCS.BSIZE+1 words. If HCLK is fast enough not to underrun, EHST.RDY do not need to be check.

The auto-increment reading operation which has that start address is 0x12345678, and burst length is 8 is shown in Figure 18.8.

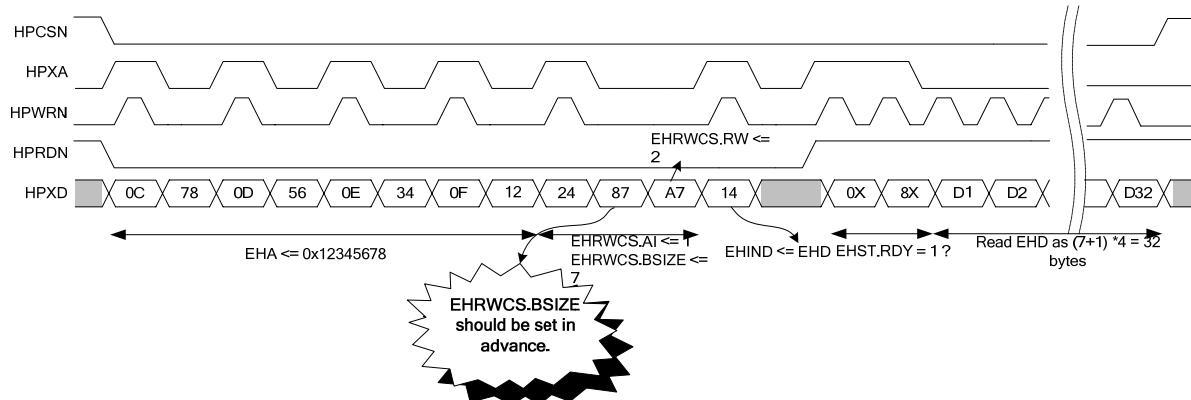


Figure 18.8 Example of Reading from the on-chip System Bus (68 interface, 8bits)

Notice that EHRWCS.RW is cleared to 0 automatically after the single reading and writing operation.

Entrance to the critical section using the semaphore register

EHI supports the semaphore for improving data transfer efficiency and EHSEM register is used.

<pre> int get_sem (void) { if (EHSEM.FLG & 1) { return FALSE; } else { return TRUE; } } void release_sem (void) { EHSEM.FLG = 0; } </pre> <p style="text-align: center;">On-Chip CPU</p>	<pre> int get_sem (void) { sem = read EHSEM reg. ; if (sem & 2) { return FALSE; } else { return TRUE; } } void release_sem (void) { write 0 to EHSEM.FLG; } </pre> <p style="text-align: center;">External Host</p>
---	--

Figure 18.9 Pseudo code for getting and releasing a semaphore

18.3.3 Interrupt Request

The EHI has three kinds of interrupt requests; external software interrupt request, internal software interrupt request, and internal CS/WR interrupt request

The first, the external software interrupt request is generated via HPINT pin. Therefore, the HPINT pin should be configured as interrupt request pin. It is default value. After this configuration, the HPINT pin is controlled by EHEINT.IRQ bit directly.

The second, the external host can request an interrupt to the on-chip CPU. This internal software interrupt request is issued when EHIINT.IIRQ is set to 1.

Finally, internal CS/WR interrupt request uses HPCSN, HPWRN, HPRDN signals as interrupt source. For using HPCSN signal as the interrupt source, EHCFG.CSIRQ bit should be set to 1. When EHCFG.CSIRQ is equal to 1 and HPCSN is asserted, the interrupt request is issued. The interrupt service routine of the on-chip CPU should clear EHCFG.CSIRQ to 0 not to generate the additional interrupt request by HPCSN signal. In the case of using writing operation as interrupt source, EHCFG.WRIRQ bit should be set to 1. After that, when a writing operation is issued (80-mode: HPCSN=0 HPWRN=0, 68-mode: HPCSN=0, HPWRN=1, HPRDN=0), the interrupt is generated. The external host can issue this interrupt request regardless of ECLK.

19 DAI & CDIF (AUDIO & CD INTERFACE)

19.1 DAI (Digital Audio Interface)

The block diagram of DAI is shown in Figure 19.1.

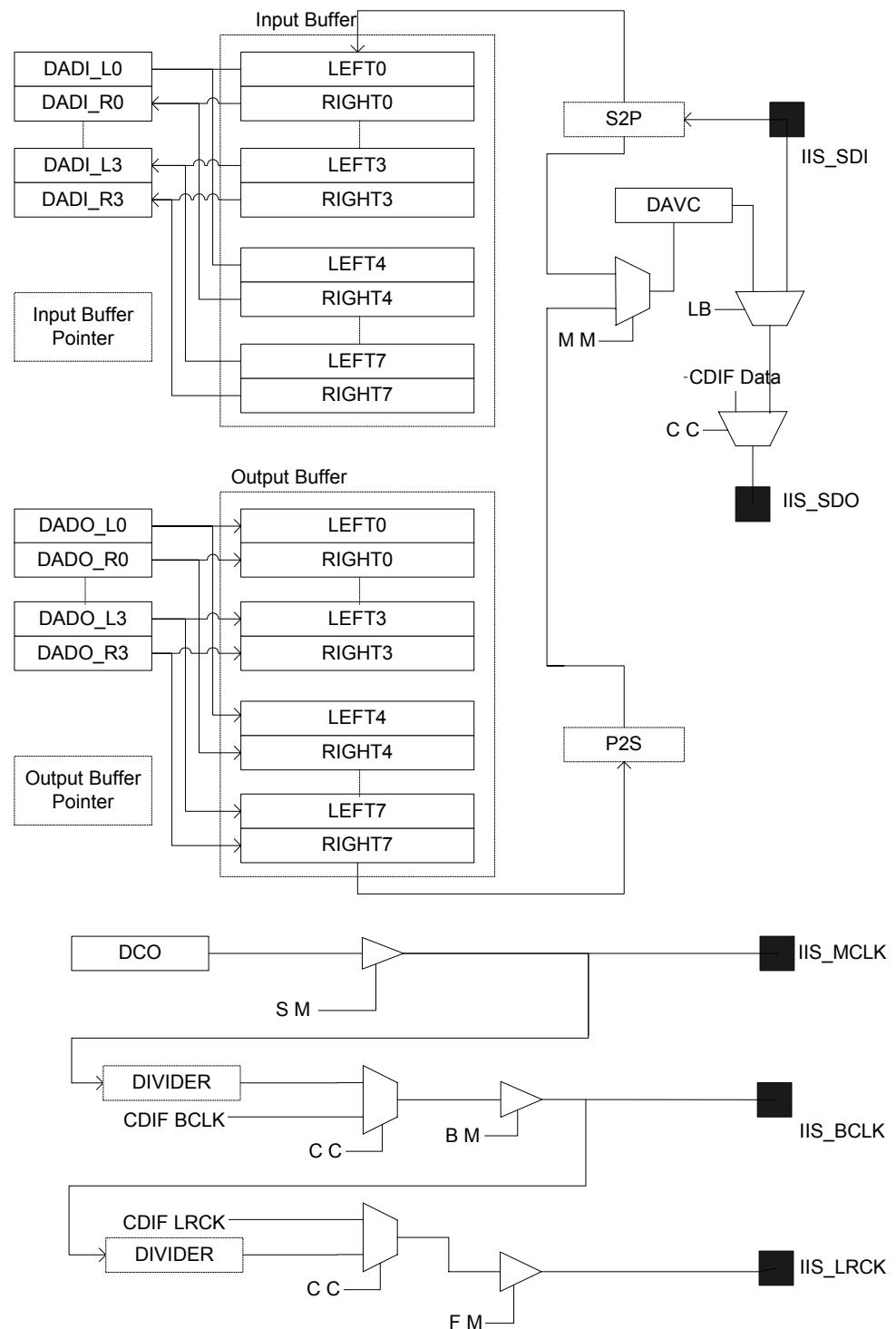


Figure 19.1 DAI Block Diagram

The TCC79XX provides digital audio interface that complies with IIS (Inter-IC Sound). The DAI has five input/output pins for IIS interface; MCLK, BCLK, LRCK, DAI, DAO.

The MCLK is the system clock pin that is used for CODEC system clock. In master mode, the MCLK can be generated from clock generator known as a DCLK, or fed from the outside of the TCC79XX in slave mode. The DAI can process 256fs, 384fs and 512fs as a system clock. 256fs means that the system clock has 256 times of sampling frequency (fs).

The BCLK is the serial bit clock for IIS data exchange. The DAI can generate 64fs, 48fs and 32fs by dividing a system clock. The polarity of BCLK can be programmed. That is, the serial bit can be stable either at the rising edge of BCLK or falling edge of BCLK.

The LRCK is the frame clock for the stereo audio channel Left and Right. The frequency of LRCK is known as the "fs" – sampling frequency. Generally, for audio application – such as MP3 Player, CD player, the fs can be set to 8kHz, 16kHz, 11.05kHz, 24kHz, 32kHz, 44.1kHz and 48kHz. For supporting the wide range of sampling frequency in audio application, the DCO function is very useful to generate a system clock. Refer the chapter of clock generator for detail information.

All three clocks (MCLK, BCLK, LRCK) are selectable as master or slave.

The DAI, DAO are the serial data input output pins respectively.

The DAI has two 8-word input/output buffers. It has a banked buffer structure so that one side of buffer is receiving/transmitting data while the other side of that can be read/written through the DADI_XX/DADO_XX registers. The maximum data word size is 24 bit. Data is justified to MSB of 32bits and zeros are padded to LSB.

There are 2 types of interrupt from IIS; transmit done interrupt, receive done interrupt. The transmit-done interrupt is generated when the 8 words are transferred successfully in the output buffer. At this interrupt, user should fill another 8 more words into the other part of the output buffer in the interrupt service routine (ISR). In this ISR routine, 8 consecutive stores of word data to the DADO registers are needed. The receive-done interrupt is generated when the 8 words are received successfully in the input buffer. At this interrupt, user should read 8 received words from the input buffer using 8 consecutive load instructions from the DADI registers.

Table 19.1 DAI Register Map (Base Address = 0xF0059000)

Name	Address	Type	Reset	Description
DADI_L0	0x00	R	-	Digital Audio Left Input Register 0
DADI_R0	0x04	R	-	Digital Audio Right Input Register 0
DADI_L1	0x08	R	-	Digital Audio Left Input Register 1
DADI_R1	0x0C	R	-	Digital Audio Right Input Register 1
DADI_L2	0x10	R	-	Digital Audio Left Input Register 2
DADI_R2	0x14	R	-	Digital Audio Right Input Register 2
DADI_L3	0x18	R	-	Digital Audio Left Input Register 3
DADI_R3	0x1C	R	-	Digital Audio Right Input Register 3
DADO_L0	0x20	R/W	-	Digital Audio Left Output Register 0
DADO_R0	0x24	R/W	-	Digital Audio Right Output Register 0
DADO_L1	0x28	R/W	-	Digital Audio Left Output Register 1
DADO_R1	0x2C	R/W	-	Digital Audio Right Output Register 1
DADO_L2	0x30	R/W	-	Digital Audio Left Output Register 2
DADO_R2	0x34	R/W	-	Digital Audio Right Output Register 2
DADO_L3	0x38	R/W	-	Digital Audio Left Output Register 3
DADO_R3	0x3C	R/W	-	Digital Audio Right Output Register 3
DAMR	0x40	R/W	0x00000000	Digital Audio Mode Register
DAVC	0x44	R/W	0x0000	Digital Audio Volume Control Register

Digital Audio Mode Register (DAMR)

0xF0059040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MS	Reserved								RXE	RXS<2:0>		TXS<1:0>		0	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	TE	RE	MD	SM	BM	FM	CC	BD<1:0>		FD<1:0>		BP	CM	MM	LB

MS [31]

Master Clock Selection Register

- | | |
|---|--|
| 0 | Internal BCLK is used. (Master Mode) |
| 1 | External BCLK is used. (Slave Mode) |

RXE [22]

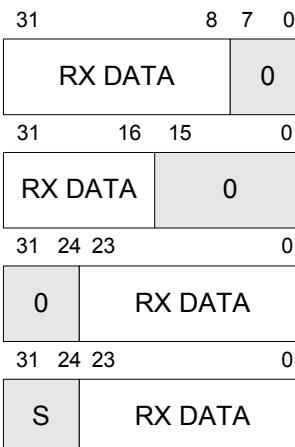
DAI RX Data Sign Extension

- | | |
|---|-----------------------------|
| 0 | Disable (zero extension) |
| 1 | Enable (sign bit extension) |

RXS [21:20]

DAI RX Shift

- | | |
|----------|------------------------------|
| 00 | Bit-pack MSB and 24bit mode. |
| 01 | Bit-pack MSB and 16bit mode. |
| 10 or 11 | Bit-pack LSB and 24bit mode. |



TXS [19:18]

DAI TX Shift

- | | |
|----|------------------------------|
| 0x | Bit-pack MSB mode. |
| 10 | Bit-pack LSB and 24bit mode. |
| 11 | Bit-pack LSB and 16bit mode. |

EN [15]

DAI Master Enable

- | | |
|---|--------------------|
| 0 | Disable DAI module |
| 1 | Enable DAI module |

TE [14]

DAI Transmitter Enable

- | | |
|---|-------------------------|
| 0 | Disable DAI transmitter |
| 1 | Enable DAI transmitter |

RE [13]

DAI Receiver Enable

- | | |
|---|----------------------|
| 0 | Disable DAI receiver |
| 1 | Enable DAI receiver |

MD [12]		DAI Bus Mode
0	Set DAI bus as IIS bus mode	
1	Set DAI bus as MSB justified mode	
SM [11]		DAI System Clock Master Select
0	Set that DAI system clock is come from external pin	
1	Set that DAI system clock is generated by the clock generator block	

The DAI system clock in clock generator is known as DCLK. Its frequency can be determined by PCK_DAI from the CKC.

BM [10]		DAI Bit Clock Master Select
0	Set that DAI bit clock is from external pin	
1	Set that DAI bit clock is generated by dividing DAI system clock	
FM [9]		DAI Frame Clock Master Select
0	Set that DAI frame clock is come from external pin	
1	Set that DAI frame clock is generated by dividing DAI bit clock	
CC [8]		CDIF Clock Select
0	Disable CDIF Clock master mode	
1	Enable CDIF Clock master mode	
BD [7:6]		DAI Bit Clock Divider select
00	Select Div 4 (256fs->64fs)	
01	Select Div 6 (384fs->64fs)	
10	Select Div 8 (512fs->64fs, 384fs->48fs, 256fs->32fs)	
11	Select Div16 (512fs->32fs)	
FD [5:4]		DAI Frame Clock Divider select
00	Select Div 32 (32fs->fs)	
01	Select Div 48 (48fs->fs)	
10	Select Div 64 (64fs->fs)	

The combination of BD & FD determines that the ratio between main system clock and the sampling frequency. The multiplication between the division factor of BD and FD must be equal to this ratio.

BP [3]		DAI Bit Clock Polarity
0	Set that data is captured at positive edge of bit clock	
1	Set that data is captured at negative edge of bit clock	
CM [2]		CDIF Monitor Mode
0	Disable CDIF monitor mode	
1	Enable CDIF monitor mode. Data bypass from CDIF	
MM [1]		DAI Monitor Mode
0	Disable DAI monitor mode	
1	Enable DAI monitor mode. Transmitter should be enabled. (TE = 1)	

LB [0]	DAI Loop-back Mode
0	Disable DAI Loop back mode
1	Enable DAI Loop back mode

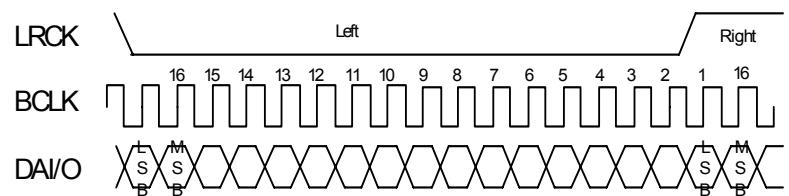
Digital Audio Volume Control Register (DAVC)

0xF0059044

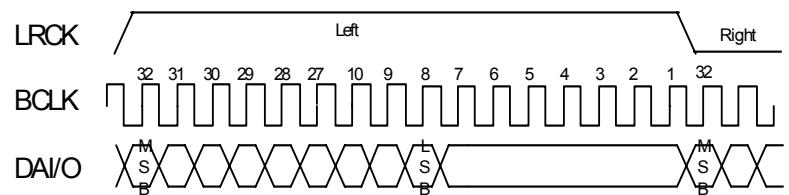
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

The volume of audio output can be manipulated by this register. It has -6dB unit so the output volume can be set from 0 dB to -90 dB as the following table.

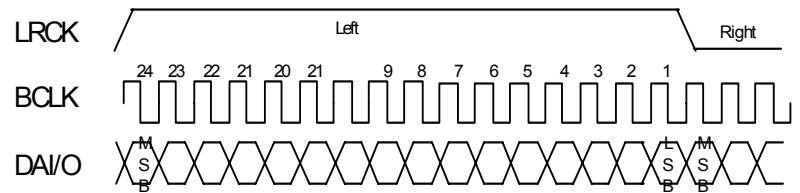
VC [4:0]	DAI Volume control
00000	0dB
00001	-6dB
00010	-12dB
00011	-18dB
00100	-24dB
00101	-30dB
00110	-36dB
00111	-42dB
01000	-48dB
01001	-54dB
01010	-60dB
01011	-66dB
01100	-72dB
01101	-78dB
01110	-84dB
01111	-90dB
10000	-96dB



MD=0 (IIS mode), BP=0, BCLK = 32fs



MD=1 (MSB justified mode), BP=0, BCLK=64fs



MD=1 (MSB justified mode), BP=1, BCLK=48fs

Figure 19.2 DAI Bus Timing Diagram

19.2 CDIF (CD Media Interface)

The block diagram of CDIF is illustrated in Figure 19.3.

The TCC79XX provides CD-ROM interface for feasible implementation of CD-ROM application such as CD-MP3 player. The CDIF supports the industry standard IIS format and the LSB justified format used as the most popular format for CD-ROM interface by Sony and Samsung.

The CDIF has three pins for interface; CBCLK, CLRCK, CDAI. The CBCLK is the bit clock input pin of which frequency can be programmed by CICR for selection of 48fs and 32fs. The CLRCK is the frame clock input pin that indicates the channel of CD stereo digital audio data. The CDAI is the input data pin.

The CDIF has nine registers; CDDI_0 to CDDI_3 and CICR. The CDDI_0 to the CDDI_3 are the banked read only registers for access of data input buffer. The data input buffer is composed of sixteen 32 bit wide registers of which upper 16 bit is left channel data and lower is right channel data.

The CDIF receive the serial data from CDAI pin and store the data into the buffer through the serial to parallel register. Whenever the half of buffer is filled, the receive interrupt is generated. Only the half of input buffer can be accessible through the CDDI_0 to the CDDI_3.

Table 19.2 CDIF Register Map (Base Address = 0xF0059080)

Name	Address	Type	Reset	Description
CDDI_0	0x80	R		CD Digital Audio Input Register 0
CDDI_1	0x84	R		CD Digital Audio Input Register 1
CDDI_2	0x88	R		CD Digital Audio Input Register 2
CDDI_3	0x8C	R		CD Digital Audio Input Register 3
CICR	0x90	R/W	0x0000	CD Interface Control Register

CD Data Input (CDDI0)

0xF0059080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Left Channel Data															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Right Channel Data															

CD Data Input (CDDI1)

0xF0059084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Left Channel Data															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Right Channel Data															

CD Data Input (CDDI2)

0xF0059088

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Left Channel Data															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Right Channel Data															

CD Data Input (CDDI3)

0xF005908C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Left Channel Data															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Right Channel Data															

CD Interface Control Register (CICR)

0xF0059090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								EN	Reserved				BS	MD	BP

EN [7]	CDIF Enable
0	Disable CDIF
1	Enable CDIF
BS [3:2]	CDIF Bit Clock select
00	64fs
01	32fs
10	48fs
MD [1]	Interface Mode select
0	Select IIS format
1	Select LSB justified format
BP [0]	CDIF Bit Clock Polarity
0	Set that data is captured at positive edge of bit clock
1	Set that data is captured at negative edge of bit clock

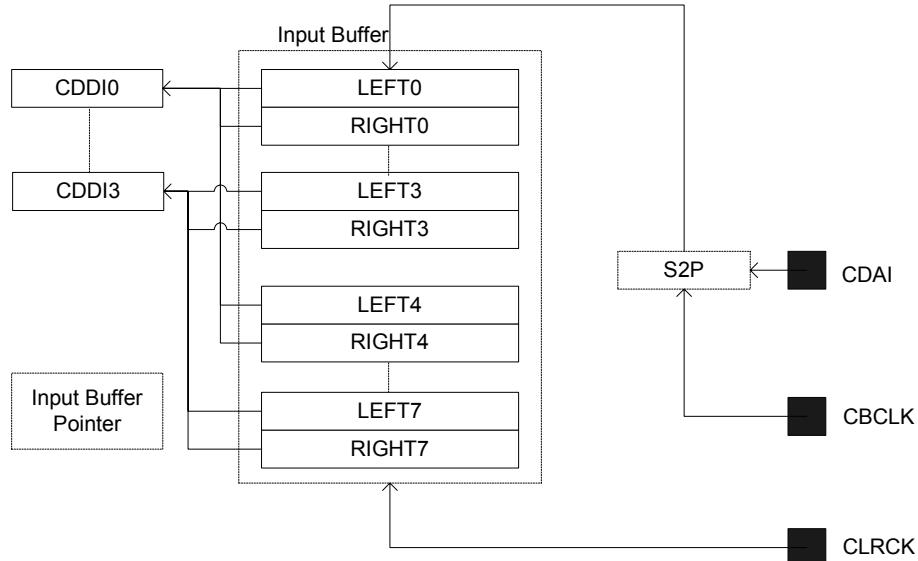
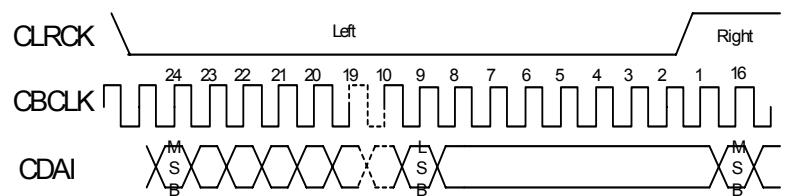
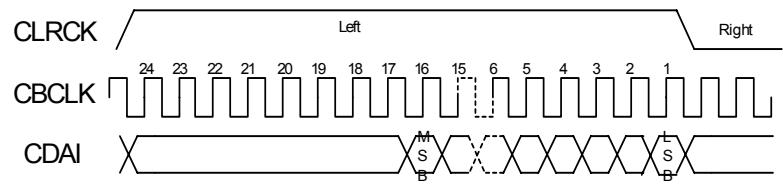


Figure 19.3 CDIF Block Diagram



MD=0 (IIS mode), BP=0, CBCLK=48fs



MD=1 (LSB justified mode), BP=0, CBCLK=48fs
Figure 19.4 CDIF Bus Timing Diagram

20 GPSB (GENERAL PURPOSE SERIAL BUS)

20.1 Functional Description

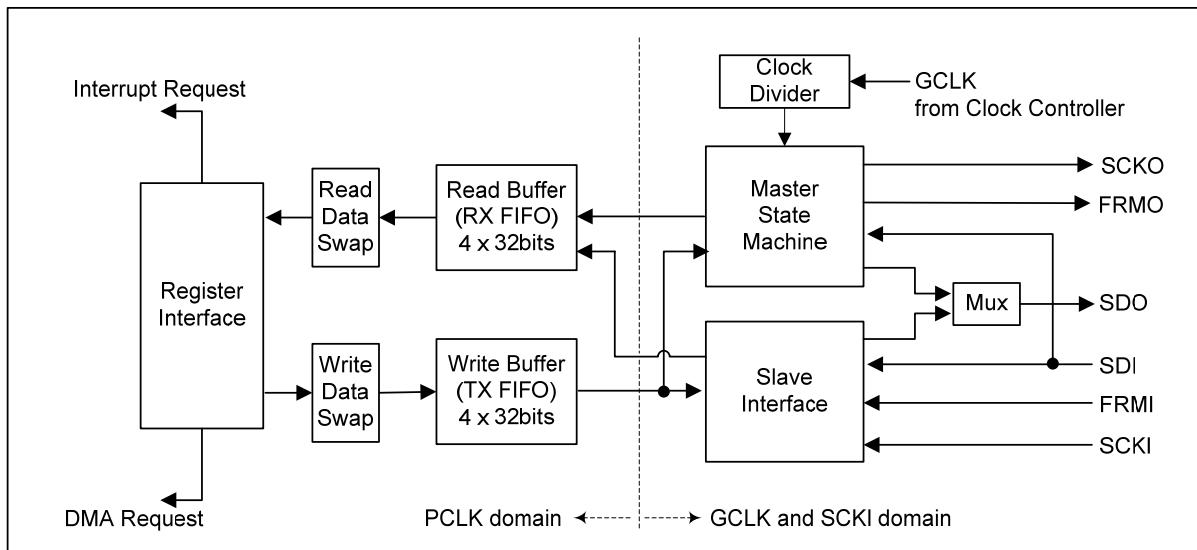


Figure 20.1 GPSB Interface Block Diagram

The Figure 20.1 shows the block diagram of GPSB interface circuit. The following figure shows the overall hardware block diagram for GPSB.

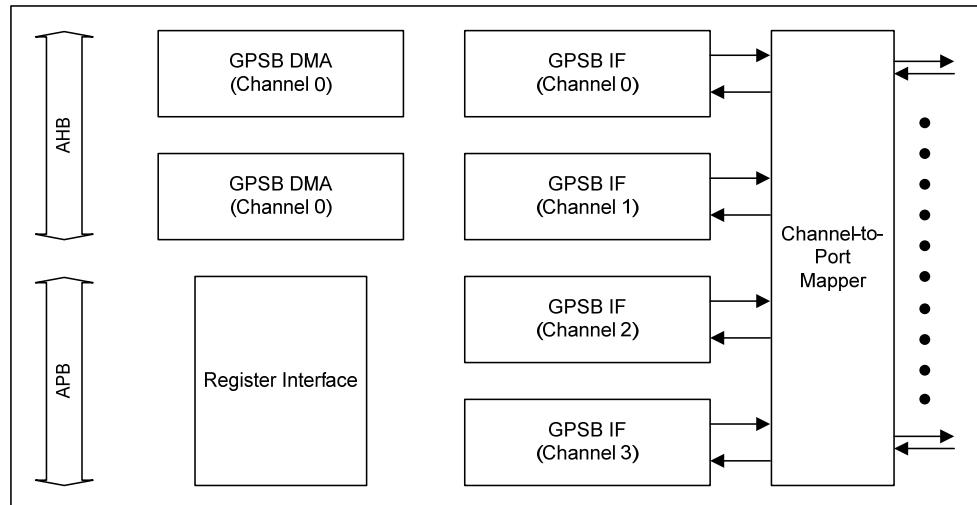


Figure 20.2 Overall GPSB Block Diagram

The TCC79XX provides 4 channel General Purpose Serial Bus Controller, which can be configured as a SPI compatible master/slave and MPEG-2 TS interface. The channel 0 and 1 have the DMA function which can transfer the bulk data. The channel 2 and 3 can interface the general purpose DMA for fast and bulk data transfer. The GPSB DMA can be used for MPEG2-TS serial interface with PID matching function.

The “Channel-to-Port Mapper” is for mapping the hardware channel and external ports. Each channel can be mapped to one of the ports.

20.2 Feature

The feature of GPSB in TCC79XX supports Motorola SPI, TI SSP compatible timing and changes the programmable master / slave configuration, timing parameters, data bit width, shift direction, polarity, data swap and so on.

The frequency of the SCKO pin is determined by the Clock Divider block which uses GCLK as clock source. The GCLK is generated by the Clock Controller Module.

20.3 Register Description

Table 20.1 GPSB Register Map (Base Address = 0xF0057000)

Ch	Name	Addr. Offset	Type	Reset	Description
Channel 0	PORT	0x000	R/W	0x0000	Data port
	STAT	0x004	R/W	0x0000	Status register
	INTEN	0x008	R/W	0x0000	Interrupt enable
	MODE	0x00C	R/W	0x0000	Mode register
	CTRL	0x010	R/W	0x0000	Control register
	TXBASE	0x020	R/W	0x0000	TX base address register
	RXBASE	0x024	R/W	0x0000	RX base address register
	PACKET	0x028	R/W	0x0000	Packet register
	DMACTR	0x02C	R/W	0x0000	DMA control register
	DMASTR	0x030	R/W	0x0000	DMA status register
Channel 1	DMAICR	0x034	R/W	0x0000	DMA interrupt control register
	PORT	0x100	R/W	0x0000	Data port
	STAT	0x104	R/W	0x0000	Status register
	INTEN	0x108	R/W	0x0000	Interrupt enable
	MODE	0x10C	R/W	0x0000	Mode register
	CTRL	0x110	R/W	0x0000	Control register
	TXBASE	0x120	R/W	0x0000	TX base address register
	RXBASE	0x124	R/W	0x0000	RX base address register
	PACKET	0x128	R/W	0x0000	Packet register
	DMACTR	0x12C	R/W	0x0000	DMA control register
Channel 2	DMASTR	0x130	R/W	0x0000	DMA status register
	DMAICR	0x130	R/W	0x0000	DMA status register
	PORT	0x200	R/W	0x0000	Data port
	STAT	0x204	R/W	0x0000	Status register
	INTEN	0x208	R/W	0x0000	Interrupt enable
	MODE	0x20C	R/W	0x0000	Mode register
	CTRL	0x210	R/W	0x0000	Control register
	PORT	0x300	R/W	0x0000	Data port
	STAT	0x304	R/W	0x0000	Status register
	INTEN	0x308	R/W	0x0000	Interrupt enable
Channel 3	MODE	0x30C	R/W	0x0000	Mode register
	CTRL	0x310	R/W	0x0000	Control register
Port Config	PCFG	0x800	R/W	0x03020100	Port Configuration Register
PID Table	PIDT	0xF00	R/W	-	

PORT Register (PORT)

0xF0057n³⁹00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Data Read / Write Port. Any data written to this register is sent to the write(TX) FIFO. Any data read from this register is from the read(RX) FIFO. Read data is valid only if RBVCNT is not zero. Although this port can be written even when the core disabled, no serial bus cycle is generated until the core is enabled with valid parameters.

³⁹ n = Channel Number 0 ~ 3

Status Register (STATUS)

0xF0057n⁴⁰04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
WBVCNT																
RBVCNT																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-								WOR	RUR	WUR	ROR	RF	WE	RNE	WTH	RTH

Field	Name	Reset	RW	Description
31 ~ 28	-	-	-	Undefined
27 ~ 24	WBVCNT	-	RW	Write(Transmit) FIFO valid entry count Maximum value dependent on the data bit width
23 ~ 20	-	-	-	Undefined
19 ~ 16	RBVCNT	-	RW	Read(Receive) FIFO valid entry count Maximum value dependent on the data bit width
15 ~ 9	-	-	-	Undefined
8	WOR	-	R/C	Write FIFO over-run error flag When writing 1, it is cleared. If INTEN.RC = 1, it is also cleared when reading it.
7	RUR	-	R/C	Read FIFO under-run error flag When writing 1, it is cleared. If INTEN.RC = 1, it is also cleared when reading it.
6	WUR	-	R/C	Write FIFO under-run error flag When writing 1, it is cleared. If INTEN.RC = 1, it is also cleared when reading it.
5	ROR	-	R/C	Read FIFO over-run error flag When writing 1, it is cleared. If INTEN.RC = 1, it is also cleared when reading it.
4	RF	-	R	Read FIFO full flag
3	WE	-	R	Write FIFO empty flag
2	RNE	-	R	Read FIFO not empty flag
1	WTH	-	R	Write FIFO valid entry count is under threshold.
0	RTH	-	R	Read FIFO valid entry increased over threshold.

⁴⁰ n = Channel Number 0 ~ 3

Interrupt Enable Register (INTEN)

0xF0057n⁴¹08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DW	DR	--		SHT	SBT	SHR	SBR	-	CFGWTH		-		CFGRTTH		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RC		--											IRQEN		

Field	Name	RW	Reset	Description
31	DW	RW	0	DMA request enable for TX FIFO '1' for enable, '0' for disable
30	DR	RW	0	DMA request enable for RX FIFO '1' for enable, '0' for disable
27	SHT	RW	0	TX half-word swap in word
26	SBT	RW	0	TX byte swap in half-word
25	SHR	RW	0	RX half-word swap in word
24	SBR	RW	0	RX byte swap in half-word
22 ~ 20	CFGWTH	RW	0	Transmit FIFO threshold for interrupt/DMA request
18 ~ 16	CFGRTTH	RW	0	Receive FIFO threshold for interrupt/DMA request
				Clear status[8:0] at the end of read cycle.
15	RC	RW	0	When this bit is set as "1", status[8:0] is all cleared whenever GPSBSTAT register is read.
14 ~ 9	-	-	-	Undefined
8 ~ 0	IRQEN	RW	0	Interrupt enable signals [8] : TX FIFO over-run error [7] : RX FIFO under-run error [6] : TX FIFO under-run error [5] : RX FIFO over-run error [4] : RX FIFO full [3] : TX FIFO empty [2] : RX FIFO not empty [1] : Valid entry count of TX FIFO is under threshold [0] : Valid entry count of RX FIFO is more than threshold

⁴¹ n = Channel Number 0 ~ 3

Mode Register (MODE)

0xF0057n⁴²0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIVLDV								TRE	THL	TSU	PCS	PCD	PWD	PRD	PCK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRF	CWF	-	BWS				SD	LB	SDO	CTF	EN	SLV	MD		

Field	Name	RW	Reset	Description
31 ~ 24	DIVLDV	RW	0	Clock divider load value FSCKO = FGCLK / ((DIVLDV+1)*2)
23	TRE	RW	0	Master recovery time tRECV = (TRE + 1) * (SCKO period)
22	THL	RW	0	Master hold time tHOLD = (THL + 1) * (SCKO period)
21	TSU	RW	0	Master setup time tSETUP = (TSU + 1) * (SCKO period)
20	PCS	RW	0	Polarity control for CS(FRM) – Master Only '0' for active low (default) / must be '0' for SSP '1' for active high
19	PCD	RW	0	Polarity control for CMD(FRM) – Master only '0' for active low / must be '1' for SSP '1' for active high
18	PWD	RW	0	Polarity control for transmitting data – Master Only '0' for falling edge of SCK '1' for rising edge of SCK / must be '1' for SSP
17	PRD	RW	0	Polarity control for receiving data – Master only '0' for rising edge of SCK '1' for falling edge of SCK / must be '1' for SSP
16	PCK	RW	0	Polarity control for serial clock '0' for master SCKO start from "low" for SPI timing 0, and "high" for SSP timing '0' for slave SCKI not inverted For SPI timing 0 and 3
				'1' for master SCKO starts from "1" For SPI timing 2 and 3.
				'1' for slave

⁴² n = Channel Number 0 ~ 3

					SCKI inverted. For SPI timing 1, 2 and SSP timing
15	CRF	RW	0	Clear receive FIFO counter	
14	CWF	RW	0	Clear transmit FIFO counter	
12 ~ 8	BWS	RW	0	Bit width selection Data bit width == BWS + 1 Valid range for BWS is 7 ~ 31	
7	SD	RW	0	The FIFOs are configured according to BWS[4] as follows, "BWS[4]==1", 4x32 bit. "BWS[4]==0", 8x16bits.	
6	LB	RW	0	Data shift direction control '0' for shift left (MSB first)	
5	SDO	R/W	0	Data loop-back enable * SDO is feedback to SDI internally and the incoming data from external I/O is ignored. SDO output is not affected by this bit.	
4	CTF	RW	0	SDO output disable (slave mode only) '0' for enable '1' for disable	
3	EN	RW	0	Continuous transfer mode enable '0' for single mode, '1' for continuous mode	
2	SLV	RW	0	If set to continuous mode, the CS signal keeps the active state until that 'CTF' is cleared and transmit FIFO is empty. Operation enable bit '0' for disable '1' for enable	
1 ~ 0	MD	RW	0	Slave mode configuration '0' for master mode '1' for slave mode	
				Operation mode	
				"00" for SPI compatible "01" for SSP compatible "1x" reserved for future use	

Control Register (CTRL)

0xF0057n⁴³10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
LCW	LCR	-	CMDEND								-	CMDSTART							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
-		RDSTART								PLW	-	PSW							

Field	Name	RW	Reset	Description
31	LCW	RW	0	Last clock disable for write cycle '1' for enable
30	LCR	RW	0	Last clock disable for read cycle '1' for enable
29	-	-	-	Undefined
28 ~ 24	CMDEND	RW	0	Command end position
20 ~ 16	CMDSTART	RW	0	Command start position
12 ~ 8	RDSTART	RW	0	Read data start position
7	PLW	RW	0	Polarity control for write command
6 ~ 5	-	-	-	Undefined
4 ~ 0	PSW	RW	0	Write command position

⁴³ n = Channel Number 0 ~ 3

TX Base Register (TX base)

0xF0057m⁴⁴20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TX_BASE [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_BASE [15:0]															

RX Base Register (RX base)

0xF0057m 24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RX_BASE [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_BASE [15:0]															

Packet Register (PACKET)

0xF0057m 28

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

Field	Name	RW	Reset	Description
28 ~ 16	COUNT	RW	0	Packet number information (COUNT + 1)
12 ~ 0	SIZE	RW	0	Packet size information

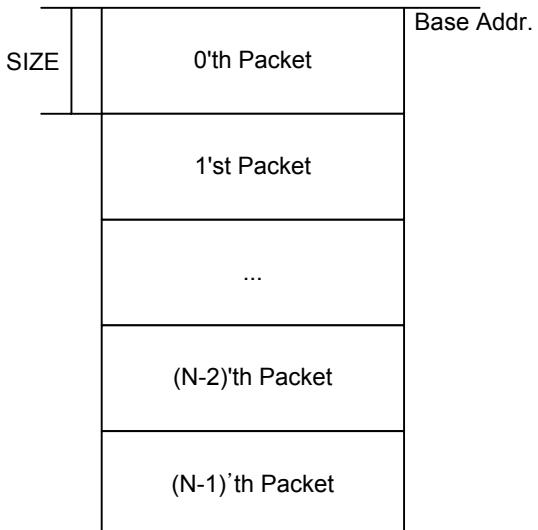


Figure 20.3 Packet Structure

⁴⁴ m = Channel Number 0 ~ 1

DMA Control Register (DMA CTRL)

0xF0057m⁴⁵2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTE	DRE	CT	END									MP	MS		TXAM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXAM									MD		-	PCLR	-	EN

Field	Name	RW	Reset	Description
31	DTE	RW	0	Transmit DMA request enable '1' for enable
30	DRE	RW	0	Receive DMA request enable '1' for enable
29	CT	RW	0	Continuous mode enable (Master) '1' for enable
28	END	RW	0	Byte endian mode register '0' for little-endian '1' for big-endian
17 ~ 16	TXAM	RW	0	<i>If the byte or half-word transfer mode of GPSB mode were used, this field should be '1'. The value of '0' is allowed for word transfer only.</i>
19	MP	RW	0	PID match mode register '1' for enable (MPEG2-TS mode)
18	MS	RW	0	Sync byte match control register '1' for enable (MPEG2-TS mode)
15 ~ 14	RXAM	RW	0	TX addressing mode '0' : Multiple Packet '1' : Fixed address (base) '2', '3' : Single Packet
5 ~ 4	MD	RW	0	RX addressing mode '0' : Multiple Packet '1' : Fixed address (base) '2', '3' : Single Packet
2	PCLR	W	0	DMA mode register Clear TX/RX packet counter
0	EN	RW	0	DMA enable register '1' for enable

⁴⁵ m = Channel Number 0 ~ 1

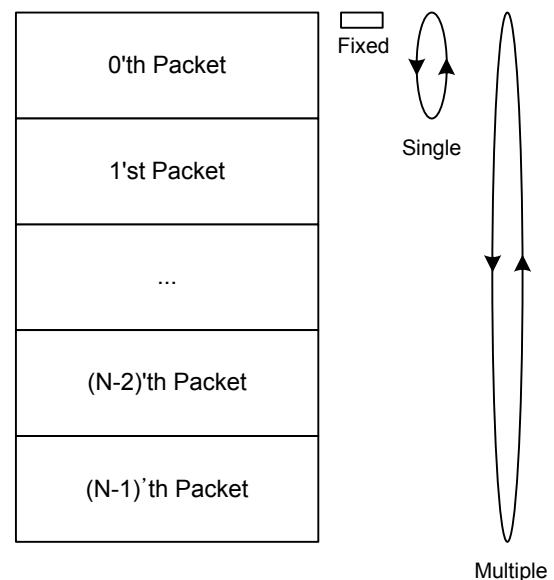


Figure 20.4 TX/RX Addressing Modes

DMA STATUS Register (DMA_STAT)

0xF0057m⁴⁶30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															TXPCNT

Field	Name	RW	Reset	Description
29 ~ 17	RXPCNT	R	0	Receive packet count register
12 ~ 0	TXPCNT	R	0	Transmit packet count register

DMA IRQ Register (DMAICR)

0xF0057m34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-		ISD	ISP												IRQS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															IRQPCNT

Field	Name	RW	Reset	Description
29	ISD	R/C	0	IRQ status for “Done Interrupt” When the number of packets which DMA transfers/receives is equal to the number of packets specified in PACKET register, it is issued. Therefore, it is not valid in continuous mode. When writing 1, it is cleared.
28	ISP	R/C	0	IRQ status for “Packet Interrupt”. It is issued every IRQPCNT packets which DMA transfers/receives. When writing 1, it is cleared.
20	IRQS	R/W	0	IRQ select register '0' for receiving '1' for transmitting
17	IED	R/W	0	IRQ enable for “Done Interrupt”
16	IEP	R/W	0	IRQ enable for “Packet Interrupt”
12 ~ 0	IRQPCNT	R/W	0	IRQ packet count register

⁴⁶ m = Channel Number 0 ~ 1

PID Table (PIDT)

0xF0057F00 ~ 0xF0057F80

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1	CH0														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															
PID															

Field	Name	RW	Reset	Description
31	CH1	R/W	X	Channel 1 enable
30	CH0	R/W	X	Channel 0 enable
12 ~ 0	PID	R/W	X	PID value

- Before starting the PID matching, the CH1 and CH0 fields of all the table entries should be initialized.

Port Configuration Register (PCFG)

0xF0057800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1															
CH0															

Field	Name	RW	Reset	Description
31 ~ 24	CH3	R/W	3	Channel 3 port mapping register
23 ~ 16	CH2	R/W	2	Channel 2 port mapping register
15 ~ 8	CH1	R/W	1	Channel 1 port mapping register
7 ~ 0	CH0	R/W	0	Channel 0 port mapping register

- The port map value for each channel should be different.

Channel IRQ Status Register (CIRQST)

0xF0057804

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								0	ISTC3	0	ISTC2	ISTD1	ISTC1	ISTD0	ISTC0

Field	Name	RW	Reset	Description
6	ISTC3	R	0	GPSB core IRQ status register for channel 3
4	ISTC2	R	0	GPSB core IRQ status register for channel 2
3	ISTD1	R	0	GPSB DMA IRQ status register for channel 1
2	ISTC1	R	0	GPSB core IRQ status register for channel 1
1	ISTD0	R	0	GPSB DMA IRQ status register for channel 0
0	ISTC0	R	0	GPSB core IRQ status register for channel 0

* If IRQ status of each channel were cleared, the corresponding field would be read '0'.

* The IRQ mode of GPSB channel is preferred to "level-trigger" mode.

20.4 GPSB Timing Diagram

The table and figures on the following page are the examples of GPSB MODE values to be programmed for SPI and SSP interface.

Timing	CPOL/CPHA	Master	Slave
	PWD,PRD,PCK		
SPI timing 0	00	000	0
SPI timing 1	01	110	1
SPI timing 2	10	001	1
SPI timing 3	11	111	0
SSP timing	NA	110	1

The figures below show the timing of each case listed in the table DIVLDV = 0 (GCLK / 2), Single Transfer Mode assumed

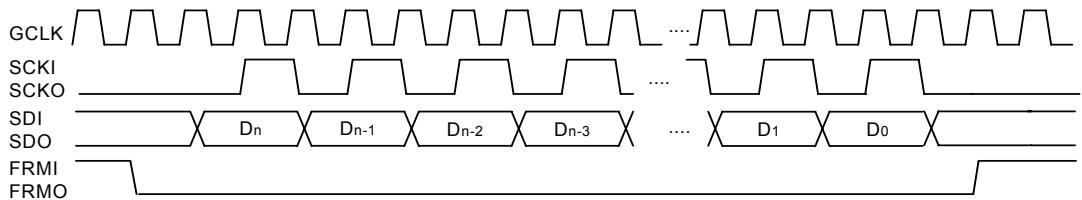


Figure 20.5 SPI Timing 0

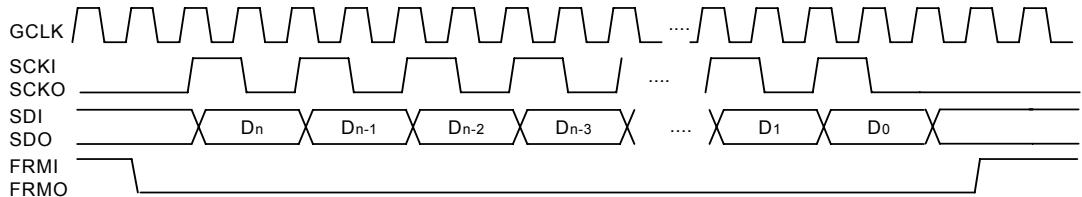


Figure 20.6 SPI Timing 1

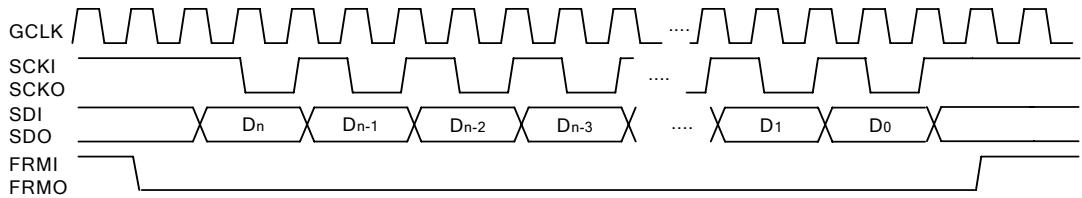


Figure 20.7 SPI Timing 2

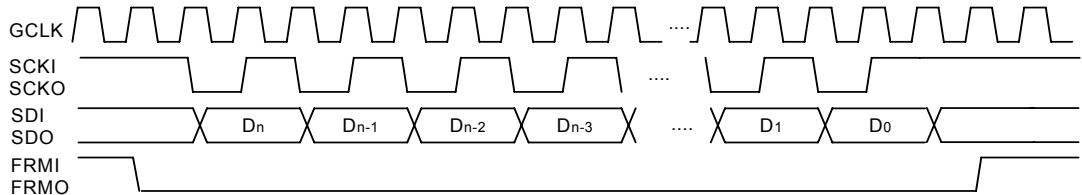


Figure 20.8 SPI Timing 3

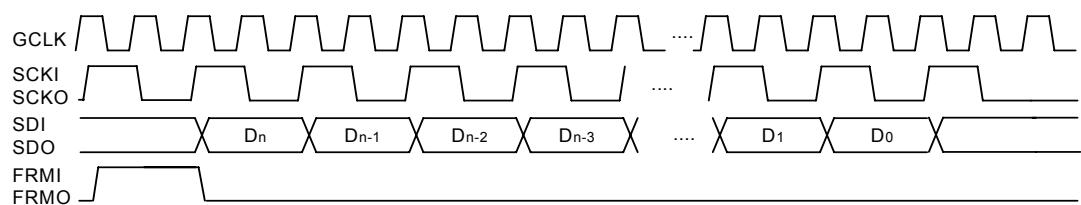


Figure 20.9 SSP Timing

20.5 MPEG2-TS Interface

The figures on the following page are the examples of MPEG2-TS bit stream. The packet of MPEG2-TS (Transport Stream) is composed of two groups that are the 188-bytes Payload data and 16bit Parity data. The Payload data including TS header within Sync byte is shown below Figure 20.10.

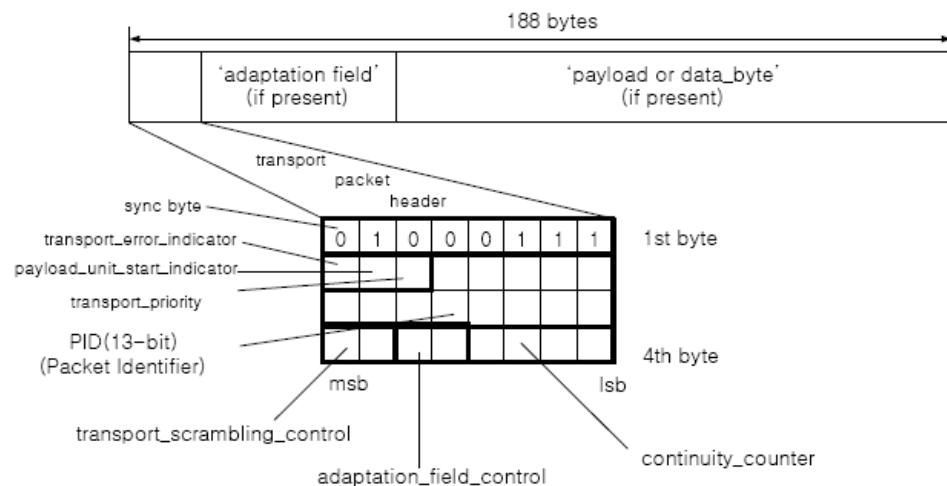


Figure 20.10 Bitstream of MPEG2-TS

In the Figure 20.10, the first byte of TS packet header is '0x47'. This sync byte arranges the serial stream data that is processed, stored, and read.

The input of MPEG2-TS interface is TS_CLK and TS packet data. TS packet data is composed in 204-byte unit and is separated by SYNC signal. Figure 20.11 shows the TS timing information.

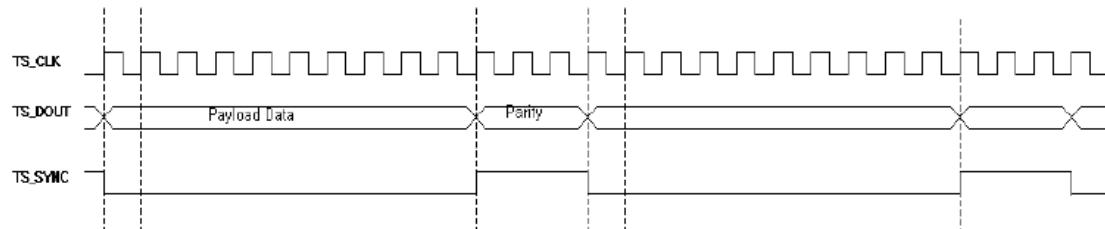


Figure 20.11 MPEG2-TS timing information

21 SPDIF TRANSMITTER

21.1 Overview

The SPDIF (or AES/EBU, IEC950 standards) is a point-to-point protocol for serial transmission of digital audio through a single transmission line. The transmission medium can be either electrical or optical (e.g. TosLink). It provides two channels for audio data, a method for communicating control information, and some error detection capabilities. The control information is transmitted as one bit per sample and accumulates in a block structure. The data is by-phase encoded, which enables the receiver to extract a clock from the data. Coding violations, defined as preambles, are used to identify sample and block boundaries.

The SPDIF interfaces are found on most CD/DVD players, audio equipment and computer sound cards.

The transmitter architecture is shown below.

The sub-frame assembler creates 32bit data words from sample data, register settings and optionally channel status/user data buffers. A parity bit is added in each sub-frame. The frame assembler adds preambles to create a frame of two sub-frames. 192 frames add up to a block. The block structure is bi-phase encoded before transmitting.

The size of the sample buffer is 4words. The sample buffer is addressed by setting the most significant address (0x00000200) bit to '1'. The sample buffer is divided in two equal parts, lower and upper, and the user will be notified when either is emptied of audio data.

The channel status can be generated from a dedicated 192bit buffer. Two interrupts can be generated when the transmitter reads from the buffer, one in the middle and one at the end. The user data buffer operates in an identical way.

Table 21.1 SPDIF Register Map (Base Address = 0xF005C000)

Name	Address	Type	Reset	Description
TxVersion	0x00	R		Version Register
TxConfig	0x04	R/W		Configuration Register
TxChStat	0x08	R/W		Channel Status Control Register
TxIntMask	0x0C	R/W		Interrupt Mask Register
TxIntStat	0x10	R/W		Interrupt Status Register
UserData	0x80~0xDC	W	-	User Data Buffer
ChStatus	0x100~0x15C	W	-	Channel Status Buffer
TxBuffer	0x200~0x210	W	-	Transmit Data Buffer
DMACFG	0x400	R/W	-	Additional Configuration for DMA

21.2 Register Descriptions

SPDIF Transmitter Version Register 0xF005C000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CSB	UDB			AW				DW			VER			

CSB [12]	Channel Status Buffer Available Bit
0	Channel status buffer is not available
1	Channel status buffer is available
UDB[11]	User Data Buffer Available Bit
0	User data buffer is not available
1	User data buffer is available
AW[10:5]	Value of Address Width
n	Value of Address Width
DW[4]	Value of Data Width
n	Value of Data Width
VER[3:0]	Design Version
n	Version Number

SPDIF Transmit Configuration Register

0xF005C004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				0	MODE									0	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					RATIO				UDATEN	CHSTEN	0	IEN	TXD	TXEN	

MODE[23:20]		Sample Format Mode Indicator
n		16 + n Bits Transmit Sample Format
9 ~ 15		Reserved
RATIO[15:8]		Clock Divider Ratio
n		Clock divider for the transmit frequency. The SPDIF clock is divided by a factor of (RATIO + 1) to generate the serial transmit clock.
UDATEN[7:6]		User Data Enable Bits
0		User data A & B set to 0.
1		User data A & B generated from UserData bit 7-0
2		User data A generated from UserData bit 7-0, B generated from UserData bit 15-8
3		Reserved
CHSTEN[5:4]		Channel Status Enable Bits
0		Channel status A & B generated from TxChStat
1		Channel status A & B generated from ChStat bit 7-0
2		Channel status A generated from ChStat bit 7-0, B generated from ChStat bit 15-8
3		Reserved
IEN[2]		Interrupt Output Enable Bit
n		'1' for enabling the interrupt output.
<hr/>		
TXD[1]		Data Valid Bit
n		'1' for data being valid.
<hr/>		
TXEN[0]		Transmitter Enable Bit
n		'1' for enabling the transmission.

SPDIF Transmit Channel Status Register

0xF005C008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
0																			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0								FREQ		0		GSTS		PRE		CPY		AU	

FREQ[7:6]		Sampling Frequency
0	44.1kHz	
1	48kHz	
2	32kHz	
3	Sample Rate Converter	
GSTS[3]		Status Generation
0	No indication	
1	Original/Commercially Pre-recorded data	
PRE[2]		Pre-emphasis
0	None	
1	50/15us	
CPY[1]		Copyright
0	Copy inhibited	
1	Copy permitted	
AU[0]		Data Format
0	Audio Format	
1	Data Format	

SPDIF Transmit Interrupt Mask Register

0xF005C00C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												HCSB	LCSB	HSB	LSB

BIT	NAME	DESCRIPTION
4	HCSB	Higher Channel Status/User Data Buffer Empty '1' for enable for higher channel status/user data buffer empty interrupt
3	LCSB	Lower Channel Status/User Data Buffer Empty '1' for enable for lower channel status/user data buffer empty interrupt
2	HSB	Higher Data Buffer Empty '1' for enable for higher data buffer empty interrupt
1	LSB	Lower Data Buffer Empty '1' for enable for lower data buffer empty interrupt

SPDIF Transmit Interrupt Status Register

0xF005C010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												HCSB	LCSB	HSB	LSB

Bit	Name	Description
4	HCSB[4]	Higher Channel Status/User Data Buffer Empty '1' for higher channel status/user data buffer empty interrupt activated
3	LCSB[3]	Lower Channel Status/User Data Buffer Empty '1' for lower channel status/user data buffer empty interrupt activated
2	HSB[2]	Higher Data Buffer Empty '1' for higher data buffer empty interrupt activated
1	LSB[1]	Lower Data Buffer Empty '1' for lower data buffer empty interrupt activated

SPDIF Transmit User Data Buffer Register

0xF005C080 ~ 0xF005C0DC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHBUD								CHAUD							

The user data is 24 bytes per sample block. Bit 0 of byte 0 is transmitted first. Bit 7 of byte 23 is transmitted last.

Bit	Name	Description
15:8	CHBUD[15:8]	User Data for Channel B 8 Bits User Data for Channel B
7:0	CHAUD[7:0]	User Data for Channel A 8 Bits User Data for Channel A

SPDIF Transmit Channel Status Buffer Register

0xF005C100 ~ 0xF005C15C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHBCS								CHACS							

The channel status data is 24 bytes per sample block. Bit 0 of byte 0 is transmitted first. Bit 7 of byte 23 is transmitted last.

bit	name	description
15:8	CHBCS[15:8]	Channel Status for Channel B 8 Bits Channel Status for Channel B
7:0	CHACS[7:0]	Channel Status for Channel A 8 Bits Channel Status for Channel A

SPDIF Transmit Sample Data Buffer Register

0xF005C200 ~ 0xF005C3FC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATH								DATL							

The format of data words in transmit sample buffer. Channel A is transmitted first, and must be stored on even addresses, while channel B is stored on odd addresses in the sample buffer.

BIT	name	description
23:16	DATH	Upper 8 Bits for Sample Buffer Data Audio data if > 16 bits resolution. Unused bits are 0
15:0	DATL	Lower 16 Bits for Sample Buffer Data Audio data (16 bits mode). Bits 0 is LSB

DMA Configuration (DMACFG)

0xF005C400

Field	Name	RW	Reset	Description
31 ~ 21	Reserved	R	0	Reserved
20 ~ 16	VCNT	R	X	FIFO Valid Entry Count
15 ~ 12	Reserved	R	0	Reserved
11	DRQEN1	RW	0	DMA Request Enable for User Data Buffer
10	DRQEN0	RW	0	DMA Request Enable for Sample Data Buffer
9	AMODE1	RW	0	Sample Data Buffer Address Mode. Buffer would be written with sequence below. (0, 2, 4, 6, 1, 3, 5, 7)
8	AMODE0	RW	0	0 : Ignore Address (FIFO Mode) 1 : Enable Address (16 Entries can be written with address)
7	FIFOCLR	RW	0	Clear FIFO. Should be written with "0" for normal operation
6 ~ 4	Reserved	R	0	Reserved
3 ~ 0	FIFOTH	RW	X	FIFO Threshold for DMA Request DMA request will be asserted if VCNT <= FIFOTH

21.3 Operation & Timing Diagram

Resetting

Except for the system reset, the transmitter can be disabled by clearing the TXEN bit in the TxConfig register.

Selecting transmit data rate

The data rate of the SPDIF signal is a function of the SPDIF clock from CKC and the RATIO bits in the TxConfig register. The bit rate is 64 times of the sampling frequency – each sample is encoded as 32 bits and there are two channels. Sample frequency is given by the following equation:

$$S_{FREQ} = C_{SPDIF} / (128 \times (RATIO + 1));$$

Example : Data rate is 48kHz and C_{SPDIF} is 12.288MHz. The RATIO bits must then be set to 1. Output data rate is 3.072Mbps.

Selecting data format

Any sample resolution from 16 to 24 bit can be transmitted. Data format is selected by the MODE bits in the TxConfig register.

Setting up channel status bits

If output format is standard consumer audio, set CHSTEN to 0 in TxConfig, and set TxChStat to desired format. Otherwise set up the ChStatus buffer with desired channel status data (192bits). If the channel status bits do not change from blocks to blocks, it is only necessary to program the buffer once. Otherwise the HCSBF/LCSBF bits in TxIntMask must be set, and the buffer will need to be updated when every half block is transmitted.

Setting up user data bits

User data bits are normally set to zero, but if required user data can be transmitted using the UserData buffer. If the user data bits do not change from block to block, it is only necessary to program the buffer once. Otherwise the HCSBF/LCSBF bits in TxIntMask must be set, and the buffer will need to be updated every block.

Preparing sample buffer

Before the TXDATA bit in TxConfig is set, fill up the complete sample buffer with audio data. The transmitter will generate an interrupt when lower half or upper half of sample buffer is emptied.

Start transmission

Transmission of SPDIF signal starts when the TXEN bit in TxConfig is set. If TXDATA bit is not set, the transmitted data will be all zeros with the sub-frame validity bit set.

Once the TXDATA bit is set, audio data from the sample buffer will be transmitted and the validity bit is cleared.

22 UART

22.1 Overview

The TCC79XX has the 4-channel UART that can be used in programming the system software. Additionally, these channels can be used as the smart card interface.

The block diagram of UART is shown in Figure 22.1.

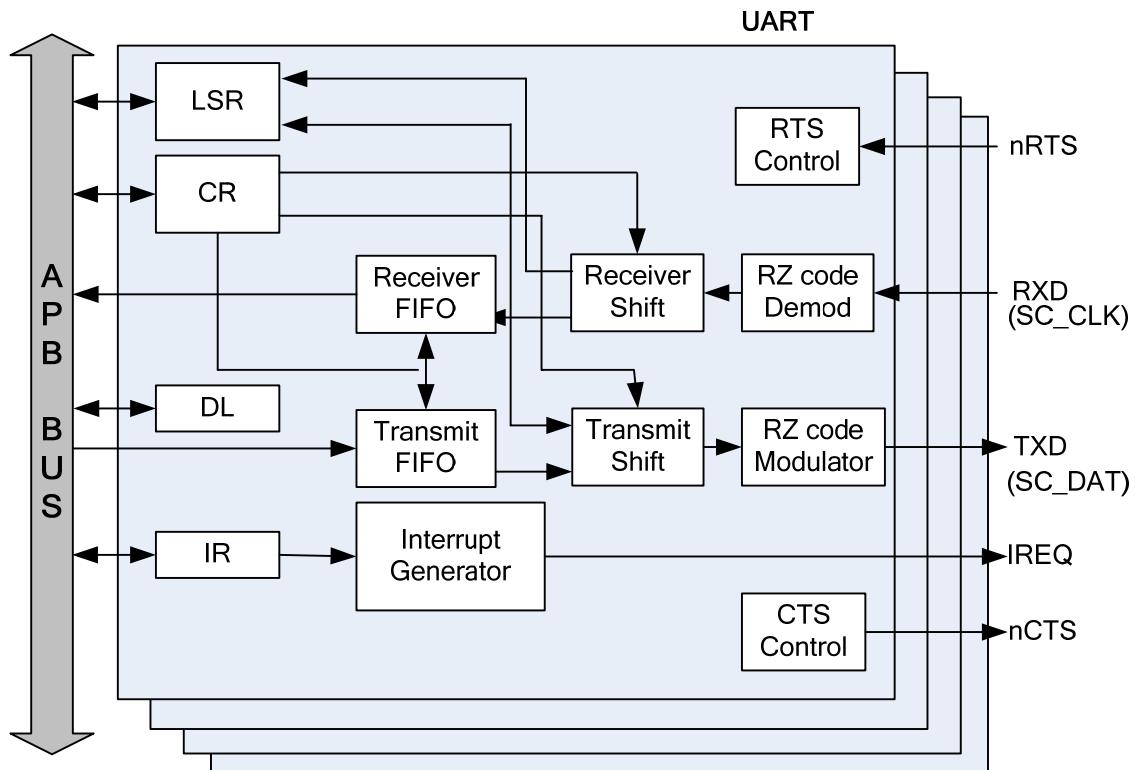
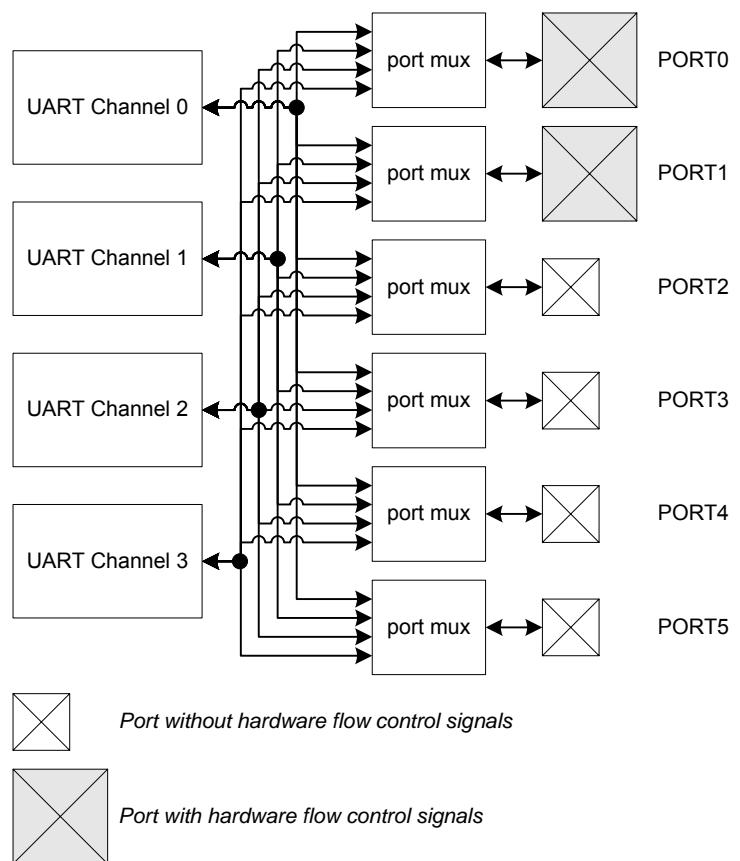


Figure 22.1 UART Block Diagram

Each channel can be connected to the one of 6 UART ports. UART port 0 and port 1 have flow-control signals, but UART port 3, 4, 5, and 6 does not have them. Although all channels have the hardware flow control function, channels should be mapped to UART port 0 or port 1 to use the hardware flow control function.

For smart card interface, RXD signal is used as the smart card clock (SC_CLK) and TXD signal is used as the smart card data (SC_DAT). All UART ports and channels are available for smart card interface.



22.2 Register Description

The base address of UART channel 0, 1, 2, and 3 are 0xF0055000, 0xF0055100, 0xF0055200, and 0xF0055300 respectively.

Table 22.1 UART Register Map

Name	Address	Type	Reset	Description
RBR	0x00	R	Unknown	Receiver Buffer Register(DLAB = 0)
THR	0x00	W	0x00	Transmitter Holding Register (DLAB=0)
DLL	0x00	R/W	0x00	Divisor Latch (LSB) (DLAB=1)
IER	0x04	R/W	0x00	Interrupt Enable Register (DLAB=0)
DLM	0x04	R/W	0x00	Divisor Latch (MSB) (DLAB=1)
IIR	0x08	R	Unknown	Interrupt Ident. Register (DLAB=0)
FCR	0x08	W	0xC0	FIFO Control Register (DLAB=1)
LCR	0x0C	R/W	0x03	Line Control Register
MCR	0x10	R/W	0x00	MODEM Control Register
LSR	0x14	R	Unknown	Line Status Register
MSR	0x18	R	Unknown	MODEM Status Register
SCR	0x1C	R/W	0x00	Scratch Register
AFT	0x20	R/W	0x00	AFC Trigger Level Register
UCR	0x24	R/W	0x00	UART Control Register
SRBR	0x40	R	Unknown	Rx Buffer Register
STHR	0x44	W	0x00	Transmitter Holding Register
SDLL	0x48	R/W	0x00	Divisor Latch (LSB)
SDLM	0x4C	R/W	0x00	Divisor Latch (MSB)
SIER	0x50	R/W	0x00	Interrupt Enable Register
SCCR	0x60	R/W	0x00	Smart Card Control Register
STC	0x64	R/W	0x00	Smart Card TX Count Register

The base address of “Port Mux Registers” is 0xF0055400.

Table 22.2 UART Port Mux Register

Name	Address	Type	Reset	Description
CHSEL	0x00	R/W	0x3210	Channel Selection Register
CHST	0x00	R	0x0000	Channel Status Register

22.3 UART Controller Register Map

Receiver Buffer Register (RBR) 0xF0055n⁴⁷00(DLAB=0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															
Received Data (when reading)															

The RBR is actually the 16byte FIFO, a received data from external device is stored in the RBR and CPU(or DMA) can read this register by Rx interrupt(or Rx DMA Request).

Transmitter Holding Register (THR) 0xF0055n00(DLAB=0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															
Transmitting Data (when writing)															

The THR is actually the 16byte FIFO. To transmit data to external device, CPU(or DMA) should write data to the THR.

Divisor Latch Register (DLL) 0xF0055n00(DLAB=1)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															
Divisor Latch LSB															

Divisor Latch Register (DLM) 0xF0055n04(DLAB=1)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															
Divisor Latch MSB															

This is for generation of the desired baud rate clock.

The value can be calculated as follows.

$$\{DLM, DLL\} = f_{UART} / (16 * \text{desired baud rate})$$

For example,

If UART clock frequency is 48MHz (from CKC) and the baud-rate you want is 115,200 bps, the divisor value should be 26(48M/ (115200 x16)) in decimal.

In UART interface with general DMA, the UART clock frequency (from CKC) and con-

⁴⁷ n = Channel Number 0 ~ 3.

figuration should be set by the following description.

- f_{UART} frequency
- CHCTRL.BST of GDMA = 0 (DMA Arbitration Mode Enable).

$$f_{UART} \times 4 \geq f_{BUS}$$

- CHCTRL.BST of GDMA = 1 (DMA Arbitration Mode Disable).

$$f_{UART} \times 3 \geq f_{BUS}$$

The following setting value(DMA/UART setting value) is example of RX/TX operation with DMA for the TCC79XX.

- GDMA Setting Value

CHCTRL.BSIZE = 0 : 1Read/1Write (every UART request).

CHCTRL.WSIZE = 0 : 8bit data (1 byte).

- UART Setting Value

FCR.TXT = 0 : TX Empty Condition in TX FIFO.

FCR.RXT = 0 : Rx Available data Condition (minimum 1 character available data is received in Rx FIFO).

Interrupt Register (IER)

0xF0055n⁴⁸04(DLAB=0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															EMS1

0

EMSI ELSI ETXI ERXI

EMS1 [3]	Type	Interrupt Enable Bit
0	R/W	Disable MODEM status interrupt
1	R/W	Enable MODEM status interrupt

ELSI [2]	Type	Interrupt Enable Bit
0	R/W	Disable receiver line status interrupt
1	R/W	Enable receiver line status interrupt

ETXI [1]	Type	Interrupt Enable Bit
0	R/W	Disable transmitter holding register empty interrupt
1	R/W	Enable transmitter holding register empty interrupt

ERXI [0]	Type	Interrupt Enable Bit
0	R/W	Disable received data available interrupt
1	R/W	Enable received data available interrupt

⁴⁸ n = Channel Number 0 ~3.

Interrupt Ident. Register (IIR)

0xF0055n⁴⁹08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				STF											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					0			0	0	0	0	IID[2:0]			IPF

STF [27]	Type	Smart Card TX done Flag
1	R	It represents that TX is done.

IID [3:1]	Type	Interrupt ID
011	R	Receiver line status
010	R	Received data available
110	R	Character timeout indication
001	R	Transmitter holding register empty
000	R	MODEM status

IID [3:1]	Type	Interrupt Source
011	R	Overrun/parity error/framing error/break error
010	R	Receiver data available or trigger level reached
110	R	While the last 4 characters are received, no characters have been removed from or input to the RX FIFO and there is at least 1 character in the RX FIFO.
001	R	Transmitter holding register empty
000	R	Clear to send data set ready or ring indicator or data carrier detect

IID [3:1]	Type	Interrupt Reset
011	R	Reading the line status register
010	R	Reading the receiver buffer register or the FIFO drops below the trigger level
110	R	Reading the receiver buffer register
001	R	Reading the IIR register(if source of interrupt) or writing into the transmitter holding register
000	R	Reading the MODEM status register

IPF [0]	Type	Interrupt Flag
0	R	Interrupt pending
1	R	Interrupt has not generated

⁴⁹ n = Channel Number 0 ~ 3.

FIFO Control Register (FCR)

0xF0055n⁵⁰08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								RXT[1:0]		TXT[1:0]		DRTE	TXFR	RXFR	FE

RXT [7:6]	Type	RX FIFO Trigger Level(bytes)
00	W	1 byte
01	W	4 bytes
10	W	8 bytes
11	W	14 bytes

TXT [5:4]	Type	TX FIFO Trigger Level(bytes)
00	W	Possible to Write 16 byte (EMPTY)
01	W	Possible to Write 8 byte
10	W	Possible to Write 4 byte
11	W	Possible to Write 1 byte

These bits are used to configure the writable amount at FIFO when the TX interrupt or GDMA TX request is generated.

DRTE [3]	Type	DMA Mode Select
0	W	DMA transfer is depend on RxDE or TxDI bit.
1	W	Enable both Rx & Tx DMA transfer (Regardless of RxDE or TxDI status)

TXFR [2]	Type	TX FIFO Reset
1	W	Reset TX FIFO counter and FIFO data

RXFR [1]	Type	RX FIFO Reset
1	W	Reset RX FIFO counter and FIFO data

FE [0]	Type	FIFO Enable
1	W	Enable TX FIFO and RX FIFO.

⁵⁰ n = Channel Number 0 ~ 3.

Line Control Register (LCR)

0xF0055n⁵¹0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								DLAB	SB	SP	EPS	PEN	STB	WLS[1:0]	

DLAB [7]	Type	Divisor Latch Access
1	R/W	Access the divisor latches of the baud generator
0	R/W	Access the receiver buff, the transmitter holding register, or the interrupt enable register.
SB [6]	Type	Set Break
1	R/W	The serial output is forced to the spacing(logic 0) state
0	R/W	Disable the break
SP [5]	Type	Stick Parity
1	R/W	When PEN[3], EPS[4], and SP[5] are set to 1, the parity bit is transmitted and checked as a logic 0. If PEN[3] and SP[5] are set to 1 and EPS[4] is set to 0, then the parity bit is transmitted and checked as a logic 1
0	R/W	Disable stick parity
EPS [4]	Type	Even Parity Select
1	R/W	Generate or check even parity
0	R/W	Generate or check odd parity
PEN [3]	Type	Parity Enable
1	R/W	A parity bit is generated(TX) or checked(RX)
STB [2]	Type	Number of Stop Bits
1	R/W	One stop bit is generated in the transmitted data
0	R/W	When 5-bit word length is selected, one and a half stop bits are generated. When either 6, 7, or 8-bit word length is selected, two stop bits are generated.
WLS [1:0]	Type	Word Length Select
00	R/W	5 bits
01	R/W	6 bits
11	R/W	7 bits
11	R/W	8 bits

⁵¹ n = Channel Number 0 ~ 3.

Modem Control Register (MCR)

0xF0055n⁵²10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							0		RS	AFE	LOOP	-	-	RTS	-

RS [6]	Type	RTS Deassert Condition Control Bit
0	R/W	nRTS is de-asserted at the Rx Stop Condition
1	R/W	nRTS is de-asserted at the Rx Start Condition (Reset value)
AFE [5]	Type	Hardware Auto Flow Control Enable Bit
0	R/W	Disable Automatic Flow Control
1	R/W	Enable Automatic Flow Control
LOOP [4]	Type	Loop Back Mode Enable
0	R/W	Disable local loop back feature
1	R/W	Enable local loop back feature
RTS [1]	Type	Request To Send
0	R/W	Set the nRTS line to high state
1	R/W	Reset the nRTS line to low state
AFE	RTS	Flow Control Configuration
1	1	Both nCTS and nRTS automatic flow control is activated
1	0	Only nCTS automatic flow control is activated
0	X	Both nCTS and nRTS automatic flow control is de-activated (Same as TCC77x series, only support for S/W Flow Control)

⁵² n = Channel Number 0 ~ 3.

Line Status Register (LSR)

0xF0055n⁵³14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								ERF	TEMPT	THRE	BI	FE	PE	OE	DR

ERF [7]	Type	Divisor Latch Access
1	R	In the FIFO mode this bit is set when there is at least one parity error, framing error or break indication in the FIFO
0	R	In the 16450 mode
TEMPT [6]	Type	Transmitter Empty
1	R	Transmitter holding register and the transmitter shift register are both empty
THRE [5]	Type	Transmitter Holding Register Empty
1	R	UART is ready to accept a new char for transmission
BI [4]	Type	Break Interrupt
1	R	The received data input is held in the spacing(logic 0) state for longer than a full word transmission time
FE [3]	Type	Framing Error
1	R	The received character did not have a valid stop bit
PE [2]	Type	Parity Error
1		The received data character does not have the correct even or odd parity
OE [1]	Type	Overrun Error
1	R	The receiver buffer register was not read by the CPU before the next character was transferred into the receiver buffer register.
DR [0]	Type	Data Ready
1	R	The receiver data ready

⁵³ n = Channel Number 0 ~ 3.

Modem Status Register(MSR)

0xF0055n⁵⁴18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								reserved				CTS	reserved		DCTS

CTS [4]	Type	Clear to Send
	R	This bit is the complement of the Clear to Send input
DCTS [0]	Type	Delta Clear to Send
	R	This bit is the Delta Clear to Send indicator.

Scratch Register (SCR)

0xF0055n1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Scratch Data[7:0]							

This 8-bit Read/Write Register does not control the UART in anyway. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.

AFC Trigger Level Register (AFT)

0xF0055n20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								ATL[3:0]				DTL[3:0]			

ATL	Type	[3:0]	nRTS Assert Trigger Level
N	R/W	nRTS assert trigger level	
DTL	Type	[3:0]	nRTS Deassert Trigger Level
N	R/W	nRTS de-assert trigger level	

⁵⁴ n = Channel Number 0 ~ 3.

Control Register (UCR)

0xF0055n⁵⁵24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							0					RWA	TWA	RxDE	TxDE

RWA	Type	[3]	Rx Word Access
0	R/W	Rx FIFO access to byte	
1	R/W	Rx FIFO access to word (4 bytes)	

TWA	Type	[2]	Tx Word Access
0	R/W	Tx FIFO access to byte	
1	R/W	Tx FIFO access to word (4 bytes)	

RxDE	Type	[1]	Rx DMA Enable
0	R/W	Rx DMA disable	
1	R/W	Rx DMA enable	

TxDE	Type	[0]	Tx DMA Enable
0	R/W	Tx DMA disable	
1	R/W	Tx DMA enable	

⁵⁵ n = Channel Number 0 ~ 3.

Receiver Buffer Register (SRBR)																0xF0055n ⁵⁶ 40
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Received Data (Read Only)
0																Received Data (Read Only)

Transmitter Holding Register (STHR)																0xF0055n44
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Transmitting Data (Write Only)
0																Transmitting Data (Write Only)

Divisor Latch Register (SDLL)																0xF0055n48
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Divisor Latch LSB
0																Divisor Latch LSB

Divisor Latch Register (SDLM)																0xF0055n4C
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Divisor Latch MSB
0																Divisor Latch MSB

Interrupt Register (SIER)																0xF0055n50
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	EMSI ELSI ETXI ERXI
0																EMSI ELSI ETXI ERXI

SRBR, STHR, SDLL, SDLM, SIER registers are copy of the RBR, THR, DLL, DLM, IER registers. These registers can be accessed without concern of DLAB state.

⁵⁶ n = Channel Number 0 ~ 3.

SmartCard Configuration Register (SCCR)

0xF0055n⁵⁷60

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEN	STEN	DEN	P	STF											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DIV [15:0]

SEN	Type	[31]	Smart Card Interface Enable
1	R/W		The smart card interface is enabled
STEN	Type	[30]	SmartCard Tx Enable
1	R/W		Enable
DEN	Type	[29]	SmartCard Clock Divider Enable
1	R/W		Enable
P	Type	[28]	SmartCard Clock Post Divider Enable
1	R/W		Enable
STF	Type	[27]	SmartCard Tx done Flag
1	R		It represents that all characters specified by S_CNT are transmitted. It can be TX done interrupt source.
1	W		When '1' is written to it, it is cleared.
STE	Type	[27]	SmartCard Tx done Interrupt Enable
1	R		Enable Tx done Interrupt
DIV	Type	[15:0]	SmartCard Clock Divisor
1	R/W		$F_{out} = F_{uart} / ((DIV+1) * 2^P)$

⁵⁷ n = Channel Number 0 ~ 3.

SmartCard Tx Count (STC)

0xF0055n64

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C_CNT[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S_CNT[15:0]															

C_CNT	Type	[31:16]	Current Tx Count
N	R	It represents current TX count when smart card interface is enabled	
S_CNT	Type	[15:0]	Start Tx Count
N	R/W	To transmit 1 character, bit7 of SCR register should be set to 1 and S_CNT should be set to 2. In case two or more characters are transmitted, S_CNT should be set to the number of the transmitted characters and bit7 of SCR register should be set to 0.	When smart card interface is enabled, it specifies TX Count except for transmitting 1 character.

22.4 Port Mux Register

This LSI has 6 UART ports, and each port can be allocated for one of UART channels. Base address is 0xF0055400

Channel Selection Register (CHSEL)																0xF0055400
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CH0
0	CH3				0	CH2				0	CH1				0	CH0

The value of CH0, CH1, CH2, and CH3 is the corresponding port number. Therefore, when CH0 is set to 3, UART channel0 is connected to UART port 3. Note that the value for each channel should be different.

Channel Status Register																0xF0055404
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CH3
																CH0

When one or more UART channels are issuing an interrupt request, the corresponding bit is set to 1.

23 CKC

23.1 Overview

The block diagram of CKC is shown in Figure 23.1.

The CKC block has 4 primary clock sources from 2 PLLs, XIN, and XTIN. Each PLL output clock can be divided by a corresponding clock divider. XIN and XTIN can also generate divided clocks. These 4 primary clock sources and 4 divided clock sources can be used for generating the CPU clock and the bus clock.

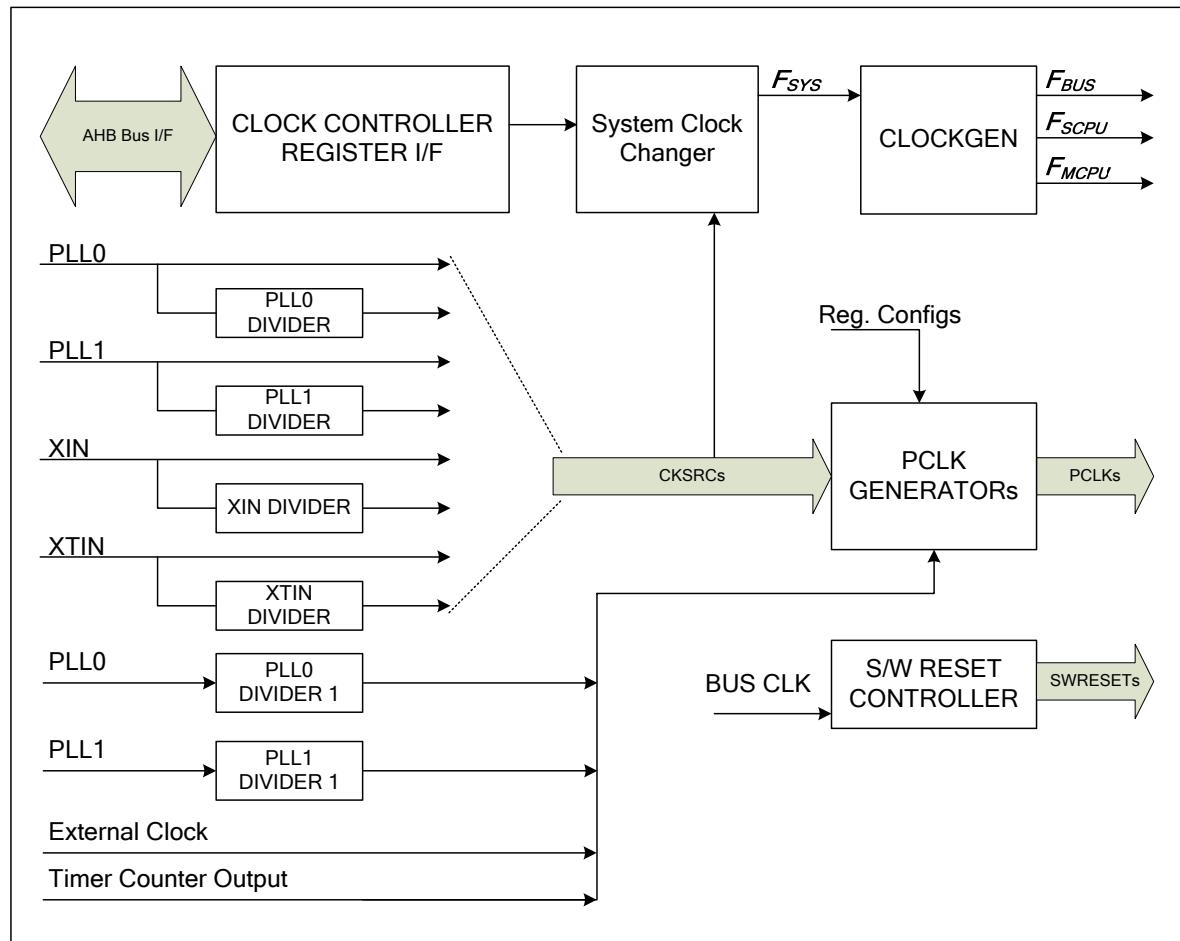


Figure 23.1 CKC Block Diagram

The output clock of safe clock changer, which generates glitch-free clock from 8 independent clock sources, can be used for making the main operating clocks (CPU clock and bus clock).

The peripheral clock generator makes the corresponding hardware clock from 4 timer/counter outputs, another two PLL divided clocks, external clock, and 48MHz clock from USB PHY as well as 4 primary clock sources and 4 divided clocks.

This processor has 2 operating modes, halt-mode and power-down mode. In the halt mode, the CPU clock and bus clock are masked until the FIQ or IRQ interrupt. In power-down mode, all the oscillator circuits are disabled and all the clocks are masked until the pre-defined external interrupt.

Table 23.1 CKC Register Map (Base Address = 0xF3000000)

Name	Address	Type	Reset	Description
CLKCTRL	0x00	R/W	0x800FF011	CPU & Bus Clock Control Register
PLL0CFG	0x04	R/W	0x00015003	PLL0 Configuration Register
PLL1CFG	0x08	R/W	0x00015003	PLL1 Configuration Register
CLKDIVC	0x0C	R/W	0x81828383	Clock Divider Configuration Register
CLKDIVC1	0x10	R/W	0x00000000	PLL1 Divider Configuration Register
MODECTR	0x14	R/W	0x00010000	Operating Mode Control Register
BCLKCTR0	0x18	R/W	0xFFFFFFFF	Bus Clock Mask Control Register for Group 0
BCLKCTR1	0x1C	R/W	0xFFFFFFFF	Bus Clock Mask Control Register for Group 1
SWRESET0	0x20	R/W	0x00800000	Software Reset Control Register for Group 0
SWRESET1	0x24	R/W	0x00000000	Software Reset Control Register for Group 1
WDTCTRL	0x28	R/W	0x00000000	Watchdog Control Register
PCK_USB11H	0x80	R/W	0x14000000	USB11H Clock Control Register
PCK_SDMMC	0x84	R/W	0x14000000	SD/MMC Clock Control Register
PCK_MSTICK	0x88	R/W	0x14000000	Memory Stick Clock Control Register
PCK_I2C	0x8C	R/W	0x14000000	I2C Clock Control Register
PCK_LCD	0x90	R/W	0x14000000	LCD Clock Control Register
PCK_CAM	0x94	R/W	0x14000000	Camera Clock Control Register
PCK_UART0	0x98	R/W	0x14000000	UART0 Clock Control Register
PCK_UART1	0x9C	R/W	0x14000000	UART1 Clock Control Register
PCK_UART2	0xA0	R/W	0x14000000	UART2 Clock Control Register
PCK_UART3	0xA4	R/W	0x14000000	UART3 Clock Control Register
PCK_TCT	0xA8	R/W	0x14000000	Clock Output via GPIO_A[3] during CLK_OUT1 mode
PCK_TCX	0xAC	R/W	0x14000000	Clock Output via GPIO_A[2] during CLK_OUT0 mode
PCK_TCZ	0xB0	R/W	0x14000000	Timer32 T-Clock Control Register
PCK_DAI	0xB4	R/W	0x0A000000	DAI Clock Control Register
PCK_GPSB0	0xB8	R/W	0x14000000	GPSB0 Clock Control Register
PCK_GPSB1	0xBC	R/W	0x14000000	GPSB1 Clock Control Register
PCK_GPSB2	0xC0	R/W	0x14000000	GPSB2 Clock Control Register
PCK_GPSB3	0xC4	R/W	0x14000000	GPSB3 Clock Control Register
PCK_ADC	0xC8	R/W	0x14000000	ADC Clock Control Register
PCK_SPDIF	0xCC	R/W	0x14000000	SPDIF TX Clock Control Register
PCK_RFREQ	0xD0	R/W	0x14000000	SDRAM Refresh Clock Control Register
PCK_SCALER	0xD4	R/W	0x14000000	CIF Scaler Clock Control Register
PCK_EHI0	0xD8	R/W	0x14000000	EHI0 Clock Control Register
PCK_EHI1	0xDC	R/W	0x14000000	EHI1 Clock Control Register
BCKVCFG	0xE0	R/W	0xFFFFFFFF	Bus Clock Mask Register for Video Block
MCLKCTRL	0xE8	R/W	0x8000000F	Main Processor Clock Control Register
SCLKCTRL	0xEC	R/W	0x8000000F	Sub Processor Clock Control Register
BCLKMSKE0	0xF0	R/W	0xFFFFFFFF	Bus Clock Mask Enable Register for Group 0
BCLKMSKE1	0xF4	R/W	0xFFFFFFFF	Bus Clock Mask Enable Register for Group 1
OPTION0	0xFC	R/W	0x00000000	Option Register

23.2 Register Descriptions

MCLKCTRL, SCLKCTRL Register

0xF30000E8, 0xF30000EC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACC								0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

0 XB 0 XDIV

Field	Name	RW	Reset	Description	
31	ACC	RW	1	Access Permission for MDIV(SDIV) and MB(SB) Register	
				0b : Can be accessed through M(S)CLKCTRL	
				1b : Can be accessed through CLKCTRL	
30 ~ 9	-	-	-	Undefined	
8	XB	RW	0	Same as MB or SB field of CLKCTRL Register	
7 ~ 4	-	-	-	Undefined	
3 ~ 0	XDIV	RW	0	Same as MDIV or SDIV field of CLKCTRL Register	

PLL0CFG & PLL1CFG Register

0xF3000004/008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PD						-									S
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

M

0

P

Field	Name	RW	Reset	Description
5 ~ 0	P	RW	0x03	PLL P value (p = P) $1 \leq P \leq 63$
15 ~ 8	M	RW	0xC0	PLL M value (m = M) $16 \leq M \leq 255$
18 ~ 16	S	RW	0x01	PLL S value (s = S) $0 \leq S \leq 5$
30 ~ 18	-	-	-	Undefined
31	PD	RW	0x1	PLL power down register

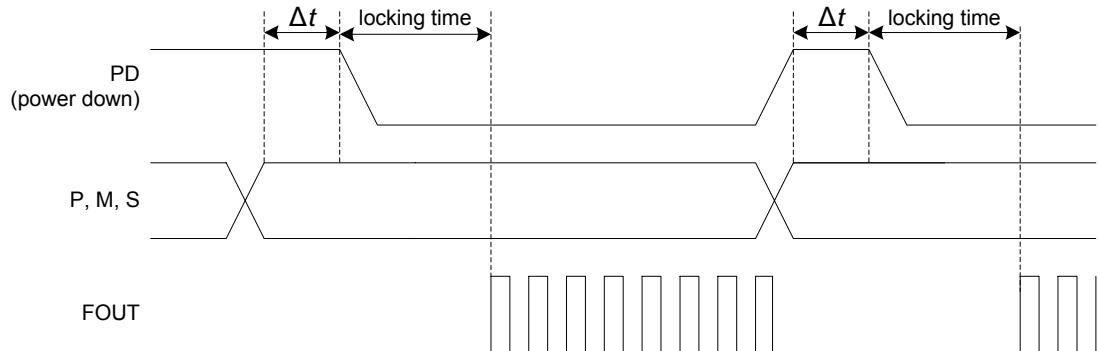
The PLL has the following constraints.

$$F_{VCO0} = (m * F_{IN}) / (p) : 250\text{MHz} \sim 600\text{MHz} \text{ (} F_{IN} \text{ is XIN oscillator)}$$

$$F_{PLL0} = F_{VCO0} / (2^s)$$

Whenever P, M, and S value are changed, the PD bit needs to be 1 for Δt ($>5\text{ns}$) and to be 0 again to restart the PLL with new setting values.

You must get the proper divider setting values from APPENDIX A.



$\Delta t > 5\text{ns}$

locking time $> 300\text{us}$

CLKDIVC Register

0xF300000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P0E	-							P1E	-						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XE	-							XTE	-						XTDIV

Field	Name	RW	Reset	Description
5 ~ 0	XTDIV	RW	0x3	XTIN divisor value FXTDIV = FXTIN / (XTDIV + 1) XTDIV should not be zero
6	-	-	-	Undefined
7	XTE	RW	0x1	XTIN divider enable register
13 ~ 8	XDIV	RW	0x3	XIN divisor value FXDIV = FXIN / (XDIV + 1) XDIV should not be zero
14	-	-	-	Undefined
15	XE	RW	0x1	XIN divider enable register
21 ~ 16	P1DIV	RW	0x2	PLL1 divisor value FP1DIV = FPLL1 / (P1DIV + 1) P1DIV should not be zero
22	-	-	-	Undefined
23	P1E	RW	0x1	PLL1 divider enable register
29 ~ 24	P0DIV	RW	0x1	PLL0 divisor value FP0DIV = FPLL0 / (P0DIV + 1) P0DIV should not be zero
30	-	-	-	Undefined
31	P0E	RW	0x1	PLL0 divider enable register

CLKDIVC1 Register

0xF3000010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P1E	-	P1DIV						P0E	-	P0DIV					

Field	Name	RW	Reset	Description
5 ~ 0	P1DIV	RW	0x0	PLL1 divisor value FP1DIV1 = FPLL1 / (P1DIV + 1) P1DIV should not be zero
6	-	-	-	Undefined
7	P1E	RW	0x0	PLL1 divider enable register
13 ~ 8	P0DIV	RW	0x0	PLL0 divisor value FP0DIV1 = FPLL0 / (P0DIV + 1) P0DIV should not be zero
14	-	-	-	Undefined
15	P0E	RW	0x0	PLL0 divider enable register
31 ~ 16	-	-	-	Undefined

MODECTR Register

0xF3000014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	PDCK	PD	-	-	-	-	-	-	-	HALT

Field	Name	RW	Reset	Description
0	HALT	RW	0x0	Halt mode enable register
7 ~ 1	-	-	-	Undefined
8	PD	RW	0x0	Power down mode enable register
9	PDCK	RW	0x0	Power down clock selection register 0b : XTIN selected 1b : XIN selected
31 ~ 10	-	-	-	Undefined

BCLKCTR0 & BCLKCTR1 Register

0xF3000018/1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCLKEN[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCLKEN[15:00]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCLKEN[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCLKEN[47:32]															

Field	Name	RW	Reset	Description
63 ~ 0	BCLKEN	RW	0xFFFFFFFFFFFFFF	Bus clock enable register (refer to Figure 23.2) 0b : bus clock disabled 1b : bus clock enabled

SWRESET0 & SWRESET1 Register

0xF3000020/24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWRESET [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRESET[15:00]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWRESET [63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRESET[47:32]															

Field	Name	RW	Reset	Description
63 ~ 0	SWRESET	RW	0x000000000000800000	Software reset for hardware (refer to Figure 23.2) 0b : reset inactive state 1b : reset active state

BCLKMASKE0 & BCLKMASKE1 Register

0xF30000F0/F4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BMASKE [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BMASKE[15:00]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BMASKE [63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BMASKE[47:32]															

Field	Name	RW	Reset	Description
63 ~ 0	BMASKE	RW	0xFFFFFFFFFFFFFF	Dynamic mask enable for bus clock (refer to Figure 23.2) 0b : Dynamic mask disabled 1b : Dynamic mask enabled

Corresponding Controller for Each Bits			
0	USB1.1 Host Controller	1	USB2.0 Function Controller
2	LCD Controller	3	Camera Interface Controller
4	HDD Controller	5	DMA Controller
6	SD/MMC Controller	7	Memory Stick Controller
8	I2C Controller	9	NAND Flash Controller
10	EHI 0 Controller	11	UART Controller Channel 0
12	UART Controller Channel 1	13	GPSB Channel 0
14	DAI Controller	15	UART Controller Channel 2
16	ECC Controller	17	S/PDIF Transmitter
18	GPSB Channel 1	19	Sub processor peripherals
20	Graphic 2D	21	RTC
22	External Memory Controller	23	Sub processor
24	ADC	25	Video Controller Hardware
26	Timer	27	Main processor
28	Vectored Interrupt Controller	29	Memory-to-Memory Scaler
30	Mailbox	31	Main Bus Components (should be '1')
32	UART Channel 3	33	SRAM Interface
34	GPSB Channel 2	35	GPSB Channel 3
36	EHI 1 Controller	37	Video Encoder

Figure 23.2 Hardware Block Group

PCK_XXX Register

0xF3000080 ~ 0xF30000DC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-			EN		SEL						-				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															DIV

Field	Name	RW	Reset	Description
11 ~ 0	DIV	RW	0x0	Clock divisor for each hardware FPCK_XXX = FCKS / (DIV + 1) FCKS is selected clock by SEL In case of DIV being 0, the FPCK_XXX = FCKS, FCKS should be under 500MHz
23 ~ 12	-	-	-	Undefined
27 ~ 24	SEL	RW	0x4	Source clock selection register Refer to Clock Source Table
28	EN	RW	0x1	Clock divider enable register * If DIV is 0, the output clock can't be disabled.
31 ~ 29	-	-	-	Undefined

Clock Source Table				
0	PLL0 direct output	1	PLL1 direct output	
2	PLL0 divider 0 output	3	PLL1 divider 0 output	
4	XIN direct	5	XIN divider output	
6	XTIN direct output	7	XTIN divider output	
8	PLL0 divider 1 output	9	PLL1 divider 1 output	
10	External clock ⁵⁸	11	Timer/Counter 1 output	
12	Timer/Counter 2 output	13	Timer/Counter 3 output	
14	Timer/Counter 4 output	15	48M Clock from USB PHY	

Offset	Register	Corresponding Hardware	Offset	Register	Corresponding Hardware
0x80	PCK_USB11H	USB11H	0xB0	PCK_TCZ	32-bit Timer TCLK
0x84	PCK_SDMMC	SD/MMC	0xB4	PCK_DAI	Refer to next page.
0x88	PCK_MSTICK	Memory Stick	0xB8	PCK_GPSB0	GPSB Channel 0
0x8C	PCK_I2C	I2C	0xBC	PCK_GPSB1	GPSB Channel 1
0x90	PCK_LCD	LCD	0xC0	PCK_GPSB2	GPSB Channel 2
0x94	PCK_CAM	CIF PXCLK	0xC4	PCK_GPSB3	GPSB Channel 3
0x98	PCK_UART0	UART Channel 0	0xC8	PCK_ADC	ADC
0x9C	PCK_UART1	UART Channel 1	0xCC	PCK_SPDIF	SPDIF TX
0xA0	PCK_UART2	UART Channel 2	0xD0	PCK_RFREQ	SDRAM Refresh Clock
0xA4	PCK_UART3	UART Channel 3	0xD4	PCK_SCALER	CIF Scaler
0xA8	PCK_TCT	CLK_OUT1	0xD8	PCK_EHI0	EHI 0
0xAC	PCK_TCX	CLK_OUT0	0xDC	PCK_EHI1	EHI 1

⁵⁸ GPIOA[7] can be configured as the external clock function.

PCK_DAI Register

0xF30000B4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MD	-		EN		SEL						-				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DIV

Field	Name	RW	Reset	Description
15 ~ 0	DIV	RW	0x0	Clock divisor for each hardware In Divider Mode, FPCK_DAI = FCKS / (DIV + 1) In DCO Mode, FPCK_DAI = FCKS * ((65536-DIV) / (65536)) FCKS is selected clock by SEL DIV should not be "ZERO"
23 ~ 16	-	-	-	Undefined
27 ~ 24	SEL	RW	0xA	Source clock selection register Refer to Clock Source Table
28	EN	RW	0x1	Clock divider enable register
30 ~ 29	-	-	-	Undefined
31	MD	RW	0x0	Mode selection register 1 : DIVIDER mode 0 : DCO mode

* If you want the peripheral clock generator to be disabled, you should select the lowest frequency clock source for reducing the power consumption.

Clock Source Table			
0	PLL0 direct output	1	PLL1 direct output
2	PLL0 divider 0 output	3	PLL1 divider 0 output
4	XIN direct	5	XIN divider output
6	XTIN direct output	7	XTIN divider output
8	PLL0 divider 1 output	9	PLL1 divider 1 output
10	External clock ⁵⁹	11	Timer/Counter 1 output
12	Timer/Counter 2 output	13	Timer/Counter 3 output
14	Timer/Counter 4 output	15	48M Clock from USB PHY

⁵⁹ GPIOA[7] can be configured as the external clock function.

BCKVCFG Register

0xF30000E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCKMVEN[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCKMV[15:00]															

Field	Name	RW	Reset	Description	
					Bus clock dynamic mask enable for video hardware
31 ~ 16	BCKMVEN	RW	0xFFFF	0b	: Dynamic clock masking disabled
					1b : Dynamic clock masking enabled
					* Recommended to 0xFFFF
					Bus Clock Mask for Video Hardware
15 ~ 0	BCKMV	RW	0xFFFF	0b	: Bus clock masked (disabled)
					1b : Bus clock unmasked (enabled)
					* Recommended to 0xFFFF

OPTION0 Register

0xF30000FC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															

Field	Name	RW	Reset	Description
31 ~ 1	-	RW	0	Undefined
0	BD	RW	0x1	Bus Clock Duty Control Register 0b : High pulse period is FSYS clock cycle 1b : High pulse period is near 50% duty
* Recommended to '0'				

Watchdog Control Register (WDTCTRL)

0xF3000028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WE	WC							-							WS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

WDTCNT

Field	Name	RW	Reset	Description
31	WE	RW	0	Watchdog Enable Register 0b : Disabled 1b : Enabled
30	WC	RW	0	Watchdog Clear Register 0b : Not-Clear 1b : Clear
16	WS	RW	0	Watchdog Status Indication Register
15 ~ 0	WDTCNT	RW	0x0	Watchdog Count Register Counter value = WDTCNT * 2^16 In case of 12MHz and WDTCNT being 1, the period of watchdog is about 5.2ms.

- * If POR reset activated, all the bit fields are initialized with reset value.
- * If watchdog reset activated, only "WE" field are cleared.
- * The clock source for watchdog counter is XIN.
- * If watchdog enabled, the watchdog counter should be cleared by WC field until the WDTCNT being 0.

23.3 Operation & Timing Diagram

23.3.1 Clock Change Operation

An example of the timing diagram for changing the clock is shown below, Figure 23.3.

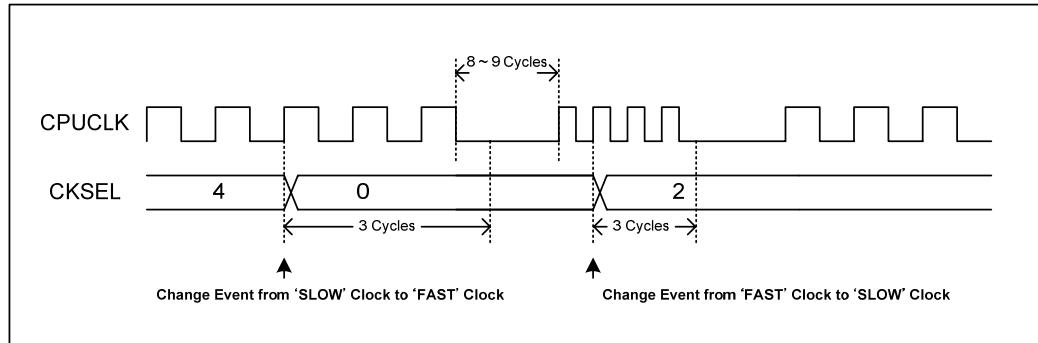


Figure 23.3 Clock Change Timing Diagram

An example shows two changing sequences. The first sequence is for changing from slower to faster frequency.

When the CPU write the 'CKSEL' register bits with specified value ('0' in this example), the glitch-free circuit in the CKC hardware first stops the current clock after 3 clock cycles. And then in the 3 clock period, the clock multiplexer changes from the current clock to the next clock. Finally, the wake-up circuits enables the clock output. The changing sequence from 'faster' to 'slower' clocks is same procedure.

23.3.2 Procedure for Configuration of Peripheral Clocks

The Figure 23.4 shows the configuration procedure for peripheral clock.

The programmer should take care of some cautions in configuring the peripheral clock for corresponding hardware. (e.g., MS, DAI, etc.)

The controller should be in reset state by setting the SWRESET bit before configuring the hardware clock and configuring the corresponding ports. If you configure the clock and ports in not-reset state, which is not-initialized state, the unexpected operations can occur.

The hardware starts operating by clearing the SWRESET bit after the configuration for the clock and ports of corresponding controller.

If you want to close the hardware operation, you should make the controller into reset-state by SWRESET bit and stop the clock and release the port.

If the controller is in reset-state (SWRESET='1'), the program can't access the control register for corresponding hardware.

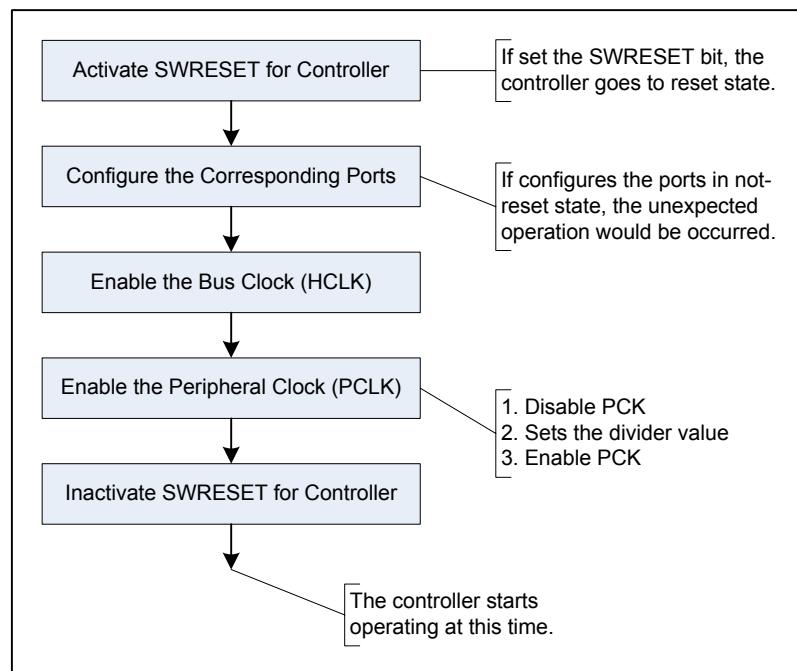


Figure 23.4 Peripheral Clock Configuration Procedure

23.3.3 Reference PMS Table for Target Frequency

You must get the proper divider setting values from APPENDIX A.

23.3.4 Enter Halt Mode

The timing diagram of entering halt mode is shown below, Figure 23.5.

To enter the halt mode, all you do is writing the '1' into 'HALT' bit.

If the program sets the 'HALT' bit, the halt mode controller stops the CPU clock after 3 clock cycles. After then, the CPU can't access the bus, TCM, and cache components any more. This causes that the clocks to CPU would be stopped, the processor do not operate any more, and the power consumption could be reduced.

But it is prefer to use the waiting for interrupt in the ARM technical reference manual than this mode. But if you use this mode, the instruction prefetching from the memory should not be occurred on the instruction bus during entering this mode. It can be achieved by that the code for entering the halt-mode should be in the instruction TCM or in the instruction cache memory with locked status to avoid the instruction prefetching transfer.

The ARM926EJ-S core has the same function, the waiting-for-interrupt function described in ARM926EJ-S technical reference manual is to halt the processor core and wait until IRQ or FIQ interrupt. In this mode, the internal clock does not propagate to reduce the power consumption.

23.3.5 Exit Halt Mode

The timing diagram of exit halt mode is shown below, Figure 23.5.

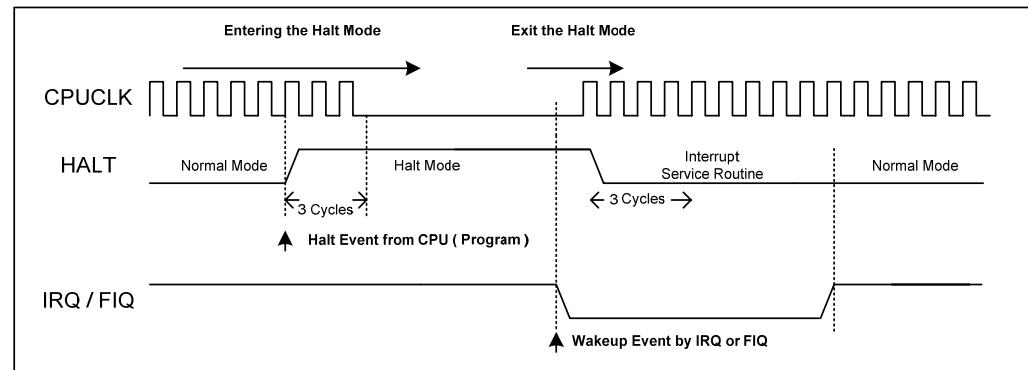


Figure 23.5 Timing Diagram to Enter and Exit Halt Mode

The only way to exit halt mode is that the IRQ or FIQ interrupt occur. In halt mode, the FIQ or IRQ interrupt wakes up the halt-mode controller and enables the CPU clocks.

After enabling the CPU clock, the processor in halt mode goes to interrupt service routine and goes to normal mode after the interrupt service routine. And you can re-enter the halt mode when no jobs to process.

23.3.6 Enter Power-Down Mode

The timing diagram of entering power-down mode is shown below, Figure 23.6.

In the power-down mode, all the on-chip clocks are disabled including main oscillator (XIN). To enter the power-down mode, all you do is set the 'PWRDN' bit by software.

The sequence for entering the power-down mode is as follows.

- (1) Change the CPU clock with XIN or XTIN
- (2) Set the selection bit for power-down clock, XIN or XTIN.

(3) Set the 'PWRDN' bit.

Once the 'PWRDN' bit is set, the CKC power-down controller disables the oscillator after 1.5 or 2.5 cycles. After some cycles, the main crystal does not oscillate any more until the external interrupt is generated. The corresponding external interrupts are selected by WKEN register in "VECTORED INTERRUPT CONTROLLER" on page 24-1.

In the power-down mode, the power consumption is lowest among the various powered-on operating modes. But, the wake-up time is longest.

If you selected the XTIN with power-down mode clock, the XTIN oscillator is not disabled because the XTIN oscillator is always turned-on.

23.3.7 Exit Power-Down Mode

The timing diagram of exit power-down mode is shown below, Figure 23.6.

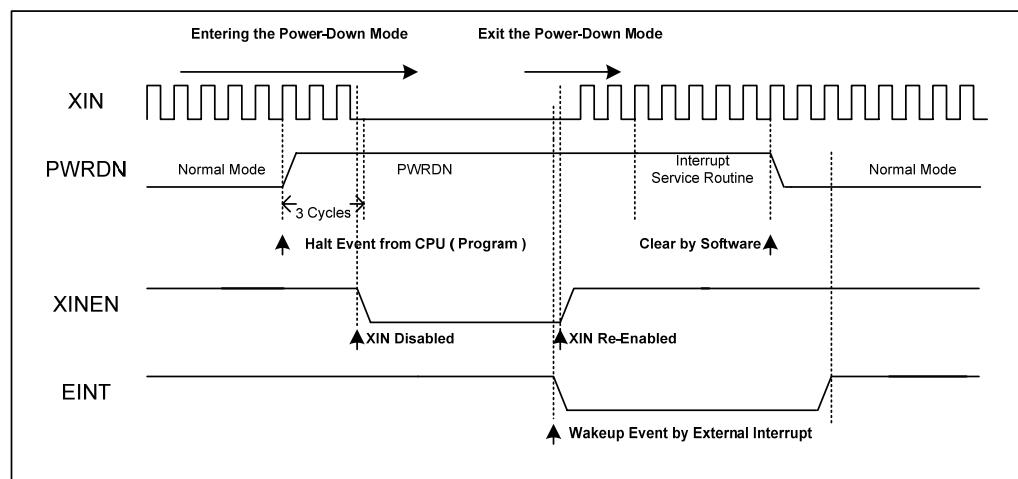


Figure 23.6 Timing Diagram of Enter and Exit Power-Down Mode

The only way to exit the power down mode is external interrupt, which does not controlled by oscillator (ex, key, host, etc).

The external interrupt wakes up the power-down controller in CKC and enables the main oscillator (XIN). The clocks from main oscillator goes through the CKC blocks via on-chip clock-noise filter, which consumes the long wake-up time, and the processor starts interrupt service.

24 VECTORED INTERRUPT CONTROLLER

24.1 Overview

The following figure represents the block diagram of interrupt controller. The interrupt controller can manage up to 32 interrupt sources. In the TCC79XX, there are four external interrupt sources that can be detected various kind of method that is a rising edge/ falling edge / level high / level low. The external interrupt sources can be selected among the various GPIO ports and fed reliably into interrupt controller with dedicated noise filters.

There are two types of interrupt in ARM family processor; IRQ type, FIQ type.

Interrupt controller can select these two types for each interrupt sources separately.

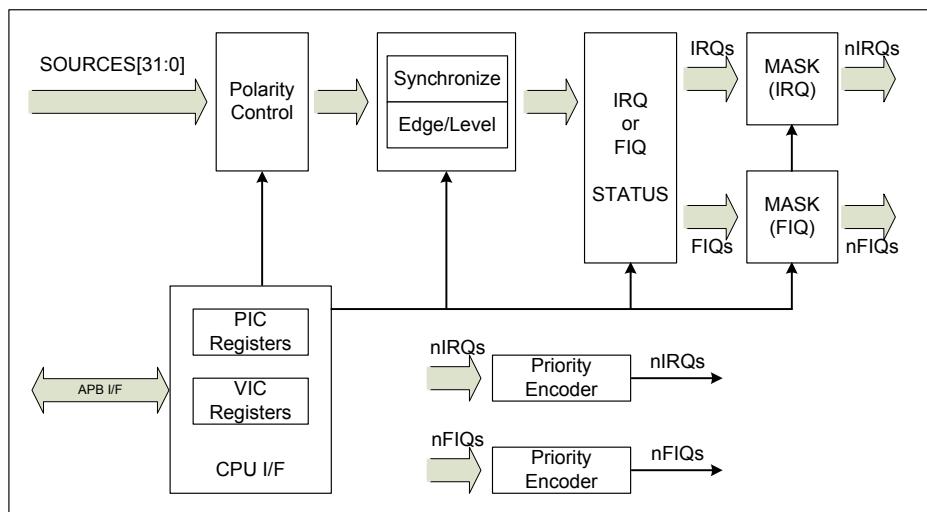


Figure 24.1 Block Diagram of Vectored Interrupt Controller

Table 24.1 Priority Interrupt Controller Register Map (Base Address = 0xF3001000)

Name	Address	Type	Reset	Description
IEN	0x00	R/W	0x00000000	Interrupt Enable Register
CLR	0x04	R/W	0x00000000	Interrupt Clear Register
STS	0x08	R	Unknown	Interrupt Status Register
SEL	0x0C	R/W	0x00000000	IRQ or FIR Selection Register
SRC	0x10	R	Unknown	Source Interrupt Status Register
MSTS	0x14	R	0x00000000	Masked Status Register
TIG	0x18	R/W	0x00000000	Test Interrupt Generation Register
POL	0x1C	R/W	0x00000000	Interrupt Polarity Register
IRQ	0x20	R	0x00000000	IRQ Raw Status Register
FIQ	0x24	R	Unknown	FIQ Status Register
MIRQ	0x28	R	0x00000000	Masked IRQ Status Register
MFIQ	0x2C	R	0x00000000	Masked FIQ Status Register
MODE	0x30	R/W	0x00000000	Trigger Mode Register – Level or Edge
SYNC	0x34	R/W	0xFFFFFFFF	Synchronization Enable Register
WKEN	0x38	R/W	0x00000000	Wakeup Event Enable Register
MODEA	0x3C	R/W	0x00000000	Both Edge or Single Edge Register
INTMSK	0x40	R/W	0xFFFFFFFF	Interrupt Output Masking Register
ALLMSK	0x44	R/W	0x00000003	All Mask Register

Table 24.2 Vectored Interrupt Controller Register Map (Base Address = 0xF3001080)

Name	Address	Type	Reset	Description
VAIRQ	0x00	R	0x800000XX	IRQ Vector Register
VAFIQ	0x04	R	0x800000XX	FIQ Vector Register
VNIRQ	0x08	R	0x800000XX	IRQ Vector Number Register
VNFIQ	0x0C	R	0x800000XX	FIQ Vector Number Register
VCTRL	0x10	R/W	0x00000000	Vector Control Register
PRI04	0x20	R/W	0x03020100	Priorities for Interrupt 0 ~ 3
PRI08	0x24	R/W	0x07060504	Priorities for Interrupt 4 ~ 7
PRI12	0x28	R/W	0x0B0A0908	Priorities for Interrupt 8 ~ 11
PRI16	0x2C	R/W	0x0F0E0D0C	Priorities for Interrupt 12 ~ 15
PRI20	0x30	R/W	0x13121110	Priorities for Interrupt 16 ~ 19
PRI24	0x34	R/W	0x17161514	Priorities for Interrupt 20 ~ 23
PRI28	0x38	R/W	0x1B1A1918	Priorities for Interrupt 24 ~ 27
PRI32	0x3C	R/W	0x1F1E1D1C	Priorities for Interrupt 28 ~ 31

24.2 Priority Interrupt Controller

Interrupt Enable Register (IEN)

0xF3001000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTEX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) For each bit, '1' means that corresponding interrupt is enabled and '0' for disabled

Field	Name	RW	Reset	Description
0	EI0	RW	0x0	External interrupt 0 enable
1	EI1	RW	0x0	External interrupt 1 enable
2	EI2	RW	0x0	External interrupt 2 enable
3	EI3	RW	0x0	External interrupt 3 enable
4	RTC	RW	0x0	RTC interrupt enable
5	GPSB	RW	0x0	GPSB Interrupt Enable
6	TC0	RW	0x0	Timer 0 interrupt enable
7	TC1	RW	0x0	Timer 1 interrupt enable
8	SCORE	RW	0x0	Sub processor interrupt enable (Mailbox)
9	SPDTEX	RW	0x0	SPDIF transmitter interrupt enable
10	SEL0	RW	0x0	Refer to IRQSEL0 on page 24-5.
11	SEL1	RW	0x0	Refer to IRQSEL1 on page 24-5.
12	SC	RW	0x0	Mem-to-Mem scaler interrupt enable
13	I2C	RW	0x0	I2C interrupt enable
14	DAIRX	RW	0x0	DAI receive interrupt enable
15	DAITX	RW	0x0	DAI transmit interrupt enable
16	CDRX	RW	0x0	CDIF receive interrupt enable
17	HPI	RW	0x0	EHI channel 0 internal interrupt enable
18	UT	RW	0x0	UART interrupt enable
19	SEL2	RW	0x0	Refer to IRQSEL2 on page 24-5.
				Graphic Engine 2D Hardware Interrupt Enable
20	G2D	RW	0x0	Note that the interrupt request signal of graphic engine is active low. Therefore, G2D bit of POL register should be set to 1.
21	UD	RW	0x0	USB 2.0 device interrupt enable
22	UH	RW	0x0	USB 1.1 host interrupt enable
23	DMA	RW	0x0	DMA controller interrupt enable
24	HDD	RW	0x0	HDD controller interrupt enable
25	SEL3	RW	0x0	Refer to IRQSEL3 on page 24-5
26	NFC	RW	0x0	Nand flash controller interrupt enable
27	SD	RW	0x0	SD/MMC interrupt enable
28	CAM	RW	0x0	Camera interrupt enable
29	LCD	RW	0x0	LCD controller interrupt enable
30	ADC	RW	0x0	ADC interrupt enable
31	SEL4	RW	0x0	Refer to IRQSEL4 on page 24-5

The interrupt controller can receive up to 8 external interrupts simultaneously, which are EI0 ~ EI7. And each external interrupt can become one of the following 32 interrupt sources. EINT0SEL and EINT1SEL register are used for selecting these interrupt sources.

NUM	EINTn(n=even, 0)	EINTn(n=odd)
0	URXD3	URXD3
1	UTXD2	UTXD2
2	AIN[7]	AIN[7]
3	AIN[1]	AIN[1]
4	AIN[0]	AIN[0]
5	GPIOA[11]	GPIOA[11]
6	GPIOA[7]	GPIOA[7]
7	GPIOA[6]	GPIOA[6]
8	SDO1	SDO1
9	SCMD1	SCMD1
10	GPIOB[15]	GPIOB[15]
11	GPIOB[14]	GPIOB[14]
12	GPIOB[13]	GPIOB[13]
13	GPIOB[4]	GPIOB[4]
14	GPIOB[3]	GPIOB[3]
15	GPIOB[0]	GPIOB[0]
16	GPIOA[5]	GPIOA[5]
17	GPIOA[3]	GPIOA[3]
18	GPIOA[2]	GPIOA[2]
19	SDO0	SDO0
20	SCMD0	SCMD0
21	HPXD[3]	HPXD[3]
22	HPXD[11]	HPXD[11]
23	HPRDN	HPRDN
24	HPCSN_L	HPCSN_L
25	GPIOF[26]	GPIOF[26]
26	GPIOF[27]	GPIOF[27]
27	GPIOC[30]	GPIOC[30]
28	LPD[18]	LPD[18]
29	LPD[23]	LPD[23]
30	PMKUP	PMKUP
31	USB_VBOFF	USB_VBON

EINTSEL0

0xF005A134

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EINT3SEL															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EINT1SEL															
EINT0SEL															

EINTSEL1

0xF005A138

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EINT7SEL															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EINT5SEL															
EINT4SEL															

EINT0SEL selects EI0 interrupt source. For example, when EINT0SEL is set to 29, LPD[23] port is used for the external interrupt EINT0. When EINT7SEL is set to 0, URXD3 port is used for the external interrupt EI7.

IRQSEL

0xF005A13C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				0		IRQSEL4		IRQSEL3		IRQSEL2		IRQSEL1		IRQSEL0	

An interrupt source of SEL0, SEL1, SEL2, SEL3, and SEL4 can be programmed. These are determined by IRQSEL register.

This register selects an interrupt request source. For example, when IRQSEL0 is equal to 2, SEL0 bit of IEN register is allocated for the external interrupt 4

IRQSEL0 is used for selecting an interrupt source for SEL0 bit of IEN register

IRQSEL0[1:0]	RESET	NOTE
0		VIDEO CORE interrupt
1		EHI channel 0 control signal interrupt
2	0	External interrupt 4, EI4
3		-

IRQSEL1 is used for selecting an interrupt source for SEL1 bit of IEN register

IRQSEL1[3:2]	RESET	NOTE
0		GPSB interrupt
1		EHI channel 1 control signal interrupt
2	0	External interrupt 5, EI5
3		-

IRQSEL2 is used for selecting an interrupt source for SEL2 bit of IEN register

IRQSEL2[5:4]	RESET	NOTE
0		UART interrupt
1		EHI channel 1 internal interrupt
2	0	-
3		-

IRQSEL3 is used for selecting an interrupt source for SEL3 bit of IEN register

IRQSEL3[7:6]	RESET	NOTE
0		Memory Stick interrupt
1		External Interrupt 6, EI6
2	0	-
3		-

IRQSEL4 is used for selecting an interrupt source for SEL4 bit of IEN register

IRQSEL4[9:8]	RESET	NOTE
0		GPSB interrupt
1		ECC interrupt
2	0	External Interrupt 7, EI7
3		USB 2.0 DMA interrupt

Interrupt Clear Register (CLR)

0xF3001004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) For each bit, the interrupt status is cleared by writing '1' for corresponding bit.

Interrupt Status Register (STS)

0xF3001008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) For each bit, if it is '1', the corresponding interrupt was activated.

Interrupt Select Register (SEL)

0xF300100C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) For each bit, if it is '1', the corresponding interrupt is propagated to nIRQ otherwise to nFIQ.

Interrupt Source Status Register (SRC)

0xF3001010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) This represents the status for each interrupt source by XOR with interrupt input and polarity register for the corresponding interrupt source.

Masked Interrupt Status Register (MSTS)

0xF3001014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) This register represents the interrupt status for enabled sources. If a interrupt is not enabled, the corresponding bit is '0'.

Test Interrupt Source Register (TST)

0xF3001018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) In the case of the corresponding bit in SRC register is '1', the interrupt is activated by writing '1'.

Interrupt Polarity Control Register (POL)

0xF300101C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) If the interrupt signal is active-high, the corresponding bit should be '0', otherwise '1' – active-low.

IRQ Raw Status Register (IRQ)

0xF3001020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTEX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) This represents the raw status for IRQ register.

FIQ Raw Status Register (FIQ)

0xF3001024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTEX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) This represents the raw status for FIQ register.

Masked IRQ Status Register (MIRQ)

0xF3001028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTEX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) This represents the masked status for IRQ register.

Masked FIQ Status Register (MFIQ)

0xF300102C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTEX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) This represents the masked status for FIQ register.

Mode Register (MODE)

0xF3001030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTEX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) If the corresponding bit is '1', the interrupt acts as level-triggered mode, otherwise edge-triggered mode.

Synchronization Enable Register (SYNC)

0xF3001034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTEX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) If the corresponding bit is '1', the controller synchronizes the corresponding interrupt source with bus clock - HCLK.

Wakeup Enable Register (WKEN)

0xF3001038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTEX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) If the corresponding bit is '0', the CPU clock or bus clock can be waked up by the corresponding interrupt source.

Edge Trigger Mode Register (MODEA)

0xF300103C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTEX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) If the corresponding bit is '0' in case of the edge-triggered mode, the interrupt propagated to nFIQ or nIRQ in single edge, otherwise in both edge.

IRQ Output Mask Register (INTMSK)

0xF3001040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL4	ADC	LCD	CAM	SD	NFC	SEL3	HDD	DMA	UH	UD	G2D	SEL2	UT	HPI	CDRX
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAITX	DAIRX	I2C	SC	SEL1	SEL0	SPDTEX	SCORE	TC1	TC0	GPSB	RTC	EI3	EI2	EI1	EI0

*) If the corresponding bit is '1', interrupt are passed to IRQ or FIQ.

All Register (ALLMSK)

0xF3001044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- - - - FIQ IRQ

Field	Name	RW	Reset	Description
0	IRQ	RW	0x1	IRQ mask register When it is 0, IRQ interrupt is masked.
1	FIQ	RW	0x1	FIQ mask register When it is 0, FIQ interrupt is masked.
31 ~ 2	-	-	-	Undefined

24.3 Vectored Interrupt Controller

The vectored interrupt controller can make it possible to service the corresponding interrupt with any other unnecessary comparison operation. The vectored interrupt controller has the 4 registers for this purpose, VAIRQ, VAFIQ, VNIRQ, and VNIRQ.

The VAIRQ and VAFIQ represent the vectored address offset for current activated interrupt in word-based address. For example, if the 2nd priority interrupt has been activated to IRQ, the VAIRQ represents the '4' in VA field below with '0' INV field. The VAFIQ is same as VAIRQ.

The VNIRQ and VNFIQ represent the vectored number offset for current interrupt. For example, if the 2nd priority interrupt has been activated to IRQ, the VNIRQ represents the '1' in VA field below. You can use these features for fast handler service.

IRQ Vector Register (VAIRQ)

0xF3001080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INV									0						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

0

VA

INV [31]	Type	Invalid Field
0	R	Valid for Current Interrupt Source.
1		Invalid for Current Interrupt Source.

VA [6:0]	Type	Invalid Field
n	R	Interrupt Vector Address Offset in Word-Address. This is one of '0', '4', '8', ...

FIQ Vector Register (VAFIQ)

0xF3001084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INV									0						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

0

VA

INV [31]	Type	Invalid Field
0	R	Valid for Current Interrupt Source.
1		Invalid for Current Interrupt Source.

VA [6:0]	Type	Invalid Field
n	R	Interrupt Vector Address Offset in Word-Address. This is one of '0', '4', '8', ...

IRQ Number Register (VNIRQ)

0xF3001088

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INV								0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INV [31]	Type	Invalid Field
0	R	Valid for Current Interrupt Source.
1		Invalid for Current Interrupt Source.

VN [4:0]	Type	Invalid Field
n	R	Interrupt Vector Address Offset in Word-Address. This is one of '0', '1', '2', ...

FIQ Number Register (VNFIQ)

0xF300108C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INV								0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INV [31]	Type	Invalid Field
0	R	Valid for Current Interrupt Source.
1		Invalid for Current Interrupt Source.

VN [4:0]	Type	Invalid Field
n	R	Interrupt Vector Address Offset in Word-Address. This is one of '0', '1', '2', ...

Vector Control Register (VCTRL)

0xF3001090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCL	FPOL	FFLG	IFLG	FHD	IHD	0									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															

RCL [31]	Type	Clear Interrupt Status
0	R/W	The write operation is needed for clearing interrupt status.
1		The reading the VN or VA register clears the interrupt status.
FPOL [30]	Type	Valid Flag Polarity
0	R/W	The INV field means valid for '0', invalid for '1'.
1		The INV field means valid for '1', invalid for '0'.
FFLG [29]	Type	Valid Flag Enable
0	R/W	Invalid flag enable for FIQ vector registers.
1		Valid flag enable for FIQ vector registers.
IFLG [28]	Type	Valid Flag Enable
0	R/W	Invalid flag enable for IRQ vector registers.
1		Valid flag enable for IRQ vector registers.
FHD [27]	Type	Hold Enable
0	R/W	Disable the holding vector for FIQ until cleared.
1		Enable the holding vector for FIQ until cleared.
IHD [26]	Type	Hold Enable
0	R/W	Disable the holding vector for IRQ until cleared.
1		Enable the holding vector for IRQ until cleared.

Priority Configuration Register (PRIxx)

0xF30010A0 ~ 0xF30010BC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			PRI3					0			PRI2				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0			PRI1					0			PRI0				

After resetting the system, the priority of each interrupt controller is determined with default value. The default priority of interrupt source 0 is 0 – highest priority and the default priority of interrupt source 31 is 31 – lowest priority. If you want to change the priority of each interrupt sources, you have to write the priority value to the corresponding field.

Be advised that the controller can do the unpredicted operation in case the priorities of different interrupt sources are set with the same value.

PRI0 [4:0]	Type	Priority Configuration Register
n	R/W	0xF30010A0 : Priority for Interrupt 0
		0xF30010A4 : Priority for Interrupt 4
		0xF30010A8 : Priority for Interrupt 8
		0xF30010AC : Priority for Interrupt 12
		0xF30010B0 : Priority for Interrupt 16
		0xF30010B4 : Priority for Interrupt 20
		0xF30010B8 : Priority for Interrupt 24
		0xF30010BC : Priority for Interrupt 28
PRI1 [12:8]	Type	Priority Configuration Register
n	R/W	0xF30010A0 : Priority for Interrupt 1
		0xF30010A4 : Priority for Interrupt 5
		0xF30010A8 : Priority for Interrupt 9
		0xF30010AC : Priority for Interrupt 13
		0xF30010B0 : Priority for Interrupt 17
		0xF30010B4 : Priority for Interrupt 21
		0xF30010B8 : Priority for Interrupt 25
		0xF30010BC : Priority for Interrupt 29
PRI2 [20:16]	Type	Priority Configuration Register
n	R/W	0xF30010A0 : Priority for Interrupt 2
		0xF30010A4 : Priority for Interrupt 6
		0xF30010A8 : Priority for Interrupt 10
		0xF30010AC : Priority for Interrupt 14
		0xF30010B0 : Priority for Interrupt 18
		0xF30010B4 : Priority for Interrupt 22
		0xF30010B8 : Priority for Interrupt 26
		0xF30010BC : Priority for Interrupt 30

PRI3 [28:24]		Type	Priority Configuration Register
			0xF30010A0 : Priority for Interrupt 3
			0xF30010A4 : Priority for Interrupt 7
			0xF30010A8 : Priority for Interrupt 11
n	R/W		0xF30010AC : Priority for Interrupt 15
			0xF30010B0 : Priority for Interrupt 19
			0xF30010B4 : Priority for Interrupt 23
			0xF30010B8 : Priority for Interrupt 27
			0xF30010BC : Priority for Interrupt 31

25 TIMER / COUNTER**25.1 Overview**

The TCC79XX has four 16-bit, two 20-bit, and one 32-bit timers/counters. 16-bit and 20-bit timer/counters have three registers for basic operation modes. Refer to register description table for details. When operating in counter modes, External interrupt pin is used as counting clock for that counter.

The main clock frequency of timer counter is determined by XIN frequency. With the 12bit internal basic counter, the timer counter can generate various intervals from microseconds to seconds unit.

32-bit timer is included in timer counter, but watchdog timer is in CKC block. Refer to 23. CKC on page 23-1.

The following figure represents the block diagram of timer/counter.

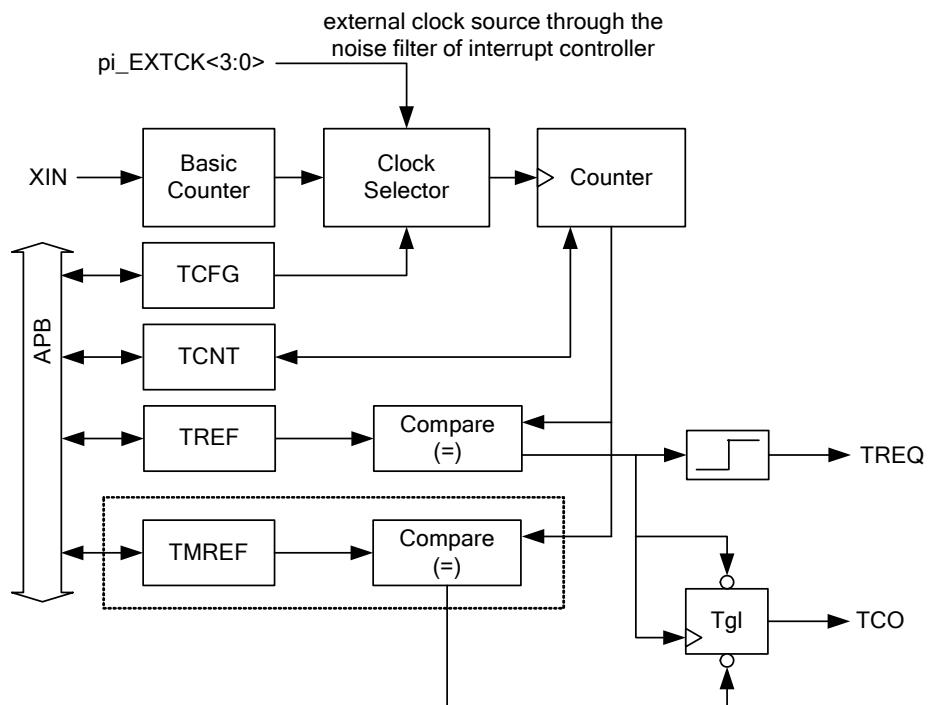


Figure 25.1 16-bit and 20bit Timer/Counter Block Diagram

The following table explains the registers of each timer counter. The address of each timer counter is 16bytes aligned. The base address of timer counter is 0xF3003000.

The number n represents for each timer/counter. In case of timer/counter 4, 5 (that is n = 4 or 5) the TREF, TCNT register has 20bit resolution. It can be used for generating long time events.

25.2 Register Description

Table 25.1 Timer/Counter Register Map (Base Address = 0xF3003000)

Name	Address	Type	Reset	Description
TCFG0	0x00	R/W	0x00	Timer/Counter 0 Configuration Register
TCNT0	0x04	R/W	0x0000	Timer/Counter 0 Counter Register
TREF0	0x08	R/W	0xFFFF	Timer/Counter 0 Reference Register
TMREF0	0x0C	R/W	0x0000	Timer/Counter 0 Middle Reference Register
TCFG1	0x10	R/W	0x00	Timer/Counter 1 Configuration Register
TCNT1	0x14	R/W	0x0000	Timer/Counter 1 Counter Register
TREF1	0x18	R/W	0xFFFF	Timer/Counter 1 Reference Register
TMREF1	0x1C	R/W	0x0000	Timer/Counter 1 Middle Reference Register
TCFG2	0x20	R/W	0x00	Timer/Counter 2 Configuration Register
TCNT2	0x24	R/W	0x0000	Timer/Counter 2 Counter Register
TREF2	0x28	R/W	0xFFFF	Timer/Counter 2 Reference Register
TMREF2	0x2C	R/W	0x0000	Timer/Counter 2 Middle Reference Register
TCFG3	0x30	R/W	0x00	Timer/Counter 3 Configuration Register
TCNT3	0x34	R/W	0x0000	Timer/Counter 3 Counter Register
TREF3	0x38	R/W	0xFFFF	Timer/Counter 3 Reference Register
TMREF3	0x3C	R/W	0x0000	Timer/Counter 3 Middle Reference Register
TCFG4	0x40	R/W	0x00	Timer/Counter 4 Configuration Register
TCNT4	0x44	R/W	0x00000	Timer/Counter 4 Counter Register
TREF4	0x48	R/W	0xFFFF	Timer/Counter 4 Reference Register
TCFG5	0x50	R/W	0x00	Timer/Counter 5 Configuration Register
TCNT5	0x54	R/W	0x00000	Timer/Counter 5 Counter Register
TREF5	0x58	R/W	0xFFFF	Timer/Counter 5 Reference Register
TIREQ	0x60	R/W	0x0000	Timer/Counter n Interrupt Request Register
TWDCFG	0x70	R/W	0x0000	Reserved
TWDCLR	0x74	W	-	Reserved
TC32EN	0x80	R/W	0x00007FFF	32-bit Counter Enable / Pre-scale Value
TC32LDV	0x84	R/W	0x00000000	32-bit Counter Load Value
TC32CMP0	0x88	R/W	0x00000000	32-bit Counter Match Value 0
TC32CMP1	0x8C	R/W	0x00000000	32-bit Counter Match Value 1
TC32PCNT	0x90	R/W	-	32-bit Counter Current Value (pre-scale counter)
TC32MCNT	0x94	R/W	-	32-bit Counter Current Value (main counter)
TC32IRQ	0x98	R/W	0x0000----	32-bit Counter Interrupt Control

Timer/Counter n Configuration Register (TCFGn)

0xF30030n0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					0	STOP	CC	POL		TCKSEL		IEN	PWM	CON	EN

STOP [9]		STOP Mode
0	Continuous counting mode.	
1	If TCNTn is equal to the TREFn, the TCNTn counter stop to increment.	
CC [8]		Clear Count
0	TCNTn is not cleared.	
1	TCNTn is cleared to zero.	
POL [7]		TCK Polarity
0	TCNTn is incremented at rising edge of the selected counting clock	
1	TCNTn is incremented at falling edge of the selected counting clock	
TCKSEL [6:4]		TCK Select
$k = 0 \sim 4$	TCK is internally generated from divider circuit. It is driven by TCLK, and this value determines the division factor of this circuit. Division factor is $2^{(k+1)}$.	
$k = 5, 6$	TCK is internally generated from divider circuit. It is driven by TCLK, and this value determines the division factor of this circuit. Division factor is 2^{2k}	
$k = 7$	TCK is the external pin shared by external interrupt signal. In TCC79XX, there are 4 external pins for this purpose, so this configuration is valid only for timer/counter 3 ~ 0. (not for timer/counter 5, 4)	
IEN [3]		Interrupt Enable
1	Enable Timer/Counter interrupt	
0	Disable Timer/Counter interrupt	
PWM [2]		PWM Mode Enable
	Enable PWM mode	
1	Timer/Counter output is changed at every time the TCNTn is equal to TREFn and TMREFn value. It can be used to generate PWM waveform, by changing TMREFn while fixing TREFn. (where, TREFn > TMREFn)	
	Disable PWM mode	
0	Timer/Counter output can be changed only when the TCNTn is equal to TREFn. It can be used to generate a rectangular pulse of variable frequency.	

The output of Timer/Counter 0, 1, 2, and 3 can be monitored through GPIOA[7], GPIOA[6], GPIOA[5], and GPIOA[4] respectively. Refer to "PORTCFG4" on page 33-10 and "PORTCFG7" on page 33-13

CON [1]		Continue Counting
0	When the TCNTn is reached to TREFn, TCNTn restarts counting from 0 at the next pulse of selected clock source.	
1	The TCNTn continues counting from the TREFn.	

If the STOP bit is set, this bit is meaningless.

EN [0]	Timer/Counter Enable
0	Timer counter is disabled.
1	Timer counter is enabled.

Following figure illustrates the basic behavior of timer / counter.

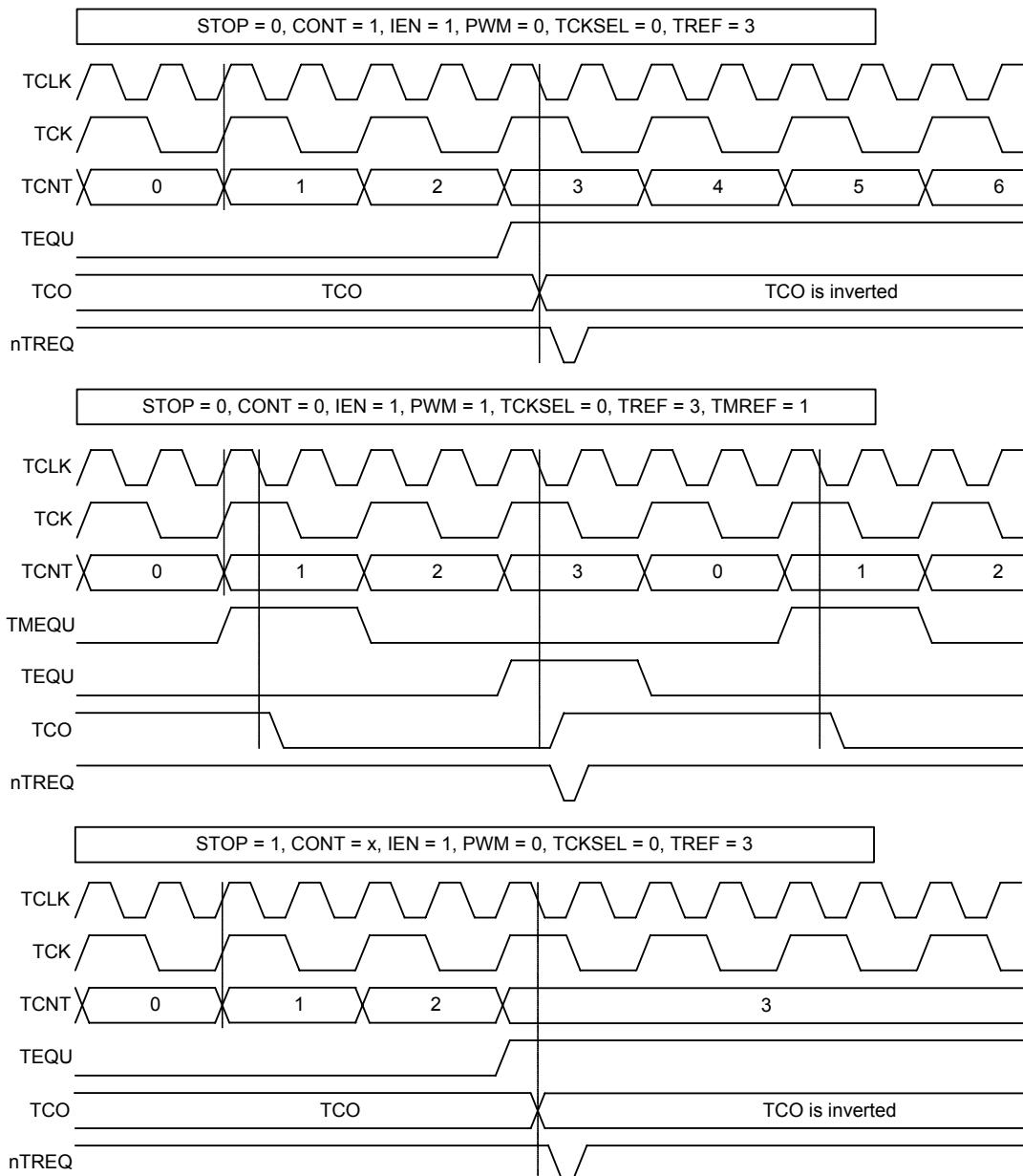


Figure 25.2 Timing diagram of timer/counter

Timer/Counter n Counting Register (TCNTn)

0xF30030n4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCNTn[19:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCNTn[15:0]															

TCNTn is increased by 1 at every pulse of selected clock source. TCNTn can be set to any value by writing to this register. In case of timer 4 and timer 5, it has 20 bits, otherwise it has 16 bits.

Timer/Counter n Counting Reference Register (TREFn)

0xF30030n8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															TREFn[19:16]
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TREFn[15:0]															

When TCNTn is reached at TREFn and the CON flag of TCFGn register is set to 1, the TCNTn is cleared to 0 at the next pulse of selected clock source. According to the TCFGn settings, various kinds of operations may be done. In case of timer 4 and timer 5, it has 20 bit, otherwise it has 16 bit.

Timer/Counter n Middle Reference Register (TMREFn)

0xF30030nC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMREFn[15:0]															

When TCNTn is reached at TMREFn and the PWM flag of TCFGn register is set to 1, the timer output of TCOOn is cleared to 0 at the negative edge of that pulse of selected clock source. The TCOOn is set to 1 when the TCNTn is reached at TREFn. (refer TREFn). So you can generate PWM signal by modifying TMREFn(n=0~3) between 0 ~ (TREFn-1).

Timer/Counter Interrupt Request Register (TIREQ) 0xF3003060

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	TWF	TF5	TF4	TF3	TF2	TF1	TF0	0	TWI	TI5	TI4	TI3	TI2	TI1	TI0

TWF [14]		Watchdog Timer Flag	
1		Watchdog timer has reached to its reference value.	
TFn [13:8]		Timer/Counter n Flag	
1		Timer/counter n has reached to its reference value.	
TWI [6]	Type	Watchdog Timer Interrupt Request Flag	
1	Read	Watchdog timer has generated its interrupt.	
1	Write	Watchdog timer interrupt is cleared.	
TIn [5:0]	Type	Timer/Counter n Interrupt Request Flag	
1	Read	Timer/counter n has generated its interrupt.	
1	Write	Timer/counter n interrupt flag is cleared.	

If a timer n has reached its reference value, the TFn is set. (bit n represents for Timer n). If its interrupt request is enabled by set bit 3 of TCFGn register, then the TIn is set. If the TC bit of IEN register is set, the timer interrupt is really generated and this TIREQ register can be used to determine which timer has requested the interrupt. After checking these flags, user can clear these TFn and TIn field by writing "1" to corresponding TFn or TIn bit field.

As illustrated in the figure below, TC32 consists of a pre-scale counter, main counter and two comparators. The pre-scale counter is a simple 24-bit up-counter which always counts from zero to PRESCALE value programmed in TC32EN register. The 32-bit main counter is incremented only when the prescale counter reaches PRESCALE value.

The clock input of TC32 module is PCK_TCZ from CKC Refer to 23. CKC on page 23-1.

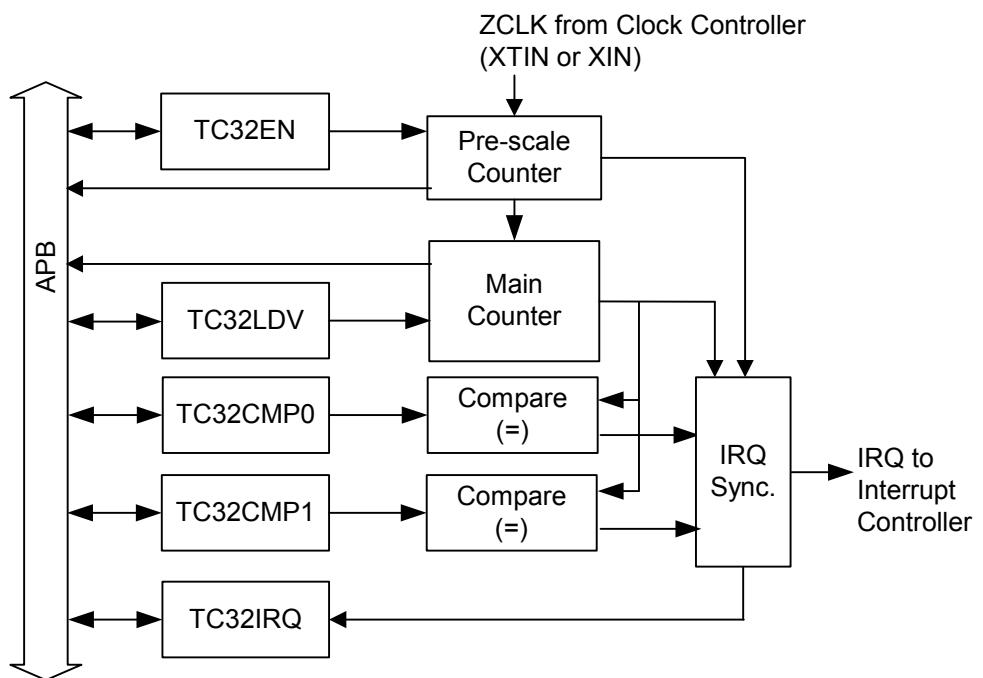


Figure 25.3 32-bit Counter Block Diagram

Possible counter modes are described in the table below.

Table 25.2 TC32 Count Mode

Mode	TC32EN Register Bits				Main Counter Operation	
	LOADZERO	LDM1	LDM0	Start Count Value	End Count Value	
0	0	0	0	LOADVAL	0xFFFFFFFF	
1	0	0	1	LOADVAL	CMP0 (if LOADVAL < CMP0)	
2	0	1	0	LOADVAL	CMP1 (if LOADVAL < CMP1)	
3	0	1	1	LOADVAL	CMP0 (if LOADVAL < CMP0 ≤ CMP1) or CMP1 (if LOADVAL < CMP1 ≤ CMP0)	
4	1	0	0	0	LOADVAL – 1	
5	1	0	1	0	CMP0 (if LOADVAL > CMP0)	
6	1	1	0	0	CMP1 (if LOADVAL > CMP1)	
7	1	1	1	0	CMP0 (if LOADVAL > CMP1 ≥ CMP0) or CMP1 (if LOADVAL > CMP0 ≥ CMP1)	

Refer to register descriptions below for CMP0, CMP1 and LOADVAL.

Mode0 can be used as 1Hz counter mode, if PRESCALE = 0x007FF, STOPMODE = 0, ZCLK = XTIN (32.768kHz)

TC32 Enable / Pre-scale Value Register (TC32EN) 0xF3003080

Bit	Name	Default	R/W	Description
31:30	Reserved	0	R	
29	LDM1	0	R/W	Re-load counter when the counter value is matched with CMP1. LOADZERO bit below selects the counter load(start) value.
28	LDM0	0	R/W	Re-load counter when the counter value is matched with CMP0 LOADZERO bit below selects the counter load(start) value.
27	Reserved	0	R	
26	STOPMODE	0	R/W	0 = Free Running Mode, 1 = Stop Mode.
25	LOADZERO	0	R/W	By default, counter starts from LOADVAL. When this bit is enabled (1), the counter is forced to count from "0" to "LOADVAL - 1".
24	ENABLE	0	R/W	Counter Enable
23:0	PREScale	0x007FFF	R/W	Pre-scale counter load value. The pre-scale counter always runs from "0" up to PREScale. The default value is for 1Hz counter when ZCLK = XTIN (32.768kHz).

TC32 Load Value Register (TC32LDV) 0xF3003084

Bit	Name	Default	R/W	Description
31:0	LOADVAL	0x00000000	R/W	Counter Load Value.

The counter is restarted whenever one of the TC32En and TC32LDV is written.

TC32 Match Value 0 Register (TC32CMP0) 0xF3003088

Bit	Name	Default	R/W	Description
31:0	CMP0	0x00000000	R/W	Counter Match Value

TC32 Match Value 1 Register (TC32CMP1) 0xF300308C

Bit	Name	Default	R/W	Description
31:0	CMP1	0x00000000	R/W	Counter Match Value

TC32 Pre-scale Counter Current Value Register (TC32PCNT) 0xF3003090

Bit	Name	Default	R/W	Description
31:24	Reserved	0x00	R	
23:0	PCNT	0x0000000	R	Pre-scale counter current value. The AHB system clock must be three times faster than the frequency of ZCLK to read valid value.

TC32 Main Counter Current Value Register (TC32MCNT) 0xF3003094

Bit	Name	Default	R/W	Description
31:0	MCNT	0x00000000	R	Main counter current value. When RSYNC is enabled, the AHB system clock must be faster than the frequency calculated below. $(ZCLK \text{ frequency}) / (\text{PREScale} + 1) * 3$

TC32 Interrupt Control Register (TC32IRQ)

0xF3003098

Bit	Name	Default	R/W	Description
31	IRQCLR	0	R/W	Interrupt Clear Control. When this bit is 0, interrupt status bits (IRQRSTAT) are cleared by reading this register. When this bit is set, IRQSTAT bits are cleared only if written with non-zero value.
30	RSYNC	0	R/W	Synchronization control for Counter Current Value Registers (TC32PCNT and TC32MCNT). 0 = Enable, 1 = Disable.
29:24	BITSEL	0x00	R/W	Counter bit selection value for interrupt generation. Any one of the counter bits {MCNT[31:0], PCNT[23:0]} selected by BITSEL is used to generate an interrupt. 0x00 ~ 0x17 : PCNT[0] ~ PCNT[23] 0x18 ~ 0x38: MCNT[0] ~ MCNT[31]
23:21	Reserved	0	R/W	
20	IRQEN[4]	0	R/W	Enable Interrupt at the rising edge of a counter bit selected by BITSEL.
19	IRQEN[3]	0	R/W	Enable Interrupt at the end of pre-scale count
18	IRQEN[2]	0	R/W	Enable Interrupt at the end of count
17	IRQEN[1]	0	R/W	Enable Interrupt when the counter value matched with CMP1
16	IRQEN[0]	0	R/W	Enable Interrupt when the counter value matched with CMP0
15:13	Reserved	0	R/W	
12:8	IRQRSTAT	0x00	R/W	Interrupt Raw Status. Refer to the description for IRQEN above.
7:5	Reserved	0	R/W	
4:0	IRQMSTAT	0x00	R/W	Masked Interrupt Status = IRQRSTAT & IRQEN

26 I2C CONTROLLER

26.1 Functional Description

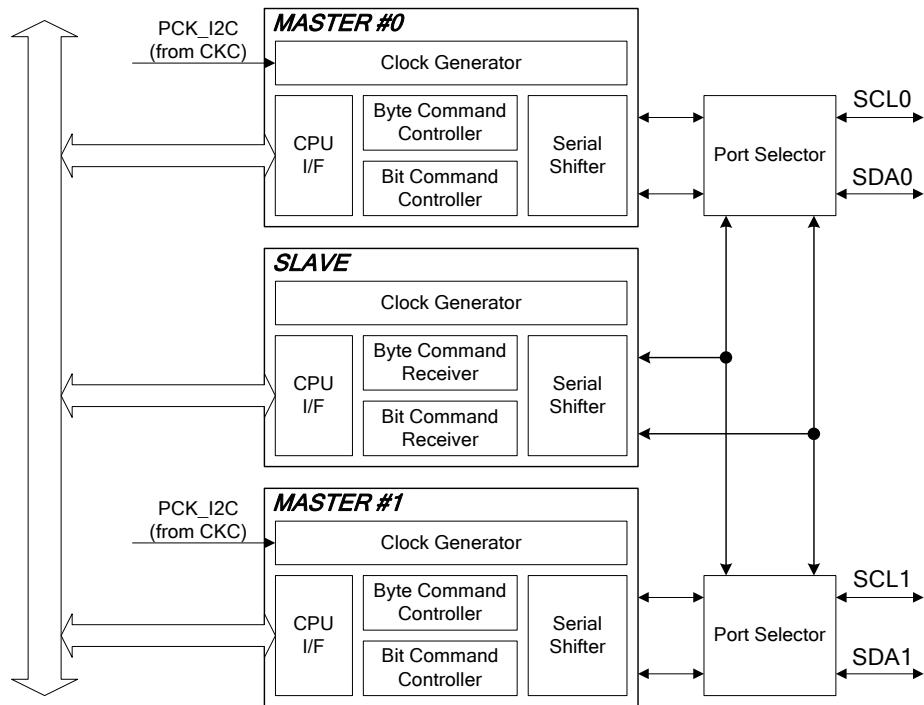


Figure 26.1 I2C Block Diagram

The I2C controller in the TCC79XX has two masters and one slave controller. The each master can control the dedicated I2C bus and the slave controller can select the one of two I2C buses. For each I2C bus can be selected by the register.

26.2 I2C Slave Function

- Supports Fast mode (400Kbps)
- Supports wait state by clock stretching
- 4byte TX FIFO, 4byte RX FIFO and 8byte Buffer

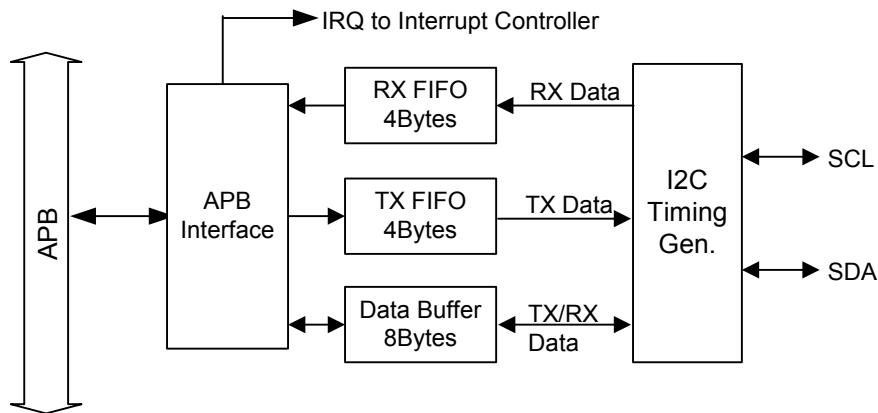


Figure 26.2 I2C Slave Block Diagram

The following figure and table show an example format for accessing Buffers and FIFOs in the I2C slave core.

	MSB	LSB	MSB	LSB	MSB	LSB
S	SLAVE ADDRESS	R/W	A	SUB ADDRESS	A	DATA BYTE0
	7bits	1	1	8bits	8bits	A

S: Start Condition
 R/W: Read (1) or Write(0)
 A: Acknowledge from/to the core

SUB ADDRESS[7:0]	DATA[7:0] Source / Destination	Note
0x00 ~ 0x03	MB0 (Data Buffer 0) Byte 0 ~ 3	
0x04 ~ 0x07	MB1 (Data Buffer 1) Byte 0 ~ 3	
0x08 ~ 0x7F	Reserved	NACK is sent to a master
0x80 ~ 0xFF	RX/TX FIFO	All addresses are directed to the RX/TX FIFO

26.3 Related Blocks

After the signals are enabled, PCK_I2C (the main clock of I2C) must be enabled and configured to the proper frequency. Refer to PCK_I2C related descriptions on page 23-10.

For internal synchronization, the APB clock frequency must be faster than the PCK_I2C frequency.

$$f_{PCK_I2C} \leq f_{HCLK} / 4.0$$

26.4 Register Description

Table 26.1 I2C Register Map (Base Address = 0xF0052000)

Ch	Name	Addr. Offset	Type	Reset	Description
Master 0	PRES	0x00	R/W	0xFFFF	Clock Prescale register
	CTRL	0x04	R/W	0x0000	Control Register
	TXR	0x08	W	0x0000	Transmit Register
	CMD	0x0C	W	0x0000	Command Register
	RXR	0x10	R	0x0000	Receive Register
	SR	0x14	R	0x0000	Status Register
	TIME	0x18	R/W	0x0000	Timing Control Register
Master 1	PRES	0x40	R/W	0xFFFF	Clock Prescale register
	CTRL	0x44	R/W	0x0000	Control Register
	TXR	0x48	W	0x0000	Transmit Register
	CMD	0x4C	W	0x0000	Command Register
	RXR	0x50	R	0x0000	Receive Register
	SR	0x54	R	0x0000	Status Register
	TIME	0x58	R/W	0x0000	Timing Control Register
Slave	PORT	0x80	R/W	-	Data Access port (TX/RX FIFO)
	CTL	0x84	R/W	0x00000000	Control register
	ADDR	0x88	W	0x00000000	Address register
	INT	0x8C	W	0x00000000	Interrupt Enable Register
	STAT	0x90	R	0x00000000	Status Register
	MBF	0x9C	R/W	0x00000000	Buffer Valid Flag
	MB0	0xA0	R/W	0x00000000	Data Buffer 0 (Byte 3 ~ 0)
Status	MB1	0xA4	R/W	0x00000000	Data Buffer 1 (Byte 7 ~ 4)
	IRQSTR	0xC0	R	0x00000000	IRQ Status Register

Prescale Register (PRES)

0xF0052000, 0xF0052040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Clock Prescale data

This register is used to prescale the SCL clock line. Due to the structure of the I2C interface, the core uses a 5*SCL clock internally. The prescale register must be programmed to this 5*SCL frequency (minus 1). Change the value of the prescale register only when 'EN' bit is cleared.

Example :

CLK Input frequency = 8MHz , Desired SCL frequency = 100KHz

Prescale = (8MHz / 100KHz) – 1 = 15

Control Register (CTR)

0xF0052004, 0xF0052044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				0				EN	IEN	MOD					RESERVED

EN [7]

I2C Core enable bit

- | | |
|---|----------|
| 0 | Disabled |
| 1 | Enabled |

IEN [6]

I2C Core interrupt enable bit

- | | |
|---|----------|
| 0 | Disabled |
| 1 | Enabled |

MOD [5]

I2C Data Width

- | | |
|---|------------|
| 0 | 8bit Mode |
| 1 | 16bit Mode |

Transmit Register (TXR)

0xF0052008, 0xF0052048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Transmit Data

When CTRL[5] is set, in case of 16Bit Mode is selected, Transmit Data bit width become 16 bit. Default mode is 8bit mode.

Command Register (CMD)

0xF005200C, 0xF005204C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								STA	STO	RD	WR	ACK	RESERVE	IACK	

STA [7]	Start Condition Generation
0	Disabled
1	Enabled
STO [6]	Stop Condition Generation.
0	Disabled
1	Enabled
RD [5]	Read From Slave
0	Disabled
1	Enabled
WR [4]	Write to Slave
0	Disabled
1	Enabled
ACK [3]	Sent ACK
0	Enabled
1	Disabled
IACK [0]	Interrupt Acknowledge
0	-
1	Clear a pending interrupt

Receive Register (RXR)

0xF0052010, 0xF0052050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Receive Data

When CTRL[5] is set, in case of 16Bit Mode is selected, Transmit Data bit width become 16 bit. Default mode is 8bit mode.

Status Register (SR)

0xF0052014, 0xF0052054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RxACK	BUSY	AL				TIP	IF

RxACK [7]

Received acknowledge from slave

- | | |
|---|-------------------------|
| 0 | Acknowledge received |
| 1 | No Acknowledge received |

BUSY [6]

I2C Bus Busy

- | | |
|---|---------------------------------|
| 0 | '0' after STOP signal detected |
| 1 | '1' after START signal detected |

AL[5]

Arbitration lost

- | | |
|---|------------------------------------|
| 0 | The core does not lose arbitration |
| 1 | The core loses arbitration |

Arbitration is lost when a STOP signal is detected, but non requested master drives SDA high, but SDA is low

TIP [1]

Transfer in progress

- | | |
|---|-------------------|
| 0 | Transfer Complete |
| 1 | Transferring Data |

IF [0]

Interrupt Flag

- | | |
|---|----------------------|
| 0 | - |
| 1 | Interrupt is pending |

Timing Register (TR)

0xF0052018, 0xF0052058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RC										CKSEL	FC				

RC [15:8]	Recovery time counter load value
value	“0” disables recovery time counter.
	The recovery time counter is enabled and loaded with RC[7:0] whenever a STOP condition is issued by the core. Execution of a new command written to CMD register is delayed until the counter is expired.
	Actual wait time = (I2CCLK period) * PRES[15:0] * 5 * RC[7:0]

CKSEL [5]	Clock Source Select
0	I2CCLK from Clock controller
1	PCLK (HCLK) divided by 2

Recommended if PCLK is not variable during system operation.

FC[4:0]	Noise filter counter load value
value	“0” disables noise filter. SCL and SDA inputs are checked for stability until the counter is expired.

Data Port (DPORT)

0xF0052080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	TXVC	-	-	-	-	-	-	-	RXVC	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PORT

TXVC [26:24]

TX buffer valid entry count

value TX buffer valid entry count

RXVC [18:16]

RX buffer valid entry count

value RX buffer valid entry count

PORT[7:0]

TX/RX FIFO access port

value TX / RX FIFO access port

Master Write Cycle to RX FIFO

	MSB	LSB	MSB	LSB	MSB	LSB									
S	WRITE SLAVE ADDRESS	W(0)	A	WRITE 1XXXXXXXb	A	WRITE DATA BYTE0	A	WRITE DATA BYTE1	A	WRITE DATA BYTE2	A	WRITE DATA BYTE3	A		

7bits 1 1 8bits

8bits

Master Read Cycle to TX FIFO

	MSB	LSB	MSB	LSB	MSB	LSB									
S	WRITE SLAVE ADDRESS	R(1)	A	always read as 11111111b	A	READ DATA BYTE0	A	READ DATA BYTE1	A	READ DATA BYTE2	A	READ DATA BYTE3	A		

7bits 1 1 8bits

8bits

The first byte is always read as 0xFF. The first byte must be discarded by the master

Control Register (CTL)

0xF0052084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV	-						FC								DRQEN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	TXTH	-		RXTH	-		--		RCLR	WS	SDA	CLR	-	EN	

SLV [31:30]		Pin Configuration for Slave Core
2 master operation		
00/11		I2C Channel 0 : Master 0 I2C Channel 1 : Master 1
1 master / 1 slave		
01		I2C Channel 0 : Slave I2C Channel 1 : Master 1
1 master / 1 slave		
10		I2C Channel 0 : Master 0 I2C Channel 1 : Slave
FC[24:20]		Filter Counter Load Value
value		“0” disables noise filter. SCL and SDA inputs are checked for stability until the counter is expired.
DRQEN [19:16]		DMA Request Enable
value		DMA request enable
TXTH [13:12]		TX FIFO threshold for DMA Request
value		TX FIFO threshold value
RXTH [9:8]		RX FIFO threshold for DMA Request
value		RX FIFO threshold value
RCLR [5]		Clear Interrupt Status at read cycle
0		-
1		Clear
WS [4]		Wait Status Control by SCL Stretching
0		Enable
1		Disable
SDA [3]		Reserved for test
0		-
1		-

CLR [2]	Clear FIFO
0	-
1	Clear

EN [0]	Enable for this slave core
0	Disable
1	Enable

Address Register (ADDR)																0xF0052088	
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																	
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																ADDR	-

ADDR [7:1]	Device Address
value	Slave address

Interrupt Register (INT)

0xF005208C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQSTAT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQEN															

IRQST[bit]	Interrupt Status
27th bit	All bytes of data buffer have been read by a master
26th bit	All bytes of data buffer have been written by a master
25th bit	Data buffer has been read by a master
24th bit	Data buffer has been written by a master
23rd bit	TX FIFO under run
22nd bit	RX FIFO over run
21st bit	TX bus cycle started with TX FIFO empty
20th bit	RX FIFO full
19th bit	TX FIFO empty
18th bit	RX FIFO not empty
17th bit	TX FIFO Level (TXVC <= TXTH)
16th bit	RX FIFO Level (RXVC <= RXTH)

IRQEN[bit]	Interrupt Enable
11th bit	All byte of data buffer has been read by a master
10th bit	All byte of data buffer has been written by a master
9th bit	Data buffer has been read by a master
8th bit	Data buffer has been written by a master
7th bit	TX FIFO under run
6th bit	RX FIFO over run
5th bit	TX bus cycle started with TX FIFO empty
4th bit	RX FIFO full
3rd bit	TX FIFO empty
2nd bit	RX FIFO not empty
1st bit	TX FIFO Level (TXVC <= TXTH)
0th bit	RX FIFO Level (RXVC <= RXTH)

Note: When RCLR bit of Control Register is “1”, IRQSTAT bits are cleared at the end of read cycle.

When RCLR bit of Control Register is “0”, IRQSTAT bits are cleared by writing “1”.

Status Register (STAT)

0xF0052090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	DDIR	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SADR

DDIR [3]

Data Direction

0 RX

1 TX

SADR[7:0]

Slave address received

value Slave address received from address cycle.

Buffer Valid Flag Register (MBF)

0xF005209C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

MBFT [23:16]	Buffer TX Flag
-----------------	----------------

value	<p>Buffer TX flag for byte 7 ~ 0.</p> <p>When the data buffer 0/1 is read by an external master, the corresponding bit is set. MBFT[3:0] is cleared when the CPU writes to MB0 and MBFT[7:4] is cleared when MB1 is written by the CPU</p>
-------	--

MBFR [7:0]	Buffer RX Flag
---------------	----------------

value	<p>Buffer RX flag for byte 7 ~ 0.</p> <p>When the data buffer 0/1 is read by an external master, the corresponding bit is set. MBFT[3:0] is cleared when the CPU writes to MB0 and MBFT[7:4] is cleared when MB1 is written by the CPU</p>
-------	--

Data buffer 0(MB0)

0xF00520A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Data byte 3								Data byte 2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data byte 1								Data byte 0							

Data buffer 1(MB1)

0xF00520A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Data byte 7								Data byte 6							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data byte 5								Data byte 4							

NOTE: Deviation from the standard.

The MB0 and MB1 registers can not be read with the standard I2C read timing sequence. Always TX FIFO is accessed with the standard read timing. To read MB0 or MB1, the sequence below must be followed. If a master is not capable of this non-standard sequence, MB0 and MB1 can not be read.

	MSB	LSB	MSB	LSB	MSB	LSB	
S	WRITE SLAVE ADDRESS	R(1)	A	WRITE 00000XXXb	A	READ DATA BYTE0	A
	7bits	1	1	8bits	8bits		

IRQSTR

0xF00520C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
0																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
														ST2	ST1	ST0

ST2

IRQ Status of I2C Slave Controller

IRQ Status Flag for I2C Slave Controller

0/1 This would be read as '1' when the IRQ status of I2C slave controller is activated.

ST1

IRQ Status of I2C Master Controller Channel 1

IRQ Status Flag for I2C Master Controller of Channel 1.

0/1 This would be read as '1' when the IRQ status of I2C master controller channel 1 is activated.

ST0

IRQ Status of I2C Master Controller Channel 0

IRQ Status Flag for I2C Master Controller of Channel 0.

0/1 This would be read as '1' when the IRQ status of I2C master controller channel 0 is activated.

27 ADC INTERFACE**27.1 Overview**

The TCC79XX has 8 channel general purpose low-power ADC for battery level detection, key detection, remote control interface, touch screen interface, etc. It is a CMOS type 10bit A/D converter with 8-channel analog input multiplexer.

- Resolution : 10-bit
- Maximum Conversion Rate : 1MSPS
- Main Clock : 5MHz (Max.)
- Standby Mode
- Input Range : 0.0V ~ VDDA_ADC

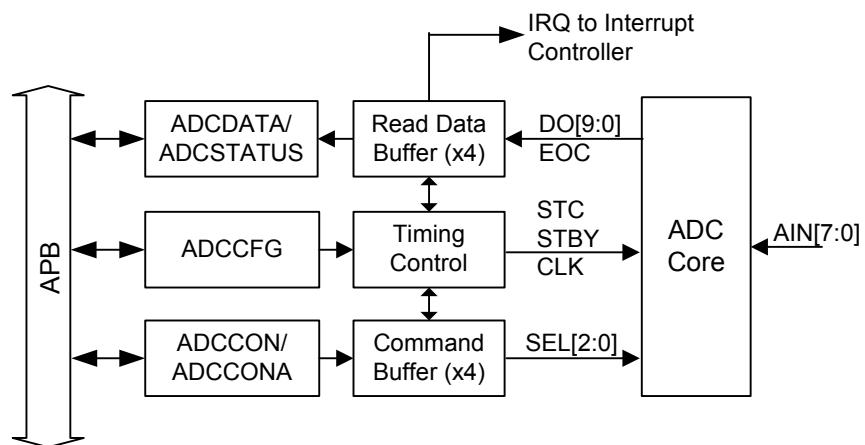


Figure 27.1 ADC Controller Block Diagram

Except for the APB interface, the ADC controller module runs with PCK_ADC from the Clock Generator module. The clock input is always divided before sent to the ADC core. The PCK_ADC register of Clock Generator and CLKDIV bits of ADCCFG register must be programmed to get desired frequency. The maximum frequency of CLK signal in Figure 27.1 must not exceed 5MHz. Refer to PCK_ADC on page 23-10

When one of the ADCCON or ADCCONA register is written with a channel number (SEL[2:0]), the SEL value is posted to the Command Buffer. The ADC Core starts conversion cycle as long as the Command Buffer is not empty. After the conversion cycle is completed, the result is written in Read Data Buffer. The data can be read from either ADCDATA or ADCSTATUS register. Up to four different SEL values can be posted to the Command Buffer. When the buffer is full, data written to ADCCON/ADCCONA registers are ignored. Various operating options can be set by using ADCCFG register.

27.2 ADC Controller Register Description

Table 27.1 ADC Controller Register Map (Base Address = 0xF3004000)

Name	Address	Type	Reset	Description
ADCCON	0x00	R/W	0x00000018	ADC Control Register
ADCDATA	0x04	R	Unknown	ADC Data Register
ADCCONA	0x80	R/W	0x00000018	ADC Control Register A
ADCSTATUS	0x84	R/W	Unknown	ADC Status Register
ADCCFG	0x88	R/W	0x00002400	ADC Configuration Register

ADC Control Register (ADCCON) 0xF3004000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Field	Name	RW	Reset	Description
4	STB	RW	0x0	ADC goes to standby mode when this bit is '1', otherwise starts operating
2 ~ 0	ASEL	RW	0x0	ADC Input Select AIN n pin is selected as ADC input signal

ADC Data Register (ADCDATA) 0xF3004004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															

Field	Name	RW	Reset	Description
10 ~ 1	ADATA	RW	-	ADC data = adc * All the AD input levels must be within the range that is from 0V to VDDADC (the main power level of ADC). Do not exceed the limit.
0	FLG	RW	0x0	If it is '1', it indicates that A/D conversion has finished and data is stable. Otherwise A/D conversion is on processing and data is unstable.

ADC Control Register A (ADCCONA)																0xF3004080			
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																0xF3004080			
Reserved																			
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																			
Reserved																			

*) This register has the same functionality as that of ADCCON register. Only the register address is different.

ADC Status Register (ADCSTATUS)																0xF3004084			
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																0xF3004084			
R WBVCNT R RBVCNT Reserved RSELV																RSELV			
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																RSELV			
Reserved RBDATA																RBDATA			

Field	Name	RW	Reset	Description
31	R	R	0	Reserved
30 ~ 28	WBVCNT	R	0x0	Command write buffer valid entry count. Up to 4 entries with different ASEL values can be posted to command buffer.
27	R	R	0	Reserved
26 ~ 24	RBVCNT	R	0	Read data buffer valid entry count. Up to 4 entries.
23 ~ 19	Reserved	R	0	Reserved
18 ~ 16	RSEL	R	X	Input channel number for current read data. Up to 4 entries.
15 ~ 10	Reserved	R	0	Reserved
9 ~ 0	RBDATA	R	X	Read buffer data

ADC Configuration Register (ADCCFG) 0xF3004088

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKDIV				DLYSTC			NEOC	0	FIFOTH		IRQE	R8	APD	SM	

Field	Name	RW	Reset	Description
15 ~ 12	CLKDIV	RW	0x2	Clock divisor value. ADCLK is divided by ((CCLKDIV + 1) * 2).
11 ~ 8	DLYSTC	RW	0x4	Delay from SEL to STC (Start of Conversion) in ADC core CLK count. Whenever SEL value changes, delay is inserted.
7	NEOC	RW	0	For test purpose only. Must be written as "0"
6	Reserved	RW	0	Reserved
5 ~ 4	FIFOTH	RW	0	FIFO threshold for interrupt assertion. Interrupt will be asserted only if FIFOTH < (# of valid entry).
3	IRQE	RW	0	Interrupt enable
2	R8	RW	0	When this bit is "1", two LSBs are truncated. (shift right). Only "ADCSTATUS" register is affected by this bit.
1	ADP	RW	0	Auto power down enable. This bit is effective only if SM bit (described below) is "1". After conversion cycle is done, the ADC core is forced to power down mode.
0	SM	RW	0	Single Mode Enable. When it is set to 0, ADC conversion cycle is repeated endlessly with the input selected by ASEL bits. When it is set to 1, only one cycle is executed.

28 RTC (REAL-TIME CLOCK)

28.1 Overview

The Real Time Clock (RTC) unit can operate by the backup battery although the system power turns off. The RTC transmits data to CPU as BCD (binary coded decimal) values. The data includes second, minute, hour, date, day of the week, month, and year. The RTC unit works with an external 32.768kHz crystal and also can perform the alarm function. The block diagram is shown in Figure 16.1.

Features

- Clock and calendar functions (BCD display): seconds, minutes, hours, date, day of week, month, year
- Leap year generator
- Wake-up (PMWKUP) signal generation: support on the power down mode (PWDN)
- Alarm interrupt (ALMINT) in normal operation mode
- Power Supply Voltage: System power supply(3.0V, 1.2V), Backup battery (3.0V)

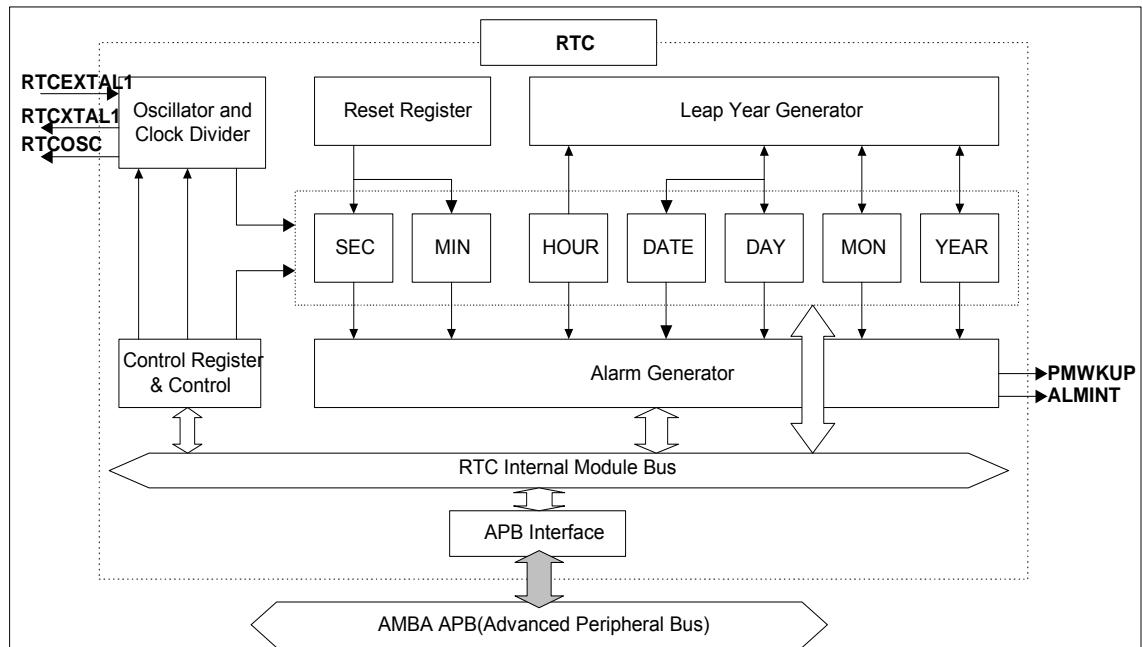


Figure 28.1 RTC Block Diagram

28.2 Function Description

28.2.1 System Clock Frequency Control

The leap year generator calculates whether the last date of each month is 28, 29, 30 or 31 based on data from BCDDAY, BCDMON, and BCDYEAR. This also considers leap years in deciding the last date. A 16 bit counter can just represent four BCD digits, so it can decide whether any year is a leap year or not.

28.2.2 System Power Operation

It is required to set bit 1 of the RTCCON register for interfacing between CPU and RTC logic. An one second error can occur when the CPU reads or writes data into BCD counters and this can cause the change of the higher time units. When the CPU reads/writes data to/from the BCD counters, another time unit may be changed if BCDSEC register is overflowed. To avoid this problem, the CPU should reset BCDSEC register to 00h. The reading sequence of the BCD counters is BCDYEAR, BCDMON, BCDDATE, BCDDAY, BCDHOUR, BCDMIN, and BCDSEC. It is required to read it again from BCDYEAR to BCDSEC if BCDSEC is zero.

28.2.3 Backup Battery Operation

The RTC logic is driven by backup battery if the system power turns off. The interfaces of the CPU and RTC logic are blocked and the battery drives the oscillation circuit, clock divider (DIVIDER), and BCD counters (RTCFCN) to minimize power dissipation.

28.2.4 Alarm Function

The RTC generates alarm signal at specified time in the power down mode or normal operation mode. In normal operation mode, the alarm interrupt (ALMINT) is activated and in the power down mode the power management wake up (PMWKUP) signal is activated. The RTC alarm register, RTCALM, determines the alarm enable and the condition of the alarm time setting.

28.3 RTC Operation

28.3.1 Initial Settings of Registers after Power-on

All the registers should be set after the power is turned on.

28.3.2 RTC Time Setting

The following figure shows how to set the time when clock is stopped. This works when the entire calendar or clock is to be set.

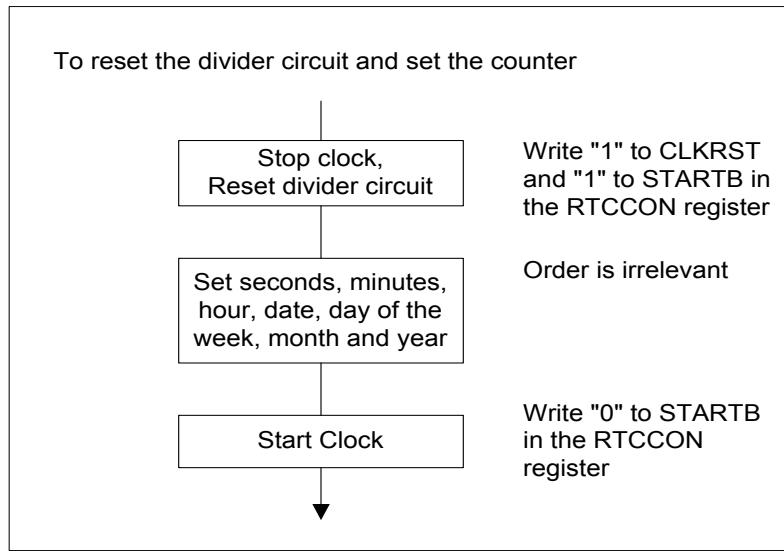


Figure 28.2 The RTC Time Setting Sequence

28.3.3 RTC Alarm Time Setting

The following figure shows how to use the alarm function. Alarms can be generated using the seconds, minutes, hours, days of week, date, month, year or any combination of these. Set the ALMEN bit (bit 7) in the register on which the alarm is placed to "1", and then set the alarm time. Clear the ALMEN bit in the register on which the alarm is placed to "0".

When the INTMODE bit of RTCIM register is high, and the clock and alarm times match, "1" is set in the PEND bit of RTCPEND register. The detection of alarm can be checked with reading the PEND bit.

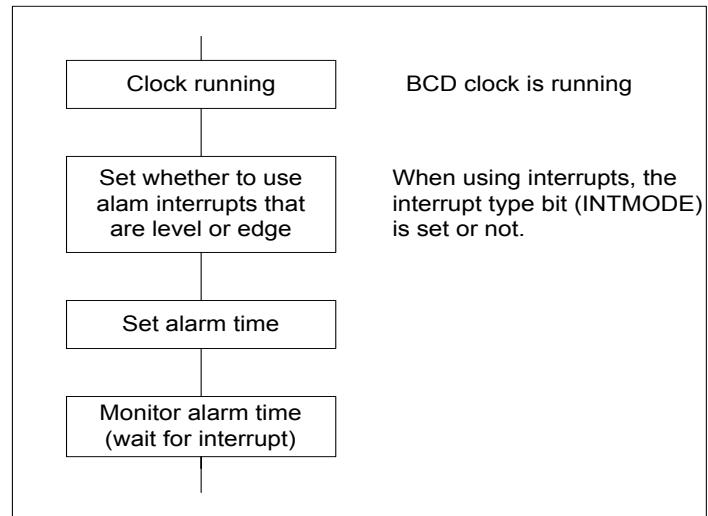


Figure 28.3 RTC Alarm Time Setting Sequence

28.3.4 RTCPEND Clear

The following figure shows how to Clear RTC interrupt

There are two types of interrupt.

- Alarm Interrupt (ALMINT).
- Wake-Up Interrupt (WKUPINT).

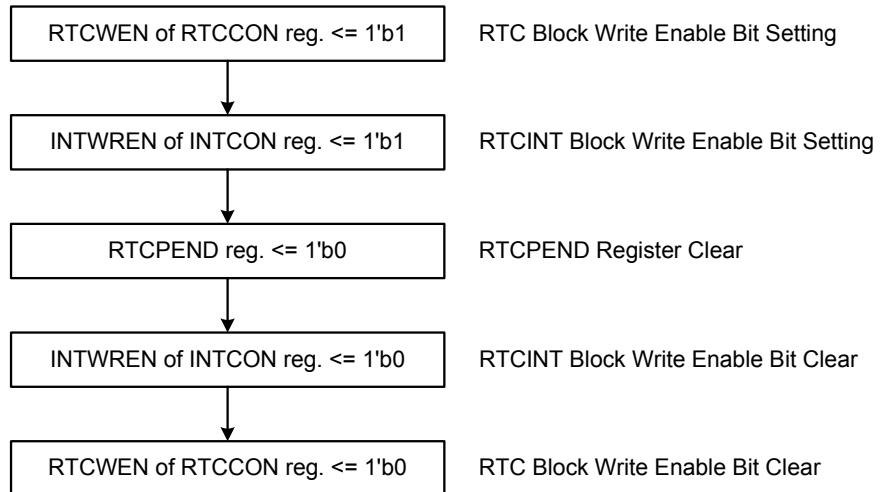


Figure 28.4 PEND Clear Sequence

28.3.5 RTC Operation

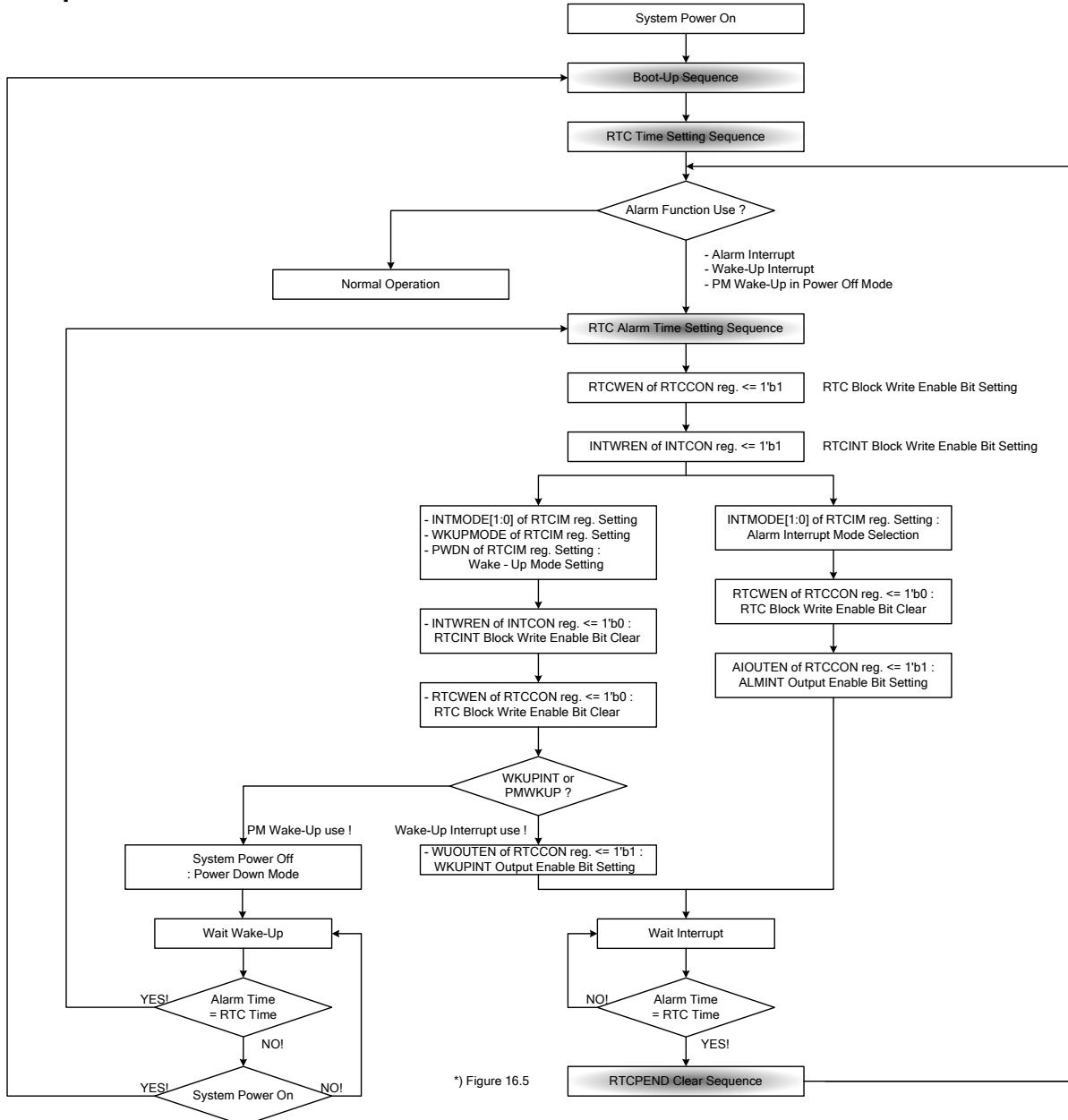


Figure 28.5 RTC Operation Process Flow Chart.

28.3.6 Crystal Oscillator Circuit

Crystal oscillator circuit constants (recommended values) are shown in the following table and the RTC crystal oscillator circuit in the following figure.

Table 28.1 Recommended Oscillator Circuit Constants

fosc	C1	C2	Rf
32.768 kHz	from 10 pF to 22 pF	from 10 pF to 22 pF	> 5MΩ

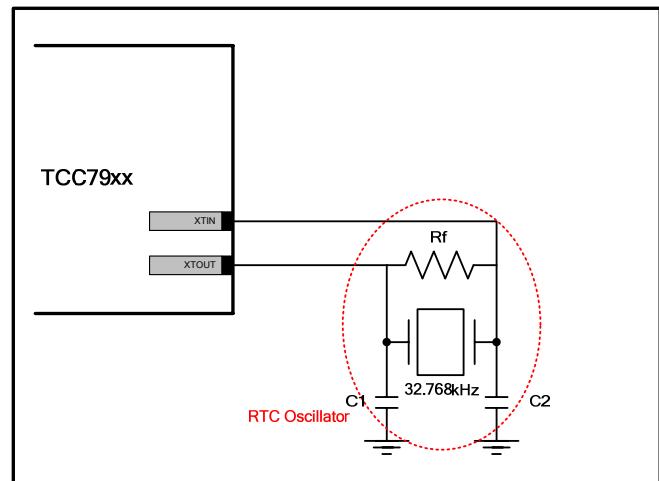


Figure 28.6 Example of Crystal Oscillator Circuit Connection

28.4 Programmer's model

28.4.1 Register memory map

Table 28.2 RTC Register Map (Base Address = 0xF3002000)

Register	Address	R/W	Reset value	Description
RTCCON	0x00	R/W	0x00	RTC Control Register
INTCON	0x04	R/W	-	RTC Interrupt Control Register
RTCALM	0x08	R/W	-	RTC Alarm Control Register
ALMSEC	0x0C	R/W	-	Alarm Second Data Register
ALMMIN	0x10	R/W	-	Alarm Minute Data Register
ALMHOUR	0x14	R/W	-	Alarm Hour Data Register
ALMDATE	0x18	R/W	-	Alarm Date Data Register
ALMDAY	0x1C	R/W	-	Alarm Day of Week Data Register
ALMMON	0x20	R/W	-	Alarm Month Data Register
ALMYEAR	0x24	R/W	-	Alarm Year Data Register
BCDSEC	0x28	R/W	-	BCD Second Register
BCDMIN	0x2C	R/W	-	BCD Minute Register
BCDHOUR	0x30	R/W	-	BCD Hour Register
BCDDATE	0x34	R/W	-	BCD Date Register
BCDDAY	0x38	R/W	-	BCD Day of Week Register
BCDMON	0x3C	R/W	-	BCD Month Register
BCDYEAR	0x40	R/W	-	BCD Year Register
RT CIM	0x44	R/W	-	RTC Interrupt Mode Register
RTCPEND	0x48	R/W	-	RTC Interrupt Pending Register

28.4.2 Register Description

RTC Control Register (RTCCON)

0xF3002000

RTCCON register consists of 8-bits such as STARTB that controls of running the normal counters, RTCWEN that controls the read/write enable of the BCD registers, CLKSEL, CNTSEL, and CLKRST for BCD counters testing.

RTCWEN bit controls all interfaces between the CPU and the RTC, so it should be set to '1' in an initialization routine to enable data transfer after a system reset. Instead of working BCD with 1Hz, CLKSEL bit enables the operation of BCD counters with an external clock which is applied through the pin EXTAL1 to the test BCD counters. CNTSEL bit converts the dependent

The operation of BCD counters into independent counters for the test. CLKRST resets the frequency divided logic in the RTC.

OSCEN bit controls the path from input of Crystal to the output of divider logic. If this bit is high, the output of divider is 1 Hz clock. This bit is implemented to test the oscillator circuit and divider block.

INTWREN bit controls the path from the RTCIF Block to the RTCINT Block.

Table 28.3 RTC Control Register (RTCCON)

RTCCON	Bit	Initial State	Description	
STARTB	[0]	0	RTC start bit	
			0: RUN	1: Halt
RTCWEN	[1]	0	RTC write enable bit	
			0: Disable	1: Enable
			BCD counter test clock set bit	
CLKSEL	[2]	0	0: EXTAL1 divided clock (1 Hz)	
			1: Reserved (EXTAL1: 32.768 kHz)	
			BCD count test type set bit	
CNTSEL	[3]	0	0: Merge BCD counters	
			1: Reserved (Separate BCD counters)	
CLKRST	[4]	0	RTC clock count set bit	
			0: No reset	1: reset
			Oscillator and Divider circuit test enable bit	
OSCEN	[5]	0	0: Disable	1: Enable
			Alarm Interrupt Output Enable	
AIOUTEN	[6]	0	0: Disable	1: Enable
			Wake Up Output Enable	
WUOUTEN	[7]	0	0: Disable	1: Enable

RTC Interrupt Control Register (INTCON)

0xF3002004

Table 28.4 RTC Interrupt Control Register (INTCON)

INTCON	Bit	Initial State	Description
INTWREN	[0]	Undef.	Interrupt Block Write Enable bit 0: Disable 1: Enable
STATUS	[2:1]	Undef.	User Define Status Register

*) To Write INTCON register, you must set RTCWEN (RTCCON[1]).

RTC Alarm Control Register (RTCALM)

0xF3002008

RTCALM register determines the alarm enable and the condition of the alarm time setting. Note that RTCALM register generates the alarm signal through ALMINT in normal operation mode.

Alarms can be generated using the seconds, minutes, hours, day of week, date, month, year or any combination of these. Set the ALMEN bit (bit 7) in the register on which the alarm is placed to "1", and then set the alarm time. Clear the ALMEN bit in the register on which the alarm is placed to "0".

Table 28.5 RTC Alarm Control Register (RTCALM)

RTCALM	Bit	Initial State	Description
SECEN	[0]	Undef.	Second alarm enable bit 0 : Disable 1 : Enable
MINEN	[1]	Undef.	Minute alarm enable bit 0 : Disable 1 : Enable
HOUREN	[2]	Undef.	Hour alarm enable bit 0 : Disable 1 : Enable
DATEEN	[3]	Undef.	Date alarm enable bit 0 : Disable 1 : Enable
DAYEN	[4]	Undef.	Day of week alarm enable bit 0 : Disable 1 : Enable
MONEN	[5]	Undef	Mon alarm enable bit 0 : Disable 1 : Enable
YEAREN	[6]	Undef	Year alarm enable bit 0 : Disable 1 : Enable
ALMEN	[7]	Undef.	Alarm global enable bit 0: Disable 1 : Enable

*) To Write RTCALM register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0]).

To Read RTCALM register, you must set INTWREN (INTCON[0]).

Alarm Second Data Register (ALMSEC)

0xF300200C

Table 28.6 Alarm Second Data Register (ALMSEC)

ALMSEC	Bit	Initial State	Description
SECDATA	[6:0]	Undef.	BCD value for alarm second bits [3:0] bit is from 0 to 9 [6:4] bit is from 0 to 5

*) To Write ALMSEC register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0]).

To Read ALMSEC register, you must set INTWREN (INTCON[0]).

Alarm Minute Data Register (ALMMIN)

0xF3002010

Table 28.7 Alarm Minute Data Register (ALMMIN)

ALMMIN	Bit	Initial State	Description
MINDATA	[6:0]	Undef.	BCD value for alarm second bits [3:0] bit is from 0 to 9 [6:4] bit is from 0 to 5

*) To Write ALMMIN register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMMIN register, you must set INTWREN (INTCON[0]).

Alarm Hour Data Register (ALMHOUR)

0xF3002014

Table 28.8 Alarm Hour Data Register (ALMHOUR)

ALMHOUR	Bit	Initial State	Description
HOURDATA	[5:0]	Undef.	BCD value for alarm hour bits [3:0] bit is from 0 to 9 [5:4] bit is from 0 to 2

*) To Write ALMHOUR register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMHOUR register, you must set INTWREN (INTCON[0]).

Alarm Date Data Register (ALMDATE)

0xF3002018

Table 28.9 Alarm Date Data Register (ALMDATE)

ALMDATE	Bit	Initial State	Description
DATEDATA	[5:0]	Undef.	BCD value for alarm date, from 0 to 28, 29, 30, 31 (decimal: 01 ~ 31) [3:0] bit is from 0 to 9 [5:4] bit is from 0 to 3

*) To Write ALMDATE register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMDATE register, you must set INTWREN (INTCON[0]).

Alarm Day of Week Data Register (ALMDAY)

0xF300201C

Table 28.10 Alarm Day of Week Data Register (ALMDAY)

ALMDAY	Bit	Initial State	Description
DAYDATA	[2:0]	Undef.	BCD value for alarm day bits [2:0] bit is from 0 to 6 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday

*) To Write ALMDAY register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMDAY register, you must set INTWREN (INTCON[0]).

Alarm Month Data Register (ALMMON)

0xF3002020

Table 28.11 Alarm Month Data Register (ALMMON)

ALMMON	Bit	Initial State	Description
BCD value for alarm month bits			
MONDATA	[4:0]	Undef.	[3:0] bit is from 0 to 9 [4] bit is from 0 to 1

*) To Write ALMMON register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMMON register, you must set INTWREN (INTCON[0]).

Alarm Year Data Register (ALMYEAR)

0xF3002024

Table 28.12 Alarm Year Data Register (ALMYEAR)

ALMYEAR	Bit	Initial State	Description
BCD value for alarm year bits			
YEARDATA	[15:0]	Undef.	[7:0] bit is from 0 to 99 [15:8] bit is from 0 to 99

*) To Write ALMYEAR register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read ALMYEAR register, you must set INTWREN (INTCON[0]).

BCD Second Data Register (BCDSEC)

0xF3002028

Table 28.13 BCD Second Data Register (BCDSEC)

BCDSEC	Bit	Initial State	Description
BCD value for second bits			
SECDATA	[6:0]	Undef.	[3:0] bit is from 0 to 9 [6:4] bit is from 0 to 5

*) To Write BCDSEC register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]).

To Read BCDSEC register, you must set INTWREN (INTCON[0]).

BCD Minute Data Register (BCDMIN)

0xF300202C

Table 28.14 BCD Minute Data Register (BCDMIN)

BCDMIN	Bit	Initial State	Description
BCD value for minute bits			
MINDATA	[6:0]	Undef.	[3:0] bit is from 0 to 9 [6:4] bit is from 0 to 5

*) To Write BCDMIN register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]).

To Read BCDMIN register, you must set INTWREN (INTCON[0]).

BCD Hour Data Register (BCDHOUR)

0xF3002030

Table 28.15 BCD Hour Data Register (BCDHOUR)

BCDHOUR	Bit	Initial State	Description
HOURDATA	[5:0]	Undef.	BCD value for hour bits [3:0] bit is from 0 to 9 [5:4] bit is from 0 to 2

*) To Write BCDHOUR register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]).

To Read BCDHOUR register, you must set INTWREN (INTCON[0]).

BCD Date Data Register (BCDDATE)

0xF3002034

Table 28.16 BCD Date Data Register (BCDDATE)

BCDDATE	Bit	Initial State	Description
DATEDATA	[5:0]	Undef.	BCD value for date bits(decimal : 01 ~ 31) [3:0] bit is from 0 to 9 [5:4] bit is from 0 to 3

*) To Write BCDDATE register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]).

To Read BCDDATE register, you must set INTWREN (INTCON[0]).

BCD Day of Week Data Register (BCDDAY)

0xF3002038

Table 28.17 BCD Day of Week Data Register (BCDDAY)

BCDDAY	Bit	Initial State	Description
DATEDAY	[2:0]	Undef.	BCD value for date bits [2:0] bit is from 0 to 6 000 : Sunday 001 : Monday 010 : Tuesday 011 : Wednesday 100 : Thursday 101 : Friday 110 : Saturday

*) To Write BCDDAY register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]).

To Read BCDDAY register, you must set INTWREN (INTCON[0]).

BCD Month Data Register (BCDMON)

0xF300203C

Table 28.18 BCD Month Data Register (BCDMON)

BCDMON	Bit	Initial State	Description
MONDATA	[4:0]	Undef.	BCD value for month bits [3:0] bit is from 0 to 9 [4] bit is from 0 to 1

*) To Write BCDMON register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]).

To Read BCDMON register, you must set INTWREN (INTCON[0]).

BCD Year Data Register (BCDYEAR)

0xF3002040

Table 28.19 BCD Year Data Register (BCDYEAR)

BCDYEAR	Bit	Initial State	Description
YEARDATA	[15:0]	Undef.	BCD value for year bits [7:0] bit is from 0 to 99 [15:8] bit is from 0 to 99

*) To Write BCDYEAR register, you must set STARTB (RTCCON[0]) and RTCWEN (RTCCON[1]).

To Read BCDYEAR register, you must set INTWREN (INTCON[0]).

RTC Interrupt Mode Register (RTCIM)

0xF3002044

When the INTMODE bit of RTCIM register is high, and the clock and alarm times match, "1" is set in the PEND bit of RTCPEND register. The detection of alarm can be checked with reading the PEND bit.

Table 28.20 RTC Interrupt Mode Register (RTCIM)

RTCIM	Bit	Initial State	Description
INTMODE	[1:0]	Undef.	Interrupt mode selection bit x0: Disable alarm interrupt mode. x1: Enable alarm interrupt mode. 01: Supports on the edge alarm interrupt. 11: Supports on the level alarm interrupt.
WKUPMODE	[2]	Undef.	Wakeup mode selection bit 0: PMWKUP active low 1: PMWKUP active high
PWDN	[3]	Undef.	Operation mode selection bit 0: Normal Operation Mode 1: Power Down Mode

*) To Write RTCIM register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read RTCIM register, you must set INTWREN (INTCON[0]).

To generate the alarm interrupt, you must be set INTMODE (RTCIM[0] = 1), PWDN (RTCIM [3] = 0) and RTCALM register should be set properly. To generate PMWKUP signal, PWDN register should be set to power down mode.

RTC Interrupt Pending Register (RTCPEND)

0xF3002048

Table 28.21 RTC Interrupt Pending Register (RTCPEND)

RTCPEND	Bit	Initial State	Description
PEND	[0]	0	Interrupt pending enable bit 0 : PEND bit is cleared. 1 : PEND bit is pending.

*) To Clear RTCPEND register, you must set RTCWEN (RTCCON[1]) and INTWREN (INTCON[0])

To Read RTCPEND register, you must set INTWREN (INTCON[0]).

29 EXTERNAL MEMORY CONTROLLER

29.1 Overview

The TCC79XX has a memory controller for various kinds of memories for digital media en-decoding system. It can manipulate SDRAM, Flash (NAND, NOR type), ROM, SRAM type memories. The data bus width can be configured for each chip select separately

The memory controller provides the power saving function for SDRAM (self refresh).

The following figure represents the block diagram of memory control unit.

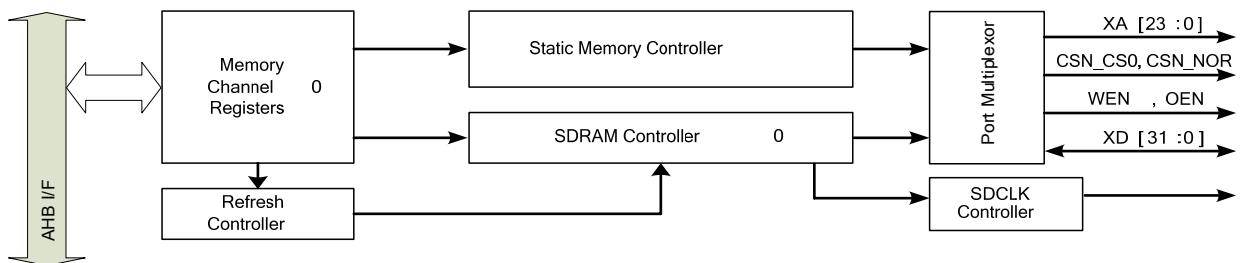


Figure 29.1 Memory Controller Block Diagram

Table 29.1 Channel 0 Memory Controller Register Map (Base Address = 0xF1000000)

Name	Address	Type	Reset	Description
SDCFG	0x00	R/W	0x6A484C00	SDRAM Configuration Register
SDFSM	0x04	R	0x00000000	SDRAM FSM Status Register
MCFG	0x08	R/W	0x01000042	Miscellaneous Configuration Register
TST	0x0C	W	0x00000000	Should not write to this – it's for TEST
CSCFG0	0x10	R/W	0x468AC809	External Chip Select 0 Config. Register (CSN_CS0)
-	0x14	R/W	0x508AD01A	Reserved
-	0x18	R/W	0x608AD03A	Reserved
CSCFG3	0x1C	R/W	0x728AD01A	External Chip Select 3 Config. Register (CSN_NOR)
CLKCFG	0x20	R/W	0x00000A05	Memory Controller Version & Periodic Clock Enable Count Register
SDCMD	0x24	R/W	-	SDRAM Command Write Register
SDCFG1	0x28	R/W	0xFFFFFFFF	Extra SDRAM Configuration Register

Table 29.2 NAND flash Register Map (Base Address = N * 0x10000000)

Name	Address	Type	Reset	Description
NDCMD	0x00	R/W	-	Command Cycle Register
NDLADR	0x04	W	-	Linear Address Cycle Register
NDRADR	0x08	W	-	Row Address Cycle Register
NDIADR	0x0C	W	-	Single Address Cycle Register
NDDATA	0x10	R/W	-	Data Access Cycle Register

*) N represents BASE field of configuration register (CSCFGx) for each chip select.

*) This table is used for reading/writing through memory port.

29.2 SDRAM Controller

SDRAM controller can control SDRAM of which column address width is 8 ~ 10bits and row address width is 11~13-bit and bank address width is 1 ~ 2-bit, and data bus is 16 or 32-bit. The SDRAM can contain most parts for system operation. (Program, data, buffer, etc can be located in SDRAM).

The SDRAM parameter such as size, refresh period, RAS to CAS delay, refresh to idle delay can be programmed by internal register. The registers for SDRAM controller are as follows. Refer to SDRAM cycle diagram in Figure 29.2

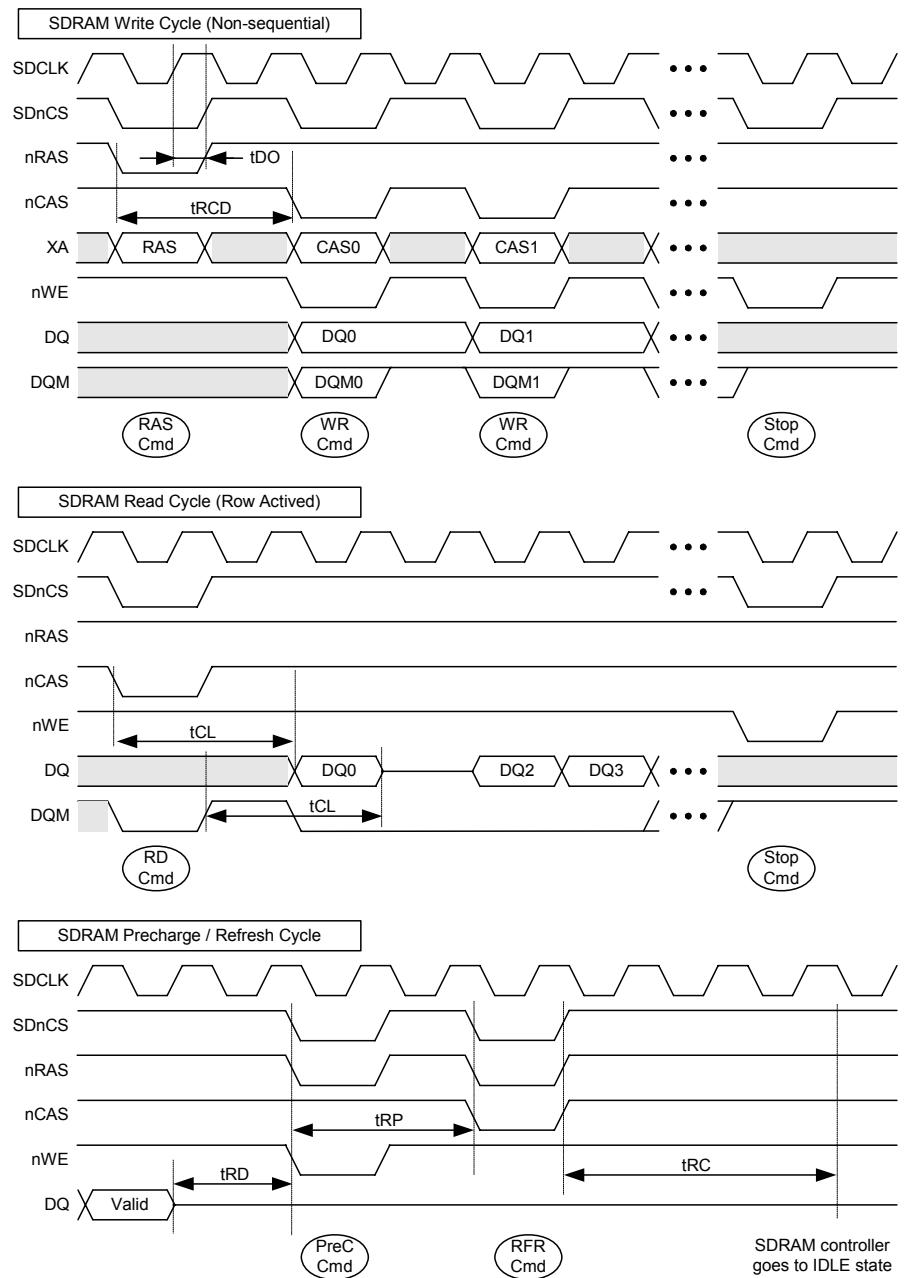


Figure 29.2 SDRAM Cycle Diagram

SDRAM Configuration Register (SDCFG)															0xF1000000		
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																	
CL BW CW RC RCD WR 0 0 0 0 0 AM 0 PPD SR															0	0	0
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0															1	0	0
RP	RW	0	0	0	0	0	0	0	0	0	AM	0	PPD	SR			

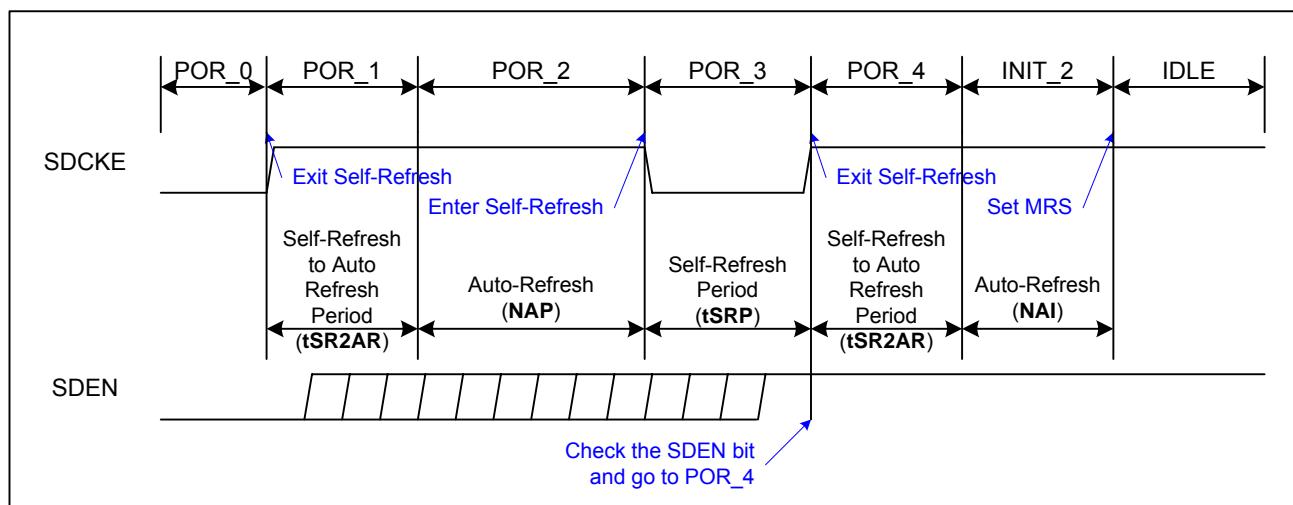
Field	Name	RW	Reset	Description
0	SR	RW	0	Self refresh enable register 0 : Disabled(exit from self-refresh state) 1 : Enabled(enter self-refresh state)
1	PPD	RW	0	Before entering self-refresh state, the instruction and data in the SDRAM should be written back.
3	AM	RW	0	Precharge power-down mode enable 0 : Disabled, 1 : Enabled
11 ~ 10	RW	RW	3	Address matching option 0 : BANK-RAS-CAS address order 1 : RAS-BANK-CAS address order
15 ~ 12	RP	RW	4	The address bit width for SDRAM banks is two.
19 ~ 18	WR	RW	2	RAS address width configuration register {0} : 12 bits RAS address width {2} : 13 bits RAS address width {1, 3} : 11 bits RAS address width
23 ~ 20	RCD	RW	4	Precharge-to-Refresh cycle parameter
27 ~ 23	RC	RW	10	Write-to-Read wait cycle parameter
29 ~ 28	CW	RW	2	RAS to CAS cycle parameter
30	BW	RW	1	Refresh-to-Idle cycle parameter
31	CL		0	CAS address width {0,1} : 8 bits CAS address width {2} : 9 bits CAS address width {3} : 10 bits CAS address width
				SDRAM Bus width 0 : 32 bits data bus 1 : 16 bits data bus
				CAS latency 0 : 2 Cycles 1 : 3 Cycles

SDRAM Configuration Register1 (SDCFG1)

0xF1000028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
tSR2AR								tSRP							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAP								NAI							

Field	Name	RW	Reset	Description
7 ~ 0	NAI	RW	255	Auto-refresh count in INIT_2 state
15 ~ 8	NAP	RW	255	Auto-refresh count in POR_2 state
23 ~ 16	tSRP	RW	255	Minimum cycles of self-refresh active period Minimum self-refresh period.
31 ~ 24	tSR2AR	RW	255	Minimum cycles of exiting self-refresh state. Self-refresh to auto-refresh cycle time



The SDRAM controller starts from the POR reset and the state transition timing is shown in the above figure.

After the system reset, the SDRAM controller makes the SDCKE signal LOW to support the boot-up from self-refresh mode.

After POR_0, the SDCKE goes to HIGH for exiting the self-refresh mode and goes to POR_2 state consecutively. In the POR_2 state, the memory controller performs the auto-refresh commands of which count is defined in NAP field.

The state of SDRAM controller will go to POR_3 state after POR_2. At POR_3 state, the SDRAM controller waits until SDEN being high. If SDEN goes to high, the state would be changed to POR_4.

In the POR_4, the SDRAM controller waits for the specified period defined by tSR2AR field. After waiting, the state goes to INIT_2 state in which the auto-refresh command would be executed NAI times.

SDRAM FSM Status Register (SDFSM)																0xF100004
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SDFSM
0																SDFSM

This register is read only and represents current status of finite state machine in the SDRAM controller. This can be used for test purpose only.

SDFSM	Description
01_0000b	SD_IDLE
01_0001b	SD_ACT_ROW
01_0010b	SD_WRITE
01_0011b	SD_READ
01_0100b	SD_WRITEA
01_0101b	SD_READA
01_0110b	SD_ACT_IDLE
01_0111b	SD_PRE_ALL
01_1000b	SD_PRE
01_1001b	SD_AUTO_REFRESH
01_1010b	SD_SELF_REFRESH
01_1111b	SD_CMD

29.3 Miscellaneous Configuration

Miscellaneous Configuration Register (MCFG)

0xF1000008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
X(don't care – read-only area)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Field	Name	RW	Reset	Description
0	ERF	RW	0	External refresh clock enable 0 : Disabled, 1 : Enabled <i>Before setting, the external refresh clock should be enabled</i>
1	CKM	RW	1	SDRAM clock masking bit 0 : Masked, 1 : Unmasked
2	RP	RW	0	Refresh pending enable bit 0 : Disabled, 1 : Enabled
3	RE	RW	0	Refresh emergent control enable 0 : Disabled, 1 : Enabled <i>If enabled and the pending count is reached to 7, the emergent refresh would be executed consecutively until the pending count being under 4.</i>
4	SDEN	RW	0	SDRAM controller enable bit 0 : Disabled, 1 : Enabled <i>Before enabled, the SDRAM configuration registers should be initialized with specific values according to the SDRAM.</i>
5	ECKE	RW	1	CKE Enable bit 0 : SDCKE(Low), 1 : SDCKE(Controlled)
6	ENCK	-	0	SDCLK dynamic control enable bit 0 : Direct Output, 1 : Gating Controlled <i>If enabled, the memory controller would mask the SDCLK to reduce the power consumption. The ratio of masked clock period is determined by CLKCFG register.</i>
7	SDW	RW	0	Wait 1 cycle for reading from the SDRAM. 0 : Disabled, 1 : Enabled
10 ~ 8	RPENDM	R	0	The Pended Refresh Count Register
12 ~ 11	BW	R	X	Read-only registers for external configuration
14	XDM	RW	1	Data output mode control register 0 : Default input mode 1 : Default output mode

If set, in the idle state, the direction of data bus is output.

External ready status bit

15	RDY	R	X	<i>This flag can monitor the state of READY pin. The READY pin is used to extend the access cycle for the external memory access by setting the URDY bit of each configuration register and also can be used as the ready flag by polling the state of this bit, especially for NAND flash interfacing.</i>
----	-----	---	---	---

29.4 External Memory Controller

External memory controller can control external memories such as NAND or NOR type flash memory and ROM, SRAM type memory. These memories are selected by nCS3 ~ nCS0 pins. The cycle parameter for accessing external memory can be configured by internal registers. In case of NAND flash, additional parameters for address, command and data cycles can be provided.

External Chip Select n Configuration Register (CSCFGx)

0xF1000010 + (x*4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSBASE				OD	WD	SIZE		TYPE		URDY	RDY	AM	PSIZE		LA[2]
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LA[1:0]	STP		PWD								HLD				

*) The reset value of each CSCFGx register means the following configuration for each chip select.

Chip Select 0 (CSN_CS0) : 16bit, SRAM, Base = 0x40000000, tSTP=1, tPW=1, tHLD=1

Chip Select 3 (CSN_NOR): 16bit, NOR, Base = 0x70000000, tSTP=2, tPW=3, tHLD=2

CSBASE [31:28]	Chip Select n Base Address
M	This is fixed and read-only CS0 : 0x4, CS3 : 0x7
OD	Half-Cycle Delay 'nOE' Signal
0	Normal STP and HLD timing would be applied.
1	When STP and HLD are zero, 1 cycle would be added to tPW and then tSTP and tHLD of nOE signal become 1/2 cycle.
WD	Half-Cycle Delay 'WE' Signal
0	Normal STP and HLD timing would be applied.
1	When STP and HLD are zero, 1 cycle would be added to tPW and then tSTP and tHLD of nWE signal become 1/2 cycle.
SIZE [25:24]	Bus Width Select
0, 1	Bus width = 32 bit
2	Bus width = 16 bit
3	Bus width = 8 bit

The SIZE of CS3 is determined by the BW during the initialization period.

TYPE [23:22]	Type of External Memory
0	NAND type
1	IDE type
2	SMEM_0 type (Ex : ROM, NOR flash) Byte-write control signal (DQM) is not needed.
3	SMEM_1 type (Ex : SRAM) Byte-write control signal (DQM) is needed.

URDY [21]	Use Ready
	<p>Ready / Busy signal monitoring is enabled</p> <p>The memory controller extends access cycle until the state of READY pin indicates that the access request has accomplished.</p>
1	<p>If the SDRAM is connected to channel 0 memory ports, it is strongly recommended that you DO NOT USE this function. These can cause that the SDRAM refresh cycle is lost by the long period of READY signal such as NAND flash writing operation.</p>
RDY [20]	Ready / Busy Select
0	<p>The READY pin indicates the READY signal.</p> <p>The memory controller extends access cycle until this pin goes to high state.</p>
1	<p>The READY pin indicates the BUSY signal.</p> <p>The memory controller extends access cycle until this pin goes to low state.</p>
*) Refer to Figure 29.3 for ready/busy cycle extension.	
AM [19]	Address Mask Bit
0	Upper half of data bus is masked to zero.
*) In case of 16bit width NAND flash, the upper half byte must be held low, during address cycles. This bit must be set to zero. But if the system uses multiple NAND flashes by sharing a chip select but separating each data to 16 or 32bit data bus of TCC78x, the AMSK must be set to 1, so the address can be fed to each NAND flashes.	
PSIZE [18:17]	Page size of NAND Flash
psize	<p>The size of one page for NAND type flash.</p> <p>It represents byte per page calculated by the following equation.</p> $1 \text{ Page} = 256 * 2\text{psize}$
*) Refer to Table 29.3 about the relationship between the address generation and each page size configuration.	
LA [16:14]	Number of Address Cycles
N	<p>The number of address command cycle for NAND type flash.</p> <p>(N+1) cycle is used for generating address cycle command.</p>
*) Refer to sub-register of NAND type memory for more information of PSIZE and CADR field.	
STP [13:11]	Number of Cycle for Setup Time (tSH)
N	N cycle is issued between the falling edge of nCS[n] and nOE / nWE.
PWD[10:3]	Number of Cycle for Pulse Width (tPW)
N (= 0~255)	(N+1) cycle is issued between the falling and rising edge of nOE / nWE.
HLD [2:0]	Number of Cycle for Hold Time (tHLD)
N	N cycle is issued between the rising edge of nOE / nWE and nCS[n].

The following figure displays the element cycle diagram for external memories.

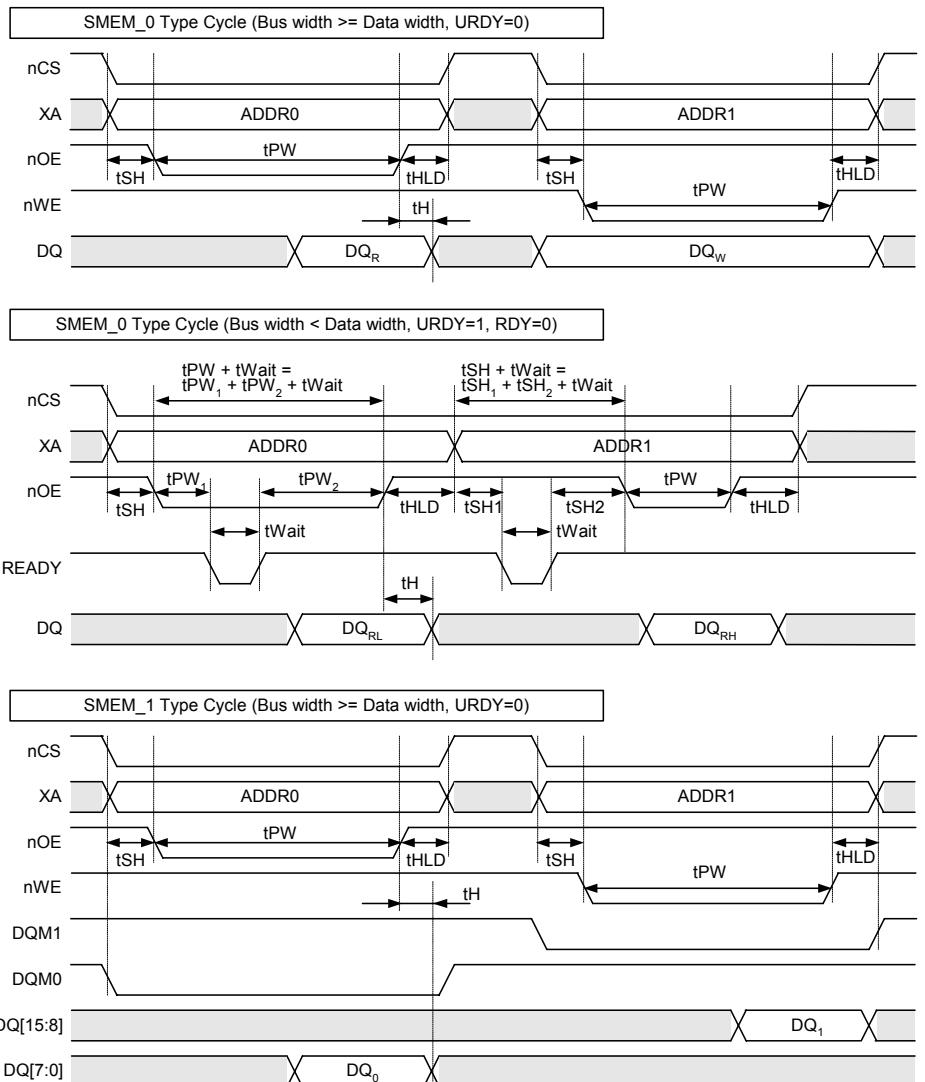


Figure 29.3 Basic Timing Diagram for External Memories

In case of IDE type memory, there are two chip-enable signals for it. In the TCC79XX, each enable signal can be controlled by offset address space. 'nCS0' reflects that the offset address range of 0 ~ 0x1F is accessed, 'nCS1' reflects that 0x20 ~ 0x3F is accessed. For larger address than 0x3F, bit5 of address value means which enable signal is activated. (0 to 'nCS0', 1 to 'nCS1')

29.5 Sub-registers of NAND type memory

In case of NAND flash type memories, there are several sub-registers for generating command, address, and data cycles. The followings describe the sub-registers. (M is base field of CSCFGx register). Except the data register (NDDATA), the sub-register has implicit size of 32bit, so the bus-width of CSCFGx register does not affect the cycle of command and address registers. It only affects the cycle of data register.

Command Cycle Register (NDCMD)

0x10000000*M

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDCMD3								NDCMD2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDCMD1								NDCMD0							

*) If bus width of NAND flash is more than 8bit, the NDCMD1 ~ 3 may be used as command register, otherwise only NDCMD0 is used as command register. The following values are an example commands for NAND flash of SAMSUNG. Refer to corresponding datasheet of NAND flash chip for more detailed list of command s.

0x00/0x01 : Page Read Command

0x80 : Page Program Command

0x60 : Block Erase Command

0x70 : Status Read Command (generated by reading from 0xM0000700 address)

Linear Address Cycle Register (NDLADR)

0x10000000 * M + 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDLADR[31:16]								NDLADR[15:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

*) By writing to this register, memory controller generates linear address cycle for NAND flash.

Row Address Cycle Register (NDRADR)

0x10000000 * M + 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDRADR[31:16]								NDRADR[15:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

*) By writing to this register, memory controller generates row address cycle for NAND flash.

Table 29.3 Page size of NAND Flash

# of Cycle	Address Generation			
	PSIZE = 0	PSIZE = 1	PSIZE = 2	PSIZE = 3
1st	ADR[7:0]	ADR[7:0]	ADR[7:0]	ADR[7:0]
2nd	ADR[15:8]	ADR[16:9]	ADR[10:8]	ADR[11:8]
3rd	ADR[23:16]	ADR[24:17]	ADR[18:11]	ADR[19:12]
4th	ADR[31:24]	ADR[31:25]	ADR[26:19]	ADR[27:20]
5th	-	-	ADR[31:27]	ADR[31:28]

*) ADR means address value that is written to NDLADR or NDRADR register. The shaded cycles represent row address cycles. That is, NAND address cycles start from there when NDRADR register is accessed.

Single Address Cycle Register (NDIADR)**0x10000000 * M + 0x0C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								NDIADR							

*) When CPU writes to this register, one cycle of address cycle is generated.

Data Register (NDDATA)**0x10000000 * M + 0x10**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDDATA3								NDDATA2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDDATA1								NDDATA0							

*) NDDATA3~1 may be used as the value of data register, otherwise only NDDATA0 is used as data register. It is dependant on the bus-width of CSCFGx register of NAND flash.

29.6 Miscellaneous Control Register

The clock for external SDRAM is controlled by this register. In the period of active, the HCLK is out with same frequency. But in the period of in-active, this register generates the divided HCLK or masks the clock outputs. This can save the power consumption in idle mode.

SDRAM Clock Control Register (CLKCFG) 0xF1000020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCLKCNT															

PCLKCNT [7:0] Masked Period of SDCLK

M This determines the period of SDCLK masked. If 0, the HCLK is out directly with same frequency.

SDRAM Command Register (SDCMD) 0xF1000024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSN	RAS	CAS	WEN	BA				CADR							

Field	Name	RW	Reset	Description
9 ~ 0	CADR	RW	0	CAS address ports
11 ~ 10	BA	RW	0	BANK address ports
12	WEN	RW	0	WEN port
13	CAS	RW	0	CAS port
14	RAS	RW	0	RAS port
15	CSN	RW	0	CSN_SD port
16	A10	RW	0	A10 port
17	CKE	RW	0	CKE port

To command the SDRAM directly, refer to the following configuration

- Burst length : The burst length should be '1' for 32 bits bus width. And '2' for 16 bits bus width
- CAS latency : The CAS latency to command directly should be same as the value of the corresponding bit-field in the SDCFG register.

30 SRAM INTERFACE

30.1 Overview

The TCC79XX has 8-bit SRAM interface (SRAMIF) with two address lines and two chip selectors.

When the corresponding SRAMIF register is accessed by a bus master, 80-type CPU interface signals are generated on SRAMIF ports. The relationship between the system address and SRAMIF CS/XA signals is shown in the Figure 30.1.

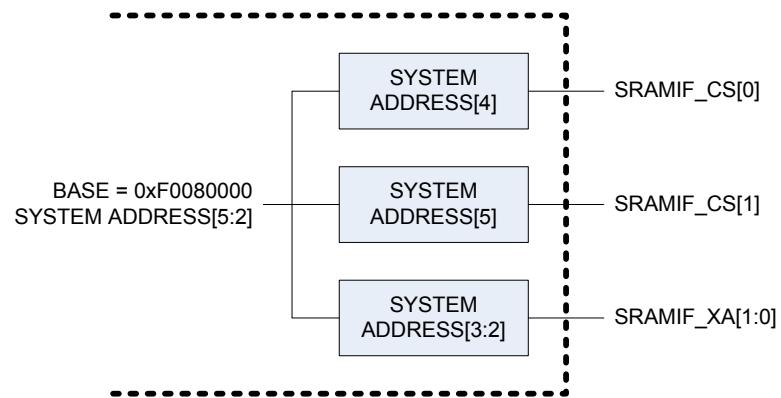


Figure 30.1 Relationship between the internal system address and SRAMIF CS/XA

30.2 Registers

All control registers for the SRAMIF are listed in Table 30.1.

Table 30.1 SRAMIF Register map (0xF0080000)

Name	Offset	Type	Reset	Description
CTRLCS0XA0	0x00	R/W	0xA0229011	Control register for CS0XA0 and CS0XA1.
CTRLCS0XA2	0x04	R/W	0xA0429021	Control register for CS0XA2 and CS0XA3.
CTRLCS1XA0	0x08	R/W	0xA0129009	Control register for CS1XA0 and CS1XA1.
CTRLCS1XA2	0x0C	R/W	0xA0229011	Control register for CS1XA2 and CS1XA3.
CS0XA0	0x10	R/W	-	If this register is read or written, reading or writing operations are generated on SRAMIF_CS0 while SRAMIF_XA = 0.
CS0XA1	0x14	R/W	-	If this register is read or written, reading or writing operations are generated on SRAMIF_CS0 while SRAMIF_XA= 1.
CS0XA2	0x18	R/W	-	If this register is read or written, reading or writing operations are generated on SRAMIF_CS0 while SRAMIF_XA= 2.
CS0XA3	0x1C	R/W	-	If this register is read or written, reading or writing operations are generated on SRAMIF_CS0 while SRAMIF_XA= 3.
CS1XA0	0x20	R/W	-	If this register is read or written, reading or writing operations are generated on SRAMIF_CS1 while SRAMIF_XA=0
CS1XA1	0x24	R/W	-	If this register is read or written, reading or writing operations are generated on SRAMIF_CS1 while SRAMIF_XA=1
CS1XA2	0x28	R/W	-	If this register is read or written, reading or writing operations are generated on SRAMIF_CS1 while SRAMIF_XA=2
CS1XA3	0x2C	R/W	-	If this register is read or written, reading or writing operations are generated on SRAMIF_CS1 while SRAMIF_XA=3

CTRLCSmXAn

0xF00800n0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BW[1]				W ⁶⁰ STP											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BW[0]				R ⁶¹ STP											

BIT	NAME	R/W	RESET	DESCRIPTION
31, 15	BW[1:0]	R/W		It determines the data width of the corresponding I/O register. 0 : 8 bits 1 : 16bits 3 : 32bits (default)
30-28 14:12	WSTP RSTP	R/W	Refer to Table 30.1	Setup Time N cycles are issued between the falling edge of nCS and the falling edge of nWR(Writing operation) or nRD(Reading operation).
27:19 11-3	WPW RPW	R/W		Pulse Width (N+1) cycles are issued between the falling edge of nWR(or nRD) and the rising edge of nWR(or nRD).
18-16 2-0	WHLD RHLD	R/W		Hold Time N cycles are issued between the rising edge of nWR(or nRD) and the rising edge of nCS.

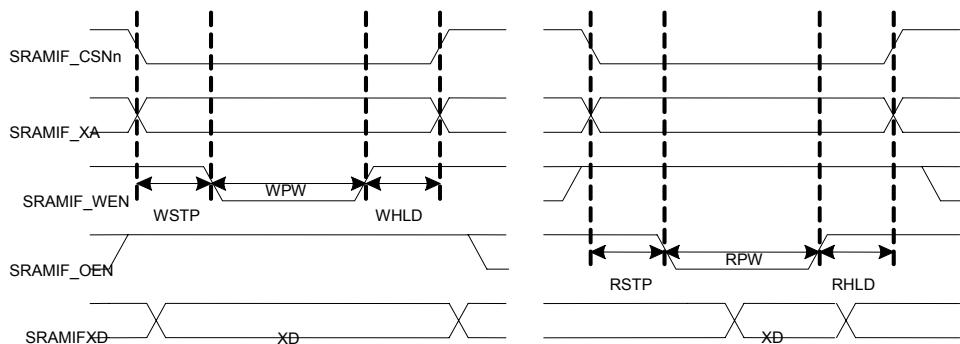


Figure 30.2 Timing configuration for SRAMIF output signals

When BW[1:0] is set to 8bits, the corresponding CSnXAm register is configured as 8-bit I/O device. Therefore, when CSnXAm register is accessed by 32-bit data operation, 4 writing or reading operations per 32-bit data are generated. Notice that the address (SRAMIF_XA[1:0]) is not incremented during these 4 operations. The address is only determined by m of CSnXAm register. Refer to Figure 30.3 (A).

When BW[1:0] is set to 32bits and CSnXAm register is accessed by 32-bit data operation, one write/read operation is only generated and least significant 8-bit data is only valid. Refer to Figure 30.3 (B).

⁶⁰ Prefix W means writing operation.

⁶¹ Prefix R means reading operation.

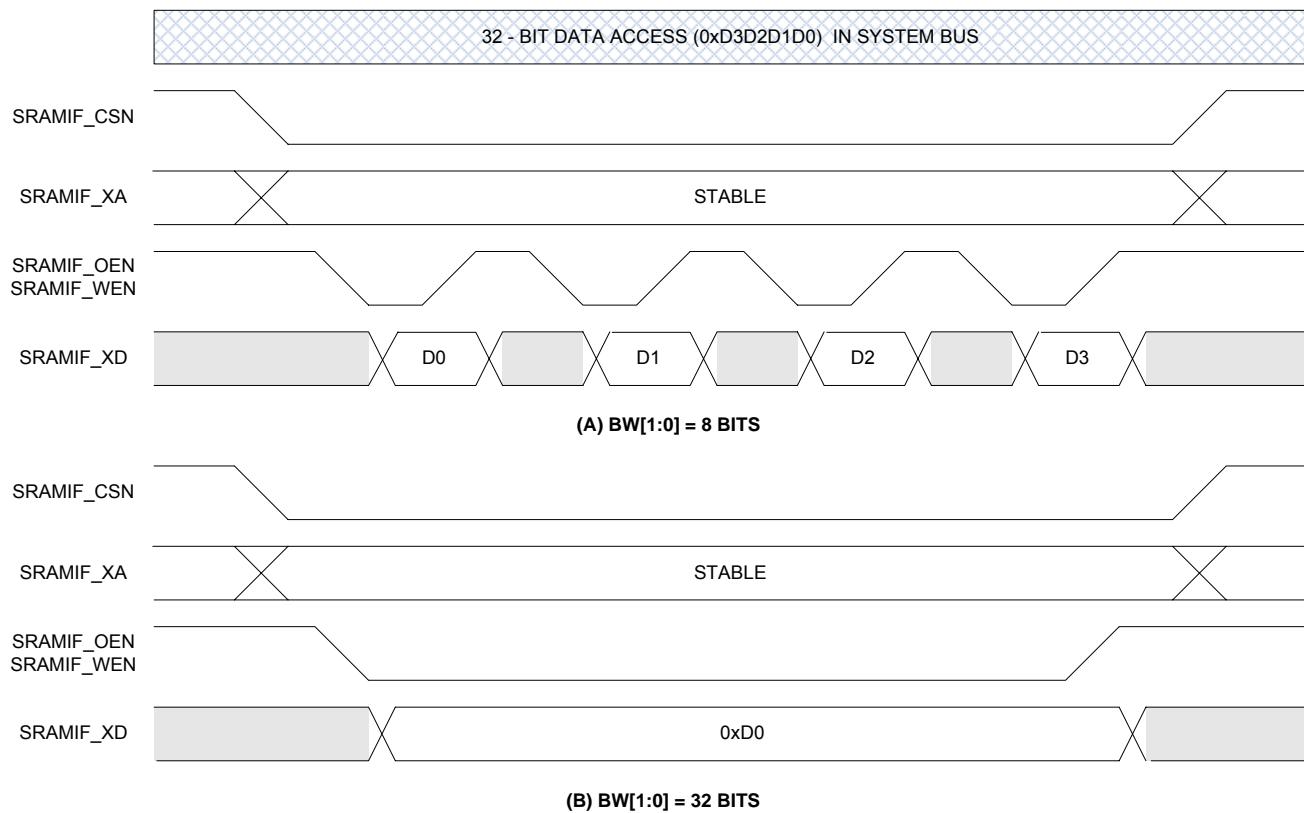


Figure 30.3 Interface Signals according to BW[1:0] Bits

31 MAILBOX**31.1 Overview**

The “MAILBOX” is used for communication between main processor and sub processor. The following figure shows the hardware block diagram.

The mailbox has two terminals for each processor, which has 8 words fifo and terminal controller. 8-word FIFO is used for transfer the messages and terminal controller is used for FIFO controller, status indicator, and interrupt generator to another processor.

The terminals of main and sub processors have the same structure.

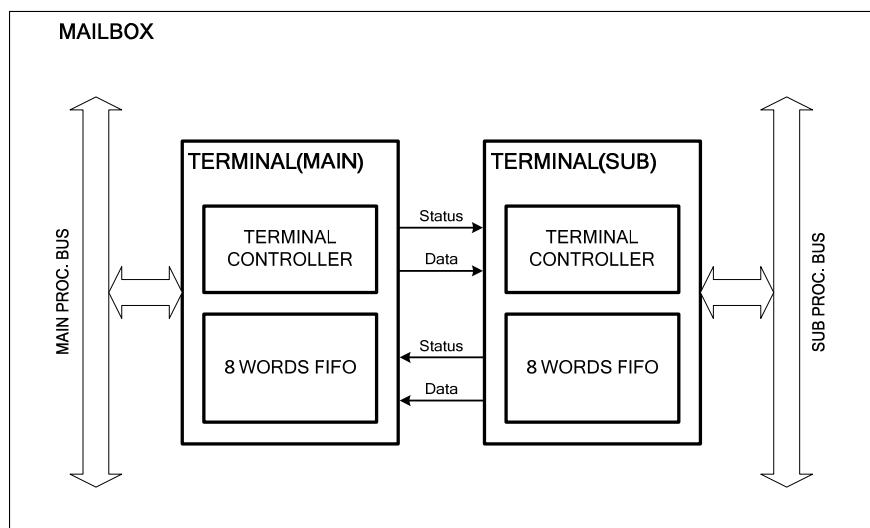


Figure 31.1 MAILBOX Hardware Block Diagram

Table 31.1 MAILBOX Register Map (Base Address = 0xF4000000)

Name	Address	Type	Reset	Description
MBOXTXD	0x00 ~ 0x1C	R/W	-	Transmit FIFO Data Region
MBOXRXD	0x20 ~ 0x3C	R/W	-	Receive FIFO Data Region
MBOXCTR	0x40	R/W	0x00015003	Mailbox Control Register
MBOXSTR	0x44	R/W	0x81828383	Mailbox Status Register

31.2 Register Descriptions

MBOXTXD Register Region 0xF4000000 ~ 0xF400001C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXD[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXD[15:0]															

Field	Name	RW	Reset	Description
31 ~ 0	TXD	W	-	Transmit FIFO Data Region If write data into the region, the data would be stored in transmit data FIFO.

* The reason for transmit FIFO data being a region is to be able to use the burst transfer by CPU.

MBOXRXD Register Region

0xF4000020 ~ 0xF400003C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXD[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXD[15:0]															

Field	Name	RW	Reset	Description
31 ~ 0	RXD	R	-	Received FIFO Data Region If read from the region, the data would be loaded in receive data FIFO.

* The reason for receive FIFO data being a region is to be able to use the burst transfer by CPU.

MBOXCTR Register

0xF4000040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEST															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									FLUSH	OEN	IEN	0	0		ILEVEL

Field	Name	RW	Reset	Description
1 ~ 0	ILEVEL	RW		Received FIFO level to generate interrupt 00b : not-empty 01b : greater equal to 2 10b : greater equal to 4 11b : full
4	IEN	RW		Received data interrupt enable
5	OEN	RW		Transmit data output enable
6	FLUSH	RW		Flush the transmit data FIFO
31	TEST	-		For Test Purpose

* If you want to transmit 4 words data to sub processor, you write the 4 words data with "OEN" being low. After writing done, you have to set "OEN" high. Then the stored data count will be passed to sub processor.

MBOXSTR Register

0xF4000044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	SCOUNT				-	-	SFUL	SEMP
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	MCOUNT				-	-	MFUL	MEMP

Field	Name	RW	Reset	Description
0	MEMP	R		Empty Status of Transmit FIFO
1	MFUL	R		Full Status of Transmit FIFO
7 ~ 4	MCOUNT	R		Stored data in Transmit FIFO (0 ~ 8, 0 is empty)
16	SEMP	RW		Empty Status of Receive FIFO
17	SFUL	RW	-	Full Status of Receive FIFO
23 ~ 20	SCOUNT	RW	-	Filled Count of counter-part (0 ~ 8, 0 is empty)

31.3 Operation & Timing Diagram

SendMessage : Transmit Data

An example of flow-chart for sending message to counter-part processor shows in the following figure.

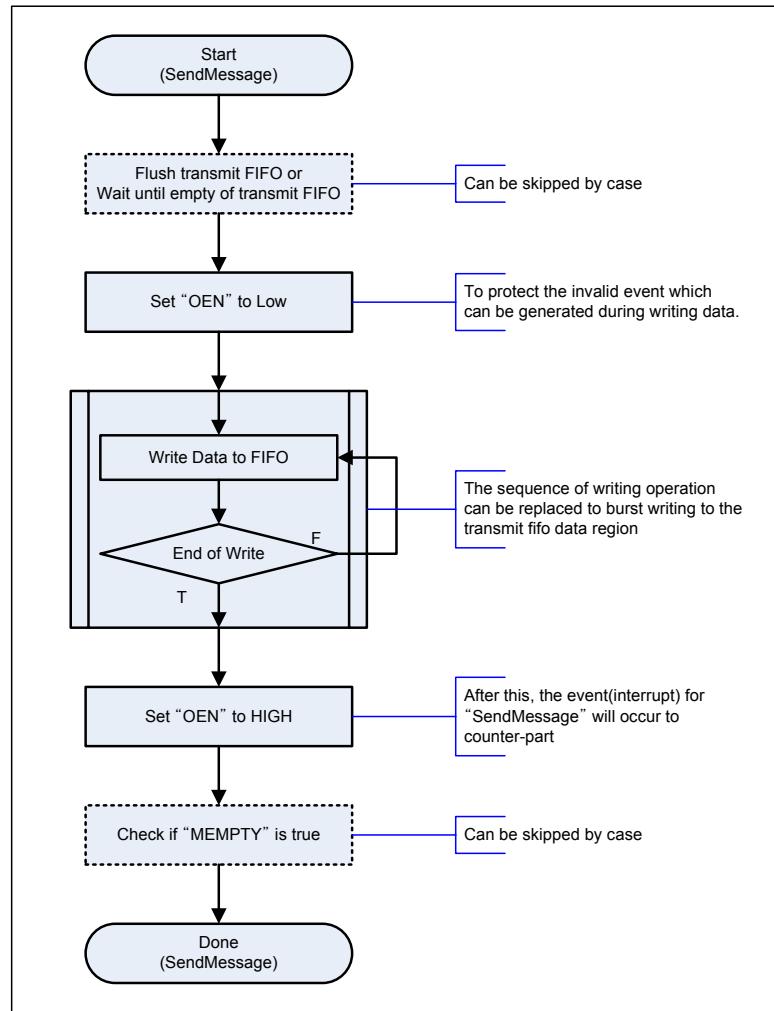


Figure 31.2 Flow-Chart for "SendMessage"

The generated interrupt to another processor would be cleared after all the messages being read.

RecvMessage : Receive Data

An example of flow-chart for receiving message from counter-part processor shows in the following figure.

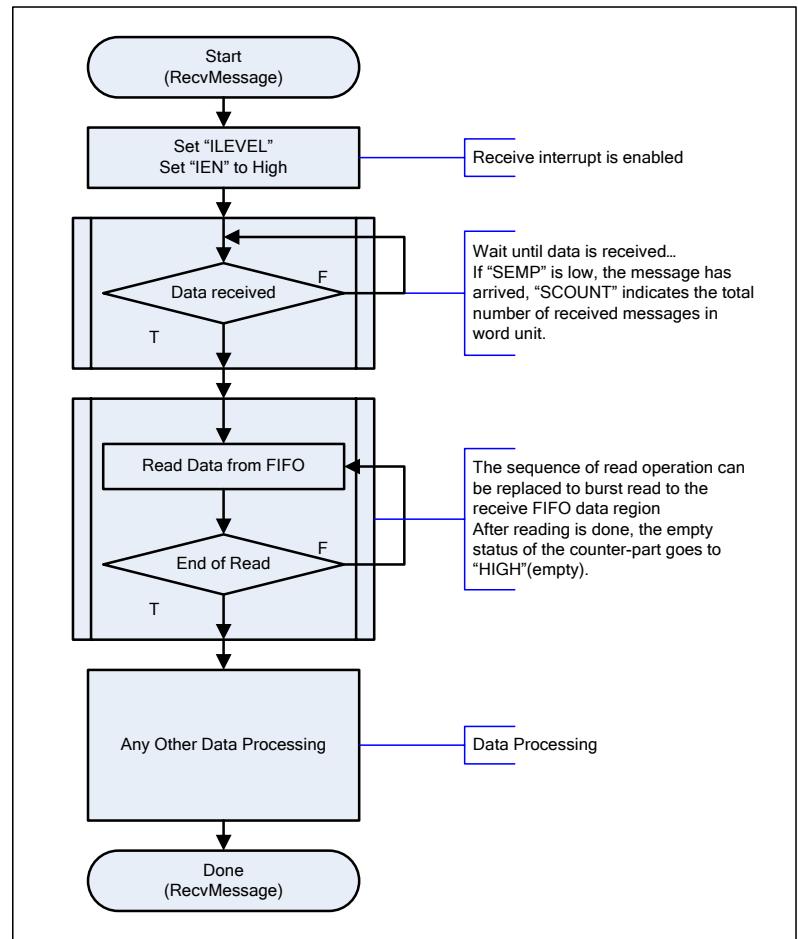


Figure 31.3 Flow-Chart for "RecvMessage"

Interrupt Type

The interrupt trigger type can be one of edge-triggered or level-triggered. In the level-triggered interrupt mode, the read operations should be in the handler routine.

32 MISCELLANEOUS CONTROLLER

The miscellaneous controller (MISC) is a very important hardware module which controls the TCC79XX system. This is composed of SYSC (SYStem Controller), IOBC (IO Bus Controller), and finally VMT (Virtual MMU Table).

The IOBC has 3 registers named as USB11H, IOBAPB, and STORAGE. The USB20D is to configure the USB 2.0 function (device) controller. The USB11H is to configure the USB 1.1 host controller. The IOBAPB register is to configure the I/O bus bridge. Finally, the STORAGE register is to configure the memory stick, UART channel 0, and ECC controller.

The SYSC has the 5 registers. The functions for remap of booting area, the JTAG interface type, and the configuration of video controller core are defined in the SYSC.

The VMT is used for MMU table for ARM926EJS specifically. If the application has no need of address translation table, the VMT is very useful hardware. You have to locate the MMU table to the specified memory area regardless of the applications which doesn't require the address translation. But if you need only cacheable and non-cacheable regions, you can use the VMT (Virtual MMU Table) for the MMU table. The detailed function is described in the following section.

Table 32.1 IOBC Register Map (Base Address = 0xF0050000)

Name	Address	Type	Reset	Description
-	0x00	R/W	0x00000000	Reserved
USB11H	0x04	R/W	0x00000000	USB 1.1 Host Configuration Register
-	0x08	R/W	0x00000000	Reserved
STORAGE	0x0C	R/W	0x00000002	Storage Device Configuration Register

Table 32.2 SYSC Register Map (Base Address = 0xF3005000)

Name	Address	Type	Reset	Description
REMAP	0x00	R/W	0x00000003	Remap Configuration Register
VCFG0	0x04	R/W	0x10008000	Video Core Configuration Register 0
ECFG0	0x08	R/W	-	Exceptional Configuration Register 0
MPCFG	0x10	R/W	0x00000000	Main Processor Configuration Register
BUSCFG	0x20	R/W	0x00001F1F	Main Bus Configuration Register

Table 32.3 VMT Register Map (Base Address = 0xF7000000)

Name	Address	Type	Reset	Description
REGION0	0x00	R/W	-	Configuration Register for Region 0
REGION1	0x04	R/W	-	Configuration Register for Region 1
REGION2	0x08	R/W	-	Configuration Register for Region 2
REGION3	0x0C	R/W	-	Configuration Register for Region 3
REGION4	0x10	R/W	-	Configuration Register for Region 4
REGION5	0x14	R/W	-	Configuration Register for Region 5
REGION6	0x18	R/W	-	Configuration Register for Region 6
REGION7	0x1C	R/W	-	Configuration Register for Region 7
TABBASE	0x8000	R	-	MMU Table Base Address

32.1 I/O Bus Controller

USB 1.1 Host Configuration Register (USB11H)

0xF0050004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															

CNT OVRC

Field	Name	RW	Reset	Description
1 ~ 0	OVRC	RW	0x0	Reserved
2	CNT	RW	0x0	Reserved
31 ~ 3	-	-	-	Undefined

Storage Controller Configuration Register (STORAGE)

0xF005000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															ECCS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															-

Field	Name	RW	Reset	Description
15 ~ 0	-	-	-	Undefined
ECC monitor bus selection register				
0 : Storage bus				
17 ~ 16	ECCS	RW	0x0	1 : I/O bus 2 : general purpose SRAM 3 : main processor data bus
31 ~ 18	-	-	-	Undefined

32.2 System Controller

Remap Control Register (REMAP)

0xF3005000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JTAG															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFGBW															

Field	Name	RW	Reset	Description
				Remap configuration register
				Refer to 2.1 Address Map on page 2-1 for more information about address remapping.
2 ~ 0	REMAP	RW	0xX	0 : Remap 0x1... to 0x0... 1 : Remap 0x2... to 0x0... 2 : unpredicted 3 : Remap 0x7... to 0x0... 4 : Do-not remap 5 : unpredicted 6 : unpredicted 7 : Remap 0xE... to 0x0...
3	-	-	-	Undefined
4	FB	R		0: Normal Boot 1: Fast Boot
7 ~ 5	-	-	-	Undefined
9 ~ 8	CFGBW	R	0xX	Boot bus width indication bits Refer to memory controller
15 ~ 10	-	-	-	Undefined
19 ~ 16	CFGBM	R	0xX	Boot mode indication bits Refer to boot mode description
23 ~ 20	-	-	-	Undefined
				JTAG mode indication bits
25 ~ 24	JTAG	RW	0xX	0,1 : main processor only 2 : sub processor only 3 : dual processor Refer to 34. JTAG Debug Interface on page 34-1.
31 ~ 26	-	-	-	Undefined

Video Controller Core Configuration Register 0 (VCFG0)

0xF3005004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ROB															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROB															

Field	Name	RW	Reset	Description
13 ~ 0	-	-	-	Undefined
31 ~ 14	ROB	RW	0x4002	Base address for sub processor boot area (0x0...)

Exceptional Configuration Register 0 (ECFG0)

0xF300500C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDW	IW	DW		RTCWAIT										ME1	ME0

Field	Name	RW	Reset	Description
14	SDW	R/W	0x1	Subcore DTCM Wait Enable Bit When the operating frequency of main ARM926EJ-S is faster than 200MHz, it should be set to 1.
13	IW	R/W	0x1	Main Core ITCM Wait Enable Bit When the operating frequency of main ARM926EJ-S is faster than 200MHz, it should be set to 1.
12	DW	R/W	0x1	Main Core DTCM Wait Enable Bit When the operating frequency of main ARM926EJ-S is faster than 200MHz, it should be set to 1.
10 ~ 8	RTCWAIT	RW	0x3	RTC Access Wait Cycle
1	ME1	RW	0x1	Bus Clock Dynamic Mask Enable for I/O Bus to Main Bus 0b : Dynamic Clock Masking Disabled (Bus clock is always enabled) 1b : Dynamic Clock Masking Enabled
0	ME0	RW	0x1	Bus Clock Dynamic Mask Enable for Main Bus to I/O Bus 0b : Dynamic Clock Masking Disabled (Bus clock is always enabled) 1b : Dynamic Clock Masking Enabled

Main Processor Configuration Register (MPCFG)

0xF3005010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INI									0						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Field	Name	RW	Reset	Description
30 ~ 0	-	-	-	Undefined
31	INI	RW	0x0	ITCM boot enable register

Main Bus Configuration Register (MBCFG)

0xF3005020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-					RR			-							INCRB

Field	Name	RW	Reset	Description
4 ~ 0	INCRB	RW	0x1F	Incremental burst lock enable 0 : unlocked transfer 1 : locked transfer Refer to Bus Table
7 ~ 5	-	-	-	Undefined
12 ~ 8	RR	RW	0x1F	Round-robin arbitration enable 0 : fixed priority arbitration 1 : round-robin arbitration * Recommended value is 0x00
31 ~ 13	-	-	-	Undefined

Bus Table

Field	Connected Resources
0	General purpose SRAM, Shared SRAM, DTCM, Boot-ROM
1	External Static memory, SDRAM channel 0 (0x2...)
2	N/A
3	Video SRAM
4	I/O bus, all the peripherals

32.3 Virtual MMU Table

Region Configuration Register (REGIONx)

0xF700000 ~ 0xF700001C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SA														-	SZ
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SZ				AP		EN		DO				-	CA	BU	-

Field	Name	RW	Reset	Description
2	BU	RW	0x0	Bufferable Register * Defined same as ARM926EJS
3	CA	RW	0x0	Cacheable Register * Defined same as ARM926EJS
8 ~ 5	DO	RW	0x0	Domain Register * Defined same as ARM926EJS
9	EN	RW	0x0	Region Enable Register
11 ~ 10	AP	RW	0x0	Access Permission Register * Defined same as ARM926EJS
Size Register				
0x13 : 1MB				
0x14 : 2MB				
...				
0x1E : 2GB				
0x1F : 4GB				
31 ~ 20	SA	RW	0x0	Section Base Address * Defined same as ARM926EJS

* All the bit fields except for SZ and EN are defined the same as "Section Descriptor" in ARM926EJS technical reference manual.

The REGION7 has the highest priority and the REGION0 has the lowest priority.

After setting "Region Configuration Registers", the address of TABBASE should be written into TLB base address for ARM926EJS.

33 PORT MULTIPLEXER & GPIO

33.1 Overview

The TCC79XX has a lot of general purpose I/Os which is able to be programmed by setting internal registers. All I/Os are set to input mode at reset. The block diagram of GPIO is in the following figure.

The GPIO in the TCC79XX has the following key features.

The I/Os have the multiple functions shared by various on-chip hardware controllers.

The GPSB, memory stick host controller, UART, and SD/MMC controller have the capability of interfacing to external device via the multi-channel ports by multiplexing the in/out signals of the corresponding hardware.

Most of the I/Os can be pulled-up or pulled-down by reconfigurable register.

The external interrupts passed to the interrupt controller can be selected from the various sources.

The GPIO controller has the special function described in “33.2.2 GPIO Register” on page 33-20. And the various I/Os are pulled-up or pulled-down at system reset to reduce the boot-up power consumption during the system reset, which prevents the reset state of the various I/Os from being floating state.

But, for this reason, if you want to design the specific I/Os with pull-up state which are pulled-down by reset configuration, you should be careful in choosing the value of resistor and the value of the pull-up/down control registers should be changed as soon as possible – for example, in the reset handler of initial boot code.

33.2 Register Description

Table 33.1 GPIO Register Map (Base Address = 0xF005A000)

Name	Addr	Type	Reset	Description
PORTCFG0	0x000	R/W		Port Configuration Register 0
PORTCFG1	0x004	R/W		Port Configuration Register 1
PORTCFG2	0x008	R/W		Port Configuration Register 2
PORTCFG3	0x00C	R/W		Port Configuration Register 3
PORTCFG4	0x010	R/W		Port Configuration Register 4
PORTCFG5	0x014	R/W		Port Configuration Register 5
PORTCFG6	0x018	R/W	Refer to the register descriptions	Port Configuration Register 6
PORTCFG7	0x01C	R/W		Port Configuration Register 7
PORTCFG8	0x020	R/W		Port Configuration Register 8
PORTCFG9	0x024	R/W		Port Configuration Register 9
PORTCFG10	0x028	R/W		Port Configuration Register 10
PORTCFG11	0x02C	R/W		Port Configuration Register 11
PORTCFG12	0x030	R/W		Port Configuration Register 12
PORTCFG13	0x034	W		Port Configuration Register 13
RESERVED	0x038	-		
RESERVED	0x03C	-		
GPADAT	0x040	R/W	0x00000000	GPA Data Register
GPAEN	0x044	R/W	0x00000000	GPA Output Enable Register
GPASET	0x048	W	-	OR function on GPA Output Data
GPACLR	0x04C	W	-	BIC function on GPA Output Data
GPAXOR	0x050	W	-	XOR function on GPA Output Data
	0x054			
RESERVED	0x058	W	-	-
	0x05C			
GPBDAT	0x060	R/W	0x00000000	GPB Data Register
GPBEN	0x064	R/W	0x00000000	GPB Output Enable Register
GPBSET	0x068	W	-	OR function on GPB Output Data
GPBCLR	0x06C	W	-	BIC function on GPB Output Data
GPBXOR	0x070	W	-	XOR function on GPB Output Data
	0x074			
RESERVED	0x078	-	-	-
	0x07C			
GPCDAT	0x080	R/W	0x00000000	GPC Data Register
GPCEN	0x084	R/W	0x00000000	GPC Output Enable Register
GPCSET	0x088	W	-	OR function on GPC Output Data
GPCCLR	0x08C	W	-	BIC function on GPC Output Data
GPCXOR	0x090	W	-	XOR function on GPC Output Data
	0x094			
RESERVED	0x098	-	-	-
	0x09C			
GPDDAT	0x0A0	R/W	0x00000000	GPD Data Register
GPDEN	0x0A4	R/W	0x00000000	GPD Output Enable Register
GPDSET	0x0A8	W	-	OR function on GPD Output Data
GPDCLR	0x0AC	W	-	BIC function on GPD Output Data
GPDXOR	0x0B0	W	-	XOR function on GPD Output Data
	0x0B4			
RESERVED	0x0B8	-	-	-

Name	Addr	Type	Reset	Description
	0x0BC			
GPEDAT	0x0C0	R/W	0x00000000	GPE Data Register
GPEEN	0x0C4	R/W	0x00000000	GPE Output Enable Register
GPESET	0x0C8	W	-	OR function on GPE Output Data
GPECLR	0x0CC	W	-	BIC function on GPE Output Data
GPEXOR	0x0D0	W	-	XOR function on GPE Output Data
	0x0D4			
RESERVED	0x0D8	-	-	-
	0x0DF			
GPFDAT	0x0E0	R/W	0x00000000	GPF Data Register
GPFEN	0x0E4	R/W	0x00000000	GPF Output Enable Register
GPFSET	0x0E8	W	-	OR function on GPF Output Data
GPFCLR	0x0EC	W	-	BIC function on GPF Output Data
GPFXOR	0x0F0	W	-	XOR function on GPF Output Data
	0x0F4			
RESERVED	0x0F8	-	-	-
	0x0FC			
CPUD0	0x100	R/W		Pull-Up/Down Control Register 0
CPUD1	0x104	R/W		Pull-Up/Down Control Register 1
CPUD2	0x108	R/W		Pull-Up/Down Control Register 2
CPUD3	0x10C	R/W		Pull-Up/Down Control Register 3
CPUD4	0x110	R/W		Pull-Up/Down Control Register 4
CPUD5	0x114	R/W	Refer to the register descriptions	Pull-Up/Down Control Register 5
CPUD6	0x118	R/W		Pull-Up/Down Control Register 6
CPUD7	0x11C	W		Pull-Up/Down Control Register 7
CPUD8	0x120	R/W		Pull-Up/Down Control Register 8
CPUD9	0x124	W		Pull-Up/Down Control Register 9
CPDRV0	0x128	R/W		Driver strength Control Register 0
CPDRV1	0x12C	R/W		Driver strength Control Register 1
AINCFG	0x130	R/W	0x000000FF	Analog Input Pad Control Register
EINTSEL0	0x134	R/W	0x00000000	External Interrupt Select Register 0
EINTSEL1	0x138	R/W	0x00000000	External Interrupt Select Register 1
IRQSEL	0x13C	R/W	0x00000000	Interrupt Select Register

33.2.1 Port Configuration

Port Configuration Register 0 (PORTCFG0) 0xF005A000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPD23				LPD22				LPD21				LPD20			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPD19				LCD18				LCD16				LCD8			

PIN	Field	RESET	0	1	2	3	4	5
LPD[23]	LPD23	0	LPD[23]		GPIOC[23]	SD_CMD(0)	MS_BUS(0)	SFRM(10)
LPD[22]	LPD22	0	LPD[22]		GPIOC[22]	SD_CLK(0)	MS_CLK(0)	SCLK(10)
LPD[21]	LPD21	0	LPD[21]		GPIOC[21]	SD_D0(0)	MS_D0(0)	SDI(10)
LPD[20]	LPD20	0	LPD[20]		GPIOC[20]	SD_D1(0)	MS_D1(0)	SDO(10)
LPD[19]	LPD19	0	LPD[19]		GPIOC[19]	SD_D2(0)	MS_D2(0)	
LPD[18]			LPD[18]		GPIOC[18]	SD_D3(0)	MS_D3(0)	
LPD[17]	LCD18	0	LPD[17]	LXD[17]	GPIOC[17]			
LPD[16]			LPD[16]	LXD[16]	GPIOC[16]			
LPD[15]			LPD[15]	LXD[15]	GPIOC[15]			
LPD[14]			LPD[14]	LXD[14]	GPIOC[14]			
LPD[13]			LPD[13]	LXD[13]	GPIOC[13]			
LPD[12]	LCD16	0	LPD[12]	LXD[12]	GPIOC[12]			
LPD[11]			LPD[11]	LXD[11]	GPIOC[11]			
LPD[10]			LPD[10]	LXD[10]	GPIOC[10]			
LPD[9]			LPD[9]	LXD[9]	GPIOC[9]			
LPD[8]			LPD[8]	LXD[8]	GPIOC[8]			
LPD[7]			LPD[7]	LXD[7]	GPIOC[7]			
LPD[6]			LPD[6]	LXD[6]	GPIOC[6]			
LPD[5]			LPD[5]	LXD[5]	GPIOC[5]			
LPD[4]	LCD8	0	LPD[4]	LXD[4]	GPIOC[4]			
LPD[3]			LPD[3]	LXD[3]	GPIOC[3]			
LPD[2]			LPD[2]	LXD[2]	GPIOC[2]			
LPD[1]			LPD[1]	LXD[1]	GPIOC[1]			
LPD[0]			LPD[0]	LXD[0]	GPIOC[0]			

In case of LPD[17:0], refer to BP of GPADAT on page 33-20.

FUNCTION	DESCRIPTION
LPD[23:0]	Pixel Data for LCD RGB Interface
LXD[17:0]	Pixel Data for LCD CPU Interface
GPIOC[23:0]	GPIO port
SD_CMD(0)	
SD_CLK(0)	
SD_D0(0)	SD Interface Port 0
SD_D1(0)	
SD_D2(0)	
SD_D3(0)	
MS_BUS(0)	Memory Stick Interface Port 0
MS_CLK(0)	
MS_D0(0)	
MS_D1(0)	
MS_D2(0)	

MS_BUS(0)	
SFRM(10)	
SCLK(10)	GPSB Port 10
SDI(10)	
SDO(10)	
GPIOC[23:0]	GPIO port

Port Configuration Register 1 (PORTCFG1)

0xF005A004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LDE				LCK				LHS				LVS			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCS				GPIOC30				GPIOC31				GPIOF27			

PIN	Field	RESET	0	1	2	3	4	5
LBIAS	LDE	2	LBIAS	LWEN	GPIOC[25]			
LCK	LCK	2	LCK	LOEN	GPIOC[26]			
LHS	LHS	2	LHS	LXA	GPIOC[27]			
LVS	LVS	2	LVS	LCSN[0]	GPIOC[28]			
GPIOC[29]	LCS	2	GPIOC[29]	LCSN[1]	GPIOC[29]			
GPIOC[30]	GPIOC30	0	GPIOC[30]	GPIOC[30]	GPIOC[30]			
GPIOC[31]	GPIOC31	0	GPIOC[31]	GPIOC[31]	GPIOC[31]			
GPIOF[27]	GPIOF27	0		GPIOF[27]	SFRM(9)	NANDRDY3		LDE_IN

In case of LBIAS, LCK, LHS, LVS, GPIOC[29], and GPIOC[30], refer to BP of GPADAT on page 33-20.

Function	Port	Description
LCD RGB Interface	LBIAS	Data Enable
	LCK	Pixel Clock
	LHS	Horizontal Sync
	LVS	Vertical Sync
LCD CPU Interface	LWEN	Write Strobe
	LOEN	Read Strobe
	LXA	RS signal
	LCSN[0]	Chip Selector 0
	LCSN[1]	Chip Selector 1
GPIO	GPIOC[31:25], GPIOF[27]	GPIO port
GPSB	SFRM(9)	GPSB Port 9
NFC	NANDRDY3	READY signal for NFC
LCD PIP	LDE_IN	Data Enable Input

Port Configuration Register 2 (PORTCFG2)

0xF005A008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIOF26				GPIOF25				HPCSN_L				HPCTRL			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPXD17				HPXD15				HPXD11				HPXD7			

PIN	Field	RESET	0	1	2	3	4	5
GPIOF[26]	GPIOF26	0	GPIOF[26]	SCLK(9)	NANDRDY2	HDDRDY	LCLK_IN	
GPIOF[25]	GPIOF25	0	GPIOF[25]	SDI(9)	NANDCSN3		LVS_IN	
HPCSN_L	HPCSN_L	1	HPCSN_L	GPIOF[24]	SDO(9)	NANDCSN2	HDDAK	LHS_IN
HPINTO			HPINTO	GPIOF[23]	MS_CLK(1)	SD_CLK(1)	HDDRQ	LPDIN[23]
HPINTO1			HPINTO1	GPIOF[22]	MS_BUS(1)	SD_CMD(1)	HDDXA[2]	LPDIN[22]
HPXA[0]	HPCTRL	1	HPXA[0]	GPIOF[21]	MS_D0(1)	SD_D0(1)	HDDXA[1]	LPDIN[21]
HPWRN			HPWRN	GPIOF[19]	MS_D2(1)	SD_D2(1)	HDDIOW	LPDIN[19]
HPRDN			HPRDN	GPIOF[18]	MS_D3(1)	SD_D3(1)	HDDIOR	LPDIN[18]
HPXD[17]	HPXD17	0	HPXD[17]	GPIOF[17]	SD_CLK(3)	SD_CLK(2)	HDDCSN1	LPDIN[17]
HPXD[16]			HPXD[16]	GPIOF[16]	SD_CMD(3)	SD_CMD(2)	HDDCSN0	LPDIN[16]
HPXD[15]			HPXD[15]	GPIOF[15]	NANDXD[15]	SD_D7(2)	HDDXD[15]	LPDIN[15]
HPXD[14]	HPXD15	0	HPXD[14]	GPIOF[14]	NANDXD[14]	SD_D6(2)	HDDXD[14]	LPDIN[14]
HPXD[13]			HPXD[13]	GPIOF[13]	NANDXD[13]	SD_D5(2)	HDDXD[13]	LPDIN[13]
HPXD[12]			HPXD[12]	GPIOF[12]	NANDXD[12]	SD_D4(2)	HDDXD[12]	LPDIN[12]
HPXD[11]			HPXD[11]	GPIOF[11]	NANDXD[11]	SD_D3(2)	HDDXD[11]	LPDIN[11]
HPXD[10]	HPXD11	0	HPXD[10]	GPIOF[10]	NANDXD[10]	SD_D2(2)	HDDXD[10]	LPDIN[10]
HPXD[9]			HPXD[9]	GPIOF[9]	NANDXD[9]	SD_D1(2)	HDDXD[9]	LPDIN[9]
HPXD[8]			HPXD[8]	GPIOF[8]	NANDXD[8]	SD_D0(2)	HDDXD[8]	LPDIN[8]
HPXD[7]	HPXD7	0	HPXD[7]	GPIOF[7]	SD_D7(3)	SFRM(8)	HDDXD[7]	LPDIN[7]

For HPCSN port, refer to PORTCFG13 register on page 33-19.

Function	Port	Description
EHI	HPCSN_L	Chip Select
	HPINTO	External Interrupt for HPCSN
	HPXA[0]	Address
	HPINTO1	External Interrupt for HPCSN_L
	HPWRN	Write Strobe
	HPRDN	Read Strobe
	HPXD[17:7]	Data
GPIO	GPIOF[26:22], GPIOF[19:7]	GPIO
	SCLK(9)	
	SDI(9)	GPSB Port 9
	SDO(9)	
GPSB	SFRM(8)	GPSB Port 8
	MS_CLK(1)	
	MS_BUS(1)	Memory Stick Port 1
	MS_D0,2,3(1)	
Memory Stick	SD_D7(3)	
	SD_CLK(3)	SD Interface Port3
	SD_CMD(3)	
	SD_CLK(1)	SD Interface Port1
SD Card		

	SD_CMD(1)	
	SD_D0,2,3(1)	
	SD_CLK(2)	
	SD_CMD(2)	SD Interface Port2
	SD_D0~7(2)	
	NANDXD[15:8]	Data
NFC	NANDRDY2	READY
	NANDCSN 3	Chip Select 3
	NANDCSN2	Chip Select 2
	HDDRDY	READY
	HDDAK	Acknowledge
	HDDRQ	Request
IDE	HDDXA[2:1]	Address
	HDDIOW	Write Enable
	HDDIOR	Read Enable
	HDDCSN1	Chip Select 1
	HDDCSN0	Chip Select 0
	HDDXD[15:7]	Data Bus
	LCLK_IN	Pixclk input
LCD PIP	LVS_IN	Vertical Sync Input
	LHS_IN	Horizontal Sync Input
	LPDIN[23:21]	
	LPDIN[19:7]	Pixel Data Input

Port Configuration Register 3 (PORTCFG3)

0xF005A00C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPXD6				HPXD5				HPXD4				HPXD3			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPXD2				HPXD1				HPXD0				SCMD0			

PIN	Field	RESET	0	1	2	3	4	5
HPXD[6]	HPXD6	0	HPXD[6]	GPIOF[6]	SD_D6(3)	SCLK(8)	HDDXD[6]	LPDIN[6]
HPXD[5]	HPXD5	0	HPXD[5]	GPIOF[5]	SD_D5(3)	SDI(8)	HDDXD[5]	LPDIN[5]
HPXD[4]	HPXD4	0	HPXD[4]	GPIOF[4]	SD_D4(3)	SDO(8)	HDDXD[4]	LPDIN[4]
HPXD[3]	HPXD3	0	HPXD[3]	GPIOF[3]	SD_D3(3)	SFRM(7)	HDDXD[3]	LPDIN[3]
HPXD[2]	HPXD2	0	HPXD[2]	GPIOF[2]	SD_D2(3)	SCLK(7)	HDDXD[2]	LPDIN[2]
HPXD[1]	HPXD1	0	HPXD[1]	GPIOF[1]	SD_D1(3)	SDI(7)	HDDXD[1]	LPDIN[1]
HPXD[0]	HPXD0	0	HPXD[0]	GPIOF[0]	SD_D0(3)	SDO(7)	HDDXD[0]	LPDIN[0]
SCMD0	SCMD0	0	SFRM(6)	GPIOD[5]				

	Function	Port	Description
	EHI	HPXD[6:0]	Data Bus
	GPIO	GPIOF[6:0], GPIOD[5]	GPIO
		SCLK(8)	
		SDI(8)	GPSB port 8
		SDO(8)	
		SFRM(7)	
		SCLK(7)	
		SDI (7)	GPSB Port 7
		SDO(7)	
		SFRM(6)	GPSB Port 6
	SD Card	SD_D6~0(3)	SD Interface Port3
	IDE	HDDXD[6:0]	Data Bus
	LCD PIP	LPDIN[6:0]	Data Bus

Port Configuration Register 4 (PORTCFG4)

0xF005A010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCLK0				SDI0				SDO0				GPIOA2			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIOA3				GPIOA4				GPIOA5				CSN_CS0			

PIN	FIELD	RESET	0	1	2	3	4	5
SCLK0	SCLK0	0	SCLK(6)	GPIOD[6]				
SDI0	SDI0	0	SDI(6)	GPIOD[7]				
SDO0	SDO0	0	SDO(6)	GPIOD[8]				
GPIOA[2]	GPIOA2	0	GPIOA[2]	CLK_OUT0				
GPIOA[3]	GPIOA3	0	GPIOA[3]	CLK_OUT1				
GPIOA[4]*	GPIOA4	0	GPIOA[4]	WDTRSTO	TC03			
GPIOA[5]*	GPIOA5	0	GPIOA[5]		TC02			
CSN_CS0	CSN_CS0	0	CSN_CS0	GPIOC[24]				

The port status of GPIOA[4] and GPIOA[5] is only reset by power-on-reset. Therefore, their function, direction, and data are not reset by watchdog.

Function	Port	Description
GPIO	GPIOA[5:2]	
	GPIOD[8:6]	GPIO
	GPIOC[24]	
GPSB	SCLK(6)	
	SDI(6)	GPSB port 6
	SDO(6)	
CKC	CLK_OUT1	PCK_TCT clock output Refer to PCK_TCT on page 23-10
	CLK_OUT0	PCK_TCX clock output Refer to PCK_TCX on page 23-10
	WDTRSTO	POR and Watch Dog Reset Output
EMC	CSN_CS0	External Memory Controller Chip Selector 0
Timer	TC03	
	TC02	Output of Timer/Counter 2 and 3

Port Configuration Register 5 (PORTCFG5)

0xF005A014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSN_NOR				GPIOB0				GPIOB1				GPIOB2			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIOB3				GPIOB4				GPIOB5				GPIOB6			

PIN	FIELD	RESET	0	1	2	3	4	5
CSN_NOR	CSN_NOR	0	CSN_NOR	GPIOF[28]				
GPIOB[0]	GPIOB0	0	GPIOB[0]	NANDXD[0]	SD_D0(5)	SFRM(3)	SRAMIF_XD[0]	
GPIOB[1]	GPIOB1	0	GPIOB[1]	NANDXD[1]	SD_D1(5)	SCLK(3)	SRAMIF_XD[1]	
GPIOB[2]	GPIOB2	0	GPIOB[2]	NANDXD[2]	SD_D2(5)	SDI(3)	SRAMIF_XD[2]	
GPIOB[3]	GPIOB3	0	GPIOB[3]	NANDXD[3]	SD_D3(5)	SDO(3)	SRAMIF_XD[3]	
GPIOB[4]	GPIOB4	0	GPIOB[4]	NANDXD[4]	SD_D4(5)	SFRM(2)	SRAMIF_XD[4]	
GPIOB[5]	GPIOB5	0	GPIOB[5]	NANDXD[5]	SD_D5(5)	SCLK(2)	SRAMIF_XD[5]	
GPIOB[6]	GPIOB6	0	GPIOB[6]	NANDXD[6]	SD_D6(5)	SDI(2)	SRAMIF_XD[6]	

Function	Port	Description
EMC	CSN_NOR	External Memory Controller Chip Selector 3
GPIO	GPIOB[6:0] GPIOF[28]	GPIO
	SFRM(3) SCLK(3) SDI(3)	GPSB port 3
GPSB	SDO(3)	
	SFRM(2) SCLK(2) SDI(2)	GPSB Port 2
SD Card	SD_D0~6(5)	Data Bus
SFRAMIF	SRAMIF_XD[6:0]	Data Bus

Port Configuration Register 6 (PORTCFG6)

0xF005A018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIOB7				GPIOB8				GPIOB9				GPIOB10			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIOB11				GPIOB12				GPIOB13				GPIOB14			

PIN	FIELD	RESET	0	1	2	3	4	5		
GPIOB[7]	GPIOB7	0	GPIOB[7]	NANDXD[7]	SD_D7(5)	SDO(2)	SRAMIF_XD[7]			
GPIOB[8]	GPIOB8	0	GPIOB[8]	NANDOEN	SD_CMD(5)	SRAMIF_OEN				
GPIOB[9]	GPIOB9	0	GPIOB[9]	NANDWEN	SD_CLK(5)	SRAMIF_WEN				
GPIOB[10]	GPIOB10	0	GPIOB[10]	NANDCSN0	SD_D0(6)	MS_D0(3)	SFRM(1)	SRAMIF_CS[0]		
GPIOB[11]	GPIOB11	0	GPIOB[11]	NANDCSN1	SD_D1(6)	MS_D1(3)	SCLK(1)	SRAMIF_CS[1]		
GPIOB[12]	GPIOB12	0	GPIOB[12]	NANDCLE	SD_D2(6)	MS_D2(3)	SDI(1)	SRAMIF_XA[0]		
GPIOB[13]	GPIOB13	0	GPIOB[13]	NANDALE	SD_D3(6)	MS_D3(3)	SDO(1)	SRAMIF_XA[1]		
GPIOB[14]	GPIOB14	0	GPIOB[14]	NANDRDY0	SD_CMD(6)	MS_BUS(3)				

Function		Port		Description	
GPIO		GPIOB[14:7]		GPIO	
		SDO(2)		GPSB port 2	
		SFRM(1)			
		SCLK(1)		GPSB port 1	
		SDI(1)			
		SDO(1)			
		SD_D7(5)			
		SD_CMD(5)		SD Card Interface port 5	
		SD_CLK(5)			
		SD_D0~3(6)		SD Card Interface port 6	
		SD_CMD(6)			
SD Card		MS_D0~3(3)		Memory Stick Interface port 3	
		MS_BUS(3)			
		SRAMIF_XD[7]		Memory Stick Interface port 3	
		SRAMIF_OEN		Data bus	
		SRAMIF_WEN		Read Enable	
		SRAMIF_CS[1:0]		Write Enable	
		SRAMIF_XA[1:0]		Chip Select	
		Address			

Port Configuration Register 7 (PORTCFG7)

0xF005A01C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIOB15				SCMD1				SCLK1				SDI1			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDO1				GPIOA6				GPIOA7				GPIOA8			

PIN	FIELD	RESET	0	1	2	3	4	5
GPIOB[15]	GPIOB15	0	GPIOB[15]	NANDRDY1	SD_CLK(6)	MS_CLK(3)		
SCMD1	SCMD1	0	SFRM(5)	GPIOD[9]				
SCLK1	SCLK1	0	SCLK(5)	GPIOD[10]				
SDI1	SDI1	0	SDI(5)	GPIOD[11]				
SDO1	SDO1	0	SDO(5)	GPIOD[12]				
GPIOA[6]	GPIOA6	0	GPIOA[6]	SPDIFTX	TCO1			
GPIOA[7]	GPIOA7	0	GPIOA[7]	EXTCLKI	TCO0			
GPIOA[8]	GPIOA8	0	GPIOA[8]	SCL1				

Function	Port	Description
GPIO	GPIOA[8:6] GPIOB[15] GPIOD[12:9]	GPIO
GPSB	SFRM(5) SCLK(5) SDI(5) SDO(5)	GPSB port 5
SPDIFTX	SPDIFTX	SPDIF TX signal
Timer	EXTCLKI	External Clock Input
	TCO1, TCO0	Output of Timer/Counter 0 and 1
SD Card	SD_CLK(6)	SD Interface port 6
Memory Stick	MS_CLK(3)	Memory Stick Interface port 3
I2C	SCL1	SCLK for I2C channel 1

Port Configuration Register 8 (PORTCFG8)

0xF005A020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIOA9				GPIOA10				GPIOA11				GPIOA12			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIN0				AIN1				AIN2				AIN3			

PIN	FIELD	RESET	0	1	2	3	4	5
GPIOA[9]	GPIOA9	0	GPIOA[9]	SDA1				
GPIOA[10]	GPIOA10	0	GPIOA[10]	CBCLK				
GPIOA[11]	GPIOA11	0	GPIOA[11]	CLRCK				
GPIOA[12]	GPIOA12	0	GPIOA[12]	CDATA				
AIN[0]	AIN0	0	AIN[0]	GPIOE[24]				
AIN[1]	AIN1	0	AIN[1]	GPIOE[25]				
AIN[2]	AIN2	0	AIN[2]	GPIOE[26]	SD_CMD(7)	MS_BUS(4)		
AIN[3]	AIN3	0	AIN[3]	GPIOE[27]	SD_CLK(7)	MS_CLK(4)		

Function	Port	Description
GPIO	GPIOA[12:9] GPIOE[27:24]	GPIO
ADC	AIN[3:0]	ADC input
CD Interface	CBCLK CLRCK CDATA	CD interface
SD Card	SD_CMD(7) SD_CLK(7)	SD Interface port 7
Memory Stick	MS_BUS(4) MS_CLK(4)	Memory Stick Interface port 4
I2C	SDA1	SDA for I2C channel 1

Port Configuration Register 9 (PORTCFG9)

0xF005A024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AIN4				AIN5				AIN6				AIN7			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UTXD0				URXD0				UCTS0				URTS0			

PIN	FIELD	RESET	0	1	2	3	4	5
AIN[4]	AIN4	0	AIN[4]	GPIOE[28]	SD_D0(7)	MS_D0(4)	SFRM(0)	
AIN[5]	AIN5	0	AIN[5]	GPIOE[29]	SD_D1(7)	MS_D1(4)	SCLK(0)	
AIN[6]	AIN6	0	AIN[6]	GPIOE[30]	SD_D2(7)	MS_D2(4)	SDI(0)	
AIN[7]	AIN7	0	AIN[7]	GPIOE[31]	SD_D3(7)	MS_D3(4)	SDO(0)	
UTXD0	UTXD0	1	UTXD(0)	GPIOE[0]				
URXD0	URXD0	1	URXD(0)	GPIOE[1]				
UCTS0	UCTS0	1	UCTS(0)	GPIOE[2]	UTXD(4)			
URTS0	URTS0	1	URTS(0)	GPIOE[3]	URXD(4)			

Function	Port	Description
GPIO	GPIOE[31:28] GPIOE[3:0]	GPIO
ADC	AIN[7:4]	ADC input
CD Interface	CBCLK CLRCK CDATA	CD interface
SD Card	SD_D0~3(7)	SD Interface port 7
Memory Stick	MS_D0~3 (4)	Memory Stick Interface port 4
GPSB	SFRM(0) SCLK(0) SDI(0) SDO(0)	GPSB port 0
UART	UTXD(0) URTXD(0) UCTS(0) URTS(0)	UART port 0
	UTXD(4) URXD(4)	UART port 4

Port Configuration Register 10 (PORTCFG10)

0xF005A028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				UTXD1			URXD1			UCTS1				URTS1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				UTXD2			URXD2			UTXD3				URXD3	

PIN	FIELD	RESET	0	1	2	3	4	5
UTXD1	UTXD1	1	UTXD(1)	GPIOE[4]				
URXD1	URXD1	1	URXD(1)	GPIOE[5]				
UCTS1	UCTS1	1	UCTS(1)	GPIOE[6]	UTXD(5)	SD_CLK(4)	MS_CLK(2)	
URTS1	URTS1	1	URTS(1)	GPIOE[7]	URXD(5)	SD_CMD(4)	MS_BUS(2)	
UTXD2	UTXD2	1	UTXD(2)	GPIOE[8]	SFRM(4)	SD_D0(4)	MS_D0(2)	
URXD2	URXD2	1	URXD(2)	GPIOE[9]	SCLK(4)	SD_D1(4)	MS_D1(2)	
UTXD3	UTXD3	1	UTXD(3)	GPIOE[10]	SDI(4)	SD_D2(4)	MS_D2(2)	
URXD3	URXD3	1	URXD(3)	GPIOE[11]	SDO(4)	SD_D3(4)	MS_D3(2)	

Function	Port	Description
GPIO	GPIOE[11:4] GPIOE[3:0]	GPIO
SD Card	SD_CLK(4) SD_CMD(4) SD_D0~3(4)	SD Interface port 4
Memory Stick	MS_CLK(2) MS_BUS(2) MS_D0~3 (2)	Memory Stick Interface port 2
GPSB	SFRM(4) SCLK(4) SDI(4) SDO(4)	GPSB port 4
UART	UTXD(1) URTXD(1) UCTS(1) URTS(1)	UART port 1
	UTXD(2) URTXD(2)	UART port 2
	UTXD(3) URTXD(3)	UART port 3
	UTXD(5) URXD(5)	UART port 5

Port Configuration Register 11 (PORTCFG11)

0xF005A02C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BM				BCLK				LRCK				MCLK			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAO				DAI				GPIOA0				CCKI			

PIN	FIELD	RESET	0	1	2	3	4	5
BM[0]		0	BM[0]	GPIOF[29]				
BM[1]	BM	0	BM[1]	GPIOF[30]				
BM[2]		0	BM[2]	GPIOF[31]				
BCLK	BCLK	0	BCLK	GPIOD[0]				
LRCK	LRCK	0	LRCK	GPIOD[1]				
MCLK	MCLK	0	MCLK	GPIOD[2]				
DAO	DAO	1	DAO	GPIOD[3]				
DAI	DAI	0	DAI	GPIOD[4]				
GPIOA[0]	GPIOA0	0	GPIOA[0]	SCL0				
GPIOA[1]		0	GPIOA[1]	SDA0				
CCKI	CCKI	0	CCKI	GPIOE[20]				

Function	Port	Description
GPIO	GPIOF[31:29] GPIOE[20] GPIOD[4:0] GPIOA[1:0]	GPIO
Boot Mode	BM[2:0]	Boot Mode Selector
DAI	BCLK LRCK MCLK DAO DAI	
I2C	SCL0 SDA0	I2C channel 0
CIF	CCKI	Clock Input

Port Configuration Register 12 (PORTCFG12)

0xF005A030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCKO				CVS				CHS				CPD0			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPD1				CPD2				CPD3				CPD4			

PIN	FIELD	RESET	0	1	2	3	4	5
CCKO	CCKO	1	CCKO	GPIOE[23]				
CVS	CVS	0	CVS	GPIOE[21]				
CHS	CHS	0	CHS	GPIOE[22]				
CPD[0]	CPD0	0	CPD[0]	GPIOE[12]				
CPD[1]	CPD1	0	CPD[1]	GPIOE[13]				
CPD[2]	CPD2	0	CPD[2]	GPIOE[14]				
CPD[3]	CPD3	0	CPD[3]	GPIOE[15]				
CPD[4]	CPD4	0	CPD[4]	GPIOE[16]				

Function	Port	Description
GPIO	GPIOE[16:12] GPIOE[23:21]	GPIO
	CCKO	Clock Output
CIF	CVS	Vertical Sync
	CHS	Horizontal Sync
	CPD[4:0]	Data Bus

Port Configuration Register 13 (PORTCFG13)

0xF005A034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CPD5				CPD6				CPD7							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

HPCSN*

PIN	FIELD	RESET	0	1	2	3	4	5
CPD[5]	CPD5	0	CPD[5]	GPIOE[17]				
CPD[6]	CPD6	0	CPD[6]	GPIOE[18]				
CPD[7]	CPD7	0	CPD[7]	GPIOE[19]				
HPCSN	HPCSN*	1	HPCSN	GPIOF[20]	MS_D1(1)	SD_D1(1)	HDDXA[0]	LPDIN20]

*) Note that HPCSN field is write-only register and always read as 0.

Function	Port	Description
GPIO	GPIOE[20:17]	GPIO
EHI	HPCSN	Chip Select
Memory Stick	MS_D1(1)	Memory Stick Interface port 1
IDE	HDDXA[0]	Address bit 0
LCD PIP	LPDIN[20]	Data
CIF	CCKO	Clock Output
	CVS	Vertical Sync
	CHS	Horizontal Sync
	CPD[4:0]	Data Bus

33.2.2 GPIO Register

GPIOA Data Register (GPADAT)																0xF005A040
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
0	RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED	GPADAT[12:0]															
BP																

If writing to this register, the written data would be stored in the output buffer. And the direction of corresponding GPIO is the output, the stored data would be out to pin. In case of reading this register, if the direction of the corresponding GPIO is out, the data read is output buffer, otherwise (input mode) the data read is the status of the corresponding pin

BP bit changes configuration of LPD[17:0], LBIAS, LCK, LHS, LVS, GPIOC[29], and GPIOC[30]. When BP bit is set to 1, LPD[17:0], LBIAS, LCK, LHS, LVS, and, GPIOC[29] is set to output direction and their output data are input data of HPXD[17:0], HPWRN, HPRDN, HPXA[0], HPCSN, HPCSN_L, and GPIOF[26] respectively (Figure 33.1).

Additionally, GPBDAT[16] is used for preventing HPCSN and HPCSN_L from propagating to LVS and GPIOC[29](Figure 33.1(b)). Oppositely, if you want to prevent HPCSN and HPCSN_L from propagating to the corresponding EHI module, HPCSN and HPCSN_L pin should be configured as GPIOF[20] and GPIOF[24] respectively.

The direction of GPIOC[30] is determined by PORTCFG1.GPIOC30, but its output data becomes input of GPIOF[26] (Figure 33.1(c)).

To read GPADAT.BP, GPAEN.BPEN should be set to 1.

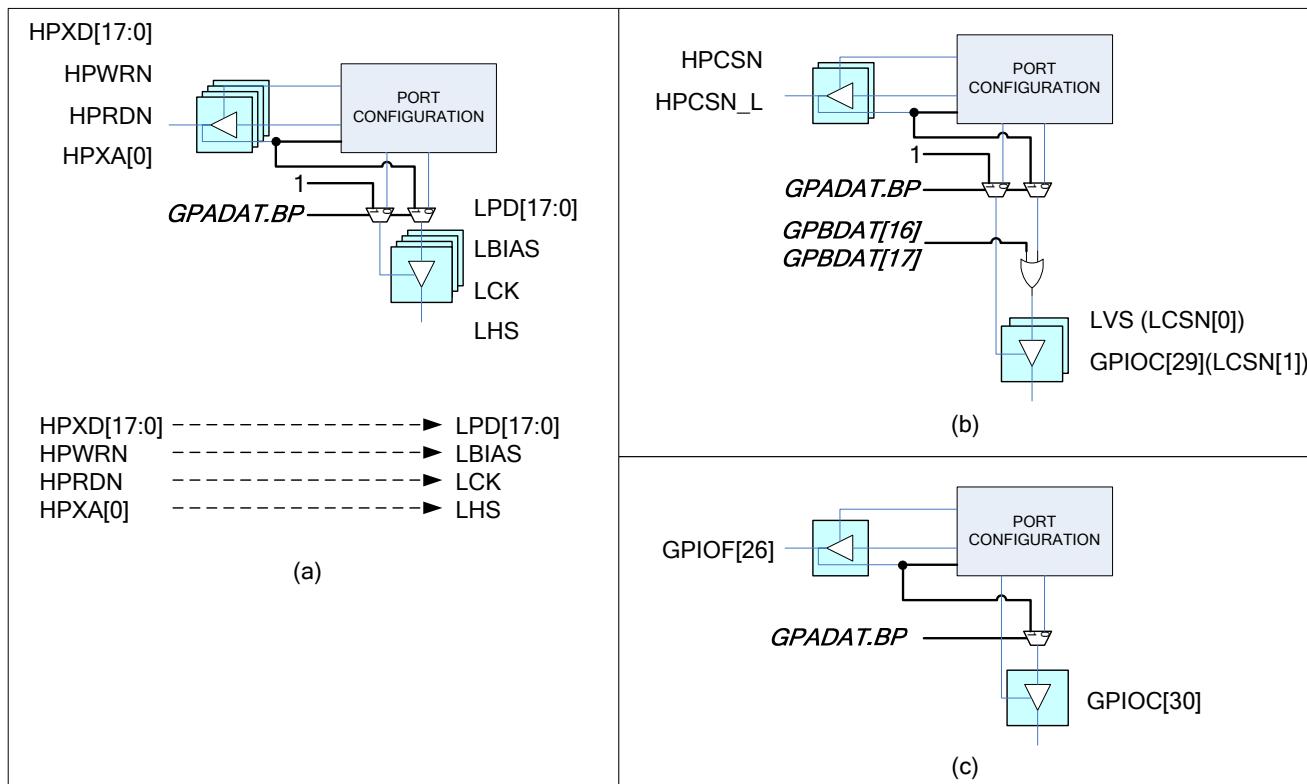


Figure 33.1 I/O Function according to BP bit

GPIOA Direction Control Register (GPAEN)

0xF005A044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BPEN		GPACON[12:0]													

If the GPACON[i] is '1'(i = 0 ~ 12), the direction of the corresponding GPIOA port is changed to output mode, otherwise to input mode.

To read GPADAT.BP, '1' should be written to GPAEN.BPEN.

GPIOA Set Register (GPASET)

0xF005A048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		WDATA[12:0]													

The equivalent function is GPADAT = GPADAT | WDATA, when GPADAT is the value of output buffer.

GPIOA Clear Register (GPACLR)

0xF005A04C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		WDATA[12:0]													

The equivalent function is GPADAT = GPADAT & ~WDATA, when GPADAT is the value of output buffer.

GPIOA XOR Register (GPAXOR)

0xF005A050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		WDATA[12:0]													

The equivalent function is GPADAT = GPADAT ^ WDATA, when GPADAT is the value of output buffer.

GPIOB Data Register (GPBDAT)

0xF005A060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED														MSK1	MSK0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPBDAT[15:0]															

If writing is done to this register, the written data would be stored in the output buffer. And the direction of corresponding GPIO is the output, the stored data would be out to pin. In case of reading this register, if the direction of the corresponding GPIO is out, the data read is output buffer, otherwise (input mode) the data read is the status of the corresponding pin.

MSK0 and MSK1 are used for masking outputs of LVS (LCSN0) and GPIOC[29] (LCSN1). Refer to Figure 33.1(b) on page 33-20. To read MSK0 and MSK1, GPBDAT.MSK0EN and GPBDAT.MSK1EN should be set to 1.

GPIOB Direction Control Register (GPBEN)

0xF005A064

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED														MSK1E	MSK0E
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPBEN[15:0]															

If the GPBEN[i] is '1' (i = 0 ~ 15), the direction of the corresponding GPIOB port is changed to output mode, otherwise to input mode.

GPIOB Set Register (GPBSET)

0xF005A068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[15:0]															

The equivalent function is GPBDAT = GPBDAT | WDATA, when GPBDAT is the value of output buffer.

GPIOB Clear Register (GPBCLR)

0xF005A06C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[15:0]															

The equivalent function is GPBDAT = GPBDAT & ~WDATA, when GPBDAT is the value of output buffer.

GPIOB XOR Register (GPBXOR)

0xF005A070

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[15:0]															

The equivalent function is GPBDAT = GPBDAT ^ WDATA, when GPBDAT is the value of output buffer.

GPIOC Data Register (GPCDAT)

0xF005A080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPCDAT[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCDAT[15:0]															

If writing is done to this register, the written data would be stored in the output buffer. And the direction of corresponding GPIO is the output, the stored data would be out to pin. In case of reading this register, if the direction of the corresponding GPIO is out, the data read is output buffer, otherwise (input mode) the data read is the status of the corresponding pin.

GPIOC Direction Control Register (GPCEN)

0xF005A084

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPCCON[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCCON[15:0]															

If the GPCCON[i] is '1' (i = 0 ~ 31), the direction of the corresponding GPIO port is changed to output mode, otherwise to input mode.

GPIOC Set Register (GPCSET)

0xF005A088

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[15:0]															

The equivalent function is GPCDAT = GPCDAT | WDATA, when GPCDAT is the value of output buffer.

GPIOC Clear Register (GPCCLR)

0xF005A08C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[15:0]															

The equivalent function is GPCDAT = GPCDAT & ~WDATA, when GPCDAT is the value of output buffer.

GPIOC XOR Register (GPCXOR)

0xF005A090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[15:0]															

The equivalent function is GPCDAT = GPCDAT ^ WDATA, when GPCDAT is the value of output buffer.

GPIOD Data Register (GPDDAT)

0xF005A0A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPDDAT[12:0]															

If writing is done to this register, the written data would be stored in the output buffer. And the direction of corresponding GPIO is the output, the stored data would be out to pin. In case of reading this register, if the direction of the corresponding GPIO is out, the data read is output buffer, otherwise (input mode) the data read is the status of the corresponding pin.

GPIOD Direction Control Register (GPDEN)

0xF005A0A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPDCON[15:0]															

If the GPDCON[i] is '1' (i = 0 ~ 12), the direction of the corresponding GPIO port is changed to output mode, otherwise to input mode.

GPIOD Set Register (GPDSET)

0xF005A0A8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[12:0]															

The equivalent function is GPDDAT = GPDDAT | WDATA, when GPDDAT is the value of output buffer.

GPIOD Clear Register (GPDCLR)

0xF005A0AC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[12:0]															

The equivalent function is GPDDAT = GPDDAT & ~WDATA, when GPDDAT is the value of output buffer.

GPIOD XOR Register (GPDXOR)

0xF005A0B0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[12:0]															

The equivalent function is GPDDAT = GPDDAT ^ WDATA, when GPDDAT is the value of output buffer.

GPIOE Data Register (GPEDAT)

0xF005A0C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPEDAT[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPEDAT[15:0]															

If writing is done to this register, the written data would be stored in the output buffer. And the direction of corresponding GPIO is the output, the stored data would be out to pin. In case of reading this register, if the direction of the corresponding GPIO is out, the data read is output buffer, otherwise (input mode) the data read is the status of the corresponding pin.

GPIOE Direction Control Register (GPEEN)

0xF005A0C4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPECON[31:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPECON[15:0]															

If the GPECON[i] is '1' (i = 0 ~ 31), the direction of the corresponding GPIOE port is changed to output mode, otherwise to input mode.

GPIOE Set Register (GPESET)

0xF005A0C8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[15:0]															

The equivalent function is GPEDAT = GPEDAT | WDATA, when GPEDAT is the value of output buffer.

GPIOE Clear Register (GPECLR)

0xF005A0CC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[15:0]															

The equivalent function is GPEDAT = GPEDAT & ~WDATA, when GPEDAT is the value of output buffer.

GPIOE XOR Register (GPDEOR)

0xF005A0D0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDATA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[15:0]															

The equivalent function is GPEDAT = GPEDAT ^ WDATA, when GPEDAT is the value of output buffer.

GPIOF Data Register (GPFDAT)															0xF005A0E0				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	GPFDAT[28:16]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	GPFDAT[15:0]			

If writing is done to this register, the writing data would be stored in the output buffer. And the direction of corresponding GPIO is the output, the stored data would be out to pin. In case of reading this register, if the direction of the corresponding GPIO is out, the data read is output buffer, otherwise (input mode) the data read is the status of the corresponding pin.

GPIOF[31:29] can be only used as input mode.

GPIOF Direction Control Register (GPFCEN)															0xF005A0E4				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	GPFCEN[28:16]			
0	0	0														GPFCEN[15:0]			

If the GPFCEN[i] is '1' (i = 0 ~ 28), the direction of the corresponding GPIO port is changed to output mode, otherwise to input mode.

GPIOF Set Register (GPFSET)															0xF005A0E8				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	N/A			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	WDATA[28:16]			

The equivalent function is GPFDAT = GPFDAT | WDATA, when GPFDAT is the value of output buffer.

GPIOF Clear Register (GPFCCLR)															0xF005A0EC				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	N/A			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	WDATA[28:16]			

The equivalent function is GPFDAT = GPFDAT & ~ WDATA, when GPFDAT is the value of output buffer.

GPIOF XOR Register (GPDFOR)															0xF005A0F0				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	N/A			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	WDATA[15:0]			

The equivalent function is GPFDAT = GPFDAT ^ WDATA, when GPFDAT is the value of output buffer.

33.2.3 Pull up/down control registers

PULL UP/DOWN CONTROL REGISTER 0(CPUD0)

0xF005A100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPD[23]		LPD[22]		LPD[21]		LPD[20]		LPD[19]		LPD[18]		LPD[17]		LPD[16]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPD[15]		LPD[14]		LPD[13]		LPD[12]		LPD[11]		LPD[10]		LPD[9]		LPD[8]	

Each field of pull up/down control registers is the same with corresponding port name and consists of two bits.

Control Bit	Pull-Up/Down Functions
00	Neither pull-up nor pull-down is enabled.
01	Pull-up enabled
10	Pull-down enabled
11	Both pull-up and pull-down are enabled – This value should be not written.

PULL UP/DOWN CONTROL REGISTER 1(CPUD1)

0xF005A104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		GPIOC[31]		GPIOC[30]		GPIOC[29]		LVS		LHS		LCK		LBIAS	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPD[0]		LPD[1]		LPD[2]		LPD[3]		LPD[4]		LPD[5]		LPD[6]		LPD[7]	

PULL UP/DOWN CONTROL REGISTER 2(CPUD2)

0xF005A108

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PULL UP/DOWN CONTROL REGISTER 3(CPUD3)

0xF005A10C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				SCMD0				SCLK0				SDI0			

PULL UP/DOWN CONTROL REGISTER 4(CPUD4)

0xF005A110

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIOA[3]		GPIOA[4]		GPIOA[5]		CSN CS0		CSN NOR		GPIOB[0]		GPIOB[1]		GPIOB[2]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIOB[3]		GPIOB[4]		GPIOB[5]		GPIOB[6]		GPIOB[7]		GPIOB[8]		GPIOB[9]		GPIOB[10]	

PULL UP/DOWN CONTROL REGISTER 5(CPUD5)

0xF005A114

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIOB[11]		GPIOB[12]		GPIOB[13]		GPIOB[14]		GPIOB[15]		SCMD1		SCLK1		SDI1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDO1		GPIOA[6]		GPIOA[7]		GPIOA[8]		GPIOA[9]		GPIOA[10]		GPIOA[11]		GPIOA[12]	

PULL UP/DOWN CONTROL REGISTER 6(CPUD6)

0xF005A118

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AIN[0]		AIN[1]		AIN[2]		AIN[3]		AIN[4]		AIN[5]		AIN[6]		AIN[7]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UTXD0		URXD0		UCTS0		URTS0		UTXD1		URXD1		UCTS1		URTS1	

PULL UP/DOWN CONTROL REGISTER 7(CPUD7*)

0xF005A11C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UTXD2		URXD2		UTXD3		URXD3								BCLK	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LRCK		MCLK		DAO		DAI		GPIOA[0]		GPIOA[1]		CCKI		CCKO	

*) Note that CPUD7 is write-only register.

PULL UP/DOWN CONTROL REGISTER 8(CPUD8)

0xF005A120

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CVS		CHS		CPD0		CPD1		CPD2		CPD3		CPD4		CPD5	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPD6		CPD7													

PULL UP/DOWN CONTROL REGISTER 9(CPUD9)

0xF005A124

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDCLK		SDCKE		OEN		CSN SD		WEN							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

*) Note that CPUD9 is write-only register.

33.2.4 Driver Strength Control Registers

The TCC79XX has the function of programmable port driver strength. The control value for driver strength of the corresponding port can become 0~3 and 3 is the strongest value.

Driver Strength CONTROL REGISTER 0(CDRV0)

0xF005A128

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIOF		GPIOA		LPD		LCD		ADC2		ADC0		CDIF		DAIF	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SD5CTRL		SD6		GPSB5		GPSB3		GPSB2		GPSB0		HPXD		HPCTRL	

Bit	Name	R/W	Reset	Controlled Ports
31-30	GPIOF	RW	2	GPIOF[27:25]
29-28	GPIOA	RW	2	GPIOA[7:0]
27-26	LPD	RW	2	LPD[23:16]
25-24	LCD	RW	2	LPD[15:0], LBIAS, LCD, LHS, LVS, GPIOC[31:29]
23-22	ADC2	RW	2	AIN[7:2]
21-20	ADC0	RW	2	AIN[1:0]
19-18	CDIF	RW	2	GPIOA[12:10]
17-16	DAIF	RW	2	BCLK, LRCK, MCLK, DAO, DAI
15-14	SD5CTRL	RW	2	GPIOB[9:8]
13-12	SD6	RW	2	GPIOB[15:10]
11-10	GPSB5	RW	2	SCMD1, SCLK1, SDI1, SDO1
9-8	GPSB3	RW	2	GPIOB[3:0]
7-6	GPSB2	RW	2	GPIOB[7:4]
5-4	GPSB0	RW	2	SCMD0, SCLK0, SDI0, SDO0
3-2	HPXD	RW	2	HPXD[17:0]
1-0	HPCTRL	RW	2	HPCSN_L, HPINTO, HPXA[1:0], HPCSN, HPWRN, HPRDN

Driver Strength CONTROL REGISTER 1(CDRV1)

0xF005A12C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								CIF		UTD3		UTD2		UTC1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UTD1	UTC0	UTD0	I2C1	I2C0	XD	XA	MCCTRL								

Bit	Name	R/W	Reset	Controlled Ports
31-24	-	R	0	Reserved
23-22	CIF	RW	2	CCKI, CCKO, CVS, CHS, CPD[7:0]
21-20	UTD3	RW	2	UTXD3, URXD3
19-18	UTD2	RW	2	UTXD2, URXD2
17-16	UTC1	RW	2	UCTS1, URTS1
15-14	UTD1	RW	2	UTXD1, URXD1
13-12	UTC0	RW	2	UCTS0, URTS0
11-10	UTD0	RW	2	UTXD0, URXD0
9-8	I2C1	RW	2	GPIOA[9:8]
7-6	I2C0	RW	2	GPIOA[1:0]
5-4	XD*	RW	1	XD[31:0]
3-2	XA*	RW	1	XA[23:0]
1-0	MCCTRL*	RW	1	SDCLK, SDCKE, CSN_CS0, CSN_NOR, CSN_SD, OEN, WEN

* If "base chip revision code" of the TCC79XX is "A", its reset value is 3.

AIN Input Configuration (AINCFG)

0xF005A130

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This register is used for enabling digital input of the corresponding AINn port.

33.2.5 Interrupt Related Registers

EINTSEL0

0xF005A134

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EINT3SEL															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EINT1SEL															

EINTSEL1

0xF005A138

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EINT7SEL															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EINT5SEL															

The interrupt controller can receive up to 8 external interrupts simultaneously, which are from EI0 to EI7. And each external interrupt can become one of the following 32 interrupt sources. EINT0SEL and EINT1SEL register are used for selecting an interrupt source.

NUM	EINTn(n=0,even)	EINTn(n=odd)
0	URXD3	URXD3
1	UTXD2	UTXD2
2	AIN[7]	AIN[7]
3	AIN[1]	AIN[1]
4	AIN[0]	AIN[0]
5	GPIOA[11]	GPIOA[11]
6	GPIOA[7]	GPIOA[7]
7	GPIOA[6]	GPIOA[6]
8	SDO1	SDO1
9	SCMD1	SCMD1
10	GPIOB[15]	GPIOB[15]
11	GPIOB[14]	GPIOB[14]
12	GPIOB[13]	GPIOB[13]
13	GPIOB[4]	GPIOB[4]
14	GPIOB[3]	GPIOB[3]
15	GPIOB[0]	GPIOB[0]
16	GPIOA[5]	GPIOA[5]
17	GPIOA[3]	GPIOA[3]
18	GPIOA[2]	GPIOA[2]
19	SDO0	SDO0
20	SCMD0	SCMD0
21	HPXD[3]	HPXD[3]
22	HPXD[11]	HPXD[11]
23	HPRDN	HPRDN
24	HPCSN_L	HPCSN_L
25	GPIOF[26]	GPIOF[26]
26	GPIOF[27]	GPIOF[27]
27	GPIOC[30]	GPIOC[30]
28	LPD[18]	LPD[18]
29	LPD[23]	LPD[23]
30	PMWKUP	PMWKUP
31	USB_VBOFF	USB_VBON

EINT0SEL selects EI0 interrupt source. For example, when EINT0SEL is set to 29,

LPD[23] port is used for the external interrupt EINT0. When EINT7SEL is set to 0, URXD3 port is used for the external interrupt EI7. Refer to 24.2 Priority Interrupt Controller on page 24-3.

IRQSEL															0xF005A13C
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			0			IRQSEL4		IRQSEL3		IRQSEL2		IRQSEL1		IRQSEL0	

An interrupt source of SEL0, SEL1, SEL2, SEL3, and SEL4 can be programmed. These are determined by IRQSEL register. SEL0, SEL1, SEL3, SEL4 can be the external interrupt 4, 5, 6, and 7 respectively.

This register selects interrupt request sources. For example, when IRQSEL0 is equal to 2, SEL0 bit of IEN register is allocated for the external interrupt 4

IRQSEL0 is used for selecting an interrupt source for SEL0 bit of IEN register Refer to 24.2 Priority Interrupt Controller on page 24-3.

IRQSEL0[1:0]	RESET	NOTE
0		VIDEO CORE interrupt
1		EHI channel 0 control signal interrupt
2	0	External interrupt 4
3		-

IRQSEL1 is used for selecting an interrupt source for SEL1 bit of IEN register

IRQSEL1[3:2]	RESET	NOTE
0		GPSB interrupt
1		EHI channel 1 control signal interrupt
2	0	External interrupt 5
3		-

IRQSEL2 is used for selecting an interrupt source for SEL2 bit of IEN register

IRQSEL2[5:4]	RESET	NOTE
0		UART interrupt
1		EHI channel 1 internal interrupt
2	0	-
3		-

IRQSEL3 is used for selecting an interrupt source for SEL3 bit of IEN register

IRQSEL3[7:6]	RESET	NOTE
0		Memory Stick interrupt
1		External Interrupt 6
2	0	-
3		-

IRQSEL3 is used for selecting an interrupt source for SEL4 bit of IEN register

IRQSEL4[9:8]	RESET	NOTE
0		GPSB interrupt
1		ECC interrupt
2	0	External Interrupt 7
3		USB 2.0 DMA interrupt

34 JTAG DEBUG INTERFACE

34.1 Internal JTAG Connections

The TCC79XX has two cores, ARM926EJ-S core as main controller and video controller for video processing. The JTAG interface circuits on the TCC79XX can be configured by external pull-up or pull-down registers. The SDO0 and SDO1 are used for configuring the JTAG.

If the SDO1 is '0', the external JTAG can control the ARM926EJ-S core as the main processor. If the SDO1 is '1', the JTAG can control the video controller or multi core for main processor and video controller connected with the chained structure.

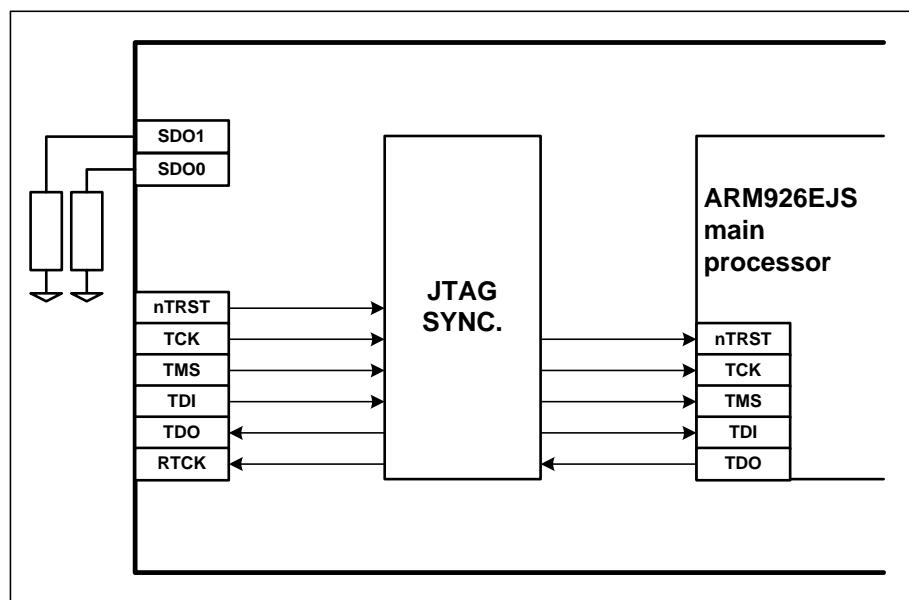


Figure 34.1 Internal JTAG Connection When SDO1 and SDO0 are '0'

The Figure 34.1 shows the JTAG connection in the TCC79XX in case that the SDO1 and SDO0 are both pulled-down. In this mode, the external JTAG can control the ARM926EJ-S main processor only. If you want to run the video controller core, it is the only way that you should download the program for video controller core and debug from the result of corresponding operation.

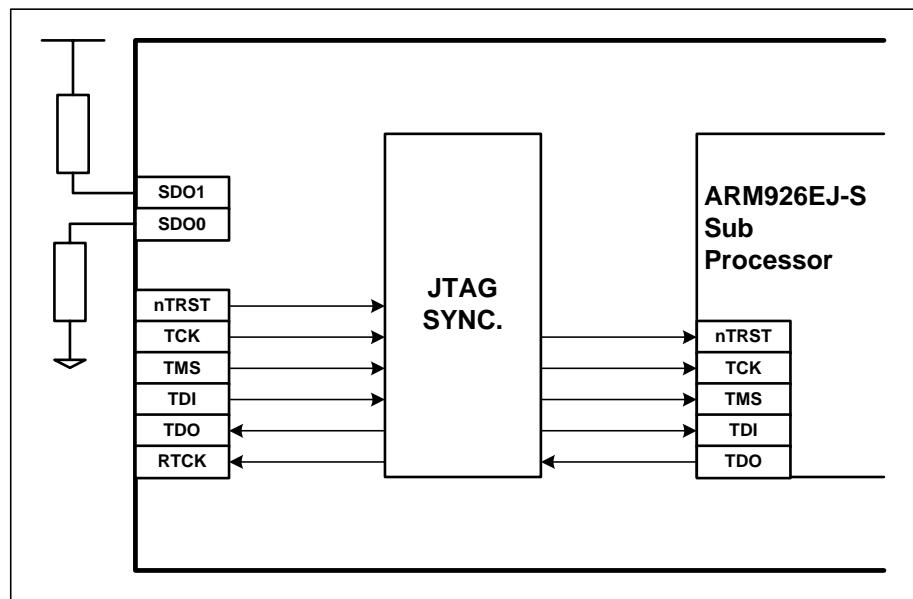


Figure 34.2 Internal JTAG Connection When SDO1 = 0 and SDO0 = 1

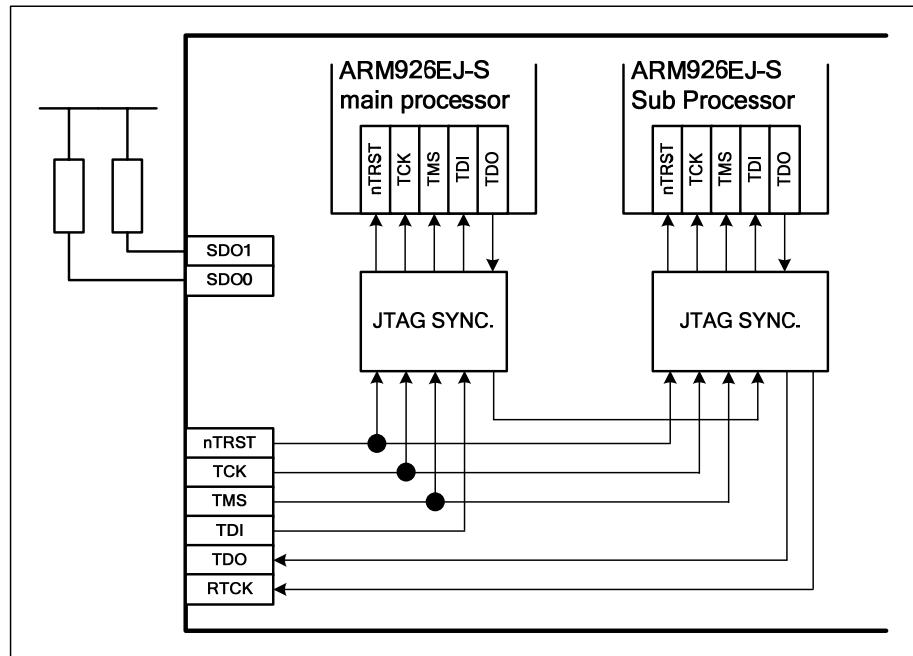


Figure 34.3 Internal JTAG Connection When SDO1 and SDO0 are '1'

The Figure 34.2 shows the JTAG connection in the TCC79XX in case that the SDO1 is pulled-up and the SDO0 is pulled-down. The Figure 34.3 shows the JTAG connection in the TCC79XX in case that the SDO1 is pulled-up and SDO0 is pulled-up. Specially, when SDO1 and SDO0 are "1", the external JTAG can control both the main ARM926EJ-S and the sub ARM926EJ-S, but, some of the debugging tools can't be used for this configuration.

34.2 External JTAG Connections

The TCC79XX has the ARM926EJ-S core as main controller, and JTAG interface for developing the application programs. It can be connected with OPENice32 of AIJI System or Multi-ICE of ARM or other third party's in-circuit emulator supporting ARM926EJ-S core.

With the use of in-circuit emulator, users can easily develop the program for their own system. It provides hardware breakpoints, internal register monitoring, memory dump, etc. Refer to user's manual of in-circuit emulator for more detail functions of it.

The Figure 34.4 shows the application circuit for JTAG interface. Care must be taken not to combine system reset with JTAG reset signal. All the input pins on TCC79XX such as TCK, TMS, TDI, nTRST have the internal pull-up resistors to prevent the corresponding pins from being floating status. If you want to design the application set board without the 4.7k resistors, the interface circuits between JTAG connector and the TCC79XX should be designed to the following guidance. The interface circuit between TCC79XX and JTAG connector works as level-shifter between two different power. If you can design with the single power, this circuit needs not to work as level-shifter. The output drivers to the TCC79XX in the interface circuit have the operating power as VDDIO and the output drivers to the JTAG connector have the operating power as VDD_JTAG. The grounds for each power can be shorted.

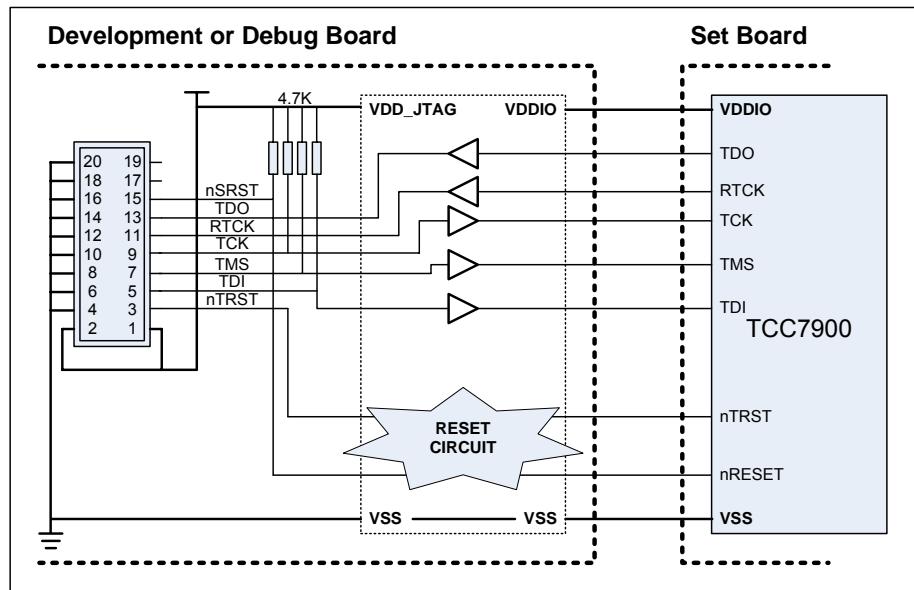


Figure 34.4 JTAG Interface Circuit Diagram

34.3 Debugging with OPENice32 & AIJI Spider

OPENice32 is a powerful and convenient emulator for ARM processor. It provides the best debugging solution for Telechips ARM processor, TCC79XX series. A powerful debugger, AIJI Spider, is supplied with OPENice32.

It also provides a TCC79XX device file and flash device file.

For more information, refer to OPENice32 manual or application note.

34.3.1 System Configuration

OPENice32 is connected to the host PC with serial, USB or Ethernet that provides high speed downloading. It is connected to the target JTAG connector with a 20-way cable.

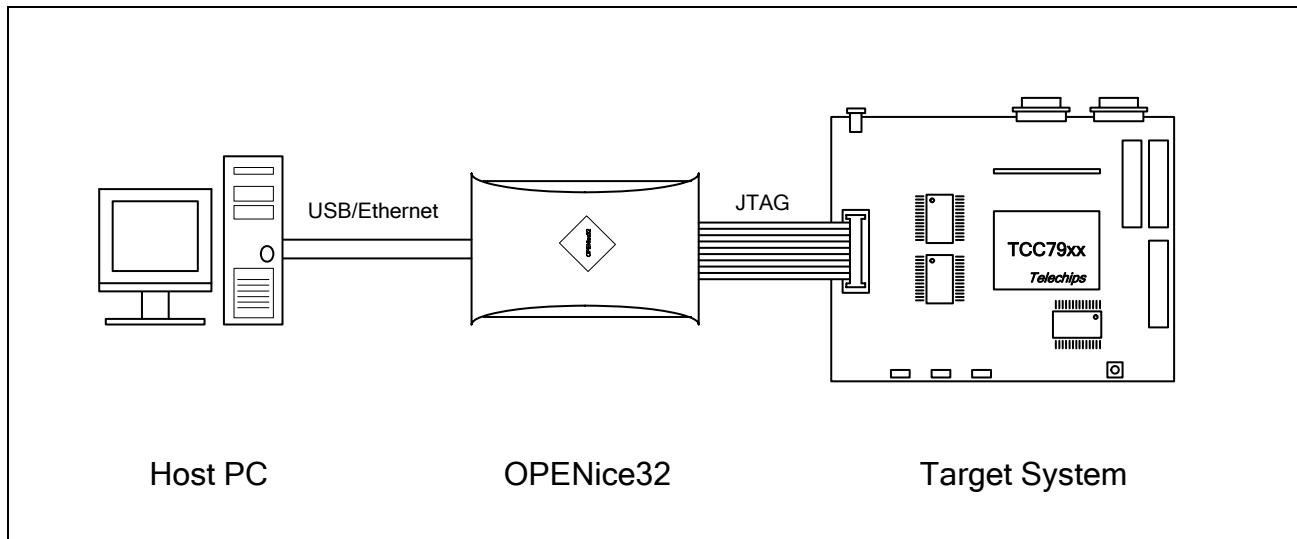


Figure 34.5 Connection between Host PC and OPENice32 and Target System

34.3.2 JTAG Connector on the Target

The following diagram shows the JTAG connector on a target system and a 20-14pin-adapter board.

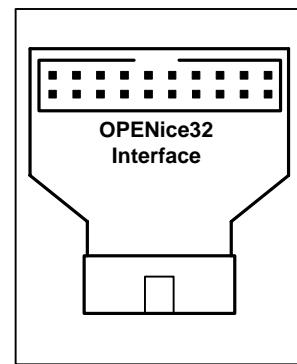
When designing a target board, users can select a 14 pin header or 20pin header. The pin configuration of header on the target board should be same as following diagram.

T_VCC	1 ■	■ 2	x	T_VCC	1 ■	■ 2	GND
nTRST	3 ■	■ 4	GND	nTRST	3 ■	■ 4	GND
TDI	5 ■	■ 6	GND	TDI	5 ■	■ 6	GND
TMS	7 ■	■ 8	GND	TMS	7 ■	■ 8	GND
TCK	9 ■	■ 10	GND	TCK	9 ■	■ 10	GND
x	11 ■	■ 12	GND	TDO	11 ■	■ 12	nSYSRST
TDO	13 ■	■ 14	GND	TDO	13 ■	■ 14	GND
nSYSRST	15 ■	■ 16	GND				
x	17 ■	■ 18	GND				
x	19 ■	■ 20	GND				

[20 pin Connector]

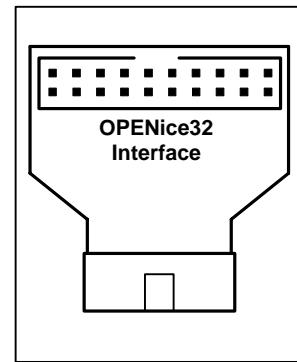
[14 pin Connector]

Figure 34.6 Pin Configuration of 20 and 14 pin Connector



[20-14 Adapter]

Figure 34.7 20-14 Adapter Board Connector



[20-14 Adapter]

Figure 34.8 20-14 Adapter Board

34.3.3 Contact Point

If you have any question regarding OPENice32, please contact AIJI System.

AIJI System Co., Ltd.
Tel: +82-32-223-6611
<http://www.aijisystem.com>

email:stroh@aijisystem.com
<http://www.mculand.com>

35 BOOT PROCEDURE

35.1 Power Up/Down Sequence for Core and I/O Power

The recommended power-up sequence for core and I/O power is described in the following figure.

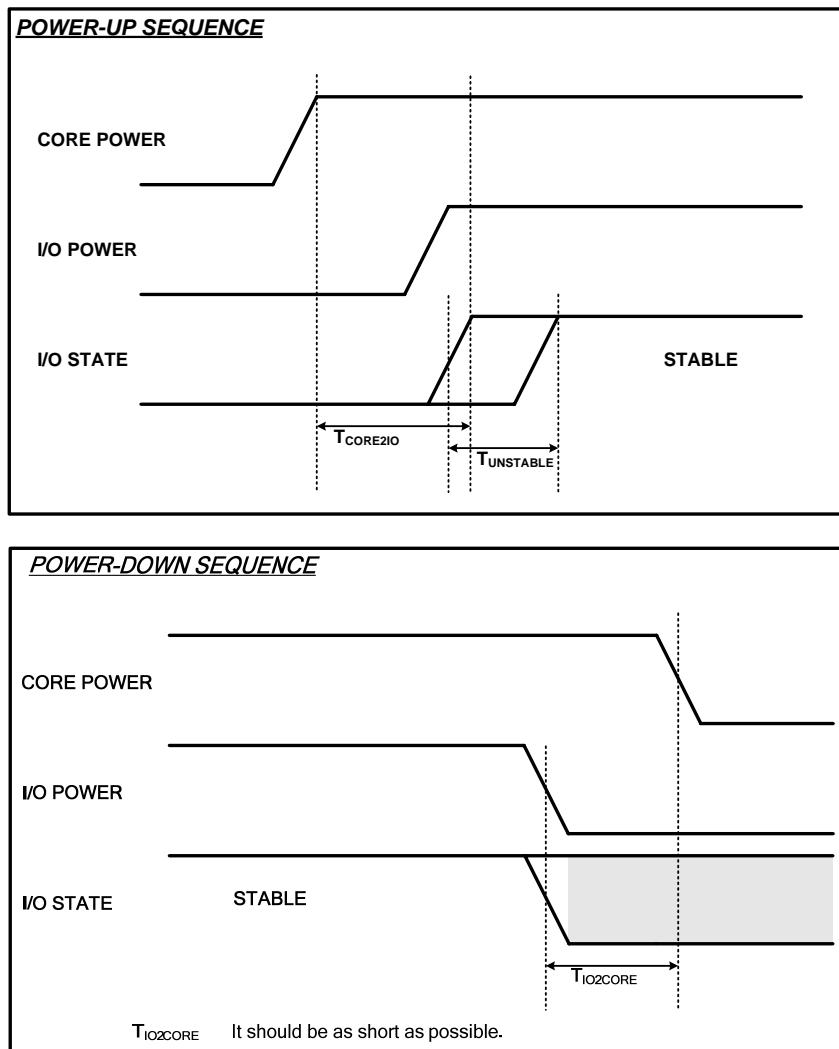


Figure 35.1 Power Up/Down Sequence for Core and I/O Power

The above timing parameter is as follows.

Name	Description	Min	Typ	Max
$T_{CORE2IO}$	It should be as short as possible, but the value should be greater than or equal to 0ns.	0ns		
$T_{UNSTABLE}$	It is dependent on the application circuit. It is caused by the meta-stability for the CMOS I/O and GPIO connections. For the sufficient power supply, it can be shorter.	8ns		

* The minimum value for the $T_{UNSTABLE}$ is the simulation result in case that the power supply is ideal at 1.65V I/O voltage.

35.2 Boot Mode

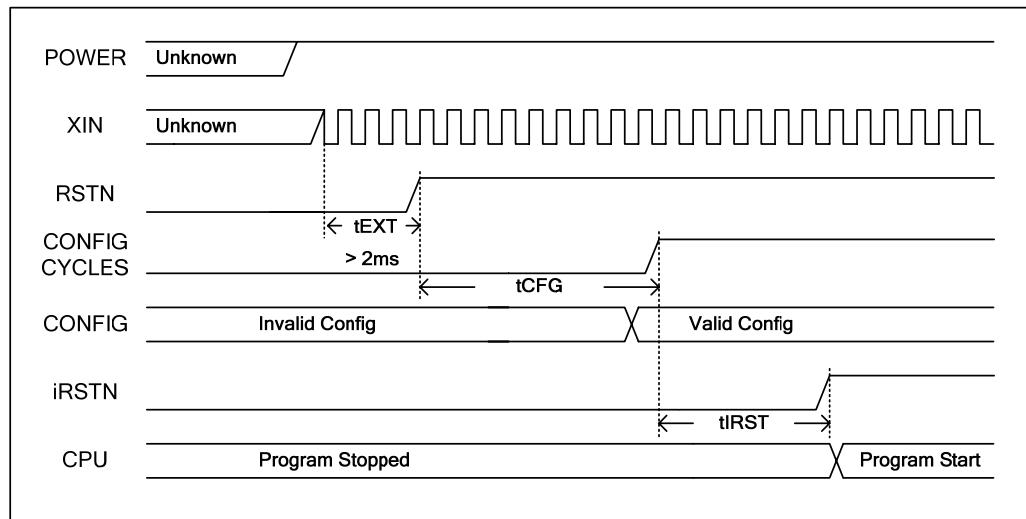
The TCC79XX has the 5 pins for booting configuration with BM[2:0], UTXD0, UTXD1.

Table 35.1: Configuration Value

BM[2:0]	DESCRIPTION	COMMENTS	
000b	EHI 80 Boot Mode	f_{CPU}	: 180 MHz
		f_{BUS}	: 90 MHz
		f_{EHI}	: 90 MHz
	I2C Master Boot (EEPROM Boot) Mode		
001b	{UTXD1} = 0 : I2C Channel 0 Boot	f_{CPU}	: 180 MHz
	{UTXD1} = 1 : I2C Channel 1 Boot	f_{BUS}	: 90 MHz
	{UTXD0} = 0 : If I2C boot failed, go to UART channel 0 boot	f_{I2C}	: 90 MHz
	{UTXD1} = 1 : If I2C boot failed, go to UART channel 1 boot		
	Serial Flash Boot (Serial EEPROM Boot) Mode		
010b	{UTXD1} = 0 : If boot failed, go to UART boot.	f_{CPU}	: 180 MHz
	{UTXD1} = 1 : If boot failed, go to USB boot.	f_{BUS}	: 90 MHz
	{UTXD0} = 0 : In UART boot, the channel is 0.	f_{I2C}	: 90 MHz
	{UTXD0} = 1 : In UART boot, the channel is 1.		
011b	USB Function Boot (USB Boot) Mode	f_{CPU}	: 180 MHz
		f_{BUS}	: 90 MHz
100b	SPI Slave Boot Mode	f_{CPU}	: 180 MHz
		f_{BUS}	: 90 MHz
	UART Boot Mode	f_{CPU}	: 180 MHz
101b	{UTXD0} = 0 : The boot port is 0. (UART port 0)	f_{BUS}	: 90 MHz
	{UTXD0} = 1 : The boot port is 1. (UART port 1)		
	NOR Flash Boot Mode		
110b	{UTXD1,UTXD0} = 10b : 16bits NOR Flash	f_{CPU}	: 180 MHz
	{UTXD1,UTXD0} = 00b : 32 bits NOR Flash	f_{BUS}	: 90 MHz
	{UTXD1,UTXD0} = 11b : 8 bits NOR Flash		
	NAND Flash Boot Mode	f_{CPU}	: 180 MHz
111b	{UTXD1} = 0b : 8bits only	f_{BUS}	: 90 MHz
	{UTXD1} = 1b : 16bits only		

In the TCC79XX, there is an internal boot ROM for system initialization process. It contains the fundamental routines for system initialization or boot procedure through various device or interface such as EHI, I2C slave, SPI, USB device, parallel NOR flash, serial EEPROM with SPI or I2C protocol, NAND flash.

There are 7 modes for booting procedure. The Figure 35.2 illustrates the timing of reset sequence at power-up.

**Figure 35.2 Reset Sequence**

* The XIN and RSTN are the external pins and other signals are internal signals.

* The BM[2:0] are in input state during the “tEXT” and “tCFG”.

In the above figure, the ‘tEXT’ time should be longer than 2ms after XIN is stable. ‘tIRST’ is fixed-time with 64 clock cycles. But the ‘tCFG’ is about 1048576 clock cycles.

35.3 Overall Procedure

The following figure shows the overall flowchart of boot code.

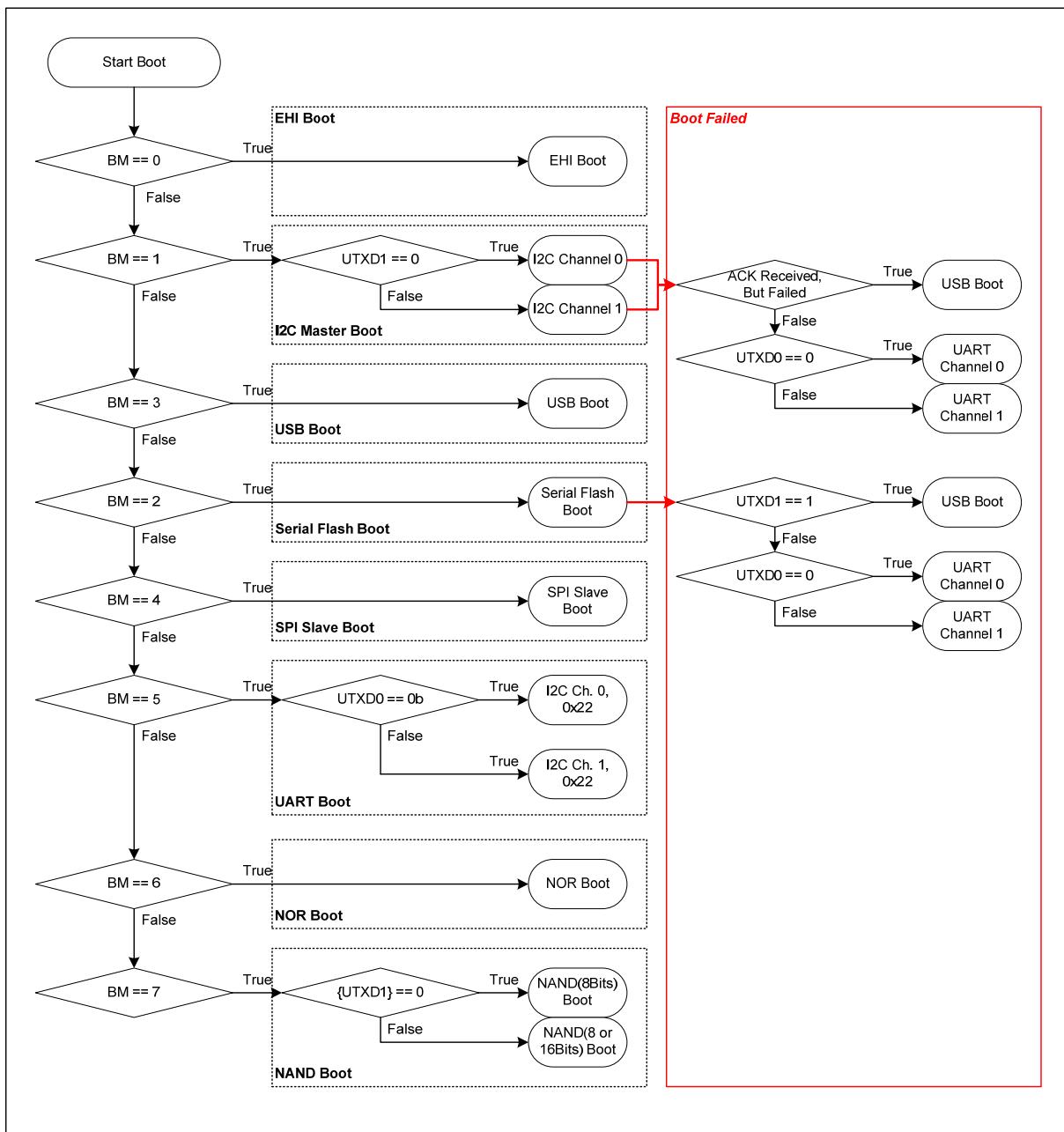


Figure 35.3 Overall Flowchart of Boot Code

35.4 EHI Boot (BM[2:0] = 000b)

In this mode, the on-chip CPU enables EHI module and sets HPINT to high. In this time, operating clock of EHI block is 90MHz. At the same time, the external host must wait until HPINT is high. After HPINT is asserted, the external host sets the ST field of EHST to INITCFG which indicates the interface bus width, the validity for this packet, and the total size of the BIP(Boot Information Packet).

After then, the on-chip CPU re-initializes the EHI with the bus width information in INITCFG.

7	6	5	4	3	2	1	0
1	0	0	0	0	BIPEN	VAL	BW
NOTE)		BIIPEN = 0 (recommend) VAL = should be 1 for valid INITCFG BW = 0 (8bits), 1 (16bits)					

Figure 35.4 INITCFG bit field

For handshaking between this LSI and the external host, the ST field of EHST register is used. The following is EHI booting procedure and refer to Figure 35.7.

- (1) After boot codes reinitialize this LSI using INITCFG, the EHST.ST is set to 0x55 by the on-chip CPU (On-chip CPU).
- (2) The external host uploads the data to this LSI. When uploading is completed, the external host set EHD register to start address of uploaded data and set EHST.ST to 0xAA. (External host).
- (3) When upload is completed, EHST.ST is not equal to 0x55. The on-chip CPU determines that type of the uploaded data is the BIP or the firmware program using the BIPEN field of INITCFG.(On-chip CPU)
- (4) If INITCFG.BIPEN is zero, the on-chip CPU jumps to start address in EHD register and EHI boot is completed. Otherwise, the uploaded data is the BIP and start address is BIP address. The BIP has much information about booting from EHI(Figure 35.5). After receiving the BIP from the external host and processing it, the on-chip CPU set the EHST.ST to 0x55. After then, the external host can upload program.(On-chip CPU)
- (5) The external host fills buffer0 that it is indicated by BIP. If it is full, the external host sets EHST.ST to 0x2C. After then, the external host fills buffer1 when it is full. And the external host sets EHST.ST to 0x25(Figure 35.6). This procedure is repeated by the external host until all of data is uploaded.(External host)

31	0
0	0
Total code size	
Buffer0 address	
Buffer1 address	
Buffer size	
Destination address	
start address	
reserved	

Figure 35.5 BIP data structure

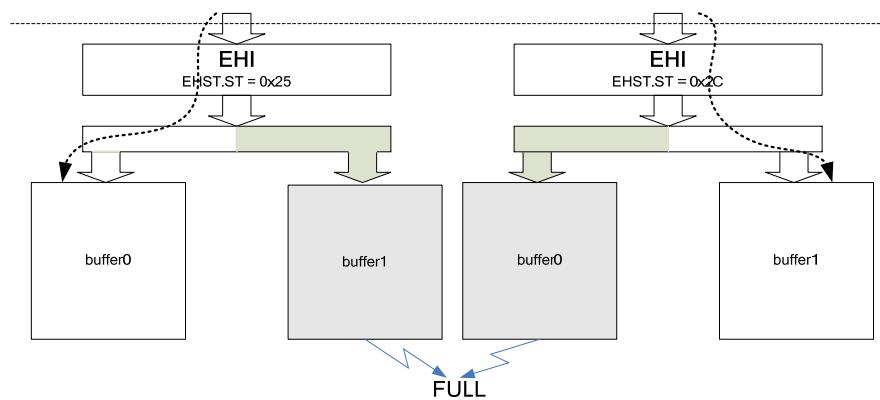


Figure 35.6 External host uploads program when BIPEN = 1

The EHI can manipulate either of 80 interfacing. The bus-width of EHI is determined by the BW field of INITCFG. During the procedure for booting from EHI, the external host can access all the address space in the TCC79XX. Therefore, if you want to read from or write to any register in TCC79XX, just do it as you want

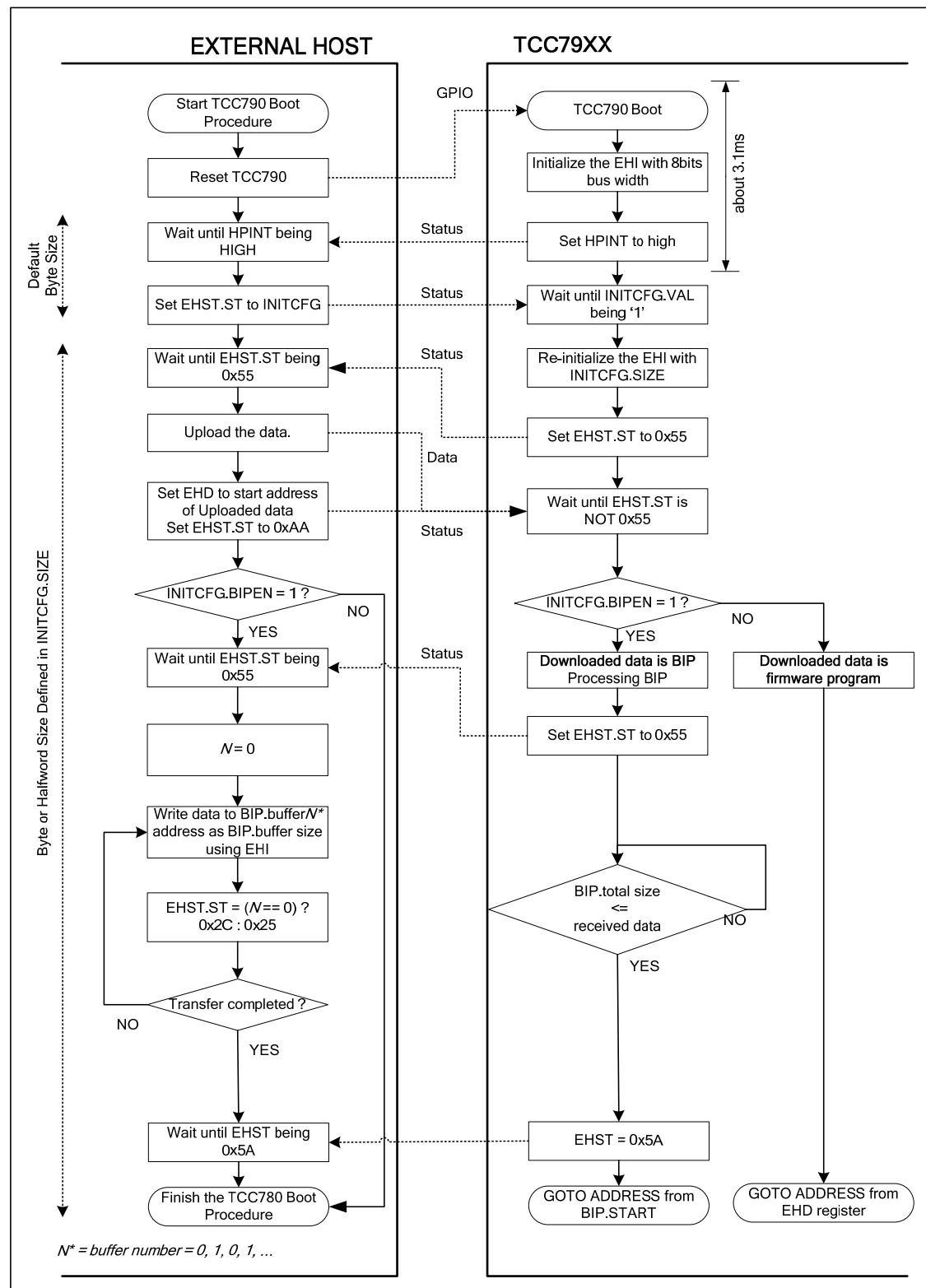


Figure 35.7 EHI Boot Procedure

35.5 USB Boot (BM == 011b)

This mode is mainly for firmware upgrade mode. In this mode, user can download a program into the user defined area. If the destination is SDRAM, user also send the configuration register value (SDCFG) for SDRAM.

When the failure occurs in some other boot modes, it may progress this boot sequence also.

The procedure of this mode is as follows.

- (1) The TCC79XX makes internal SRAM area starts from zero, and copies USB interrupt service routine to internal SRAM area.
- (2) It waits until USB connection is established.
- (3) Once it is connected, host transfers first the parameter for USB loader routine including destination address, user defined parameter and the amount of data to be transferred (with a unit of packet). If user wants to download into SDRAM, the SDCFG value must be transferred as user defined parameter.
- (4) The TCC79XX starts communicating between a host PC with fixed amount of data which is called as packet. The packet size of TCC79XX is 512 bytes.
- (5) At every successful reception of packet, it copies them where the destination address pointed, and after all amount of data has been copied, it starts program where the start address pointed.

Normally, the program downloaded is for writing user system firmware to non-volatile memory like NOR or NAND flash.

The following figure illustrates the sequence of USB boot mode described above.

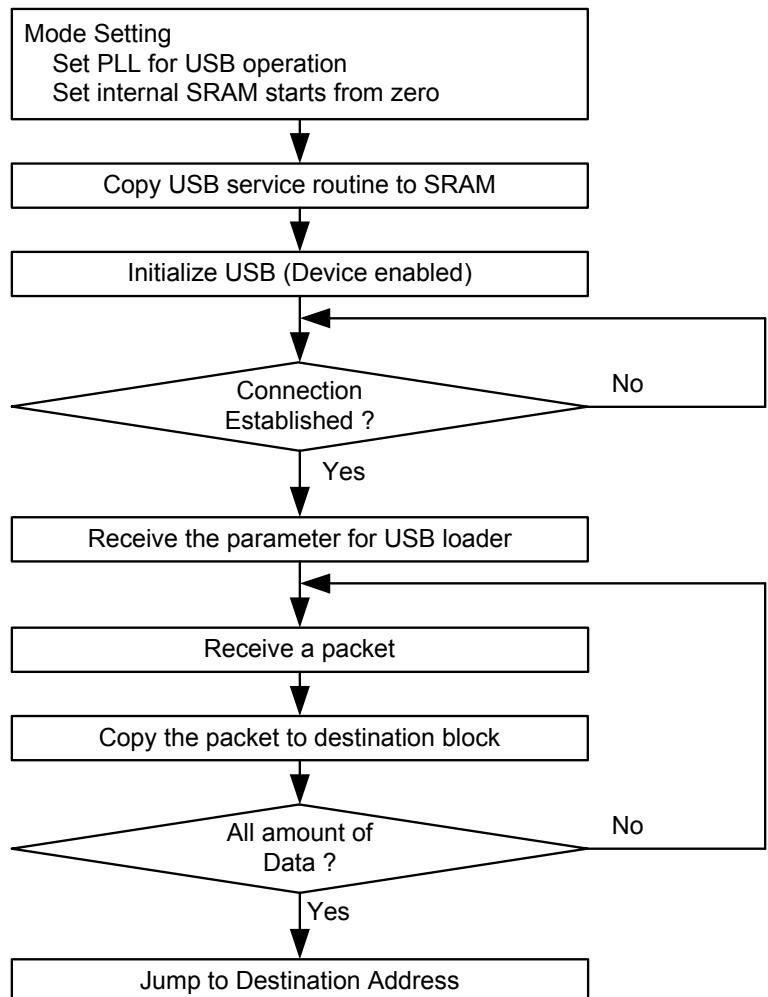


Figure 35.8 USB boot procedure

35.6 External NOR Boot (BM == 110b)

In this mode, the contents of NOR flash can be examined by checking CRC. If it is OK, it boots like a normal mode. If it is not OK, the TCC79XX automatically changes to USB boot mode so user can fix NOR contents via USB interface. NOR flash must be attached to CSN_NOR pin. The detailed procedure is as follows.

- (1) If 1st or 2nd word is 0xFFFFFFFF, TCC79XX goes to USB boot mode for F/W downloading.
- (2) If the 31'th bit of 5'th word in NOR flash is zero, the CRC checking procedure will be skipped.
- (3) The TCC79XX do the C2 (Dual CRC Checking) process on the image of external NOR flash. The C2 process is described at the Dual CRC Checking section.
- (4) After CRC process has finished and if it returns "OK", TCC79XX finishes booting procedure by jumping to the destination address.

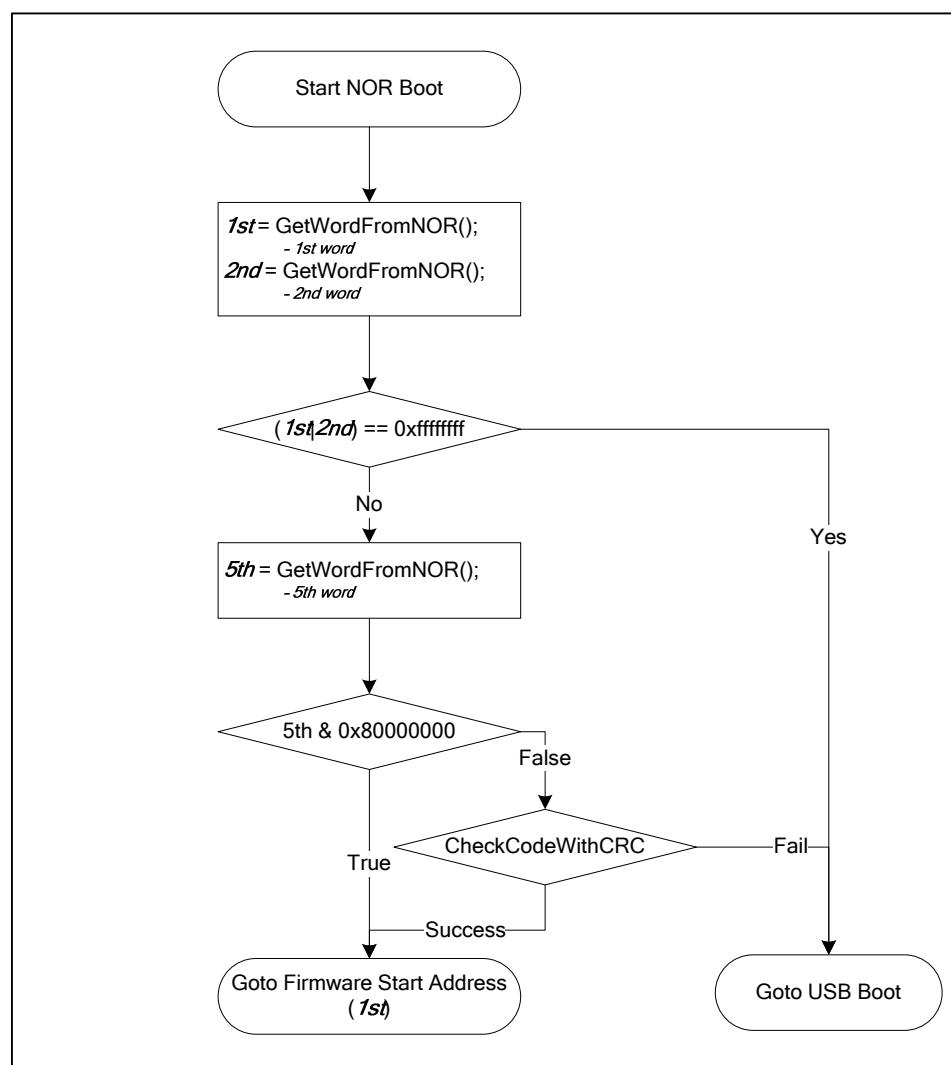


Figure 35.9 External NOR boot procedure

35.7 NFC NAND Boot (BM == 111b)

In this mode, the F/W code is read from NAND flash attached. To make use of this mode, the predefined structure such as Master Block and Master Cluster should be fused in the NAND flash.

The supportable configuration of NAND flash is as follows.

Case1) one NAND flash of 8-bit bus-width.

Case2) one NAND flash of 16-bit bus-width

Case3) two NAND flashes of 8-bit bus-width with the same chip enable signals.

In case of Case3, the UTXD1 should be 1 to regard as 16bit. So, the two NAND flashes can share the 16bit data-bus by using upper and lower 8 bit separately.

The boot sequence of this mode is as follows. If there exists any problem hard to recover during this sequence, it goes to USB boot mode automatically. It assumes the contents is stored as little endian format.

- (1) Check if device ID exists in the device ID table (Refer to Table 35.2). Determine the configuration of NAND flashes (Case1 ~ Case3). Setup the parameter for NFC block (pages per block, number of address cycles, etc.) according to device ID.
- (2) Read the last 1 word (4 bytes) in the spare area of 0 page of 0 block. It contains the block address of Master Block in upper 3 bytes, and number of Master Cluster in lower 1 byte.

31	30	8	7	0
Block Address of Master Block			Number of Master Cluster	

- (3) Load and Construct the golden image of Master Cluster from the Master Block to internal SRAM. The structure of Master Block and Master Cluster is represented in Figure 35.11.
- (4) The 1st word of golden Master Cluster determines the next procedure as follows. In case of 0x54C34396 ('T', C3, 'C', 96), the TCC79XX regard the Master Cluster as the Master Code, and finishes booting procedure by jumping to the address of 2nd word in golden Master Cluster. In case of 0x54C34996 ('T', C3, 'I', 96), the TCC79XX starts loading according to the contents of golden Master Cluster.
- (5) After finishing download of F/W, the TCC79XX progress C² (Dual CRC Checking) process on the loaded image. The C² process is described at the "35.12 Dual CRC Checking" on page 35-20.
- (6) After C² process has finished, and it is OK, the TCC79XX finishes booting procedure by jumping to the destination address which is contained in golden Master Cluster.

In loading from NAND flash, the TCC79XX uses MLC-ECC regardless of NAND type, so loading is accomplished by unit of 512+16 bytes or 512+20. The MLC-ECC can correct up to 8 symbol errors, so the F/W code can be stored with high robustness.

Table 35.2 Supported NAND flash types

Size (bytes)	Size of Page (bytes)	Pages / Block	Number of Page	Device ID x8 / x8 / x16 / x16
8M	512	16	16K	39 / E6 / 49 / 59
16M	512	16	32K	33 / 73 / 43 / 53
32M	512	32	64K	35 / 75 / 45 / 55
64M	512	32	128K	36 / 76 / 46 / 56
128M	512	32	256K	78 / 79 / 72 / 74
256M	512	32	512K	71
512M	512	32	1024K	DC
128M	2048	64	64K	A1 / F1 / B1 / C1
256M	2048	64	128K	AA / DA / BA / CA
512M	2048	64	256K	AC / DC / BC / CC
1G	2048	64	512K	A3 / D3 / B3 / C3
2G	2048	64	1024K	A5 / D5 / B5 / C5
8G	4096	128	2048K	D7/D3

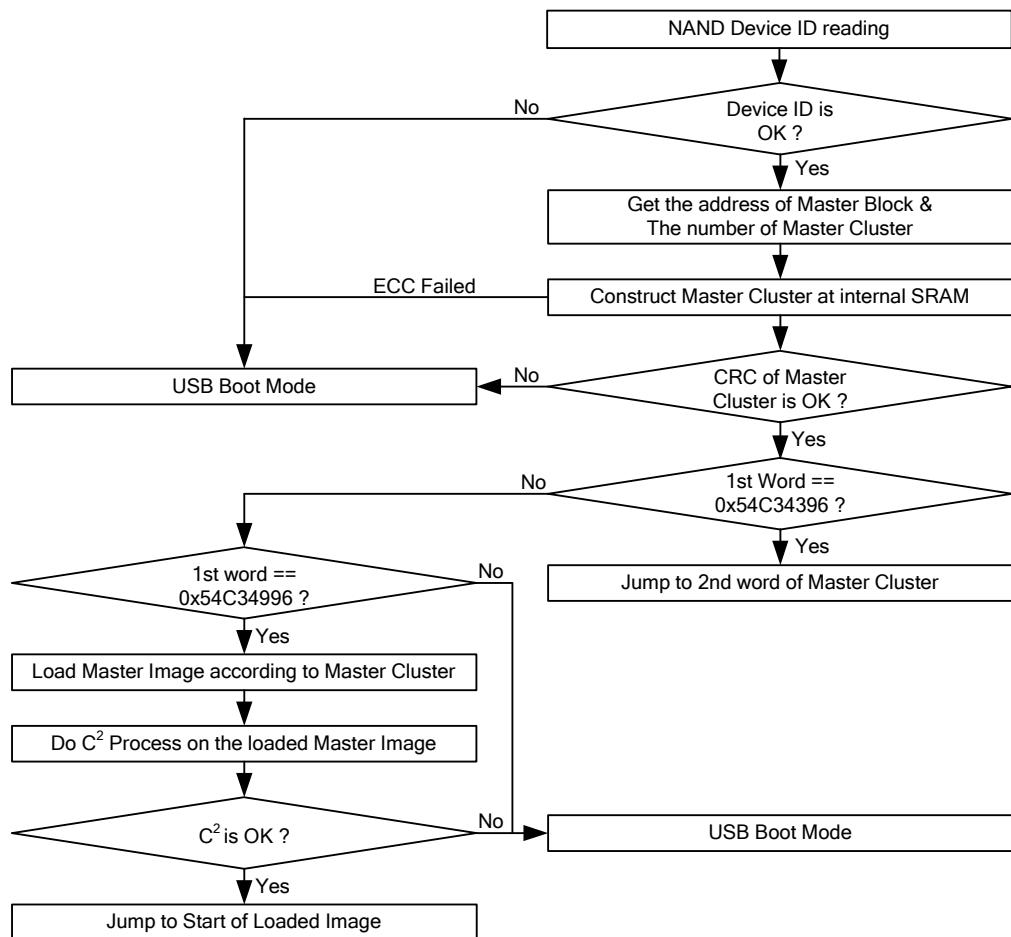


Figure 35.10 NAND boot procedure

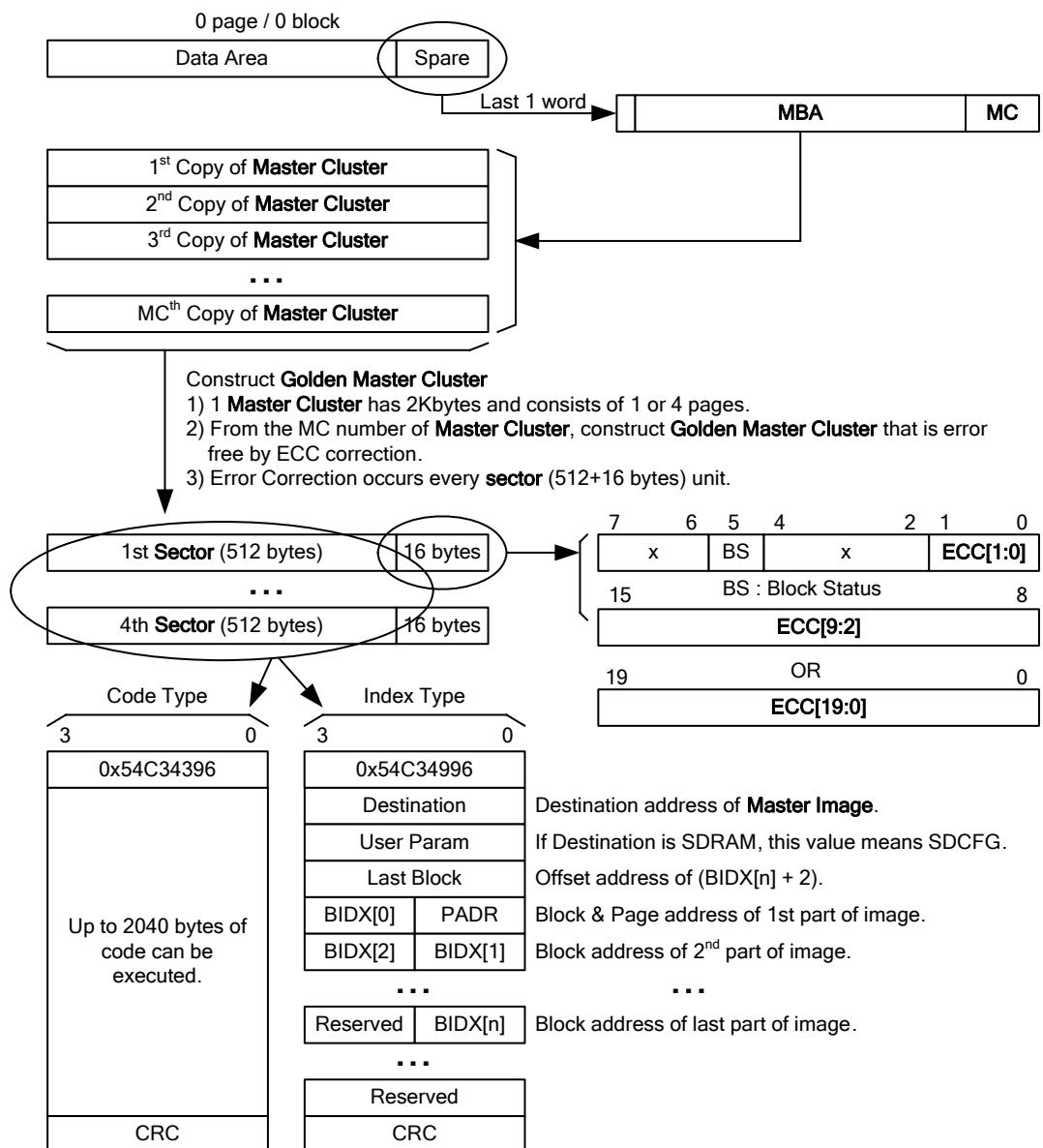


Figure 35.11 Basic Structure and Boot Flow of NAND Boot

35.8 I2C Master Boot – EEPROM Boot (BM == 001b)

In this mode, the F/W code would be read from serial EEPROM attached to the I2C ports. It interfaces with standard I2C protocol. If serial EEPROM is not attached, the TCC79XX changes to UART boot mode.

The procedure checks if there exist EEPROM first. If there exist an EEPROM, the TCC79XX do the following procedure. If certain problem which can not be solved has occurred, it goes to USB boot mode automatically.

The UTXD1 port determines the boot channel for I2C. If pulled-up, the channel 1 port would be used and otherwise the channel 0 port used.

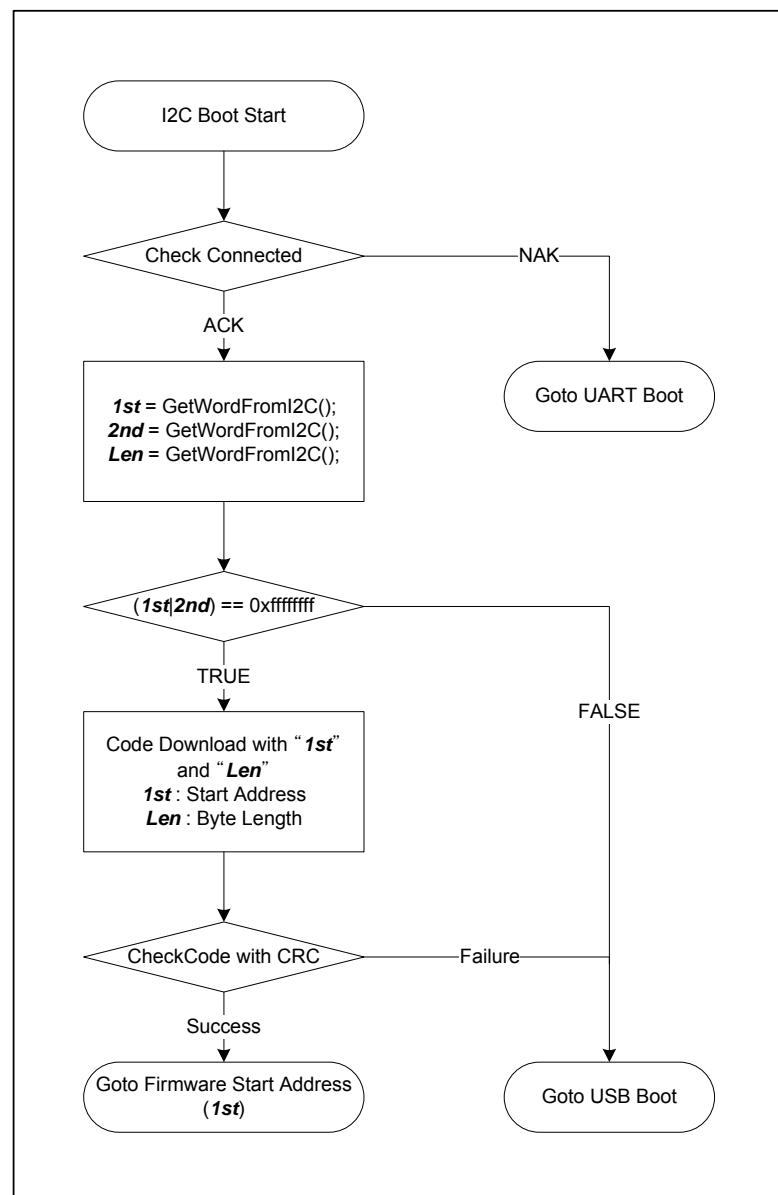


Figure 35.12 I2C Boot Procedure

35.9 Serial Flash Boot

The serial flash boot mode supports the serial NOR flash with SPI protocol. The HPXD[7:4] are used for interfacing the serial EEPROM. The UTXD1 pin determines which mode to branch after serial EEPROM booting failed. If UTXD1 is pulled-up, the boot mode would be changed to USB boot after failed and otherwise to the UART boot mode. In the case of UART boot mode, the UTXD0 determines that which port can be used in UART boot mode. If UTXD0 is pulled-up, the UART port 1 will be used and otherwise UART port 0 used.

```
typedef struct {
    unsigned short    MID;           // Unused for booting
    unsigned short    DID;           // Unused for booting
    unsigned          SectorSize;    // Unused for booting
    unsigned          BlockSize;     // Unused for booting
    unsigned          BlockNum;      // Unused for booting
    unsigned          uDest;         // Used for booting
    unsigned          uSDCFG;        // Used for booting
    unsigned          uLength;       // Used for booting
    unsigned          uExecStart;    // Used for booting
    unsigned char     Name[20];      // Unused for booting
    unsigned char     Rsv[256-56];   // Rsv[0] : DLDV value // Used for booting
    unsigned          uCRC;          // Used for booting
} sSFBootHeader;
```

Figure 35.13 Data Structure for Header Information Stored In Serial Flash

In the above figure, the data structure required for booting from serial flash is described. The variables named as uDest, uSDCFG, uLength, and uExecStart are used in the boot code. If the uDest is not equal to uExecStart, the uExecStart is used to DLDV, which determines the clock frequency of GPSB(General Purpose Serial Bus) controller.

The uLength field is total number of bytes to stored in the serial flash and should be aligned to word(4bytes).

The overall procedure for serial flash boot is shown in the following figure.

```

Void IO_GPSBM_GetWordData (unsigned *data, unsigned nwords);
// Function to read from serial flash
// "data" is destination pointer
// "nwords" is the total number of words to read

Void StartSFlashBoot ()
{
    IO_GPSB_PortConfig ();      // Port Configuration for Serial Flash Interface
    IO_GPSBM_Init (); // Initialization for GPSB controller
    IO_GPSB_FlushBuffer ();    // Flushing Buffer for Initial Loading
    IO_GPSBM_GetWordData ((unsigned *)pHeader,0,sizeof(sSFBootHeader)/4);

    if ( IsEmptyFlash () ) return (-1); // Booting Failed

    uData = IO_UTIL_CalcCRC32(0,(unsigned *)pHeader,256-4,0);
    if ( uData != pHeader->uCRC ) return (-1); // Booting Failed

    IO_UTIL_SetMEM (pHeader->uDest,pHeader->uSDCFG);
    // Initialization for Memory Controller

    if ( (pHeader->uDest&0xf0000000) != (pHeader->uExecStart&0xf0000000) ) {
        HwGPSB0->uMOD.bMOD.DLDV = pHeader->uExecStart;
        uExecStart = pHeader->uDest;
    }

    IO_GPSB_FlushBuffer ();
    IO_GPSBM_GetWordData
    ((unsigned *)pHeader->uDest, SFLASH_PAGE_DATA_SIZE, (pHeader->uLength>>2));
    // SFLASH_PAGE_DATA_SIZE : 256

    if (IO_UTIL_CheckCODE((unsigned *)pHeader->uDest, pHeader->uLength, 0))
        return 0;           /* to usb boot */

    goto_firmware (pHeader->uExecStart);
}

```

Figure 35.14 Pseudo Code for Serial Flash Booting Procedure

35.10 SPI Slave Boot

The SPI slave boot uses the ports named HPXD[7:4]. And if boot failed, the boot mode would jump to the USB boot – USB firmware download mode.

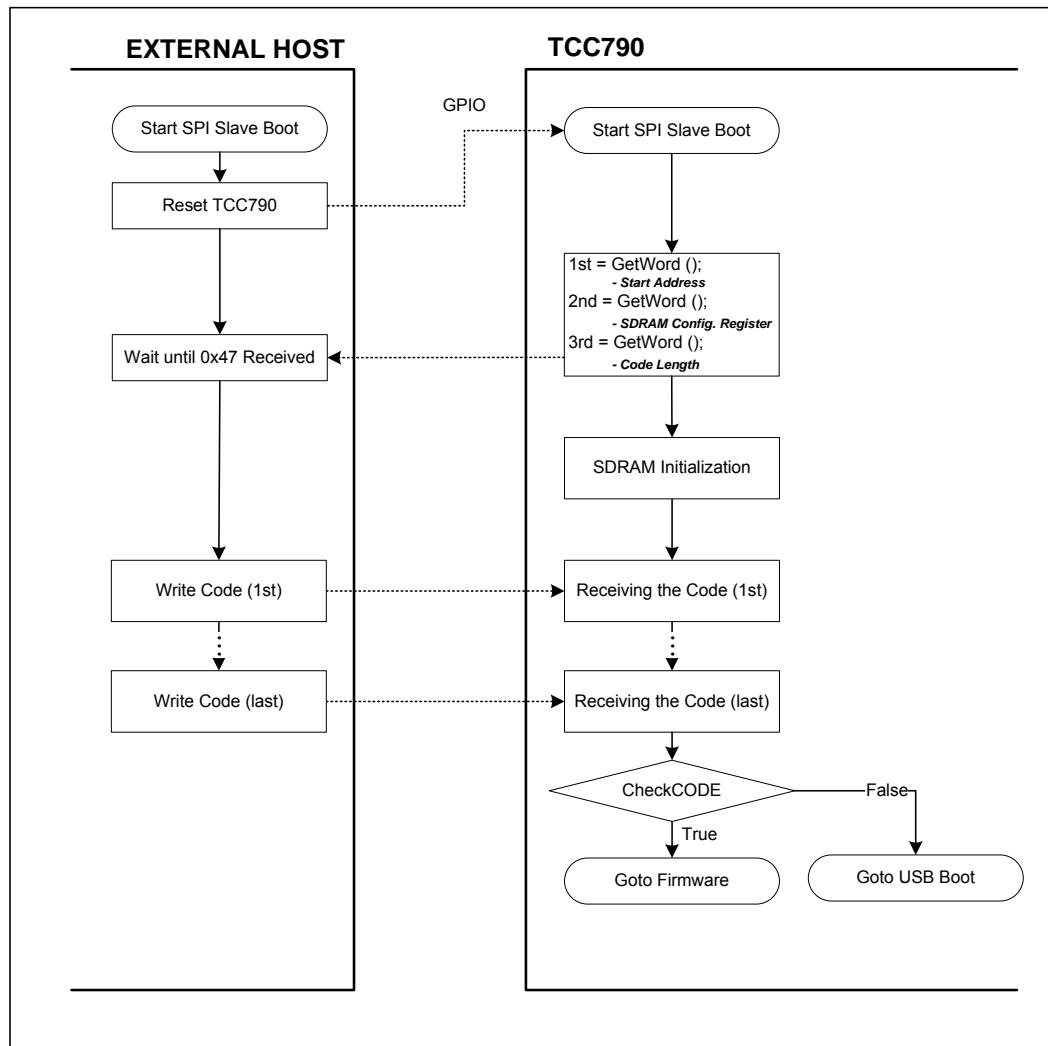


Figure 35.15 SPI Slave Boot Procedure

35.11 UART Boot

The UART boot mode can be changed from I2C boot mode or serial flash boot mode. If the I2C eeprom boot mode was failed, the UART boot would be started with the configuration of the UTXD0 pin. If UTXD0 is pulled-up, the UTXD0 and URXD0 will be used for booting and if UTXD0 is pulled-down, the UTXD1 and URXD1 will be used.

The detailed procedure is as follows. It uses little endian for word manipulation. So, the LSB byte is received first, and MSB byte is received last.

- (1) Enable UART I/F. (fUART=24MHz, 115.2kbps, Data 8bit, Non-parity, 1 Stop bit)
- (2) Send ACK (0x52, ASCII code of 'R').
- (3) Wait until ACK is received. ('H' or 'R')
- (4) Send ACK
- (5) Receive the Destination Address.
- (6) Send ACK
- (7) Receive the User Parameter (if the destination address is SDRAM, use user parameter as SDCFG value.)
- (8) Send ACK
- (9) Receive the size of code in byte unit.
- (10) Send ACK
- (11) Receive the Destination Address. (It should be same as (5))
- (12) Send ACK
- (13) If ACK in (3) is 'H', new divisor value (DLM, DLL) is received (fUART = 24MHz) and baud rate is reconfigured.
- (14) Load the code from HOST to destination address
- (15) Do the CRC checking process on the downloaded image.
- (16) After CRC checking process has finished, and it is OK, TCC79XX finishes booting procedure by jumping to the destination address. Or, it restarts UART boot from the first step.

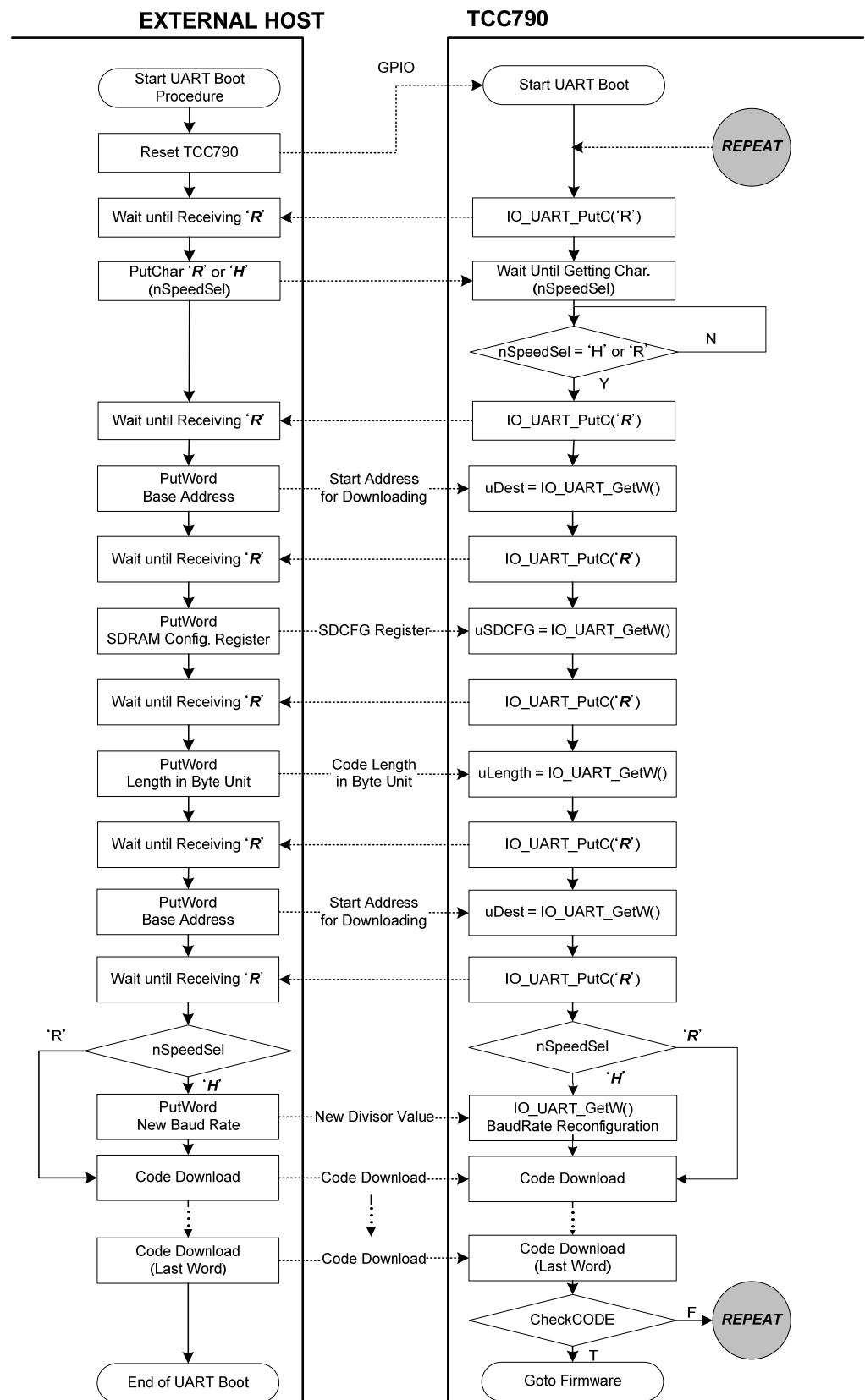


Figure 35.16 UART boot procedure

35.12 Dual CRC Checking

The TCC79XX can check the CRC for the downloaded image according to the dedicated algorithm. This is described as follows.

Calculate CRC on the downloaded image up to 64Kbytes. In calculation, the word at the offset of 0x10 and 0x14 is skipped, because the good CRC code is contained at the offset of 0x10, and the word at the offset of 0x14 means the address of user-defined CRC routine.

After the first CRC calculation has been done, and it is same as the value at the offset address of 0x10, TCC79XX check if the address of user-defined CRC routine is not 0. Or, TCC79XX goes to pre-defined boot mode (USB or UART boot) automatically.

If there exist user-defined CRC routine, call that routine.

The user-defined CRC routine checks its own CRC code and return with Equal condition if it is OK or return Not Equal condition if it fails.

If there are failures on the user-defined CRC checking, TCC79XX goes to pre-defined boot mode (USB or UART boot) automatically.

The following code shows same CRC generation algorithm used in the first CRC routine.

```
word calc_crc (word *base, word length, word *crctable)
{ // contents of crctable is acquired by gen_crc () function
    word     crcout    = 0;
    word     cnt, i, code, tmp;

    length  >>= 2;           // convert into word unit.
    for (cnt = 0; cnt < length; cnt++) {
        if (cnt == 0x04 or cnt == 0x05) // skip offset of 0x10 and 0x14
            continue;
        code = base[cnt];
        for (i = 0; i < 4; i++) {
            tmp     = code ^ crcout;
            crcout = (crcout >> 8) ^ crctable[tmp & 0xFF];
            code   = code >> 8;
        }
        return  crcout;
    }

    void gen_crc (word *crctable)
    { // Polynomial = x32 + x31 + x16 + x15 + x4 + x3 + x + 1
        word     crc, cnt, i;

        for (cnt = 0; cnt < 0x100; cnt++) {
            crc   = cnt;
            for (i = 0; i < 8; i++) {
                if (crc & 1)    crc     = (crc >> 1) ^ 0xD8018001;
                else           crc     = (crc >> 1);
            }
            crctable[cnt] = crc;
        }
    }
}
```

APPENDIX A

PLLs in the TCC79XX have the following constraints.

$$F_{VCO} = (m * F_{IN}) / (p) : (F_{IN} \text{ is XIN oscillator})$$

$$F_{PLL} = F_{VCO} / (2^s)$$

You must get the proper divider setting values from the following table. They are not allowed to use any divider setting which isn't obtained from the table. If you have any questions then consult with *Telechips*. To set the selected divider value, refer to PLL0CFG and PLL1CFG register on page 23-5.

FIN (MHz)	P	M	S	FPLL(MHz)	FVCO(MHz)
12	3	128	5	16	512
12	3	99	4	24.75	396
12	3	140	3	70	560
12	3	144	3	72	576
12	3	86	2	86	344
12	3	101	2	101	404
12	3	118	2	118	472
12	3	132	2	132	528
12	3	73	1	146	292
12	3	90	1	180	360
12	3	92	1	184	368
12	3	95	1	190	380
12	3	96	1	192	384
12	3	102	1	204	408
12	3	104	1	208	416
12	3	105	1	210	420
12	3	107	1	214	428
12	3	108	1	216	432
12	3	111	1	222	444
12	3	113	1	226	452
12	3	114	1	228	456
12	3	115	1	230	460
12	3	117	1	234	468
12	3	118	1	236	472
12	3	122	1	244	488
12	3	123	1	246	492
12	3	124	1	248	496
12	3	126	1	252	504
12	3	127	1	254	508
12	3	129	1	258	516
12	3	130	1	260	520
12	3	131	1	262	524
12	1	44	1	264	528
12	3	133	1	266	532
12	3	138	1	276	552
12	3	141	1	282	564
12	3	145	1	290	580
12	3	147	1	294	588
12	5	246	1	295.2	590.4
12	3	76	0	304	304

FIN (MHz)	P	M	S	FPLL(MHz)	FVCO(MHz)
12	3	77	0	308	308
12	2	52	0	312	312
12	3	79	0	316	316
12	3	80	0	320	320
12	3	81	0	324	324
12	3	82	0	328	328
12	2	55	0	330	330
12	3	83	0	332	332
12	3	84	0	336	336
12	3	85	0	340	340
12	3	87	0	348	348
12	3	88	0	352	352
12	2	59	0	354	354
12	3	89	0	356	356
12	1	30	0	360	360
12	6	181	0	362	362
12	3	91	0	364	364
12	4	123	0	369	369
12	6	185	0	370	370
12	1	31	0	372	372
12	3	95	0	380	380
12	4	127	0	381	381
12	1	32	0	384	384
12	3	97	0	388	388
12	2	65	0	390	390
12	3	98	0	392	392
12	4	131	0	393	393
12	1	33	0	396	396
12	4	133	0	399	399
12	3	100	0	400	400
12	3	101	0	404	404
12	5	169	0	405.6	405.6
12	6	203	0	406	406
12	1	34	0	408	408
12	6	205	0	410	410
12	3	103	0	412	412
12	3	104	0	416	416
12	5	174	0	417.6	417.6
12	1	35	0	420	420
12	3	106	0	424	424
12	3	107	0	428	428
12	4	143	0	429	429
12	6	215	0	430	430
12	1	36	0	432	432
12	3	109	0	436	436
12	3	110	0	440	440
12	6	221	0	442	442
12	1	37	0	444	444
12	3	112	0	448	448
12	2	75	0	450	450
12	3	113	0	452	452

FIN (MHz)	P	M	S	FPLL(MHz)	FVCO(MHz)
12	6	227	0	454	454
12	1	38	0	456	456
12	3	115	0	460	460
12	5	193	0	463.2	463.2
12	3	116	0	464	464
12	6	233	0	466	466
12	1	39	0	468	468
12	6	235	0	470	470
12	3	118	0	472	472
12	2	79	0	474	474
12	3	119	0	476	476
12	1	40	0	480	480
12	3	121	0	484	484
12	2	81	0	486	486
12	3	122	0	488	488
12	6	245	0	490	490
12	1	41	0	492	492
12	3	124	0	496	496
12	5	207	0	496.8	496.8
12	2	83	0	498	498
12	3	125	0	500	500

Change Log

Revision	Date	Description
1.03	Jan 23, 2009	<ul style="list-style-type: none"> ● CKC - Page 23-17: The description about "Enter Halt Mode" is added. ● Introduction - Page 1-2: "Operating Modes" are modified. ● UART -Page 22-1: IrDA is removed. ● CKC - Page 23-4: Addresses of MCLKCTRL and SCLKCTRL Register are corrected. ● RTC - Page 28-12: PWDN description for PMWKUP is corrected.
1.02	Dec 17, 2008	<ul style="list-style-type: none"> ● Port Configuration - Reset value is corrected. <ul style="list-style-type: none"> - Page 33-3: AINCFG, Page 33-18: CCKO, Page 33-17:DAO, Page 33-15:PORTCFG9, Page 33-16:PORTCFG10 ● APPENDIX A <ul style="list-style-type: none"> - <u>PLL Table is updated. (FPLL=118MHz, 295.2MHz, 354MHz)</u> ● LCD <ul style="list-style-type: none"> - Page 1-2, 6-1: Add pixel data width when LCD CPU Interface is used. ● CKC <ul style="list-style-type: none"> - Page 23-5, APPENDIX A: Add PLL PMS Table ● I2C <ul style="list-style-type: none"> - Page 26-9: RxACK is corrected. ● Port Configuration <ul style="list-style-type: none"> - Page 33-2, 33-19: R/W type of PORTCFG13 is corrected. - Page 33-3, 33-28: R/W type of CPUD7 and CPUD9 are corrected. ● Boot Procedure <ul style="list-style-type: none"> - Page 35-3: tEXT is modified.
1.01	Oct 8, 2008	<ul style="list-style-type: none"> ● Boot Procedure <ul style="list-style-type: none"> - Page 35-2: Data bit-width selection is corrected when booting from NOR. - Page 35-19: UART boot procedure is corrected.
1.00	Jul 9, 2008	<ul style="list-style-type: none"> ● LCD Interface <ul style="list-style-type: none"> - Page 5-1, 5-2: LUT in the Figure 5.1and Figure 5.2 are corrected. - Page 5-19, 5-50: S2LC address is corrected. - Page 5-25: LEWC bit width is corrected. - Page 5-26: VDF bit width is corrected. - Page 5-26: LVTIME2.FEWC bit width is corrected. - Page 5-28: Bit width of LDP5/4/3 register is corrected. - Page 5-37, 5-41, 5-45: WIDTH bit width is corrected. - Page 5-48: Addresses of DCLSA0 and DCLSA1 are corrected. - Page 5-49: Y2RM description is added ● NTSC/PAL Encoder <ul style="list-style-type: none"> - Page 7-16: PXDATA bit in VENCIF.FMT description is corrected. - Page 7-17: Example of NTSC/PAL is modified. - Page 7-21: CGMS description is added. ● Memory-to-Memory Scaler <ul style="list-style-type: none"> - Page 8-11: MSCCTR.CON is added. ● USB Controller <ul style="list-style-type: none"> - Page 9-3, 9-12: PLICR and PCR are added. - Page 9-21: Descriptions about DPPD and DMFD are corrected and added. - Page 9-25: Descriptions about TM, XCVRSEL, and OPMODE of UPCR2 register are added. ● NFC <ul style="list-style-type: none"> - Page 14-6: The descriptions about CS3SEL and CS2SEL are corrected. ● Camera Interface <ul style="list-style-type: none"> - Page 16-2: CK_CTRL in Figure 16.1 is removed and the related description is corrected.
0.20	Jun 23, 2008	<ul style="list-style-type: none">

- Page 16-13: IIS description is added.
- Graphic Engine
 - Page 17-3: Parts related to FCH2 are removed.
 - Page 17-2, 17-27, 17-28, 17-29, 17-29: Figure 17.1, Figure 17.5, Figure 17.6, Figure 17.7, and Figure 17.8 are modified.(FCH2 is removed)
 - Page 17-14: “OP_CTRL.C0/1_SEL Register is 2'b01.” Is changed to “OP_CTRL.C0/1_SEL Register is 2'b10.”
 - Page 17-15: S_CTRL description is corrected.
 - Page 17-19: OP_CTRL description is corrected.
 - Page 17-25: GE_CTRL description is corrected.
- EHI
 - Page 18-16: Figure 18.5 is modified.
- GPSB
 - Page 20-5: STATUS description is added.
 - Page 20-13: DMAICR description is added.
 - Page 20-15: “Figure 20.7 SPI Timing 2” is corrected.
- CKC
 - Page 23-14: 5.2us is changed to 5.2ms in WDTCNT description
- Vectored Interrupt Controller
 - Page 24-3: Field name “UT0” is changed to “UT”.
 - Page 24-8: ALLMSK description is added.
- Timer/Counter
 - Page 25-5: The description about TMREF4/5 is removed correctly.
- I2C
 - Page 26-1, 26-2: I2C clock name (PCK_I2C) is corrected.
 - Page 26-12: SLV description is added.
- RTC
 - Page 28-5: Figure 28.6 is added.
- External Memory Controller
 - Page 29-2: The description about supported SDRAM is corrected.
- Port Multiplexer & GPIO
 - Page 33-30: Reset values of XD, XA, and MCTRL are changed.
- Boot Procedure (new)