

# **Symphony™ DSP56724/DSP56725 Multi-Core Audio Processors Reference Manual**

Document Number: DSP56724RM  
Rev. 1  
02/2009

### **How to Reach Us:**

#### **Home Page:**

[www.freescale.com](http://www.freescale.com)

#### **Web Support:**

<http://www.freescale.com/support>

#### **USA/Europe or Locations Not Listed:**

Freescale Semiconductor, Inc.  
Technical Information Center, EL516  
2100 East Elliot Road  
Tempe, Arizona 85284  
1-800-521-6274 or  
+1-480-768-2130  
[www.freescale.com/support](http://www.freescale.com/support)

#### **Europe, Middle East, and Africa:**

Freescale Halbleiter Deutschland GmbH  
Technical Information Center  
Schatzbogen 7  
81829 Muenchen, Germany  
+44 1296 380 456 (English)  
+46 8 52200080 (English)  
+49 89 92103 559 (German)  
+33 1 69 35 48 48 (French)  
[www.freescale.com/support](http://www.freescale.com/support)

#### **Japan:**

Freescale Semiconductor Japan Ltd.  
Headquarters  
ARCO Tower 15F  
1-8-1, Shimo-Meguro, Meguro-ku  
Tokyo 153-0064  
Japan  
0120 191014 or  
+81 3 5437 9125  
[support.japan@freescale.com](mailto:support.japan@freescale.com)

#### **Asia/Pacific:**

Freescale Semiconductor China Ltd.  
Exchange Building 23F  
No. 118 Jianguo Road  
Chaoyang District  
Beijing 100022  
China  
+86 010 5879 8000  
[support.asia@freescale.com](mailto:support.asia@freescale.com)

#### **For Literature Requests Only:**

Freescale Semiconductor  
Literature Distribution Center  
P.O. Box 5405  
Denver, Colorado 80217  
1-800 441-2447 or  
+1-303-675-2140  
Fax: +1-303-675-2150  
LDCForFreescaleSemiconductor  
[@hibbertgroup.com](mailto:@hibbertgroup.com)

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2009. All rights reserved.

# Contents

Paragraph Number	Title	Page Number
---------------------	-------	----------------

## About This Book

Audience .....	xxxv
Organization.....	xxxv
Suggested Reading.....	xxxv
Conventions .....	xxxv

## Chapter 1 Introduction

1.1	Overview.....	1-1
1.2	Block Diagram .....	1-3
1.3	Features .....	1-4
1.4	Overview of Peripherals .....	1-5
1.4.1	Direct Memory Access Controller (DMA, DMA_1).....	1-5
1.4.2	Program Interrupt Controller (PIC, PIC_1) .....	1-5
1.4.3	Enhanced Serial Audio Interfaces (ESAI, ESAI_1, ESAI_2, ESAI_3) .....	1-6
1.4.4	Serial Host Interfaces (SHI, SHI_1) .....	1-6
1.4.5	Triple Timers (TEC, TEC_1).....	1-6
1.4.6	Watch Dog Timers (WDT, WDT_1).....	1-6
1.4.7	Core Integration Modules (CIM, CIM_1) .....	1-6
1.4.8	Sony/Philips Digital Interface (S/PDIF).....	1-6
1.4.9	Asynchronous Sample Rate Converter (ASRC).....	1-7
1.4.10	External Memory Controller (EMC) .....	1-7
1.4.11	Clock Generation Module (CGM).....	1-7
1.4.12	Shared Memory.....	1-8
1.4.13	Inter-Core Communication (ICC).....	1-8
1.4.14	Shared Bus Arbiters .....	1-8
1.4.15	Chip Configuration Module.....	1-8
1.4.16	JTAG Controller .....	1-8

## Chapter 2 Signal Descriptions

2.1	Signal Groupings .....	2-1
2.2	Signals in Each Functional Group .....	2-6
2.2.1	Power .....	2-6
2.2.2	Ground .....	2-6
2.2.3	SCAN.....	2-7
2.2.4	Clock and PLL .....	2-7
2.2.5	Reset Pin .....	2-8

# Contents

Paragraph Number	Title	Page Number
2.2.6	Interrupt and Mode Control .....	2-8
2.2.7	DSP Core-1 Non-Maskable Interrupt (NMI1) .....	2-10
2.2.8	Serial Host Interface (SHI and SHI_1) .....	2-10
2.2.9	Enhanced Serial Audio Interface Signals (ESAI, ESAI_1, ESAI_2, ESAI_3) .....	2-14
2.2.10	Watch Dog Timer (WDT) .....	2-26
2.2.11	External Memory Controller (EMC) .....	2-26
2.2.12	S/PDIF Audio Interface Signals .....	2-29
2.2.13	Dedicated Port G GPIOs .....	2-30
2.2.14	JTAG/OnCE Interface Signals .....	2-30

## Chapter 3 Memory Map

3.1	Overview .....	3-1
3.2	Data and Program Memory Maps .....	3-2
3.3	Peripheral Register Memory Map .....	3-3

## Chapter 4 DSP56300 Platform

4.1	Overview .....	4-1
4.2	DSP56300 Core Features .....	4-2
4.3	DSP56300 Block Descriptions .....	4-3
4.3.1	Data ALU .....	4-3
4.3.1.1	Data ALU Registers .....	4-4
4.3.1.2	Multiplier-Accumulator (MAC) .....	4-4
4.3.2	Address Generation Unit (AGU) .....	4-4
4.3.3	Program Control Unit (PCU) .....	4-5
4.3.4	Internal Buses .....	4-5
4.3.5	OnCE Module .....	4-6

## Chapter 5 Core Configuration

5.1	Introduction .....	5-1
5.2	Operating Mode Register (OMR) .....	5-1
5.3	Status Register (SR) .....	5-3
5.4	DSP Cores Operating Modes .....	5-6
5.5	Interrupt Priority Registers .....	5-9
5.6	DMA Request Sources .....	5-19
5.7	Chip ID Register .....	5-20

# Contents

Paragraph Number	Title	Page Number
<b>Chapter 6</b>		
<b>Core Integration Module (CIM, CIM_1)</b>		
6.1	Overview .....	6-1
6.1.1	Memory Map .....	6-2
6.1.2	Register Summary .....	6-2
6.2	Register Descriptions .....	6-3
6.2.1	Chip ID Register (CHIDR) .....	6-3
6.2.2	DMA Stall Register (DMAS) .....	6-4
6.2.3	OnCE Global Data Bus Register (OGDB) .....	6-5
<b>Chapter 7</b>		
<b>Clock Generation Module (CGM)</b>		
7.1	Introduction .....	7-1
7.1.1	Overview of Modes .....	7-1
7.1.2	Features .....	7-2
7.1.3	Modes of Operation .....	7-3
7.1.4	External Signal Description .....	7-4
7.2	Functional Description .....	7-4
7.2.1	Clocks .....	7-4
7.2.2	Reset .....	7-4
7.2.3	Interrupts .....	7-4
7.2.4	Internal PLL Block .....	7-4
7.2.4.1	PLL Operating Modes .....	7-5
7.2.4.2	Programming the Output Clock Frequency .....	7-6
7.2.5	Low Power Divider .....	7-6
7.3	Memory Map and Register Definition .....	7-7
7.3.1	Memory Map .....	7-7
7.3.2	Register Summary .....	7-7
7.3.3	Register Descriptions .....	7-8
7.3.3.1	Shared Peripheral Clock Enable Register (SPENA) .....	7-8
7.3.3.2	PLL Control Register (PTCL) .....	7-8
7.3.3.3	ASRC Clock Dividing Control Register (ASCDR) .....	7-11
<b>Chapter 8</b>		
<b>General Purpose Input/Output (GPIO)</b>		
8.1	Introduction .....	8-1
8.2	Programming Model .....	8-1
8.2.1	Port C, Port E, Port C1, Port E1 Signals and Registers .....	8-2

# Contents

Paragraph Number	Title	Page Number
8.2.2	Port H Signals and Registers .....	8-2
8.2.2.1	Port H Control Register (PCRH) .....	8-2
8.2.2.2	Port H Direction Register (PRRH) .....	8-3
8.2.2.3	Port H Data Register (PDRH).....	8-3
8.2.2.4	EXTAL Clocking Control for ESAI and ESAI_1 .....	8-4
8.2.3	Port H1 Signals and Registers .....	8-4
8.2.3.1	Port H1 Control Register (PCRH1) .....	8-5
8.2.3.2	Port H1 Direction Register (PRRH1) .....	8-5
8.2.3.3	Port H1 Data Register (PDRH1).....	8-6
8.2.3.4	EXTAL Clocking Control for ESAI_2 and ESAI_3 .....	8-6
8.2.4	Port A Signals and Registers .....	8-7
8.2.4.1	Port A Control Registers (PCRA, PCRA1) .....	8-8
8.2.4.2	Port A Direction Register (PRRA, PRRA1).....	8-8
8.2.4.3	Port A Data Register (PDRA, PDRA1) .....	8-8
8.2.5	Port G Signals and Registers .....	8-9
8.2.5.1	Port G Control Registers (PCRG, PCRG1).....	8-10
8.2.5.2	Port G Direction Register (PRRG, PRRG1) .....	8-10
8.2.5.3	Port G Data Register (PDRG, PDRG1) .....	8-10
8.2.6	Timer Event Counter Signals.....	8-10

## Chapter 9

### Enhanced Serial Audio Interface (ESAI, ESAI\_1, ESAI\_2, ESAI\_3)

9.1	ESAI Data and Control Pins .....	9-3
9.1.1	Serial Transmit 0 Data Pin (SDO0) .....	9-3
9.1.2	Serial Transmit 1 Data Pin (SDO1) .....	9-3
9.1.3	Serial Transmit 2/Receive 3 Data Pin (SDO2/SDI3).....	9-3
9.1.4	Serial Transmit 3/Receive 2 Data Pin (SDO3/SDI2).....	9-4
9.1.5	Serial Transmit 4/Receive 1 Data Pin (SDO4/SDI1).....	9-4
9.1.6	Serial Transmit 5/Receive 0 Data Pin (SDO5/SDI0).....	9-4
9.1.7	Receiver Serial Clock (SCKR) .....	9-5
9.1.8	Transmitter Serial Clock (SCKT) .....	9-6
9.1.9	Frame Sync for Receiver (FSR).....	9-7
9.1.10	Frame Sync for Transmitter (FST) .....	9-8
9.1.11	High Frequency Clock for Transmitter (HCKT) .....	9-8
9.1.12	High Frequency Clock for Receiver (HCKR) .....	9-8
9.2	ESAI Programming Model .....	9-9
9.2.1	ESAI Transmitter Clock Control Register (TCCR).....	9-9
9.2.1.1	TCCR Transmit Prescale Modulus Select (TPM7–TPM0)—Bits 7–0.....	9-10
9.2.1.2	TCCR Transmit Prescaler Range (TPSR)—Bit 8.....	9-11
9.2.1.3	TCCR Tx Frame Rate Divider Control (TDC4–TDC0)—Bits 13–9 .....	9-11

# Contents

Paragraph Number	Title	Page Number
9.2.1.4	TCCR Tx High Frequency Clock Divider (TFP3-TFP0)—Bits 17–14.....	9-12
9.2.1.5	TCCR Transmit Clock Polarity (TCKP)—Bit 18.....	9-13
9.2.1.6	TCCR Transmit Frame Sync Polarity (TFSP)—Bit 19 .....	9-13
9.2.1.7	TCCR Transmit High Frequency Clock Polarity (THCKP)—Bit 20.....	9-13
9.2.1.8	TCCR Transmit Clock Source Direction (TCKD)—Bit 21 .....	9-13
9.2.1.9	TCCR Transmit Frame Sync Signal Direction (TFSD)—Bit 22.....	9-13
9.2.1.10	TCCR Transmit High Frequency Clock Direction (THCKD)—Bit 23.....	9-13
9.2.2	ESAI Transmit Control Register (TCR) .....	9-14
9.2.2.1	TCR ESAI Transmit 0 Enable (TE0)—Bit 0.....	9-14
9.2.2.2	TCR ESAI Transmit 1 Enable (TE1)—Bit 1 .....	9-14
9.2.2.3	TCR ESAI Transmit 2 Enable (TE2)—Bit 2.....	9-15
9.2.2.4	TCR ESAI Transmit 3 Enable (TE3)—Bit 3.....	9-15
9.2.2.5	TCR ESAI Transmit 4 Enable (TE4)—Bit 4.....	9-15
9.2.2.6	TCR ESAI Transmit 5 Enable (TE5)—Bit 5.....	9-16
9.2.2.7	TCR Transmit Shift Direction (TSHFD)—Bit 6 .....	9-16
9.2.2.8	TCR Transmit Word Alignment Control (TWA)—Bit 7 .....	9-16
9.2.2.9	TCR Transmit Network Mode Control (TMOD1–TMOD0)—Bits 9–8.....	9-17
9.2.2.10	TCR Tx Slot and Word Length Select (TSWS4–TSWS0)—Bits 14–10.....	9-19
9.2.2.11	TCR Transmit Frame Sync Length (TFSL)—Bit 15.....	9-20
9.2.2.12	TCR Transmit Frame Sync Relative Timing (TFSR)—Bit 16.....	9-22
9.2.2.13	TCR Transmit Zero Padding Control (PADC)—Bit 17 .....	9-22
9.2.2.14	TCR Reserved Bit—Bits 18 .....	9-22
9.2.2.15	TCR Transmit Section Personal Reset (TPR)—Bit 19.....	9-22
9.2.2.16	TCR Transmit Exception Interrupt Enable (TEIE)—Bit 20.....	9-22
9.2.2.17	TCR Transmit Even Slot Data Interrupt Enable (TEDIE)—Bit 21.....	9-23
9.2.2.18	TCR Transmit Interrupt Enable (TIE)—Bit 22 .....	9-23
9.2.2.19	TCR Transmit Last Slot Interrupt Enable (TLIE)—Bit 23.....	9-23
9.2.3	ESAI Receive Clock Control Register (RCCR) .....	9-24
9.2.3.1	RCCR Receiver Prescale Modulus Select (RPM7–RPM0)—Bits 7–0 .....	9-24
9.2.3.2	RCCR Receiver Prescaler Range (RPSR)—Bit 8 .....	9-24
9.2.3.3	RCCR Rx Frame Rate Divider Control (RDC4–RDC0)—Bits 13–9 .....	9-24
9.2.3.4	RCCR Rx High Frequency Clock Divider (RFP3-RFP0)—Bits 17–14.....	9-25
9.2.3.5	RCCR Receiver Clock Polarity (RCKP)—Bit 18 .....	9-25
9.2.3.6	RCCR Receiver Frame Sync Polarity (RFSP)—Bit 19.....	9-25
9.2.3.7	RCCR Receiver High Frequency Clock Polarity (RHCKP)—Bit 20 .....	9-26
9.2.3.8	RCCR Receiver Clock Source Direction (RCKD)—Bit 21 .....	9-26
9.2.3.9	RCCR Receiver Frame Sync Signal Direction (RFSD)—Bit 22 .....	9-26
9.2.3.10	RCCR Receiver High Frequency Clock Direction (RHCKD)—Bit 23 .....	9-27
9.2.4	ESAI Receive Control Register (RCR) .....	9-28
9.2.4.1	RCR ESAI Receiver 0 Enable (RE0)—Bit 0 .....	9-28
9.2.4.2	RCR ESAI Receiver 1 Enable (RE1)—Bit 1 .....	9-28

# Contents

Paragraph Number	Title	Page Number
9.2.4.3	RCR ESAI Receiver 2 Enable (RE2)—Bit 2 .....	9-28
9.2.4.4	RCR ESAI Receiver 3 Enable (RE3)—Bit 3 .....	9-29
9.2.4.5	RCR Reserved Bits—Bits 5–4, 18–17.....	9-29
9.2.4.6	RCR Receiver Shift Direction (RSHFD)—Bit 6.....	9-29
9.2.4.7	RCR Receiver Word Alignment Control (RWA)—Bit 7.....	9-29
9.2.4.8	RCR Receiver Network Mode Control (RMOD1–RMOD0)—Bits 9–8 .....	9-29
9.2.4.9	RCR Receiver Slot and Word Select (RSWS4–RSWS0)—Bits 14–10 .....	9-30
9.2.4.10	RCR Receiver Frame Sync Length (RFSL)—Bit 15 .....	9-31
9.2.4.11	RCR Receiver Frame Sync Relative Timing (RFSR)—Bit 16.....	9-31
9.2.4.12	RCR Receiver Section Personal Reset (RPR)—Bit 19 .....	9-31
9.2.4.13	RCR Receive Exception Interrupt Enable (REIE)—Bit 20.....	9-32
9.2.4.14	RCR Receive Even Slot Data Interrupt Enable (REDIE)—Bit 21 .....	9-32
9.2.4.15	RCR Receive Interrupt Enable (RIE)—Bit 22 .....	9-32
9.2.4.16	RCR Receive Last Slot Interrupt Enable (RLIE)—Bit 23.....	9-32
9.2.5	ESAI Common Control Register (SAICR).....	9-33
9.2.5.1	SAICR Serial Output Flag 0 (OF0)—Bit 0 .....	9-33
9.2.5.2	SAICR Serial Output Flag 1 (OF1)—Bit 1 .....	9-33
9.2.5.3	SAICR Serial Output Flag 2 (OF2)—Bit 2 .....	9-33
9.2.5.4	SAICR Reserved Bits—Bits 5–3, 23–9.....	9-33
9.2.5.5	SAICR Synchronous Mode Selection (SYN)—Bit 6.....	9-33
9.2.5.6	SAICR Transmit External Buffer Enable (TEBE)—Bit 7.....	9-34
9.2.5.7	SAICR Alignment Control (ALC)—Bit 8.....	9-34
9.2.6	ESAI Status Register (SAISR) .....	9-36
9.2.6.1	SAISR Serial Input Flag 0 (IF0)—Bit 0 .....	9-36
9.2.6.2	SAISR Serial Input Flag 1 (IF1)—Bit 1 .....	9-36
9.2.6.3	SAISR Serial Input Flag 2 (IF2)—Bit 2.....	9-36
9.2.6.4	SAISR Reserved Bits—Bits 5–3, 12–11, 23–18 .....	9-36
9.2.6.5	SAISR Receive Frame Sync Flag (RFS)—Bit 6 .....	9-37
9.2.6.6	SAISR Receiver Overrun Error Flag (ROE)—Bit 7 .....	9-37
9.2.6.7	SAISR Receive Data Register Full (RDF)—Bit 8 .....	9-37
9.2.6.8	SAISR Receive Even-Data Register Full (REDF)—Bit 9 .....	9-37
9.2.6.9	SAISR Receive Odd-Data Register Full (RODF)—Bit 10 .....	9-37
9.2.6.10	SAISR Transmit Frame Sync Flag (TFS)—Bit 13.....	9-38
9.2.6.11	SAISR Transmit Underrun Error Flag (TUE)—Bit 14 .....	9-38
9.2.6.12	SAISR Transmit Data Register Empty (TDE)—Bit 15.....	9-38
9.2.6.13	SAISR Transmit Even-Data Register Empty (TEDE)—Bit 16.....	9-38
9.2.6.14	SAISR Transmit Odd-Data Register Empty (TODE)—Bit 17.....	9-39
9.2.7	ESAI Receive Shift Registers .....	9-42
9.2.8	ESAI Receive Data Registers (RX3, RX2, RX1, RX0) .....	9-42
9.2.9	ESAI Transmit Shift Registers.....	9-42
9.2.10	ESAI Transmit Data Registers (TX5, TX4, TX3, TX2, TX1, TX0) .....	9-42



# Contents

Paragraph Number	Title	Page Number
9.2.11	ESAI Time Slot Register (TSR) .....	9-42
9.2.12	Transmit Slot Mask Registers (TSMA, TSMB) .....	9-42
9.2.13	Receive Slot Mask Registers (RSMA, RSMB) .....	9-44
9.3	Operating Modes .....	9-45
9.3.1	ESAI After Reset .....	9-45
9.3.2	ESAI Initialization .....	9-45
9.3.3	ESAI Interrupt Requests .....	9-46
9.3.4	Operating Modes—Normal, Network and On-Demand .....	9-48
9.3.4.1	Normal/Network/On-Demand Mode Selection .....	9-48
9.3.4.2	Synchronous/Asynchronous Operating Modes .....	9-48
9.3.4.3	Frame Sync Selection .....	9-49
9.3.4.4	Shift Direction Selection .....	9-49
9.3.5	Serial I/O Flags .....	9-50
9.4	GPIO—Pins and Registers .....	9-50
9.4.1	Port C Control Register (PCRC) .....	9-51
9.4.2	Port C Direction Register (PRRC) .....	9-51
9.4.3	Port C Data Register (PDRC) .....	9-51
9.5	ESAI Initialization Examples .....	9-52
9.5.1	Initializing the ESAI Using Individual Reset .....	9-52
9.5.2	Initializing Only the ESAI Transmitter Section .....	9-52
9.5.3	Initializing Only the ESAI Receiver Section .....	9-53
9.6	ESAI/ESAI_2 and ESAI_1/ESAI_3 Pin Switch .....	9-53
9.7	Internal Clock Connections Between ESAI and ESAI_1, ESAI_2 and ESAI_3 .....	9-53

## Chapter 10 Serial Host Interface (SHI, SHI\_1)

10.1	Introduction .....	10-1
10.2	Serial Host Interface Internal Architecture .....	10-2
10.3	SHI Clock Generator .....	10-3
10.3.1	Serial Host Interface Programming Model .....	10-3
10.3.2	SHI Input/Output Shift Register (IOSR)—Host Side .....	10-5
10.3.3	SHI Host Transmit Data Register (HTX)—DSP Side .....	10-6
10.3.4	SHI Host Receive Data FIFO (HRX)—DSP Side .....	10-6
10.3.5	SHI Slave Address Register (HSAR)—DSP Side .....	10-7
10.3.5.1	HSAR Reserved Bits—Bits 19, 17–0 .....	10-7
10.3.6	HSAR I <sup>2</sup> C Slave Address (HA[6:3], HA1)—Bits 23–20, 18 .....	10-7
10.3.7	SHI Clock Control Register (HCKR)—DSP Side .....	10-7
10.3.7.1	Clock Phase and Polarity (CPHA and CPOL)—Bits 1–0 .....	10-7
10.3.7.2	HCKR Prescaler Rate Select (HRS)—Bit 2 .....	10-9
10.3.7.3	HCKR Divider Modulus Select (HDM[7:0])—Bits 10–3 .....	10-9

# Contents

Paragraph Number	Title	Page Number
10.3.7.4	HCKR Reserved Bits—Bits 23–14, 11.....	10-9
10.3.7.5	HCKR HFM[1:0] Bits—Bits 13–12.....	10-10
10.3.8	SHI Control/Status Register (HCSR)—DSP Side .....	10-10
10.3.8.1	HCSR Host Enable (HEN)—Bit 0 .....	10-11
10.3.8.1.1	SHI Individual Reset .....	10-11
10.3.8.2	HCSR I <sup>2</sup> C/SPI Selection (HI <sup>2</sup> C)—Bit 1 .....	10-11
10.3.8.3	HCSR Serial Host Interface Mode (HM[1:0])—Bits 3–2 .....	10-11
10.3.8.4	HCSR I <sup>2</sup> C Clock Freeze (HCKFR)—Bit 4 .....	10-12
10.3.8.5	HCSR FIFO-Enable Control (HFIFO)—Bit 5 .....	10-12
10.3.8.6	HCSR Master Mode (HMST)—Bit 6.....	10-12
10.3.8.7	HCSR Host-Request Enable (HRQE[1:0])—Bits 8–7 .....	10-13
10.3.8.8	HCSR Idle (HIDLE)—Bit 9 .....	10-13
10.3.8.9	HCSR Bus-Error Interrupt Enable (HBIE)—Bit 10.....	10-14
10.3.8.10	HCSR Transmit-Interrupt Enable (HTIE)—Bit 11.....	10-14
10.3.8.11	HCSR Receive Interrupt Enable (HRIE[1:0])—Bits 13–12.....	10-15
10.3.8.12	HCSR Host Transmit Underrun Error (HTUE)—Bit 14.....	10-15
10.3.8.13	HCSR Host Transmit Data Empty (HTDE)—Bit 15.....	10-16
10.3.8.14	HCSR Reserved Bits—Bits 23, 18 and 16 .....	10-16
10.3.8.15	Host Receive FIFO Not Empty (HRNE)—Bit 17 .....	10-16
10.3.8.16	Host Receive FIFO Full (HRFF)—Bit 19 .....	10-16
10.3.8.17	Host Receive Overrun Error (HROE)—Bit 20.....	10-16
10.3.8.18	Host Bus Error (HBER)—Bit 21 .....	10-17
10.3.8.19	HCSR Host Busy (HBUSY)—Bit 22 .....	10-17
10.4	Characteristics of the SPI Bus .....	10-17
10.5	Characteristics of the I <sup>2</sup> C Bus.....	10-18
10.5.1	Overview of I <sup>2</sup> C Bus .....	10-18
10.5.2	I <sup>2</sup> C Data Transfer Formats.....	10-20
10.6	SHI Programming Considerations .....	10-20
10.6.1	SPI Slave Mode .....	10-21
10.6.2	SPI Master Mode .....	10-22
10.6.3	I <sup>2</sup> C Slave Mode.....	10-23
10.6.3.1	Receive Data in I <sup>2</sup> C Slave Mode .....	10-23
10.6.3.2	Transmit Data In I <sup>2</sup> C Slave Mode .....	10-24
10.6.4	I <sup>2</sup> C Master Mode .....	10-25
10.6.4.1	Receive Data in I <sup>2</sup> C Master Mode.....	10-26
10.6.4.2	Transmit Data In I <sup>2</sup> C Master Mode .....	10-26
10.6.5	SHI Operation During DSP Stop .....	10-27
10.7	SHI Pin-Outs for Device Packages .....	10-27
10.7.1	SHI Pin-Outs for Small Pin Count Packages.....	10-28

# Contents

Paragraph Number	Title	Page Number
<b>Chapter 11</b>		
<b>Triple Timer Module (TEC, TEC_1)</b>		
11.1	Introduction.....	11-1
11.1.1	Overview.....	11-1
11.1.2	Triple Timer Module Block Diagram .....	11-2
11.2	Individual Timer Block Diagram .....	11-2
11.3	Operation .....	11-3
11.3.1	Timer After Reset .....	11-3
11.3.2	Timer Initialization .....	11-4
11.3.3	Timer Exceptions .....	11-4
11.4	Operating Mode .....	11-5
11.4.1	Timer GPIO (Mode 0) .....	11-5
11.4.2	Reserved Modes.....	11-7
11.4.3	Special Cases .....	11-7
11.4.4	DMA Trigger .....	11-7
11.5	Triple Timer Module Programming Model .....	11-7
11.5.1	Prescaler Counter.....	11-7
11.5.2	Timer Prescaler Load Register (TPLR) .....	11-9
11.5.3	Timer Prescaler Count Register (TPCR) .....	11-10
11.5.4	Timer Control/Status Register (TCSR).....	11-10
11.5.5	Timer Load Register (TLR) .....	11-13
11.5.6	Timer Compare Register (TCPR).....	11-13
11.5.7	Timer Count Register (TCR) .....	11-13
<b>Chapter 12</b>		
<b>Watchdog Timer (WDT, WDT_1)</b>		
12.1	Introduction.....	12-1
12.2	WDT Pin-Outs for Different Device Packages.....	12-1
12.3	WDT Registers .....	12-3
12.3.1	Watchdog Control Register (WCR).....	12-3
12.3.2	Watchdog Counter and WCNTR Register.....	12-4
12.3.3	Watchdog Modulus Register (WMR) .....	12-4
12.3.4	Watchdog Service Register (WSR).....	12-5
12.4	Watchdog Operating Modes .....	12-5
12.4.1	Wait Mode.....	12-5
12.4.2	Debug Mode .....	12-6
12.4.3	Stop Mode.....	12-6

# Contents

Paragraph Number	Title	Page Number
<b>Chapter 13</b>		
<b>Inter-Core Communication (ICC)</b>		
13.1	Introduction.....	13-1
13.1.1	Overview.....	13-1
13.1.2	Features.....	13-3
13.2	Memory Map and Register Definition.....	13-3
13.2.1	Memory Map .....	13-3
13.2.2	Register Descriptions.....	13-5
13.2.2.1	ICDR1 (ICC Data Register 1).....	13-5
13.2.2.2	ICCR1 (ICC Control Register 1).....	13-6
13.2.2.3	ICDR2 (ICC Data Register 2).....	13-6
13.2.2.4	ICCR2 (ICC Control Register 2).....	13-7
13.2.2.5	ICDR3 (ICC Data Register 3).....	13-7
13.2.2.6	ICCR3 (ICC Control Register 3).....	13-8
13.2.2.7	ICAR3 (ICC Acknowledge Register 3).....	13-8
13.2.2.8	ICDR4 (ICC Data Register 4).....	13-9
13.2.2.9	ICCR4 (ICC Control Register 4).....	13-9
13.2.2.10	ICAR4 (ICC Acknowledge Register 4).....	13-10
13.2.2.11	ICPR1 (ICC Poll Register 1).....	13-10
13.2.2.12	ICPR2 (ICC Poll Register 2).....	13-11
13.3	Programming Model.....	13-11
13.3.1	Inter-Core Maskable Interrupts.....	13-12
13.3.2	Inter-Core Non-Maskable Interrupts .....	13-13
13.3.3	Polling.....	13-13
13.3.4	Error Interrupts .....	13-14
13.3.5	Reset.....	13-15

## Chapter 14 Shared Bus Arbiter

14.1	Introduction.....	14-1
14.1.1	Overview.....	14-1
14.1.2	Features.....	14-1
14.2	Memory Map and Register Definition.....	14-2
14.3	Functional Description.....	14-2
14.3.1	Shared Bus Arbitration .....	14-2
14.3.1.1	Configuring the Arbitration Method.....	14-2
14.3.1.2	Shared Bus Master 0/1 Always Has Priority Methods .....	14-3
14.3.1.3	Round-Robin Arbitration Method .....	14-4
14.3.1.4	Cores/DMA Continuous Access Arbitration .....	14-5

# Contents

Paragraph Number	Title	Page Number
<b>Chapter 15</b>		
<b>Shared Memory</b>		
15.1	Overview .....	15-1
15.2	Block Diagram .....	15-1
<b>Chapter 16</b>		
<b>Shared Peripheral Bus</b>		
16.1	Introduction .....	16-1
16.1.1	Features .....	16-1
16.2	Memory Map and Register Definition .....	16-2
16.3	Functional Description .....	16-2
<b>Chapter 17</b>		
<b>EMC Burst Buffer</b>		
17.1	Introduction .....	17-1
17.1.1	Overview .....	17-2
17.1.2	Features .....	17-3
17.2	Memory Map and Register Definition .....	17-3
17.3	Functional Description .....	17-3
17.3.1	Burst Control .....	17-3
17.3.2	Read Access .....	17-5
17.3.2.1	Single Read .....	17-5
17.3.2.2	Burst Read .....	17-6
17.3.3	Write Access .....	17-6
17.3.3.1	Single Write .....	17-7
17.3.3.2	Burst Write .....	17-8
<b>Chapter 18</b>		
<b>S/PDIF—Sony/Philips Digital Interface</b>		
18.1	Introduction .....	18-1
18.1.1	Features .....	18-3
18.1.2	External Signal Description .....	18-3
18.1.3	Memory Map .....	18-4
18.2	Register Descriptions .....	18-5
18.2.1	S/PDIF Configuration Register (SCR) .....	18-5
18.2.2	CDText Control Register (SRCD) .....	18-6
18.2.3	PhaseConfig Register (SRPC) .....	18-7

# Contents

Paragraph Number	Title	Page Number
18.2.4	Interrupt Registers (SIE, SIS, SIC).....	18-8
18.2.5	S/PDIF Reception Registers .....	18-10
18.2.6	S/PDIF Transmission Registers .....	18-12
18.2.7	S/PDIF FreqMeas Register (SRFM).....	18-15
18.2.8	SPDIFTxClk Register (STC).....	18-15
18.3	S/PDIF Receiver .....	18-16
18.3.1	Audio Data Reception.....	18-16
18.3.2	Channel Status Reception .....	18-19
18.3.3	User Bit Reception.....	18-19
18.3.4	Validity Flag Reception .....	18-21
18.3.5	S/PDIF Receiver Interrupt Exception Definition.....	18-21
18.3.6	Standards Compliance .....	18-22
18.3.7	S/PDIF PLOCK Detection and Rxcclk Output .....	18-22
18.3.8	Measuring the Frequency of SPDIF_RcvClk.....	18-22
18.4	S/PDIF Transmitter .....	18-23
18.4.1	Audio Data Transmission .....	18-23
18.4.2	Channel Status Transmission .....	18-24
18.4.3	Validity Flag Transmission .....	18-24

## Chapter 19 Asynchronous Sample Rate Converter

19.1	Introduction.....	19-1
19.1.1	Overview.....	19-2
19.1.2	Features.....	19-3
19.1.3	Modes of Operation .....	19-4
19.1.3.1	Data Transfer Schemes .....	19-4
19.1.3.1.1	Data Input Modes .....	19-4
19.1.3.1.2	Data Output Modes.....	19-5
19.2	Memory Map and Register Definitions .....	19-5
19.2.1	Memory Map .....	19-5
19.2.2	Register Descriptions.....	19-7
19.2.2.1	ASRC Control Register (ASRCTR) .....	19-7
19.2.2.2	Interrupt Enable Register and Mask Register (ASRIER, ASRIEM).....	19-9
19.2.2.3	Channel Number Configuration Register (ASRCNCR).....	19-11
19.2.2.4	Filter Configuration Status Register (ASRCFG).....	19-12
19.2.2.5	ASRC Clock Source Register (ASRC SR).....	19-14
19.2.2.6	ASRC Clock Divider Registers (ASRC DR1, ASRC DR2).....	19-16
19.2.2.7	ASRC Status Register (ASRSTR) .....	19-18
19.2.2.8	ASRC Debug Control Register (ASRDCR).....	19-21
19.2.2.9	Parameter Registers (ASRPM1–ASRPM5) .....	19-23

# Contents

Paragraph Number	Title	Page Number
19.2.2.10	ASRC Task Queue FIFO Register (ASRTFR1) .....	19-23
19.2.2.11	Channel Counter Register (ASRCCR) .....	19-24
19.2.2.12	ASRC Data Input and Output Registers .....	19-24
19.2.2.12.1	Ideal Ratio Registers for Pair A, High/Low Part (ASRIDRHA, ASRIDRLA) .....	19-24
19.2.2.12.2	Ideal Ratio Registers for Pair B, High/Low Part (ASRIDRHB, ASRIDRLB) .....	19-25
19.2.2.12.3	Ideal Ratio Registers for Pair C, High/Low Part (ASRIDRHC, ASRIDRLC) .....	19-25
19.2.2.12.4	ASRC 76 kHz Period Register in Terms of Master Clock (ASR76K) .....	19-26
19.2.2.12.5	ASRC 56 kHz Period Register in Terms of Master Clock (ASR56K) .....	19-26
19.3	Interrupts .....	19-26
19.4	DMA Requests .....	19-27
19.5	Functional Description .....	19-28
19.5.1	Algorithm Description .....	19-28
19.5.1.1	Signal Processing Flow .....	19-28
19.5.1.2	Miscellaneous Topics .....	19-30
19.5.1.2.1	Support of Physical Clocks .....	19-30
19.5.1.2.2	Physical Clock Source Selector and Divider .....	19-31
19.5.1.2.3	Configure ASRCDR1/ASRCDR2 .....	19-33

## Chapter 20 Chip Configuration Module

20.1	Introduction .....	20-1
20.1.1	Modes of Operation .....	20-1
20.2	Memory Map and Register Definition .....	20-1
20.2.1	Memory Map .....	20-1
20.2.2	Register Descriptions .....	20-3
20.2.2.1	Reserved Register .....	20-3
20.2.2.2	EMBC (External Memory Burst Control) Register .....	20-3
20.2.2.3	EMBC Register .....	20-3
20.2.2.4	LPSC (EMC PLL Status and Control) Register .....	20-5
20.2.2.5	Pin Mux Control Register (PMC) .....	20-7
20.2.2.6	ESAI Pin Switch Control Register (EPSC) .....	20-8
20.2.2.7	ONCE Debug and Burst Control Register (ODBC) .....	20-10
20.2.2.8	Peripheral Soft Reset Control Register (PSRC) .....	20-11
20.2.2.9	Arbiter Control Register (ARCR) .....	20-12
20.3	Programming Model .....	20-13
20.3.1	ESAI/ESAI_1/ESAI_2/ESAI_3 Pin-Switching and Internal Connections .....	20-13
20.3.2	ESAI_2 Data and SPDIF Data Pin Mux .....	20-14

# Contents

Paragraph Number	Title	Page Number
20.3.3	Soft Reset.....	20-15
20.3.4	Reset.....	20-15
20.3.5	ESAI Pin Switch and Internal Clock Connections .....	20-15
20.3.5.1	ESAI and ESAI_1 Internal Clocks Connections .....	20-16
20.3.5.2	ESAI_2 and ESAI_3 Internal Clock Connections.....	20-20

## Chapter 21 External Memory Controller (EMC)

21.1	Introduction.....	21-1
21.1.1	Features.....	21-2
21.2	External Signal Descriptions .....	21-3
21.2.1	Detailed Signal Descriptions .....	21-4
21.3	Memory Map and Register Definition.....	21-7
21.3.1	Memory Map .....	21-7
21.3.2	Register Descriptions.....	21-9
21.3.2.1	Base Registers (BR0–BR7) .....	21-9
21.3.2.2	Option Registers (OR0–OR7).....	21-11
21.3.2.2.1	Address Mask .....	21-11
21.3.2.2.2	Option Registers (ORx)—GPCM Mode .....	21-12
21.3.2.2.3	Option Registers (ORx)—UPM Mode .....	21-15
21.3.2.2.4	Option Registers (ORx)—SDRAM Mode .....	21-17
21.3.2.3	UPM Memory Address Register (MAR).....	21-19
21.3.2.4	UPM Mode Registers (MxMR) .....	21-20
21.3.2.5	Memory Refresh Timer Prescaler Register (MRTPR) .....	21-23
21.3.2.6	UPM Data Register (MDR).....	21-24
21.3.2.7	SDRAM Machine Mode Register (SDMR) .....	21-25
21.3.2.8	UPM Refresh Timer (URT) .....	21-27
21.3.2.9	SDRAM Refresh Timer (SRT) .....	21-28
21.3.2.10	Transfer Error Status Register (TESR).....	21-29
21.3.2.11	Transfer Error Check Disable Register (TEDR).....	21-30
21.3.2.12	Transfer Error Interrupt Enable Register (TEIR) .....	21-31
21.3.2.13	Transfer Error Attributes Register (TEATR) .....	21-32
21.3.2.14	Transfer Error Address Register (TEAR).....	21-33
21.3.2.15	Local Bus Configuration Register (LBCR) .....	21-34
21.3.2.16	Clock Ratio Register (LCRR).....	21-35
21.4	Functional Description.....	21-36
21.4.1	Basic Architecture.....	21-38
21.4.1.1	Address and Address Space Checking .....	21-38
21.4.1.2	External Address Latch Enable Signal (LALE) .....	21-38
21.4.1.3	Data Transfer Acknowledge (TA) .....	21-38



# Contents

Paragraph Number	Title	Page Number
21.4.1.4	Data Buffer Control (LBCTL).....	21-39
21.4.1.5	Bus Monitor .....	21-39
21.4.2	General-Purpose Chip-Select Machine (GPCM).....	21-40
21.4.2.1	Timing Configuration .....	21-41
21.4.2.2	Chip-Select Assertion Timing .....	21-45
21.4.2.2.1	Programmable Wait State Configuration.....	21-45
21.4.2.2.2	Chip-Select and Write Enable Negation Timing .....	21-46
21.4.2.2.3	Relaxed Timing .....	21-46
21.4.2.2.4	Output Enable (LOE) Timing.....	21-49
21.4.2.2.5	Extended Hold Time on Read Accesses.....	21-49
21.4.2.3	External Access Termination (LGTA) .....	21-50
21.4.2.4	Boot Chip-Select Operation.....	21-51
21.4.3	SDRAM Machine .....	21-52
21.4.3.1	Supported SDRAM Configurations.....	21-52
21.4.3.2	SDRAM Power-On Initialization .....	21-53
21.4.3.3	JEDEC-Standard SDRAM Interface Commands .....	21-53
21.4.3.4	Page Hit Checking .....	21-54
21.4.3.5	Page Management.....	21-55
21.4.3.6	SDRAM Address Multiplexing .....	21-55
21.4.3.7	SDRAM Device-Specific Parameters.....	21-56
21.4.3.7.1	Precharge-to-Activate Interval.....	21-56
21.4.3.7.2	Activate-to-Read/Write Interval .....	21-56
21.4.3.7.3	Column Address to First-Data-Out—CAS Latency .....	21-57
21.4.3.7.4	Last-Data-In-to-Precharge—Write Recovery.....	21-57
21.4.3.7.5	Refresh Recovery Interval (RFRC) .....	21-58
21.4.3.7.6	External Address and Command Buffers (BUFCMD).....	21-58
21.4.3.8	SDRAM Interface Timing .....	21-59
21.4.3.9	SDRAM Read/Write Transactions.....	21-61
21.4.3.10	SDRAM MODE-SET Command Timing.....	21-62
21.4.3.11	SDRAM Refresh.....	21-62
21.4.3.11.1	SDRAM Refresh Timing .....	21-62
21.4.4	User-Programmable Machines (UPMs).....	21-63
21.4.4.1	UPM Requests .....	21-64
21.4.4.1.1	Memory Access Requests.....	21-65
21.4.4.1.2	UPM Refresh Timer Requests .....	21-65
21.4.4.1.3	Software Requests—RUN Command .....	21-66
21.4.4.1.4	Exception Requests.....	21-66
21.4.4.2	Programming the UPMs .....	21-66
21.4.4.3	UPM Signal Timing.....	21-68
21.4.4.4	The RAM Array.....	21-69
21.4.4.4.1	RAM Words.....	21-69

# Contents

Paragraph Number	Title	Page Number
21.4.4.4.2	Chip-Select Signal Timing (CST <sub>n</sub> ) .....	21-73
21.4.4.4.3	General-Purpose Signals (GxTx, GOx) .....	21-73
21.4.4.4.4	Loop Control (LOOP) .....	21-73
21.4.4.4.5	Repeat Execution of Current RAM Word (REDO) .....	21-74
21.4.4.4.6	Address Multiplexing (AMX) .....	21-74
21.4.4.4.7	Data Valid and Data Sample Control (UTA) .....	21-75
21.4.4.4.8	LGPL[5:0] Signal Negation (LAST) .....	21-75
21.4.4.4.9	The Wait Mechanism (WAEN) .....	21-75
21.4.4.5	Synchronous Sampling of UPWAIT for Early Transfer Acknowledge .....	21-77
21.4.4.6	Extended Hold Time on Read Accesses .....	21-77
21.4.4.7	Memory System Interface Example Using UPM .....	21-77
21.5	Application Information .....	21-83
21.5.1	Interfacing to Peripherals .....	21-83
21.5.1.1	Multiplexed Address and Data Bus and Unmultiplexed Address Signals .....	21-83
21.5.1.2	Peripheral Hierarchy on the External Memory Controller .....	21-84
21.5.1.3	Peripheral Hierarchy on External Memory Controller for Very High Bus Speeds .....	21-85
21.5.1.4	GPCM Timing .....	21-85
21.5.2	Bus Turnaround .....	21-86
21.5.2.1	Address Phase after Previous Read .....	21-86
21.5.2.2	Read Data Phase after Address Phase .....	21-86
21.5.2.3	UPM Cycles with Additional Address Phases .....	21-87
21.5.3	Interfacing to SDRAM .....	21-87
21.5.3.1	Basic SDRAM Capabilities of the External Memory Controller .....	21-87
21.5.3.2	Maximum Amount of SDRAM Supported .....	21-88
21.5.3.3	Example of SDRAM Usage .....	21-88
21.5.3.3.1	Maximum Row Number Due to Bank Select MUX .....	21-88
21.5.3.3.2	Bank Select Signals .....	21-89
21.5.3.3.3	SDRAM of 256-Mbit .....	21-89
21.5.3.3.4	Power-Down Mode .....	21-90
21.5.3.3.5	Self-Refresh .....	21-91
21.5.3.3.6	SDRAM Timing .....	21-91

## Chapter 22 JTAG Controller

22.1	Overview .....	22-1
22.2	Features .....	22-2
22.3	External Signal Descriptions .....	22-4

# Contents

Paragraph Number	Title	Page Number
<b>Appendix A Revision History</b>		
A.1	Changes From Revision 0 to Revision 1 .....	A-1

# Contents

**Paragraph  
Number**

**Title**

**Page  
Number**

# Figures

Figure Number	Title	Page Number
1-1	DSP56724/DSP56725 Block Diagram .....	1-3
2-1	DSP56725 80-Pin Package Pin-Out.....	2-4
2-2	DSP56724 144-Pin Package Pin-Out.....	2-5
3-1	DSP56724 and DSP56725 Memory Address Allocation .....	3-1
4-1	DSP56300 Core in DSP56724/DSP56725.....	4-2
5-1	Operating Mode Register (OMR) .....	5-1
5-2	Status Register (SR) .....	5-4
5-3	Core-0 Interrupt Priority Register P.....	5-10
5-4	Core-0 Interrupt Priority Register C .....	5-10
5-5	Core-0 Interrupt Priority Register C1 .....	5-10
5-6	Core-0 Interrupt Priority Register P1 .....	5-11
5-7	Core-1 Interrupt Priority Register P.....	5-11
5-8	Core-1 Interrupt Priority Register C .....	5-11
5-9	Core-1 Interrupt Priority Register C1 .....	5-12
5-10	Core-1 Interrupt Priority Register P1 .....	5-12
6-1	Core Integration Module Block Diagram .....	6-1
6-2	Legend.....	6-2
6-3	Chip ID Register for DSP56724 Core-0 .....	6-4
6-4	Chip ID Register for DSP56724 Core-1 .....	6-4
6-5	Chip ID Register for DSP56725 Core-0 .....	6-4
6-6	Chip ID Register for DSP56725 Core-1 .....	6-4
6-7	DMA Stall Register (DMAS).....	6-5
6-8	OnCE GDB Register (OGDB).....	6-5
7-1	CGM Function Diagram .....	7-2
7-2	Internal PLL Block Diagram.....	7-5
7-3	Shared Peripheral Clock Enable Register (SPENA).....	7-8
7-4	PLL Control Register (PCTL).....	7-8
7-5	ASRC Clock Dividing Control Register (ASCDR).....	7-11
9-1	ESAI Block Diagram .....	9-2
9-2	TCCR Fields .....	9-9
9-3	ESAI Clock Generator Functional Block Diagram.....	9-10
9-4	ESAI Frame Sync Generator Functional Block Diagram .....	9-12
9-5	TCR Register.....	9-14
9-6	Normal and Network Operation.....	9-18
9-7	Frame Length Selection .....	9-21
9-8	RCCR Register.....	9-24
9-9	ESAI Receive Control Register .....	9-28
9-10	SAICR Register.....	9-33
9-11	SAICR SYN Bit Operation .....	9-35
9-12	SAISR Register .....	9-36
9-13	ESAI Data Path Programming Model ([R/T]SHFD=0).....	9-40

# Figures

Figure Number	Title	Page Number
9-14	ESAI Data Path Programming Model ([R/T]SHFD=1).....	9-41
9-15	TSMA Register .....	9-43
9-16	TSMB Register .....	9-43
9-17	RSMA Register .....	9-44
9-18	RSMB Register .....	9-44
9-19	PCRC Register .....	9-51
9-20	PRRC Register .....	9-51
9-21	PDRC Registers .....	9-52
10-1	Serial Host Interface Block Diagram .....	10-2
10-2	SHI Clock Generator.....	10-3
10-3	SHI Programming Model—Host Side .....	10-3
10-4	SHI Programming Model—DSP Side .....	10-4
10-5	SHI I/O Shift Register (IOSR).....	10-6
10-6	SPI Data-To-Clock Timing Diagram .....	10-8
10-7	I <sup>2</sup> C Bit Transfer.....	10-18
10-8	I <sup>2</sup> C Start and Stop Events.....	10-19
10-9	Acknowledgment on the I <sup>2</sup> C Bus.....	10-19
10-10	I <sup>2</sup> C Bus Protocol For Host Write Cycle.....	10-20
10-11	I <sup>2</sup> C Bus Protocol For Host Read Cycle.....	10-20
10-12	SHI/SHI_1 Pin-Outs in Small Pin Count Packages .....	10-28
11-1	Triple Timer Block Diagram.....	11-2
11-2	Individual Timer Block Diagram .....	11-3
11-3	Timer Mode (TRM = 1) .....	11-6
11-4	Timer Mode (TRM = 0).....	11-6
11-5	Timer Module Programmer's Model .....	11-8
11-6	Timer Prescaler Load Register .....	11-9
11-7	Timer Prescaler Count Register (TPCR).....	11-10
11-8	Timer Control/Status Register (TCSR).....	11-10
12-1	Watchdog Timer Block Diagram .....	12-2
12-2	WCR Register .....	12-3
12-3	WCNTR Register .....	12-4
12-4	WMR Register .....	12-4
12-5	WSR Register.....	12-5
13-1	ICC Block Diagram .....	13-1
13-2	ICC Registers Block.....	13-2
13-3	ICDR1 Write Data Register .....	13-5
13-4	ICCR1 Control Register 1.....	13-6
13-5	ICDR2 Data Register .....	13-6
13-6	ICCR2 Control Register.....	13-7
13-7	ICDR3 Data Register .....	13-7
13-8	ICCR3 Control Register.....	13-8

# Figures

Figure Number	Title	Page Number
13-9	ICC ICAR3 Control Register .....	13-8
13-10	ICDR4 Data Register .....	13-9
13-11	ICCR4 Control Register .....	13-9
13-12	ICAR4 Control Register .....	13-10
13-13	ICC_ICPR1 Polling Register .....	13-10
13-14	ICPR2 Polling Register .....	13-11
13-15	Maskable Interrupts .....	13-12
13-16	Non-Maskable Interrupts .....	13-13
13-17	Poll Data Register .....	13-13
13-18	Error Interrupt Generation .....	13-14
13-19	Error Interrupt Status Register (EISR) .....	13-14
14-1	Shared Bus Arbiter .....	14-1
14-2	Shared Bus Master 0 Always has Priority .....	14-3
14-3	Shared Bust Master 1 Always has Priority .....	14-4
14-4	Round-Robin Arbitration Method .....	14-5
15-1	Shared Memory .....	15-1
16-1	Shared Peripheral Bus in DSP56724/DSP56725 .....	16-1
17-1	EMC Burst Buffer in DSP56724 .....	17-2
17-2	Read Access Flow .....	17-5
17-3	Write Access Flow .....	17-7
18-1	S/PDIF Transceiver Data Interface Block Diagram .....	18-1
18-2	S/PDIF Transceiver Clock Diagram .....	18-2
18-3	S/PDIF Configuration Register (SCR) .....	18-5
18-4	CDText Control Register (SRCD) .....	18-6
18-5	PhaseConfig Register (SRPC) .....	18-7
18-6	InterruptEn Register (SIE) .....	18-8
18-7	InterruptStat Register (SIS) .....	18-8
18-8	InterruptClear Register (SIC) .....	18-9
18-9	SPDIFRcvLeft Register (SRL) .....	18-10
18-10	SPDIFRcvRight Register (SRR) .....	18-10
18-11	SPDIFRxChannel_h Register (SRCSH) .....	18-11
18-12	SPDIFRxChannel_l Register (SRCSL) .....	18-11
18-13	S/PDIF QchannelRcv Register (SRQ) .....	18-11
18-14	S/PDIF QchannelRcv Register (SRQ) .....	18-12
18-15	SPDIFTxLeft Register (STL) .....	18-12
18-16	SPDIFTxRight Register (STR) .....	18-13
18-17	SPDIFTxChannelCons_h Register (STCSCH) .....	18-13
18-18	SPDIFTxChannelCons_l Register (STCSCL) Register .....	18-13
18-19	SPDIFTxChannelProf_h Register (STCSPH) .....	18-14
18-20	SPDIFTxChannelProf_l Register (STCSPL) .....	18-14
18-21	S/PDIF FreqMeas Register (SRFM) .....	18-15

# Figures

Figure Number	Title	Page Number
18-22	SPDIFTxClk Register (STC) .....	18-15
18-23	FIFO Auto-Resync Controller State Machine.....	18-17
19-1	ASRC Connections .....	19-1
19-2	ASRC Block Diagram.....	19-2
19-3	ASRC Control Register (ASRCTR) .....	19-7
19-4	Interrupt Enable Register (ASRIER) .....	19-9
19-5	Interrupt Enable Mask Register (ASRIEM).....	19-10
19-6	Channel Number Configuration Register (ASRCNCR) .....	19-11
19-7	Filter Configuration Status Register (ASRCFG) .....	19-12
19-8	Clock Source Register (ASRCSSR).....	19-14
19-9	Clock Divider Register-1 (ASRCDR1).....	19-16
19-10	Clock Divider Register-2 (ASRCDR2).....	19-17
19-11	Status Register (ASRSTR).....	19-18
19-12	Debug Control Register (ASRDCR).....	19-21
19-13	ASRC Task Queue FIFO Register 1 (ASRTFR1).....	19-23
19-14	Channel Counter Register (ASRCCR).....	19-24
19-15	Ideal Ratio Register for Pair A, High Part (ASRIDRHA) .....	19-25
19-16	Ideal Ratio Register for Pair A, Low Part (ASRIDRLA).....	19-25
19-17	Ideal Ratio Register for Pair B, High Part (ASRIDRHB).....	19-25
19-18	Ideal Ratio Register for Pair B, Low Part (ASRIDRLB).....	19-25
19-19	Ideal Ratio Register for Pair C, High Part (ASRIDRHC).....	19-25
19-20	Ideal Ratio Register for Pair C, Low Part (ASRIDRLC).....	19-26
19-21	ASRC 76 kHz Period Register (ASR76K) .....	19-26
19-22	ASRC 56kHz Period Register (ASR56K).....	19-26
19-23	Signal Processing Configurations for ASRC .....	19-28
19-24	Clock Source Selector and Divider .....	19-32
20-1	Reserved Control Register .....	20-3
20-2	External Memory Burst Buffer Control Register—DSP56724 .....	20-3
20-3	EMBC Register, DSP56725 .....	20-4
20-4	EMC PLL Status and Control Register (PSC)—DSP56724.....	20-5
20-5	LPSC Register—DSP56725 .....	20-6
20-6	Pin Mux Control Register .....	20-7
20-7	ESAI Pin Switch Control Register (EPSC).....	20-8
20-8	ONCE Debug and Burst Control Register (ODBC) .....	20-10
20-9	Peripheral Soft Reset Control Register (PSRC).....	20-11
20-10	Arbiter Control Register (ARCR).....	20-12
20-11	ESAI Internal Clock Connections and Pin-Switching .....	20-13
20-12	SPDIFIN1 and ESAI Pin Mux .....	20-14
20-13	SPDIFOUT1 and ESAI_2 Pin Mux .....	20-14
20-14	ESAI Internal Clock Control Register (EICCR for Core-0, EICCR_1 for Core-1) .....	20-15
20-15	ESAI Pin-Out and Internal Clock Connection.....	20-17



# Figures

Figure Number	Title	Page Number
20-16	No Internal Clock Is Connected Between ESAI and ESAI_1: 0x .....	20-19
20-17	An Internal Clock Is Connected Between ESAI and ESAI_1: 10 .....	20-19
20-18	An Internal Clock Is Connected Between ESAI and ESAI_1: 11 .....	20-19
20-19	ESAI_2/ESAI_3 Internal Clock Connection Is Disabled: 0x .....	20-20
20-20	An Internal Clock Is Connected Between ESAI_2 and ESAI_3: 10 .....	20-20
20-21	An Internal Clock Is Connected Between ESAI_2 and ESAI_3: 11 .....	20-21
21-1	EMC Block Diagram .....	21-2
21-2	Base Register High Part (x) .....	21-9
21-3	Base Register Low Part (x) .....	21-10
21-4	Option Register High Part—GPCM Mode(x) .....	21-12
21-5	Option Register Low Part—GPCM Mode(x) .....	21-13
21-6	Option Register High Part—UPM Mode .....	21-15
21-7	Option Register Low Part (x)—UPM Mode .....	21-16
21-8	Option Register High Part (x)—SDRAM Mode .....	21-17
21-9	Option Register Low Part (x)—SDRAM Mode .....	21-18
21-10	UPM Memory Address High Part .....	21-19
21-11	UPM Memory Address Low Part .....	21-19
21-12	UPM Mode Registers High Part .....	21-20
21-13	UPM Mode Registers Low Part .....	21-22
21-14	Memory Refresh Timer Prescaler Register .....	21-23
21-15	UPM Data Register High Part .....	21-24
21-16	UPM Data Register Low Part .....	21-24
21-17	SDRAM Machine Mode Register High Part .....	21-25
21-18	SDRAM Machine Mode Register Low Part .....	21-26
21-19	UPM Refresh Timer .....	21-28
21-20	SDRAM Refresh Timer .....	21-28
21-21	Transfer Error Status Register .....	21-29
21-22	Transfer Error Check Disable Register .....	21-30
21-23	Transfer Error Interrupt Enable Register .....	21-31
21-24	Transfer Error Attributes Register High Part .....	21-32
21-25	Transfer Error Attributes Register Low Part .....	21-32
21-26	Transfer Error Address Register High Part .....	21-33
21-27	Transfer Error Address Register Low Part .....	21-33
21-28	Local Bus Configuration Register High Part .....	21-34
21-29	Local Bus Configuration Register Low Part .....	21-34
21-30	Clock Ratio Register High Part .....	21-35
21-31	Clock Ratio Register Low Part .....	21-36
21-32	Basic Operation of Memory Controllers in EMC .....	21-37
21-33	Basic EMC Bus Cycle with LALE, TA, and LCSx .....	21-39
21-34	Local Bus to GPCM Device Interface .....	21-40

# Figures

Figure Number	Title	Page Number
21-35	GPCM Basic Read Timing (XACS = 0, ACS = 1x, TRLX = 0, CLKDIV = 4,8) .....	21-40
21-36	GPCM Basic Write Timing (XACS = 0, ACS = 00, CSNT = 1, SCY = 1, TRLX = 0, CLKDIV = 4, 8) .....	21-46
21-37	GPCM Relaxed Timing Read (XACS = 0, ACS = 1x, SCY = 1, CSNT = 0, TRLX = 1, CLKDIV = 4, 8) .....	21-47
21-38	GPCM Relaxed Timing Back-to-Back Writes (XACS = 0, ACS = 1x, SCY = 0, CSNT = 0, TRLX = 1, CLKDIV = 4, 8).....	21-47
21-39	GPCM Relaxed Timing Write (XACS = 0, ACS = 10, SCY = 0, CSNT = 1, TRLX = 1, CLKDIV = 4, 8) .....	21-48
21-40	GPCM Relaxed Timing Write (XACS = 0, ACS = 00, SCY = 1, CSNT = 1, TRLX = 1, CLKDIV = 4, 8) .....	21-48
21-41	GPCM Read Followed by Read (TRLX = 0, EHTR = 0, Fastest Timing) .....	21-49
21-42	GPCM Read Followed by Write (TRLX = 0, EHTR = 1, One-Cycle Extended Hold Time on Reads) .....	21-50
21-43	GPCM Read Followed by Write (TRLX = 1, EHTR = 0, Four-Cycle Extended Hold Time on Reads) .....	21-50
21-44	External Termination of GPCM Access.....	21-51
21-45	Connection to a 24-Bit SDRAM with 12 Address Lines.....	21-53
21-46	SDRAM Address Multiplexing .....	21-55
21-47	PRETOACT = 2 (2 Clock Cycles).....	21-56
21-48	ACTTORW = 2 (2 Clock Cycles).....	21-57
21-49	CL = 2 (2 Clock Cycles) .....	21-57
21-50	WRC = 2 (2 Clock Cycles) .....	21-58
21-51	RFRC = 4 (6 Clock Cycles) .....	21-58
21-52	BUFCMD = 1, CRR[BUFCMDC] = 2 .....	21-59
21-53	SDRAM Single-Beat Read, Page Closed, CL = 3 .....	21-59
21-54	SDRAM Single-Beat Read, Page Hit, CL = 3 .....	21-59
21-55	SDRAM Two-Beat Burst Read, Page Closed, CL = 3.....	21-59
21-56	SDRAM Four-Beat Burst Read, Page Miss, CL = 3.....	21-60
21-57	SDRAM Single Beat Write, Page Hit .....	21-60
21-58	SDRAM Three-Beat Write, Page Closed.....	21-60
21-59	SDRAM Read-after-Read Pipelined, Page Hit, CL = 3.....	21-60
21-60	SDRAM Write-after-Write Pipelined, Page Hit.....	21-61
21-61	SDRAM Read-after-Write Pipelined, Page Hit .....	21-61
21-62	SDRAM MODE-SET Command.....	21-62
21-63	SDRAM Bank-Staggered Auto Refresh Timing.....	21-63
21-64	User-Programmable Machine Functional Block Diagram.....	21-63
21-65	RAM Array Indexing .....	21-64
21-66	Memory Refresh Timer Request Block Diagram .....	21-65
21-67	UPM Clock Scheme for CRR[CLKDIV] = 2 .....	21-68

# Figures

Figure Number	Title	Page Number
21-68	UPM Clock Scheme for CRR[CLKDIV] = 4 (or 8) or CRR[CLKDIV] = 2 (or 4).....	21-69
21-69	RAM Array and Signal Generation .....	21-69
21-70	RAM Word Field Descriptions .....	21-70
21-71	LCSx Signal Selection .....	21-73
21-72	UPM Read Access Data Sampling.....	21-75
21-73	Effect of UPWAIT Signal .....	21-76
21-74	Single-Beat Read Access to FPM DRAM .....	21-78
21-75	Single-Beat Write Access to FPM DRAM .....	21-79
21-76	Burst Read Access to FPM DRAM Using LOOP (2 Beats Shown).....	21-80
21-77	Refresh Cycle to FPM DRAM.....	21-81
21-78	Exception Cycle .....	21-82
21-79	Multiplexed Address and Data Bus .....	21-83
21-80	External Memory Controller Peripheral Hierarchy .....	21-84
21-81	EMC Peripheral Hierarchy for Very High Bus Speeds.....	21-85
21-82	GPCM Address Timing.....	21-85
21-83	GPCM Data Timing .....	21-86
21-84	256-Mbit SDRAM Diagram .....	21-89
21-85	SDRAM Power-Down Mode Timing .....	21-90
21-86	SDRAM Self-Refresh Mode Timing .....	21-91
21-87	EMC De-Skew PLL .....	21-93
22-1	DSP56724/DSP56725 JTAG Block Diagram.....	22-2

# Figures

**Figure  
Number**

**Title**

**Page  
Number**

# Tables

Table Number	Title	Page Number
2-1	DSP56724/DSP56725 Shared/Dedicated Pins.....	2-1
2-2	DSP56724/DSP56725 Signal Groups.....	2-2
2-3	Power Pins.....	2-6
2-4	Ground Pins.....	2-6
2-5	SCAN Signals .....	2-7
2-6	Clock and PLL Signals .....	2-7
2-7	Reset Pin .....	2-8
2-8	Interrupt and Mode Control Signals.....	2-8
2-9	Non-Maskable Interrupt for DSP Core-1 (NMI1) .....	2-10
2-10	DSP56724/DSP56725 SHI Pin Configuration.....	2-10
2-11	Serial Host Interface Signals (SHI).....	2-11
2-12	Serial Host Interface Signals (SHI_1).....	2-14
2-13	Enhanced Serial Audio Interface Signals (ESAI) .....	2-15
2-14	Enhanced Serial Audio Interface Signals (ESAI_1) .....	2-19
2-15	Enhanced Serial Audio Interface Signals (ESAI_2) .....	2-20
2-16	Enhanced Serial Audio Interface Signals (ESAI_3) .....	2-22
2-17	WDT Signal .....	2-26
2-18	External Memory: EMC Signals.....	2-26
2-19	Digital Audio Interface: S/PDIF Signals .....	2-29
2-20	Dedicated Port G Signals and Mode Pins .....	2-30
2-21	JTAG/OnCE Interface.....	2-30
3-1	Core-0 Configuration .....	3-2
3-2	Core-1 Configuration .....	3-2
3-3	DSP Core-0 Memory Map Locations .....	3-2
3-4	DSP Core-1 Memory Map Locations .....	3-3
3-5	X-Memory Map for DSP Core-0 and Core-1 .....	3-4
3-6	Y-Memory Map for DSP Core-0 and Core-1 .....	3-4
3-7	Detailed Device X-Memory Map .....	3-5
3-8	Detailed Device Y-Memory Map.....	3-12
5-1	Operation Mode Register Bit Definitions .....	5-2
5-2	Status Register Bit Definitions.....	5-4
5-3	Core-0 Operating Modes in DSP56724 .....	5-6
5-5	DSP56724 Core-0/Core-1 Boot Modes .....	5-7
5-4	Core-1 Operating Modes in DSP56724 .....	5-7
5-6	Peripherals and DMA Interrupt Priority Level Bits.....	5-9
5-7	External Interrupts Priority Level Bits.....	5-9
5-8	Interrupt Sources Priorities within an IPL .....	5-12
5-9	Reset and Interrupt Vector Summary .....	5-15
5-10	DMA Request Sources.....	5-19
6-1	CIM Memory Map.....	6-2
6-2	CIM Register Summary .....	6-2

## Tables

Table Number	Title	Page Number
7-1	System Clock Definition .....	7-3
7-2	Signal Properties .....	7-4
7-3	PLL Output Configurations .....	7-5
7-4	Output Divider Values.....	7-6
7-5	Block Memory Map .....	7-7
7-6	Register Summary .....	7-7
7-7	Shared Peripheral Clock Enable Register Field Description .....	7-8
7-8	PLL Control Registers Field Description.....	7-8
7-9	PLL Programming Examples.....	7-9
7-10	ASRC Control Division Registers Field Description .....	7-11
8-1	GPIO Pins Summary Table .....	8-1
8-2	PCRH and PRRH Bits Functionality .....	8-3
8-3	PORT H Registers Summary .....	8-3
8-4	ESAI/EXTAL Clock Control Bits.....	8-4
8-5	PCRH1 and PRRH1 Bits Functionality .....	8-5
8-6	PORT H1 Registers Summary .....	8-5
8-7	ESAI/EXTAL Clock Control Bit .....	8-6
8-8	GPIO Port A Registers Summary .....	8-7
8-9	PCRA/A1 and PRRA/A1 Bits.....	8-8
8-10	GPIO Port G Registers Summary .....	8-9
8-11	PCRG and PRRG Bits Functionality .....	8-10
9-1	Receiver Clock Sources (Asynchronous Mode Only) .....	9-5
9-2	Transmitter Clock Sources (Asynchronous Mode Only).....	9-6
9-3	Transmitter High Frequency Clock Divider .....	9-12
9-4	Transmit Network Mode Selection .....	9-17
9-5	ESAI Transmit Slot and Word Length Selection .....	9-19
9-6	Receiver High Frequency Clock Divider.....	9-25
9-7	SCKR Pin Definition Table.....	9-26
9-8	FSR Pin Definition Table .....	9-27
9-9	HCKR Pin Definition Table .....	9-27
9-10	ESAI Receive Network Mode Selection.....	9-30
9-11	ESAI Receive Slot and Word Length Selection.....	9-30
9-12	PCRC and PRRC Bits Functionality.....	9-51
10-1	SHI Interrupt Vectors .....	10-5
10-2	SHI Internal Interrupt Priorities .....	10-5
10-3	SHI Noise Reduction Filter Mode .....	10-10
10-4	SHI Data Size.....	10-11
10-5	HREQ Function In SPI Slave Mode .....	10-13
10-6	HCSR Receive Interrupt Enable Bits.....	10-15
11-1	Timer GPIO (Mode 0).....	11-5
11-2	Timer Prescaler Load Register (TPLR) Bit Definitions .....	11-9

# Tables

Table Number	Title	Page Number
11-3	Timer Prescaler Count Register (TPCR) Bit Definitions.....	11-10
11-4	Timer Control/Status Register (TCSR) Bit Definitions .....	11-11
13-1	ICC Block Memory Map .....	13-3
13-2	ICC Registers Summary.....	13-4
13-3	ICDR1 Field Descriptions.....	13-5
13-4	ICCR1 Field Descriptions.....	13-6
13-5	ICDR2 Field Descriptions.....	13-6
13-6	ICCR2 Field Descriptions .....	13-7
13-7	ICDR3 Field Descriptions.....	13-7
13-8	ICCR3 Field Descriptions.....	13-8
13-9	ICAR3 Field Descriptions.....	13-8
13-10	ICDR4 Field Descriptions.....	13-9
13-11	ICCR4 Field Descriptions.....	13-9
13-12	ICAR4 Field Descriptions.....	13-10
13-13	CPR1 Field Descriptions.....	13-10
13-14	CPR2 Field Descriptions.....	13-11
13-15	EISR Field Description .....	13-15
14-1	Round-Robin Arbitration Method.....	14-4
17-1	External Memory X Space Burst Control .....	17-4
17-2	External Memory Y Space Burst Control .....	17-4
17-3	External Memory P Space Burst Control.....	17-4
18-1	Signal Properties .....	18-3
18-2	S/PDIF Memory Map.....	18-4
18-3	S/PDIF Configuration Register (SCR) Fields.....	18-5
18-4	CDText Control Register (SRCD) Field Descriptions .....	18-6
18-5	PhaseConfig Register (SRPC) Field Descriptions .....	18-7
18-6	Interrupt Register Field Descriptions.....	18-9
18-7	SPDIFRcvLeft Register (SRL) Fields.....	18-10
18-8	SPDIFRcvRight Register (SRR) Fields .....	18-10
18-9	SPDIFRxCChannel_h Register (SRCSH) Fields.....	18-11
18-10	SPDIFRxCChannel_l Register (SRCSL) Fields .....	18-11
18-11	S/PDIF UchannelRcv Register (SRU) Fields .....	18-12
18-12	S/PDIF QchannelRcv Register (SRQ) Fields .....	18-12
18-13	SPDIFTxLeft Register (STL) Fields.....	18-12
18-14	SPDIFTxRight Register (STR) Fields .....	18-13
18-15	SPDIFTxCChannelCons_h Register (STCSCH) Fields .....	18-13
18-16	SPDIFTxCChannelCons_l Register (STCSCL) Fields.....	18-14
18-17	SPDIFTxChannelProf_h Register (STCSPH) Fields.....	18-14
18-18	SPDIFTxCChannelProf_l Register (STCSPL) Fields .....	18-14
18-19	S/PDIF FreqMeas Register (SRFM) Fields .....	18-15
18-20	SPDIFTxCclk Register (STC) Fields .....	18-15

# Tables

Table Number	Title	Page Number
18-21	Sync Control Bits .....	18-19
18-22	S/PDIF Receiver Interrupt Exceptions .....	18-21
19-1	ASRC Specifications .....	19-3
19-2	Block Memory Map .....	19-5
19-3	ASRC Control Register Bits (ASRCTR) .....	19-7
19-4	Interrupt Enable Register (ASRIER) .....	19-9
19-5	Interrupt Enable Mask Register (ASRIEM) .....	19-10
19-6	Channel Number Configuration Register (ASRCNCR) .....	19-11
19-7	Filter Configuration Status Register (ASRCFG) .....	19-12
19-8	Clock Source Register (ASRCSSR) .....	19-14
19-9	Clock Divider Register-1 (ASRCDR1) .....	19-17
19-10	Clock Divider Register-2 (ASRCDR2) .....	19-17
19-11	Status Register (ASRSTR) .....	19-18
19-12	Debug Control Register (ASRDCR) .....	19-21
19-13	Recommended Values for Parameter Registers .....	19-23
19-14	ASRC Task Queue FIFO Register 1 (ASRTFR1) .....	19-23
19-15	Channel Counter Register (ASRCCR) .....	19-24
19-16	Interrupt Vectors .....	19-26
19-17	DMA Requests .....	19-27
19-18	Possible ASRC Configurations .....	19-29
19-19	Pre-Processing, Post-Processing Options .....	19-30
20-1	Chip Configuration Module Memory Map .....	20-1
20-2	CFG Register Summary .....	20-2
20-3	Reserved Control Register Field Description .....	20-3
20-4	EMBC Register Field Description—DSP56724 .....	20-4
20-5	Field Descriptions, DSP56725 .....	20-5
20-6	External Peripherals X,Y,P Space Set by Burst Boundary .....	20-5
20-7	LPSC Register Field Description—DSP56724 .....	20-6
20-8	Field Descriptions—DSP56725 .....	20-6
20-9	PMC Field Descriptions .....	20-7
20-10	EPSC Field Descriptions .....	20-8
20-11	ODBC Field Descriptions .....	20-10
20-12	PSRC Field Description .....	20-11
20-13	ARCR Field Descriptions .....	20-12
20-14	EICCR Field Descriptions .....	20-15
20-15	Internal Clock Connections Between ESAI and ESAI_1 .....	20-17
20-16	Internal Clock Connections Between ESAI_2 and ESAI_3 .....	20-18
20-17	ESAI and ESAI_1 Internal Clock Connections .....	20-18
20-18	ESAI_2 and ESAI_3 Internal Clock Connections .....	20-20
21-1	External Signal Descriptions .....	21-3
21-2	Detailed Signal Descriptions .....	21-4



## Tables

Table Number	Title	Page Number
21-3	EMC Registers Memory Map .....	21-7
21-4	BRHx field Descriptions .....	21-10
21-5	BRLx Field Descriptions .....	21-10
21-6	Memory Bank Sizes in Relation to Address Mask .....	21-11
21-7	ORHx—GPCM Field Descriptions .....	21-12
21-8	ORLx—GPCM Field Descriptions .....	21-13
21-9	ORHx—UPM Field Descriptions .....	21-15
21-10	ORLx—UPM Field Descriptions .....	21-16
21-11	ORHx—SDRAM Mode .....	21-17
21-12	ORLx—SDRAM Field Descriptions .....	21-18
21-13	MARH Field Descriptions .....	21-19
21-14	MARL Field Descriptions .....	21-19
21-15	MxMR High Part Field Descriptions .....	21-20
21-16	MxMRL Field Descriptions .....	21-22
21-17	MRTPR Field Descriptions .....	21-23
21-18	MDRH Field Descriptions .....	21-24
21-19	MDRL Field Descriptions .....	21-24
21-20	SDMRH Field Descriptions .....	21-25
21-21	SDMRL Field Descriptions .....	21-26
21-22	URT Field Descriptions .....	21-28
21-23	SRT Field Descriptions .....	21-29
21-24	TESR Field Descriptions .....	21-30
21-25	TEDR Field Descriptions .....	21-30
21-26	TEIR Field Descriptions .....	21-31
21-27	TEATRH Field Descriptions .....	21-32
21-28	TEATRL Field Descriptions .....	21-33
21-29	TEARH Field Descriptions .....	21-33
21-30	TEARL Field Descriptions .....	21-34
21-31	LBCRH Field Description .....	21-34
21-32	LBCRL Field Descriptions .....	21-35
21-33	LCRRH Field Description .....	21-35
21-34	LCRRL Field Descriptions .....	21-36
21-35	GPCM Write Control Signal Timing for CRR[CLKDIV] = 4 or 8 .....	21-41
21-36	GPCM Read Control Signal Timing for CRR[CLKDIV] = 4 or 8 .....	21-42
21-37	GPCM Write Control Signal Timing for CRR[CLKDIV] = 2 .....	21-43
21-38	GPCM Read Control Signal Timing for CRR[CLKDIV] = 2 .....	21-44
21-39	Boot Bank Field Values After Reset .....	21-52
21-40	SDRAM Interface Commands .....	21-54
21-41	UPM Routines Start Addresses .....	21-65
21-42	RAM Word Field Descriptions .....	21-70
21-43	MxMR Loop Field Use .....	21-74

## Tables

<b>Table Number</b>	<b>Title</b>	<b>Page Number</b>
21-44	Micron SDRAM Devices .....	21-88
21-45	Logical Address Bus Partitioning .....	21-90
21-46	SDRAM Device Address Port During Row Address Phase .....	21-90
21-47	SDRAM Device Address Port During READ/WRITE Command.....	21-90
21-48	SDRAM Capacitance .....	21-92
21-49	SDRAM AC Characteristics .....	21-92
22-1	JTAG-1 Supported Instructions.....	22-3
22-2	JTAG-0 Supported Instructions.....	22-3
22-3	JTAG Identification Register Configuration.....	22-3
22-4	JTAG Instruction Descriptions.....	22-4
22-5	External Signal Descriptions.....	22-4

## About This Book

The *Symphony™ DSP56724/DSP56725 Multi-Core Audio Processors Reference Manual* describes the features and operation of the Symphony™ DSP56724/DSP56725 Multi-Core Audio Processors, including, for example, their main features, architecture, function blocks, operation modes, pin signals, clocks, interrupts, DMA operations, and memory maps.

The DSP56724/DSP56725 Multi-Core Audio Processors are devices of the DSP5672x family of programmable CMOS DSPs, designed using dual DSP56300 24-bit cores. The DSP56724/DSP56725 are intended for automotive, consumer, and professional audio applications that require high performance for audio processing. Potential applications include A/V receivers, car audio/amplifiers, and professional audio equipment.

## Audience

The *Symphony DSP56724/DSP56725 Multi-Core Audio Processors Reference Manual* provides to the design engineer the necessary data to successfully integrate the processors into a wide variety of applications.

The intended audience for this document includes system architects, system modeling teams, IC designers, software architects/designers, and the platform integration and testing teams. The level of detail in this document is intended to provide the reader with sufficient information to validate the capabilities of the processes in the targeted applications.

## Organization

This reference manual is organized into chapters that describe the operation and programming of the processors. It includes brief summaries of the major components, as well as listings of the memory maps for the processors and shared memories.

This manual also contains chapters that describe the operations and configuration of the peripherals, including the modules that provide bootmodes, memory, and connectivity.

## Suggested Reading

The *DSP56300 Family Manual* (DSP56300FM) is suggested for a complete description of the Symphony DSP56724/DSP56725 Multi-Core Audio Processors, and is necessary to design with the devices. This document is helpful when used in conjunction with this reference manual.

## Conventions

This reference manual uses the following conventions:

- $\overline{\text{OVERBAR}}$  is used to indicate a signal that is active when pulled low: for example,  $\overline{\text{RESET}}$ .
- *Logic level one* is a voltage that corresponds to Boolean true (1) state.
- *Logic level zero* is a voltage that corresponds to Boolean false (0) state.

- To *set* a bit or bits means to establish logic level one.
- To *clear* a bit or bits means to establish logic level zero.
- A *signal* is an electronic construct whose state conveys or changes in state convey information.
- A *pin* is an external physical connection. The same pin can be used to connect a number of signals.
- *Asserted* means that a discrete signal is in active logic state.
  - *Active low* signals change from logic level one to logic level zero.
  - *Active high* signals change from logic level zero to logic level one.
- *Negated* means that an asserted discrete signal changes logic state.
  - *Active low* signals change from logic level zero to logic level one.
  - *Active high* signals change from logic level one to logic level zero.
- LSB means *least significant bit* or *bits*, and MSB means *most significant bit* or *bits*. References to low and high bytes or words are spelled out.
- Numbers preceded by a percent sign (%) are binary. Numbers preceded by a *0x* are hexadecimal.
- Courier monospaced type indicate commands, command parameters, code examples, expressions, data types, and directives.
- Italic type indicates replaceable command parameters.

# Chapter 1

## Introduction

### 1.1 Overview

The Symphony™ DSP56724 and DSP56725 Multi-Core Audio Processors are devices of the DSP5672x family of programmable CMOS DSPs, designed using multiple DSP56300 24-bit cores. The DSP56724 and DSP56725 are intended for automotive, consumer, and professional audio applications that require high performance for audio processing. These processors also support professional audio applications, including audio recording, signal processing and digital audio synthesis. Potential applications include A/V receivers, car audio/amplifiers, and professional audio equipment. Additional device features include support for digital audio compression/decompression, sound field processing, acoustic equalization and other digital audio algorithms. With two DSP56300 cores, the DSP56724 (or DSP56725) device can replace two DSP devices in designs, providing high MIPs and lower cost.

DSP56724/DSP56725 features include:

- Two DSP56300 enhanced cores: up to 500 MIPs (up to 250 MIPs/core) with up to 250-MHz clock; each core includes:
  - Highly parallel instruction set
  - Hardware debugging support (JTAG TAP, OnCE™ module)
  - Eight-channel DMA controller
  - Wait and Stop low-power standby modes
- Configurable and flexible arbitration method for the shared peripherals and shared memory blocks
- Powerful audio data communication ability:
  - Four Enhanced Serial Audio Interface (ESAI) modules to transmit and receive audio data. Two ESAI modules are provided for each core. For each ESAI, up to 4 receivers and up to 6 transmitters, master or slave. Protocols include I<sup>2</sup>S, Left-Justified, Right-Justified, Sony, AC97, network and other programmable protocols.
  - One S/PDIF module is shared by the two cores to transmit and receive audio data in IEC958 format.
- Powerful host communication port:
  - Two Serial Host Interface (SHI, SHI\_1) modules, with one module for each core. SHIs support SPI and I<sup>2</sup>C protocols, multi-master capability in I<sup>2</sup>C mode, 10-word receive FIFO, and support for 8, 16 and 24-bit words.
- Two triple-timer modules (TEC, TEC\_1), with one timer module for each core.
- Two watchdog timer modules (WDT, WDT\_1), with one watchdog timer module for each core, to prevent code runaway problems.

## Introduction

- An External Memory Controller (EMC) that can be accessed by both DSP cores, which supports SDRAM, SRAM, EPROM, flash EPROM, burstable RAM, regular DRAM devices, and extended data output DRAM devices. *Note that the EMC is only available on DSP56724 devices, and is not available on DSP56725 devices.* The EMC includes:
  - High performance SDRAM machine
  - A general-purpose chip-select machine (GPCM)
  - Up to three user-programmable machines (UPMs)
- A seamless hardware Asynchronous Sampling Rate Converter (ASRC) that is accessible to both cores, to support different sample rate audio data transmission reception. Three data sampling rate convert pairs can be supported at the same time. Different pairs can be used by different cores at the same time.
- Inter-Core Communication (ICC) module:
  - 32-Kword shared memory between the two DSP56300 cores
  - Supports a flexible arbitration system which allows multiple methods of arbitration
  - Non-maskable and maskable interrupts between the two cores
  - Poll data registers for simple data transfers
- Includes as many as 79 GPIO pins, shared with other peripherals function pins; the actual number is different for different device packages.

## 1.2 Block Diagram

Figure 1-1 shows the block diagram for the DSP56724/DSP56725.

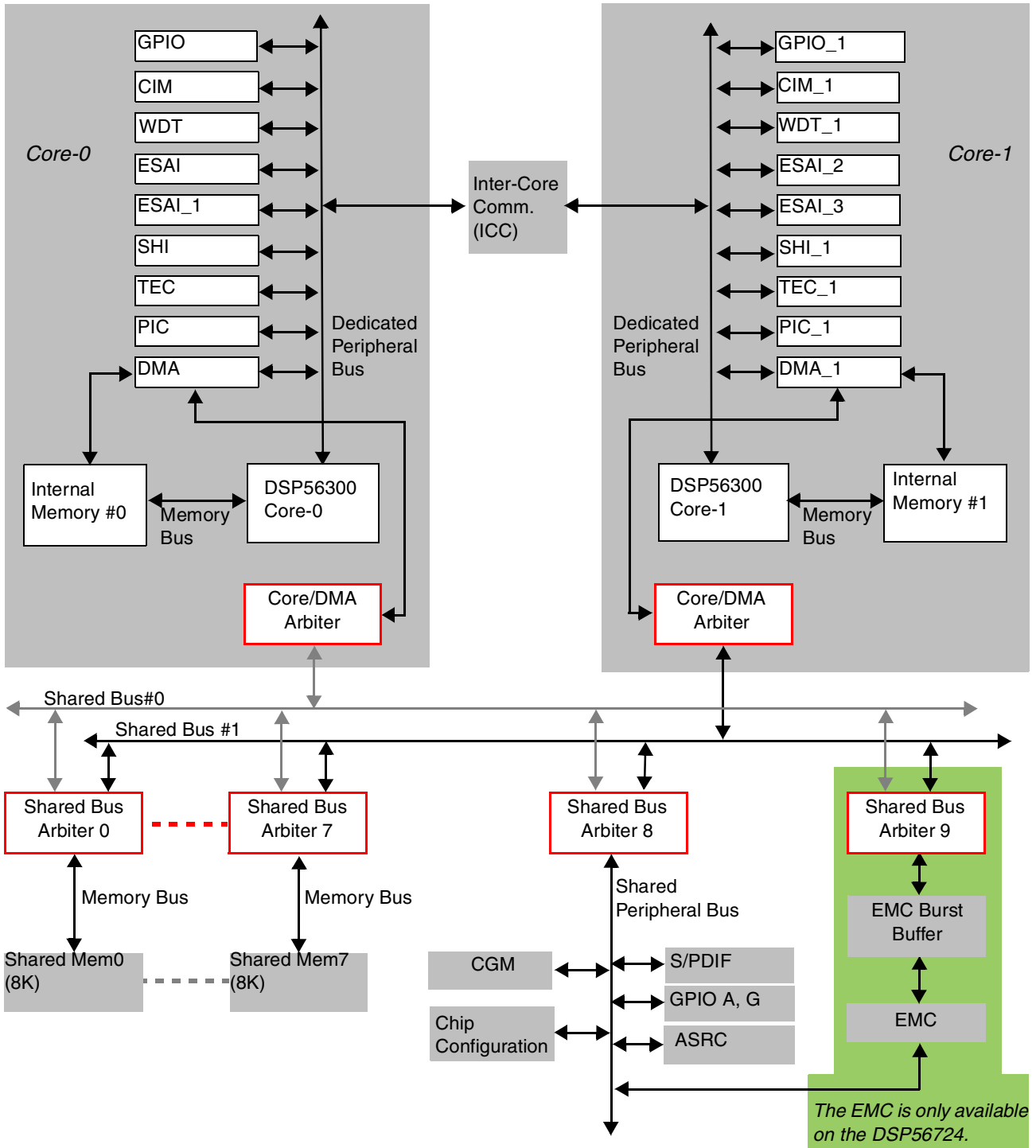


Figure 1-1. DSP56724/DSP56725 Block Diagram

## 1.3 Features

In addition to high MIPS, the DSP56724/DSP56725 provides powerful and flexible audio data communications and supports a wide variety of audio applications. This section provides a brief description of the DSP56724/DSP56725 processor features.

The DSP56724/DSP56725 has two DSP56300 DSP cores. The high throughput of the DSP56300 family of processors makes them well-suited for high-speed control, efficient signal processing, numeric processing, and audio applications. Benefits of using DSP56300 cores include:

- *Speed:* The DSP56300 family supports most high-performance DSP applications.
- *Precision:* The data paths are 24 bits wide, providing 144 dB of dynamic range. Intermediate results held in the 56-bit accumulators can range over 336 dB.
- *Parallelism:* Each on-chip execution unit, memory, and peripheral operates independently and in parallel with the other units through a sophisticated bus system. The Data ALU, AGU, and program controller operate in parallel so that the following operations can execute in a single instruction:
  - An instruction pre-fetch
  - A 24-bit × 24-bit multiplication
  - A 54-bit addition
  - Two data moves
  - Two address-pointer updates using either linear or modulo arithmetic
- *Flexibility:* While many other DSPs require external communication devices to interface with peripheral circuits (such as A/D converters, D/A converters, or processors), the DSP56300 family provides on-chip serial and parallel interfaces that support various configurations of memory and peripheral modules. The peripherals are interfaced to the DSP56300 family core through a peripheral interface bus that provides a common interface to many different peripherals.
- *Sophisticated Debugging:* Freescale's On-Chip Emulation (OnCE) technology allows simple, inexpensive, and speed-independent access to the internal registers for debugging. With the OnCE module, you can easily determine the exact status of the registers and memory locations, plus identify which instructions were executed last.
- *Phase Locked Loop (PLL)-Based Clocking:* The PLL allows the chip to use almost any available external system clock for full-speed operation, while also supplying an output clock synchronized to a synthesized internal core clock. It improves the synchronous timing of the external memory port, eliminating the timing skew common on other processors.
- *Invisible Pipeline:* The seven-stage instruction pipeline is essentially invisible to the programmer, allowing straightforward program development in either assembly language or high-level languages such as C or C++.
- *Similar Instruction Set:* The instruction mnemonics are similar to those used for microcontroller units, making an easy transition from programming microprocessors to programming the device. New microcontroller instructions, addressing modes, and bit field instructions allow for significant decreases in program code size. The orthogonal syntax controls the parallel execution units. The hardware DO loop and the repeat (REP) instructions make writing straight-line code obsolete.
- *Low Power:* Designed in CMOS, the DSP56300 family consumes very little power. Two additional low-power modes, Stop and Wait, further reduce power requirements. Wait is a low-power mode



in which the DSP56300 core shuts down, but the peripherals and interrupt controller continue to operate, so that an interrupt can bring the chip out of Wait mode. In Stop mode, even more of circuitry is shut down for the lowest power consumption. Several different methods are available to bring the chip out of Stop mode: hardware RESET, IRQA, and DE.

## 1.4 Overview of Peripherals

The peripherals include the following:

- DMA
- PIC
- ESAI
- SHI
- TEC
- WDT
- CIM
- S/PDIF
- ASRC
- EMC
- CGM
- Shared memory
- ICC
- Shared bus arbiters
- Chip configuration module
- JTAG controller

### 1.4.1 Direct Memory Access Controller (DMA, DMA\_1)

The DMA controller enables data transfers without any interactions with the DSP cores. During DMA accesses, it supports any combination of source and destination between internal memory, internal peripheral I/O, and external memory. DMA features include:

- Eight DMA channels supporting internal and external accesses
- One-, two-, and three-dimensional transfers (including circular buffering)
- End-of-block-transfer interrupts
- Triggering from interrupt lines and all peripherals

### 1.4.2 Program Interrupt Controller (PIC, PIC\_1)

The Program Interrupt Controller arbitrates among all interrupt requests (internal interrupts and the five external requests IRQA, IRQB, IRQC, IRQD, and NMI), and generates the appropriate interrupt vector address.

The Program Interrupt Controller supports the following:

- Both non-maskable and maskable interrupts
- Up to 18 DMA interrupts and 24 Peripheral interrupts
- Up to 9 non-maskable interrupts

### 1.4.3 Enhanced Serial Audio Interfaces (ESAI, ESAI\_1, ESAI\_2, ESAI\_3)

The enhanced serial audio interfaces provide full-duplex serial GPIO pins or serial communications with a variety of serial devices, including one or more industry-standard codecs, other DSPs, microprocessors and other peripherals that implement the serial peripheral interface (SPI) serial protocol. Each ESAI consists of independent transmitter and receiver sections, each with its own clock generator, and is a superset of the DSP56300 family ESSI peripherals and the DSP56000 family SAI peripherals.

### 1.4.4 Serial Host Interfaces (SHI, SHI\_1)

Each serial host interface provides a path for communications and program/coefficient data transfers between the DSP core and an external host processor. The SHI can interface directly to either of two well-known and widely used synchronous serial buses: the SPI bus and the Philips inter-integrated-circuit control (I2C) bus. The SHI supports either the SPI or I2C bus protocol, as required, from a slave or a single-master device. To minimize DSP overhead, the SHI supports single-, double- and triple-word data transfers. The SHI has a 10-word receive FIFO that permits receiving up to 30 words before generating a receive interrupt, reducing the overhead for data reception.

### 1.4.5 Triple Timers (TEC, TEC\_1)

Each Triple Timer is composed of a common 21-bit prescaler and three independent and identical general purpose 24-bit timer event counters, with each timer having its own register set. Each timer uses internal clocking, and can also interrupt the DSP after a specified number of events (clocks). Each timer can also be used to trigger DMA transfers after a specified number of events (clocks) have occurred.

### 1.4.6 Watch Dog Timers (WDT, WDT\_1)

Each watchdog timer is a 16-bit timer used to help software recover from runaway code. The timer is a free-running down-counter used to generate a reset on underflow. Software must periodically service the watchdog timer to restart the count down.

### 1.4.7 Core Integration Modules (CIM, CIM\_1)

Each DSP core has a Core Integration Module. Each core integration module includes a chip ID register, DMA stall monitor function, and OnCE global data bus (GDB) register.

### 1.4.8 Sony/Philips Digital Interface (S/PDIF)

The Sony/Philips Digital Interface (S/PDIF) audio module is a transceiver that allows the DSP to receive and transmit digital audio via this module. There is one S/PDIF in each DSP56724/DSP56725 device,

shared by the two DSP cores. The DSP provides a single S/PDIF receiver with four multiplexed inputs, and one S/PDIF transmitter with two outputs. The S/PDIF module can also transmit and receive the S/PDIF channel status (CS) and user (U) data. Not all S/PDIF pins are available on all packages.

### 1.4.9 Asynchronous Sample Rate Converter (ASRC)

Incoming audio data to the DSP can be received from various sources at different sampling rates. Outgoing audio data from the DSP can have different sampling rates, and additionally, it can be associated with output clocks that are asynchronous to the input clocks. The Asynchronous Sample Rate Converter (ASRC) converts the sampling rate of a signal associated to an input clock into a signal associated to a different output clock.

The ASRC supports concurrent sample rate conversion of up to 10 channels of about 120 dB THD+N. The sample rate conversion of each channel is associated to a pair of incoming and outgoing sampling rates.

The ASRC supports up to three sampling rate pairs. Although there is only one ASRC in the DSP56724/DSP56725 device (shared by the two DSP cores), the three sample rate pairs can be used by both DSP cores at the same time. The ASRC is hard-coded and implemented as a co-processor, requiring minimal CPU or DSP controller intervention.

### 1.4.10 External Memory Controller (EMC)

There is one EMC in each DSP56724 device, shared by the two DSP cores. Both cores can access external memory using the EMC. (DSP56725 devices do not have an EMC.) The EMC provides a seamless interface to many types of memory devices and peripherals over a shared address and data bus and dedicated control signals. The memory controller in the EMC controls a parameterized number of memory banks shared by a high performance SDRAM machine, a general-purpose chip-select machine (GPCM), and up to three user-programmable machines (UPMs).

With external latching, it supports connections to synchronous DRAM (SDRAM), SRAM, EPROM, flash EPROM, burstable RAM, regular DRAM devices, extended data output DRAM devices, and other peripherals. Support signals for external address latch (LALE) allows multiplexing of address with data lines in devices with strict pin count limitations.

### 1.4.11 Clock Generation Module (CGM)

The Clock Generation Module generates all clocks in the DSP56724/DSP56725 device; the output is a series of gated clocks. The CGM uses a low jitter phase-locked loop (PLL). The PLL has a wide range of frequency multiplications (1 to 256), predivider factors (1 to 32) and output divider (1 to 8). The CGM also has a power saving clock divider ( $2^i$ :  $i = 0$  to 7).

In functional mode, the PLL control register (PCTL) sits on the Shared Peripheral bus; both DSP cores can read and write these registers to change the chip's working frequency. Additionally, each core can independently enter stop or wait mode to save power. The shared peripherals enter power-saving mode only when both DSP cores enter the stop mode.

### 1.4.12 Shared Memory

The shared memory is a shared memory space accessible by either DSP Core-0 or DSP Core-1. The DSP56724/DSP56725 shared memory has four 8-Kword  $\times$  24-word memory blocks for a total of 32-K shared words and is located starting from 0x030000. It can be accessed as X or Y memory (with zero wait states) or as P memory (with 1 wait state).

The 8-Kword  $\times$  24 words blocks are single port SRAMs; the Shared Bus Arbiters perform arbitration when the two DSP cores try to access the same 8K x 24 SRAM block at the same time. No bus contentions occur when the two DSP cores access different 8K x 24 SRAM blocks simultaneously.

### 1.4.13 Inter-Core Communication (ICC)

Using the inter-core communication module, each DSP core can issue a maskable interrupt or non-maskable interrupt to the other core, and each core has its own write data register (which passes data to the other core when the interrupt is generated). There are also poll data registers for inter-core data exchange in the ICC. The ICC module interfaces with both cores' dedicated peripheral buses.

### 1.4.14 Shared Bus Arbiters

The Shared Bus Arbiter provides arbitration between the two DSP cores for the shared peripherals, shared memory and shared external memory interface (if available). It is a configurable arbiter, so users can choose the arbitration method via the appropriate chip configuration registers. The Shared Bus Arbiter supports using one of three arbitration schemes:

- Always round-robin method
- DSP Core-0 always has high priority
- DSP Core-1 always has high priority

### 1.4.15 Chip Configuration Module

The Chip Configuration module contains several registers which establish the mode of operation for various internal blocks, modules, and some of the peripherals. These registers include:

- Control bits of Shared Bus Arbiters
- EMC Burst Mode control bits
- Pin mux/switch control of ESAI, S/PDIF, S/PDIF Rx Clock output mux on ESAI HCKR pins
- Shared peripherals Soft Reset triggering and auto-release
- EMC PLL control and status

### 1.4.16 JTAG Controller

In the DSP56724/DSP56725 devices, two separate DSP cores are supported, each with their own OnCE and JTAG TAP controller. The two JTAG TAPs are daisy-chained, and appear to be two separate single core devices to the outside world.

# Chapter 2

## Signal Descriptions

### 2.1 Signal Groupings

Each product (DSP56724, DSP56725) is available in a variety of packages, which affects whether some modules use dedicated or shared external pins. See [Table 2-1](#).

**Table 2-1. DSP56724/DSP56725 Shared/Dedicated Pins**

Function	Module	DSP56724	DSP56725
		144-pin	80-pin
Timers	TEC	None	None
	TEC_1	None	None
Enhanced Serial Audio Interface	ESAI	No SDO0, SDO1	No SDO0, SDO1
	ESAI_1	No SDO0, SDO1, FST, FSR, SCKT, SCKR, HCKT, HCKR <sup>1</sup>	No SDO0, SDO1, FST, FSR, SCKT, SCKR, HCKT, HCKR <sup>1</sup>
	ESAI_2	No SDO0, SDO1, FST, FSR, SCKT, SCKR, HCKT, HCKR <sup>2</sup>	No SDO0, SDO1, FST, FSR, SCKT, SCKR, HCKT, HCKR <sup>3</sup>
	ESAI_3	No SDO0, SDO1	No SDO0, SDO1, HCKR
Serial Host Interface	SHI	All	All
	SHI_1	Only SS, others muxed with SHI	Only SS; others muxed with SHI
Watchdog Timer	WDT	All	All
	WDT_1	Muxed with WDT	Muxed with WDT
General Purpose I/O	GPIO	Only PG1, PG2	None
Sony/Philips Digital Interface Format	S/PDIF	Only SPDIFIN and SPDIFOUT1	None; muxed with ESAI_2's SDO2, SDO3
External Memory Controller	EMC	All	Not applicable, because DSP56725 does not have an EMC module.

<sup>1</sup> Clock and Frame Sync signals can be shared with ESAI.

<sup>2</sup> Clock and Frame Sync signals can be shared with ESAI\_3.

## Signal Descriptions

The input and output signals of the DSP56724/DSP56725 are organized into functional groups, as listed in [Table 2-2](#).

**Table 2-2. DSP56724/DSP56725 Signal Groups**

Signal Group	Signal		Number of Signals		Detailed Description
			DSP56724	DSP56725	
			144-Pin	80-Pin	
Power, Ground, Scan, Clock, Interrupts	Power ( $V_{DD}$ )		21	13	<a href="#">Table 2-3</a>
	Ground (GND)		21	18	<a href="#">Table 2-4</a>
	Scan Pins		1	1	<a href="#">Table 2-5</a>
	Reset Pin		1	1	<a href="#">Table 2-7</a>
	Clock and PLL	Port G <sup>8</sup>	4	4	<a href="#">Table 2-6</a>
	Shared External Interrupt Pins / Mode Control	Port G <sup>8</sup>	5	5	<a href="#">Table 2-8</a> <a href="#">Table 2-9</a>
DSP Core-0 Peripheral Pins	SHI	Port H <sup>1</sup>	5	5	<a href="#">Table 2-11</a>
	ESAI	Port C <sup>2</sup>	10	10	<a href="#">Table 2-13</a>
	ESAI_1	Port E <sup>3</sup>	4	4	<a href="#">Table 2-14</a>
	TEC	—	0	0	—
	WDT	No GPIO Function	1	1	<a href="#">Table 2-17</a>
DSP Core-1 Peripheral Pins	SHI_1	Port H1 <sup>4</sup>	1	1	<a href="#">Table 2-12</a>
	ESAI_2	Port C1 <sup>5</sup>	4	4	<a href="#">Table 2-15</a>
	ESAI_3	Port E1 <sup>6</sup>	10	9	<a href="#">Table 2-16</a>
	TEC_1	—	0	0	—
	WDT_1	No GPIO Function	0	0	—

**Table 2-2. DSP56724/DSP56725 Signal Groups (Continued)**

Signal Group	Signal		Number of Signals		Detailed Description
			DSP56724	DSP56725	
			144-Pin	80-Pin	
Pins of Shared Peripherals	SPDIF	Port G <sup>8</sup>	2	0	<a href="#">Table 2-19</a>
	EMC <sup>9</sup>	Port A <sup>7</sup>	48	0	<a href="#">Table 2-18</a>
	GPIO PORT G and Mode Pins	Port G <sup>8</sup>	2	0	<a href="#">Table 2-20</a>
	JTAG/OnCE Port for the two DSP Cores		4	4	<a href="#">Table 2-21</a>

**Note:**

- <sup>1</sup> Port H signals are the GPIO port signals that are multiplexed with the SHI  $\overline{\text{HREQ}}$  signal.
- <sup>2</sup> Port C signals are the GPIO port signals that are multiplexed with the ESAI signals.
- <sup>3</sup> Port E signals are the GPIO port signals that are multiplexed with the ESAI\_1 signals.
- <sup>4</sup> Port H1 signals are the GPIO port signals that are multiplexed with the SHI\_1  $\overline{\text{HREQ}}_1$  signals.
- <sup>5</sup> Port C1 signals are the GPIO port signals that are multiplexed with the ESAI\_2 signals.
- <sup>6</sup> Port E1 signals are the GPIO port signals that are multiplexed with the ESAI\_3 signals.
- <sup>7</sup> Port A signals are the GPIO port signals that are multiplexed with the EMC.
- <sup>8</sup> Port G signals are the GPIO port signals that are multiplexed with S/PDIF, shared external maskable interrupts, and PLL lock output signals.
- <sup>9</sup> DSP56724 products have an EMC; DSP56725 products do not have an EMC.

Figure 2-1 shows the DSP56725 80-pin package pin-out.

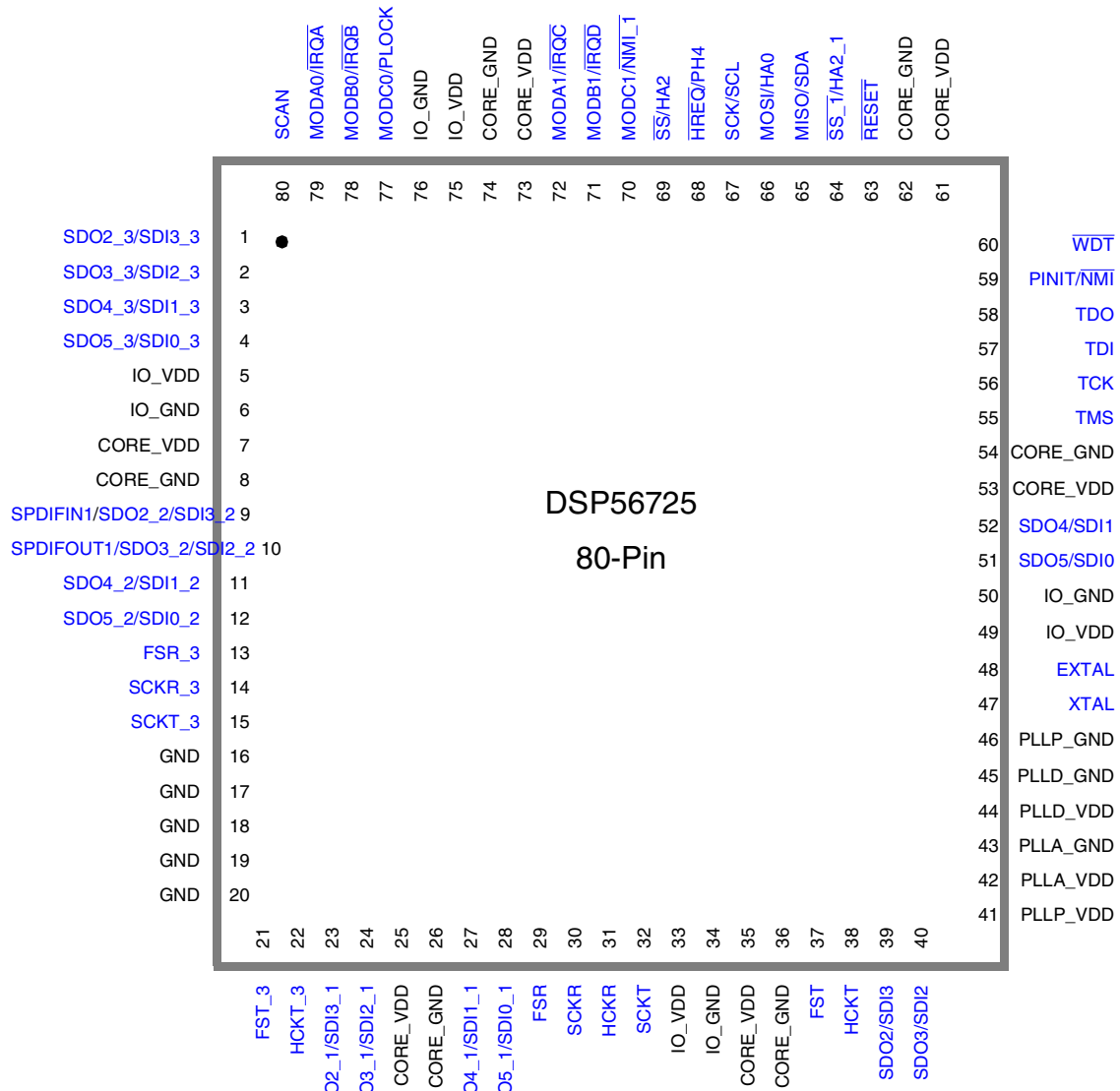


Figure 2-1. DSP56725 80-Pin Package Pin-Out



Figure 2-2 shows the DSP56724 144-pin package pin-out.

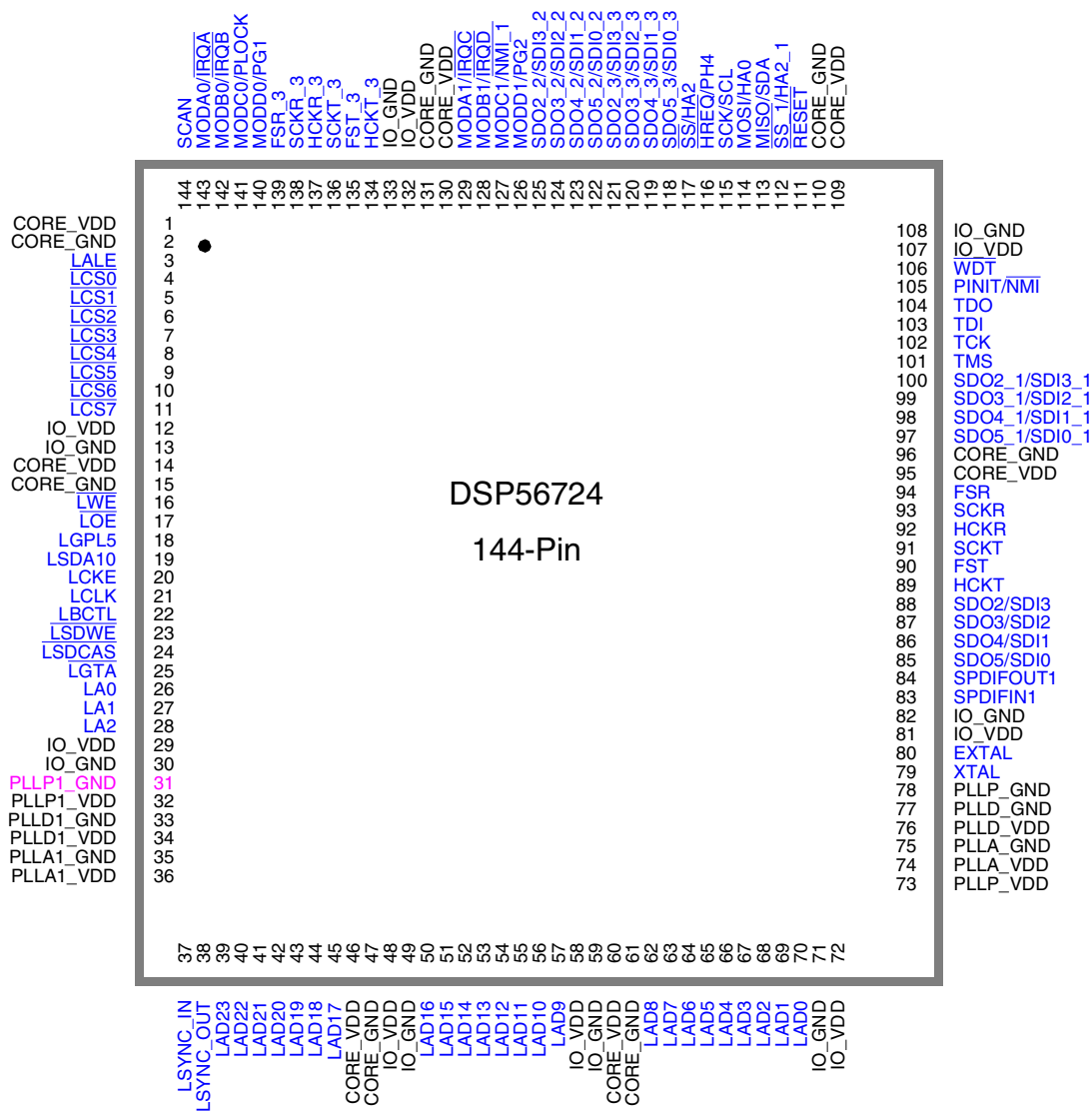


Figure 2-2. DSP56724 144-Pin Package Pin-Out

## 2.2 Signals in Each Functional Group

### 2.2.1 Power

Table 2-3 lists the power pins.

**Table 2-3. Power Pins**

Power Name	Description
PLLA_VDD PLLP_VDD PLLA1_VDD PLLP1_VDD	<b>PLL Power</b> The voltage (3.3 V) should be well-regulated and the input should be provided with an extremely low impedance path to the 3.3 V <sub>DD</sub> power rail. The user must provide adequate external decoupling capacitors.
PLLD_VDD PLLD1_VDD	<b>PLL Power</b> The voltage (1.0 V) should be well-regulated and the input should be provided with an extremely low impedance path to the 1.0 V <sub>DD</sub> power rail. The user must provide adequate external decoupling capacitors.
CORE_VDD	<b>Core Power</b> The voltage (1.0 V) should be well-regulated and the input should be provided with an extremely low impedance path to the 1.0 V <sub>DD</sub> power rail. The user must provide adequate decoupling capacitors.
IO_VDD	<b>I/O Power</b> The voltage (3.3 V) should be well-regulated and the input should be provided with an extremely low impedance path to the 3.3 V <sub>DD</sub> power rail. This is an isolated power for the SHI, SHI_1, ESAI, ESAI_1, ESAI_2, ESAI_3, Timer I/O, and other IO signals. The user must provide adequate external decoupling capacitors.

### 2.2.2 Ground

Table 2-4 lists the ground pins.

**Table 2-4. Ground Pins**

Ground Name	Description
PLLA_GND PLLP_GND PLLA1_GND PLLP1_GND	<b>PLL Ground</b> The PLL ground should be provided with an extremely low-impedance path to ground. The user must provide adequate external decoupling capacitors.
PLLD_GND PLLD1_GND	<b>PLL Ground</b> The PLL ground should be provided with an extremely low-impedance path to ground. The user must provide adequate external decoupling capacitors.
CORE_GND	<b>Core Ground</b> The Core ground should be provided with an extremely low-impedance path to ground. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors.
IO_GND	<b>I/O Ground</b> IO_GND is an isolated ground for the SHIs, ESAIs, Timer I/O and LIBU IO. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors.
GND	<b>Ground</b> This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors.

## 2.2.3 SCAN

Table 2-5 lists the scan signals.

Table 2-5. SCAN Signals

Signal Name	Type	State During Reset	Description
SCAN	Input	Input	<b>SCAN</b> Manufacturing test pin. This pin should be pulled low. Uses internal pull-down resistor.

## 2.2.4 Clock and PLL

Table 2-6 lists the clock and PLL signals.

Table 2-6. Clock and PLL Signals

Signal Name	Type	State During Reset	Description
EXTAL	Input	Input	<b>External Clock / Crystal Input</b> An external clock source must be connected to EXTAL to supply the clock to the internal clock generator and PLL.
XTAL	Output	Chip Driven	<b>Crystal Output</b> Connects the internal Crystal Oscillator output to an external crystal. If an external clock is used, leave XTAL unconnected.
PLOCK	Output	MODC0 Input	<b>PLL Lock/GPIO Port G Pin 0</b> During assertion of $\overline{\text{RESET}}$ , the PLOCK pin acts as a mode pin input, and when $\overline{\text{RESET}}$ is de-asserted, the state of the PLOCK pin is latched into the Core-0 (MDC of Core-0's OMR). After $\overline{\text{RESET}}$ is de-asserted, PLOCK is output "0"; and goes high when the internal PLL is locked.  <b>MODC0</b> MODA0, MODB0, MODC0, and MODD0 levels select one of 16 initial chip operating modes of DSP Core-0, and are latched into the DSP Core-0's OMR when the $\overline{\text{RESET}}$ signal is de-asserted.  <b>GPIO Port G0</b> When the PLOCK is configured as GPIO, this pin is individually programmable as input, output, or internally disconnected. Uses an internal pull-up resistor.
MODC0	Input		
PG0	Input, Output, or Disconnected		
PINIT/ $\overline{\text{NMI}}$	Input	Input	<b>PLL Initial/Nonmaskable Interrupt for DSP Core-0</b> During assertion of $\overline{\text{RESET}}$ , the value of PINIT/ $\overline{\text{NMI}}$ is written into the PLL Enable (PEN) bit of the PLL control register, determining whether the PLL is enabled or disabled. After $\overline{\text{RESET}}$ de-assertion and during normal instruction processing, the PINIT/ $\overline{\text{NMI}}$ Schmitt-trigger input is a negative-edge-triggered nonmaskable interrupt (NMI) request for DSP Core-0, internally synchronized to the internal system clock. Uses an internal pull-up resistor.

## 2.2.5 Reset Pin

Assert Reset to low and then high to force a reset of the DSP cores. [Table 2-7](#) lists the reset pin.

**Table 2-7. Reset Pin**

Signal Name	Type	State During Reset	Description
$\overline{\text{RESET}}$	Input	Input	$\overline{\text{RESET}}$ is an active-low, Schmitt-trigger input. When asserted, the chip is placed in the Reset state and the internal phase generator is reset. The Schmitt-trigger input allows a slowly rising input (such as a capacitor charging) to reset the chip reliably. When the $\overline{\text{RESET}}$ signal is de-asserted, the initial two cores operating modes are latched from the MODA0, MODB0, MODC0, MODD0, MODA1, MODB1, MODC1, and MODD1 inputs. The $\overline{\text{RESET}}$ signal must be asserted during power up. A stable EXTAL signal must be supplied while $\overline{\text{RESET}}$ is being asserted. Uses an internal pull-up resistor.

## 2.2.6 Interrupt and Mode Control

The interrupt and mode control signals select the operating mode of the DSP cores as the cores come out of hardware reset. After  $\overline{\text{RESET}}$  is de-asserted, these inputs are used as hardware interrupt request lines. [Table 2-8](#) lists the interrupt and mode control signals.

**Table 2-8. Interrupt and Mode Control Signals**

Signal Name	Type	State During Reset	Description
MODA0/ $\overline{\text{IRQA}}$	Input	MODA0 Input	<b>Mode Select A0/External Interrupt Request A</b> MODA0/ $\overline{\text{IRQA}}$ is an active-low Schmitt-trigger input, internally synchronized to the DSP clock. MODA0/ $\overline{\text{IRQA}}$ selects the initial Core-0 operating mode during hardware reset, and becomes a two-core shared, level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. This pin can also be programmed as GPIO. MODA0, MODB0, MODC0, and MODD0 levels select one of 16 initial chip operating modes, and are latched into the DSP Core-0's OMR when the $\overline{\text{RESET}}$ signal is de-asserted. If the processor is in the stop standby state and the MODA0/ $\overline{\text{IRQA}}$ pin is pulled to GND, the processor will exit the stop state.
PG5	Input, Output, or Disconnected		<b>GPIO Port G5</b> When the MODA0/ $\overline{\text{IRQA}}$ is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected; and can be controlled by either of the two cores. Uses an internal pull-up resistor.
MODB0/ $\overline{\text{IRQB}}$	Input	MODB0 Input	<b>Mode Select B0/External Interrupt Request B</b> MODB0/ $\overline{\text{IRQB}}$ is an active-low Schmitt-trigger input, internally synchronized to the DSP clock. MODB0/ $\overline{\text{IRQB}}$ selects the initial DSP Core-0 operating mode during hardware reset and becomes a two-core shared, level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. This pin can also be programmed as GPIO. MODA0, MODB0, MODC0, and MODD0 levels select one of 16 initial chip operating modes, and are latched into the DSP Core-0's OMR when the $\overline{\text{RESET}}$ signal is de-asserted.

**Table 2-8. Interrupt and Mode Control Signals (Continued)**

Signal Name	Type	State During Reset	Description
PG6	Input, Output, or Disconnected		<p><b>GPIO Port G6</b></p> <p>When the MODB0/<math>\overline{\text{IRQB}}</math> is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected; and can be controlled by either of the two cores. Uses an internal pull-up resistor.</p>
MODA1/ $\overline{\text{IRQC}}$	Input	MODA1 Input	<p><b>Mode Select A1/External Interrupt Request C</b></p> <p>MODA1/<math>\overline{\text{IRQC}}</math> is an active-low Schmitt-trigger input, internally synchronized to the DSP clock. MODA1/<math>\overline{\text{IRQC}}</math> selects the initial DSP Core-0 operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. This pin can also be programmed as GPIO.</p> <p>MODA1, MODB1, MODC1, and MODD1 levels select one of 16 initial chip operating modes, and are latched into the DSP Core-1 OMR when the <math>\overline{\text{RESET}}</math> signal is de-asserted.</p>
PG7	Input, Output, or Disconnected		<p><b>GPIO Port G7</b></p> <p>When the MODA1/<math>\overline{\text{IRQC}}</math> is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected; and this signal can be controlled by either of the two cores. Uses an internal pull-up resistor.</p>
MODB1/ $\overline{\text{IRQD}}$	Input	MODB1 Input	<p><b>Mode Select B1/External Interrupt Request D</b></p> <p>MODB1/<math>\overline{\text{IRQD}}</math> is an active-low Schmitt-trigger input, internally synchronized to the DSP clock. MODB1/<math>\overline{\text{IRQD}}</math> selects the initial DSP Core-1 operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. This pin can also be programmed as GPIO.</p> <p>MODA1, MODB1, MODC1, and MODD1 levels select one of 16 initial chip operating modes, and are latched into the DSP Core-1 OMR when the <math>\overline{\text{RESET}}</math> signal is de-asserted.</p>
PG8	Input, Output, or Disconnected		<p><b>GPIO Port G8</b></p> <p>When the MODB1/<math>\overline{\text{IRQD}}</math> is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected; and can be controlled by either of the two cores. Uses an internal pull-up resistor.</p>

## 2.2.7 DSP Core-1 Non-Maskable Interrupt ( $\overline{\text{NMI1}}$ )

DSP Core-1 has a dedicated NMI pin, listed in [Table 2-9](#).

**Table 2-9. Non-Maskable Interrupt for DSP Core-1 (NMI1)**

Signal Name	Type	State During Reset	Description
$\overline{\text{NMI1}}$	Input	MODC1 Input	<p><b>Nonmaskable interrupt for DSP Core-1</b> After <math>\overline{\text{RESET}}</math> deassertion and during normal instruction processing, the <math>\overline{\text{NMI1}}</math> Schmitt-trigger input is a negative-edge-triggered nonmaskable interrupt request for DSP Core-1, and is internally synchronized to the internal system clock.</p> <p><b>Operating modes</b> MODA1, MODB1, MODC1, and MODD1 levels select one of 16 initial chip operating modes of DSP Core-1, and are latched into the DSP Core-1's OMR when the <math>\overline{\text{RESET}}</math> signal is de-asserted. Uses an internal pull-up resistor.</p>
MODC1	Input		

## 2.2.8 Serial Host Interface (SHI and SHI\_1)

There are two SHI modules in each DSP56724/DSP56725 device. SHI and SHI\_1. SHI is used by DSP Core-0, while SHI\_1 is used by DSP Core-1. Each of the two SHI modules has five I/O signals that can be configured in either SPI or I<sup>2</sup>C mode.

In the DSP56724 144-pin and DSP56725 80-pin packages, the two SHI modules share one group of SHI pins, with separate  $\overline{\text{SS}}/\text{HA2}$  and  $\overline{\text{SS}}_1/\text{HA2}_1$  pins, shown in [Table 2-10](#).

**Table 2-10. DSP56724/DSP56725 SHI Pin Configuration**

Product	Package	SHI Pin Configuration
DSP56724	144-pin	Both SHI modules (SHI, SHI_1) share one group of SHI pins, except for the $\overline{\text{SS}}$ pin.
DSP56725	80-pin	Both SHI modules (SHI, SHI_1) share one group of SHI pins, except for the $\overline{\text{SS}}$ pin.

Table 2-11 lists the SHI signals.

**Table 2-11. Serial Host Interface Signals (SHI)**

Signal Name	Signal Type	State During Reset	Description
SCK	Input or Output	Tri-stated	<p><b>SPI Serial Clock</b></p> <p>When the SPI is configured as a master, the SCK signal is an output; when the SPI is configured as a slave, the SCK signal is a Schmitt-trigger input.</p> <p>When the SPI is configured as a master, the SCK signal is derived from the internal SHI clock generator.</p> <p>When the SPI is configured as a slave, the SCK signal is an input, and the clock signal from the external master synchronizes the data transfer. The SCK signal is ignored by the SPI if it is defined as a slave and the slave select (<math>\overline{SS}</math>) signal is not asserted.</p> <p>In both the master and slave SPI devices, data is shifted on one edge of the SCK signal and is sampled on the opposite edge where data is stable. Edge polarity is determined by the SPI transfer protocol.</p>
SCL	Input or Output		<p><b>I<sup>2</sup>C Serial Clock</b></p> <p>SCL carries the clock for I<sup>2</sup>C bus transactions in the I<sup>2</sup>C mode. When the SPI is configured as a master, SCL is an open-drain output; when the SPI is configured as a slave, SCL is a Schmitt-trigger input.</p> <p>SCL should be connected to V<sub>DD</sub> through a pull-up resistor.</p> <p>This signal is tri-stated during hardware, software and individual reset. Thus, there is no need for an external pull-up in this state.</p> <p>This pin is shared by SHI and SHI_1 in DSP56725 80-pin and DSP56724 144-pin packages.</p> <p>Uses an internal pull-up resistor.</p>

**Table 2-11. Serial Host Interface Signals (SHI) (Continued)**

Signal Name	Signal Type	State During Reset	Description
MISO	Input or Output	Tri-stated	<p><b>SPI Master-In-Slave-Out</b>                      When the SPI is configured as a master, MISO is the master data input line. The MISO signal is used in conjunction with the MOSI signal for transmitting and receiving serial data.                      When the SPI is configured as a master, MISO is a Schmitt-trigger input; when the SPI is configured as a slave, MISO is an output, and is tri-stated when <math>\overline{SS}</math> is deasserted.                      An external pull-up resistor is not required for SPI operation.</p>
SDA	Input or Open-drain Output	Tri-stated	<p><b>I<sup>2</sup>C Data and Acknowledge</b>                      In I<sup>2</sup>C mode, SDA is a Schmitt-trigger input when receiving and an open-drain output when transmitting. SDA should be connected to V<sub>DD</sub> through a pull-up resistor. SDA carries the data for I<sup>2</sup>C transactions. The data in SDA must be stable during the high period of SCL. The data in SDA is only allowed to change when SCL is low. When the bus is free, SDA is high.                      In start and stop events, the SDA line is only allowed to change during the time SCL is high.                      A start event is a high-to-low transition of the SDA line while SCL is high. A stop event is a low-to-high transition of SDA while SCL is high.</p> <p>This signal is tri-stated during hardware, software and individual reset. Thus, there is no need for an external pull-up in this state.                      This pin is shared by SHI and SHI_1 in DSP56725 80-pin and DSP56724 144-pin packages.                      Uses an internal pull-up resistor.</p>
MOSI	Input or Output	Tri-stated	<p><b>SPI Master-Out-Slave-In</b>                      When the SPI is configured as a master, MOSI is the master data output line. The MOSI signal is used in conjunction with the MISO signal for transmitting and receiving serial data.                      When the SPI is configured as a slave, MOSI is the slave data input line, and is a Schmitt-trigger input.</p>
HA0	Input	Tri-stated	<p><b>I<sup>2</sup>C Slave Address 0</b>                      When configured for I<sup>2</sup>C slave mode, HA0 is used to form the slave device address, and is a Schmitt-trigger input.                      When configured for I<sup>2</sup>C master mode, HA0 is ignored.</p> <p>This signal is tri-stated during hardware, software and individual reset. Thus, there is no need for an external pull-up in this state.                      This pin is shared by SHI and SHI_1 in DSP56725 80-pin and DSP56724 144-pin packages.                      Uses an internal pull-up resistor.</p>



**Table 2-11. Serial Host Interface Signals (SHI) (Continued)**

Signal Name	Signal Type	State During Reset	Description
$\overline{SS}$	Input	Tri-stated	<p><b>SPI Slave Select</b> When configured for SPI Slave mode, <math>\overline{SS}</math> is used to enable the SPI slave for transfer, and is an active low Schmitt-trigger input. When configured for SPI master mode, <math>\overline{SS}</math> should be kept de-asserted (pulled high). If <math>\overline{SS}</math> is asserted while configured in SPI master mode, a bus error condition is flagged. If <math>\overline{SS}</math> is de-asserted while configured in SPI master mode, the SHI ignores SCK clocks and keeps the MISO output signal in the high-impedance state.</p>
HA2	Input		<p><b>I<sup>2</sup>C Slave Address 2</b> When configured for I<sup>2</sup>C slave mode, HA2 is used to form the slave device address, and is a Schmitt-trigger input. When configured for I<sup>2</sup>C master mode, HA2 is ignored.</p> <p>This signal is tri-stated during hardware, software and individual reset. Thus, there is no need for an external pull-up in this state. Uses an internal pull-up resistor.</p>
$\overline{HREQ}$	Input or Output Input, Output, or Disconnected	Tri-stated	<p><b>SHI_1's Host Request</b> When configured for SPI master mode, this signal is an active low Schmitt-trigger input. When asserted by the external slave device, <math>\overline{HREQ}</math> will trigger the start of the data word transfer by the master. After finishing the data word transfer, the master will await the next assertion of <math>\overline{HREQ}</math> to proceed to the next transfer. This pin can also be programmed as GPIO.</p> <p>When configured for SPI slave mode, this signal is an active low output. <math>\overline{HREQ}</math> is asserted to indicate that the SHI is ready for the next data word transfer; <math>\overline{HREQ}</math> is de-asserted at the first clock pulse of the new data word transfer.</p> <p><b>Port H4</b> When <math>\overline{HREQ}</math> is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected.</p> <p>This signal is tri-stated during hardware, software and individual reset, or when the HREQ1-HREQ0 bits in the HCSR are cleared. There is no need for an external pull-up in this state. This pin is shared by SHI and SHI_1 in DSP56725 80-pin and DSP56724 144-pin packages. Uses an internal pull-up resistor.</p>
PH4			

Table 2-12 lists the SHI\_1 signals.

**Table 2-12. Serial Host Interface Signals (SHI\_1)**

Signal Name	Signal Type	State during Reset	Description
$\overline{SS}_1$	Input	Tri-stated	<p><b>SHI_1's SPI Slave Select</b>                      When configured for SPI_1 Slave mode, <math>\overline{SS}_1</math> is used to enable the SPI_1 slave for transfer, and is an active low Schmitt-trigger input.                      When configured for SPI_1 master mode, <math>\overline{SS}_1</math> should be kept de-asserted (pulled high). If <math>\overline{SS}</math> is asserted while configured in SPI_1 master mode, a bus error condition is flagged. If <math>\overline{SS}_1</math> is de-asserted while configured in SPI_1 master mode, the SHI_1 ignores SCK_1 clocks and keeps the MISO_1 output signal in the high-impedance state</p>
HA2_1	Input		<p><b>SHI_1's I<sup>2</sup>C Slave Address 2</b>                      When configured for I<sup>2</sup>C slave mode, HA2_1 is used to form the slave device address, and is a Schmitt-trigger input.                      When configured for I<sup>2</sup>C master mode, HA2_1 is ignored.</p> <p>This signal is tri-stated during hardware, software and individual reset. Thus, there is no need for an external pull-up resistor in this state.                      Uses an internal pull-up resistor.</p>

### 2.2.9 Enhanced Serial Audio Interface Signals (ESAI, ESAI\_1, ESAI\_2, ESAI\_3)

There are four groups of ESAI pins: ESAI, ESAI\_1, ESAI\_2 and ESAI\_3. ESAI and ESAI\_1 pins are used by the DSP Core-0. ESAI\_2 and ESAI\_3 are used by DSP Core-1. Table 2-13 through Table 2-16 show the pins for each ESAI group.

Pin switching between ESAI modules features are supported: ESAI can switch pins with ESAI\_2, and ESAI\_1 can switch pins with ESAI\_3. The switch controls are pin by pin.

**Table 2-13. Enhanced Serial Audio Interface Signals (ESAI)**

Signal Name	Signal Type	State during Reset	Description
HCKR	Input or Output	GPIO Disconnected	<p><b>ESAI's High Frequency Clock for Receiver</b> When programmed as an input, this signal provides a high frequency clock source for the ESAI receiver (as an alternative to the DSP core clock). When programmed as an output, this signal can serve as a high-frequency sample clock (for example, for DACs or as an additional system clock).</p> <p><b>GPIO Port C2</b> When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected.</p> <p><b>S/PDIF Receive Clock</b>— This Pin can be used as S/PDIF receive clock output; this clock is generated by the internal S/PDIF's DPLL, controlled by the ERC0 bits in Pin MUX Control Register of the Chip Configuration Module.</p> <p>The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.</p>
PC2	Input, Output, or Disconnected		
SRCK	Output		
HCKT	Input or Output	GPIO Disconnected	<p><b>ESAI's High Frequency Clock for Transmitter</b> When programmed as an input, this signal provides a high frequency clock source for the ESAI transmitter (as an alternative to the DSP core clock). When programmed as an output, this signal can serve as a high frequency sample clock (for example, for external DACs) or as an additional system clock.</p> <p><b>GPIO Port C5</b> When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected.</p> <p><b>S/PDIF Transmit Clock</b>— This Pin can be used as S/PDIF transmit clock input; controlled by the ClkSrc_Sel bits in the S/PDIF PhaseConfig Register.</p> <p>The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.</p>
PC5	Input, Output, or Disconnected		
STCLK	Input		

**Table 2-13. Enhanced Serial Audio Interface Signals (ESAI) (Continued)**

Signal Name	Signal Type	State during Reset	Description
FSR	Input or Output	GPIO Disconnected	<p><b>ESAI's Frame Sync for Receiver</b>            This is the receiver frame sync input/output signal.            In the asynchronous mode (SYN=0), the FSR pin operates as the frame sync input or output used by all the enabled receivers.            In the synchronous mode (SYN=1), the FSR pin operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1).            When the FSR pin is configured as serial flag pin, its direction is determined by the RFSD bit in the RCCR register.            When configured as the output flag OF1, the FSR pin will reflect the value of the OF1 bit in the SAICR register, and the data in the OF1 bit will show up at the pin synchronized to the frame sync in normal mode or to the slot in network mode.            When configured as the input flag IF1, the data value at the FSR pin will be stored in the IF1 bit in the SAISR register, synchronized by the frame sync in normal mode or by the slot in network mode.</p>
PC1	Input, Output, or Disconnected		<p><b>GPIO Port C1</b>            When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected.            The default state after reset is GPIO disconnected.            Uses an internal pull-down resistor.</p>
FST	Input or Output	GPIO Disconnected	<p><b>ESAI's Frame Sync for Transmitter</b>            This is the transmitter frame sync input/output signal.            For synchronous mode, this signal is the frame sync for both transmitters and receivers.            For asynchronous mode, FST is the frame sync for the transmitters only.            The direction is determined by the transmitter frame sync direction (TFSD) bit in the ESAI transmit clock control register (TCCR).</p>
PC4	Input, Output, or Disconnected		<p><b>GPIO Port C4</b>            When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected.            The default state after reset is GPIO disconnected.            Uses an internal pull-down resistor.</p>

**Table 2-13. Enhanced Serial Audio Interface Signals (ESAI) (Continued)**

Signal Name	Signal Type	State during Reset	Description
SCKR	Input or Output	GPIO Disconnected	<p><b>ESAI's Receiver Serial Clock</b> SCKR provides the receiver serial bit clock for the ESAI. The SCKR operates as a clock input or output used by all the enabled receivers in the asynchronous mode (SYN=0), or as serial flag 0 pin in the synchronous mode (SYN=1). When the SCKR pin is configured as serial flag pin, its direction is determined by the RCKD bit in the RCCR register. When configured as the output flag OF0, the SCKR pin will reflect the value of the OF0 bit in the SAICR register, and the data in the OF0 bit will show up at the pin synchronized to the frame sync in normal mode or to the slot in network mode. When configured as the input flag IF0, the data value at the SCKR pin will be stored in the IF0 bit in the SAISR register, synchronized by the frame sync in normal mode or by the slot in network mode.</p>
PC0	Input, Output, or Disconnected	GPIO Disconnected	<p><b>GPIO Port C0</b> When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.</p>
SCKT	Input or Output	GPIO Disconnected	<p><b>ESAI's Transmitter Serial Clock</b> SCKT provides the serial bit rate clock for the ESAI. SCKT is a clock input or output used by all enabled transmitters and receivers in synchronous mode, or by all enabled transmitters in asynchronous mode.</p>
PC3	Input, Output, or Disconnected	GPIO Disconnected	<p><b>GPIO Port C3</b> When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.</p>
SDO5	Output	GPIO Disconnected	<p><b>ESAI's Serial Data Output 5</b> When programmed as a transmitter, SDO5 is used to transmit data from the TX5 serial transmit shift register.</p>
SDI0	Input	GPIO Disconnected	<p><b>ESAI's Serial Data Input 0</b> When programmed as a receiver, SDI0 is used to receive serial data into the RX0 serial receive shift register.</p>
PC6	Input, Output, or Disconnected	GPIO Disconnected	<p><b>GPIO Port C6</b> When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. After the Reset pin is deasserted, this pin's function is GPIO disconnected. The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.</p>

**Table 2-13. Enhanced Serial Audio Interface Signals (ESAI) (Continued)**

Signal Name	Signal Type	State during Reset	Description
SDO4	Output	GPIO Disconnected	<b>ESAI's Serial Data Output 4</b> When programmed as a transmitter, SDO4 is used to transmit data from the TX4 serial transmit shift register.
SDI1	Input		<b>ESAI's Serial Data Input 1</b> When programmed as a receiver, SDI1 is used to receive serial data into the RX1 serial receive shift register.
PC7	Input, Output, or Disconnected		<b>GPIO Port C7</b> When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. After the Reset pin is deasserted, this pin's function is GPIO disconnected  The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.
SDO3	Output	GPIO Disconnected	<b>ESAI's Serial Data Output 3</b> When programmed as a transmitter, SDO3 is used to transmit data from the TX3 serial transmit shift register.
SDI2	Input		<b>ESAI's Serial Data Input 2</b> When programmed as a receiver, SDI2 is used to receive serial data into the RX2 serial receive shift register.
PC8	Input, Output, or Disconnected		<b>GPIO Port C8</b> When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected.  The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.
SDO2	Output	GPIO Disconnected	<b>ESAI's Serial Data Output 2</b> When programmed as a transmitter, SDO2 is used to transmit data from the TX2 serial transmit shift register
SDI3	Input		<b>ESAI's Serial Data Input 3</b> When programmed as a receiver, SDI3 is used to receive serial data into the RX3 serial receive shift register.
PC9	Input, Output, or Disconnected		<b>GPIO Port C9</b> When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected.  The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.

**Table 2-14. Enhanced Serial Audio Interface Signals (ESAI\_1)**

Signal Name	Signal Type	State during Reset	Description
SDO5_1	Output	GPIO Disconnected	<b>ESAI_1's Serial Data Output 5</b> When programmed as a transmitter, SDO5_1 is used to transmit data from the TX5 serial transmit shift register.
SDI0_1	Input		<b>ESAI_1's Serial Data Input 0</b> When programmed as a receiver, SDI0_1 is used to receive serial data into the RX0 serial receive shift register.
PE6	Input, Output, or Disconnected		<b>GPIO Port E6</b> When the ESAI_1 is configured as GPIO, PE6 is individually programmable as input, output, or internally disconnected.  The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.
SDO4_1	Output	GPIO Disconnected	<b>ESAI_1's Serial Data Output 4</b> When programmed as a transmitter, SDO4_1 is used to transmit data from the TX4 serial transmit shift register.
SDI1	Input		<b>ESAI_1's Serial Data Input 1</b> When programmed as a receiver, SDI1 is used to receive serial data into the RX1 serial receive shift register.
PE7	Input, Output, or Disconnected		<b>GPIO Port E7</b> When the ESAI_1 is configured as GPIO, PE7 is individually programmable as input, output, or internally disconnected.  The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.
SDO3_1	Output	GPIO Disconnected	<b>ESAI_1's Serial Data Output 3</b> When programmed as a transmitter, SDO3_1 is used to transmit data from the TX3 serial transmit shift register.
SDI2_1	Input		<b>ESAI_1's Serial Data Input 2</b> When programmed as a receiver, SDI2_1 is used to receive serial data into the RX2 serial receive shift register.
PE8	Input, Output, or Disconnected		<b>GPIO Port E8</b> When the ESAI_1 is configured as GPIO, PE8 is individually programmable as input, output, or internally disconnected.  The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.

**Table 2-14. Enhanced Serial Audio Interface Signals (ESAI\_1) (Continued)**

Signal Name	Signal Type	State during Reset	Description
SDO2_1	Output	GPIO Disconnected	<b>ESAI_1's Serial Data Output 2</b> When programmed as a transmitter, SDO2_1 is used to transmit data from the TX2 serial transmit shift register.
SDI3_1	Input		<b>ESAI_1's Serial Data Input 3</b> When programmed as a receiver, SDI3_1 is used to receive serial data into the RX3 serial receive shift register.
PE9	Input, Output, or Disconnected		<b>GPIO Port E9</b> When the ESAI_1 is configured as GPIO, PE9 is individually programmable as input, output, or internally disconnected.  The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.

**Table 2-15. Enhanced Serial Audio Interface Signals (ESAI\_2)**

Signal Name	Signal Type	State during Reset	Description
SDO5_2	Output	GPIO Disconnected	<b>ESAI_2's Serial Data Output 5</b> When programmed as a transmitter, SDO5_2 is used to transmit data from the TX5 serial transmit shift register.
SDI0_2	Input		<b>ESAI_2's Serial Data Input 0</b> When programmed as a receiver, SDI0_2 is used to receive serial data into the RX0 serial receive shift register.
PC6_1	Input, Output, or Disconnected		<b>GPIO Port C6_1</b> When the ESAI_2 is configured as GPIO, PC6_1 is individually programmable as input, output, or internally disconnected.  The default state after reset is GPIO disconnected. Uses internal pull-down resistor in the DSP56724 144-pin package. Uses internal pull-up resistor in DSP56725 80-pin package.
SDO4_2	Output	GPIO Disconnected	<b>ESAI_2's Serial Data Output 4</b> When programmed as a transmitter, SDO4_2 is used to transmit data from the TX4 serial transmit shift register.
SDI1	Input		<b>ESAI_2's Serial Data Input 1</b> When programmed as a receiver, SDI1 is used to receive serial data into the RX1 serial receive shift register.
PC7_1	Input, Output, or Disconnected		<b>GPIO Port C7_1</b> When the ESAI_2 is configured as GPIO, PC7_1 is individually programmable as input, output, or internally disconnected.  The default state after reset is GPIO disconnected. Uses internal pull-down resistor in the DSP56724 144-pin package. Uses internal pull-up resistor in DSP56725 80-pin package.



**Table 2-15. Enhanced Serial Audio Interface Signals (ESAI\_2) (Continued)**

Signal Name	Signal Type	State during Reset	Description
SDO3_2	Output	GPIO Disconnected	<b>ESAI_2's Serial Data Output 3</b> When programmed as a transmitter, SDO3_2 is used to transmit data from the TX3 serial transmit shift register.
SDI2_2	Input		<b>ESAI_2's Serial Data Input 2</b> When programmed as a receiver, SDI2_2 is used to receive serial data into the RX2 serial receive shift register.
PC8_1	Input, Output, or Disconnected		<b>GPIO Port C8_1</b> When the ESAI_2 is configured as GPIO, PC8_1 is individually programmable as input, output, or internally disconnected.
SPDIFOUT1	Input		<b>S/PDIF Audio Output Line1</b> — In DSP56725 80-Pin Package, this pin can be used as S/PDIF Output Line 1. Controlled by the spdifout_en bit of the Pin MUX Control Register.  The default state after reset is GPIO disconnected. Uses internal pull-down resistor in the DSP56724 144-pin package. Uses internal pull-up resistor in DSP56725 80-pin package.
SDO2_2	Output	GPIO Disconnected	<b>ESAI_2's Serial Data Output 2</b> When programmed as a transmitter, SDO2_2 is used to transmit data from the TX2 serial transmit shift register
SDI3_2	Input		<b>ESAI_2's Serial Data Input 3</b> When programmed as a receiver, SDI3_2 is used to receive serial data into the RX3 serial receive shift register.
PC9_1	Input, Output, or Disconnected		<b>GPIO Port C9_1</b> When the ESAI_2 is configured as GPIO, PC9_1 is individually programmable as input, output, or internally disconnected.
SPDIFIN1	Input		<b>S/PDIF Audio Input Line1</b> — In DSP56725 80-Pin Package, this pin can be used as S/PDIF Input Line 1. Controlled by the spdifin1_en bit of the Pin MUX Control Register.  The default state after reset is GPIO disconnected. Uses internal pull-down resistor in the DSP56724 144-pin package. Uses internal pull-up resistor in DSP56725 80-pin package.

**Table 2-16. Enhanced Serial Audio Interface Signals (ESAI\_3)**

Signal Name	Signal Type	State during Reset	Description
HCKR_3	Input or Output	GPIO Disconnected	<p><b>ESAI_3's High Frequency Clock for Receiver</b> When programmed as an input, HCKR_3 provides a high frequency clock source for the ESAI receiver (as an alternative to the DSP core clock). When programmed as an output, HCKR_3 can serve as a high-frequency sample clock (for example, for external DACs) or as an additional system clock.</p> <p><b>GPIO Port E2_1</b> When the ESAI_3 is configured as GPIO, PE2_1 is individually programmable as input, output, or internally disconnected.</p> <p><b>S/PDIF Receive Clock</b>— This Pin can be used as S/PDIF receive clock output; this clock is generated by the internal S/PDIF's DPLL, S/PDIF Receive Clock output controlled by the ERC3 bits in Pin MUX Control Register of the Chip Configuration Module.</p> <p>The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.</p>
PE2_1	Input, Output, or Disconnected		
SRCK	Output		
HCKT_3	Input or Output	GPIO Disconnected	<p><b>ESAI_3's High Frequency Clock for Transmitter</b> When programmed as an input, HCKT_3 provides a high frequency clock source for the ESAI_3 transmitter (as an alternative to the DSP core clock). When programmed as an output, HCKT_3 can serve as a high frequency sample clock (for example, for external DACs) or as an additional system clock.</p> <p><b>GPIO Port E5_1</b> When the ESAI_3 is configured as GPIO, PE5_1 is individually programmable as input, output, or internally disconnected.</p> <p><b>S/PDIF Transmit Clock</b>— This Pin can be used as S/PDIF transmit clock input; controlled by the ClkSrc_Sel bits in the S/PDIF PhaseConfig Register.</p> <p>The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.</p>
PE5_1	Input, Output, or Disconnected		
STCLK	Input		

**Table 2-16. Enhanced Serial Audio Interface Signals (ESAI\_3) (Continued)**

Signal Name	Signal Type	State during Reset	Description
FSR_3	Input or Output	GPIO Disconnected	<p><b>ESAI_3's Frame Sync for Receiver</b>            FSR_3 is the receiver frame sync input/output signal. In the asynchronous mode (SYN=0), the FSR_3 pin operates as the frame sync input or output used by all the enabled receivers. In the synchronous mode (SYN=1), the FSR_3 pin operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1).            When the FSR_3 pin is configured as serial flag pin, its direction is determined by the RFSD bit in the RCCR register.            When configured as the output flag OF1, the FSR_3 pin will reflect the value of the OF1 bit in the SAICR register, and the data in the OF1 bit will show up at the pin synchronized to the frame sync in normal mode or to the slot in network mode.            When configured as the input flag IF1, the data value at the FSR_3 pin will be stored in the IF1 bit in the SAISR register, synchronized by the frame sync in normal mode or by the slot in network mode.</p>
PE1_1	Input, Output, or Disconnected		<p><b>GPIO Port E1_1</b>            When the ESAI_3 is configured as GPIO, PE1_1 is individually programmable as input, output, or internally disconnected.             The default state after reset is GPIO disconnected.            Uses an internal pull-down resistor.</p>
FST_3	Input or Output	GPIO Disconnected	<p><b>ESAI_3's Frame Sync for Transmitter</b>            FST_3 is the transmitter frame sync input/output signal. For synchronous mode, FST_3 is the frame sync for both transmitters and receivers. For asynchronous mode, FST_3 is the frame sync for the transmitters only.            The direction is determined by the transmitter frame sync direction (TFSD) bit in the ESAI_3 transmit clock control register (TCCR).</p>
PE4_1	Input, Output, or Disconnected		<p><b>GPIO Port E4_1</b>            When the ESAI_3 is configured as GPIO, PE4_1 is individually programmable as input, output, or internally disconnected.             The default state after reset is GPIO disconnected.            Uses an internal pull-down resistor.</p>

**Table 2-16. Enhanced Serial Audio Interface Signals (ESAI\_3) (Continued)**

Signal Name	Signal Type	State during Reset	Description
SCKR_3	Input or Output	GPIO Disconnected	<p><b>ESAI_3's Receiver Serial Clock</b> SCKR_3 provides the receiver serial bit clock for the ESAI_3. The SCKR_3 pin operates as a clock input or output used by all the enabled receivers in the asynchronous mode (SYN=0), or as serial flag 0 pin in the synchronous mode (SYN=1). When the SCKR_3 pin is configured as serial flag pin, its direction is determined by the RCKD bit in the RCCR register. When configured as the output flag OF0, the SCKR_3 pin will reflect the value of the OF0 bit in the SAICR register, and the data in the OF0 bit will show up at the pin synchronized to the frame sync in normal mode or to the slot in network mode. When configured as the input flag IF0, the data value at the SCKR_3 pin will be stored in the IF0 bit in the SAISR register, synchronized by the frame sync in normal mode or by the slot in network mode.</p>
PE0_1	Input, Output, or Disconnected		<p><b>GPIO Port E0_1</b> When the ESAI_3 is configured as GPIO, PE0_1 is individually programmable as input, output, or internally disconnected.  The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.</p>
SCKT_3	Input or Output	GPIO Disconnected	<p><b>ESAI_3's Transmitter Serial Clock</b> SCKT_3 provides the serial bit rate clock for the ESAI_3. SCKT_3 is a clock input or output used by all enabled transmitters and receivers in synchronous mode, or by all enabled transmitters in asynchronous mode.</p>
PE3_1	Input, Output, or Disconnected		<p><b>GPIO Port E3_1</b> When the ESAI_3 is configured as GPIO, PE3_1 is individually programmable as input, output, or internally disconnected.  The default state after reset is GPIO disconnected. Uses an internal pull-down resistor.</p>
SDO5_3	Output	GPIO Disconnected	<p><b>ESAI_3's Serial Data Output 5</b> When programmed as a transmitter, SDO5_3 is used to transmit data from the TX5 serial transmit shift register.</p>
SDI0_3	Input		<p><b>ESAI_3's Serial Data Input 0</b> When programmed as a receiver, SDI0_3 is used to receive serial data into the RX0 serial receive shift register.</p>
PE6_1	Input, Output, or Disconnected		<p><b>GPIO Port E6_1</b> When the ESAI_3 is configured as GPIO, PE6_1 is individually programmable as input, output, or internally disconnected. After the Reset pin is deasserted, this pins function is GPIO disconnected.  The default state after reset is GPIO disconnected. Uses internal pull-down resistor in the DSP56724 144-pin package. Uses an internal pull-up resistor in DSP56725 80-pin package.</p>

**Table 2-16. Enhanced Serial Audio Interface Signals (ESAI\_3) (Continued)**

Signal Name	Signal Type	State during Reset	Description
SDO4_3	Output	GPIO Disconnected	<b>ESAI_3's Serial Data Output 4</b> When programmed as a transmitter, SDO4_3 is used to transmit data from the TX4 serial transmit shift register.
SDI1	Input		<b>ESAI_3's Serial Data Input 1</b> When programmed as a receiver, SDI1 is used to receive serial data into the RX1 serial receive shift register.
PE7_1	Input, Output, or Disconnected		<b>GPIO Port E7_1</b> When the ESAI_3 is configured as GPIO, PE7_1 is individually programmable as input, output, or internally disconnected. After Reset pin is deasserted, this pins function is GPIO disconnected  The default state after reset is GPIO disconnected.
SDO3_3	Output	GPIO Disconnected	<b>ESAI_3's Serial Data Output 3</b> When programmed as a transmitter, SDO3_3 is used to transmit data from the TX3 serial transmit shift register.
SDI2_3	Input		<b>ESAI_3's Serial Data Input 2</b> When programmed as a receiver, SDI2_3 is used to receive serial data into the RX2 serial receive shift register.
PE8_1	Input, Output, or Disconnected		<b>GPIO Port E8_1</b> When the ESAI_3 is configured as GPIO, PE8_1 is individually programmable as input, output, or internally disconnected.  The default state after reset is GPIO disconnected. Uses an internal pull-down resistor in the DSP56724 144-pin package. Uses an internal pull-up resistor in the DSP56725 80-pin package.
SDO2_3	Output	GPIO Disconnected	<b>ESAI_3's Serial Data Output 2</b> When programmed as a transmitter, SDO2_3 is used to transmit data from the TX2 serial transmit shift register
SDI3_3	Input		<b>ESAI_3's Serial Data Input 3</b> When programmed as a receiver, SDI3_3 is used to receive serial data into the RX3 serial receive shift register.
PE9_1	Input, Output, or Disconnected		<b>GPIO Port E9_1</b> When the ESAI_3 is configured as GPIO, PE9_1 is individually programmable as input, output, or internally disconnected.  The default state after reset is GPIO disconnected. Uses internal pull-down resistor in the DSP56724 144-pin package. Uses an internal pull-up resistor in the DSP56725 80-pin package.

## 2.2.10 Watch Dog Timer (WDT)

The block  $\overline{\text{WDT}}$  pin is used by DSP Core-0, shown in [Table 2-17](#).

**Table 2-17. WDT Signal**

Signal Name	Type	State after Reset	Description
$\overline{\text{WDT}}$	Output	WDT output	This signal is asserted low when the hardware watchdog timer counts down to zero. This pin is controlled by both WDT and WDT_1 modules, and is asserted when the watchdog timer counts down to zero in either WDT or WDT_1 modules.

## 2.2.11 External Memory Controller (EMC)

There is an external memory interface controller (EMC) in the DSP56724. (There is no EMC in the DSP56725.) The EMC module is shared by both DSP cores. All of the EMC signals are available in the DSP56724 144-pin package. [Table 2-18](#) lists the EMC signals.

**Table 2-18. External Memory: EMC Signals**

Signal Name	Type	State during Reset	Description
LALE	Output	LALE function	<p><b>External Address Latch Enable</b></p> <p>The EMC provides control for an external address latch, which allows address and data to be multiplexed on the device pins.</p> <p><i>Asserted</i>—LALE is asserted for at least 1/2 bus clock cycle for each memory-controller transaction. If <math>\text{ORx}[\text{EAD}] = 1</math>, LALE is asserted for <math>(N+1/2)</math> bus clock cycles. <math>N</math> is the number of bus clock cycles defined by the EADC field in the CRR. Note that no other control signals are asserted when LALE is asserted.</p> <p><i>Negated</i>—LALE is negated at the negative edge of bus clock during address phase. LALE is negated 1/2 bus clock cycle earlier than the next positive edge of bus clock, to get additional hold time for external latch device.</p> <p>Uses an internal pull-down resistor.</p>
$\overline{\text{LCS}}[7:0]$	Output	$\overline{\text{LCS}}[7:0]$ function	<p><b>Chip Selects</b></p> <p>Eight mutually exclusive chip selects are provided.</p> <p><i>Asserted/Negated</i>—Used to enable specific memory devices or peripherals connected to the EMC. <math>\overline{\text{LCS}}[7:0]</math> are provided on a per-bank basis, with <math>\overline{\text{LCS}}0</math> corresponding to the chip select for memory bank 0, which has the memory type and attributes defined by BR0 and OR0.</p> <p>Uses an internal pull-up resistor.</p>
$\overline{\text{LWE}}/\text{LSDDQM}$	Output	$\overline{\text{LWE}}/\text{LSDDQM}$	<p><b>GPCM Write Enable / SDRAM Data Mask</b></p> <p><i>Asserted/Negated</i>—For GPCM operation, <math>\overline{\text{LWE}}</math> is asserted for writing. For SDRAM operation, LSDDQM functions as the DQM or data mask signals provided by JEDEC-compliant SDRAM devices. When the EMC wishes to mask a write or disable read data output from the SDRAM, LSDDQM is driven high.</p> <p>Uses an internal pull-up resistor.</p>

Table 2-18. External Memory: EMC Signals (Continued)

Signal Name	Type	State during Reset	Description
LSDA10/ LGPL0	Output	LSDA10/ LGPL0	<b>SDRAM A10 / General Purpose Line 0</b> <i>Asserted/Negated</i> —For SDRAM accesses, this signal represents address bit 10. When the row address is driven, this signal drives the value of address bit 10. When the column address is driven, this signal forms part of the SDRAM command. This signal is one of six general purpose signals when in UPM mode and drives a value programmed in the UPM array. Uses an internal pull-up resistor.
$\overline{\text{LSDWE}}$ / LGPL1	Output	$\overline{\text{LSDWE}}$ / LGPL1	<b>SDRAM Write Enable / General-Purpose Line 1</b> <i>Asserted/Negated</i> —This signal is connected to the SDRAM device WE input and acts as the SDRAM write enable when accessing SDRAM. This signal is one of six general purpose signals when in UPM mode, and drives a value programmed in the UPM array. Uses an internal pull-up resistor.
$\overline{\text{LOE}}$ / LSDRAS/ LGPL2	Output	$\overline{\text{LOE}}$ / LSDRAS/ LGPL2	<b>GPCM Output Enable / SDRAM RAS / General-Purpose Line 2</b> <i>Asserted/Negated</i> —This signal controls the output buffer of memory when accessing memory/devices in GPCM mode. For SDRAM accesses, this signal is the row address strobe (RAS). This signal is one of six general purpose lines when in UPM mode, and drives a value programmed in the UPM array. Uses an internal pull-up resistor.
$\overline{\text{LSDCAS}}$ / LGPL3	Output	$\overline{\text{LSDCAS}}$ / LGPL3	<b>SDRAM CAS / General-Purpose Line 3</b> <i>Asserted/Negated</i> —In SDRAM mode, drives the column address strobe (CAS). This signal is one of six general purpose signals when in UPM mode, and drives a value programmed in the UPM array. Uses an internal pull-up resistor.
$\overline{\text{LGTA}}$ / LGPL4/ UPWAIT	Input/Output	$\overline{\text{LGTA}}$ / LGPL4/ UPWAIT	<b>GPCM Terminate Access / General-Purpose Line 4 / UPM Wait</b> <i>Asserted/Negated</i> —This signal is an input in GPCM mode and is used for transaction termination. This signal may also be configured as one of six general purpose output signals when in UPM mode or as an input to force the UPM controller to wait for the memory/device. Uses an internal pull-up resistor.
LGPL5	Input/Output	LGPL5	<b>General-Purpose Line 5</b> <i>Asserted/Negated</i> —This signal is one of six general purpose signals when in UPM mode, and drives a value programmed in the UPM array. Uses an internal pull-up resistor.
LBCTL	Output	LBCTL	<b>Data Buffer Control</b> When a GPCM- or UPM-controlled bank is accessed, the memory controller activates a data buffer control signal (BCTL) for the external memory. Access to an SDRAM machine-controlled bank does not activate the buffer control. The buffer control can be disabled by setting ORx[BCTLD]. <i>Asserted/Negated</i> —The LBCTL pin normally functions as a Write/ $\overline{\text{Read}}$ control for a bus transceiver connected to the LAD lines. Because LBCTL remains high after reset and during address phases, an external data buffer must not drive the LAD lines in conflict with the EMC when LBCTL is high. Uses an internal pull-up resistor.

**Table 2-18. External Memory: EMC Signals (Continued)**

Signal Name	Type	State during Reset	Description
LA[2:0]	Output	GPIO Disconnected	<b>External Memory Non-Multiplexed Address LSBs</b> All bits driven on LA[2:0] are defined for 24-bit port sizes. <i>Asserted/Negated</i> —Even though the EMC shares an address and data bus, up to three least significant bits of the RAM address always appear on the dedicated address pins LA[2:0]. These may be used, unlatched, in place of LAD[2:0], to connect the three least significant bits of the address for address phases. For some RAM devices, such as fast-page DRAM, LA[2:0] serve as the column address offset during a burst access.
PA[26:24]	Input or Output or Disconnected		<b>GPIO Port A, Pin26–Pin24</b> When the EMC is configured as GPIO, these signals are individually programmable as input, output, or internally disconnected. PA24 is multiplexed with LA0; PA25 is multiplexed with LA1; PA26 is multiplexed with LA2. The default state after reset for these signals is GPIO disconnected. Internal Pull-Down Resistor for these 3 signals.
LAD[23:0]	Input/Output	GPIO Disconnected	<b>Multiplexed Address and Data Bus</b> For configuration of a port size in BRx[PS] as 32 bits, all of LAD[23:0] needs to be connected to the external RAM data bus, with LAD[23] occupying the most significant bit. <i>Asserted/Negated</i> —LAD[23:0] is the shared 24-bit address and data bus through which external RAM devices transfer data and receive addresses. <i>Assertion/Negation</i> —During assertion of LALE, LAD[23:0] are driven with the RAM address for the access to follow. External logic should propagate the address on LAD[23:0] while LALE is asserted, and latch the address upon negation of LALE. After LALE is negated, LAD[23:0] are either driven by write data or are made high-impedance by the EMC to sample read data driven by an external device. Following the last data transfer of a write access, LAD[23:0] are again taken into a high-impedance state.
PA[23:0]	Input or Output or Disconnected	—	<b>GPIO Port A, Pin[23:0]</b> When the EMC is configured as GPIO, these signals are individually programmable as input, output, or internally disconnected. PA23 is multiplexed with LAD23, PA22 is multiplexed with LAD22,..., and PA0 is multiplexed with LAD0. The default state after reset for these signals is GPIO disconnected. Internal Pull-Down Resistor for these signals.
LCKE	Output	LCKE	<b>External Memory Clock Enable</b> <i>Asserted/Negated</i> —LCKE is the bus clock enable signal (CKE) for JEDEC-standard SDRAM devices. This signal is asserted during normal SDRAM operation. Uses an internal pull-up resistor.
LCLK	Output	LCLK	<b>External Memory Clocks</b> <i>Asserted/Negated</i> —LCLK drive external memory clock signal. If the EMC phase-locked loop (PLL) is enabled (see CRR[DBYP]), the bus clock phase is shifted earlier than transitions on other EMC signals (such as LAD[23:0] and LCSx) by a time delay matching the delay of the PLL timing loop set up between LSYNC_OUT and LSYNC_IN. Uses an internal pull-down resistor.



**Table 2-18. External Memory: EMC Signals (Continued)**

Signal Name	Type	State during Reset	Description
LSYNC_OUT	Output	LSYNC_OUT	<p><b>PLL Synchronization Out</b>  <i>Asserted/Negated</i>—A replica of the bus clock, appearing on LSYNC_OUT should be propagated through a passive timing loop and returned to LSYNC_IN for achieving correct PLL lock.  <i>Assertion/Negation</i>—The time delay of the timing loop should be such that it compensates for the round-trip flight time of LCLK and clocked drivers in the system. No load other than a timing loop should be placed on LSYNC_OUT. Uses an internal pull-down resistor.</p>
LSYNC_IN	Input	LSYNC_IN	<p><b>PLL Synchronization Input</b>  <i>Asserted/Negated</i>—See the description of LSYNC_OUT. Uses an internal pull-down resistor.</p>

## 2.2.12 S/PDIF Audio Interface Signals

Table 2-19 lists the S/PDIF signals.

**Table 2-19. Digital Audio Interface: S/PDIF Signals**

Signal Name	Type	State During Reset	Description
SPDIFIN1  PG9	Input  Input, Output or Disconnected	GPIO Disconnected	<p><b>S/PDIF Input Line 1</b> IEC958 data in biphase mark format.</p> <p><b>GPIO Port G9</b> When the S/PDIF is configured as GPIO, these signals are individually programmable as input, output, or internally disconnected.</p> <p>This signal can be used by the GPIO port G function. GPIO functions are controlled by GPIO port G registers. There is a bit in the chip configuration registers that set it to S/PDIF.</p> <p>The default state after reset for these signals is GPIO disconnected. Uses an internal pull-down resistor.</p>
SPDIFOUT1  PG13	Input  Input or Output or Disconnected	GPIO Disconnected	<p><b>Audio Data Output Line 1</b> IEC958 data in biphase mark format. (Consumer C channel).</p> <p><b>GPIO Port G13</b> When the S/PDIF is configured as GPIO, these signals are individually programmable as input, output, or internally disconnected.</p> <p>This signal can be used by the GPIO port G function. GPIO functions are controlled by GPIO port G registers. There is a bit in the chip configuration registers that set it to S/PDIF.</p> <p>The default state after reset for these signals is GPIO disconnected. Uses an internal pull-down resistor.</p>

## 2.2.13 Dedicated Port G GPIOs

Table 2-20. Dedicated Port G Signals and Mode Pins

Signal Name	Type	State During Reset	Description
PG1	Input, Output, or Disconnected	MODD0 Input	<b>Port G1</b> When the PLOCK is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected.
MODD0	Input		MODA0 MODB0, MODC0, and MODD0 levels select one of 16 initial chip operating modes of DSP Core-0, and are latched into the DSP Core-0's OMR when the $\overline{\text{RESET}}$ signal is deasserted. Uses an internal pull-down resistor. This signal is only available in the DSP56724 144-pin package. This signal is not available in the DSP56725 80-pin package.
PG2	Input, Output, or Disconnected	MODD1 Input	<b>Port G2</b> When the PLOCK is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected.
MODD1	Input		MODA1, MODB1, MODC1, and MODD1 levels select one of 16 initial chip operating modes of DSP Core-1, and are latched into the DSP Core-1's OMR when the $\overline{\text{RESET}}$ signal is deasserted. Uses an internal pull-down resistor. This signal is only available in the DSP56724 144-pin package. This signal is not available in the DSP56725 80-pin package.

## 2.2.14 JTAG/OnCE Interface Signals

Table 2-21 list the JTAGE/OnCE interface signals.

Table 2-21. JTAG/OnCE Interface

Signal Name	Signal Type	State During Reset	Description
TCK	Input	Input	<b>Test Clock</b> TCK is a test clock input signal used to synchronize the JTAG test logic. It uses an internal pull-up resistor.
TDI	Input	Input	<b>Test Data Input</b> TDI is a test data serial input signal used for test instructions and data. TDI is sampled on the rising edge of TCK and uses an internal pull-up resistor.
TDO	Output	Tri-States	<b>Test Data Output</b> TDO is a test data serial output signal used for test instructions and data. TDO is tri-statable and is actively driven in the shift-IR and shift-DR controller states. TDO changes on the falling edge of TCK.
TMS	Input	Input	<b>Test Mode Select</b> TMS is an input signal used to sequence the test controller's state machine. TMS is sampled on the rising edge of TCK and uses an internal pull-up resistor.

# Chapter 3 Memory Map

## 3.1 Overview

The memory space of each DSP56300 core is partitioned into three main parts: program memory space, X data memory space, and Y data memory space. The data memory space is divided into X and Y data memory to work with the two address ALUs and to feed two operands simultaneously to the data ALU. Memory space includes internal RAM and ROM, and can be expanded off-chip.

Figure 3-1 shows the memory address allocations for the DSP56724 and DSP56725 DSPs.

	Program Memory Space	X Data Memory Space	Y Data Memory Space	
	Program ROM	Internal I/O	Internal I/O	0xFFFFF
		X ROM	Y ROM	0xFFFF0
0xF8000	External Memory	External Memory	External Memory	
0x04000				
	Reserved	Reserved	Reserved	0x04000
0x03800	Shared Memory	Shared Memory	Shared Memory	0x03800
0x03000				
0x00000	Internal RAM	Internal RAM	Internal RAM	

**Figure 3-1. DSP56724 and DSP56725 Memory Address Allocation**

In the DSP56724/DSP56725, the lowest addresses are used for on-chip internal Program, X, and Y RAM. The DSP56724 supports using a memory switch mode to increase the size of program RAM as needed. Some X RAM bank(s) and/or Y RAM bank(s) can be switched to Program RAM. Program, X, and Y RAM allocation on each DSP can be configured as five types of memory maps: a default mode plus four additional memory maps based on two bit settings (MSW0, MSW1).

Eight blocks of 4 Kwords shared memory (RAM) are accessible in the DSP56724/DSP56725. The shared memory blocks occupy addresses from 0x030000 to 0x037FFF (including 0x037FFF), accessible by both DSP cores. When the DSP cores access the shared memory, the Program, X, and Y memory addresses are mapped into same physical location, which means that there is no difference in accessing the shared memory from Program, X, or Y memory space.

## Memory Map

External Memory addresses from 0x040000 are used for internal RAM/ROM expansion. In the DSP56724, the expansion is implemented via the EMC module, using the EMC's external interface signals.

On-chip peripherals are connected to the core using the peripheral bus or Shared Peripheral bus. The on-chip peripherals use the addresses above 0xFFF000 (including 0xFFF000).

### 3.2 Data and Program Memory Maps

The on-chip memory configuration for each DSP is affected by the state of the memory switch control bits in the Operating Mode Register (OMR). These bits are the Master Memory Switch Mode (MS) bit, the Memory Switch Mode 0 (MSW0) bit, and the Memory Switch Mode 1 (MSW1) bit.

**Table 3-1. Core-0 Configuration**

Bit Settings			Memory Space		
MSW1	MSW0	MS	Program RAM	X Data RAM	Y Data RAM
—	—	0	4 Kwords	28 Kwords	24 Kwords
0	0	1	40 Kwords	8 Kwords	8 Kwords
0	1	1	24 Kwords	16 Kwords	16 Kwords
1	0	1	16 Kwords	24 Kwords	16 Kwords
1	1	1	8 Kwords	24 Kwords	24 Kwords

**Table 3-2. Core-1 Configuration**

Bit Settings			Memory Space		
MSW1	MSW0	MS	Program RAM	X Data RAM	Y Data RAM
—	—	0	2 Kwords	12 Kwords	10 Kwords
0	0	1	16 Kwords	4 Kwords	4 Kwords
0	1	1	12 Kwords	8 Kwords	4 Kwords
1	0	1	8 Kwords	8 Kwords	8 Kwords
1	1	1	4 Kwords	12 Kwords	8 Kwords

**Table 3-3. DSP Core-0 Memory Map Locations**

Configuration	Program RAM	X Data RAM	Y Data RAM
MSW = NA, MS = 0	4 Kwords	28 Kwords	24 Kwords
	1 × 4-Kword block	1 × 4-Kword block 3 × 8-Kword block	3 × 8-Kword block
	0x000000–0x000FFF	0x000000–0x006FFF	0x000000–0x005FFF
MSW1 = 1, MSW0 = 1, MS = 1	8 Kwords	24 Kwords	24 Kwords
	0x000000–0x001FFF	0x000000–0x005FFF	0x000000–0x005FFF

**Table 3-3. DSP Core-0 Memory Map Locations (Continued)**

Configuration	Program RAM	X Data RAM	Y Data RAM
MSW1 = 1, MSW0 = 0, MS = 1	16 Kwords	24 Kwords	16 Kwords
	0x000000–0x003FFF	0x000000–0x005FFF	0x000000–0x003FFF
MSW1 = 0, MSW0 = 1, MS = 1	24 Kwords	16 Kwords	16 Kwords
	0x000000–0x005FFF	0x000000–0x003FFF	0x000000–0x003FFF
MSW1 = 0, MSW0 = 0, MS = 1	40 Kwords	8 Kwords	8 Kwords
	0x000000–0x009FFF	0x000000–0x001FFF	0x000000–0x001FFF

**Table 3-4. DSP Core-1 Memory Map Locations**

Configuration	Program RAM	X Data RAM	Y Data RAM
MSW = NA, MS = 0	2 Kwords	12 Kwords	10 Kwords
	1 × 2-Kword block	3 × 4-Kword block	1 × 4-Kword block
	0x000000–0x0007FF	0x000000–0x002FFF	0x000000–0x0027FF
MSW1 = 1, MSW0 = 1, MS = 1	4 Kwords	12 Kwords	8 Kwords
	0x000000–0x000FFF	0x000000–0x002FFF	0x000000–0x001FFF
MSW1 = 1, MSW0 = 0, MS = 1	8 Kwords	8 Kwords	8 Kwords
	0x000000–0x001FFF	0x000000–0x001FFF	0x000000–0x001FFF
MSW1 = 0, MSW0 = 1, MS = 1	12 Kwords	8 Kwords	4 Kwords
	0x000000–0x002FFF	0x000000–0x001FFF	0x000000–0x000FFF
MSW1 = 0, MSW0 = 0, MS = 1	16 Kwords	4 Kwords	4 Kwords
	0x000000–0x003FFF	0x000000–0x000FFF	0x000000–0x000FFF

### 3.3 Peripheral Register Memory Map

The dedicated and shared peripherals for each DSP core are the same, and the peripherals' register memory map for both cores are the same.

DSP Core-0's on-chip peripherals X-Memory map are listed in [Table 3-7](#) DSP Core-0's on-chip peripheral's Y-Memory Map is listed in [Table 3-8](#).

The X-Memory map and Y-Memory map of DSP Core-1 are essentially the same as the DSP Core-0; the difference being the names of the dedicated peripherals. For example, DSP Core-0 is PIC, while DSP Core-1 is PIC\_1.

## Memory Map

Legend for the following tables: **Yellow** indicates a dedicated peripheral, while **blue** indicates a shared peripheral.

**Table 3-5. X-Memory Map for DSP Core-0 and Core-1**

Address Range	Blocks	
	DSP Core-0	DSP Core-1
X: 0xFFFFF7F-0xFFFF7C	PIC	PIC_1
X: 0xFFFF7B-0xFFFF78	CIM	CIM_1
X: 0xFFFF77-0xFFFF60	PIC	PIC_1
X: 0xFFFF6F-0xFFFF65	CIM	CIM_1
X: 0xFFFF64-0xFFFFD0	DMA Control and DMA channels	DMA_1 Control and DMA_1 channels
X: 0xFFFFBF-0xFFFFA0	ESAI, GPIO PORT C	ESAI_2, GPIO PORT C1
X: 0xFFFF9A-0xFFFF98	GPIO port H	GPIO port H
X: 0xFFFF97-0xFFFF90	SHI	SHI_1
X: 0xFFFF8F-0xFFFF80	Triple Timer (TEC)	Triple Timer (TEC_1)
X: 0xFFFF7F-0xFFFF7C	CGM	
X: 0xFFFF7B-0xFFFF78	Reserved	
X: 0xFFFF77-0xFFFF60	S/PDIF	
X: 0xFFFF5F-0xFFFE6C	Reserved	
X: 0xFFFE6B-0xFFFE00	EMC	
X: 0xFFFDFF-0xFFE000	Reserved	

**Table 3-6. Y-Memory Map for DSP Core-0 and Core-1**

Address Range	Blocks	
	DSP Core-0	DSP Core-1
Y: 0xFFFFF7F-0xFFFFF8	GPIO Port G	
Y: 0xFFFFF77-0xFFFFF0	GPIO Port A	
Y: 0xFFFFFEF-0xFFFFE8	Reserved	
Y: 0xFFFFFE7-0xFFFFE0	Chip Configuration Registers	
Y: 0xFFFFDF-0xFFFFDC	Reserved	Reserved
Y: 0xFFFFDB-0xFFFFD0	ICC	ICC
Y: 0xFFFFCF-0xFFFFCB	Reserved	Reserved
Y: 0xFFFFCA	ESAI/ESIA_1 internal clock control	ESAI_2/ESIA_3 internal clock control
Y: 0xFFFFC9	Reserved	Reserved
Y: 0xFFFFC8	EMC/ICC Error Status Register	EMC/ICC Error Status Register
Y: 0xFFFFC8-0xFFFFC4	Reserved	Reserved

**Table 3-6. Y-Memory Map for DSP Core-0 and Core-1 (Continued)**

	Address Range	Blocks	
		DSP Core-0	DSP Core-1
	Y: 0xFFFFC3–0xFFFFC0	WDT	WDT_1
	Y: 0xFFFFBF–0xFFFFB0	Reserved	Reserved
	Y: 0xFFFFAF–0xFFFFA0	Reserved	Reserved
	Y: 0xFFFF9F–0xFFFF80	ESAI_1	ESAI_3
	Y: 0xFFFF7F–0xFFFC40	Reserved	
	Y: 0xFFFC3F–0xFFFC00	ASRC	
	Y: 0xFFFBFF–0xFFE000	Reserved	

**Table 3-7. Detailed Device X-Memory Map**

Peripherals	Address	Register Name <sup>1</sup>
PIC, PIC_1	X: 0xFFFFF7	Interrupt Priority Register Core (IPR-C)
	X: 0xFFFFF6	Interrupt Priority Register Peripheral (IPR-P)
	X: 0xFFFFF5	Reserved
CIM, CIM_1	X: 0xFFFFF4	OnCE Global Data Register (OGDB)
PIC, PIC_1	X: 0xFFFFF3	Interrupt Priority Register Core (IPR_C1)
	X: 0xFFFFF2	Interrupt Priority Register Peripheral (IPR_P1)
	X: 0xFFFFF1	Reserved
CIM, CIM_1	X: 0xFFFFF0	DMA stall register (DMAS).
	X: 0xFFFFE8	Reserved
	X: 0xFFFFE7	CHIP ID Register (CHIDR)
DMA, DMA_1	X: 0xFFFFE6	DMA Status Register (DSTR)
	X: 0xFFFFE5	DMA Offset Register 0 (DOR0)
	X: 0xFFFFE4	DMA Offset Register 1 (DOR1)
	X: 0xFFFFE3	DMA Offset Register 2 (DOR2)
	X: 0xFFFFE2	DMA Offset Register 3 (DOR3)
DMA, DMA_1 Channel 0	X: 0xFFFFE1	DMA Source Address Register (DSR0)
	X: 0xFFFFE0	DMA Destination Address Register (DDR0)
	X: 0xFFFFDF	DMA Counter (DCO0)
	X: 0xFFFFDE	DMA Control Register (DCR0)

**Table 3-7. Detailed Device X-Memory Map (Continued)**

Peripherals	Address	Register Name <sup>1</sup>
DMA, DMA_1 Channel 1	X: 0xFFFFFEB	DMA Source Address Register (DSR1)
	X: 0xFFFFFEA	DMA Destination Address Register (DDR1)
	X: 0xFFFFFE9	DMA Counter (DCO1)
	X: 0xFFFFFE8	DMA Control Register (DCR1)
DMA, DMA_1 Channel 2	X: 0xFFFFFE7	DMA Source Address Register (DSR2)
	X: 0xFFFFFE6	DMA Destination Address Register (DDR2)
	X: 0xFFFFFE5	DMA Counter (DCO2)
	X: 0xFFFFFE4	DMA Control Register (DCR2)
DMA, DMA_1 Channel 3	X: 0xFFFFFE3	DMA Source Address Register (DSR3)
	X: 0xFFFFFE2	DMA Destination Address Register (DDR3)
	X: 0xFFFFFE1	DMA Counter (DCO3)
	X: 0xFFFFFE0	DMA Control Register (DCR3)
DMA, DMA_1 Channel 4	X: 0xFFFFFDF	DMA Source Address Register (DSR4)
	X: 0xFFFFFDE	DMA Destination Address Register (DDR4)
	X: 0xFFFFFDD	DMA Counter (DCO4)
	X: 0xFFFFFDC	DMA Control Register (DCR4)
DMA, DMA_1 Channel 5	X: 0xFFFFFDB	DMA Source Address Register (DSR5)
	X: 0xFFFFFDA	DMA Destination Address Register (DDR5)
	X: 0xFFFFFD9	DMA Counter (DCO5)
	X: 0xFFFFFD8	DMA Control Register (DCR5)
DMA, DMA_1 Channel 6	X: 0xFFFFFD7	DMA Source Address Register (DSR6)
	X: 0xFFFFFD6	DMA Destination Address Register (DDR6)
	X: 0xFFFFFD5	DMA Counter (DCO6)
	X: 0xFFFFFD4	DMA Control Register (DCR6)
DMA, DMA_1 Channel 7	X: 0xFFFFFD3	DMA Source Address Register (DSR7)
	X: 0xFFFFFD2	DMA Destination Address Register (DDR7)
	X: 0xFFFFFD1	DMA Counter (DCO7)
	X: 0xFFFFFD0	DMA Control Register (DCR7)
GPIO Port C, Port C1	X: 0xFFFFFBF	PORT C/C1 Control Register (PCRC)
	X: 0xFFFFFBE	PORT C/C1 Direction Register (PRRC)
	X: 0xFFFFFBD	PORT C/C1 GPIO Data Register (PDRC)



**Table 3-7. Detailed Device X-Memory Map (Continued)**

Peripherals	Address	Register Name <sup>1</sup>
ESAI, ESAI_2	X: 0xFFFFBC	ESAI/ESAI_2 Receive Slot Mask Register B (RSMB)
	X: 0xFFFFBB	ESAI/ESAI_2 Receive Slot Mask Register A (RSMA)
	X: 0xFFFFBA	ESAI/ESAI_2 Transmit Slot Mask Register B (TSMB)
	X: 0xFFFFB9	ESAI/ESAI_2 Transmit Slot Mask Register A (TSMA)
	X: 0xFFFFB8	ESAI/ESAI_2 Receive Clock Control Register (RCCR)
	X: 0xFFFFB7	ESAI/ESAI_2 Receive Control Register (RCR)
	X: 0xFFFFB6	ESAI/ESAI_2 Transmit Clock Control Register (TCCR)
	X: 0xFFFFB5	ESAI/ESAI_2 Transmit Control Register (TCR)
	X: 0xFFFFB4	ESAI/ESAI_2 Common Control Register (SAICR)
	X: 0xFFFFB3	ESAI/ESAI_2 Status Register (SAISR)
	X: 0xFFFFB2–X: 0xFFFFAC	Reserved
	X: 0xFFFFAB	ESAI/ESAI_2 Receive Data Register 3 (RX3)
	X: 0xFFFFAA	ESAI/ESAI_2 Receive Data Register 2 (RX2)
	X: 0xFFFFA9	ESAI/ESAI_2 Receive Data Register 1 (RX1)
	X: 0xFFFFA8	ESAI/ESAI_2 Receive Data Register 0 (RX0)
	X: 0xFFFFA7	Reserved
	X: 0xFFFFA6	ESAI/ESAI_2 Time Slot Register (TSR)
ESAI, ESAI_2	X: 0xFFFFA5	ESAI/ESAI_2 Transmit Data Register 5 (TX5)
	X: 0xFFFFA4	ESAI/ESAI_2 Transmit Data Register 4 (TX4)
	X: 0xFFFFA3	ESAI/ESAI_2 Transmit Data Register 3 (TX3)
	X: 0xFFFFA2	ESAI/ESAI_2 Transmit Data Register 2 (TX2)
	X: 0xFFFFA1	ESAI/ESAI_2 Transmit Data Register 1 (TX1)
	X: 0xFFFFA0	ESAI/ESAI_2 Transmit Data Register 0 (TX0)
	X: 0xFFFF9F–X: 0xFFFF9B	Reserved
GPIO port H/H1	X: 0xFFFF9A	Port H Control Register (PCRH)
	X: 0xFFFF99	Port H Direction Register (PRRH)
	X: 0xFFFF98	Port H GPIO Data Register (PDRH)
SHI, SHI_1	X: 0xFFFF97–X: 0xFFFF95	Reserved
	X: 0xFFFF94	SHI Receive FIFO (HRX)
	X: 0xFFFF93	SHI Transmit Register (HTX)
	X: 0xFFFF92	SHI I2C Slave Address Register (HSAR)
	X: 0xFFFF91	SHI Control/Status Register (HCSR)
	X: 0xFFFF90	SHI Clock Control Register (HCKR)

**Table 3-7. Detailed Device X-Memory Map (Continued)**

Peripherals	Address	Register Name <sup>1</sup>
TEC, TEC_1	X: 0xFFFF8F	Timer 0 Control/Status Register (TCSR0)
	X: 0xFFFF8E	Timer 0 Load Register (TLR0)
	X: 0xFFFF8D	Timer 0 Compare Register (TCPR0)
	X: 0xFFFF8C	Timer 0 Count Register (TCR0)
	X: 0xFFFF8B	Timer 1 Control/Status Register (TCSR1)
	X: 0xFFFF8A	Timer 1 Load Register (TLR1)
	X: 0xFFFF89	Timer 1 Compare Register (TCPR1)
	X: 0xFFFF88	Timer 1 Count Register (TCR1)
	X: 0xFFFF87	Timer 2 Control/Status Register (TCSR2)
	X: 0xFFFF86	Timer 2 Load Register (TLR2)
	X: 0xFFFF85	Timer 2 Compare Register (TCPR2)
	X: 0xFFFF84	Timer 2 Count Register (TCR2)
	X: 0xFFFF83	Timer Prescaler Load Register (TPLR)
	X: 0xFFFF82	Timer Prescaler Count Register (TPCR)
	X: 0xFFFF81–X: 0xFFFF80	Reserved
CGM	X: 0xFFFF7F	Reserved
	X: 0xFFFF7E	The ASRC internal generated reference clock divisor (ASCDR)
	X: 0xFFFF7D	PLL Control Register (PCTL)
	X: 0xFFFF7C	Shared Peripheral Clock Enable Register (SPENA)
	X: 0xFFFF7B–X: 0xFFFF75	Reserved

**Table 3-7. Detailed Device X-Memory Map (Continued)**

Peripherals	Address	Register Name <sup>1</sup>
S/PDIF	X: 0xFFFF74	S/PDIF Transmit Clock Control Register (STC)
	X: 0xFFFF73–X: 0xFFFF72	Reserved
	X: 0xFFFF71	Frequency Measurement (SRFM)
	X: 0xFFFF70	S/PDIF Transmit Professional C Channel Lo (STCSPL)
	X: 0xFFFF6F	S/PDIF Transmit Professional C Channel Hi (STCSPH)
	X: 0xFFFF6E	S/PDIF Transmit Consumer C Channel Lo (STCSCL)
	X: 0xFFFF6D	S/PDIF Transmit Consumer C Channel Hi (STCSCH)
	X: 0xFFFF6C	S/PDIF Transmit Right Channel (STR)
	X: 0xFFFF6B	S/PDIF Transmit Left Channel (STL)
	X: 0xFFFF6A	S/PDIF Receiver Q Channel (SRQ)
	X: 0xFFFF69	S/PDIF Receiver U Channel (SRU)
	X: 0xFFFF68	S/PDIF Receiver Channel Status 24–47 (SRCSL)
	X: 0xFFFF67	S/PDIF Receiver Channel Status 00–23 (SRCSH)
	X: 0xFFFF66	S/PDIF Receiver Right (SRR)
	X: 0xFFFF65	S/PDIF Receiver Left (SRL)
	X: 0xFFFF64	S/PDIF Interrupt STAT/CLR (SIS/SIC)
	X: 0xFFFF63	S/PDIF Interrupt Register (SIE)
	X: 0xFFFF62	S/PDIF Phase Configuration Register (SRPC)
	X: 0xFFFF61	S/PDIF CD Text Control Register (SRCD)
	X: 0xFFFF60	S/PDIF Configuration Register (SCR)
X: 0xFFFF5F–X: 0xFFFE70	Reserved	

**Table 3-7. Detailed Device X-Memory Map (Continued)**

Peripherals	Address	Register Name <sup>1</sup>
EMC	X: 0xFFFE6F–X: 0xFFFE6C	Reserved
	X: 0xFFFE6B	EMC Clock Ratio Register high part (CRRH)
	X: 0xFFFE6A	EMC Clock Ratio Register low part (CRRL)
	X: 0xFFFE69	EMC Configuration Register high part (BCRH)
	X: 0xFFFE68	EMC Configuration Register low part (BCRL)
	X: 0xFFFE67–X: 0xFFFE62	Reserved
	X: 0xFFFE61	EMC Transfer Error Address Register high part (TEARH)
	X: 0xFFFE60	EMC Transfer Error Address Register low part (TEARL)
	X: 0xFFFE5F	EMC Transfer Error Attributes Register high part (TEATRH)
	X: 0xFFFE5E	EMC Transfer Error Attributes Register low part (TEATRL)
	X: 0xFFFE5D	EMC Transfer Error Interrupt Register (TEIR)
	X: 0xFFFE5C	Reserved
	X: 0xFFFE5B	EMC Transfer Error Disable Register (TEDR)
	X: 0xFFFE5A	Reserved
	X: 0xFFFE59	EMC Transfer Error Status Register (TESR)
	X: 0xFFFE58	Reserved
	X: 0xFFFE57–X: 0xFFFE54	Reserved
	X: 0xFFFE53	SDRAM Refresh Timer (SRT)
	X: 0xFFFE52	Reserved
	X: 0xFFFE51	EMC UPM Refresh Timer (URT)
	X: 0xFFFE50	Reserved
	X: 0xFFFE4F–X: 0xFFFE4C	Reserved
	X: 0xFFFE4B	EMC SDRAM Mode Register high part (SDMRH)
	X: 0xFFFE4A	EMC SDRAM Mode Register low part (SDMRL)
	X: 0xFFFE49–X: 0xFFFE46	Reserved
	X: 0xFFFE45	EMC UPM Data Register high part (MDRH)
	X: 0xFFFE44	EMC UPM Data Register low part (MDRL)

**Table 3-7. Detailed Device X-Memory Map (Continued)**

Peripherals	Address	Register Name <sup>1</sup>
EMC	X: 0xFFFE43	EMC Memory Refresh Timer Prescaler Register (MRTPR)
	X: 0xFFFE42–X: 0xFFFF3E	Reserved
	X: 0xFFFE3D	EMC UPMC Mode Register high part (MCMRH)
	X: 0xFFFE3C	EMC UPMC Mode Register low part (MCMRL)
	X: 0xFFFE3B	EMC UPMB Mode Register high part (MBMRH)
	X: 0xFFFE3A	EMC UPMB Mode Register low part (MBMRL)
	X: 0xFFFE39	EMC UPMA Mode Register high part (MAMRH)
	X: 0xFFFE38	EMC UPMA Mode Register low part (MAMRL)
	X: 0xFFFE37–X: 0xFFFE36	Reserved
	X: 0xFFFE35	EMC UPM Address Register high part (MARH)
	X: 0xFFFE34	EMC UPM Address Register low part (MARL)
	X: 0xFFFE33–X: 0xFFFE20	Reserved
	X: 0xFFFE1F	EMC Options Register 7 high part (ORH7)
	X: 0xFFFE1E	EMC Options Register 7 low part (ORL7)
	X: 0xFFFE1D	EMC Base Register 7 high part (BRH7)
	X: 0xFFFE1C	EMC Base Register 7 low part (BRL7)
	X: 0xFFFE1B	EMC Options Register 6 high part (ORH6)
	X: 0xFFFE1A	EMC Options Register 6 low part (ORL6)
	X: 0xFFFE19	EMC Base Register 6 high part (BRH6)
	X: 0xFFFE18	EMC Base Register 6 low part (BRL6)
	X: 0xFFFE17	EMC Options Register 5 high part (ORH5)
	X: 0xFFFE16	EMC Options Register 5 low part (ORL5)
	X: 0xFFFE15	EMC Base Register 5 high part (BRH5)
	X: 0xFFFE14	EMC Base Register 5 low part (BRL5)
	X: 0xFFFE13	EMC Options Register 4 high part (ORH4)
	X: 0xFFFE12	EMC Options Register 4 low part (ORL4)
	X: 0xFFFE11	EMC Base Register 4 high part (BRH4)
X: 0xFFFE10	EMC Base Register 4 low part (BRL4)	

**Table 3-7. Detailed Device X-Memory Map (Continued)**

Peripherals	Address	Register Name <sup>1</sup>
EMC	X: 0xFFFE0F	EMC Options Register 3 high part (ORH3)
	X: 0xFFFE0E	EMC Options Register 3 low part (ORL3)
	X: 0xFFFE0D	EMC Base Register 3 high part (BRH3)
	X: 0xFFFE0C	EMC Base Register 3 low part (BRL3)
	X: 0xFFFE0B	EMC Options Register 2 high part (ORH2)
	X: 0xFFFE0A	EMC Options Register 2 low part (ORL2)
	X: 0xFFFE09	EMC Base Register 2 high part (BRH2)
	X: 0xFFFE08	EMC Base Register 2 low part (BRL2)
	X: 0xFFFE07	EMC Options Register 1 high part (ORH1)
	X: 0xFFFE06	EMC Options Register 1 low part (ORL1)
	X: 0xFFFE05	EMC Base Register 1 high part (BRH1)
	X: 0xFFFE04	EMC Base Register 1 low part (BRL1)
	X: 0xFFFE03	EMC Options Register 0 high part (ORH0)
	X: 0xFFFE02	EMC Options Register 0 low part (ORL0)
	X: 0xFFFE01	EMC Base Register 0 high part (BRH0)
	X: 0xFFFE00	EMC Base Register 0 low part (BRL0)
	X: 0xFFDFFF–X: 0xFFF000	Reserved

**Note:**
<sup>1</sup> Includes short name and long name.

**Table 3-8. Detailed Device Y-Memory Map**

Peripherals	Address	Register Name <sup>1</sup>
GPIO PORT G	Y: 0xFFFFF	Reserved
	Y: 0xFFFFFE	Port G Control Register 1 (PCRG1)
	Y: 0xFFFFFD	Port G GPIO Direction Register 1 (PRRG1)
	Y: 0xFFFFFC	Port G GPIO Data Register 1 (PDRG1)
	Y: 0xFFFFFB	Reserved
	Y: 0xFFFFFA	Port G Control Register (PCRG)
	Y: 0xFFFFF9	Port G GPIO Direction Register (PRRG)
	Y: 0xFFFFF8	Port G GPIO Data Register (PDRG)

**Table 3-8. Detailed Device Y-Memory Map (Continued)**

Peripherals	Address	Register Name <sup>1</sup>
GPIO PORT A	Y: 0xFFFFF7	Reserved
	Y: 0xFFFFF6	Port A Control Register 1 (PCRA_1)
	Y: 0xFFFFF5	Port A GPIO Direction Register 1 (PRRA_1)
	Y: 0xFFFFF4	Port A GPIO Data Register 1 (PDRA_1)
	Y: 0xFFFFF3	Reserved
	Y: 0xFFFFF2	Port A Control Register (PCRA)
	Y: 0xFFFFF1	Port A GPIO Direction Register (PRRA)
	Y: 0xFFFFF0	Port A GPIO Data Register (PDRA)
	Y: 0xFFFFEF–Y: 0xFFFFE7	Reserved
Chip Configuration	Y: 0xFFFFE6	External Memory Burst Control Register (EMBC)
	Y: 0xFFFFE5	EMC PLL Status & Control Register (PSC)
	Y: 0xFFFFE4	Chip Pin Mux Control (PMCR)
	Y: 0xFFFFE3	ESAI Pin Switch Control Register (EPSC)
	Y: 0xFFFFE2	Once Debug and Burst Control Register (ODBC)
	Y: 0xFFFFE1	Shared Peripheral Software Reset Control Register (SPSR)
	Y: 0xFFFFE0	Shared Bus Arbiters Control Register (OACR)
	Y: 0xFFFFDF–Y: 0xFFFFDC	Reserved

**Table 3-8. Detailed Device Y-Memory Map (Continued)**

Peripherals	Address	Register Name <sup>1</sup>
ICC Inter-Core Communication	Y: 0xFFFFDB	ICC Data Register 1 (ICDR1) For non-maskable interrupt <i>to</i> the other core.
	Y: 0xFFFFDA	ICC Control Register 1 (ICCR1) For non-maskable interrupt <i>to</i> the other core.
	Y: 0xFFFFD9	ICC Data Register 2 (ICDR2) For non-maskable Interrupt <i>from</i> the other core.
	Y: 0xFFFFD8	ICC Control Register 2 (ICCR2) For non-maskable Interrupt <i>from</i> the other core.
	Y: 0xFFFFD7	ICC Data Register 3 (ICDR3) For maskable Interrupt <i>to</i> the other core.
	Y: 0xFFFFD6	ICC Control Register 3 (ICCR3). For maskable Interrupt <i>to</i> the other core.
	Y: 0xFFFFD5	ICC Acknowledge Registers3 (ICAR3) The other core's acknowledge for the maskable interrupt <i>to</i> the other core.
	Y: 0xFFFFD4	ICC Data Register 4 (ICDR4) For maskable Interrupt <i>from</i> the other core.
	Y: 0xFFFFD3	ICC Control Register 4 (ICCR4) For maskable Interrupt <i>from</i> the other core.
	Y: 0xFFFFD2	ICC Acknowledge Register 4 (ICAR4) Acknowledge for the maskable interrupt <i>from</i> the other core.
	Y: 0xFFFFD1	ICC Poll Register 1 (ICPR1) Read poll data <i>from</i> the other core.
	Y: 0xFFFFD0	ICC Poll Register 2 (ICPR2) Write poll data <i>to</i> the other core.
	Y: 0xFFFFCF–Y: 0xFFFFCB	Reserved
	Y: 0xFFFFCA	ESAI Internal Clock Connect Control Register
	Y: 0xFFFFC9	Reserved
	Y: 0xFFFFC8	EMC/ICC Error Status Register
Y: 0xFFFFC7–Y: 0xFFFFC4	Reserved	
WDT,WDT_1	Y: 0xFFFFC3	Watchdog Service Register (WSR)
	Y: 0xFFFFC2	Watchdog Count Register (WCNTR)
	Y: 0xFFFFC1	Watchdog Modulus Register (WMR)
	Y: 0xFFFFC0	Watchdog Control Register (WCR)
	Y: 0xFFFFBF–Y: 0xFFFFA0	Reserved



**Table 3-8. Detailed Device Y-Memory Map (Continued)**

Peripherals	Address	Register Name <sup>1</sup>
ESAI_1,ESAI_3	Y: 0xFFFF9F	Port E/E1 Control Register (PCRE)
	Y: 0xFFFF9E	Port E/E1 Direction Register (PPRE)
	Y: 0xFFFF9D	Port E/E1 GPIO Data Register (PDRE)
	Y: 0xFFFF9C	ESAI_1/3 Receive Slot Mask Register B (RSMB_1)
	Y: 0xFFFF9B	ESAI_1/3 Receive Slot Mask Register A (RSMA_1)
	Y: 0xFFFF9A	ESAI_1/3 Transmit Slot Mask Register B (TSMB_1)
	Y: 0xFFFF99	ESAI_1/3 Transmit Slot Mask Register A (TSMA_1)
	Y: 0xFFFF98	ESAI_1/3 Receive Clock Control Register (RCCR_1)
	Y: 0xFFFF97	ESAI_1/3 Receive Control Register (RCR_1)
	Y: 0xFFFF96	ESAI_1/3 Transmit Clock Control Register (TCCR_1)
	Y: 0xFFFF95	ESAI_1/3 Transmit Control Register (TCR_1)
	Y: 0xFFFF94	ESAI_1/3 Common Control Register (SAICR_1)
	Y: 0xFFFF93	ESAI_1/3 Status Register (SAISR_1)
	Y: 0xFFFF92–Y: 0xFFFF8C	Reserved
	Y: 0xFFFF8B	ESAI_1/3 Receive Data Register 3 (RX3_1)
	Y: 0xFFFF8A	ESAI_1/3 Receive Data Register 2 (RX2_1)
	Y: 0xFFFF89	ESAI_1/3 Receive Data Register 1 (RX1_1)
	Y: 0xFFFF88	ESAI_1/3 Receive Data Register 0 (RX0_1)
	Y: 0xFFFF87	Reserved
	Y: 0xFFFF86	ESAI_1/3 Time Slot Register (TSR_1/3)
	Y: 0xFFFF85	ESAI_1/3 Transmit Data Register 5 (TX5_1)
	Y: 0xFFFF84	ESAI_1/3 Transmit Data Register 4 (TX4_1)
	Y: 0xFFFF83	ESAI_1/3 Transmit Data Register 3 (TX3_1)
	Y: 0xFFFF82	ESAI_1/3 Transmit Data Register 2 (TX2_1)
	Y: 0xFFFF81	ESAI_1/3 Transmit Data Register 1 (TX1_1)
	Y: 0xFFFF80	ESAI_1/3 Transmit Data Register 0 (TX0_1)

**Table 3-8. Detailed Device Y-Memory Map (Continued)**

Peripherals	Address	Register Name <sup>1</sup>
ASRC	Y: 0xFFFF7F–Y: 0xFFFC1E	Reserved
	Y: 0xFFFC1D	Data Output Register for Pair C (ASRDOC)
	Y: 0xFFFC1C	Data Input Register for Pair C (ASRDIC)
	Y: 0xFFFC1B	Data Output Register for Pair B (ASRDOB)
	Y: 0xFFFC1A	Data Input Register for Pair B (ASRDIB)
	Y: 0xFFFC19	Data Output Register for Pair A (ASRDOA)
	Y: 0xFFFC18	Data Input Register for Pair A (ASRDIA)
	Y: 0xFFFC17	Channel Counter Register (ASRCCR)
	Y: 0xFFFC16	Reserved
	Y: 0xFFFC15	Taskque FIFO Register 1 (ASRTFR1)
	Y: 0xFFFC14	Parameter Register 5(ASRPM5)
	Y: 0xFFFC13	Parameter Register 4(ASRPM4)
	Y: 0xFFFC12	Parameter Register 3(ASRPM3)
	Y: 0xFFFC11	Parameter Register 2(ASRPM2)
	Y: 0xFFFC10	Parameter Register 1(ASRPM1)
	Y: 0xFFFC0F	Debug Control Register -1 (ASRDCR-1).
	Y: 0xFFFC0E	Debug Control Register (ASRDCR).
	Y: 0xFFFC0D	Memory Access Data Register (ASRMAD)
	Y: 0xFFFC0C	Memory Access Address Register (ASRMAA)
	Y: 0xFFFC0B–Y: 0xFFFC09	Reserved
	Y: 0xFFFC08	ASRC Status Register (ASRSTR)
	Y: 0xFFFC07	ASRC Clock Divider Register (ASRCDR-2)
	Y: 0xFFFC06	ASRC Clock Divider Register (ASRCDR-1)
	Y: 0xFFFC05	ASRC Clock Source Register (ASRCSR)
	Y: 0xFFFC04	Filter Configuration Status Register (ASRCFG)
	Y: 0xFFFC03	Channel Number Configuration Register (ASRCNCR)
	Y: 0xFFFC02	Interrupt Enable Mask Register (ASRIEM)
	Y: 0xFFFC01	Interrupt Enable Register (ASRIER)
	Y: 0xFFFC00	ASRC Control Register (ASRCTR)
	Y: 0xFFFFBFF–Y: 0xFFFF000	Reserved

**Note:**

<sup>1</sup> Include short name and long name.

## Chapter 4

# DSP56300 Platform

### 4.1 Overview

The DSP56724 and DSP56725 have two DSP56300 platforms, which are identical. Each DSP56300 platform includes a DSP56300 core, a direct memory access unit (DMA), a program interrupt controller (PIC), and a Core/DMA Arbiter.

[Figure 4-1](#) provides the block diagram for the DSP56300 Core in DSP56724 and DSP56725.

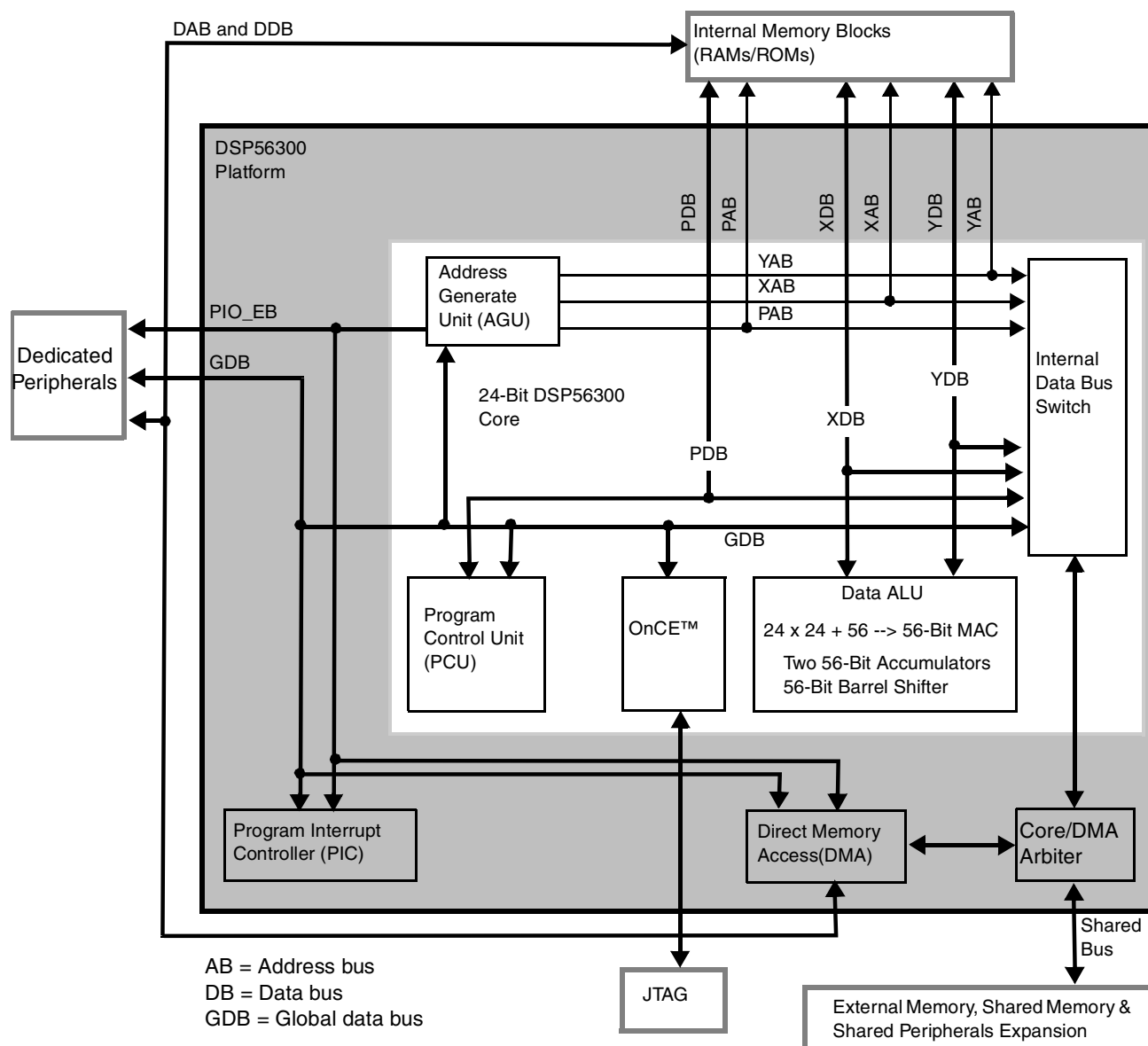


Figure 4-1. DSP56300 Core in DSP56724/DSP56725

The dedicated peripheral bus interface includes PM\_EB and GDB signals, and connects to dedicated peripherals. Peripherals can also be connected via the shared bus.

The DSP56300 core is a high-performance, single clock-cycle-per-instruction engine that provides up to twice the performance of Freescale's popular DSP56000 core family, while also retaining code compatibility with it.

## 4.2 DSP56300 Core Features

The DSP56300 core family provides a new level of performance in speed and power, provided by its rich instruction set and low power dissipation, thus enabling a new generation of wireless, telecommunications and multimedia products. Significant architectural enhancements to the DSP56300 core family include a

barrel shifter, 24-bit addressing, an instruction patch module and direct memory access (DMA). DSP56300 core features include the following:

- DSP56300 modular chassis
- Up to 250 million instructions per second (MIPS) with up to 250-MHz clock with 1.2-V internal logic supply
- Object code-compatible with DSP56000 core
- Data ALU with  $24 \times 24$  bit multiplier-accumulator and 56-bit barrel shifter plus support for 16-bit arithmetic
- Program control with support for position-independent code and instruction patches
- 8-channel DMA controller
- Support for internal address-tracing plus OnCE for hardware/software debugging
- STOP and WAIT low-power standby modes

### 4.3 DSP56300 Block Descriptions

The DSP56300 core provides five main functional blocks:

- Data arithmetic logic unit (Data ALU)
- Address generation unit (AGU)
- Program control unit (PCU)
- Internal Data Bus Switch
- OnCE module

DSP56300 core features are described fully in the *DSP56300 Family Manual*.

#### 4.3.1 Data ALU

The Data ALU performs all the arithmetic and logical operations on data operands in the DSP56300 core. Data ALU features include:

- Fully pipelined 24-bit  $\times$  24-bit parallel multiplier-accumulator (MAC)
- Bit field unit, comprising a 56-bit parallel barrel shifter (fast shift and normalization, bit stream generation and parsing)
- Conditional ALU instructions
- 24-bit or 16-bit arithmetic support under software control
- Four 24-bit input general purpose registers: X1, X0, Y1 and Y0
- Six Data ALU registers (A2, A1, A0, B2, B1 and B0), that are concatenated into two general purpose 56-bit accumulators (A and B), plus accumulator shifters
- Two data bus shifter/limiter circuits

### 4.3.1.1 Data ALU Registers

The Data ALU registers can be read or written over the X memory data bus (XDB) and the Y memory data bus (YDB), as 24- or 48-bit operands (or as 16- or 32-bit operands in 16-bit arithmetic mode). The source operands for the Data ALU, which can be 24, 48, or 56 bits (16, 32, or 40 bits in 16-bit arithmetic mode), always originate from Data ALU registers. The results of all Data ALU operations are stored in an accumulator.

All the Data ALU operations are performed in two clock cycles (in pipeline fashion) so that a new instruction can be initiated on every clock, yielding an effective execution rate of one instruction per clock cycle. The destination of every arithmetic operation can be used as a source operand for the immediately following arithmetic operation without a time penalty (without a pipeline stall).

### 4.3.1.2 Multiplier-Accumulator (MAC)

The MAC unit comprises the main arithmetic processing unit of the DSP56300 core and performs all of the calculations on data operands. For arithmetic instructions, the MAC accepts as many as three input operands and outputs one 56-bit result with the following form: Extension:Most Significant Product:Least Significant Product (EXT:MSP:LSP).

The multiplier executes 24-bit  $\times$  24-bit, parallel, fractional multiplies, between two's-complement signed, unsigned, or mixed operands. The 48-bit product is right-justified and added to the 56-bit contents of either the A or B accumulator. A 56-bit result can be stored as a 24-bit operand. The LSP can either be truncated or rounded into the MSP. Rounding is performed if specified.

## 4.3.2 Address Generation Unit (AGU)

The Address Generation Unit performs effective address calculations using integer arithmetic necessary to address data operands in memory, and contains the registers used to generate the addresses. The AGU implements four types of arithmetic (linear, modulo, multiple wrap-around modulo, reverse-carry), and operates in parallel with other chip resources to minimize address-generation overhead.

The AGU is divided into two halves, each with its own Address ALU. Each Address ALU has four sets of register triplets. Each register triplet is composed of an address register, an offset register and a modifier register. The two Address ALUs are identical. Each Address ALU contains a full 24-bit adder (called an offset adder).

A second full adder (called a modulo adder) adds the summed result of the first full adder to a modulo value that is stored in its respective modifier register. A third full adder (called a reverse-carry adder) is also provided.

The offset adder and the reverse-carry adder are in parallel and share common inputs. The only difference between the offset and reverse-carry adders is that the carry propagates in opposite directions. Test logic determines which of the three summed results of the full adders is output.

Each Address ALU can update one address register (from its respective address register file) during one instruction cycle. The contents of the associated modifier register specifies the type of arithmetic to be used in the address register update calculation. The modifier value is decoded in the Address ALU.

### 4.3.3 Program Control Unit (PCU)

The Program control unit performs instruction prefetch, instruction decoding, hardware DO loop control and exception processing. The PCU implements a seven-stage pipeline and controls the different processing states of the DSP56300 core. The PCU consists of the following hardware blocks:

- Program decode controller (PDC)
- Program address generator (PAG)
- Program interrupt controller (PIC)

The Program Decode controller decodes the 24-bit instruction loaded into the instruction latch and generates all signals necessary for pipeline control. The Program Address Generator contains all the hardware needed for program address generation, system stack and loop control. The Program Interrupt Controller arbitrates among all interrupt requests (internal interrupts, as well as the five external requests:  $\overline{\text{IRQA}}$ ,  $\overline{\text{IRQB}}$ ,  $\overline{\text{IRQC}}$ ,  $\overline{\text{IRQD}}$  and  $\overline{\text{NMI}}$ ) and generates the appropriate interrupt vector address.

PCU features include the following:

- Position-independent code support
- Addressing modes optimized for DSP applications (including immediate offsets)
- On-chip instruction cache controller
- On-chip memory-expandable hardware stack
- Nested hardware DO loops
- Fast auto-return interrupts

The PCU implements its functions using the following registers:

- PC: Program Counter Register
- SR: Status Register
- LA: Loop Address Register
- LC: Loop Counter Register
- VBA: Vector Base Address Register
- SZ: Stack Size Register
- SP: Stack Pointer
- OMR: Operating Mode Register
- SC: Stack Counter Register

The PCU also includes a hardware system stack (SS).

### 4.3.4 Internal Buses

To provide data exchange between blocks, the following buses are implemented:

- Peripheral input/output expansion bus (PIO\_EB) to peripherals
- Global data bus (GDB) between registers in the DMA, AGU, OnCE, and PCU, as well as the memory-mapped registers in the peripherals
- DMA data bus (DDB), which carries DMA data between memories and/or peripherals

- DMA address bus (DAB), which carries DMA addresses to memories and peripherals
- Program Data Bus (PDB), which carries program data between the core and internal memory
- X memory Data Bus (XDB), which carries X data between the core and internal memory
- Y memory Data Bus (YDB), which carries Y data between the core and internal memory
- Program address bus (PAB), which carries program memory addresses throughout the core
- X memory address bus (XAB), which carries X memory addresses throughout the core
- Y memory address bus (YAB), which carries Y memory addresses throughout the core
- Shared Bus for external memory expansion or external shared memory-mapped peripherals or memory. The Shared Bus is an enhanced feature which replaces the Port A external memory interface from the DSP56300 family. All accesses via the Shared Bus behave as a zero wait state SSRAM access from the Port A external memory interface, potentially extended by a transfer acknowledge. For this reason, all Program memory accesses by the DSP core over the Shared Bus take one additional wait state to complete.

All internal buses on the DSP56300 family members are 24-bit buses.

### 4.3.5 OnCE Module

An On-chip Emulation (OnCE) port supports hardware and software development on the DSP56300 core processor. It allows non-intrusive interaction with the core and its peripherals, so that developers can examine registers, memory, or on-chip peripherals. This facilitates hardware and software development on the DSP56300 core processor. OnCE module functions are provided through the JTAG TAP pins.



# Chapter 5

## Core Configuration

### 5.1 Introduction

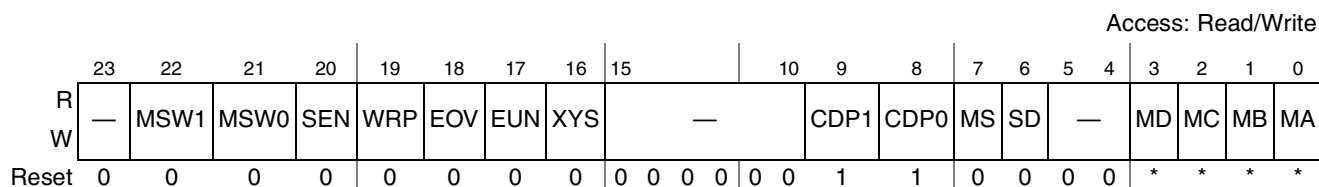
This chapter contains configuration details specific to the two DSP cores of the DSP56724/DSP56725 device, which includes:

- Operating modes register (OMR)
- Status Register (SR)
- Operating modes
- Interrupt sources and priorities
- DMA request sources
- Chip ID

For more information about specific registers or modules in the DSP56300 core, see the *DSP56300 Family Manual (DSP56300FM)*.

### 5.2 Operating Mode Register (OMR)

Both DSP cores have the operating mode register (OMR) as shown in [Figure 5-1](#). See the *DSP56300 Family Manual (DSP56300FM)* for a description of all the OMR bits.



**Note:** \* After reset, these bits reflect the corresponding value of the mode input (that is, MODD, MODC, MODB, or MODA, respectively).

**Figure 5-1. Operating Mode Register (OMR)**

OMR (Table 5-1) is a 24-bit register that is partitioned into the following bytes:

- OMR[23:16], System Stack Control/Status (SCS) Byte: Controls and monitors the stack extension in the data memory. The SCS byte is referenced implicitly by an instruction such as DO, JSR, or RTI, or referenced directly by the MOVEC instruction.
- OMR[15:8], Extended Chip Operating Mode (EOM) Byte: Determines the operating mode of the chip. This byte is affected only by hardware reset and by instructions directly referencing the OMR (that is, ANDI, ORI, and other instructions, such as MOVEC, that specify OMR as a destination).
- OMR[7:0], Chip Operating Mode (COM) Byte: Determines the operating mode of the chip. This byte is affected only by hardware reset and by instructions directly referencing the OMR (that is, ANDI, ORI, and other instructions, such as MOVEC, that specify OMR as a destination). During hardware reset, the chip operating mode bits (MD, MC, MB, and MA) are loaded from the external mode select pins MODD, MODC, MODB, and MODA, respectively.

**Table 5-1. Operation Mode Register Bit Definitions**

Bits	Name	Description
23	—	Reserved. Write to zero for future compatibility.
22—21	MSW1	Memory Switch Mode 1 See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i> , and <a href="#">Chapter 3, “Memory Map.”</a>
21	MSW0	Memory Switch Mode 0 See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i> , and <a href="#">Chapter 3, “Memory Map.”</a>
20	SEN	Stack Extension Enable See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i>
19	WRP	Extended Stack Wrap Flag See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i>
18	EOV	Extended Stack Overflow Flag See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i>
17	ENU	Extended Stack Underflow Flag See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i>
16	XYS	Stack Extension Space Select See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i>
15—10	—	Reserved. Write to zero for future compatibility.
9	CDP1	Core-DMA Priority 1 Specifies the priority between core accesses and DMA accesses to the on-chip shared memory, shared peripherals and external memory bus. For a detailed description of the priority, see the <i>DSP56300 FM 5.4.1.1 Operation Mode Register (OMR)</i> and <i>5.4.1.2 Status Register (SR)</i> .
8	CDP0	Core-DMA Priority 0 Specifies the priority between core accesses and DMA accesses to the on-chip shared memory, shared peripherals and external memory bus. For a detailed description of the priority, see the <i>DSP56300 FM 5.4.1.1 Operation Mode Register (OMR)</i> and <i>5.4.1.2 Status Register (SR)</i> .
7	MS	Master Memory Switch Mode See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i>
6	SD	Stop Delay See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i>

**Table 5-1. Operation Mode Register Bit Definitions (Continued)**

Bits	Name	Description
5–4	—	Reserved. Write to zero for future compatibility.
3	MD	Operating Mode D See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i>
2	MC	Operating Mode C See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i>
1	MB	Operating Mode B See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i>
0	MA	Operating Mode A See the <i>DSP56300 FM 5.4.1.1 Operation Mode Register(OMR)</i>

### 5.3 Status Register (SR)

The Status Register (SR) ([Figure 5-2](#) and [Table 5-2](#)) is a 24-bit register that consists of three 8-bit control registers. These three registers are defined within the SR primarily for compatibility with other Freescale DSPs.

- **Extended Mode Register (EMR) (SR[23:16]):**

Defines the current system state of the processor. The EMR bits are affected by hardware reset, exception processing, DO FOREVER instructions, ENDDO (end current DO loop) instructions, BRKcc instructions, RTI (return from interrupt) instructions, TRAP instructions, and instructions that specify the Status Register (SR) as their destination (for example, MOVEC). During hardware reset, all EMR bits are cleared.

- **Mode Register (MR) (SR[15:8]):**

Defines the current system state of the processor. The MR bits are affected by hardware reset, exception processing, DO instructions, ENDDO (end current DO loop) instructions, RTI (return from interrupt) instructions, TRAP instructions, and instructions that directly reference the Mode Register (MR) (for example, ANDI, ORI, or instructions, such as MOVEC, that specify the Status Register (SR) as the destination). During hardware reset, the interrupt mask bits are set and all other bits are cleared.

- **Condition Code Register (CCR) (SR[7:0]):**

Defines the results of previous arithmetic computations. The CCR register bits are affected by Data Arithmetic Logic Unit (Data ALU) operations, parallel move operations, instructions that directly reference the CCR register (ORI and ANDI), and by instructions that specify the Status Register (SR) as a destination (for example, MOVEC). Parallel move operations affect only the S and L bits of the CCR register. During hardware reset, all CCR register bits are cleared.

The Status Register is pushed onto the System Stack when the following conditions are true:

- Program looping is initialized
- A JSR is performed, including long interrupts

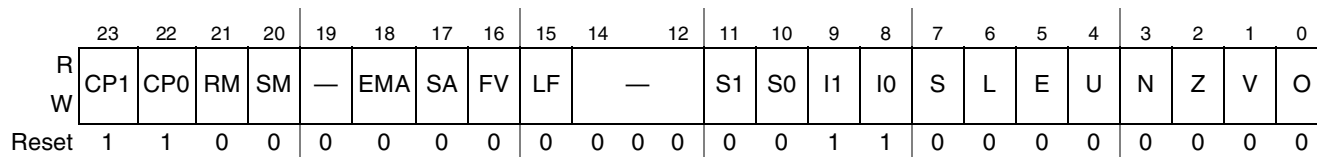


Figure 5-2. Status Register (SR)

Table 5-2. Status Register Bit Definitions

Bits	Name	Description																																
23–22	CP[1:0]	<p><b>Core Priority:</b> Under the control of CDP[1:0] bits in the Operating Mode Register (OMR), the Core Priority bits (CP1, CP0) specify the priority of core accesses to the internal shared memory, peripherals, as well as the external memory bus. The CP[1:0]bits are compared against the priority bits of the active DMA channel:</p> <ul style="list-style-type: none"> <li>• If the core priority is greater than the DMA priority, the DMA waits for a free time slot on the external shared bus.</li> <li>• If the core priority is less than the DMA priority, the core waits for a free time slot on the external shared bus.</li> <li>• If the core priority equals the DMA priority, the core and DMA take turns accessing in a round-robin pattern (for example, ... P, X, Y, DMA, P, X, Y, ...).</li> </ul> <p>The core priority bits are set during hardware reset.</p>																																
		<table border="1"> <thead> <tr> <th>Priority Mode</th> <th>Core Priority</th> <th>DMA Priority</th> <th>OMR (CDP[1:0])</th> <th>SR (CP[1:])</th> </tr> </thead> <tbody> <tr> <td rowspan="4">Dynamic</td> <td>0 (Lowest)</td> <td rowspan="4">Determined by DCRn (DPR[1:0]) for active DMA channels.</td> <td>00</td> <td>00</td> </tr> <tr> <td>1</td> <td>00</td> <td>01</td> </tr> <tr> <td>2</td> <td>00</td> <td>10</td> </tr> <tr> <td>3 (Highest)</td> <td>00</td> <td>11</td> </tr> <tr> <td rowspan="3">Static</td> <td colspan="2">Core &lt; DMA</td> <td>01</td> <td>xx</td> </tr> <tr> <td colspan="2">Core = DMA</td> <td>10</td> <td>xx</td> </tr> <tr> <td colspan="2">Core &gt; DMA</td> <td>11</td> <td>xx</td> </tr> </tbody> </table>	Priority Mode	Core Priority	DMA Priority	OMR (CDP[1:0])	SR (CP[1:])	Dynamic	0 (Lowest)	Determined by DCRn (DPR[1:0]) for active DMA channels.	00	00	1	00	01	2	00	10	3 (Highest)	00	11	Static	Core < DMA		01	xx	Core = DMA		10	xx	Core > DMA		11	xx
		Priority Mode	Core Priority	DMA Priority	OMR (CDP[1:0])	SR (CP[1:])																												
		Dynamic	0 (Lowest)	Determined by DCRn (DPR[1:0]) for active DMA channels.	00	00																												
			1		00	01																												
			2		00	10																												
			3 (Highest)		00	11																												
		Static	Core < DMA		01	xx																												
Core = DMA			10	xx																														
Core > DMA			11	xx																														
21	RM	Rounding Mode See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .																																
20	SM	Arithmetic Saturation Mode See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .																																
19	—	Reserved. Write zeroes for future compatibility.																																

**Table 5-2. Status Register Bit Definitions (Continued)**

Bits	Name	Description
18	EMA	Extended Modulo Addressing Control bit to enable modulo ranges of up to 24 bits when enabled (when bit is set). When 24 bit modulo addressing is enabled: A linear modifier requires $Mn = 0xFFFFFFFF$ ; A reverse-carry modifier requires $Mn = 0x0000000$ ; A modulo modifier requires $Mn = \text{modulus} - 1$ , where modulus can range from 2 to $2^{23}$ ; A multiple wrap-around modulo modifier requires bit 23 of $Mn$ to be set, bit 22 to be clear, and the remaining bits set to one less than the modulus (which must be a power of two from $2^1$ to $2^{22}$ ). When disabled, the existing 16-bit modulo range is supported for backwards compatibility, as defined in the <i>DSP56300 Family Manual</i> . If an RTI instruction is executed and EMA changes due to restoring the Status Register from the stack, the first instruction after RTI does not use the correct value of EMA. It is recommended that EMA be restored from the stack before executing an RTI instruction.
17	SA	Sixteen-Bit Arithmetic Mode See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .
16	FV	DO FOREVER Flag See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .
15	LF	DO Loop Flag See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .
14–12	—	Reserved Write to zero for future compatibility.
11–10	S[1:0]	Scaling Mode See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .
9–8	I[1:0]	Interrupt Mask See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .
7	S	Scaling See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .
6	L	Limit See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .
5	E	Extension See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .
4	U	Unnormalized See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .
3	N	Negative See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .
2	Z	Zero See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .
1	V	Overflow See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .
0	C	Carry See the <i>DSP56300 Family Manual 5.4.1.2 Status Register (SR)</i> .

## 5.4 DSP Cores Operating Modes

The operating modes are defined in [Table 5-3](#) and [Table 5-4](#). During reset, Core-0's OMR: MA, MB, MC, MD bits are latched from the MODA0, MODB0, MODC0, and MODD0 pins. During reset, Core-1's OMR:MA, MB, MC, MD bits are latched from MODA1, MODB1, MODC1, and MODD1 pins.

**Table 5-3. Core-0 Operating Modes in DSP56724**

Mode	DSP56724 External Pins				Reset Vector	Description
	MODD0	MODC0	MODB0	MODA0		
	OMR:MD	OMR:MC	OMR:MB	OMR:MA		
0	0	0	0	0	0xFFFFFE	Boot via SHI (SPI)
1	0	0	0	1		Boot via SHI (I2C Filter)
2	0	0	1	0		Jump to PROM (SPI)
3	0	0	1	1		Jump to PROM (I2C Filter)
4	0	1	0	0		Boot via Core-1
5	0	1	0	1		Boot via SHI Master (SPI-EEPROM)
6	0	1	1	0		Boot via SHI Master (I2C-EEPROM)
7	0	1	1	1		Boot via GPIO Master (SPI-EEPROM) PE6/PE7/PE8/PE9
8	1	0	0	0		Boot via External Memory word-wide. Not available for DSP56725.
9	1	0	0	1		Boot via External Memory byte-wide. Not available for DSP56725.
A	1	0	1	0		Reserved
B	1	0	1	1		Reserved
C	1	1	0	0		Reserved
D	1	1	0	1		Reserved
E	1	1	1	0		Reserved
F	1	1	1	1		Reserved

Table 5-4. Core-1 Operating Modes in DSP56724

Mode	DSP56724 External Pins				Reset Vector	Description
	MODD1	MODC1	MODB1	MODA1		
	OMR:MD	OMR:MC	OMR:MB	OMR:MA		
0	0	0	0	0	0xFFFFFE	Boot via SHI (SPI)
1	0	0	0	1		Boot via SHI (I2C Filter)
2	0	0	1	0		Jump to PROM (SPI)
3	0	0	1	1		Jump to PROM (I2C Filter)
4	0	1	0	0		Boot via Core-0
5	0	1	0	1		Boot via SHI Master (SPI-EEPROM)
6	0	1	1	0		Boot via SHI Master (I2C-EEPROM)
7	0	1	1	1		Boot via GPIO Master (SPI-EEPROM) PE6/PE7/PE8/PE9
8	1	0	0	0		Boot via External Memory word-wide. Not available in DSP56725 packages.
9	1	0	0	1		Boot via External Memory byte-wide. Not available in DSP56725 packages.
A	1	0	1	0		Reserved
B	1	0	1	1		Reserved
C	1	1	0	0		Reserved
D	1	1	0	1		Reserved
E	1	1	1	0		Reserved
F	1	1	1	1		Reserved

Table 5-5. DSP56724 Core-0/Core-1 Boot Modes

Mode	Name	Description
Mode 0	Boot via SHI (SPI)	In Mode 0, the internal PRAM is loaded from the Serial Host Interface (SHI). The SHI operates in the SPI slave mode, with 24-bit word width. The bootstrap code expects to read a single 24-bit word specifying the number of program words, another 24-bit word specifying the address to start loading the program words, and then a 24-bit word for each program word to be loaded. The program words will be stored in contiguous PRAM memory locations starting at the specified starting address. After reading the program words, program execution starts from the same address where loading started.
Mode 1	Boot via SHI (I <sup>2</sup> C Filter)	Mode 1 boot mode uses the same operation as Mode 0 (Boot via SHI (SPI)), except that the SHI interface operates in the I <sup>2</sup> C slave mode, with HCKFR set to 1 and the 100 ns filter enabled.
Mode 2	Jump to PROM (SPI)	The DSP starts fetching instructions from the starting address of the on-chip Program ROM. SHI operates in SPI slave mode.

**Table 5-5. DSP56724 Core-0/Core-1 Boot Modes (Continued)**

Mode	Name	Description
Mode 3	Jump to PROM (I <sup>2</sup> C Filter)	The DSP starts fetching instructions from the starting address of the on-chip Program ROM. SHI operates in I <sup>2</sup> C mode with the 100 ns filter enabled.
Mode 4	Boot from Other Core	When bit 23 of the ICPR1 register (y: 0xFFFFD1) is set, the DSP starts fetching instructions from the shared memory area. The DSP fetches instructions from the shared memory starting at the address indicated in bits 17–0 of the ICPR1 register. The bootstrap code expects to read a 24-bit word specifying the number of program words, another 24-bit word specifying the address to start loading the program words, and then a 24-bit word for each program word to be loaded. The program words will be stored in contiguous PRAM memory locations starting at the specified starting address. After reading the program words, program execution starts from the same address where loading started.
Mode 5	Boot via SHI Master (SPI-EEPROM)	In Mode 5, the internal memory (PRAM, XRAM, or YRAM) is loaded from an external serial EEPROM or FLASH in SPI mode. PH4 (HREQ) is used to determine the range of memory to be loaded. When PH4 is cleared, the 2-byte addressing format is used. When PH4 is set, the 3-byte addressing format is used. Mode 5 supports using ST M95xxx, M25Pxx and the Atmel AT25xxx family of FLASH/EEPROM memories.
Mode 6	Boot via SHI Master (I <sup>2</sup> C-EEPROM)	In Mode 6, the internal memory (PRAM, XRAM, or YRAM) is loaded from an external serial EPROM in I <sup>2</sup> C mode with the 100 ns filter enabled. Mode 6 supports using ST M24256 and the Atmel AT24C256 memories.
Mode 7	Boot via GPIO (SPI-EEPROM/FLASH)	In Mode 7, the internal memory (PRAM, XRAM, or YRAM) is loaded from an external serial EPROM in SPI mode via the GPIO pins. (Core-0 GPIO pins: PE6 - Chip Select, PE7 - Data in, PE8 - Data out and PE9 - clock) or (Core-1 GPIO pins: PC6_2 - Chip Select, PC7_2 - Data in, PC8_2 - Data out and PC9_2- clock) Mode 7 supports using ST M95256 and Atmel AT25256 memories.
Mode 8	Boot via EMC (Word-Wide EERPOM/FLASH)	In Mode 8, the internal memory (PRAM) is loaded from an external EEPROM or FLASH in word-wide mode. The bootstrap code reads the first word in external memory (address 0x800000). The bootstrap code expects to read a 24-bit word specifying the number of program words, another 24-bit word specifying the address to start loading the program words, and then a 24-bit word for each program word to be loaded. The program words will be stored in contiguous PRAM memory locations starting at the specified starting address. After reading the program words, program execution starts from the same address where loading started.
Mode 9	Boot via EMC (Byte-Wide EERPOM/FLASH)	Mode 9 boot mode uses the same operation as Mode 8, except that the data is accessed in byte-wide mode, and three bytes form a 24-bit word with big-endian format.
Mode A	—	Reserved
Mode B	—	Reserved
Mode C	—	Reserved
Mode D	—	Reserved
Mode E	—	Reserved
Mode F	—	Reserved



## 5.5 Interrupt Priority Registers

There are two PIC blocks in the DSP56724/DSP56725 device, with one PIC block for each DSP core. The PIC has also been enhanced to support additional DMA and peripheral interrupts. Two additional registers (IPR-C1, IPR-P1) have been added to the PIC to allow an additional 12 DMA interrupts and an additional 12 peripheral interrupts.

- IPR-C is dedicated for DSP56724/DSP56725: 4 external interrupts and the first 6 DMA channels interrupts.
- IPR-P is dedicated for DSP56724/DSP56725: 12 peripheral interrupt requests.
- IPR-C1 is dedicated for an additional 12 DMA channels interrupts; only 2 additional DMA channels are used in the DSP56724/DSP56725.
- IPR-P1 is dedicated for an additional 12 peripheral interrupt sources; only parts of the additional interrupts are used in the DSP56724/DSP56725.

The Interrupt Priority registers are shown in [Figure 5-3](#) through [Figure 5-10](#). The Interrupt Priority Level bits are defined in [Table 5-6](#) and [Table 5-7](#). The interrupt priorities are shown in [Table 5-8](#). The interrupt vectors are shown in [Table 5-9](#).

**Table 5-6. Peripherals and DMA Interrupt Priority Level Bits**

IPL bits		Interrupts Enabled	Interrupt Priority Level
(x)xxL1	(x)xxL0		
0	0	No	—
	1	Yes	0
1	0	Yes	1
	1	Yes	2

**Table 5-7. External Interrupts Priority Level Bits**

IPL bits			Interrupts Enabled	Interrupt Priority Level	Interrupt Trigger Mode
IxL2	IxL1	IxL0			
0	0	0	No	—	Level Triggered
		1	Yes	0	
	1	0	Yes	1	
		1	Yes	2	
1	0	0	No	—	Negative Edge Triggered
		1	Yes	0	
	1	0	Yes	1	
		1	Yes	2	

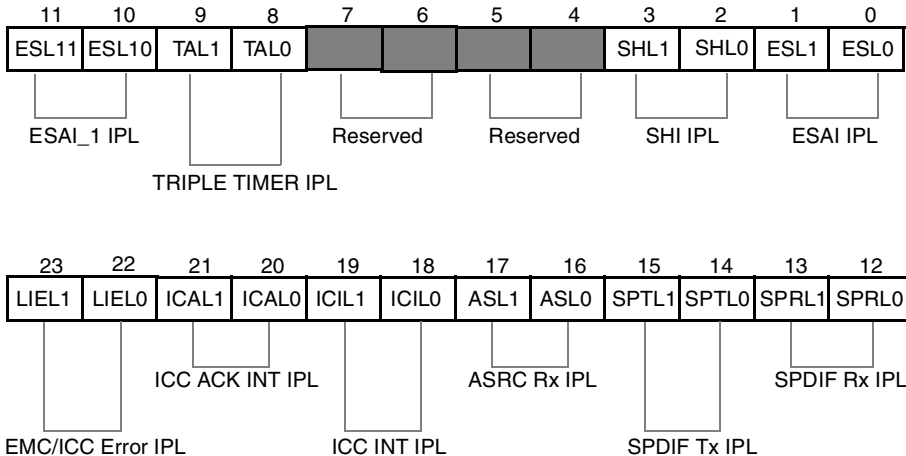


Figure 5-3. Core-0 Interrupt Priority Register P

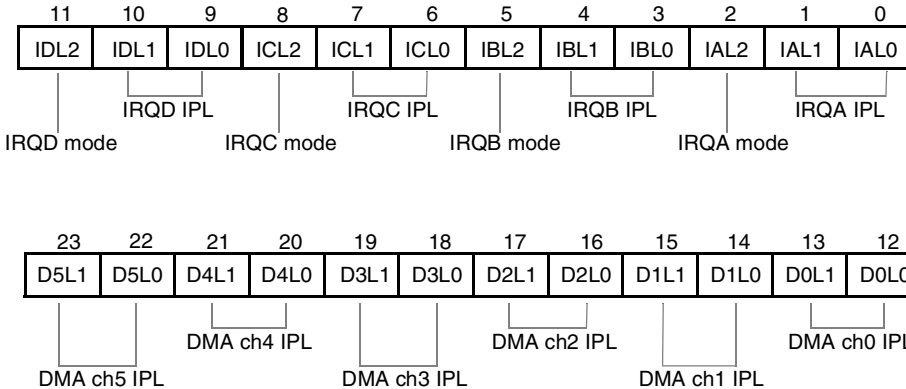


Figure 5-4. Core-0 Interrupt Priority Register C

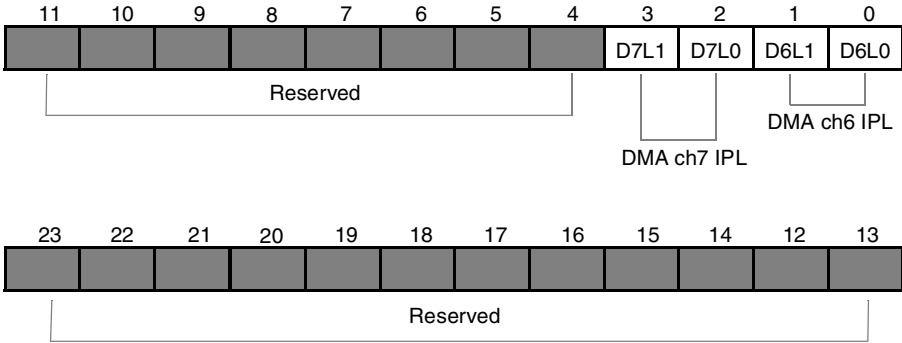


Figure 5-5. Core-0 Interrupt Priority Register C1

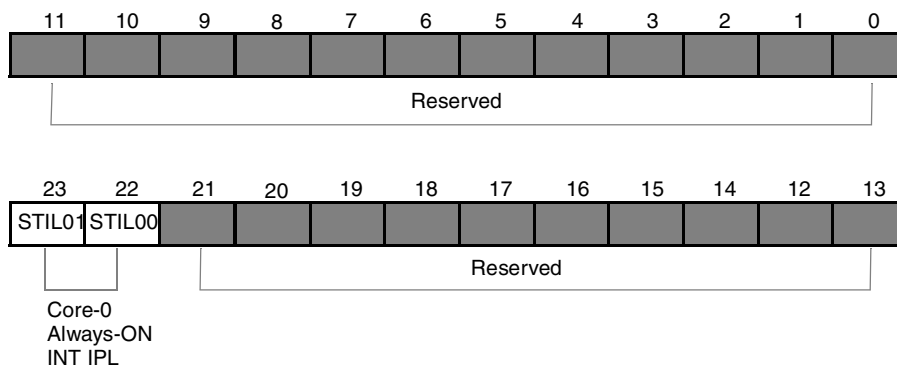


Figure 5-6. Core-0 Interrupt Priority Register P1

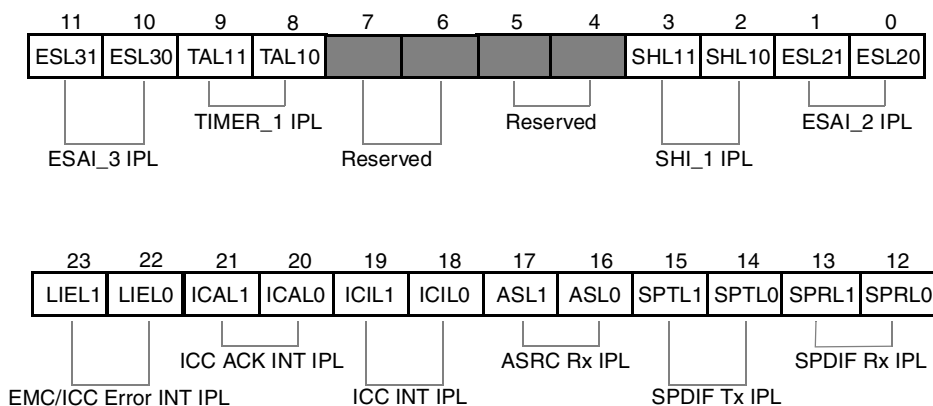


Figure 5-7. Core-1 Interrupt Priority Register P

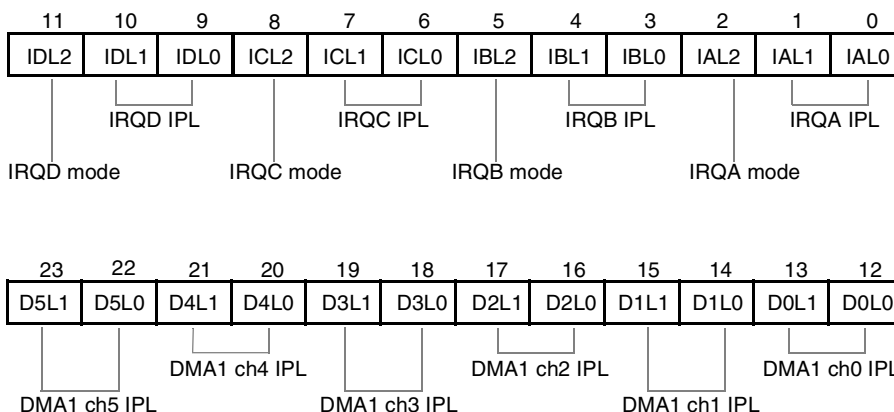
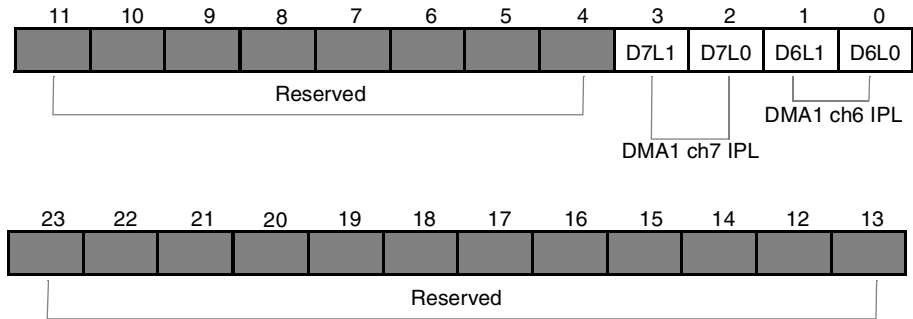
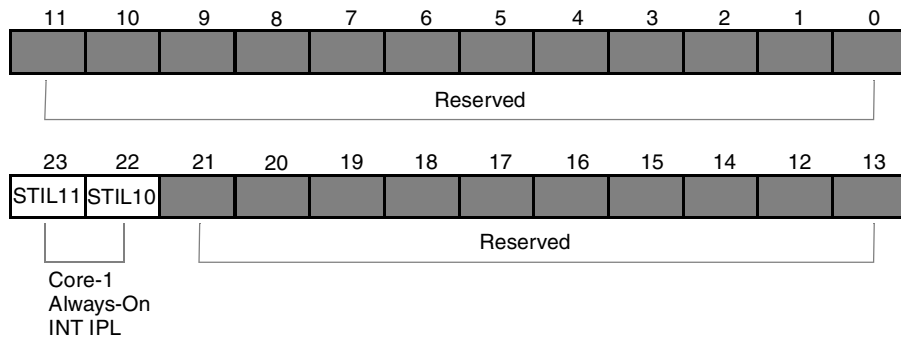


Figure 5-8. Core-1 Interrupt Priority Register C

## Core Configuration



**Figure 5-9. Core-1 Interrupt Priority Register C1**



**Figure 5-10. Core-1 Interrupt Priority Register P1**

**Table 5-8. Interrupt Sources Priorities within an IPL**

Priority Level	Interrupt Source	Group
Level 3 (non-maskable)		
Highest	RESET	—
	Stack Error	
	Illegal Instruction	
	Debug Request Interrupt	
	Trap	
	Non-Maskable Interrupt (NMI) from External	
	DMA Stall Interrupt	
Lowest	Inter-Core Non-Maskable Interrupt (from the other core)	
Level 0–2 (maskable)		

**Table 5-8. Interrupt Sources Priorities within an IPL (Continued)**

Priority Level	Interrupt Source	Group
Highest	IRQA	IRQ
	IRQB	
	IRQC	
	IRQD	
	DMA Channel 0	DMA
	DMA Channel 1	
	DMA Channel 2	
	DMA Channel 3	
	DMA Channel 4	
	DMA Channel 5	
	DMA Channel 6	
	DMA Channel 7	
Highest	ESAI/ESAI_2 Receive Data With Exception	ESAI/ESAI_2
	ESAI/ESAI_2 Receive Even Data	
	ESAI/ESAI_2 Receive Data	
	ESAI/ESAI_2 Receive Last Slot	
	ESAI/ESAI_2 Transmit Data with Exception Status	
	ESAI/ESAI_2 Transmit Last Slot	
	ESAI/ESAI_2 Transmit Even Data	
	ESAI/ESAI_2 Transmit Data	
	SHI/SHI_1 Bus Error	SHI/SHI_1
	SHI/SHI_1 Receive Overrun Error	
	SHI/SHI_1 Transmit Underrun Error	
	SHI/SHI_1 Receive FIFO Full	
	SHI/SHI_1 Transmit Data	
	SHI/SHI_1 Receive FIFO Not Empty	

**Table 5-8. Interrupt Sources Priorities within an IPL (Continued)**

Priority Level	Interrupt Source	Group
Highest	TEC/TEC_1 Timer0 Overflow	TEC/TEC_1
	TEC/TEC_1 Timer0 Compare	
	TEC/TEC_1 Timer1 Overflow	
	TEC/TEC_1 Timer1 Compare	
	TEC/TEC_1 Timer2 Overflow	
	TEC/TEC_1 Timer2 Compare	
	ESAI_1/3 Receive Data	ESAI_1/3
	ESAI_1/3 Receive Even Data	
	ESAI_1/3 Receive Data With Exception	
	ESAI_1/3 Receive Last Slot	
	ESAI_1/3 Transmit Data	
	ESAI_1/3 Transmit Even Data	
	ESAI_1/3 Transmit Data with Exception Status	
	ESAI_1/3 Transmit Last Slot	
	S/PDIF RcvChannelNew	S/PDIF
	S/PDIF RcvValidityBitNotSet	
	S/PDIF RcvIllegalSymbol	
	S/PDIF RcvParityError	
	S/PDIF RxUChannelFull	
	S/PDIF RxUChannelOver	
	S/PDIF RxQChannelFull	
	S/PDIF RxQChannelOver	
	S/PDIF RxUQSyncFound	
	S/PDIF RxUQFrameError	
	S/PDIF Rx Over/Under	
	S/PDIF Rx Resync	
	S/PDIF Lock Loss	
	S/PDIF Rcv FIFO Full	
	S/PDIF Lock Interrupt	
	S/PDIF Tx UnderOver	
	S/PDIF Tx Resync	
	S/PDIF Tx FIFO Empty	

**Table 5-8. Interrupt Sources Priorities within an IPL (Continued)**

Priority Level	Interrupt Source	Group
Highest	ASRC FP Wait State Interrupt	ASRC
	ASRC Overload Interrupt	
	ASRC Data Output C Interrupt	
	ASRC Data Output B Interrupt	
	ASRC Data Output A Interrupt	
	ASRC Data Input C Interrupt	
	ASRC Data Input B Interrupt	
	ASRC Data Input A Interrupt	
	Inter-Core Maskable Interrupt (from the other core)	
	Inter-Core Maskable Acknowledge interrupt (from the other core)	
	EMC/ICC Access Error Interrupt	
Lowest	Always Active Interrupt	

**Table 5-9. Reset and Interrupt Vector Summary**

Interrupt Starting Address	Priority Level Range	Description	Notes
VBA: 0x00	3	RESET	—
VBA: 0x02	3	Stack Error	
VBA: 0x04	3	Illegal Instruction	
VBA: 0x06	3	Debug Request Interrupt	
VBA: 0x08	3	Trap	
VBA: 0x0A	3	Non-Maskable Interrupt (NMI) (external)	
VBA: 0x0C	3	Reserved	
VBA: 0x0E	3	DMA Stall Interrupt	CIM interrupt for Core-0 and CIM_1 interrupt for Core-1
VBA: 0x10	0–2	IRQA	Shared by both cores.
VBA: 0x12	0–2	IRQB	
VBA: 0x14	0–2	IRQC	
VBA: 0x16	0–2	IRQD	

**Table 5-9. Reset and Interrupt Vector Summary (Continued)**

Interrupt Starting Address	Priority Level Range	Description	Notes
VBA: 0x18	0–2	DMA Channel 0	DMA Interrupts for Core-0 and DMA_1 interrupts for Core-1.
VBA: 0x1A	0–2	DMA Channel 1	
VBA: 0x1C	0–2	DMA Channel 2	
VBA: 0x1E	0–2	DMA Channel 3	
VBA: 0x20	0–2	DMA Channel 4	
VBA: 0x22	0–2	DMA Channel 5	
VBA: 0x24	0–2	DMA Channel 6	
VBA: 0x26	0–2	DMA Channel 7	
VBA: 0x28	3	Inter-Core Non-Maskable Interrupt (from the other core)	ICC interrupts for Core-0 and Core-1
VBA: 0x2A	0–2	Inter-Core Maskable Interrupt (from the other core)	
VBA: 0x2C	0–2	Inter-Core Maskable Acknowledge Interrupt (from the other core)	
VBA: 0x2E	0–2	Reserved	
VBA: 0x30	0–2	ESAI/ESAI_2 Receive Data	ESAI Output interrupts to Core-0; ESAI_2 Output interrupts to Core-1.
VBA: 0x32	0–2	ESAI/ESAI_2 Receive Even Data	
VBA: 0x34	0–2	ESAI/ESAI_2 Receive Data With Exception	
VBA: 0x36	0–2	ESAI/ESAI_2 Receive Last Slot	
VBA: 0x38	0–2	ESAI/ESAI_2 Transmit Data	
VBA: 0x3A	0–2	ESAI/ESAI_2 Transmit Even Data	
VBA: 0x3C	0–2	ESAI/ESAI_2 Transmit Data with Exception Status	
VBA: 0x3E	0–2	ESAI/ESAI_2 Transmit Last Slot	
VBA: 0x40	0–2	SHI/SHI_1 Transmit	SHI Output interrupts to Core-0; SHI_1 Output interrupts to Core-1.
VBA: 0x42	0–2	SHI/SHI_1 Transmit Underrun Error	
VBA: 0x44	0–2	SHI/SHI_1 Receive FIFO Not Empty	
VBA: 0x46	0–2	Reserved	
VBA: 0x48	0–2	SHI/SHI_1 Receive FIFO Full	
VBA: 0x4A	0–2	SHI/SHI_1 Receive Overrun Error	
VBA: 0x4C	0–2	SHI/SHI_1 Bus Error	
VBA: 0x4E	0–2	Reserved	—
VBA: 0x50	0–2		
VBA: 0x52	0–2		



**Table 5-9. Reset and Interrupt Vector Summary (Continued)**

Interrupt Starting Address	Priority Level Range	Description	Notes
VBA: 0x54	0–2	TEC/TEC_1 Timer0 Compare	TEC Output interrupts to Core-0; TEC_1 Output interrupts to Core-1.
VBA: 0x56	0–2	TEC/TEC_1 Timer0 Overflow	
VBA: 0x58	0–2	TEC/TEC_1 Timer1 Compare	
VBA: 0x5A	0–2	TEC/TEC_1 Timer1 Overflow	
VBA: 0x5C	0–2	TEC/TEC_1 Timer2 Compare	
VBA: 0x5E	0–2	TEC/TEC_1 Timer2 Overflow	
VBA: 0x60	0–2	Reserved	—
VBA: 0x62	0–2		
VBA: 0x66	0–2	Reserved	—
VBA: 0x68	0–2		
VBA: 0x6A	0–2		
VBA: 0x6C	0–2		
VBA: 0x6E	0–2		
VBA: 0x70	0–2		
VBA: 0x72	0–2	ESAI_1/3 Receive Even Data	
VBA: 0x74	0–2	ESAI_1/3 Receive Data With Exception	
VBA: 0x76	0–2	ESAI_1/3 Receive Last Slot	
VBA: 0x78	0–2	ESAI_1/3 Transmit Data	
VBA: 0x7A	0–2	ESAI_1/3 Transmit Even Data	
VBA: 0x7C	0–2	ESAI_1/3 Transmit Data with Exception Status	
VBA: 0x7E	0–2	ESAI_1/3 Transmit Last Slot	

**Table 5-9. Reset and Interrupt Vector Summary (Continued)**

Interrupt Starting Address	Priority Level Range	Description	Notes
VBA: 0x80	0–2	S/PDIF RcvCChannelNew	S/PDIF
VBA: 0x82	0–2	S/PDIF RcvValidityBitNotSet	
VBA: 0x84	0–2	S/PDIF RcvIllegalSymbol	
VBA: 0x86	0–2	S/PDIF RcvParityError	
VBA: 0x88	0–2	S/PDIF RxUChannelFull	
VBA: 0x8A	0–2	S/PDIF RxUChannelOver	
VBA: 0x8C	0–2	S/PDIF RxQChannelFull	
VBA: 0x8E	0–2	S/PDIF RxQChannelOver	
VBA: 0x90	0–2	S/PDIF RxUQSyncFound	
VBA: 0x92	0–2	S/PDIF RxUQFrameError	
VBA: 0x94	0–2	S/PDIF Rx Over/Under	
VBA: 0x96	0–2	S/PDIF Rx Resync	
VBA: 0x98	0–2	S/PDIF Lock Loss	
VBA: 0x9A	0–2	S/PDIF Rcv FIFO Full	
VBA: 0x9C	0–2	S/PDIF Lock Interrupt	
VBA: 0x9E	0–2	Reserved	
VBA: 0xA0	0–2	S/PDIF Tx Over/Under	
VBA: 0xA2	0–2	S/PDIF Tx Resync	
VBA: 0xA4	0–2	Reserved	
VBA: 0xA6	0–2	Reserved	
VBA: 0xA8	0–2	Reserved	
VBA: 0xAA	0–2	S/PDIF Tx FIFO Empty	
VBA: 0xAC to VBA: 0xAE	0–2	Reserved	
VBA: 0xB0	0–2	ASRC Data Input A Interrupt	ASRC
VBA: 0xB2	0–2	ASRC Data Input B Interrupt	
VBA: 0xB4	0–2	ASRC Data Input C Interrupt	
VBA: 0xB6	0–2	ASRC Data output A Interrupt	
VBA: 0xB8	0–2	ASRC Data output B Interrupt	
VBA: 0xBA	0–2	ASRC Data output C Interrupt	
VBA: 0xBC	0–2	ASRC Overload Interrupt	
VBA: 0xBE	0–2	ASRC Internal FP Wait States	

**Table 5-9. Reset and Interrupt Vector Summary (Continued)**

Interrupt Starting Address	Priority Level Range	Description	Notes
VBA: 0xC0 to VBA: 0xFA	0–2	Reserved	—
VBA: 0xFC	0–2	Always-On Interrupt (Always active, users can mask it or enable it by setting the corresponding Priority bits in IPRP1 bit 23 and 22.)	—
VBA: 0xFE	0–2	EMC/ICC Access Error Interrupt	—

## 5.6 DMA Request Sources

In previous DSP563xx products, 6 DMA channels were supported. In the DSP56724/DSP56725, the DMA blocks are updated, and up to 8 DMA channels can be supported.

Additional registers for the additional two DMA channels are included in the DMA modules. In the DSP56724/DSP56725, each DMA channel receives its own 32 request lines, allowing more flexibility in the DMA request sources for the different channels, and potentially support for a greater number of DMA request sources.

Each DMA channel’s Request Source bits (DRS4-DRS0 bits in the DMA Control/Status registers) encode the source of DMA requests used to trigger the DMA channels’ transfers. The DMA request sources may be internal peripherals, or external devices requesting service through the IRQA, IRQB, IRQC and IRQD pins.

The additional registers for the two additional DMA channels are the same as the registers for the other DMA channels. The previous section shows the addresses of these registers. The DMA status register is slightly different, as follows, with 8 channels of DMA:

- DSTR[6]: DTD6, DMA channel 6 (the seventh channel) transfer has finished.
- DSTR[7]: DTD7, DMA channel 7 (the eighth channel) transfer has finished.
- DSTR[10:9]: DCH[2:0]; when DCH[2:0] = 6, it indicates that the active channel is DMA channel 6; when DCH[2:0] = 7, it indicates that the active channel is DMA channel 7.

Table 5-10 shows the DMA request sources for all of the 8 DMA channels. All of the 34 DMA request lines are covered by two request line subsets: one line subset is for DMA channels 0–5, while the other line subset is for DMA channels 6–7. External request lines are only supported by DMA channel 0–5.

**Table 5-10. DMA Request Sources**

		Source Select Bits BSR[4:0] of DMA Channel 0–5	Source Select Bits BSR[4:0] of DMA Channel 6–7
1	External IRQA	00000	No support for these requests.
2	External IRQB	00001	
3	External IRQC	00010	
4	External IRQD	00011	
5	Transfer Done from DMA Channel 0	00100	00100

**Table 5-10. DMA Request Sources (Continued)**

		Source Select Bits BSR[4:0] of DMA Channel 0–5	Source Select Bits BSR[4:0] of DMA Channel 6–7
6	Transfer Done from DMA Channel 1	00101	00101
7	Transfer Done from DMA Channel 2	00110	00110
8	Transfer Done from DMA Channel 3	00111	00111
9	Transfer Done from DMA Channel 4	01000	01000
10	Transfer Done from DMA Channel 5	01001	01001
11	Transfer Done from DMA Channel 6	No support for these requests.	00000
12	Transfer Done from DMA Channel 7		00001
13	Reserved	01010	01010
14	ESAI/ESAI_2 receive data(RDF=1)	01011	01011
15	ESAI/ESAI_2 transmit data(TDE=1)	01100	01100
16	SHI HTX Empty	01101	01101
17	SHI FIFO Not Empty	01110	01110
18	SHI FIFO Full	01111	01111
19	TIMER0	10010	10010
20	TIMER1	10011	10011
21	TIMER2	10100	10100
22	ESAI_1/ESAI_3 Receive Data (RDF=1)	10101	10101
23	ESAI_1/ESAI_3 Transmit Data (TDE=1)	10110	10110
24	S/PDIF Xmt (SPDIFTxEmpty = 1)	10111	10111
25	SPDIF Rcv (PDIR1 full = 1)	11000	11000
26	Reserved	11001	11001
27	ASRC Rx 0	11010	11010
28	ASRC Rx 1	11011	11011
29	ASRC Rx 2	11100	11100
30	ASRC Tx 0	11101	11101
31	ASRC Tx 1	11110	11110
32	ASRC Tx 2	11111	11111

## 5.7 Chip ID Register

For more information about the CHIP ID Register (CHIDR), see [Chapter 6, “Core Integration Module \(CIM, CIM\\_1\).”](#)

# Chapter 6

## Core Integration Module (CIM, CIM\_1)

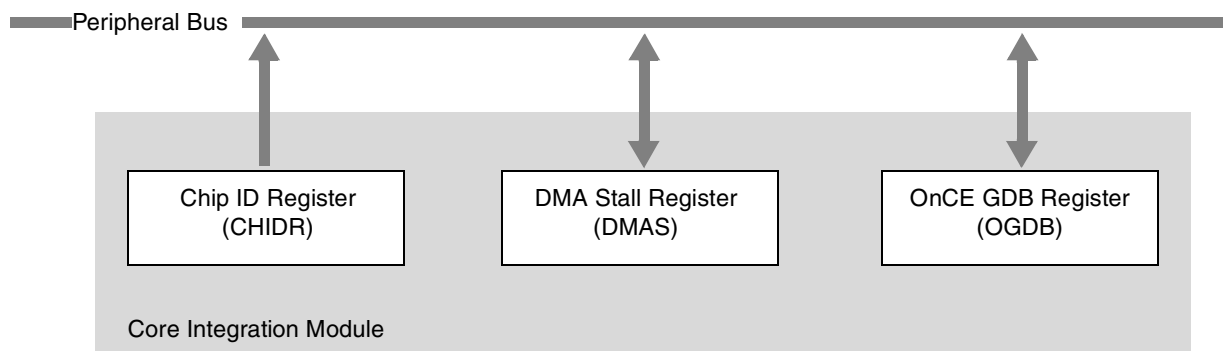
### 6.1 Overview

There are two Core Integration Module (CIM) modules in the DSP56724/DSP56725 devices: CIM and CIM\_1. CIM is used by DSP Core-0, while CIM\_1 is used by DSP Core-1. Both CIM blocks are identical, so only one CIM block is described in detail here.

The CIM block contains three registers: Chip ID number register, OnCE global data bus (GDB) register, and DMA Stall register. In more detail:

- The Chip ID Register contains the chip ID number.
- The CIM includes a DMA Monitor that optionally supports a non-maskable interrupt after the DMA has been stalled due to internal memory contention, for more than  $N$  cycles (where  $N$  can be from 2 to  $2^{24}$  cycles).
- The OnCE GDB register is a 24 bit register that can be read through the JTAG port, and is used for passing data between the chip and an external command controller.

Figure 6-1 shows block diagram for the core integration module.



**Figure 6-1. Core Integration Module Block Diagram**

## 6.1.1 Memory Map

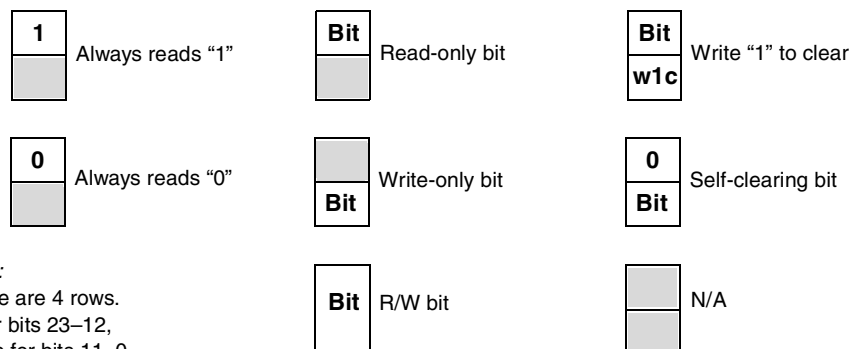
Table 6-1 shows the CIM memory map.

Table 6-1. CIM Memory Map

Offset or Address	Register	Access	Reset Value	
X: 0xFFFFF5	Chip ID Register (CHIDR)	R	DSP56724 device: 0x000724 for Core-0 0x010724 for Core-1	DSP56725 device: 0x000725 for Core-0 0x010725 for Core-1
X: 0xFFFFF8	DMA Stall Register (DMAS)	R/W	0x000000	
X: 0xFFFFFC	OnCE Global Data Bus Register (OGDB)	R/W	0x000000	

## 6.1.2 Register Summary

Figure 6-2 shows the legend for Table 6-2.



How to read Table 6-2:

For each register, there are 4 rows. The first 2 rows are for bits 23–12, and the last 2 rows are for bits 11–0.

Figure 6-2. Legend

Table 6-2 summarizes the CIM register.

Table 6-2. CIM Register Summary

Register		23	22	21	20	19	18	17	16	15	14	13	12
		11	10	9	8	7	6	5	4	3	2	1	0
DSP56724 Core-0 Chip ID Register (CHIDR) X: 0xFFFFF5	R	0	0	0	0	0	0	0	0	0	0	0	0
	W												
	R	0	1	1	1	0	0	1	0	0	0	0	0
	W												
DSP56724 Core-1 Chip ID Register (CHIDR) X: 0xFFFFF5	R	0	0	0	0	0	0	0	1	0	0	0	0
	W												
	R	0	1	1	1	0	0	1	0	0	0	0	0
	W												

**Table 6-2. CIM Register Summary (Continued)**

Register		23	22	21	20	19	18	17	16	15	14	13	12
		11	10	9	8	7	6	5	4	3	2	1	0
DSP56725 Core-0 Chip ID Register (CHIDR) X: 0xFFFFF5	R	0	0	0	0	0	0	0	0	0	0	0	0
	W												
	R	0	1	1	1	0	0	1	0	0	0	0	1
	W												
DSP56725 Core-1 Chip ID Register (CHIDR) X: 0xFFFFF5	R	0	0	0	0	0	0	0	1	0	0	0	0
	W												
	R	0	1	1	1	0	0	1	0	0	0	0	1
	W												
DMA Stall Register (DMAS) Y: 0xFFFFF8	R												
	W	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12
	R												
	W	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
OnCE GDB Register (OGDB) Y: 0xFFFFFC	R												
	W	O23	O22	O21	O20	O19	O18	O17	O16	O15	O14	O13	O12
	R												
	W	O11	O10	O9	O8	O7	O6	O5	O4	O3	O2	O1	O0

## 6.2 Register Descriptions

### 6.2.1 Chip ID Register (CHIDR)

The Chip ID Register is a 24-bit read-only register that contains the Chip ID number. There is a Chip ID Register for each DSP core in the DSP5672x.

- For the DSP56724, the Chip ID Register value is 0x000724 for Core-0 and 0x010724 for Core-1.
- For the DSP56725, the Chip ID Register value is 0x000725 for Core-0 and 0x010725 for Core-1.

Figure 6-3 summarizes the chip ID register for DSP56724 core-0.

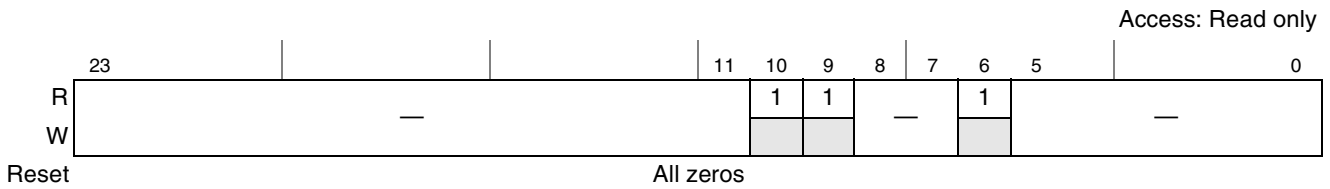


Figure 6-3. Chip ID Register for DSP56724 Core-0

Figure 6-4 summarizes the chip ID register for DSP56724 core-1.

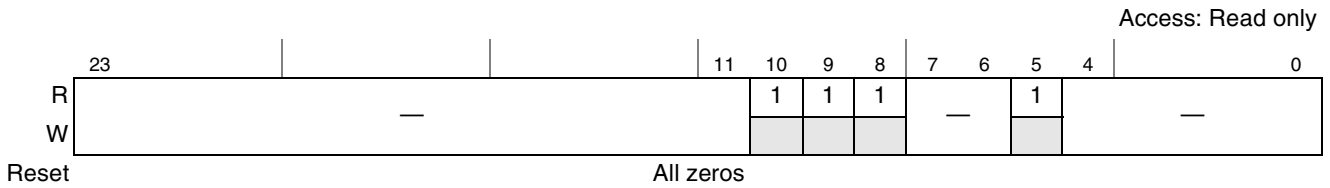


Figure 6-4. Chip ID Register for DSP56724 Core-1

Figure 6-5 summarizes the chip ID register for DSP56725 core-0.

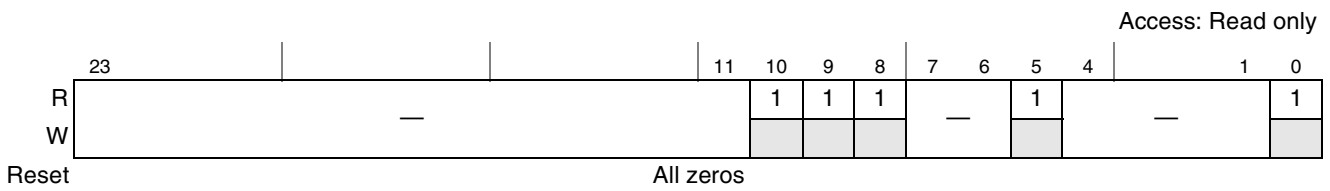


Figure 6-5. Chip ID Register for DSP56725 Core-0

Figure 6-6 summarizes the chip ID register for DSP56725 core-1.

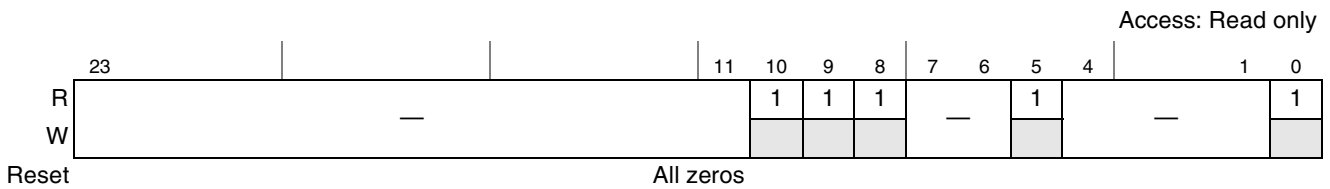


Figure 6-6. Chip ID Register for DSP56725 Core-1

## 6.2.2 DMA Stall Register (DMAS)

The DMA Stall Register is 24-bit read/write register that defines the threshold value of DMA counter of CIM. The DMA Stall Register and an associated interrupt (DMA Stall non-maskable interrupt) allow a limit to be placed on the number of cycles that the DMA is stalled, due to internal memory contention for a single DMA memory access.

- When the DMA Stall Register is set to zero, the DMA Stall Interrupt is disabled.
- When the DMA Stall Register is set to a non-zero value, a stall counter will keep track of the number of cycles the DMA is stalled due to internal memory contention for a single memory



access. If the stall counter is ever larger than the value stored in the DMA Stall Register, the DMA Stall non-maskable Interrupt will be asserted.

- The DMA Stall Interrupt remains asserted until the internal memory contention ends (usually due to the interrupt routine) or until the DMA Stall Register is written with zero. The stall counter clears when the internal memory contention ends or when the DMA Stall Register is written with zero.

Table 6-7 shows the DMA stall register.

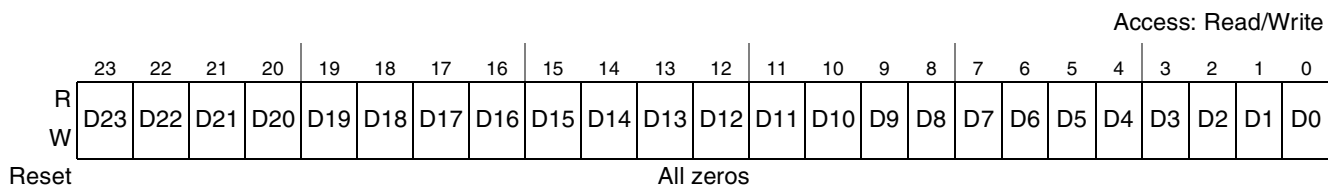


Figure 6-7. DMA Stall Register (DMAS)

### 6.2.3 OnCE Global Data Bus Register (OGDB)

The OnCE GDB Register is 24-bit read/write register that can be read through the JTAG port, and is used for passing data between the chip and an external command controller.

Figure 6-8 shows the OnCE GDB register.

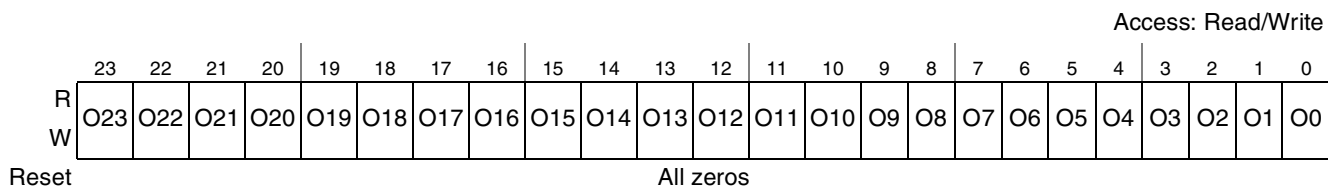


Figure 6-8. OnCE GDB Register (OGDB)



# Chapter 7

## Clock Generation Module (CGM)

### 7.1 Introduction

#### 7.1.1 Overview of Modes

The clock generation module (CGM) generates all clocks in the DSP56724/DSP56725. The system clock is either from an oscillator (bypass mode) or from a phase-locked loop (normal function mode). The system clock is properly controlled and gated. The CGM also has a low jitter PLL inside. The PLL has a wide input frequency range (2 MHz to 250 MHz), PFD comparison frequency range (2 MHz to 8 MHz) and output clock range (25 MHz to 500 MHz). The CGM also has a clock divider ( $2^i$ :  $i = 0$  to 7) after the PLL for saving power.

The CGM can be controlled from either of the DSP cores via the shared peripheral bus. In the CGM, the PLL control register (PCTL), which sits on the shared peripheral bus, can be read and written by both DSP cores to change the device's working frequency. Each DSP core can enter the STOP or WAIT mode for power saving. The shared peripherals, the shared memory block, and the external memory interface (if present), can enter power-saving mode only when both cores enter STOP mode.

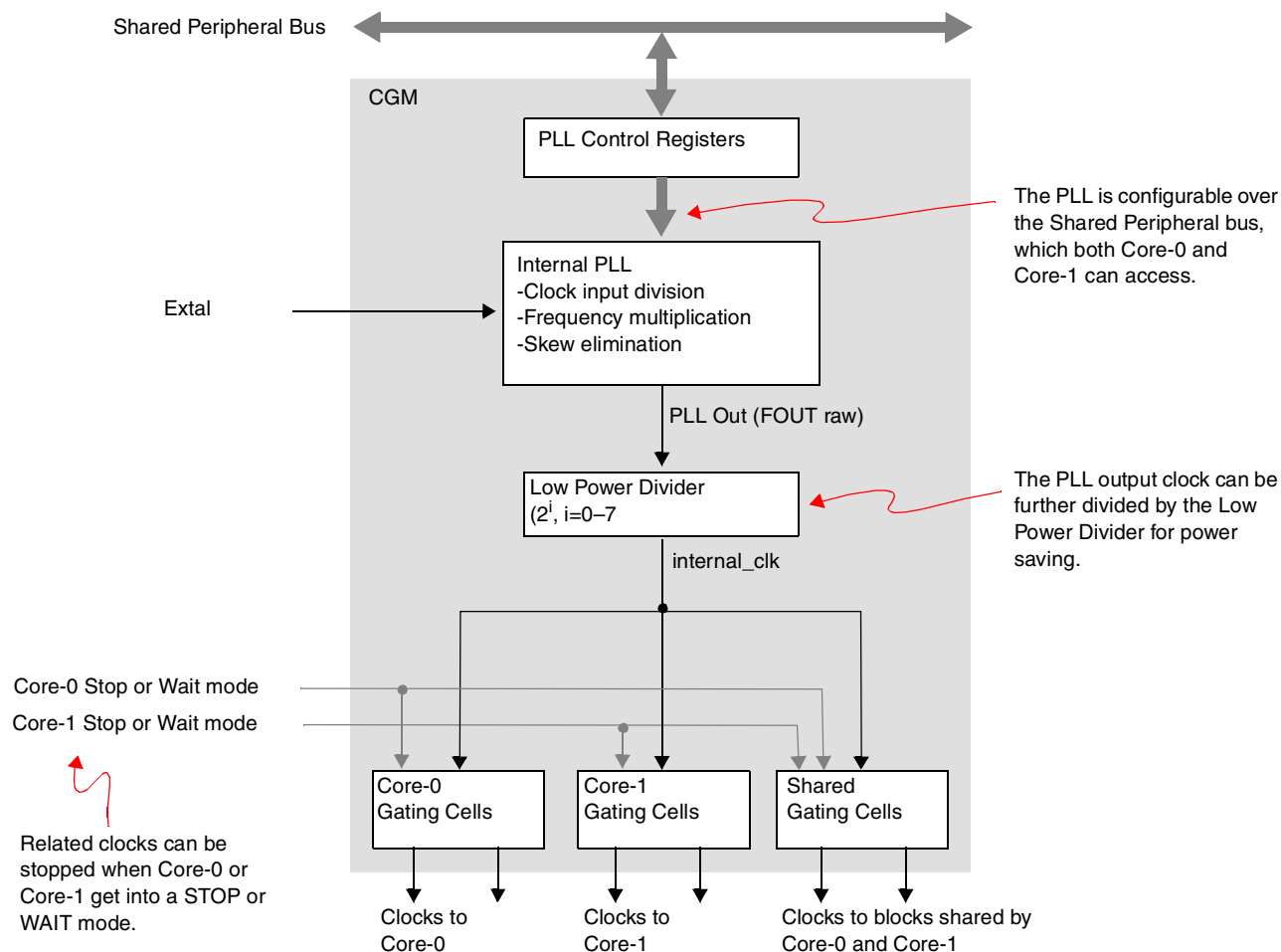
#### NOTES

A reset must be applied when the external oscillator frequency changes, to prevent a potential glitch on the PLL clock.

In the current PLL default configuration, if the oscillator clock frequency is *not between* 24 MHz and 24.61 MHz, the PLL must be bypassed (set `PINIT_NMI = 0` during reset). Otherwise, the PLL output clock is unstable because the PFD or VCO input is out of range. In case that the external oscillator clock frequency is *not between* 24 MHz and 24.61 MHz, be sure to bypass the PLL first, and then give the PLL the proper configuration by programming the PCTL register.

## Clock Generation Module (CGM)

Figure 7-1 shows the CGM function diagram.



**Figure 7-1. CGM Function Diagram**

### 7.1.2 Features

The CGM includes the following features:

- Allows changing the low-power divide factor ( $DF = 2^i, i = 0-7$ ) without losing lock.
- Provides glitch-free output clocks to DSP Cores and peripherals.
- Provides a wide range of system clocks.
- Performs power-saving by gating clocks for both cores and peripherals.

### 7.1.3 Modes of Operation

The CGM provides all necessary clocks to the DSP56724/DSP56725 devices. For stable operations, the system clock is defined in [Table 7-1](#).

**Table 7-1. System Clock Definition**

Scenario		PLL.PD	PLL.BP	Other Config	PLL.LD	CGM Clock <sup>1</sup>	System Clock <sup>2</sup>		
							Core Clock	Peripheral Clock	
During Reset	pinit_nmi = 1	0	1	By default	0 <sup>3</sup>	Fosc	Fosc	Fosc	
	pinit_nmi = 0								
After Reset	PLL Normal mode <sup>4</sup>	Before PLL.LD	0	0	—	0	Fosc	0	0
		After PLL.LD	0	0	—	1	Target	Target <sup>5</sup>	Target
	PLL Bypass mode <sup>6</sup>		0	1	—	0	Fosc	Fosc	Fosc
	System Wait mode		0	0/1 <sup>7</sup>	—	0/1	Fosc/Target	0 <sup>8</sup>	Fosc/Target
	System Stop mode		pstp <sup>9</sup>	0/1 <sup>10</sup>	—	0/1	Fosc/Target	0	0

<sup>1</sup> CGM Clock: the clocks in CGM module. It is always active for proper switching and wake-up.

<sup>2</sup> System Clock: the clock used by all DSP56724/DSP56725 modules.

<sup>3</sup> The PLL.LD will be low if the PLL doesn't detect a lock condition.

<sup>4</sup> The mode that PLL will get into after reset, normal mode or bypass mode, is decided by the PINIT\_NMI pin during reset.

<sup>5</sup> Target: the frequency (user-configured) in the PCTL register.

<sup>6</sup> It is the PLL's bypass mode. It is configured by either Core-0 or Core-1, over the Shared Peripheral bus.

<sup>7</sup> It is NOT a conflict if the PLL is in bypass mode while the system is in wait mode.

<sup>8</sup> In wait mode, for power saving, the Core clock is stopped even though the Peripheral clock is still active.

<sup>9</sup> Whether the PLL should be powered down or not in stop mode is decided by the PSTP bit in the PCTL register.

<sup>10</sup> It is NOT a conflict if the PLL is in bypass mode while the system is in stop mode.

During assertion of a hardware reset, the value of the PINIT input pin is written into the PCTL PLL Enable (PEN) bit. After a hardware reset is de-asserted, a write operation to PCTL will change the PEN bit accordingly, and the PLL will then ignore the PINIT pin.

During reset, the system clock is bypassed to extal.

After reset, if PINIT = 0 then the PLL will be in bypass mode and the system clock becomes extal immediately. Otherwise, the PLL is in normal function mode and the system will be driven by the PLL output clock after a PLL lock time (< 0.2 ms).

For the sake of power-saving, the CGM can switch to WAIT or STOP modes whenever the DSP Cores execute corresponding commands. The device can get out of WAIT and STOP modes if it gets the corresponding interrupts.

## 7.1.4 External Signal Description

Table 7-2 lists the signal properties.

**Table 7-2. Signal Properties**

Signal Name	Function	I/O	Reset	Pull-Up
PINIT	During assertion of a hardware reset, the value of the PINIT input pin is written into the PCTL PLL Enable (PEN) bit. After a hardware reset is de-asserted, the PLL ignores the PINIT pin. The default PCTL setting when PINIT is asserted is 0x2B60C2.	Input	Input	Pull-Up
EXTAL	An external clock is required to drive the DSP. The external clock is input via the EXTAL pin, passing the clock through the PLL and clock generator for optional frequency multiplication.	Input	Input	—
XTAL	An external crystal between the values of 10 MHz and 25 MHz can be driven from the XTAL pin. The external crystal should be connected to both the XTAL and EXTAL pins to provide the source clock frequency.	Output	Chip-Driven	—
PLL lock	On-chip system PLL lock indicator	Output	0	—

## 7.2 Functional Description

### 7.2.1 Clocks

The CGM has two clock sources, EXTAL and TCLK. The PLL is used in processing the EXTAL clock. The CGM outputs various clocks.

### 7.2.2 Reset

The CGM can only be reset by external asynchronous reset.

### 7.2.3 Interrupts

The CGM does not generate any interrupts.

### 7.2.4 Internal PLL Block

This section describes the PLL control components and its operation.

Figure 7-2 shows the internal PLL block diagram.

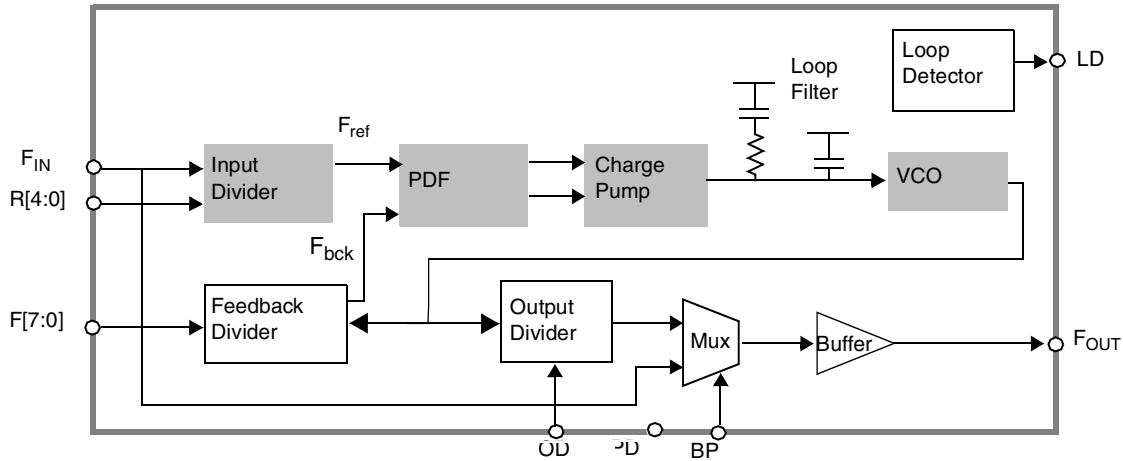


Figure 7-2. Internal PLL Block Diagram

The PLL has a programmable output frequency (from 25 to 500 MHz), configured using a 5-bit input divider, an 8-bit feedback divider and a 2-bit output divider. A 50% duty cycle for output clocks can be achieved by enabling the output divider. The PLL can also be used as a clock buffer in a bypass mode, which bypasses and powers down the PLL. A full power-down mode is also available.

7.2.4.1 PLL Operating Modes

There are three PLL operating modes: normal, bypass, and power-down.

- Normal Mode: The output clock frequency is programmable using the divider setting of  $R[4:0]$ ,  $F[7:0]$  and  $OD[1:0]$ . When the divider settings are changed, the PLL must enter the power-down mode ( $PD = HIGH$ ) for more than 50 ns (a PD timing requirement, see Table 7-3).
- Bypass Mode: The  $F_{IN}$  is buffered directly to  $F_{OUT}$ , bypassing the PLL, which is powered down. A TRDY time (pull-in + lock time) is required for the PLL to lock when switching from Bypass Mode to Normal Mode.
- Power-down Mode: The entire PLL cell is powered down internally, and  $F_{OUT}$  is set to 1 V. A TRDY time (pull-in + lock time) is required for the PLL to lock when switching from Power-down Mode to Normal Mode.

Table 7-3. PLL Output Configurations

PD	BP	OD	Description	Fout
0	0	0	Normal Mode	$F_{REF} * NF$
		1		$F_{REF} * NF/2$
		2		$F_{REF} * NF/4$
		3		$F_{REF} * NF/8$
0	1	x	Bypass Mode	$F_{IN}$
1	x	x	Power-Down Mode	1

## Clock Generation Module (CGM)

In *Normal Mode*, for the PLL to function properly, it is necessary to set suitable integer values for the dividers (NR for the input divider, NF for the feedback divider and NO for the output divider). The divider values are set using the digital binary inputs of R[4:0], F[7:0] and OD[1:0].

- Input Divider Value (**NR**):  

$$\mathbf{NR} = 16 \times R4 + 8 \times R3 + 4 \times R2 + 2 \times R1 + R0 + 1 = R[4:0] + 1$$
- Feedback Divider Value (**NF**):  

$$\mathbf{NF} = 128 \times F7 + 64 \times F6 + 32 \times F5 + 16 \times F4 + 8 \times F3 + 4 \times F2 + 2 \times F1 + F0 + 1 = F[7:0] + 1$$
- Output Divider Value (**NO**):  $\mathbf{NO} = 2^{\mathbf{OD}[1:0]}$

**Table 7-4. Output Divider Values**

OD[1:0]	0	1	2	3
NO	1	2	4	8

### 7.2.4.2 Programming the Output Clock Frequency

The following equations describe how the output frequency is calculated.

$$\mathbf{F_{REF}} = \mathbf{F_{IN}} / \mathbf{NR} \quad \text{Eqn. 7-1}$$

$$\mathbf{F_{VCO}} = \mathbf{F_{OUT}} \times \mathbf{NO} \quad \text{Eqn. 7-2}$$

$$\mathbf{F_{OUT}} = \mathbf{F_{IN}} \times \mathbf{NF} / (\mathbf{NR} \times \mathbf{NO}) \quad \text{Eqn. 7-3}$$

where FREF is the comparison frequency for the PFD.

For proper operation in normal mode, the following constraints *must* be satisfied.

When programming the Output Clock Frequency:

$$2 \text{ MHz} \leq \text{Fref} \leq 8 \text{ MHz}$$

$$200 \text{ MHz} \leq \text{Fvco} \leq 500 \text{ MHz}$$

See [Table 7-9](#).

### 7.2.5 Low Power Divider

The low power divider allows the system clock to be reduced (saving power) without causing the PLL to lose lock. The following equation describes how the system clock is calculated:

$$\mathbf{F_{sys}} = \mathbf{F_{out}} / 2^{\mathbf{DF}[2:0]}$$

The change in system clock frequency takes effect on the following rising edge of the system clock after the Divide Factor bits have updated.



## 7.3 Memory Map and Register Definition

### 7.3.1 Memory Map

Table 7-5 shows the CGM memory map.

Table 7-5. Block Memory Map

Address	Register	Access	Reset Value	Section/Page
X: 0xFFFF7C(SPENA)	Shared Peripheral Clock Enable Registers.	R/W	0x000001	<a href="#">7.3.3.1/7-8</a>
X: 0xFFFF7D (PCTL)	PLL Control Registers	R/W	0x2B60C2 <sup>1</sup>	<a href="#">7.3.3.2/7-8</a>
X: 0xFFFF7E(ASCDR)	ASRC Control Division Registers	R/W	0x000022	<a href="#">7.3.3.3/7-11</a>

<sup>1</sup> The default value of PCTL should be 0x2B\_60C2 if PINIT = 1 during reset.

### 7.3.2 Register Summary

Table 7-6 summarizes the registers.

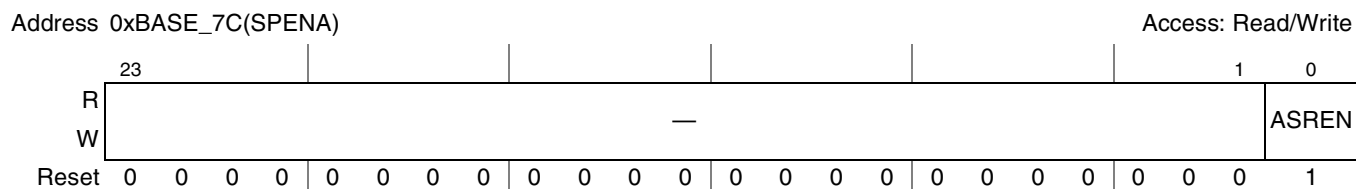
Table 7-6. Register Summary

Name		23	22	21	20	19	18	17	16	15	14	13	12
		11	10	9	8	7	6	5	4	3	2	1	0
X: 0xFFFF7C (SPENA)	R/W												
	Reset												
	R/W												ASREN
	Reset												1
X: 0xFFFF7D (PCTL)	R/W			PLKM	R4	R3	R2	R1	R0	OD1	OD0	PEN	PSTP
	Reset		0	1	0	1	0	1	1	0	1	PINIT	0
	R/W		DF2	DF1	DF0	F7	F6	F5	F4	F3	F2	F1	F0
	Reset	0	0	0	0	1	1	0	0	0	0	1	0
X: 0xFFFF7E (ASCDR)	R/W												
	Reset												
	R/W						ASDF6	ASDF5	ASDF4	ASDF3	ASDF2	ASDF1	ASDF0
	Reset						0	1	0	0	0	1	0

### 7.3.3 Register Descriptions

#### 7.3.3.1 Shared Peripheral Clock Enable Register (SPENA)

The Shared Peripheral Clock Enable register can enable or disable the clock of some shared peripherals, shown in [Figure 7-3](#) and [Table 7-7](#).



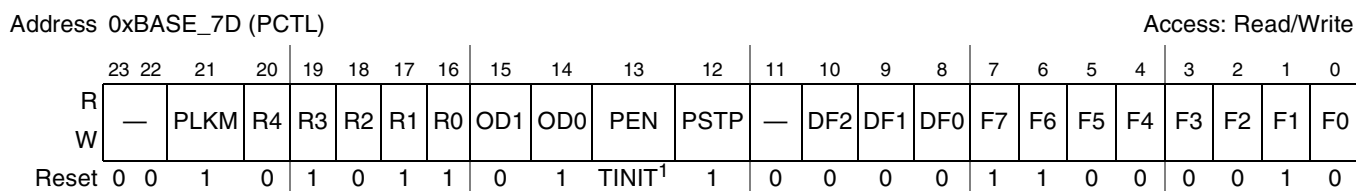
**Figure 7-3. Shared Peripheral Clock Enable Register (SPENA)**

**Table 7-7. Shared Peripheral Clock Enable Register Field Description**

Bits	Name	Description
23–1	—	Reserved
0	ASREN	ASRC Clock Enable 0 Turn off the ASRC clock 1 Turn on the ASRC clock

#### 7.3.3.2 PLL Control Register (PTCL)

There are PLL control and clock dividing control registers on the Shared Peripheral bus, shown in [Figure 7-4](#) and [Table 7-8](#). The PLL control register controls the PLL, and also controls the low-power divider.



**Figure 7-4. PLL Control Register (PCTL)**

<sup>1</sup> Reset value of PEN is 1 when TINIT is set,; reset value of PEN is 0 when TINIT is cleared.

**Table 7-8. PLL Control Registers Field Description**

Bits	Name	Description
23–22	—	Reserved. Must be written “0” to preserve future compatibility.
21	PLKM	PLL LOCK MUX The PLOCK MUX (PLKM) bit is a read/write bit that controls the operation of the PLOCK/GP0 pin. When PLKM is set, the PLOCK/GP0 pin operates as the PLL lock indicator (PLOCK). When the PLKM bit is cleared, the PLOCK/GP0 pin operates as the 0 bit of the GPIO port G.

**Table 7-8. PLL Control Registers Field Description (Continued)**

Bits	Name	Description
20–16	R4–R0	Input Divider Defines the input divider's value that is applied to the input frequency. R[4:0] can be any value from 0–31. $NR = R[4:0] + 1$ $F_{out}(pll) = (F_{in} * NF) / (NR * NO)$
15–14	OD1–OD0	Output Divider Defines the output divider value. The output divide factor divides the VCO output frequency by a factor of 1, 2, 4 or 8: OD1,OD0 = 2'b00: divide by 1 (NO = 1) OD1,OD0 = 2'b01: divide by 2 (NO = 2) OD1,OD0 = 2'b10: divide by 4 (NO = 4) OD1,OD0 = 2'b11: divide by 8 (NO = 8) $F_{out}(pll) = (F_{in} * NF) / (NR * NO)$
13	PEN	PLL Enable Enables PLL operation. When PEN is set, the PLL is enabled and the internal clocks are derived from the PLL VCO output. When PEN is cleared, the PLL is disabled and the internal clocks are derived directly from the EXTAL signal. When the PLL is disabled, the VCO stops to minimize power consumption. The PEN bit may be set or cleared by software at any time during the device operation.
12	PSTP	PLL Stop State Controls the PLL and on-chip crystal oscillator behavior during the Stop processing state. When PSTP is set, the PLL remains operating while the chip is in the Stop state. When PSTP is cleared and the device enters the Stop state, the PLL is disabled, to further reduce power consumption. PSTP, PEN = 2'b0x: PLL disabled. (Only in system Stop mode.) PSTP, PEN = 2'b10: PLL is enabled always but is in bypass mode. PSTP, PEN = 2'b11: PLL is enabled always and is in non-bypass mode.
11	—	Reserved. Must be written "0" to preserve future compatibility.
10–8	DF2–DF0	Division Factor Define the division factor (DF) of the low-power divider. These bits specify the DF as a power of two in the range of $2^0$ to $2^7$ . Changing the value of the DF[2:0] bits does not cause a loss of lock condition. Whenever possible, changes of the operating frequency of the device (for example, to enter a low-power mode) should be made by changing the value of the DF[2:0] bits rather than by changing the F[7:0] bits. The operating frequency: $F_{sys} = F_{out} / (2^{DF[2:0]})$
7–0	F7–F0	Multiplication Factor Defines the multiplication factor (MF) that is applied to the PLL input frequency. The MF can be any integer from 0 to 255. $NF = F[7:0] + 1$ . $F_{out}(pll) = (F_{in} * NF) / (NR * NO)$

Table 7-9 shows examples for PLL programming.

**Table 7-9. PLL Programming Examples**

Extal (MHz)	NR (= R + 1)	Fref (= Fosc/NR) (2~8 MHz)	NF (= F + 1)	Fvco (= Fref*NF) (200~400 MHz)	NO (= 2^OD)	PLL Output (= Fvco/NO) (MHz)	PLL Setting (0x)
24.576	12	2.048	122	249.856	1	249.856	2B2079

Table 7-9. PLL Programming Examples (Continued)

Extal (MHz)	NR (= R + 1)	Fref (= Fosc/NR) (2~8 MHz)	NF (= F + 1)	Fvco (= Fref*NF) (200~400 MHz)	NO (=2^OD)	PLL Output (= Fvco/NO) (MHz)	PLL Setting (0x)
24.576	12	2.048	98	200.704	1	200.704	2B4061
24.576	12	2.048	195	399.360	2	199.680	2B60C2
24.576	12	2.048	194	397.312	2	198.656	2B60C1
24.576	12	2.048	191	391.168	2	195.584	2B60BE
24.576	12	2.048	181	370.688	2	185.344	2B60B4
24.576	12	2.048	171	350.208	2	175.104	2B60AA
24.576	12	2.048	157	321.536	2	160.768	2B609C
24.576	12	2.048	180	368.640	2	184.320	2B60B3
24.576	12	2.048	147	301.056	2	150.528	2B6092
24.576	12	2.048	98	200.704	8	25.088	2BE061
24.576	4	6.144	65	399.360	2	199.680	236040
24.576	4	6.144	33	202.752	1	202.752	234020
12.288	6	2.048	98	200.704	1	200.704	254061
12.288	6	2.048	195	399.360	2	199.680	2560C2
12.288	6	2.048	194	397.312	2	198.656	2560C1
12.288	6	2.048	185	378.880	2	189.440	2560B8
12.288	6	2.048	176	360.448	2	180.224	2560AF
12.288	2	6.144	64	393.216	2	196.608	21603F
12.288	2	6.144	33	202.752	1	202.752	214020
11.290	5	2.258	89	200.955	1	200.955	244059
11.290	5	2.258	177	399.652	2	199.826	2460B0
11.290	5	2.258	176	397.394	2	198.697	2460AF
11.290	2	5.645	70	395.136	2	197.568	216045
11.290	2	5.645	69	389.491	2	194.746	216044
11.290	2	5.645	36	203.213	1	203.213	214023
11.290	2	5.645	37	208.858	1	208.858	214024

**NOTE:**

- The line highlighted with **Green** is the default PLL configuration.
- The lines highlighted with **Yellow** are Fout >200 MHz. The duty cycle may not be 50%.
- The lines highlighted with **Blue** are the recommended PLL settings for each Extal frequency.

### 7.3.3.3 ASRC Clock Dividing Control Register (ASCDR)

The clock dividing control register instructs the CGM to generate the corresponding divided clocks to the ASRC module, shown in Figure 7-5 and Table 7-10.

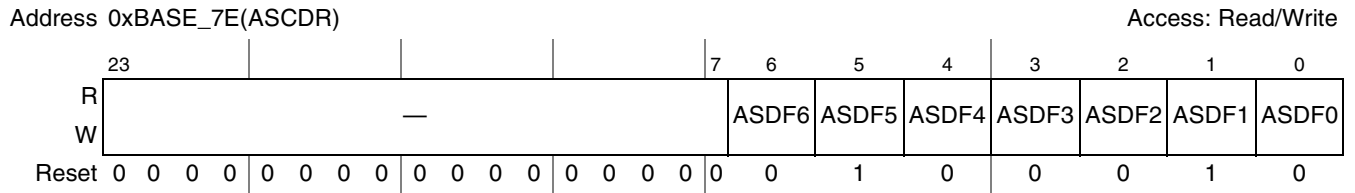


Figure 7-5. ASRC Clock Dividing Control Register (ASCDR)

Table 7-10. ASRC Control Division Registers Field Description

Bits	Name	Description
6–0	ASDF6–ASDF0	ASRC Divider Factor Defines the division factor of asrc_divider. This divided clock is used in the ASRC module as an enable. The reset value of ASDF is calculated as $199.68/5.644 - 1 = 34$ .



**Clock Generation Module (CGM)**

# Chapter 8

## General Purpose Input/Output (GPIO)

### 8.1 Introduction

The DSP56724/DSP56725 provides up to 79 bidirectional signals that can be configured as GPIO function signals or as peripheral function signals. All of these signals (except PG0, PG5, PG6, PG7, PG8, PH4, and PH4\_1) are GPIO by default after reset. The techniques for register programming for all GPIO functionality is similar between these interfaces. This section describes how signals may be used as GPIO.

### 8.2 Programming Model

The signals description section of this manual describes the special uses of these signals in detail. There are ten groups of these signals, which can be controlled separately or as groups and are shown in [Table 8-1](#).

**Table 8-1. GPIO Pins Summary Table**

GPIO Group		Function-Shared Pins	Available Pins in Different Packages	
			DSP5724	DSP56725
			144-Pin	80-Pin
1	Port C	Shared with ESAI signals	10	10
2	Port H	Shared with SHI signals	1	1
3	Port E	Shared with ESAI_1 signals	4	4
4	Timer GPIO	Shared with timer event counter (TEC) signals	0	0
5	Port G	Shared with S/PDIF signals, PLOCK, External IRQs, and some dedicated GPIO Port G signals.	7	5
6	Port C1	Shared with ESAI_2 signals	4	4
7	Port H1	Shared with SHI_1 signals	0	0
8	Port E1	Shared with ESAI_3 signals	10	9
9	Timer_1 GPIO	Shared with timer event counter (TEC_1) signals	0	0
10	Port A	Shared with EMC signals	27	0
Total GPIO Pin Number			65	33

## 8.2.1 Port C, Port E, Port C1, Port E1 Signals and Registers

Each of the 12 port signals that are not used as ESAI, ESAI\_1, ESAI\_2, or ESAI\_3 signals can be configured individually as a GPIO signal. The GPIO functionality of the port signals is controlled by three registers, described in [Chapter 9, “Enhanced Serial Audio Interface \(ESAI, ESAI\\_1, ESAI\\_2, ESAI\\_3\)”](#).

### Port C—ESAI Signal

- Port C control register (PCRC)
- Port C direction register (PRRC)
- Port C data register (PDRC)

### Port E—ESAI\_1 Signal

- Port E control register (PCRE)
- Port E direction register (PRRE)
- Port E data register (PDRE)

### Port C1—ESAI\_2 Signal

- Port C1 control register (PCRC1)
- Port C1 direction register (PRRC1)
- Port C1 data register (PDRC1)

### Port E1—ESAI\_3 Signal

- Port E control register (PCRE1)
- Port E direction register (PRRE1)
- Port E data register (PDRE1)

## 8.2.2 Port H Signals and Registers

The SHI’s HREQ can be configured as a GPIO signal. The GPIO functionality of Port H is controlled by three registers:

- Port H control register (PCRH)
- Port H direction register (PRRH)
- Port H data register (PDRH)

### 8.2.2.1 Port H Control Register (PCRH)

The read/write Port H Control Register (PCRH) and the Port H Direction Register (PRRH) together control the functionality of the dedicated GPIO pin. Each PH bit controls the functionality of the corresponding port pin. The port-pin configuration is shown in [Table 8-2](#). Hardware and software reset sets all PCRH bits.



### 8.2.2.2 Port H Direction Register (PRRH)

The read/write Port H Direction Register (PRRH) and the Port H Control Register (PCRH) together control the functionality of the dedicated GPIO pins. The port-pin configuration is shown in [Table 8-2](#). Hardware and software reset sets all PRRH bits.

**Table 8-2. PCRH and PRRH Bits Functionality**

PDH[i]	PH[i]	Port Pin[i] Function
0	0	Disconnected
0	1	GPIO input
1	0	GPIO output
1	1	Respective Functionality (SHI HREQ)

**Table 8-3. PORT H Registers Summary**

Register		23	22	21	20	19	18	17	16	15	14	13	12
		11	10	9	8	7	6	5	4	3	2	1	0
PDRH X:FFFF98	R/W	ETI1	ETO1	ERI1	ERO1	ETI0	ETO0	ERI0	ERO0				
	Reset	0	0	0	0	0	0	0	0				
	R/W								PD4				
	Reset								0				
PRRH X:FFFF99	R/W												
	Reset												
	R/W								PDH4				
	Reset								1				
PCRH X:FFFF9A	R/W												
	Reset												
	R/W								PH4				
	Reset								1				

### 8.2.2.3 Port H Data Register (PDRH)

The read/write Port H Data Register is used to read/write data to/from the dedicated GPIO pins. Bits PD(4:0) are used to read or write data from/to the corresponding port pins (if they are configured as GPIO).

- If a port pin [i] is configured as a GPIO input, then the corresponding PD[i] bit reflects the value present on this pin.
- If a port pin [i] is configured as a GPIO output, then the value written into the corresponding PD[i] bit is reflected on this pin.
- If a port pin [i] is configured as disconnected, then the corresponding PD[i] bit does not reflect the value present on this pin.

### 8.2.2.4 EXTAL Clocking Control for ESAI and ESAI\_1

The ESAI/EXTAL clock bits optionally direct the EXTAL clock to the ESAI clocking chain for generating the corresponding high frequency clock, bit clock and frame sync clock. There are 8 ESAI/EXTAL clock control bits as described in [Table 8-4](#). These 8 ESAI/EXTAL clock control bits are cleared upon reset.

**Table 8-4. ESAI/EXTAL Clock Control Bits**

ESAI / EXTAL Clock Control Bit	Description
ETI1	When the ETI1 bit is set, the EXTAL clock can be used to generate the ESAI_1 transmitter clocks: HCKT_1, SCKT_1 and FST_1. When the ETI1 bit is cleared, the Fosc clock can be used to generate the ESAI_1 transmitter clocks: HCKT_1, SCKT_1 and FST_1.
ETO1	When the ETO1 bit is set, the EXTAL clock is directed to the HCKT_1 pin. When the ETO1 bit is cleared, the EXTAL clock is not directed to the HCKT_1 pin.
ERI1	When the ERI1 bit is set, the EXTAL clock can be used to generate the ESAI_1 receiver clocks: HCKR_1, SCKR_1 and FSR_1. When the ERI1 bit is cleared, the Fosc clock can be used to generate the ESAI_1 transmitter clocks: HCKR_1, SCKR_1 and FSR_1.
ERO1	When the ERO1 bit is set, the EXTAL clock is directed to the HCKR_1 pin. When the ERO1 bit is cleared, the EXTAL clock is not directed to the HCKR_1 pin.
ETI0	When the ETI0 bit is set, the EXTAL clock can be used to generate the ESAI transmitter clocks: HCKT, SCKT and FST. When the ETI0 bit is cleared, the Fosc clock can be used to generate the ESAI transmitter clocks: HCKT, SCKT and FST.
ETO0	When the ETO0 bit is set, the EXTAL clock is directed to the HCKT pin. When the ETO0 bit is cleared, the EXTAL clock is not directed to the HCKT pin.
ERI0	When the ERI0 bit is set, the EXTAL clock can be used to generate the ESAI receiver clocks: HCKR, SCKR and FSR. When the ERI0 bit is cleared, the Fosc clock can be used to generate the ESAI transmitter clocks: HCKR, SCKR and FSR.
ERO0	When the ERO0 bit is set, the EXTAL clock is directed to the HCKR pin. When the ERO0 bit is cleared, the EXTAL clock is not directed to the HCKR pin.

### 8.2.3 Port H1 Signals and Registers

The SHI\_1's HREQ can be configured as a GPIO signal. The GPIO functionality of Port H1 is controlled by three registers:

- Port H1 control register (PCRH1)
- Port H direction register (PRRH1)
- Port H data register (PDRH1)

### 8.2.3.1 Port H1 Control Register (PCRH1)

The read/write Port H1 Control Register (PCRH1) and the Port H Direction Register (PRRH1) together control the functionality of the dedicated GPIO pin. The port-pin configuration is shown in [Table 8-5](#). Hardware and software reset sets all PCRH1 bits.

### 8.2.3.2 Port H1 Direction Register (PRRH1)

The read/write Port H1 Direction Register (PRRH1) and the Port H1 Control Register (PCRH1) together control the functionality of the dedicated GPIO pins. The port-pin configuration is shown in [Table 8-5](#). Hardware and software reset sets all PRRH1 bits.

**Table 8-5. PCRH1 and PRRH1 Bits Functionality**

PDH1[i]	PH1[i]	Port Pin[i] Function
0	0	Disconnected
0	1	GPIO input
1	0	GPIO output
1	1	Respective Functionality (SHI_1's HREQ)

**Table 8-6. PORT H1 Registers Summary**

Register		23	22	21	20	19	18	17	16	15	14	13	12
		11	10	9	8	7	6	5	4	3	2	1	0
PDRH_1 X:FFFF98	R/W	ETI1_1	ETO1_1	ERI1_1	ERO1_1	ETI0_1	ETO0_1	ERI0_1	ERO0_1				
	Reset	0	0	0	0	0	0	0	0				
	R/W								PD4_1				
	Reset								0				
PRRH_1 X:FFFF99	R/W												
	Reset								0				
	R/W								PDH4_1				
	Reset								1				
PCRH_1 X:FFFF9A	R/W												
	Reset												
	R/W								PH4_1				
	Reset								1				

### 8.2.3.3 Port H1 Data Register (PDRH1)

The read/write Port H1 Data Register is used to read/write data to/from the dedicated GPIO pins. Bits PD1(4:0) are used to read/write data from/to the corresponding port pins if they are configured as GPIO.

- If a port pin [i] is configured as a GPIO input, then the corresponding PD1[i] bit reflects the value present on this pin.
- If a port pin [i] is configured as a GPIO output, then the value written into the corresponding PD[i] bit is reflected on this pin.
- If a port pin [i] is configured as disconnected, then the corresponding PD[i] bit does not reflect the value present on this pin.

### 8.2.3.4 EXTAL Clocking Control for ESAI\_2 and ESAI\_3

ESAI/EXTAL clock bits optionally direct the EXTAL clock to the ESAI clocking chain to generate the corresponding high frequency clock, bit clock, and framesync clock. There are 8 ESAI/EXTAL clock control bits as described in [Table 8-7](#). These 8 ESAI/EXTAL clock control bits are cleared upon reset.

**Table 8-7. ESAI/EXTAL Clock Control Bit**

ESAI/EXTAL Clock Control Bit	Description
ETI3_1	When the ETI3_1 bit is set, the EXTAL clock can be used to generate the ESAI_3 transmitter clocks: HCKT_3, SCKT_3 and FST_3. When the ETI3_1 bit is cleared, the Fosc clock can be used to generate the ESAI_3 transmitter clocks: HCKT_3, SCKT_3 and FST_3.
ETO3_1	When the ETO3_1 bit is set, the EXTAL clock is directed to the HCKT_3 pin. When the ETO3_1 bit is cleared, the EXTAL clock is not directed to the HCKT_3 pin.
ERI3_1	When the ERI3_1 bit is set, the EXTAL clock can be used to generate the ESAI_3 receiver clocks: HCKR_3, SCKR_3 and FSR_3. When the ERI3_1 bit is cleared, the Fosc clock can be used to generate the ESAI_3 transmitter clocks: HCKR_3, SCKR_3 and FSR_3.
ERO3_1	When the ERO3_1 bit is set, the EXTAL clock is directed to the HCKR_3 pin. When the ERO3_1 bit is cleared, the EXTAL clock is not directed to the HCKR_3 pin.
ETI2_1	When the ETI2_1 bit is set, the EXTAL clock can be used to generate the ESAI_2 transmitter clocks: HCKT_2, SCKT_2 and FST_2. When the ETI2_1 bit is cleared, the Fosc clock can be used to generate the ESAI_2 transmitter clocks: HCKT_2, SCKT_2 and FST_2.
ETO2_1	When the ETO2_1 bit is set, the EXTAL clock is directed to the HCKT_2 pin. When the ETO2_1 bit is cleared, the EXTAL clock is not directed to the HCKT_2 pin.
ERI2_1	When the ERI2_1 bit is set, the EXTAL clock can be used to generate the ESAI_2 receiver clocks: HCKR_2, SCKR_2 and FSR_2. When the ERI2_1 bit is cleared, the Fosc clock can be used to generate the ESAI_2 transmitter clocks: HCKR_2, SCKR_2 and FSR_2.
ERO2_1	When the ERO2_1 bit is set, the EXTAL clock is directed to the HCKR_2 pin. When the ERO2_1 bit is cleared, the EXTAL clock is not directed to the HCKR_2 pin.

## 8.2.4 Port A Signals and Registers

Some of the EMC pins can also be configured individually as a GPIO PORT A signal. The GPIO functionality of Port A is controlled by the following registers:

- Port A control register (PCRA)
- Port A direction register (PRRA)
- Port A data register (PDRA)
- Port A control register1 (PCRA1)
- Port A direction register1 (PRRA1)
- Port A data register1 (PDRA1)

These GPIO port A registers are summarized in [Table 8-8](#).

**Table 8-8. GPIO Port A Registers Summary**

Register		23	22	21	20	19	18	17	16	15	14	13	12
		11	10	9	8	7	6	5	4	3	2	1	0
PDRA Y: FFFFF0	R/W	PD23	PD22	PD21	PD20	PD19	PD18	PD17	PD16	PD15	PD14	PD13	PD12
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
PRRA Y: FFFFF1	R/W	PDA23	PDA22	PDA21	PDA20	PDA19	PDA18	PDA17	PDA16	PDA15	PDA14	PDA13	PDA12
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	PDA11	PDA10	PDA9	PDA8	PDA7	PDA6	PDA5	PDA4	PDA3	PDA2	PDA1	PDA0
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
PCRA Y: FFFFF2	R/W	PA23	PA22	PA21	PA20	PA19	PA18	PA17	PA16	PA15	PA14	PA13	PA12
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
PDRA1 Y: FFFFF4	R/W												
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	R/W										PD26	PD25	PD24
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
PRRA1 Y: FFFFF5	R/W												
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	R/W										PDA26	PDA25	PDA24
	Reset	0	0	0	0	0	0	0	0	0	0	0	0

**Table 8-8. GPIO Port A Registers Summary (Continued)**

Register		23	22	21	20	19	18	17	16	15	14	13	12
		11	10	9	8	7	6	5	4	3	2	1	0
PCRA1 Y: FFFFF6	R/W												
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	R/W										PA26	PA25	PA24
	Reset	0	0	0	0	0	0	0	0	0	0	0	0

### 8.2.4.1 Port A Control Registers (PCRA, PCRA1)

The read/write 24-bit Port A Control Registers (PCRA, PCRA1) and the Port A Direction Registers (PRRA, PRRA1) together control the functionality of the Port A GPIO pins. Each of the PA[47:0] bits controls the functionality of the corresponding port pin. The port-pin configuration is shown in [Table 8-9](#).

### 8.2.4.2 Port A Direction Register (PRRA, PRRA1)

The read/write 24-bit Port A Direction Registers (PRRA, PRRA1) and the Port A Control Register (PCRA, PCRA1) together control the functionality of the dedicated GPIO pins. The port-pin configuration is shown in [Table 8-9](#).

**Table 8-9. PCRA/A1 and PRRA/A1 Bits**

PDA[i]	PA[i]	Port Pin[i] Function
0	0	Disconnected
0	1	GPIO input
1	0	GPIO output
1	1	EMC Function

### 8.2.4.3 Port A Data Register (PDRA, PDRA1)

The read/write 24-bit Port A Data Registers (PDRA, PDRA1) are used to read/write data from/to the dedicated GPIO pins. Bits PD(47:0) are used to read/write data from/to the corresponding port pins if they are configured as GPIO.

- If a port pin [i] is configured as a GPIO input, then the corresponding PD[i] bit reflects the value present on this pin.
- If a port pin [i] is configured as a GPIO output, then the value written into the corresponding PD[i] bit is reflected on this pin.
- If a port pin [i] is configured as disconnected, then the corresponding PD[i] bit does not reflect the value present on this pin.

## 8.2.5 Port G Signals and Registers

The 35 GPIO port G pins are function multiplexed with PLOCK, External Maskable Interrupts (IRQs), S/PDIF pins, and some of the dedicated GPIO port G pins. The GPIO functionality of Port G is controlled by six (6) registers: Port G control register (PCRG), Port G direction register (PRRG), Port G data register (PDRG), Port G control register 1(PCRG1), Port G direction register 1(PRRG1), Port G data register 1(PDRG1), shown in [Table 8-10](#).

**Table 8-10. GPIO Port G Registers Summary**

Register		23	22	21	20	19	18	17	16	15	14	13	12
		11	10	9	8	7	6	5	4	3	2	1	0
PDRG Y: FFFFFFF8	R/W			PD21	PD20	PD19	PD18			PD15	PD14	PD13	PD12
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	PD11	PD10	PD9	PD8	PD7	PD6	PD5		PD3	PD2	PD1	PD0
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
PRRG Y: FFFFFFF9	R/W			PDG21	PDG20	PDG19	PDG18			PDG15	PDG14	PDG13	PDG12
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	PDG11	PDG10	PDG9	PDG8	PDG7	PDG6	PDG5		PDG3	PDG2	PDG1	PDG0
	Reset	0	0	0	1	1	1	1	0	0	0	0	1
PCRG Y: FFFFFFFA	R/W			PG21	PG20	PG19	PG18			PG15	PG14	PG13	PG12
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	PG11	PG10	PG9	PG8	PG7	PG6	PG5		PG3	PG2	PG1	PG0
	Reset	0	0	0	1	1	1	1	0	0	0	0	1
PDRG1 Y: FFFFFFFC	R/W									PD39	PD38	PD37	PD36
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	PD35	PD34	PD33	PD32	PD31	PD30	PD29	PD28	PD27	PD26	PD25	PD24
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
PRRG1 Y: FFFFFFFD	R/W									PDG39	PDG38	PDG37	PDG36
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	PDG35	PDG34	PDG33	PDG32	PDG31	PDG30	PDG29	PDG28	PDG27	PDG26	PDG25	PDG24
	Reset	0	0	0	0	0	0	0	0	0	0	0	0
PCRG1 Y: FFFFFFFE	R/W									PG39	PG38	PG37	PG36
	Res	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	PG35	PG34	PG33	PG32	PG31	PG30	PG29	PG28	PG27	PG26	PG25	PG24
	Res	0	0	0	0	0	0	0	0	0	0	0	0

### 8.2.5.1 Port G Control Registers (PCRG, PCRG1)

The read/write 24-bit Port G Control Registers (PCRG, PCRG1) and the Port G Direction Registers (PRRG, PRRG1) together control the functionality of the Port G GPIO pins. Each of the PG[39:0] bits controls the functionality of the corresponding port pin. The port-pin configuration are shown in [Table 8-11](#).

External Interrupt pins and the PLL lock output will act as non-GPIO functions after reset; so corresponding bits of PRRG and PCRG will be set to “1” after reset.

### 8.2.5.2 Port G Direction Register (PRRG, PRRG1)

The read/write 24-bit Port G Direction Registers (PRRG, PRRG1) and the Port G Control Register (PCRG, PCRG1) together control the functionality of the dedicated GPIO pins. For the port-pin configuration, see [Table 8-9](#).

External Interrupt pins and the PLL lock output will act as non-GPIO functions after reset; so corresponding bits of PRRG and PCRG will be set to “1” after reset. The port-pin configuration are shown in [Table 8-11](#).

**Table 8-11. PCRG and PRRG Bits Functionality**

PDG[i]	PG[i]	Port Pin[i] Function
0	0	Disconnected
0	1	GPIO input
1	0	GPIO output
1	1	S/PDIF,... function

### 8.2.5.3 Port G Data Register (PDRG, PDRG1)

The read/write 24-bit Port G Data Registers (PDRG, PDRG1) are used to read/write data from/to the dedicated GPIO pins. Bits PD[39:0] are used to read/write data from/to the corresponding port pins if they are configured as GPIO.

- If a port pin [i] is configured as a GPIO input, then the corresponding PD[i] bit reflects the value present on this pin.
- If a port pin [i] is configured as a GPIO output, then the value written into the corresponding PD[i] bit is reflected on this pin.
- If a port pin [i] is configured as disconnected, then the corresponding PD[i] bit does not reflect the value present on this pin.

### 8.2.6 Timer Event Counter Signals

There are two identical Timer Event Counter (TEC) blocks (TEC, TEC\_1), one timer block for each DSP core. The timer event counter signals are controlled by the appropriate timer control status register (TCSR), which is described in [Chapter 11, “Triple Timer Module \(TEC, TEC\\_1\).”](#)



## Chapter 9

# Enhanced Serial Audio Interface (ESAI, ESAI\_1, ESAI\_2, ESAI\_3)

The enhanced serial audio interface (ESAI) provides a full-duplex serial port for serial communication with a variety of serial devices, including industry-standard codecs, S/PDIF transceivers, and other DSPs. The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator. It is a superset of the 56300 Family ESSI peripheral and of the 56000 Family SAI peripheral.

### NOTE

The DSP56724/DSP56725 has four ESAI modules: ESAI, ESAI\_1, ESAI\_2, and ESAI\_3. ESAI and ESAI\_1 are used by DSP Core-0; ESAI\_2 and ESAI\_3 are used by DSP Core-1. The only difference between ESAI and ESAI\_2 is that ESAI/ESAI\_1 is used by DSP Core-0 and that ESAI\_2/ESAI\_3 is used by Core-1. There are no other differences. This chapter describes the ESAI module.

The ESAI block diagram is shown in [Figure 9-1](#). The ESAI is called synchronous because all serial transfers are synchronized to a clock. Additional synchronization signals are used to delineate the word frames. The normal mode of operation is used to transfer data at a periodic rate, one word per period. The network mode is similar in that it is also intended for periodic transfers; however, it supports up to 32 words (time slots) per period. This mode can be used to build time division multiplexed (TDM) networks. In contrast, the on-demand mode is intended for non-periodic transfers of data and to transfer data serially at high speed when the data becomes available.

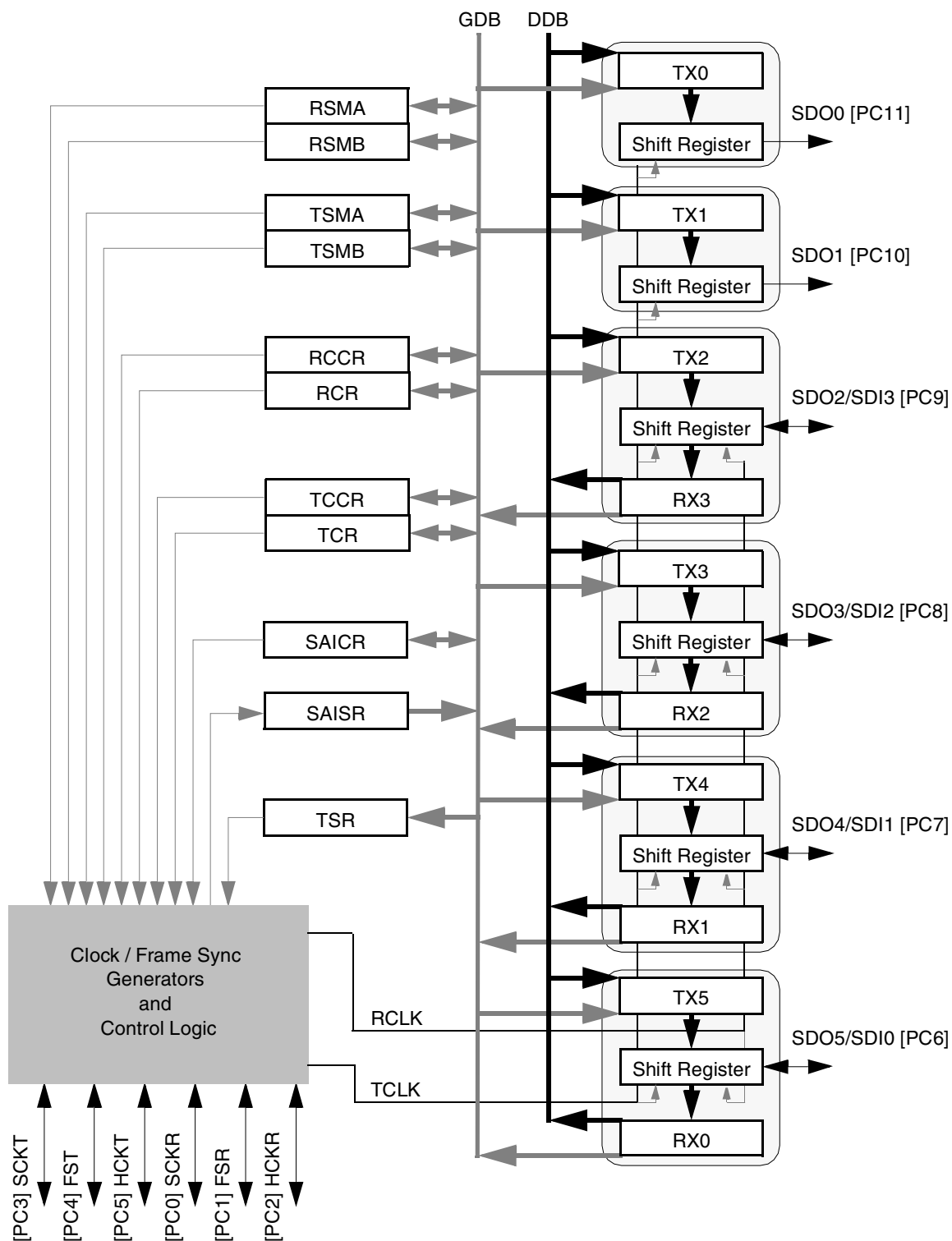


Figure 9-1. ESAI Block Diagram

## 9.1 ESAI Data and Control Pins

Three to twelve pins are required for operation, depending on the operating mode selected and the number of transmitters and receivers enabled. The SDO0 and SDO1 pins are used by transmitters 0 and 1 only. The SDO2/SDI3, SDO3/SDI2, SDO4/SDI1 and SDO5/SDI0 pins are shared by transmitters 2 to 5 with receivers 0 to 3. The actual mode of operation is selected under software control. All transmitters operate fully synchronized under control of the same transmitter clock signals. All receivers operate fully synchronized under control of the same receiver clock signals.

### 9.1.1 Serial Transmit 0 Data Pin (SDO0)

SDO0 is used for transmitting data from the TX0 serial transmit shift register. SDO0 is an output when data is being transmitted from the TX0 shift register. In the on-demand mode with an internally generated bit clock, the SDO0 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO0 may be programmed as a general-purpose I/O pin (PC11) when the ESAI SDO0 function is not being used.

### 9.1.2 Serial Transmit 1 Data Pin (SDO1)

SDO1 is used for transmitting data from the TX1 serial transmit shift register. SDO1 is an output when data is being transmitted from the TX1 shift register. In the on-demand mode with an internally generated bit clock, the SDO1 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO1 may be programmed as a general-purpose I/O pin (PC10) when the ESAI SDO1 function is not being used.

### 9.1.3 Serial Transmit 2/Receive 3 Data Pin (SDO2/SDI3)

SDO2/SDI3 is used as the SDO2 for transmitting data from the TX2 serial transmit shift register when programmed as a transmitter pin, or as the SDI3 signal for receiving serial data to the RX3 serial receive shift register when programmed as a receiver pin. SDO2/SDI3 is an input when data is being received by the RX3 shift register. SDO2/SDI3 is an output when data is being transmitted from the TX2 shift register. In the on-demand mode with an internally generated bit clock, the SDO2/SDI3 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO2/SDI3 may be programmed as a general-purpose I/O pin (PC9) when the ESAI SDO2 and SDI3 functions are not being used.

### 9.1.4 Serial Transmit 3/Receive 2 Data Pin (SDO3/SDI2)

SDO3/SDI2 is used as the SDO3 signal for transmitting data from the TX3 serial transmit shift register when programmed as a transmitter pin, or as the SDI2 signal for receiving serial data to the RX2 serial receive shift register when programmed as a receiver pin. SDO3/SDI2 is an input when data is being received by the RX2 shift register. SDO3/SDI2 is an output when data is being transmitted from the TX3 shift register. In the on-demand mode with an internally generated bit clock, the SDO3/SDI2 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO3/SDI2 may be programmed as a general-purpose I/O pin (PC8) when the ESAI SDO3 and SDI2 functions are not being used.

### 9.1.5 Serial Transmit 4/Receive 1 Data Pin (SDO4/SDI1)

SDO4/SDI1 is used as the SDO4 signal for transmitting data from the TX4 serial transmit shift register when programmed as transmitter pin, or as the SDI1 signal for receiving serial data to the RX1 serial receive shift register when programmed as a receiver pin. SDO4/SDI1 is an input when data is being received by the RX1 shift register. SDO4/SDI1 is an output when data is being transmitted from the TX4 shift register. In the on-demand mode with an internally generated bit clock, the SDO4/SDI1 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO4/SDI1 may be programmed as a general-purpose I/O pin (PC7) when the ESAI SDO4 and SDI1 functions are not being used.

### 9.1.6 Serial Transmit 5/Receive 0 Data Pin (SDO5/SDI0)

SDO5/SDI0 is used as the SDO5 signal for transmitting data from the TX5 serial transmit shift register when programmed as transmitter pin, or as the SDI0 signal for receiving serial data to the RX0 serial shift register when programmed as a receiver pin. SDO5/SDI0 is an input when data is being received by the RX0 shift register. SDO5/SDI0 is an output when data is being transmitted from the TX5 shift register. In the on-demand mode with an internally generated bit clock, the SDO5/SDI0 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO5/SDI0 may be programmed as a general-purpose I/O pin (PC6) when the ESAI SDO5 and SDI0 functions are not being used

### 9.1.7 Receiver Serial Clock (SCKR)

SCKR is a bidirectional pin providing the receivers serial bit clock for the ESAI interface. The direction of this pin is determined by the RCKD bit in the RCCR register. The SCKR operates as a clock input or output used by all the enabled receivers in the asynchronous mode (SYN=0), or as serial flag 0 pin in the synchronous mode (SYN=1).

When this pin is configured as serial flag pin, its direction is determined by the RCKD bit in the RCCR register. When configured as the output flag OF0, this pin reflects the value of the OF0 bit in the SAICR register, and the data in the OF0 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When this pin is configured as the input flag IF0, the data value at the pin is stored in the IF0 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

SCKR may be programmed as a general-purpose I/O pin (PC0) when the ESAI SCKR function is not being used.

#### NOTE

Although the external ESAI serial clock can be independent of and asynchronous to the DSP system clock, the DSP clock frequency must be at least four times the external ESAI serial clock frequency and each ESAI serial clock phase must exceed the minimum of two DSP clock periods.

For SCKR pin mode definitions, see [Table 9-7](#).

[Table 9-1](#) provides a list of asynchronous-mode receiver clock sources. For more information about EXTAL/ESAI0/2 clocking control bits (ERI0,ERO0), see [Chapter 8, “General Purpose Input/Output \(GPIO\).”](#)

**Table 9-1. Receiver Clock Sources (Asynchronous Mode Only)**

RHCKD	RFSD	RCKD	ERI0	ERO0	Receiver Bit-Clock Source	OUTPUTS		
0	0	0	N/A	N/A	SCKR	—	—	—
0	0	1	N/A	N/A	HCKR	—	—	SCKR
0	1	0	N/A	N/A	SCKR	—	FSR	—
0	1	1	N/A	N/A	HCKR	—	FSR	SCKR
1	0	0	0	0	SCKR	HCKR	—	—
1	0	0	0	1	SCKR	HCKR	—	—
1	0	0	1	0	SCKR	HCKR	—	—
1	0	0	1	1	SCKR	HCKR	—	—
1	0	1	0	0	Fsys	HCKR	—	SCKR
1	0	1	0	1	Fsys	HCKR	—	SCKR
1	0	1	1	0	EXTAL	HCKR	—	SCKR

**Table 9-1. Receiver Clock Sources (Asynchronous Mode Only) (Continued)**

RHCKD	RFSD	RCKD	ERIO	ERO0	Receiver Bit-Clock Source	OUTPUTS		
1	0	1	1	1	EXTAL	HCKR	—	SCKR
1	1	0	0	0	SCKR	HCKR	FSR	—
1	1	0	0	1	SCKR	HCKR	FSR	—
1	1	0	1	0	SCKR	HCKR	FSR	—
1	1	0	1	1	SCKR	HCKR	FSR	—
1	1	1	0	0	Fsys	HCKR	FSR	SCKR
1	1	1	0	1	Fsys	HCKR	FSR	SCKR
1	1	1	1	0	EXTAL	HCKR	FSR	SCKR
1	1	1	1	1	EXTAL	HCKR	FSR	SCKR

### 9.1.8 Transmitter Serial Clock (SCKT)

SCKT is a bidirectional pin providing the transmitters serial bit-clock for the ESAI interface. The direction of this pin is determined by the TCKD bit in the TCCR register. The SCKT is a clock input or output used by all the enabled transmitters in the asynchronous mode (SYN=0) or by all the enabled transmitters and receivers in the synchronous mode (SYN=1).

Table 9-2 provides a list of asynchronous-mode transmitter clock sources.

**Table 9-2. Transmitter Clock Sources (Asynchronous Mode Only)**

THCKD	TFSD	TCKD	ETIO	ETO0	Transmitter Bit Clock Source	OUTPUTS		
0	0	0	N/A	N/A	SCKT	—	—	—
0	0	1	N/A	N/A	HCKT	—	—	SCKT
0	1	0	N/A	N/A	SCKT	—	FST	—
0	1	1	N/A	N/A	HCKT	—	FST	SCKT
1	0	0	0	0	SCKT	HCKT	—	—
1	0	0	0	1	SCKT	HCKT	—	—
1	0	0	1	0	SCKT	HCKT	—	—
1	0	0	1	1	SCKT	HCKT	—	—
1	0	1	0	0	Fsys	HCKT	—	SCKT
1	0	1	0	1	Fsys	HCKT	—	SCKT
1	0	1	1	0	EXTAL	HCKT	—	SCKT
1	0	1	1	1	EXTAL	HCKT	—	SCKT

**Table 9-2. Transmitter Clock Sources (Asynchronous Mode Only) (Continued)**

THCKD	TFSD	TCKD	ETI0	ETO0	Transmitter Bit Clock Source	OUTPUTS		
						HCKT	FST	SCKT
1	1	0	0	0	SCKR	HCKT	FST	—
1	1	0	0	1	SCKR	HCKT	FST	—
1	1	0	1	0	SCKR	HCKT	FST	—
1	1	0	1	1	SCKR	HCKT	FST	—
1	1	1	0	0	Fsys	HCKT	FST	SCKT
1	1	1	0	1	Fsys	HCKT	FST	SCKT
1	1	1	1	0	EXTAL	HCKT	FST	SCKT
1	1	1	1	1	EXTAL	HCKT	FST	SCKT

SCKT may be programmed as a general-purpose I/O pin (PC3) when the ESAI SCKT function is not being used. For more information about EXTAL/ESAI0/2 clocking control bits (ETI0,ETO0), see [Chapter 8](#), “General Purpose Input/Output (GPIO).”

#### NOTE

Although the external ESAI serial clock can be independent of and asynchronous to the DSP system clock, the DSP clock frequency must be at least four times the external ESAI serial clock frequency and each ESAI serial clock phase must exceed the minimum of 2 DSP clock periods.

### 9.1.9 Frame Sync for Receiver (FSR)

FSR is a bidirectional pin providing the receivers frame sync signal for the ESAI interface. The direction of this pin is determined by the RFSD bit in RCR register. In the asynchronous mode (SYN=0), the FSR pin operates as the frame sync input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1). For FSR pin mode definitions, see [Table 9-8](#); for receiver clock signals, see [Table 9-1](#).

When this pin is configured as serial flag pin, its direction is determined by the RFSD bit in the RCCR register. When configured as the output flag OF1, this pin reflects the value of the OF1 bit in the SAICR register, and the data in the OF1 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When configured as the input flag IF1, the data value at the pin is stored in the IF1 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

FSR may be programmed as a general-purpose I/O pin (PC1) when the ESAI FSR function is not being used.

### 9.1.10 Frame Sync for Transmitter (FST)

FST is a bidirectional pin providing the frame sync for both the transmitters and receivers in the synchronous mode (SYN=1) and for the transmitters only in asynchronous mode (SYN=0) (see [Table 9-2](#)). The direction of this pin is determined by the TFSD bit in the TCR register. When configured as an output, this pin is the internally generated frame sync signal. When configured as an input, this pin receives an external frame sync signal for the transmitters (and the receivers in synchronous mode).

FST may be programmed as a general-purpose I/O pin (PC4) when the ESAI FST function is not being used.

### 9.1.11 High Frequency Clock for Transmitter (HCKT)

HCKT is a bidirectional pin providing the transmitters high frequency clock for the ESAI interface. The direction of this pin is determined by the THCKD bit in the TCCR register. In the asynchronous mode (SYN=0), the HCKT pin operates as the high frequency clock input or output used by all enabled transmitters. In the synchronous mode (SYN=1), it operates as the high frequency clock input or output used by all enabled transmitters and receivers. When programmed as input this pin is used as an alternative high frequency clock source to the ESAI transmitter rather than the DSP main clock. When programmed as output it can serve as a high frequency sample clock (to external DACs for example) or as an additional system clock (see [Table 9-1](#)).

HCKT may be programmed as a general-purpose I/O pin (PC5) when the ESAI HCKT function is not being used.

### 9.1.12 High Frequency Clock for Receiver (HCKR)

HCKR is a bidirectional pin providing the receivers high frequency clock for the ESAI interface. The direction of this pin is determined by the RHCKD bit in the RCCR register. In the asynchronous mode (SYN=0), the HCKR pin operates as the high frequency clock input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as the serial flag 2 pin. For HCKR pin mode definitions, see [Table 9-9](#); for receiver clock signals, see [Table 9-1](#).

When this pin is configured as serial flag pin, its direction is determined by the RHCKD bit in the RCCR register. When configured as the output flag OF2, this pin reflects the value of the OF2 bit in the SAICR register, and the data in the OF2 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When configured as the input flag IF2, the data value at the pin is stored in the IF2 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

HCKR may be programmed as a general-purpose I/O pin (PC2) when the ESAI HCKR function is not being used.



## 9.2 ESAI Programming Model

The ESAI can be viewed as five control registers, one status register, six transmit data registers, four receive data registers, two transmit slot mask registers, two receive slot mask registers and a special-purpose time slot register. The following paragraphs give detailed descriptions and operations of each bit in the ESAI registers.

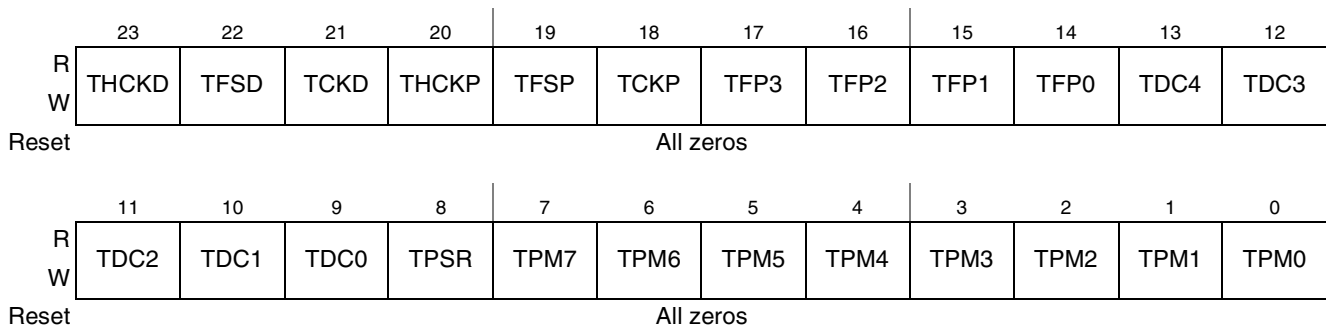
The ESAI pins can also function as GPIO pins (Port C), described in [Section 9.4, “GPIO—Pins and Registers.”](#)

### 9.2.1 ESAI Transmitter Clock Control Register (TCCR)

The read/write Transmitter Clock Control Register (TCCR) controls the ESAI transmitter clock generator bit and frame sync rates, the bit clock and high frequency clock sources and the directions of the HCKT, FST and SCKT signals. In the synchronous mode (SYN=1), the bit clock defined for the transmitter determines the receiver bit clock as well. TCCR also controls the number of words per frame for the serial data. Hardware and software reset clear all the bits of the TCCR register. [Figure 9-2](#) shows the TCCR.

Address X: 0xFFFFB6

Access: Read/Write



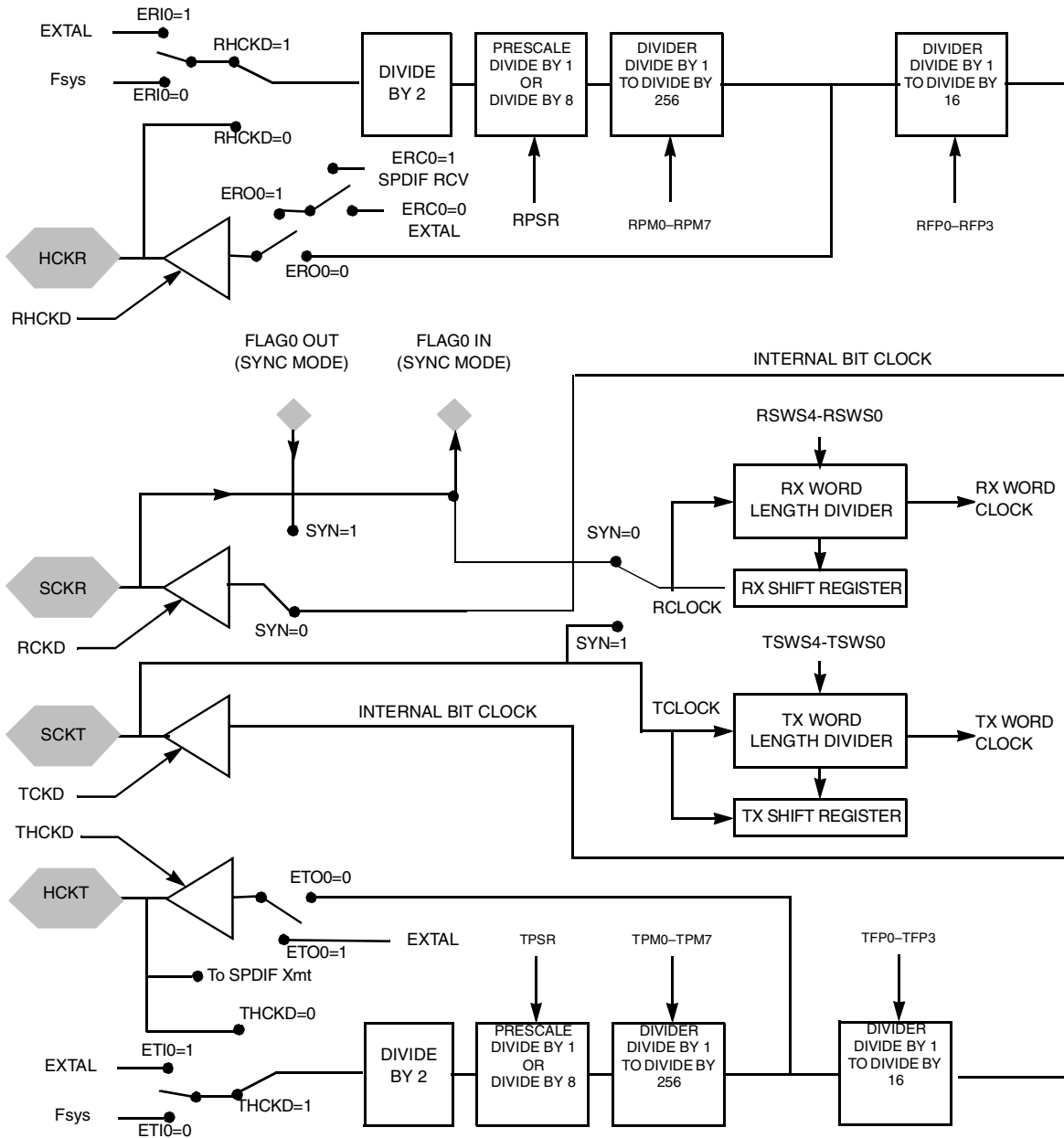
**Figure 9-2. TCCR Fields**

#### NOTE

Care should be taken in asynchronous mode whenever the frame sync clock (FSR, FST) is not sourced directly from its associated bit clock (SCKR,SCKT). Proper phase relationships must be maintained between these clocks to guarantee proper operation of the ESAI.

### 9.2.1.1 TCCR Transmit Prescale Modulus Select (TPM7–TPM0)—Bits 7–0

The TPM7–TPM0 bits specify the divide ratio of the prescale divider in the ESAI transmitter clock generator. A divide ratio from 1 to 256 (TPM[7:0]=0x00 to 0xFF) may be selected. The bit clock output is available at the transmit serial bit clock (SCKT) pin of the DSP. The bit clock output is also available internally for use as the bit clock to shift the transmit and receive shift registers. The ESAI transmit clock generator functional diagram is shown in Figure 9-3.



**Notes:**

1. ETIx, ETOx, ERIx and EROx bit descriptions are covered in [Section 8.2.2.3, “Port H Data Register \(PDRH\).”](#)
2. Fsys is the DSP56300 Core internal clock frequency.

**Figure 9-3. ESAI Clock Generator Functional Block Diagram**

### 9.2.1.2 TCCR Transmit Prescaler Range (TPSR)—Bit 8

The TPSR bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When TPSR is set, the fixed prescaler is bypassed. When TPSR is cleared, the fixed divide-by-eight prescaler is operational (see [Figure 9-3](#)). The maximum internally generated bit clock frequency is  $F_{sys}/4$ ; the minimum internally generated bit clock frequency is  $F_{sys}/(2 \times 8 \times 256) = F_{sys}/4096$ .

#### NOTE

Do not use the combination  $TPSR=1$ ,  $TPM7-TPM0=0x00$ , and  $TFP3-TFP0=0x0$  which causes synchronization problems when using the internal DSP clock as source ( $TCKD=1$  or  $THCKD=1$ ).

### 9.2.1.3 TCCR Tx Frame Rate Divider Control (TDC4–TDC0)—Bits 13–9

The TDC4–TDC0 bits control the divide ratio for the programmable frame rate dividers used to generate the transmitter frame clocks.

In network mode, this ratio may be interpreted as the number of words per frame minus one. The divide ratio may range from 2 to 32 ( $TDC[4:0]=00001$  to  $11111$ ) for network mode. A divide ratio of one ( $TDC[4:0]=00000$ ) in network mode is a special case (on-demand mode).

In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 ( $TDC[4:0]=00000$  to  $11111$ ) for normal mode. In normal mode, a divide ratio of 1 ( $TDC[4:0]=00000$ ) provides continuous periodic data word transfers. A bit-length frame sync ( $TFSL=1$ ) must be used in this case.

The ESAI frame sync generator functional diagram is shown in Figure 9-4.

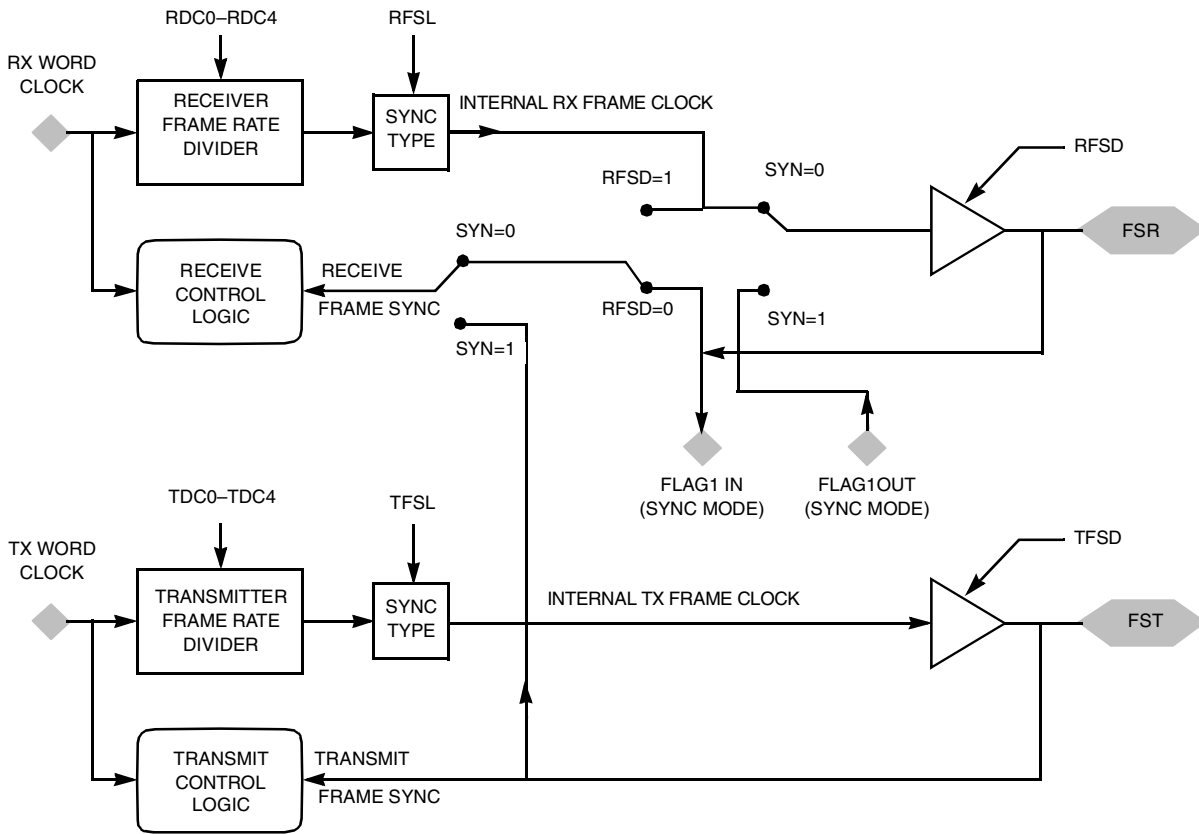


Figure 9-4. ESAI Frame Sync Generator Functional Block Diagram

### 9.2.1.4 TCCR Tx High Frequency Clock Divider (TFP3-TFP0)—Bits 17–14

The TFP3–TFP0 bits control the divide ratio of the transmitter high frequency clock to the transmitter serial bit clock when the source of the high frequency clock and the bit clock is the internal DSP clock. When the HCKT input is being driven from an external high frequency clock, the TFP3–TFP0 bits specify an additional division ratio in the clock divider chain. Table 9-3 shows the specification for the divide ratio. Figure 9-3 shows the ESAI high frequency clock generator functional diagram.

Table 9-3. Transmitter High Frequency Clock Divider

TFP3–TFP0	Divide Ratio
0x0	1
0x1	2
0x2	3
0x3	4
...	...
0xF	16

### 9.2.1.5 TCCR Transmit Clock Polarity (TCKP)—Bit 18

The Transmitter Clock Polarity (TCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If TCKP is cleared the data and the frame sync are clocked out on the rising edge of the transmit bit clock and latched in on the falling edge of the transmit bit clock. If TCKP is set the falling edge of the transmit clock is used to clock the data out and frame sync and the rising edge of the transmit clock is used to latch the data and frame sync in.

### 9.2.1.6 TCCR Transmit Frame Sync Polarity (TFSP)—Bit 19

The Transmitter Frame Sync Polarity (TFSP) bit determines the polarity of the transmit frame sync signal. When TFSP is cleared, the frame sync signal polarity is positive, that is, the frame start is indicated by a high level on the frame sync pin. When TFSP is set, the frame sync signal polarity is negative, that is, the frame start is indicated by a low level on the frame sync pin.

### 9.2.1.7 TCCR Transmit High Frequency Clock Polarity (THCKP)—Bit 20

The Transmitter High Frequency Clock Polarity (THCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If THCKP is cleared the data and the frame sync are clocked out on the rising edge of the transmit high frequency bit clock and latched in on the falling edge of the transmit bit clock. If THCKP is set the falling edge of the transmit clock is used to clock the data out and frame sync and the rising edge of the transmit clock is used to latch the data and frame sync in.

### 9.2.1.8 TCCR Transmit Clock Source Direction (TCKD)—Bit 21

The Transmitter Clock Source Direction (TCKD) bit selects the source of the clock signal used to clock the transmit shift registers in the asynchronous mode (SYN=0) and the transmit shift registers and the receive shift registers in the synchronous mode (SYN=1). When TCKD is set, the internal clock source becomes the bit clock for the transmit shift registers and word length divider and is the output on the SCKT pin. When TCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCKT pin, and an external clock source may drive this pin (see [Table 9-2](#)).

### 9.2.1.9 TCCR Transmit Frame Sync Signal Direction (TFSD)—Bit 22

TFSD controls the direction of the FST pin. When TFSD is cleared, FST is an input; when TFSD is set, FST is an output (see [Table 9-2](#)).

### 9.2.1.10 TCCR Transmit High Frequency Clock Direction (THCKD)—Bit 23

THCKD controls the direction of the HCKT pin. When THCKD is cleared, HCKT is an input; when THCKD is set, HCKT is an output (see [Table 9-2](#)).

## 9.2.2 ESAI Transmit Control Register (TCR)

The read/write Transmit Control Register (TCR) controls the ESAI transmitter section. Interrupt enable bits for the transmitter section are provided in this control register. Operating modes are also selected in this register. Figure 9-5 shows the register.

Address X: 0xFFFFB5

Access: Read/Write

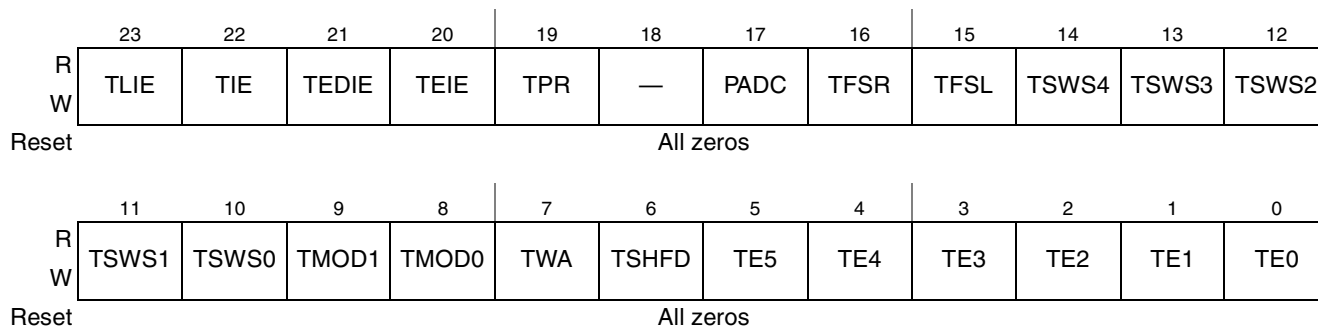


Figure 9-5. TCR Register

### NOTE

Hardware and software reset clear all the bits in the TCR register.

#### 9.2.2.1 TCR ESAI Transmit 0 Enable (TE0)—Bit 0

TE0 enables the transfer of data from TX0 to the transmit shift register #0. When TE0 is set and a frame sync is detected, the transmit #0 portion of the ESAI is enabled for that frame. When TE0 is cleared, the transmitter #0 is disabled after completing transmission of data currently in the ESAI transmit shift register. The SDO0 output is tri-stated, and any data present in TX0 is not transmitted, that is, data can be written to TX0 with TE0 cleared, but data is not transferred to the transmit shift register #0.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE0 and setting it again disables the transmitter #0 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO0 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE0 can be left enabled.

#### 9.2.2.2 TCR ESAI Transmit 1 Enable (TE1)—Bit 1

TE1 enables the transfer of data from TX1 to the transmit shift register #1. When TE1 is set and a frame sync is detected, the transmit #1 portion of the ESAI is enabled for that frame. When TE1 is cleared, the transmitter #1 is disabled after completing transmission of data currently in the ESAI transmit shift register. The SDO1 output is tri-stated, and any data present in TX1 is not transmitted, that is, data can be written to TX1 with TE1 cleared, but data is not transferred to the transmit shift register #1.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE1 and setting it again disables the transmitter #1 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO1 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE1 can be left enabled.

### 9.2.2.3 TCR ESAI Transmit 2 Enable (TE2)—Bit 2

TE2 enables the transfer of data from TX2 to the transmit shift register #2. When TE2 is set and a frame sync is detected, the transmit #2 portion of the ESAI is enabled for that frame. When TE2 is cleared, the transmitter #2 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to TX2 when TE2 is cleared but the data is not transferred to the transmit shift register #2.

The SDO2/SDI3 pin is the data input pin for RX3 if TE2 is cleared and RE3 in the RCR register is set. If both RE3 and TE2 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE3 and TE2 should not be set at the same time.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE2 and setting it again disables the transmitter #2 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO2/SDI3 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE2 can be left enabled.

### 9.2.2.4 TCR ESAI Transmit 3 Enable (TE3)—Bit 3

TE3 enables the transfer of data from TX3 to the transmit shift register #3. When TE3 is set and a frame sync is detected, the transmit #3 portion of the ESAI is enabled for that frame. When TE3 is cleared, the transmitter #3 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to TX3 when TE3 is cleared but the data is not transferred to the transmit shift register #3.

The SDO3/SDI2 pin is the data input pin for RX2 if TE3 is cleared and RE2 in the RCR register is set. If both RE2 and TE3 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE2 and TE3 should not be set at the same time.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE3 and setting it again disables the transmitter #3 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO3/SDI2 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE3 can be left enabled.

### 9.2.2.5 TCR ESAI Transmit 4 Enable (TE4)—Bit 4

TE4 enables the transfer of data from TX4 to the transmit shift register #4. When TE4 is set and a frame sync is detected, the transmit #4 portion of the ESAI is enabled for that frame. When TE4 is cleared, the

transmitter #4 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to TX4 when TE4 is cleared but the data is not transferred to the transmit shift register #4.

The SDO4/SDI1 pin is the data input pin for RX1 if TE4 is cleared and RE1 in the RCR register is set. If both RE1 and TE4 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE1 and TE4 should not be set at the same time.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE4 and setting it again disables the transmitter #4 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO4/SDI1 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE4 can be left enabled.

### 9.2.2.6 TCR ESAI Transmit 5 Enable (TE5)—Bit 5

TE5 enables the transfer of data from TX5 to the transmit shift register #5. When TE5 is set and a frame sync is detected, the transmit #5 portion of the ESAI is enabled for that frame. When TE5 is cleared, the transmitter #5 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to TX5 when TE5 is cleared but the data is not transferred to the transmit shift register #5.

The SDO5/SDI0 pin is the data input pin for RX0 if TE5 is cleared and RE0 in the RCR register is set. If both RE0 and TE5 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE0 and TE5 should not be set at the same time.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE5 and setting it again disables the transmitter #5 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO5/SDI0 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE5 can be left enabled.

### 9.2.2.7 TCR Transmit Shift Direction (TSHFD)—Bit 6

The TSHFD bit causes the transmit shift registers to shift data out MSB first when TSHFD equals zero or LSB first when TSHFD equals one (see [Figure 9-13](#) and [Figure 9-14](#)).

### 9.2.2.8 TCR Transmit Word Alignment Control (TWA)—Bit 7

The Transmitter Word Alignment Control (TWA) bit defines the alignment of the data word in relation to the slot. This is relevant for the cases where the word length is shorter than the slot length. If TWA is cleared, the data word is left-aligned in the slot frame during transmission. If TWA is set, the data word is right-aligned in the slot frame during transmission.



Because the data word is shorter than the slot length, the data word is extended until achieving the slot length, according to the following:

1. If the data word is left-aligned (TWA=0), and zero padding is disabled (PADC=0), the last data bit is repeated after the data word has been transmitted. If zero padding is enabled (PADC=1), zeroes are transmitted after the data word has been transmitted.
2. If the data word is right-aligned (TWA=1), and zero padding is disabled (PADC=0), the first data bit is repeated before the transmission of the data word. If zero padding is enabled (PADC=1), zeroes are transmitted before the transmission of the data word.

### 9.2.2.9 TCR Transmit Network Mode Control (TMOD1–TMOD0)—Bits 9–8

The TMOD1 and TMOD0 bits are used to define the network mode of ESAI transmitters, as shown in [Table 9-4](#). In the normal mode, the frame rate divider determines the word transfer rate—one word is transferred per frame sync during the frame sync time slot, as shown in [Figure 9-6](#). In network mode, it is possible to transfer a word for every time slot, as shown in [Figure 9-6](#). For further details, see [Section 9.3, “Operating Modes.”](#)

To comply with AC-97 specifications, TSWS4–TSWS0 should be set to 00011 (20-bit slot, 20-bit word length), TFSL and TFSR should be cleared, and TDC4–TDC0 should be set to 0x0C (13 words in frame). If TMOD[1:0]=0x11 and the above recommendations are followed, the first slot and word will be 16 bits long, and the next 12 slots and words will be 20 bits long, as required by the AC97 protocol.

**Table 9-4. Transmit Network Mode Selection**

TMOD1	TMOD0	TDC4–TDC0	Transmitter Network Mode
0	0	0x0–0x1F	Normal Mode
0	1	0x0	On-Demand Mode
0	1	0x1–0x1F	Network Mode
1	0	X	Reserved
1	1	0x0C	AC97

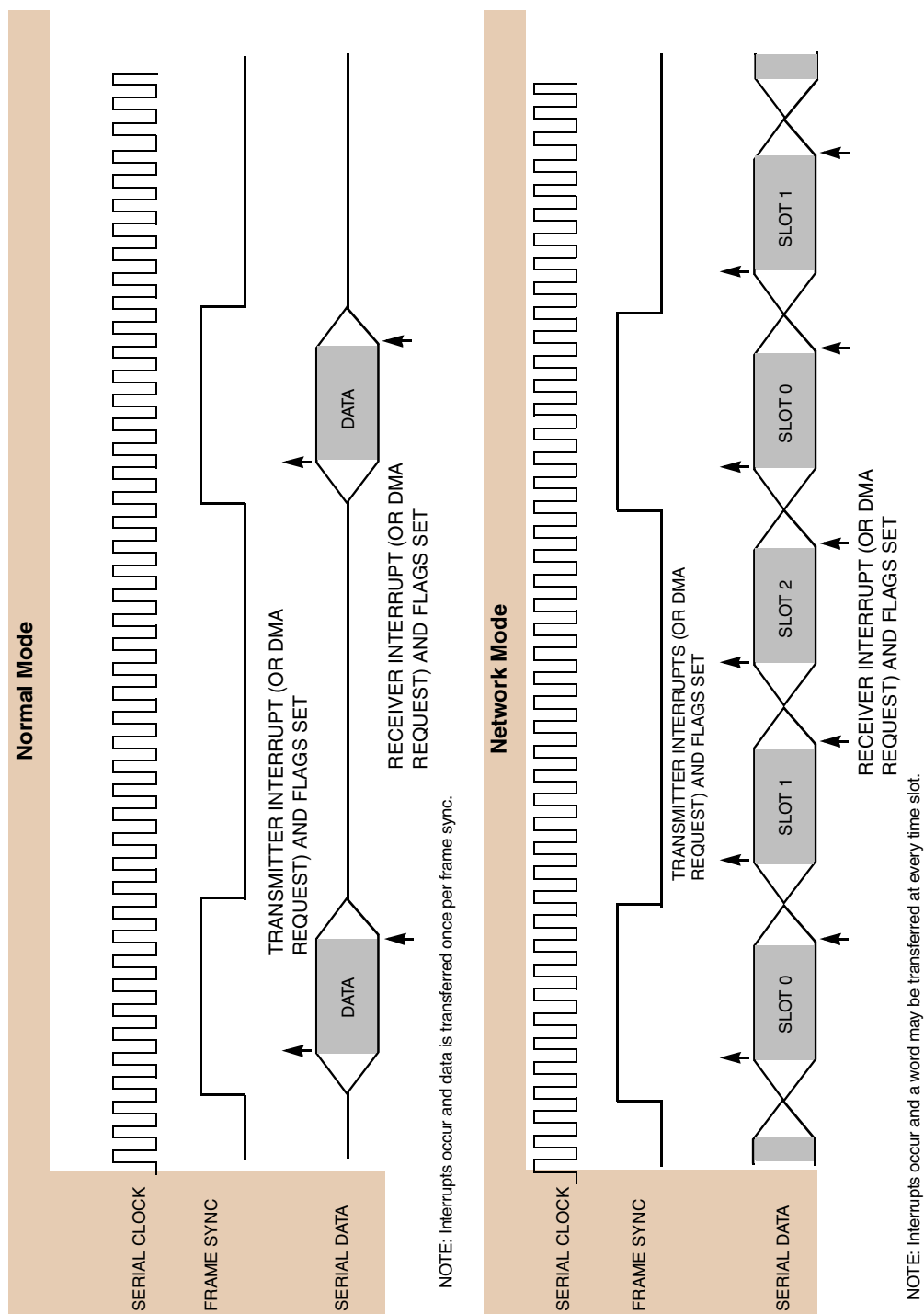


Figure 9-6. Normal and Network Operation

### 9.2.2.10 TCR Tx Slot and Word Length Select (TSWS4–TSWS0)—Bits 14–10

The TSWS4–TSWS0 bits are used to select the length of the slot and the length of the data words being transferred via the ESAI. The word length must be equal to or shorter than the slot length. The possible combinations are shown in [Table 9-5](#). See also the ESAI data path programming model in [Figure 9-13](#) and [Figure 9-14](#).

**Table 9-5. ESAI Transmit Slot and Word Length Selection**

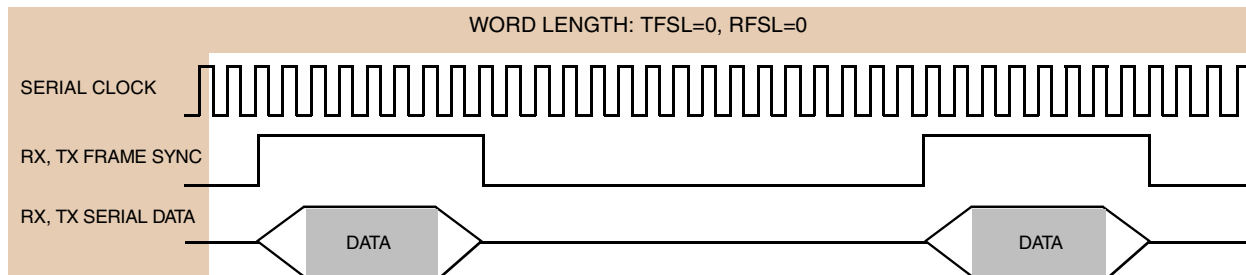
TSWS4	TSWS3	TSWS2	TSWS1	TSWS0	SLOT LENGTH	WORD LENGTH
0	0	0	0	0	8	8
0	0	1	0	0	12	8
0	0	0	0	1		12
0	1	0	0	0	16	8
0	0	1	0	1		12
0	0	0	1	0		16
0	1	1	0	0	20	8
0	1	0	0	1		12
0	0	1	1	0		16
0	0	0	1	1		20
1	0	0	0	0	24	8
0	1	1	0	1		12
0	1	0	1	0		16
0	0	1	1	1		20
1	1	1	1	0		24
1	1	0	0	0	32	8
1	0	1	0	1		12
1	0	0	1	0		16
0	1	1	1	1		20
1	1	1	1	1		24

**Table 9-5. ESAI Transmit Slot and Word Length Selection (Continued)**

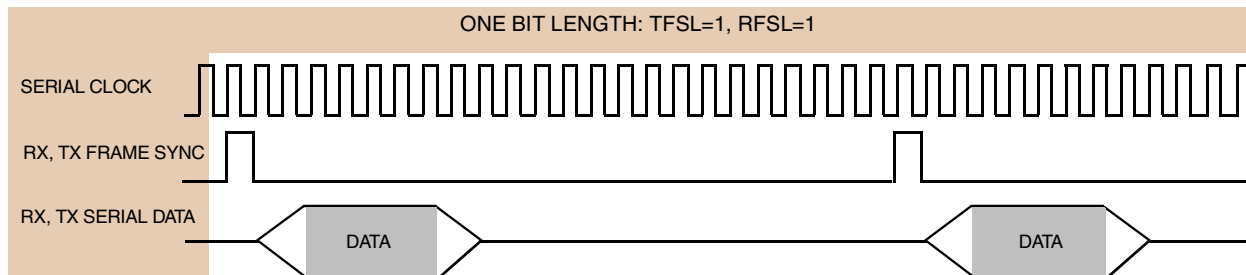
TSWS4	TSWS3	TSWS2	TSWS1	TSWS0	SLOT LENGTH	WORD LENGTH
0	1	0	1	1	Reserved	
0	1	1	1	0		
1	0	0	0	1		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		

### 9.2.2.11 TCR Transmit Frame Sync Length (TFSL)—Bit 15

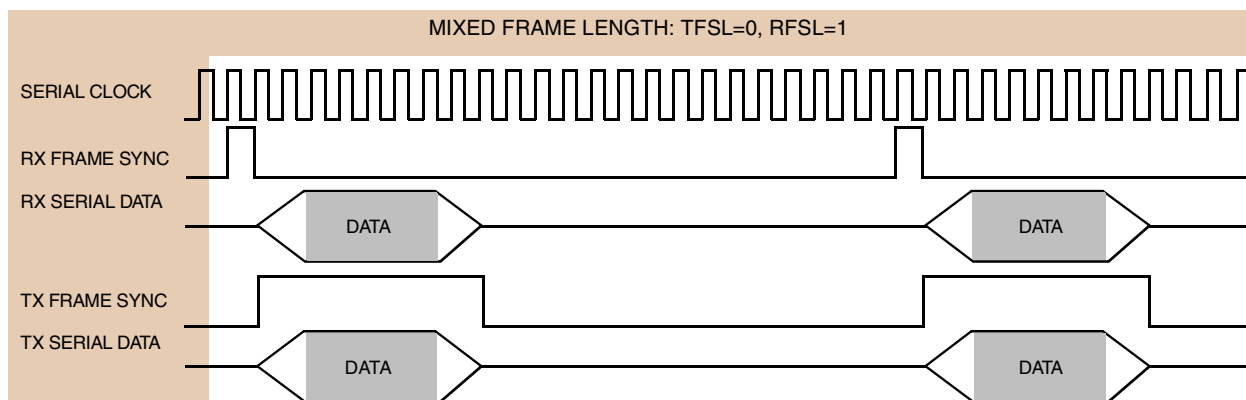
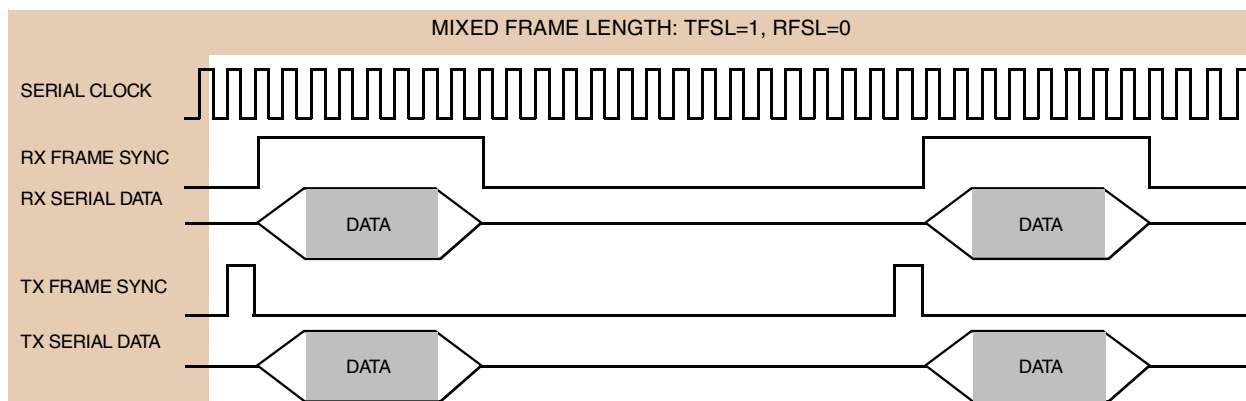
The TFSL bit selects the length of frame sync to be generated or recognized. If TFSL is cleared, a word-length frame sync is selected. If TFSL is set, a 1-bit clock period frame sync is selected. See [Figure 9-7](#) for examples of frame length selection.



**NOTE:** Frame sync occurs while data is valid.



**NOTE:** Frame sync occurs for one bit time preceding the data.



**Figure 9-7. Frame Length Selection**

### 9.2.2.12 TCR Transmit Frame Sync Relative Timing (TFSR)—Bit 16

TFSR determines the relative timing of the transmit frame sync signal as referred to the serial data lines, for a word length frame sync only (TFSL=0). When TFSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When TFSR is set the word length frame sync starts one serial clock cycle earlier, that is, together with the last bit of the previous data word.

### 9.2.2.13 TCR Transmit Zero Padding Control (PADC)—Bit 17

When PADC is cleared, zero padding is disabled. When PADC is set, zero padding is enabled. PADC, in conjunction with the TWA control bit, determines the way that padding is done for operating modes where the word length is less than the slot length. See the TWA bit description in [Section 9.2.2.8, “TCR Transmit Word Alignment Control \(TWA\)—Bit 7,”](#) for more details.

Because the data word is shorter than the slot length, the data word is extended until achieving the slot length, according to the following rule:

1. If the data word is left-aligned (TWA=0), and zero padding is disabled (PADC=0), the last data bit is repeated after the data word has been transmitted. If zero padding is enabled (PADC=1), zeroes are transmitted after the data word has been transmitted.
2. If the data word is right-aligned (TWA=1), and zero padding is disabled (PADC=0), the first data bit is repeated before the transmission of the data word. If zero padding is enabled (PADC=1), zeroes are transmitted before the transmission of the data word.

### 9.2.2.14 TCR Reserved Bit—Bits 18

This bit is reserved. It reads as zero, and it should be written with zero for future compatibility.

### 9.2.2.15 TCR Transmit Section Personal Reset (TPR)—Bit 19

The TPR control bit is used to put the transmitter section of the ESAI in the personal reset state. The receiver section is not affected. When TPR is cleared, the transmitter section may operate normally. When TPR is set, the transmitter section enters the personal reset state immediately. When in the personal reset state, the status bits are reset to the same state as after hardware reset. The control bits are not affected by the personal reset state. The transmitter data pins are tri-stated while in the personal reset state; if a stable logic level is desired, the transmitter data pins should be defined as GPIO outputs, or external pull-up or pull-down resistors should be used. The transmitter clock outputs drive zeroes while in the personal reset state. Note that to leave the personal reset state by clearing TPR, the procedure described in [Section 9.5, “ESAI Initialization Examples,”](#) should be followed.

### 9.2.2.16 TCR Transmit Exception Interrupt Enable (TEIE)—Bit 20

When TEIE is set, the DSP is interrupted when both TDE and TUE in the SAISR status register are set. When TEIE is cleared, this interrupt is disabled. Reading the SAISR status register followed by writing to all the data registers of the enabled transmitters clears TUE, thus clearing the pending interrupt.

### 9.2.2.17 TCR Transmit Even Slot Data Interrupt Enable (TEDIE)—Bit 21

The TEDIE control bit is used to enable the transmit even slot data interrupts. If TEDIE is set, the transmit even slot data interrupts are enabled. If TEDIE is cleared, the transmit even slot data interrupts are disabled. A transmit even slot data interrupt request is generated if TEDIE is set and the TEDE status flag in the SAISR status register is set. Even time slots are all even-numbered time slots (0, 2, 4, etc.) when operating in network mode. The zero time slot in the frame is marked by the frame sync signal and is considered to be even. Writing data to all the data registers of the enabled transmitters or to TSR clears the TEDE flag, thus servicing the interrupt.

Transmit interrupts with exception have higher priority than transmit even slot data interrupts, therefore if exception occurs (TUE is set) and TEIE is set, the ESAI requests an ESAI transmit data with exception interrupt from the interrupt controller.

### 9.2.2.18 TCR Transmit Interrupt Enable (TIE)—Bit 22

The DSP is interrupted when TIE and the TDE flag in the SAISR status register are set. When TIE is cleared, this interrupt is disabled. Writing data to all the data registers of the enabled transmitters or to TSR clears TDE, thus clearing the interrupt.

Transmit interrupts with exception have higher priority than normal transmit data interrupts, therefore if exception occurs (TUE is set) and TEIE is set, the ESAI requests an ESAI transmit data with exception interrupt from the interrupt controller.

### 9.2.2.19 TCR Transmit Last Slot Interrupt Enable (TLIE)—Bit 23

TLIE enables an interrupt at the beginning of last slot of a frame in network mode. When TLIE is set the DSP is interrupted at the start of the last slot in a frame in network mode regardless of the transmit mask register setting. When TLIE is cleared the transmit last slot interrupt is disabled. TLIE is disabled when TDC[4:0]=0x00000 (on-demand mode). The use of the transmit last slot interrupt is described in [Section 9.3.3, “ESAI Interrupt Requests.”](#)

### 9.2.3 ESAI Receive Clock Control Register (RCCR)

The read/write Receive Clock Control Register (RCCR) controls the ESAI receiver clock generator bit and frame sync rates, word length, and number of words per frame for the serial data. The RCCR control bits are described in the following paragraphs. [Figure 9-8](#) shows the ESAI Receive Clock Control register. Hardware and software reset clear all the bits of the RCCR register.

Address X: 0xFFFFB8

Access: Read/Write



**Figure 9-8. RCCR Register**

#### 9.2.3.1 RCCR Receiver Prescale Modulus Select (RPM7–RPM0)—Bits 7–0

The RPM7–RPM0 bits specify the divide ratio of the prescale divider in the ESAI receiver clock generator. A divide ratio from 1 to 256 (RPM[7:0]=0x00 to 0xFF) may be selected. The bit clock output is available at the receiver serial bit clock (SCKR) pin of the DSP. The bit clock output is also available internally for use as the bit clock to shift the receive shift registers. The ESAI receive clock generator functional diagram is shown in [Figure 9-3](#).

#### 9.2.3.2 RCCR Receiver Prescaler Range (RPSR)—Bit 8

The RPSR controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When RPSR is set, the fixed prescaler is bypassed. When RPSR is cleared, the fixed divide-by-eight prescaler is operational (see [Figure 9-3](#)). The maximum internally generated bit clock frequency is  $F_{sys}/4$ , the minimum internally generated bit clock frequency is  $F_{sys}/(2 \times 8 \times 256) = F_{sys}/4096$ .

**NOTE**

Do not use the combination RPSR=1 and RPM7-RPM0=0x00, which causes synchronization problems when using the internal DSP clock as source (RHCKD=1 or RCKD=1).

#### 9.2.3.3 RCCR Rx Frame Rate Divider Control (RDC4–RDC0)—Bits 13–9

The RDC4–RDC0 bits control the divide ratio for the programmable frame rate dividers used to generate the receiver frame clocks.



In network mode, this ratio may be interpreted as the number of words per frame minus one. The divide ratio may range from 2 to 32 (RDC[4:0]=00001 to 11111) for network mode. A divide ratio of one (RDC[4:0]=00000) in network mode is a special case (on-demand mode).

In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 (RDC[4:0]=00000 to 11111) for normal mode. In normal mode, a divide ratio of one (RDC[4:0]=00000) provides continuous periodic data word transfers. A bit-length frame sync (RFSL=1) must be used in this case.

The ESAI frame sync generator functional diagram is shown in [Figure 9-4](#).

### 9.2.3.4 RCCR Rx High Frequency Clock Divider (RFP3-RFP0)—Bits 17–14

The RFP3–RFP0 bits control the divide ratio of the receiver high frequency clock to the receiver serial bit clock when the source of the receiver high frequency clock and the bit clock is the internal DSP clock. When the HCKR input is being driven from an external high frequency clock, the RFP3–RFP0 bits specify an additional division ration in the clock divider chain. [Table 9-6](#) provides the specification of the divide ratio. [Figure 9-3](#) shows the ESAI high frequency generator functional diagram.

**Table 9-6. Receiver High Frequency Clock Divider**

RFP3–RFP0	Divide Ratio
0x0	1
0x1	2
0x2	3
0x3	4
...	...
0xF	16

### 9.2.3.5 RCCR Receiver Clock Polarity (RCKP)—Bit 18

The Receiver Clock Polarity (RCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If RCKP is cleared the data and the frame sync are clocked out on the rising edge of the receive bit clock and the frame sync is latched in on the falling edge of the receive bit clock. If RCKP is set the falling edge of the receive clock is used to clock the data and frame sync out and the rising edge of the receive clock is used to latch the frame sync in.

### 9.2.3.6 RCCR Receiver Frame Sync Polarity (RFSP)—Bit 19

The Receiver Frame Sync Polarity (RFSP) determines the polarity of the receive frame sync signal. When RFSP is cleared the frame sync signal polarity is positive, that is, the frame start is indicated by a high level on the frame sync pin. When RFSP is set the frame sync signal polarity is negative, that is, the frame start is indicated by a low level on the frame sync pin.

### 9.2.3.7 RCCR Receiver High Frequency Clock Polarity (RHCKP)—Bit 20

The Receiver High Frequency Clock Polarity (RHCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If RHCKP is cleared the data and the frame sync are clocked out on the rising edge of the receive high frequency bit clock and the frame sync is latched in on the falling edge of the receive bit clock. If RHCKP is set the falling edge of the receive clock is used to clock the data and frame sync out and the rising edge of the receive clock is used to latch the frame sync in.

### 9.2.3.8 RCCR Receiver Clock Source Direction (RCKD)—Bit 21

The Receiver Clock Source Direction (RCKD) bit selects the source of the clock signal used to clock the receive shift register in the asynchronous mode (SYN=0) and the IF0/OF0 flag direction in the synchronous mode (SYN=1).

In the asynchronous mode, when RCKD is set, the internal clock source becomes the bit clock for the receive shift registers and word length divider and is the output on the SCKR pin. In the asynchronous mode when RCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCKR pin, and an external clock source may drive this pin.

In the synchronous mode when RCKD is set, the SCKR pin becomes the OF0 output flag. If RCKD is cleared, the SCKR pin becomes the IF0 input flag. See [Table 9-1](#) and [Table 9-7](#).

**Table 9-7. SCKR Pin Definition Table**

Control Bits		SCKR PIN
SYN	RCKD	
0	0	SCKR input
0	1	SCKR output
1	0	IF0
1	1	OF0

### 9.2.3.9 RCCR Receiver Frame Sync Signal Direction (RFSD)—Bit 22

The Receiver Frame Sync Signal Direction (RFSD) bit selects the source of the receiver frame sync signal when in the asynchronous mode (SYN=0) and the IF1/OF1/Transmitter Buffer Enable flag direction in the synchronous mode (SYN=1).

In the asynchronous mode, when RFSD is set, the internal clock generator becomes the source of the receiver frame sync and is the output on the FSR pin. In the asynchronous mode, when RFSD is cleared, the receiver frame sync source is external; the internal clock generator is disconnected from the FSR pin, and an external clock source may drive this pin.

In the synchronous mode when RFSD is set, the FSR pin becomes the OF1 output flag or the Transmitter Buffer Enable, according to the TEBE control bit. If RFSD is cleared, the FSR pin becomes the IF1 input flag. See [Table 9-1](#) and [Table 9-8](#).

**Table 9-8. FSR Pin Definition Table**

Control Bits			FSR Pin
SYN	TEBE	RFSD	
0	X	0	FSR input
0	X	1	FSR output
1	0	0	IF1
1	0	1	OF1
1	1	0	reserved
1	1	1	Transmitter Buffer Enable

### 9.2.3.10 RCCR Receiver High Frequency Clock Direction (RHCKD)—Bit 23

The Receiver High Frequency Clock Direction (RHCKD) bit selects the source of the receiver high frequency clock when in the asynchronous mode (SYN=0) and the IF2/OF2 flag direction in the synchronous mode (SYN=1).

In the asynchronous mode, when RHCKD is set, the internal clock generator becomes the source of the receiver high frequency clock and is the output on the HCKR pin. In the asynchronous mode, when RHCKD is cleared, the receiver high frequency clock source is external; the internal clock generator is disconnected from the HCKR pin, and an external clock source may drive this pin.

When RHCKD is cleared, HCKR is an input; when RHCKD is set, HCKR is an output.

In the synchronous mode when RHCKD is set, the HCKR pin becomes the OF2 output flag. If RHCKD is cleared, the HCKR pin becomes the IF2 input flag. See [Table 9-1](#) and [Table 9-9](#).

**Table 9-9. HCKR Pin Definition Table**

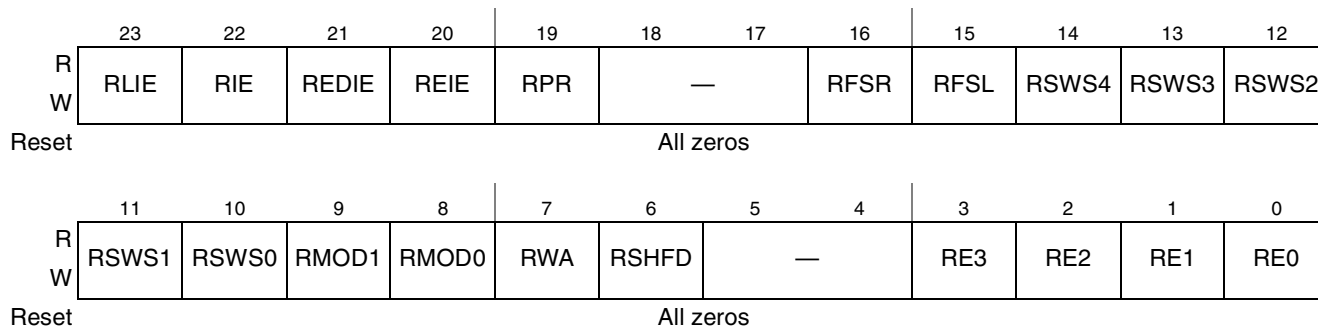
Control Bits		HCKR PIN
SYN	RHCKD	
0	0	HCKR input
0	1	HCKR output
1	0	IF2
1	1	OF2

## 9.2.4 ESAI Receive Control Register (RCR)

The read/write Receive Control Register (RCR) controls the ESAI receiver section. Interrupt enable bits for the receivers are provided in this control register. The receivers are enabled in this register (0,1,2 or 3 receivers can be enabled) if the input data pin is not used by a transmitter. Operating modes are also selected in this register. Figure 9-9 shows the ESAI Receive Control register.

Address X: 0xFFFFB7

Access: Read/Write



**Figure 9-9. ESAI Receive Control Register**

### NOTE

Hardware and software reset clear all the bits in the RCR register.

The ESAI RCR bits are described in the following paragraphs.

#### 9.2.4.1 RCR ESAI Receiver 0 Enable (RE0)—Bit 0

When RE0 is set and TE5 is cleared, the ESAI receiver 0 is enabled and samples data at the SDO5/SDI0 pin. TX5 and RX0 should not be enabled at the same time (RE0=1 and TE5=1). When RE0 is cleared, receiver 0 is disabled by inhibiting data transfer into RX0. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the RX0 data register.

If RE0 is set while some of the other receivers are already in operation, the first data word received in RX0 will be invalid and must be discarded.

#### 9.2.4.2 RCR ESAI Receiver 1 Enable (RE1)—Bit 1

When RE1 is set and TE4 is cleared, the ESAI receiver 1 is enabled and samples data at the SDO4/SDI1 pin. TX4 and RX1 should not be enabled at the same time (RE1=1 and TE4=1). When RE1 is cleared, receiver 1 is disabled by inhibiting data transfer into RX1. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the RX1 data register.

If RE1 is set while some of the other receivers are already in operation, the first data word received in RX1 will be invalid and must be discarded.

#### 9.2.4.3 RCR ESAI Receiver 2 Enable (RE2)—Bit 2

When RE2 is set and TE3 is cleared, the ESAI receiver 2 is enabled and samples data at the SDO3/SDI2 pin. TX3 and RX2 should not be enabled at the same time (RE2=1 and TE3=1). When RE2 is cleared,

receiver 2 is disabled by inhibiting data transfer into RX2. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the RX2 data register.

If RE2 is set while some of the other receivers are already in operation, the first data word received in RX2 will be invalid and must be discarded.

#### 9.2.4.4 RCR ESAI Receiver 3 Enable (RE3)—Bit 3

When RE3 is set and TE2 is cleared, the ESAI receiver 3 is enabled and samples data at the SDO2/SDI3 pin. TX2 and RX3 should not be enabled at the same time (RE3=1 and TE2=1). When RE3 is cleared, receiver 3 is disabled by inhibiting data transfer into RX3. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the RX3 data register.

If RE3 is set while some of the other receivers are already in operation, the first data word received in RX3 will be invalid and must be discarded.

#### 9.2.4.5 RCR Reserved Bits—Bits 5–4, 18–17

These bits are reserved. They read as zero, and they should be written with zero for future compatibility.

#### 9.2.4.6 RCR Receiver Shift Direction (RSHFD)—Bit 6

The RSHFD bit causes the receiver shift registers to shift data in MSB first when RSHFD is cleared or LSB first when RSHFD is set (see [Figure 9-13](#) and [Figure 9-14](#)).

#### 9.2.4.7 RCR Receiver Word Alignment Control (RWA)—Bit 7

The Receiver Word Alignment Control (RWA) bit defines the alignment of the data word in relation to the slot. This is relevant for the cases where the word length is shorter than the slot length. If RWA is cleared, the data word is assumed to be left-aligned in the slot frame. If RWA is set, the data word is assumed to be right-aligned in the slot frame.

If the data word is shorter than the slot length, the data bits which are not in the data word field are ignored.

For data word lengths of less than 24 bits, the data word is right-extended with zeroes before being stored in the receive data registers.

#### 9.2.4.8 RCR Receiver Network Mode Control (RMOD1–RMOD0)—Bits 9–8

The RMOD1 and RMOD0 bits are used to define the network mode of the ESAI receivers, as shown in [Table 9-10](#). In the normal mode, the frame rate divider determines the word transfer rate—one word is transferred per frame sync during the frame sync time slot, as shown in [Figure 9-6](#). In network mode, it is possible to transfer a word for every time slot, as shown in [Figure 9-6](#). For more details, see [Section 9.3](#), “Operating Modes.”

To comply with AC-97 specifications, RSWS4–RSWS0 should be set to 00011 (20-bit slot, 20-bit word); RFSL and RFSR should be cleared, and RDC4–RDC0 should be set to 0x0C (13 words in frame).

**Table 9-10. ESAI Receive Network Mode Selection**

RMOD1	RMOD0	RDC4–RDC0	Receiver Network Mode
0	0	0x0–0x1F	Normal Mode
0	1	0x0	On-Demand Mode
0	1	0x1–0x1F	Network Mode
1	0	X	Reserved
1	1	0x0C	AC97

### 9.2.4.9 RCR Receiver Slot and Word Select (RSWS4–RSWS0)—Bits 14–10

The RSWS4–RSWS0 bits are used to select the length of the slot and the length of the data words being received via the ESAI. The word length must be equal to or shorter than the slot length. The possible combinations are shown in [Table 9-11](#). See also the ESAI data path programming model in [Figure 9-13](#) and [Figure 9-14](#).

**Table 9-11. ESAI Receive Slot and Word Length Selection**

RSWS4	RSWS3	RSWS2	RSWS1	RSWS0	SLOT LENGTH	WORD LENGTH
0	0	0	0	0	8	8
0	0	1	0	0	12	8
0	0	0	0	1		12
0	1	0	0	0	16	8
0	0	1	0	1		12
0	0	0	1	0		16
0	1	1	0	0	20	8
0	1	0	0	1		12
0	0	1	1	0		16
0	0	0	1	1		20
1	0	0	0	0	24	8
0	1	1	0	1		12
0	1	0	1	0		16
0	0	1	1	1		20
1	1	1	1	0		24
1	1	0	0	0	32	8
1	0	1	0	1		12
1	0	0	1	0		16
0	1	1	1	1		20
1	1	1	1	1		24

**Table 9-11. ESAI Receive Slot and Word Length Selection (Continued)**

RSWS4	RSWS3	RSWS2	RSWS1	RSWS0	SLOT LENGTH	WORD LENGTH
0	1	0	1	1	Reserved	
0	1	1	1	0		
1	0	0	0	1		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		

#### 9.2.4.10 RCR Receiver Frame Sync Length (RFSL)—Bit 15

The RFSL bit selects the length of the receive frame sync to be generated or recognized. If RFSL is cleared, a word-length frame sync is selected. If RFSL is set, a 1-bit clock period frame sync is selected. See [Figure 9-7](#) for examples of frame length selection.

#### 9.2.4.11 RCR Receiver Frame Sync Relative Timing (RFSR)—Bit 16

RFSR determines the relative timing of the receive frame sync signal as referred to the serial data lines, for a word length frame sync only. When RFSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When RFSR is set the word length frame sync starts one serial clock cycle earlier, that is, together with the last bit of the previous data word.

#### 9.2.4.12 RCR Receiver Section Personal Reset (RPR)—Bit 19

The RPR control bit is used to put the receiver section of the ESAI in the personal reset state. The transmitter section is not affected. When RPR is cleared, the receiver section may operate normally. When RPR is set, the receiver section enters the personal reset state immediately. When in the personal reset state, the status bits are reset to the same state as after hardware reset. The control bits are not affected by the personal reset state. The receiver data pins are disconnected while in the personal reset state. Note that to leave the personal reset state by clearing RPR, the procedure described in [Section 9.5, “ESAI Initialization Examples,”](#) should be followed.

#### 9.2.4.13 RCR Receive Exception Interrupt Enable (REIE)—Bit 20

When REIE is set, the DSP is interrupted when both RDF and ROE in the SAISR status register are set. When REIE is cleared, this interrupt is disabled. Reading the SAISR status register followed by reading the enabled receivers data registers clears ROE, thus clearing the pending interrupt.

#### 9.2.4.14 RCR Receive Even Slot Data Interrupt Enable (REDIE)—Bit 21

The REDIE control bit is used to enable the receive even slot data interrupts. If REDIE is set, the receive even slot data interrupts are enabled. If REDIE is cleared, the receive even slot data interrupts are disabled. A receive even slot data interrupt request is generated if REDIE is set and the REDF status flag in the SAISR status register is set. Even time slots are all even-numbered time slots (0, 2, 4, etc.) when operating in network mode. The zero time slot is marked by the frame sync signal and is considered to be even. Reading all the data registers of the enabled receivers clears the REDF flag, thus servicing the interrupt.

Receive interrupts with exception have higher priority than receive even slot data interrupts, therefore if exception occurs (ROE is set) and REIE is set, the ESAI requests an ESAI receive data with exception interrupt from the interrupt controller.

#### 9.2.4.15 RCR Receive Interrupt Enable (RIE)—Bit 22

The DSP is interrupted when RIE and the RDF flag in the SAISR status register are set. When RIE is cleared, this interrupt is disabled. Reading the receive data registers of the enabled receivers clears RDF, thus clearing the interrupt.

Receive interrupts with exception have higher priority than normal receive data interrupts, therefore if exception occurs (ROE is set) and REIE is set, the ESAI requests an ESAI receive data with exception interrupt from the interrupt controller.

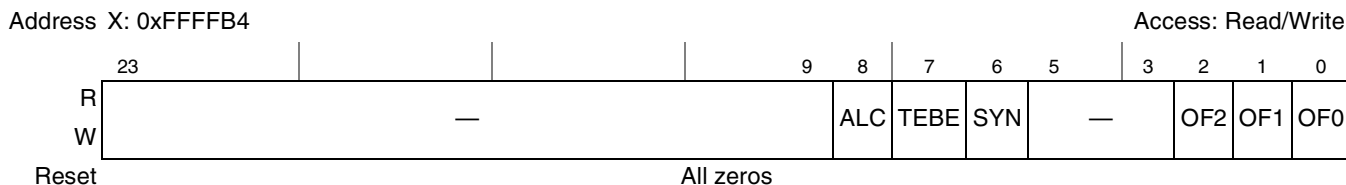
#### 9.2.4.16 RCR Receive Last Slot Interrupt Enable (RLIE)—Bit 23

RLIE enables an interrupt after the last slot of a frame ended in network mode only. When RLIE is set the DSP is interrupted after the last slot in a frame ended regardless of the receive mask register setting. When RLIE is cleared the receive last slot interrupt is disabled. Hardware and software reset clear RLIE. RLIE is disabled when RDC[4:0]=00000 (on-demand mode). The use of the receive last slot interrupt is described in [Section 9.3.3, “ESAI Interrupt Requests.”](#)



## 9.2.5 ESAI Common Control Register (SAICR)

The read/write Common Control Register (SAICR) contains control bits for functions that affect both the receive and transmit sections of the ESAI. [Figure 9-10](#) shows the ESAI Common Control register.



**Figure 9-10. SAICR Register**

Hardware and software reset clear all the bits in the SAICR register.

### 9.2.5.1 SAICR Serial Output Flag 0 (OF0)—Bit 0

The Serial Output Flag 0 (OF0) is a data bit used to hold data to be send to the OF0 pin. When the ESAI is in the synchronous clock mode (SYN=1), the SCKR pin is configured as the ESAI flag 0. If the receiver serial clock direction bit (RCKD) is set, the SCKR pin is the output flag OF0, and data present in the OF0 bit is written to the OF0 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

### 9.2.5.2 SAICR Serial Output Flag 1 (OF1)—Bit 1

The Serial Output Flag 1 (OF1) is a data bit used to hold data to be send to the OF1 pin. When the ESAI is in the synchronous clock mode (SYN=1), the FSR pin is configured as the ESAI flag 1. If the receiver frame sync direction bit (RFSK) is set and the TEBE bit is cleared, the FSR pin is the output flag OF1, and data present in the OF1 bit is written to the OF1 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

### 9.2.5.3 SAICR Serial Output Flag 2 (OF2)—Bit 2

The Serial Output Flag 2 (OF2) is a data bit used to hold data to be send to the OF2 pin. When the ESAI is in the synchronous clock mode (SYN=1), the HCKR pin is configured as the ESAI flag 2. If the receiver high frequency clock direction bit (RHCKD) is set, the HCKR pin is the output flag OF2, and data present in the OF2 bit is written to the OF2 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

### 9.2.5.4 SAICR Reserved Bits—Bits 5–3, 23–9

These bits are reserved. They read as zero, and they should be written with zero for future compatibility.

### 9.2.5.5 SAICR Synchronous Mode Selection (SYN)—Bit 6

The Synchronous Mode Selection (SYN) bit controls whether the receiver and transmitter sections of the ESAI operate synchronously or asynchronously with respect to each other (see [Table 9-11](#)). When SYN is cleared, the asynchronous mode is chosen and independent clock and frame sync signals are used for the

transmit and receive sections. When SYN is set, the synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals.

When in the synchronous mode (SYN=1), the transmit and receive sections use the transmitter section clock generator as the source of the clock and frame sync for both sections. Also, the receiver clock pins SCKR, FSR and HCKR now operate as I/O flags. See [Table 9-7](#) for the effects of SYN on the receiver clock pins.

#### 9.2.5.6 SAICR Transmit External Buffer Enable (TEBE)—Bit 7

The Transmitter External Buffer Enable (TEBE) bit controls the function of the FSR pin when in the synchronous mode. If the ESAI is configured for operation in the synchronous mode (SYN=1), and TEBE is set while FSR pin is configured as an output (RFSD=1), the FSR pin functions as the transmitter external buffer enable control to enable the use of an external buffers on the transmitter outputs. If TEBE is cleared, the FSR pin functions as the serial I/O flag 1. See [Table 9-7](#) for a summary of the effects of TEBE on the FSR pin.

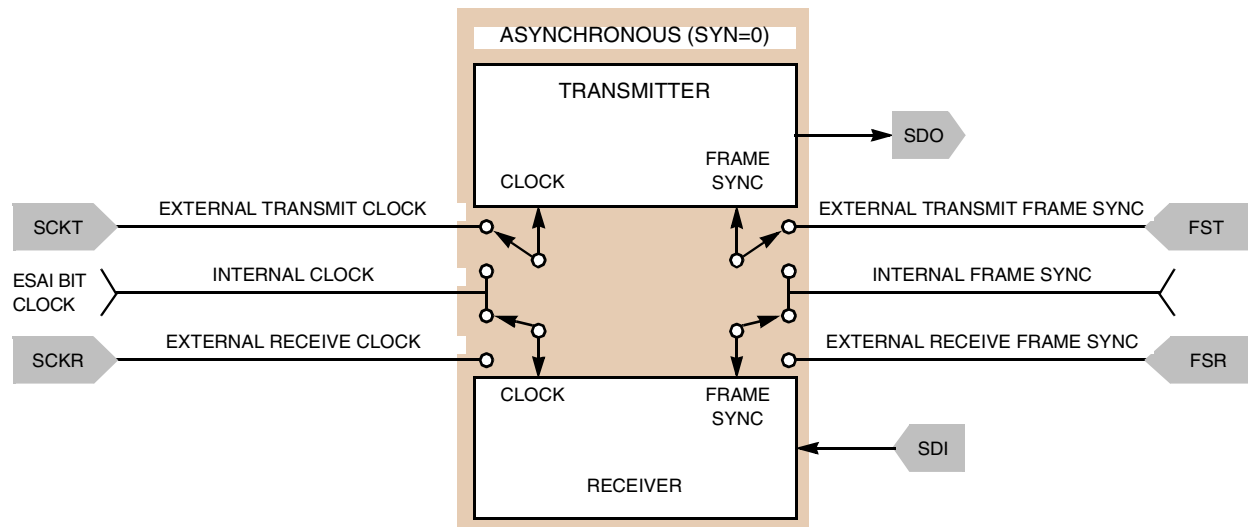
#### 9.2.5.7 SAICR Alignment Control (ALC)—Bit 8

The ESAI is designed for 24-bit fractional data, thus shorter data words are left aligned to the MSB (bit 23). Some applications use 16-bit fractional data. In those cases, shorter data words may be left aligned to bit 15. The Alignment Control (ALC) bit supports these applications.

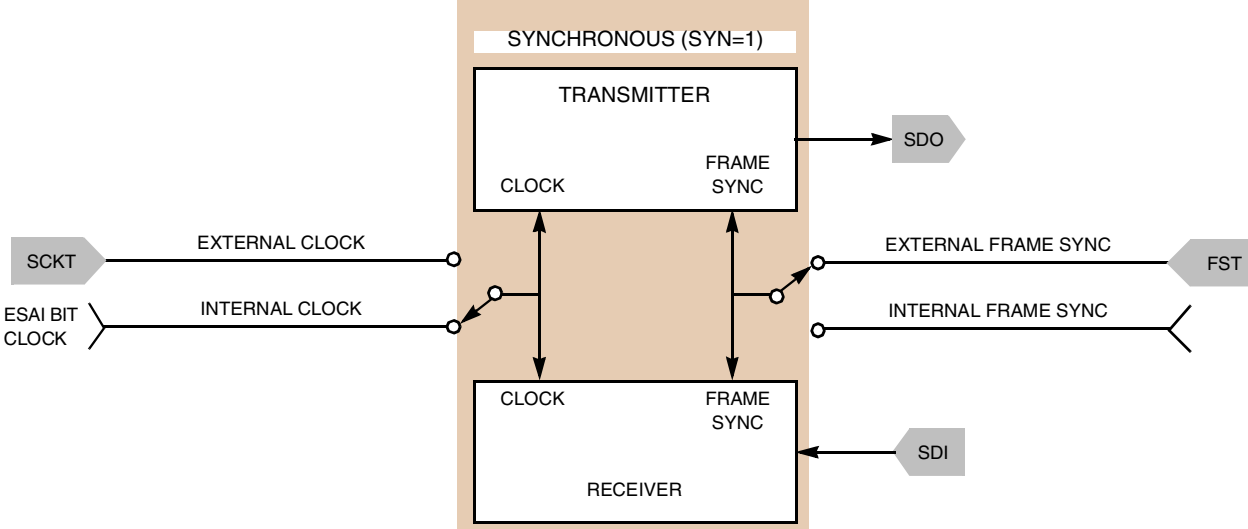
If ALC is set, transmitted and received words are left aligned to bit 15 in the transmit and receive shift registers. If ALC is cleared, transmitted and received word are left aligned to bit 23 in the transmit and receive shift registers.

#### NOTE

While ALC is set, 20-bit and 24-bit words may not be used, and word length control should specify 8-, 12-, or 16-bit words; otherwise, results are unpredictable.



**NOTE:** Transmitter and receiver may have different clocks and frame syncs.



**NOTE :** Transmitter and receiver have the same clocks and frame syncs.

**Figure 9-11. SAICR SYN Bit Operation**

## 9.2.6 ESAI Status Register (SAISR)

The Status Register (SAISR) is a read-only status register used by the DSP to read the status and serial input flags of the ESAI. The status bits are described in the following paragraphs. Figure 9-12 shows the ESAI Status register.

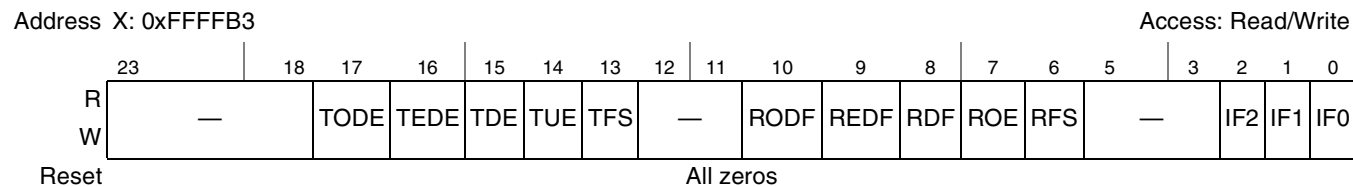


Figure 9-12. SAISR Register

### 9.2.6.1 SAISR Serial Input Flag 0 (IF0)—Bit 0

The IF0 bit is enabled only when the SCKR pin is defined as ESAI in the Port Control Register, SYN=1 and RCKD=0, indicating that SCKR is an input flag and the synchronous mode is selected. Data present on the SCKR pin is latched during reception of the first received data bit after frame sync is detected. The IF0 bit is updated with this data when the receiver shift registers are transferred into the receiver data registers. IF0 reads as a zero when it is not enabled. Hardware, software, ESAI individual and STOP reset clear IF0.

### 9.2.6.2 SAISR Serial Input Flag 1 (IF1)—Bit 1

The IF1 bit is enabled only when the FSR pin is defined as ESAI in the Port Control Register, SYN =1, RFSD=0 and TEBE=0, indicating that FSR is an input flag and the synchronous mode is selected. Data present on the FSR pin is latched during reception of the first received data bit after frame sync is detected. The IF1 bit is updated with this data when the receiver shift registers are transferred into the receiver data registers. IF1 reads as a zero when it is not enabled. Hardware, software, ESAI individual and STOP reset clear IF1.

### 9.2.6.3 SAISR Serial Input Flag 2 (IF2)—Bit 2

The IF2 bit is enabled only when the HCKR pin is defined as ESAI in the Port Control Register, SYN=1 and RHCKD=0, indicating that HCKR is an input flag and the synchronous mode is selected. Data present on the HCKR pin is latched during reception of the first received data bit after frame sync is detected. The IF2 bit is updated with this data when the receive shift registers are transferred into the receiver data registers. IF2 reads as a zero when it is not enabled. Hardware, software, ESAI individual and STOP reset clear IF2.

### 9.2.6.4 SAISR Reserved Bits—Bits 5–3, 12–11, 23–18

These bits are reserved for future use. They read as zero.

### 9.2.6.5 SAISR Receive Frame Sync Flag (RFS)—Bit 6

When set, RFS indicates that a receive frame sync occurred during reception of the words in the receiver data registers. This indicates that the data words are from the first slot in the frame. When RFS is clear and a word is received, it indicates (only in the network mode) that the frame sync did not occur during reception of that word. RFS is cleared by hardware, software, ESAI individual, or STOP reset. RFS is valid only if at least one of the receivers is enabled (REx=1).

#### NOTE

In normal mode, RFS always reads as a one when reading data because there is only one time slot per frame—the “frame sync” time slot.

### 9.2.6.6 SAISR Receiver Overrun Error Flag (ROE)—Bit 7

The ROE flag is set when the serial receive shift register of an enabled receiver is full and ready to transfer to its receiver data register (RXx) and the register is already full (RDF=1). If REIE is set, an ESAI receive data with exception (overrun error) interrupt request is issued when ROE is set. Hardware, software, ESAI individual and STOP reset clear ROE. ROE is also cleared by reading the SAISR with ROE set, followed by reading all the enabled receive data registers.

### 9.2.6.7 SAISR Receive Data Register Full (RDF)—Bit 8

RDF is set when the contents of the receive shift register of an enabled receiver is transferred to the respective receive data register. RDF is cleared when the DSP reads the receive data register of all enabled receivers or cleared by hardware, software, ESAI individual, or STOP reset. If RIE is set, an ESAI receive data interrupt request is issued when RDF is set.

### 9.2.6.8 SAISR Receive Even-Data Register Full (REDF)—Bit 9

When set, REDF indicates that the received data in the receive data registers of the enabled receivers have arrived during an even time slot when operating in the network mode. Even time slots are all even-numbered slots (0, 2, 4, 6, and so on). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. The zero time slot is considered even. REDF is set when the contents of the receive shift registers are transferred to the receive data registers. REDF is cleared when the DSP reads all the enabled receive data registers or cleared by hardware, software, ESAI individual, or STOP resets. If REDIE is set, an ESAI receive even slot data interrupt request is issued when REDF is set.

### 9.2.6.9 SAISR Receive Odd-Data Register Full (RODF)—Bit 10

When set, RODF indicates that the received data in the receive data registers of the enabled receivers have arrived during an odd time slot when operating in the network mode. Odd time slots are all odd-numbered slots (1, 3, 5, and so on). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. RODF is set when the contents of the receive shift registers are transferred to the receive data registers. RODF is cleared when the DSP reads all the enabled receive data registers or cleared by hardware, software, ESAI individual, or STOP resets.

### 9.2.6.10 SAISR Transmit Frame Sync Flag (TFS)—Bit 13

When set, TFS indicates that a transmit frame sync occurred in the current time slot. TFS is set at the start of the first time slot in the frame and cleared during all other time slots. Data written to a transmit data register during the time slot when TFS is set is transmitted (in network mode), if the transmitter is enabled, during the second time slot in the frame. TFS is useful in network mode to identify the start of a frame. TFS is cleared by hardware, software, ESAI individual, or STOP reset. TFS is valid only if at least one transmitter is enabled, that is, one or more of TE0, TE1, TE2, TE3, TE4 and TE5 are set.

#### NOTE

In normal mode, TFS always reads as a one when transmitting data because there is only one time slot per frame—the “frame sync” time slot.

### 9.2.6.11 SAISR Transmit Underrun Error Flag (TUE)—Bit 14

TUE is set when at least one of the enabled serial transmit shift registers is empty (no new data to be transmitted) and a transmit time slot occurs. When a transmit underrun error occurs, the previous data (which is still present in the TX registers that were not written) is retransmitted. If TEIE is set, an ESAI transmit data with exception (underrun error) interrupt request is issued when TUE is set. Hardware, software, ESAI individual and STOP reset clear TUE. TUE is also cleared by reading the SAISR with TUE set, followed by writing to all the enabled transmit data registers or to TSR.

### 9.2.6.12 SAISR Transmit Data Register Empty (TDE)—Bit 15

TDE is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TDE indicates that data should be written to all the TX registers of the enabled transmitters or to the time slot register (TSR). TDE is cleared when the DSP writes to all the transmit data registers of the enabled transmitters, or when the DSP writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TDE is set. Hardware, software, ESAI individual and STOP reset clear TDE.

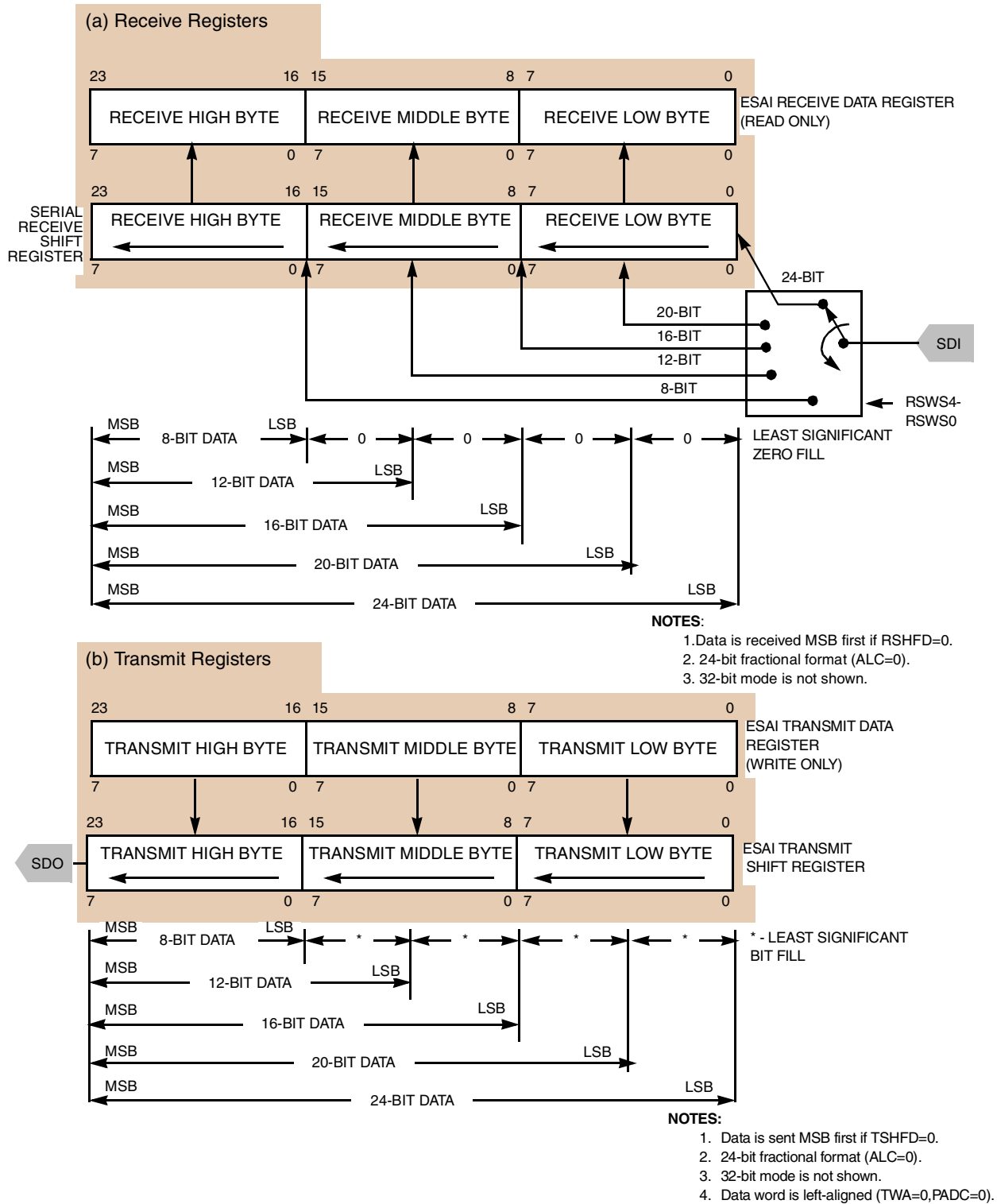
### 9.2.6.13 SAISR Transmit Even-Data Register Empty (TEDE)—Bit 16

When set, TEDE indicates that the enabled transmitter data registers became empty at the beginning of an even time slot. Even time slots are all even-numbered slots (0, 2, 4, 6, etc.). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. The zero time slot is considered even. This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TEDE indicates that data should be written to all the TX registers of the enabled transmitters or to the time slot register (TSR). TEDE is cleared when the DSP writes to all the transmit data registers of the enabled transmitters, or when the DSP writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TEDE is set. Hardware, software, ESAI individual and STOP reset clear TEDE.

#### 9.2.6.14 SAISR Transmit Odd-Data Register Empty (TODE)—Bit 17

When set, TODE indicates that the enabled transmitter data registers became empty at the beginning of an odd time slot. Odd time slots are all odd-numbered slots (1, 3, 5, and so on). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TODE indicates that data should be written to all the TX registers of the enabled transmitters or to the time slot register (TSR). TODE is cleared when the DSP writes to all the transmit data registers of the enabled transmitters, or when the DSP writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TODE is set. Hardware, software, ESAI individual and STOP reset clear TODE.

## Enhanced Serial Audio Interface (ESAI, ESAI\_1, ESAI\_2, ESAI\_3)



**Figure 9-13. ESAI Data Path Programming Model ([R/T]SHFD=0)**



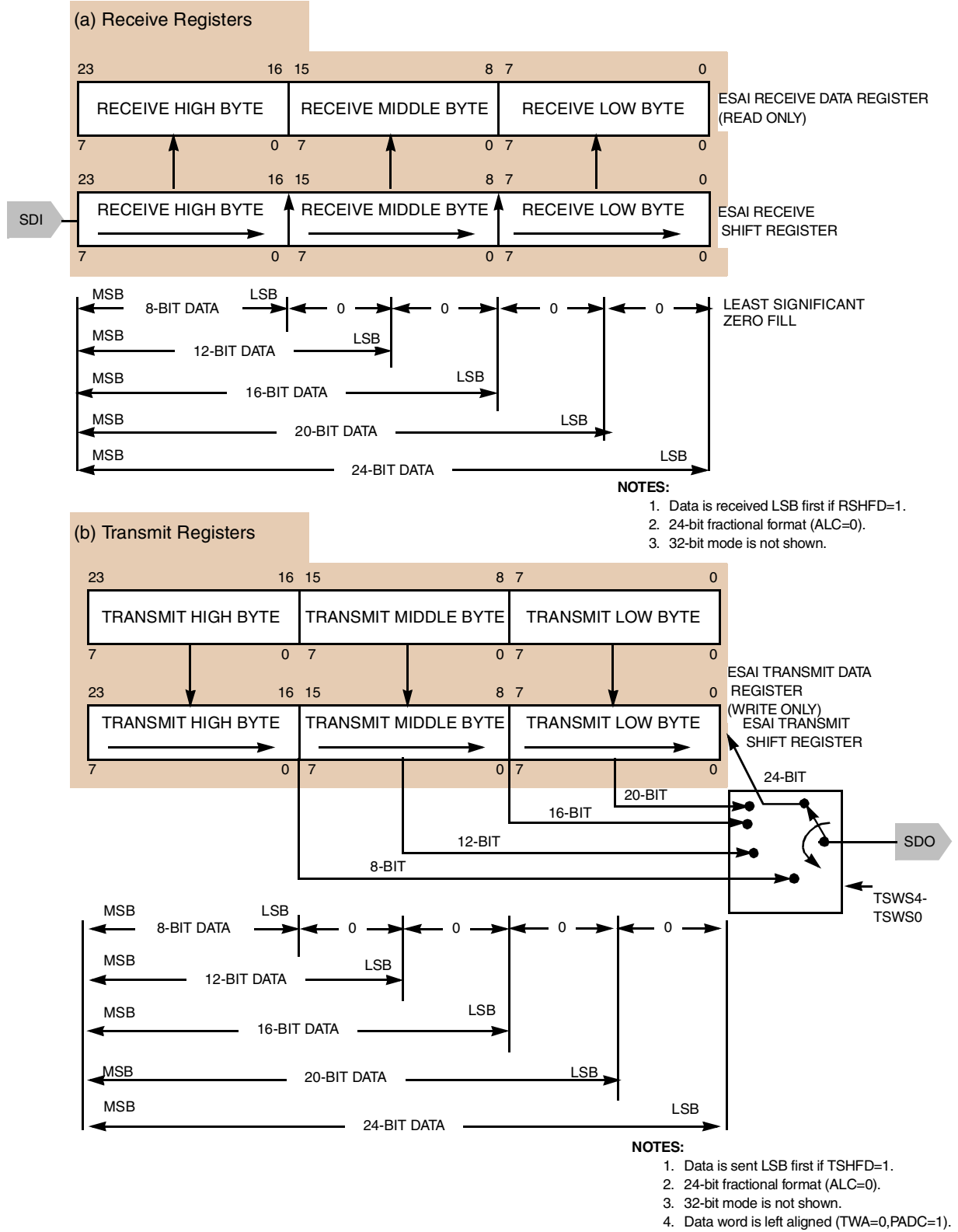


Figure 9-14. ESAI Data Path Programming Model ([R/T]SHFD=1)

## 9.2.7 ESAI Receive Shift Registers

The receive shift registers (Figure 9-13 and Figure 9-14) receive the incoming data from the serial receive data pins. Data is shifted in by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. Data is assumed to be received MSB first if RSHFD=0 and LSB first if RSHFD=1. Data is transferred to the ESAI receive data registers after 8, 12, 16, 20, 24, or 32 serial clock cycles were counted, depending on the slot length control bits in the RCR register.

## 9.2.8 ESAI Receive Data Registers (RX3, RX2, RX1, RX0)

RX3, RX2, RX1, and RX0 are 24-bit read-only registers that accept data from the receive shift registers when they become full (Figure 9-13 and Figure 9-14). The data occupies the most significant portion of the receive data registers, according to the ALC control bit setting. The unused bits (least significant portion and 8 most significant bits when ALC=1) read as zeros. The DSP is interrupted whenever RXx becomes full if the associated interrupt is enabled.

## 9.2.9 ESAI Transmit Shift Registers

The transmit shift registers contain the data being transmitted (Figure 9-13 and Figure 9-14). Data is shifted out to the serial transmit data pins by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. The number of bits shifted out before the shift registers are considered empty and may be written to again can be 8, 12, 16, 20, 24 or 32 bits (determined by the slot length control bits in the TCR register). Data is shifted out of these registers MSB first if TSHFD=0 and LSB first if TSHFD=1.

## 9.2.10 ESAI Transmit Data Registers (TX5, TX4, TX3, TX2, TX1, TX0)

TX5, TX4, TX3, TX2, TX1 and TX0 are 24-bit write-only registers. Data to be transmitted is written into these registers and is automatically transferred to the transmit shift registers (Figure 9-13 and Figure 9-14). The data written (8, 12, 16, 20, or 24 bits) should occupy the most significant portion of the TXx according to the ALC control bit setting. The unused bits (least significant portion and the 8 most significant bits when ALC=1) of the TXx are don't care bits. The DSP is interrupted whenever the TXx becomes empty if the transmit data register empty interrupt has been enabled.

## 9.2.11 ESAI Time Slot Register (TSR)

The write-only Time Slot Register (TSR) is effectively a null data register that is used when the data is not to be transmitted in the available transmit time slot. The transmit data pins of all the enabled transmitters are in the high-impedance state for the respective time slot where TSR has been written. The Transmitter External Buffer Enable pin (FSR pin when SYN=1, TEBE=1, RFSD=1) disables the external buffers during the slot when the TSR register has been written.

## 9.2.12 Transmit Slot Mask Registers (TSMA, TSMB)

The Transmit Slot Mask Registers (TSMA and TSMB) are two read/write registers used by the transmitters in network mode to determine for each slot whether to transmit a data word and generate a

transmitter empty condition (TDE=1), or to tri-state the transmitter data pins. TSMA and TSMB should each be considered as containing half a 32-bit register TSM. Bit number N in TSM (TS\*\*) is the enable/disable control bit for transmission in slot number N. Figure 9-15 and Figure 9-16 show the Transmit Slot Mask registers.

Address X: 0xFFFFB9

Access: Read/Write

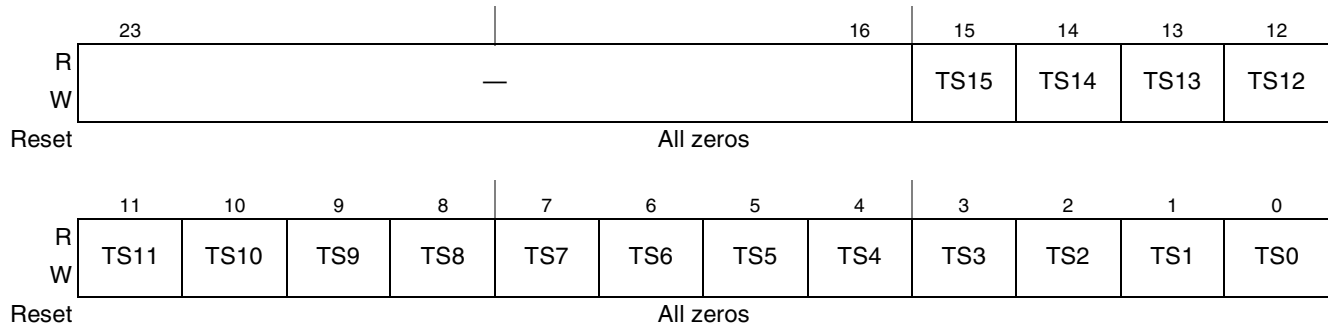


Figure 9-15. TSMA Register

Address X: 0xFFFFBA

Access: Read/Write

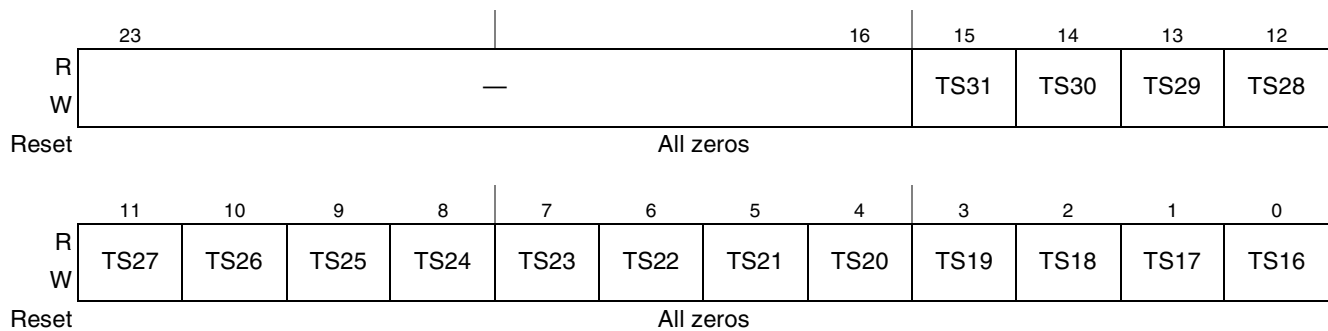


Figure 9-16. TSMB Register

When bit number N in TSM is cleared, all the transmit data pins of the enabled transmitters are tri-stated during transmit time slot number N. The data is still transferred from the transmit data registers to the transmit shift registers but neither the TDE nor the TUE flags are set. This means that during a disabled slot, no transmitter empty interrupt is generated. The DSP is interrupted only for enabled slots. Data that is written to the transmit data registers when servicing this request is transmitted in the next enabled transmit time slot.

When bit number N in TSM register is set, the transmit sequence is as usual: data is transferred from the TX registers to the shift registers and transmitted during slot number N, and the TDE flag is set.

Using the slot mask in TSM does not conflict with using TSR. Even if a slot is enabled in TSM, the user may choose to write to TSR instead of writing to the transmit data registers TXx. This causes all the transmit data pins of the enabled transmitters to be tri-stated during the next slot.

Data written to the TSM affects the next frame transmission. The frame being transmitted is not affected by this data and would comply to the last TSM setting. Data read from TSM returns the last written data.

After hardware or software reset, the TSM register is preset to 0xFFFFFFFF, which means that all 32 possible slots are enabled for data transmission.

**NOTE**

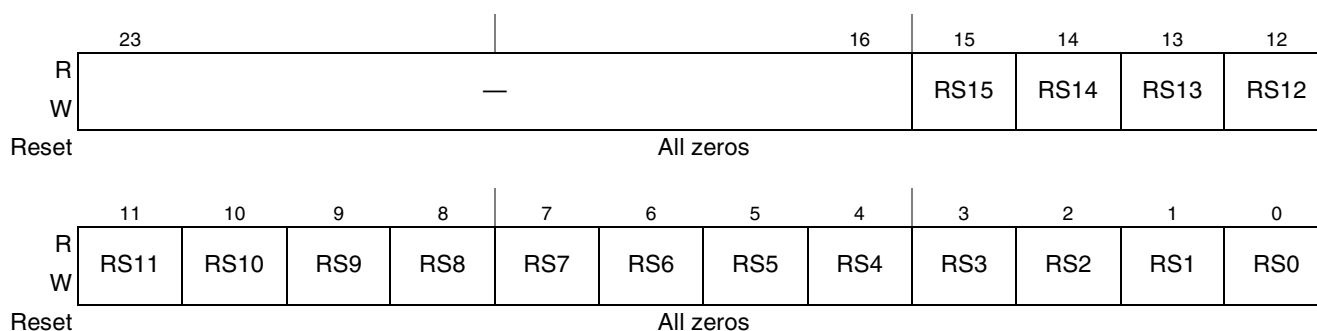
When operating in normal mode, bit 0 of the mask register must be set, otherwise no output is generated.

**9.2.13 Receive Slot Mask Registers (RSMA, RSMB)**

The Receive Slot Mask Registers (RSMA and RSMB) are two read/write registers used by the receiver in network mode to determine for each slot whether to receive a data word and generate a receiver full condition (RDF=1), or to ignore the received data. RSMA and RSMB should be considered as each containing half of a 32-bit register RSM. See Bit number N in RSM (RS\*\*) is an enable/disable control bit for receiving data in slot number N. Figure 9-17 and Figure 9-18 show the Receive Slot Mask registers.

Address X: 0xFFFFBB

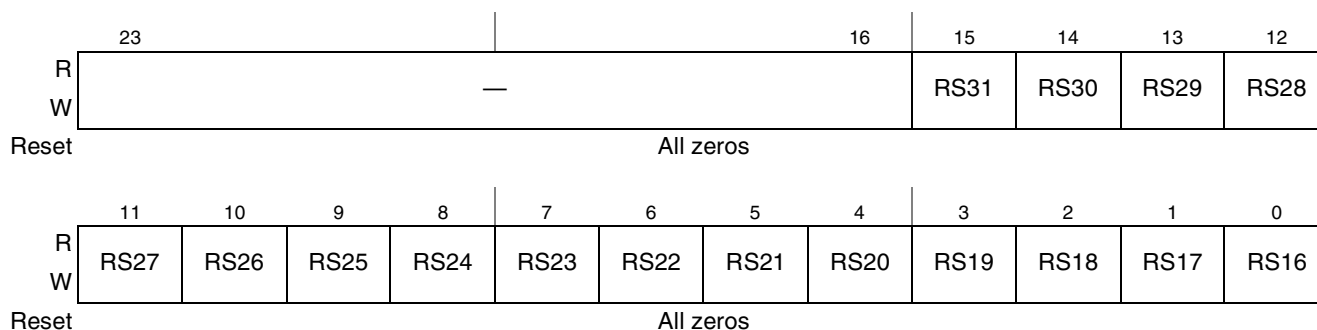
Access: Read/Write



**Figure 9-17. RSMA Register**

Address X: 0xFFFFBC

Access: Read/Write



**Figure 9-18. RSMB Register**

When bit number N in the RSM register is cleared, the data from the enabled receivers input pins are shifted into their receive shift registers during slot number N. The data is not transferred from the receive shift registers to the receive data registers, and neither the RDF nor the ROE flag is set. This means that during a disabled slot, no receiver full interrupt is generated. The DSP is interrupted only for enabled slots.

When bit number N in the RSM is set, the receive sequence is as usual: data which is shifted into the enabled receivers shift registers is transferred to the receive data registers and the RDF flag is set.

Data written to the RSM affects the next received frame. The frame being received is not affected by this data and would comply to the last RSM setting. Data read from RSM returns the last written data.

After hardware or software reset, the RSM register is preset to 0xFFFFFFFF, which means that all 32 possible slots are enabled for data reception.

#### NOTE

When operating in normal mode, bit 0 of the mask register must be set to one, otherwise no input is received.

## 9.3 Operating Modes

ESAI operating mode are selected by the ESAI control registers (TCCR, TCR, RCCR, RCR, and SAICR). The main operating mode are described in the following paragraphs.

### 9.3.1 ESAI After Reset

Hardware or software reset clears the port control register bits and the port direction control register bits, which configure all ESAI I/O pins as disconnected. The ESAI is in the individual reset state while all ESAI pins are programmed as GPIO or disconnected, and it is active only if at least one of the ESAI I/O pins is programmed as an ESAI pin.

### 9.3.2 ESAI Initialization

The correct way to initialize the ESAI is as follows:

1. Hardware, software, ESAI individual, or STOP reset.
2. Program ESAI control and time slot registers.
3. Write data to all the enabled transmitters.
4. Configure at least one pin as ESAI pin.

During program execution, all ESAI pins may be defined as GPIO or disconnected, causing the ESAI to stop serial activity and enter the individual reset state. All status bits of the interface are set to their reset state; however, the control bits are not affected. This procedure allows the DSP programmer to reset the ESAI separately from the other internal peripherals. During individual reset, internal DMA accesses to the data registers of the ESAI are not valid and data read is undefined.

The DSP programmer must use an individual ESAI reset when changing the ESAI control registers (except for TEIE, REIE, TLIE, RLIE, TIE, RIE, TE0-TE5, RE0-RE3) to ensure proper operation of the interface.

## NOTE

If the ESAI receiver section is already operating with some of the receivers, enabling additional receivers on the fly, that is, without first putting the ESAI receiver in the personal reset state, by setting their REx control bits will result in erroneous data being received as the first data word for the newly enabled receivers.

### 9.3.3 ESAI Interrupt Requests

The ESAI can generate eight different interrupt requests (ordered from the highest to the lowest priority):

1. **ESAI Receive Data with Exception Status:**

Occurs when the receive exception interrupt is enabled (REIE=1 in the RCR register), at least one of the enabled receive data registers is full (RDF=1) and a receiver overrun error has occurred (ROE=1 in the SAISR register). ROE is cleared by first reading the SAISR and then reading all the enabled receive data registers.

2. **ESAI Receive Even Data:**

Occurs when the receive even slot data interrupt is enabled (REDIE=1), at least one of the enabled receive data registers is full (RDF=1), the data is from an even slot (REDF=1) and no exception has occurred (ROE=0 or REIE=0).

Reading all enabled receiver data registers clears RDF and REDF.

3. **ESAI Receive Data:**

Occurs when the receive interrupt is enabled (RIE=1), at least one of the enabled receive data registers is full (RDF=1), no exception has occurred (ROE=0 or REIE=0) and no even slot interrupt has occurred (REDF=0 or REDIE=0).

Reading all enabled receiver data registers clears RDF.

4. **ESAI Receive Last Slot Interrupt:**

Occurs, if enabled (RLIE=1), after the last slot of the frame has ended (in network mode only) regardless of the receive mask register setting. The receive last slot interrupt may be used for resetting the receive mask slot register, reconfiguring the DMA channels and reassigning data memory pointers. Using the receive last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame is serviced with the new setting without synchronization problems. Note that the maximum receive last slot interrupt service time should not exceed N-1 ESAI bits service time (where N is the number of bits in a slot).

5. **ESAI Transmit Data with Exception Status:**

Occurs when the transmit exception interrupt is enabled (TEIE=1), at least one transmit data register of the enabled transmitters is empty (TDE=1) and a transmitter underrun error has occurred (TUE=1). TUE is cleared by first reading the SAISR and then writing to all the enabled transmit data registers, or to the TSR register.

6. **ESAI Transmit Last Slot Interrupt:**

Occurs, if enabled (TLIE=1), at the start of the last slot of the frame in network mode regardless of the transmit mask register setting. The transmit last slot interrupt may be used for resetting the transmit mask slot register, reconfiguring the DMA channels and reassigning data memory

pointers. Using the transmit last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame is serviced with the new setting without synchronization problems. Note that the maximum transmit last slot interrupt service time should not exceed  $N-1$  ESAI bits service time (where  $N$  is the number of bits in a slot).

#### 7. **ESAI Transmit Even Data:**

Occurs when the transmit even slot data interrupt is enabled ( $TEDIE=1$ ), at least one of the enabled transmit data registers is empty ( $TDE=1$ ), the slot is an even slot ( $TEDE=1$ ) and no exception has occurred ( $TUE=0$  or  $TEIE=0$ ).

Writing to all the TX registers of the enabled transmitters or to TSR clears this interrupt request.

#### 8. **ESAI Transmit Data:**

Occurs when the transmit interrupt is enabled ( $TIE=1$ ), at least one of the enabled transmit data registers is empty ( $TDE=1$ ), no exception has occurred ( $TUE=0$  or  $TEIE=0$ ) and no even slot interrupt has occurred ( $TEDE=0$  or  $TEDIE=0$ ).

Writing to all the TX registers of the enabled transmitters, or to the TSR clears this interrupt request.

### 9.3.4 **Operating Modes—Normal, Network and On-Demand**

The ESAI has three basic operating modes and many data/operation formats.

#### 9.3.4.1 **Normal/Network/On-Demand Mode Selection**

Selecting between the normal mode and network mode is accomplished by clearing or setting the  $TMOD0$ - $TMOD1$  bits in the TCR register for the transmitter section, as well as in the  $RMOD0$ - $RMOD1$  bits in the RCR register for the receiver section.

For normal mode, the ESAI functions with one data word of I/O per frame (per enabled transmitter or receiver). The normal mode is typically used to transfer data to/from a single device.

For the network mode, 2 to 32 time slots per frame may be selected. During each frame, 0 to 32 data words of I/O may be received/transmitted. In either case, the transfers are periodic. The frame sync signal indicates the first time slot in the frame. Network mode is typically used in time division multiplexed (TDM) networks of codecs, DSPs with multiple words per frame, or multi-channel devices.

Selecting the network mode and setting the frame rate divider to zero ( $DC=00000$ ) selects the on-demand mode. This special case does not generate a periodic frame sync. A frame sync pulse is generated only when data is available to transmit. The on-demand mode requires that the transmit frame sync be internal (output) and the receive frame sync be external (input). Therefore, for simplex operation, the synchronous mode could be used; however, for full-duplex operation, the asynchronous mode must be used. Data transmission that is data driven is enabled by writing data into each TX. Although the ESAI is double buffered, only one word can be written to each TX, even if the transmit shift register is empty. The receive and transmit interrupts function as usual using TDE and RDF; however, transmit underruns are impossible for on-demand transmission and are disabled.

### 9.3.4.2 Synchronous/Asynchronous Operating Modes

The transmit and receive sections of the ESAI may be synchronous or asynchronous, that is, the transmitter and receiver sections may use common clock and synchronization signals (synchronous operating mode), or they may have their own separate clock and sync signals (asynchronous operating mode). The SYN bit in the SAICR register selects synchronous or asynchronous operation. Because the ESAI is designed to operate either synchronously or asynchronously, separate receive and transmit interrupts are provided.

When SYN is cleared, the ESAI transmitter and receiver clocks and frame sync sources are independent. If SYN is set, the ESAI transmitter and receiver clocks and frame sync come from the transmitter section (either external or internal sources).

Data clock and frame sync signals can be generated internally by the DSP or may be obtained from external sources. If internally generated, the ESAI clock generator is used to derive high frequency clock, bit clock and frame sync signals from the DSP internal system clock.

### 9.3.4.3 Frame Sync Selection

The frame sync can be either a bit-long or word-long signal. The transmitter frame format is defined by the TFSL bit in the TCR register. The receiver frame format is defined by the RFSL bit in the RCR register.

1. In the word-long frame sync format, the frame sync signal is asserted during the entire word data transfer period. This frame sync length is compatible with codecs, SPI serial peripherals, serial A/D and D/A converters, shift registers and telecommunication PCM serial I/O.
2. In the bit-long frame sync format, the frame sync signal is asserted for one bit clock immediately before the data transfer period. This frame sync length is compatible with Intel and National components, codecs and telecommunication PCM serial I/O.

The relative timing of the word length frame sync as referred to the data word is specified by the TFSR bit in the TCR register for the transmitter section and by the RFSR bit in the RCR register for the receive section. The word length frame sync may be generated (or expected) with the first bit of the data word, or with the last bit of the previous word. TFSR and RFSR are ignored when a bit length frame sync is selected.

Polarity of the frame sync signal may be defined as positive (asserted high) or negative (asserted low). The TFSP bit in the TCCR register specifies the polarity of the frame sync for the transmitter section. The RFSP bit in the RCCR register specifies the polarity of the frame sync for the receiver section.

The ESAI receiver looks for a receive frame sync leading edge (trailing edge if RFSP is set) only when the previous frame is completed. If the frame sync goes high before the frame is completed (or before the last bit of the frame is received in the case of a bit frame sync or a word length frame sync with RFSR set), the current frame sync is not recognized, and the receiver is internally disabled until the next frame sync. Frames do not have to be adjacent, that is, a new frame sync does not have to immediately follow the previous frame. Gaps of arbitrary periods can occur between frames. Enabled transmitters are tri-stated during these gaps.

When operating in the synchronous mode (SYN=1), all clocks including the frame sync are generated by the transmitter section.



### 9.3.4.4 Shift Direction Selection

Some data formats, such as those used by codecs, specify MSB first while other data formats, such as the AES-EBU digital audio interface, specify LSB first. The MSB/LSB first selection is made by programming RSHFD bit in the RCR register for the receiver section and by programming the TSHFD bit in the TCR register for the transmitter section.

### 9.3.5 Serial I/O Flags

Three ESAI pins (FSR, SCKR and HCKR) are available as serial I/O flags when the ESAI is operating in the synchronous mode (SYN=1). Their operation is controlled by RCKD, RFSD, TEBE bits in the RCR, RCCR and SAICR registers. The output data bits (OF2, OF1 and OF0) and the input data bits (IF2, IF1 and IF0) are double buffered to/from the HCKR, FSR and SCKR pins. Double buffering the flags keeps them in sync with the TX and RX data lines.

Each flag can be separately programmed. Flag 0 (SCKR pin) direction is selected by RCKD, RCKD=1 for output and RCKD=0 for input. Flag 1 (FSR pin) is enabled when the pin is not configured as external transmitter buffer enable (TEBE=0) and its direction is selected by RFSD, RFSD=1 for output and RFSD=0 for input. Flag 2 (HCKR pin) direction is selected by RHCKD, RHCKD=1 for output and RHCKD=0 for input.

When programmed as input flags, the SCKR, FSR and HCKR logic values, respectively, are latched at the same time as the first bit of the receive data word is sampled. Because the input was latched, the signal on the input flag pin (SCKR, FSR or HCKR) can change without affecting the input flag until the first bit of the next receive data word. When the received data words are transferred to the receive data registers, the input flag latched values are then transferred to the IF0, IF1 and IF2 bits in the SAISR register, where they may be read by software.

When programmed as output flags, the SCKR, FSR and HCKR logic values are driven by the contents of the OF0, OF1 and OF2 bits in the SAICR register respectively, and they are driven when the transmit data registers are transferred to the transmit shift registers. The value on SCKR, FSR and HCKR is stable from the time the first bit of the transmit data word is transmitted until the first bit of the next transmit data word is transmitted. Software may change the OF0-OF2 values thus controlling the SCKR, FSR and HCKR pin values for each transmitted word. The normal sequence for setting output flags when transmitting data is as follows: wait for TDE (transmitter empty) to be set; first write the flags, and then write the transmit data to the transmit registers. OF0, OF1, and OF2 are double buffered so that the flag states appear on the pins when the transmit data is transferred to the transmit shift register, that is, the flags are synchronous with the data.

## 9.4 GPIO—Pins and Registers

The GPIO functionality of the ESAI port is controlled by three registers: Port C control register (PCRC), Port C direction register (PRRC) and Port C data register (PDRC).

### 9.4.1 Port C Control Register (PCRC)

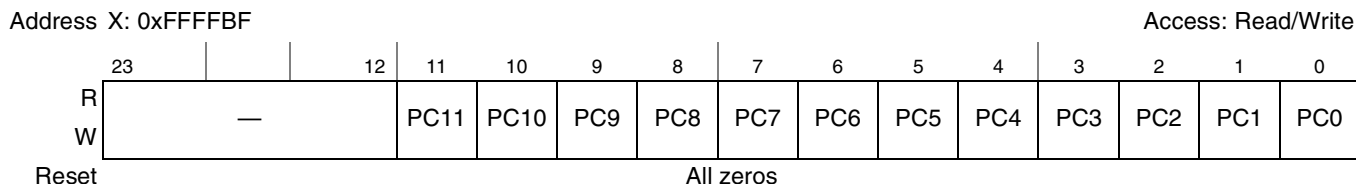
The read/write 24-bit Port C Control Register (PCRC) in conjunction with the Port C Direction Register (PRRC) controls the functionality of the ESAI GPIO pins. Each of the PC(11:0) bits controls the functionality of the corresponding port pin. [Table 9-12](#) provides the port pin configurations. Hardware and software reset clear all PCRC bits.

### 9.4.2 Port C Direction Register (PRRC)

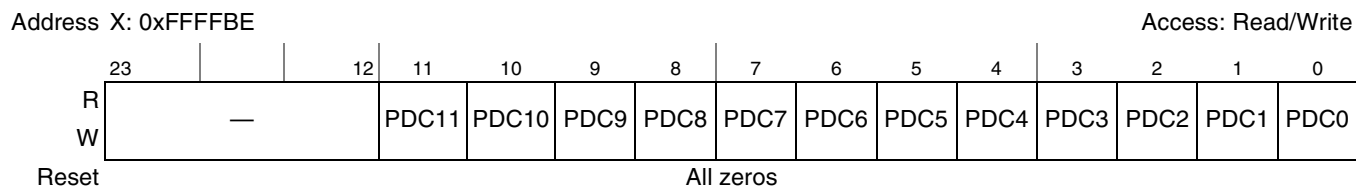
The read/write 24-bit Port C Direction Register (PRRC) in conjunction with the Port C Control Register (PCRC) controls the functionality of the ESAI GPIO pins. [Table 9-12](#) provides the port pin configurations. Hardware and software reset clear all PRRC bits.

**Table 9-12. PCRC and PRRC Bits Functionality**

PDC[i]	PC[i]	Port Pin[i] Function
0	0	disconnected
0	1	GPIO input
1	0	GPIO output
1	1	ESAI



**Figure 9-19. PCRC Register**



**Figure 9-20. PRRC Register**

### 9.4.3 Port C Data Register (PDR)

The read/write 24-bit Port C Data Register is used to read or write data to/from ESAI GPIO pins. Bits PD(11:0) are used to read or write data from/to the corresponding port pins if they are configured as GPIO. If a port pin [i] is configured as a GPIO input, the corresponding PD[i] bit reflects the value present on this pin. If a port pin [i] is configured as a GPIO output, the value written into the corresponding PD[i] bit is reflected on this pin. If a port pin [i] is configured as disconnected, the corresponding PD[i] bit is not reset and contains undefined data. [Figure 9-21](#) shows the Port C Data register.

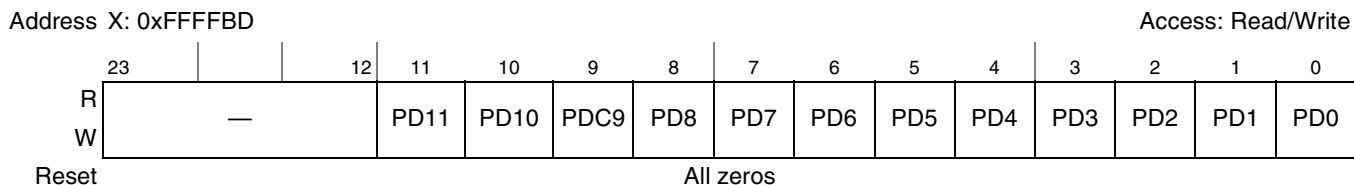


Figure 9-21. PDRC Registers

## 9.5 ESAI Initialization Examples

### 9.5.1 Initializing the ESAI Using Individual Reset

1. The ESAI should be in its individual reset state (PCRC = 0x000 and PRRC = 0x000). In the individual reset state, both the transmitter and receiver sections of the ESAI are simultaneously reset. The TPR bit in the TCR register may be used to reset the transmitter section only. The RPR bit in the RCR register may be used to reset the receiver section only.
2. Configure the control registers (TCCR, TCR, RCCR, RCR) according to the operating mode, but do not enable transmitters (TE5–TE0 = 0x0) or receivers (RE3–RE0 = 0x0). It is possible to set the interrupt enable bits which are in use during the operation (no interrupt occurs).
3. Enable the ESAI by setting the PCRC register and PRRC register bits according to pins which are in use during operation.
4. Write the first data to be transmitted to the transmitters which are in use during operation. This step is needed even if DMA is used to service the transmitters.
5. Enable the transmitters and receivers.
6. From now on ESAI can be serviced either by polling, interrupts, or DMA.

Operation proceeds as follows:

1. For internally generated clock and frame sync, these signals are active immediately after ESAI is enabled (step 3).
2. Data is received only when one of the receive enable (REx) bits is set and after the occurrence of frame sync signal (either internally or externally generated).
3. Data is transmitted only when the transmitter enable (TE<sub>x</sub>) bit is set and after the occurrence of frame sync signal (either internally or externally generated). The transmitter outputs remain tri-stated after TE<sub>x</sub> bit is set until the frame sync occurs.

### 9.5.2 Initializing Only the ESAI Transmitter Section

1. It is assumed that the ESAI is operational; that is, at least one pin is defined as an ESAI pin.
2. The transmitter section should be in its personal reset state (TPR = 1).
3. Configure the control registers TCCR and TCR according to the operating mode, making sure to clear the transmitter enable bits (TE0–TE5). TPR must remain set.
4. Take the transmitter section out of the personal reset state by clearing TPR.

5. Write first data to the transmitters which will be used during operation. This step is needed even if DMA is used to service the transmitters.
6. Enable the transmitters by setting their TE bits.
7. Data is transmitted only when the transmitter enable (TE<sub>x</sub>) bit is set and after the occurrence of frame sync signal (either internally or externally generated). The transmitter outputs remain tri-stated after TE<sub>x</sub> bit is set until the frame sync occurs.
8. From now on the transmitters are operating and can be serviced either by polling, interrupts, or DMA.

### 9.5.3 Initializing Only the ESAI Receiver Section

1. It is assumed that the ESAI is operational; that is, at least one pin is defined as an ESAI pin.
2. The receiver section should be in its personal reset state (RPR = 1).
3. Configure the control registers RCCR and RCR according to the operating mode, making sure to clear the receiver enable bits (RE0–RE3). RPR must remain set.
4. Take the receiver section out of the personal reset state by clearing RPR.
5. Enable the receivers by setting their RE bits.
6. From now on the receivers are operating and can be serviced either by polling, interrupts, or DMA.

## 9.6 ESAI/ESAI\_2 and ESAI\_1/ESAI\_3 Pin Switch

To support more flexible applications, a pin switch between ESAI and ESAI\_2 (or ESAI\_1 and ESAI\_3) can be enabled. Each pin of ESAI can be assigned to ESAI\_2 with same function; and at the same time, the same function pin of the ESAI\_2 is assigned to ESAI. In the same way, each functional pin can also be switched between ESAI\_1 and ESAI3. The switch feature of each pair of function pins can be controlled independently; the control bits for all pairs of ESAI/ESAI\_2 and ESAI\_1/ESAI\_3 pins are in the Chip Configuration Registers; details are in [Chapter 20, “Chip Configuration Module.”](#)

This feature is especially useful in smaller packages; Using the pin switch feature, the bonded out ESAI group pins can be used by the DSP Core-0’s ESAI, or by DSP Core-1’s ESAI\_2 block when the switch feature is enabled for all of the ESAI pins. Even more, some of the pins can be assigned to ESAI while the other pins are being used by ESAI\_2; when active, these pins switch control bits.

GPIO function is part of the function of the ESAI blocks, so the pin switch feature also affects the GPIO function. For example, some of the ESAI pins can be configured as GPIO Port C pins, controlled by Core-0; if the control of these pins is switched, then these pins can be used as GPIO Port C1, controlled by Core-1.

## 9.7 Internal Clock Connections Between ESAI and ESAI\_1, ESAI\_2 and ESAI\_3

In smaller packages, the clock and frame sync pins of ESAI\_1 and ESAI\_2 are not be bonded out, while the clock and frame sync pins ESAI, ESAI\_3 are bonded out. Using internal clock connection logic, ESAI\_1 can be connected to the ESAI pin-out clocks and frame sync pins internally, and also ESAI\_2 can

be connected to the ESAI\_3 pin-out clocks and frame sync pins internally. For more information about this, see [Chapter 20, “Chip Configuration Module”](#).



# Chapter 10

## Serial Host Interface (SHI, SHI\_1)

### 10.1 Introduction

The serial host interface (SHI) is a serial I/O interface that provides a communication path for program/coefficient data transfers between the DSP and an external host processor. The SHI can also communicate with other serial peripheral devices.

The SHI supports two well-known and widely used synchronous serial buses: the serial peripheral interface (SPI) bus and the Philips Inter-Integrated-Circuit Control (I<sup>2</sup>C) bus. The SHI supports both bus protocols, as either a slave or a single-master device. To minimize DSP overhead, the SHI supports 8-bit, 16-bit and 24-bit data transfers. The SHI has a 1 or 10-word receive FIFO that permits receiving up to 30 bytes before generating a receive interrupt, reducing the overhead for data reception.

#### NOTE

The DSP56724/DSP56725 has two SHI modules: SHI, SHI\_1. The only difference between SHI and SHI\_1 is that SHI is used by DSP Core-0 and SHI\_1 is used by DSP Core-1. Only SHI is described in detail in this chapter.

When configured in the SPI mode, the SHI can perform the following functions:

- Identify its slave selection (in slave mode)
- Simultaneously transmit (shift out) and receive (shift in) serial data
- Directly operate with 8-, 16- and 24-bit words
- Generate vectored interrupts separately for receive and transmit events, and update status bits
- Generate a separate vectored interrupt for a receive exception
- Generate a separate vectored interrupt for a bus-error exception
- Generate the serial clock signal (in master mode)
- Trigger DMA interrupts to service the transmit and receive events

When configured in the I<sup>2</sup>C mode, the SHI can perform the following functions:

- Detect/generate start and stop events
- Identify its slave (ID) address (in slave mode)
- Identify the transfer direction (receive/transmit)
- Transfer data byte-wise according to the SCL clock line
- Generate ACK signal following a byte receive
- Inspect ACK signal following a byte transmit
- Directly operate with 8-, 16- and 24-bit words

## Serial Host Interface (SHI, SHI\_1)

- Generate vectored interrupts separately for receive and transmit events, and update status bits
- Generate a separate vectored interrupt for a receive exception
- Generate a separate vectored interrupt for a bus error exception
- Generate the clock signal (in master mode)
- Trigger DMA interrupts to service the transmit and receive events

## 10.2 Serial Host Interface Internal Architecture

The DSP core views the SHI as a memory-mapped peripheral in the X data memory space. The DSP uses the SHI as a normal memory-mapped peripheral using standard polling, interrupt programming techniques, or DMA transfers. Memory mapping enables the DSP to communicate with the SHI registers using standard instructions and addressing modes. In addition, the MOVEP instruction allows interface-to-memory and memory-to-interface data transfers without going through an intermediate register. The DMA controller can be used to service the receive or transmit data paths. The single master configuration allows the DSP to directly connect to dumb peripheral devices; for that purpose, a programmable baud-rate generator is included to generate the clock signal for serial transfers. The host side invokes the SHI for communication and data transfers with the DSP through a shift register that can be accessed serially, using either the I<sup>2</sup>C or the SPI bus protocols. Figure 10-1 shows the SHI block diagram.

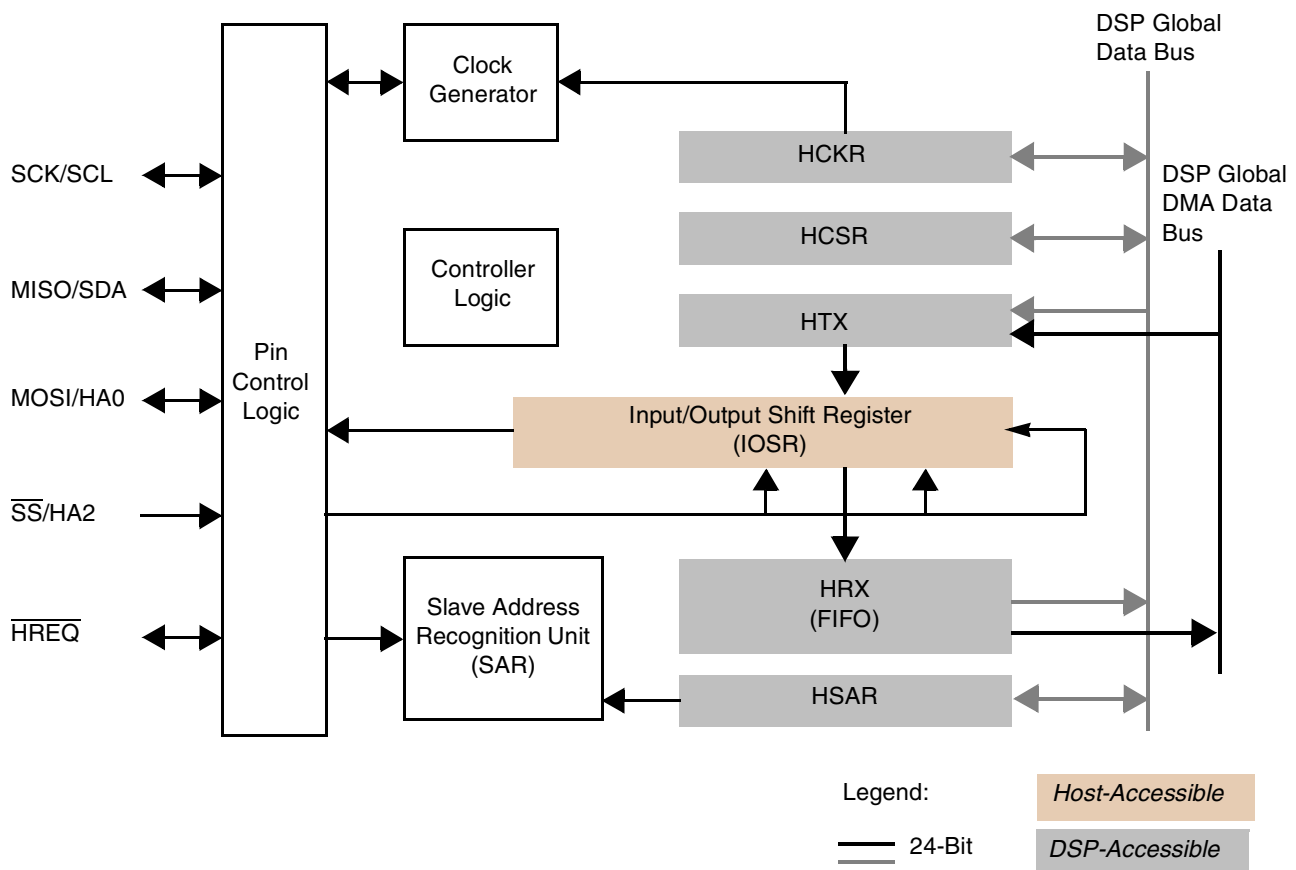


Figure 10-1. Serial Host Interface Block Diagram



## 10.3 SHI Clock Generator

If the interface operates in master mode, the SHI clock generator generates the SHI serial clock. If the interface operates in slave mode, the SHI clock generator is disabled, except if the interface is in I<sup>2</sup>C mode when the HCKFR bit is set in the HCKR register.

When the SHI operates in slave mode, the clock is external and is input to the SHI (HMST = 0).

Figure 10-2 shows the internal clock path connections. It is the user's responsibility to select the proper clock rate within the correct range (defined in the I<sup>2</sup>C and SPI bus specifications).

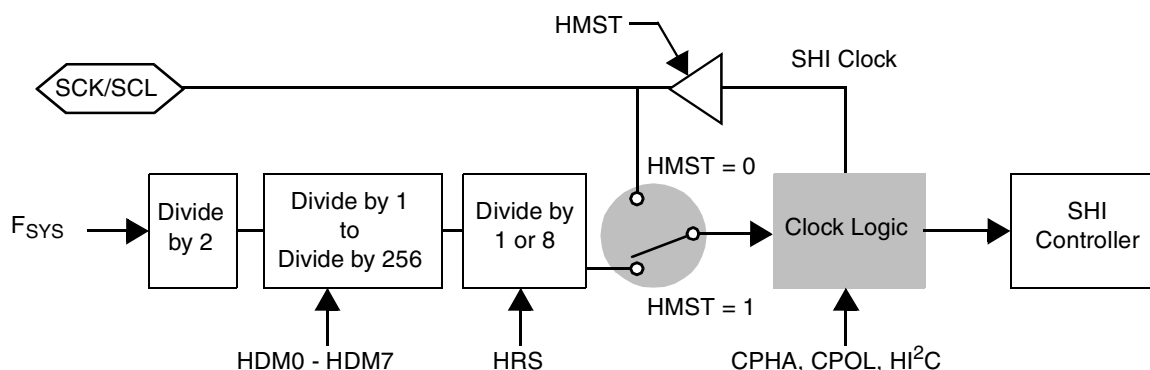


Figure 10-2. SHI Clock Generator

### 10.3.1 Serial Host Interface Programming Model

The Serial Host Interface programming model has two parts: a Host side and a DSP side.

- Host side—See Figure 10-3 and Section 10.3.2, “SHI Input/Output Shift Register (IOSR)—Host Side.”
- DSP side—See Figure 10-4 and Section 10.3.3, “SHI Host Transmit Data Register (HTX)—DSP Side,” through Section 10.3.8, “SHI Control/Status Register (HCSR)—DSP Side.”

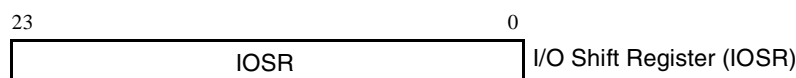


Figure 10-3. SHI Programming Model—Host Side

AA0418

# Serial Host Interface (SHI, SHI\_1)

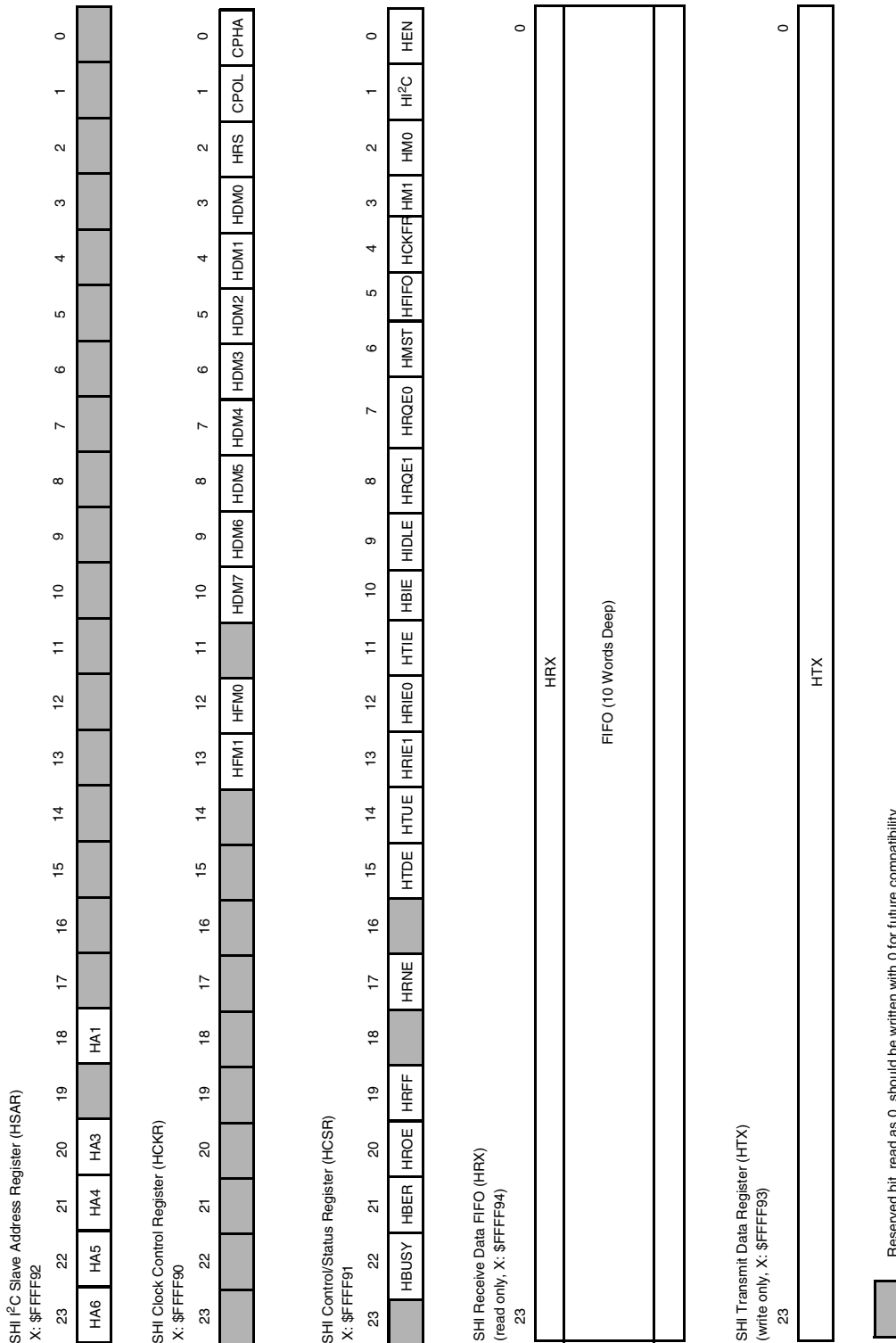


Figure 10-4. SHI Programming Model—DSP Side

For the SHI interrupt vector table, see [Table 10-1](#) For the exception priorities generated by the SHI, see [Table 10-2](#)

**Table 10-1. SHI Interrupt Vectors**

Program Address	Interrupt Source
VBA:0x0040	SHI Transmit Data
VBA:0x0042	SHI Transmit Underrun Error
VBA:0x0044	SHI Receive FIFO Not Empty
VBA:0x0048	SHI Receive FIFO Full
VBA:0x004A	SHI Receive Overrun Error
VBA:0x004C	SHI Bus Error

**Table 10-2. SHI Internal Interrupt Priorities**

Priority	Interrupt
Highest	SHI Bus Error
—	SHI Receive Overrun Error
—	SHI Transmit Underrun Error
—	SHI Receive FIFO Full
—	SHI Transmit Data
Lowest	SHI Receive FIFO Not Empty

### 10.3.2 SHI Input/Output Shift Register (IOSR)—Host Side

The variable length Input/Output Shift Register (IOSR) can be viewed as a serial-to-parallel and parallel-to-serial buffer in the SHI. The IOSR register is involved with every data transfer in both directions (read and write). In compliance with the I<sup>2</sup>C and SPI bus protocols, the data is shifted in and out MSB first. See [Figure 10-5](#)

- In 8-bit data transfer modes, the most significant byte of the IOSR is used as the shift register.
- In 16-bit data transfer modes, the two most significant bytes become the shift register.
- In 24-bit transfer modes, the shift register uses all three bytes of the IOSR.

#### NOTE

The IOSR register cannot be accessed directly either by the host processor or by the DSP core. The IOSR register is fully controlled by the SHI controller logic.

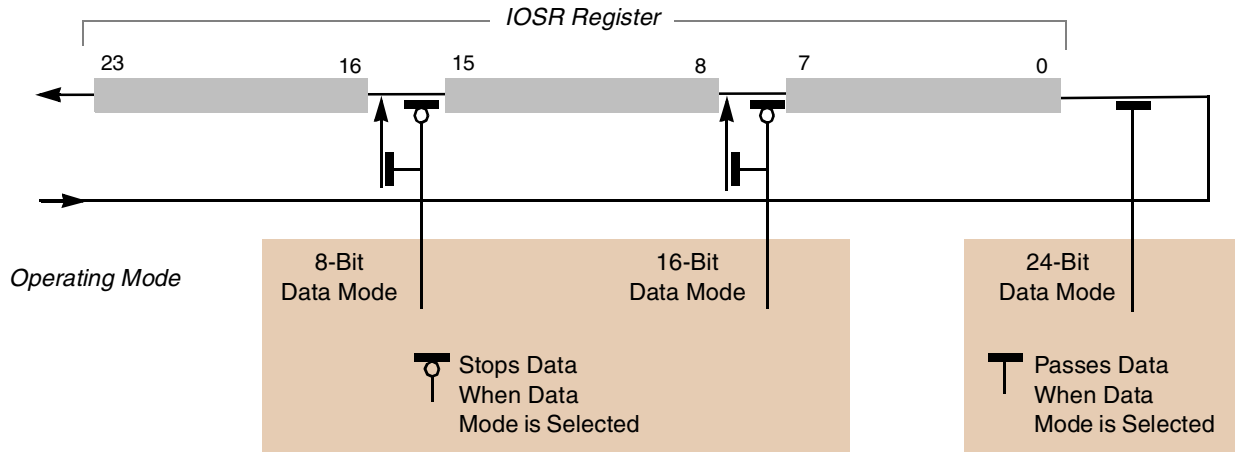


Figure 10-5. SHI I/O Shift Register (IOSR)

### 10.3.3 SHI Host Transmit Data Register (HTX)—DSP Side

The host transmit data register (HTX) is used for DSP-to-Host data transfers, and is 24 bits wide. Writing to the HTX register using DSP core instructions or by DMA transfers clears the HTDE flag. The DSP may program the HTIE bit to cause a host transmit data interrupt when the HTDE bit is set. (See [Section 10.3.8.10, “HCSR Transmit-Interrupt Enable \(HTIE\)—Bit 11.”](#)) To prevent overwriting the previous data, data should not be written to the HTX register until the HTDE bit is set. The HTX register is reset to the empty state when in stop mode, and also during hardware, software, and individual resets. In the different data transfer modes, the following occurs:

- In 8-bit data transfer mode, the most significant byte of the HTX register is transmitted.
- In 16-bit data transfer mode, the two most significant bytes of the HTX register are transmitted.
- In 24-bit data transfer mode, all the contents of HTX register are transferred.

### 10.3.4 SHI Host Receive Data FIFO (HRX)—DSP Side

The 24-bit host receive data FIFO (HRX) is a 10-word deep, First-In-First-Out (FIFO) register used for Host-to-DSP data transfers. The serial data is received via the shift register and then loaded into the HRX FIFO. In the different data transfer modes, the following actions occur:

- In 8-bit data transfer mode, the most significant byte of the shift register is transferred to the HRX FIFO (the other bits are cleared).
- In 16-bit data transfer mode, the two most significant bytes are transferred (the least significant byte is cleared) to the HTX FIFO.
- In 24-bit data transfer mode, all 24 bits are transferred to the HRX FIFO.

The HRX FIFO may be read by the DSP while the HRX FIFO is being loaded from the shift register. Reading all data from the HRX FIFO clears the HRNE flag. The HRX FIFO may be read using DSP core instructions or by DMA transfers. The HRX FIFO is reset to the empty state when the chip is in stop mode, and also during hardware reset, software, and individual resets.

### 10.3.5 SHI Slave Address Register (HSAR)—DSP Side

The 24-bit slave address register (HSAR) is used when the SHI operates in the I<sup>2</sup>C slave mode; the HSAR register is ignored in the other operational modes. The HSAR register holds 5 bits of the 7-bit slave device address. The SHI also acknowledges the general call address specified by the I<sup>2</sup>C protocol (8 zeroes comprising a 7-bit address and a  $\overline{R/\overline{W}}$  bit), but treats any following data bytes as regular data. That is, the SHI does not differentiate between its dedicated address and the general call address. Note that the host processor cannot access the HSAR register.

#### 10.3.5.1 HSAR Reserved Bits—Bits 19, 17–0

These bits are reserved; they read as zero and should be written with zeroes for future compatibility.

### 10.3.6 HSAR I<sup>2</sup>C Slave Address (HA[6:3], HA1)—Bits 23–20,18

Part of the I<sup>2</sup>C slave device address is stored in the read/write HA[6:3] and HA1 bits of the HSAR register. The full 7-bit slave device address is formed by combining the HA[6:3] and HA1 bits with the HA0 and HA2 pins, to obtain the HA[6:0] slave device address. Whenever an I<sup>2</sup>C master device initiates an I<sup>2</sup>C bus transfer, the full 7-bit slave device address is compared to the received address byte. During hardware or software resets, HA[6:3] = 1011 and the HA1 bit is cleared; this results in a default slave device address of 1011[HA2]0[HA0].

### 10.3.7 SHI Clock Control Register (HCKR)—DSP Side

The HCKR register is a 24-bit read/write register that controls the SHI clock generator. The HCKR bits should be changed only while the SHI is in the individual reset state (HEN = 0 in the HCSR register).

For proper SHI clock set-up, please consult the data sheet. The programmer should not use the combination HRS = 1 and HDM[7:0] = 00000000, since that combination may cause synchronization problems and improper operations (it is an illegal combination).

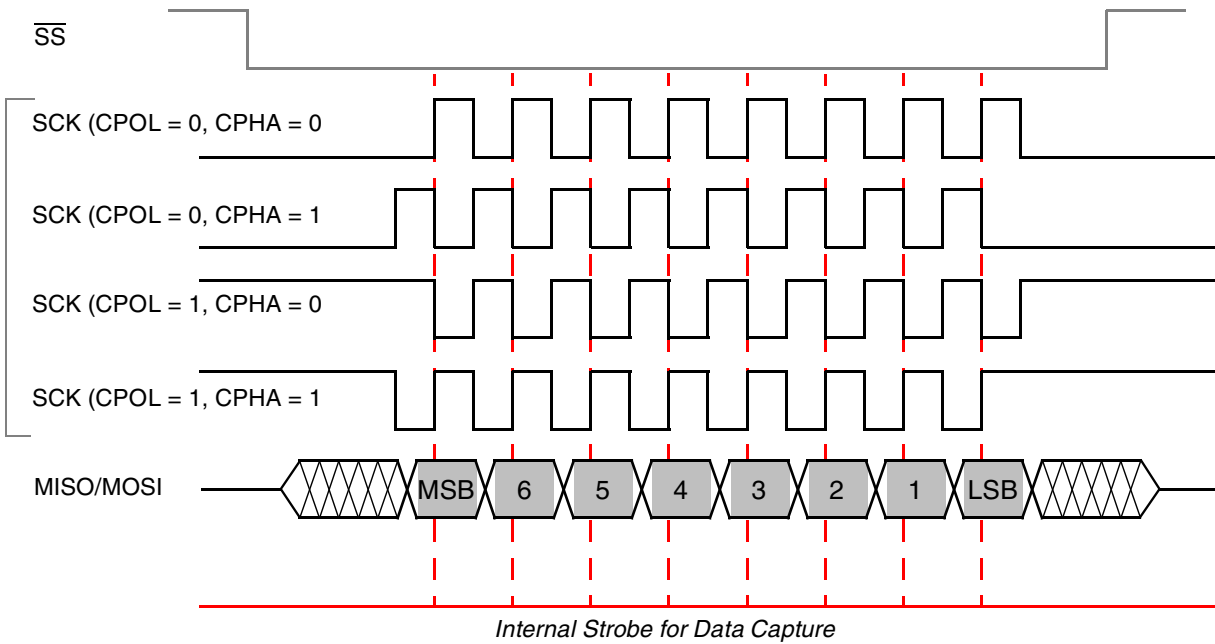
The HCKR bits are cleared during hardware or software resets, except for CPHA, which is set. The HCKR register is not affected by the stop state. The HCKR bits are described in the following sections.

#### 10.3.7.1 Clock Phase and Polarity (CPHA and CPOL)—Bits 1–0

The Clock Phase (CPHA) bit controls the relationship between the data on the master-in-slave-out (MISO) pin and master-out-slave-in (MOSI) pin, and also the clock produced or received at the SCK pin. The CPOL bit determines the clock polarity (1 = active-high clock, 0 = active-low clock).

The clock phase and polarity should be identical for both the master and slave SPI devices. The CPHA and CPOL bits are functional only when the SHI operates in the SPI mode; the CPHA and CPOL bits are ignored in the I<sup>2</sup>C mode. During hardware and software resets, the CPHA bit is set and the CPOL bit is cleared.

When operating in the SPI mode, you can select any one of four combinations of serial clock (SCK) phase and polarity. (See Figure 10-6.)



**Figure 10-6. SPI Data-To-Clock Timing Diagram**

If the CPOL bit is cleared, it produces a steady-state low value at the SCK pin of the master device whenever data is not being transferred. If the CPOL bit is set, it produces a high value at the SCK pin of the master device whenever data is not being transferred.

The CPHA and CPOL bits together select the desired clock-to-data relationship. In general, the CPHA bit selects the clock edge that captures data and allows it to change states. It has its greatest impact on the first bit transmitted (MSB) in that it does or does not allow a clock transition before the data capture edge.

- When the SHI is in slave mode and CPHA bit = 0, the  $\overline{SS}$  line must be deasserted and asserted by the external master between each successive word transfer. Also, the  $\overline{SS}$  line must remain asserted between successive bytes within a word. The DSP core should write the next data word to the HTX register when the HTDE bit = 1, thereby clearing the HTDE bit. However, the data is transferred to the shift register for transmission only when the  $\overline{SS}$  line is deasserted. The HTDE bit is set when the data is transferred from the HTX register to the shift register.
- When the SHI is in slave mode and CPHA bit = 1, the  $\overline{SS}$  line may remain asserted between successive word transfers. Also, the  $\overline{SS}$  line must remain asserted between successive bytes within a word. The DSP core should write the next data word to HTX register when HTDE bit = 1, thereby clearing the HTDE bit. The HTX register data is transferred to the shift register for transmission as soon as the shift register is empty. The HTDE bit is set when the data is transferred from the HTX register to the shift register.
- When the SHI is in master mode and CPHA bit = 0, the DSP core should write the next data word to the HTX register when HTDE bit = 1, thereby clearing the HTDE bit. The data is transferred immediately to the shift register for transmission. The HTDE bit is set only at the end of the data word transmission.

- The master is responsible for deasserting and asserting the slave device  $\overline{SS}$  line between word transmissions.
- When the SHI is in master mode and CPHA bit = 1, the DSP core should write the next data word to the HTX register when HTDE bit = 1, thereby clearing the HTDE bit. The HTX register data is transferred to the shift register for transmission as soon as the shift register is empty. The HTDE bit is set when the data is transferred from the HTX register to the shift register.

### 10.3.7.2 HCKR Prescaler Rate Select (HRS)—Bit 2

The HRS bit controls a prescaler in series with the clock generator divider. The HRS bit is used to extend the range of the divider when slower clock rates are desired.

- When the HRS bit is set, the prescaler is bypassed.
- When the HRS bit is cleared, the fixed divide-by-eight prescaler is operational.

When the SHI operates in the slave mode, the HRS bit is ignored, except during I<sup>2</sup>C mode when the HCKFR bit is set. The HRS bit is cleared during hardware and software resets.

#### NOTE

Use the equations in the SHI data sheet to determine the value of the HRS bit for the required serial clock frequency.

### 10.3.7.3 HCKR Divider Modulus Select (HDM[7:0])—Bits 10–3

The HDM[7:0] bits specify the divide ratio of the clock generator divider. A divide ratio between 1 and 256 (HDM[7:0] = 0x00 to 0xFF) can be selected. When the SHI operates in slave mode, the HDM[7:0] bits are ignored, except in HI<sup>2</sup>C mode when the HCKFR bit is set. The HDM[7:0] bits are cleared during hardware and software resets.

#### NOTE

Use the equations in the SHI data sheet to determine the value of the HDM[7:0] bits for the required serial clock frequency.

### 10.3.7.4 HCKR Reserved Bits—Bits 23–14, 11

These bits in HCKR are reserved; they are read as zero and should be written with zeroes for future compatibility.

### 10.3.7.5 HCKR HFM[1:0] Bits—Bits 13–12

The read/write control bits HFM[1:0] specify the operational mode of the noise reduction filters, as described in Table 10-3. The filters are designed to eliminate undesired spikes that might occur on the clock and data-in lines and allow the SHI to operate in noisy environments when required. One filter is located in the input path of the SCK/SCL line and the other is located in the input path of the data line (i.e., the SDA line when in I<sup>2</sup>C mode, the MISO line when in SPI master mode, and the MOSI line when in SPI slave mode).

**Table 10-3. SHI Noise Reduction Filter Mode**

HFM1	HFM0	Description
0	0	Bypassed (Disabled)
0	1	Very Narrow Spike Tolerance
1	0	Narrow Spike Tolerance
1	1	Wide Spike Tolerance

When HFM[1:0] = 00, the filter is bypassed (spikes are not filtered out). This mode is useful when higher bit-rate transfers are required and the SHI operates in a noise-free environment.

When HFM[1:0] = 01, the very narrow-spike-tolerance filter mode is selected. In this mode the filters eliminate spikes with durations of up to 10 ns. This mode is useful when very high bit-rate transfers are required and the SHI operates in a nearly noise-free environment.

When HFM[1:0] = 10, the narrow-spike-tolerance filter mode is selected. In this mode the filters eliminate spikes with durations of up to 50 ns. This mode is suitable for use in mildly noisy environments and imposes some limitations on the maximum achievable bit-rate transfer.

When HFM[1:0] = 11, the wide-spike-tolerance filter mode is selected. In this mode the filters eliminate spikes up to 100 ns. This mode is recommended for use in noisy environments; the bit-rate transfer is strictly limited. The wide-spike-tolerance filter mode is highly recommended for use in I<sup>2</sup>C bus systems as it fully conforms to the I<sup>2</sup>C bus specification and improves noise immunity.

#### NOTE

HFM[1:0] are cleared during hardware reset and software reset.

After changing the filter bits in the HCKR register to a non-bypass mode (HFM[1:0] is not equal to '00'), the programmer should wait at least ten times the tolerable spike width before enabling the SHI (setting the HEN bit in the HCSR register). Similarly, after changing the HI<sup>2</sup>C bit in the HCSR register or the CPOL bit in the HCKR register, while the filter mode bits are in a non-bypass mode (HFM[1:0] is not equal to '00'), the programmer should wait at least ten times the tolerable spike width before enabling the SHI (setting HEN bit in the HCSR register).

### 10.3.8 SHI Control/Status Register (HCSR)—DSP Side

The HCSR is a 24-bit register that controls the SHI operation and indicates its status. The control bits are read/write; the status bits are read-only. The HCSR register bits are described in the following sections.



When in the stop state or during individual reset, the HCSR status bits are reset to their hardware-reset state, while the HCSR control bits are not affected.

### 10.3.8.1 HCSR Host Enable (HEN)—Bit 0

The read/write control bit HEN, when set, enables the SHI. When HEN is cleared, the SHI is disabled (the SHI is in the individual reset state). When HEN is cleared, the HCKR and the HCSR control bits are not affected. When operating in master mode, the HEN bit should be cleared only when the SHI is idle (HBUSY = 0). The HEN bit is cleared during hardware and software resets.

#### 10.3.8.1.1 SHI Individual Reset

While the SHI is in the individual reset state, the SHI input pins are inhibited, output and bidirectional pins are disabled (high impedance), and the HCSR status bits and transmit/receive paths are reset to the same state produced by a hardware or software reset. After the HEN bit is cleared, SHI enters the individual reset state after a one-instruction-cycle delay.

### 10.3.8.2 HCSR I<sup>2</sup>C/SPI Selection (HI<sup>2</sup>C)—Bit 1

The read/write control bit HI<sup>2</sup>C selects whether the SHI operates in the I<sup>2</sup>C or SPI mode:

- When the HI<sup>2</sup>C bit is cleared, then SHI operates in the SPI mode.
- When the HI<sup>2</sup>C bit is set, then SHI operates in the I<sup>2</sup>C mode.

The HI<sup>2</sup>C bit affects the functionality of the SHI pins, as described in [Chapter 2, “Signal Descriptions.”](#) Before changing the HI<sup>2</sup>C bit, it is recommended that an SHI individual reset be generated (HEN bit cleared). The HI<sup>2</sup>C bit is cleared during hardware and software resets.

### 10.3.8.3 HCSR Serial Host Interface Mode (HM[1:0])—Bits 3–2

The read/write control bits HM[1:0] select the size of the data words to be transferred, as shown in [Table 10-4](#). The HM[1:0] bits should be modified only when the SHI is idle (HBUSY = 0). The HM[1:0] bits are cleared during hardware and software resets.

**Table 10-4. SHI Data Size**

HM1	HMO	Description
0	0	8-bit data
0	1	16-bit data
1	0	24-bit data
1	1	Reserved

#### 10.3.8.4 HCSR I<sup>2</sup>C Clock Freeze (HCKFR)—Bit 4

The read/write control bit HCKFR determines the behavior of the SHI when the SHI is unable to service the master request, while operating in the I<sup>2</sup>C slave mode. The HCKFR bit is used only in the I<sup>2</sup>C slave mode; the HCKFR bit is ignored otherwise.

- If the HCKFR bit is set, then the SHI holds the clock line to GND if the SHI is not ready to send data to the master (during a read transfer) or if the input FIFO is full (when the master attempts to execute a write transfer). In this way, the master can detect that the slave is not ready for the requested transfer, without causing an error condition in the slave. When the HCKFR bit is set for transmit sessions, the SHI clock generator must be programmed to generate the same serial clock as produced by the external master, otherwise erroneous operations may result. The programmed frequency should be in the range of 1 to 0.75 times the external clock frequency.
- If the HCKFR bit is cleared, then any attempt from the master to execute a transfer (when the slave is not ready) results in an overrun or underrun error condition.

Before changing the HCKFR bit, it is recommended that an SHI individual reset be generated (HEN bit is cleared). The HCKFR bit is cleared during hardware and software resets.

#### 10.3.8.5 HCSR FIFO-Enable Control (HFIFO)—Bit 5

The read/write control bit HFIFO selects the receive FIFO size:

- When the HFIFO bit is cleared, then the FIFO has one level.
- When the HFIFO bit is set, then the FIFO has 10 levels.

Before changing the HFIFO bit, it is recommended that an SHI individual reset be generated (HEN bit is cleared). The HFIFO bit is cleared during hardware and software resets.

#### 10.3.8.6 HCSR Master Mode (HMST)—Bit 6

The read/write control bit HMST determines the SHI operating mode:

- If the HMST bit is set, then the SHI interface operates in the master mode.
- If the HMST bit is cleared, then the SHI interface operates in the slave mode.

The SHI supports a single-master configuration in both I<sup>2</sup>C and SPI modes. When configured as an SPI master, the SHI drives the SCK line and controls the direction of the data lines using MOSI and MISO. In SPI master mode, the  $\overline{SS}$  line must be held deasserted; if the  $\overline{SS}$  line is asserted (when in SPI master mode), a bus error is generated (the HCSR HBER bit is set—see [Section 10.3.8.18, “Host Bus Error \(HBER\)—Bit 21”](#)).

When configured as an I<sup>2</sup>C master, the SHI controls the I<sup>2</sup>C bus by generating start events, clock pulses, and stop events for the transmission/reception of serial data.

Before changing the HMST bit, it is recommended that an SHI individual reset be generated (HEN bit is cleared). The HMST bit is cleared during hardware and software resets.

### 10.3.8.7 HCSR Host-Request Enable (HRQE[1:0])—Bits 8–7

The read/write control bits HRQE[1:0] control the  $\overline{\text{HREQ}}$  pin, as follows:

- When the HRQE[1:0] bits are cleared, the  $\overline{\text{HREQ}}$  pin is disabled and held in the high impedance state.
- If either of the HRQE[1:0] bits are set and the SHI is in master mode, then the  $\overline{\text{HREQ}}$  pin becomes an input controlling SCK: deasserting the  $\overline{\text{HREQ}}$  pin suspends SCK.
- If either of the HRQE[1:0] bits are set and the SHI is in SPI slave mode, then the  $\overline{\text{HREQ}}$  pin becomes an output, and its operation is defined in [Table 10-5](#)

**Table 10-5.  $\overline{\text{HREQ}}$  Function In SPI Slave Mode**

HRQE1	HRQE0	$\overline{\text{HREQ}}$ Pin Operation
0	0	High impedance
0	1	Asserted if IOSR is ready to receive a new word
1	0	Asserted if IOSR is ready to transmit a new word
1	1	SPI mode: Asserted if IOSR is ready to transmit and receive

The HRQE[1:0] bits should be changed only when the SHI is idle (HBUSY bit = 0). The HRQE[1:0] bits are cleared during hardware and software resets.

### 10.3.8.8 HCSR Idle (HIDLE)—Bit 9

The read/write control/status bit HIDLE is used only in the I<sup>2</sup>C master mode; the HIDLE bit is ignored otherwise. It is only possible to set the HIDLE bit during writes to the HCSR register. The HIDLE bit is cleared by writing to the HTX register.

- To ensure correct transmission of the slave device address byte, the HIDLE bit should be set only when the HTX register is empty (HTDE bit = 1). After the HIDLE bit is set, a write to the HTX register clears the HIDLE bit and causes the generation of a stop event, followed by a start event, and then the transmission of the 8 MSBs of the data as the slave device address byte.
- While the HIDLE bit is cleared, data written to the HTX register is transmitted as is.

After the SHI completes transmitting a word:

- If the HIDLE bit is cleared and there is no new data in the HTX register, then the clock is suspended after sampling ACK.
- If the HIDLE bit is set and there is no new data in the HTX register, then a stop event is generated.

The HIDLE bit determines the acknowledge that the receiver sends after a byte is received correctly.

- If the HIDLE bit is cleared, then the byte's reception is acknowledged by sending a 0 bit on the SDA line at the ACK clock tick.
- If the HIDLE bit is set, then the byte's reception is not acknowledged (a 1 bit is sent). It is used to signal an end-of-data to a slave transmitter by not generating an ACK on the last byte. As a result, the slave transmitter must release the SDA line to allow the master to generate the stop event.

If the SHI completes receiving a word and the HRX FIFO is full, the clock is suspended before transmitting an ACK. While the HIDL bit is cleared the bus is busy, that is, the start event was sent but no stop event was generated. Setting the HIDL bit causes a stop event after receiving the current word.

The HIDL bit is set while the SHI is not in I<sup>2</sup>C master mode, while the chip is in the stop state, and during hardware, software, and individual resets.

**NOTE**

Programmers should ensure that before setting the HIDL bit, all DMA channel service to the HTX register is disabled.

**10.3.8.9 HCSR Bus-Error Interrupt Enable (HBIE)—Bit 10**

The read/write control bit HBIE enables the SHI bus-error interrupt:

- If the HBIE bit is cleared, then bus-error interrupts are disabled, and the HBER status bit must be polled to determine if an SHI bus error occurred.
- If both the HBIE and HBER bits are set, then the SHI requests an SHI bus-error interrupt service from the interrupt controller. The HBIE bit is cleared by hardware and software resets.

**NOTE**

Clearing the HBIE bit masks a pending bus-error interrupt only after a one instruction cycle delay. If the HBIE bit is cleared in a long interrupt service routine, it is recommended that at least one other instruction is between the instruction (that clears the HBIE bit) and the RTI instruction at the end of the interrupt service routine.

**10.3.8.10 HCSR Transmit-Interrupt Enable (HTIE)—Bit 11**

The read/write control bit HTIE is used to enable the SHI transmit data interrupts:

- If the HTIE bit is cleared, then the transmit interrupts are disabled, and the HTDE status bit must be polled to determine if the HTX register is empty.
- If both the HTIE and HTDE bits are set and the HTUE bit is cleared, then the SHI requests an SHI transmit-data interrupt service from the interrupt controller.
- If both the HTIE and HTUE bits are set, then the SHI requests an SHI transmit-underrun-error interrupt service from the interrupt controller.

The HTIE bit is cleared by hardware and software resets.

**NOTE**

Clearing the HTIE bit masks a pending transmit interrupt only after a one instruction cycle delay. If the HTIE bit is cleared in a long interrupt service routine, it is recommended that at least one other instruction separates the instruction (that clears the HTIE bit) and the RTI instruction at the end of the interrupt service routine.

### 10.3.8.11 HCSR Receive Interrupt Enable (HRIE[1:0])—Bits 13–12

The read/write control bits HRIE[1:0] are used to enable the SHI receive-data interrupts:

- If the HRIE[1:0] bits are cleared, then the receive interrupts are disabled, and the HRNE and HRFF status bits must be polled to determine if there is data in the receive FIFO.
- If the HRIE[1:0] bits are not cleared, then receive interrupts are generated according to [Table 10-6](#)

The HRIE[1:0] bits are cleared by hardware and software resets.

**Table 10-6. HCSR Receive Interrupt Enable Bits**

HRIE[1:0]	Status Bits	Interrupt
00	—	Disabled
01	If HRNE = 1 and HROE = 0	Receive FIFO is not empty
	If HROE = 1	Receive Overrun Error
10	—	Reserved
11	If HRFF = 1 and HROE = 0	Receive FIFO is full
	If HROE = 1	Receive Overrun Error

#### NOTE

Clearing the HRIE[1:0] bits masks a pending receive interrupt only after a one instruction cycle delay. If the HRIE[1:0] bits are cleared in a long interrupt service routine, it is recommended that at least one other instruction separates the instruction (that clears the HRIE[1:0] bits) and the RTI instruction at the end of the interrupt service routine.

### 10.3.8.12 HCSR Host Transmit Underrun Error (HTUE)—Bit 14

The read-only status bit HTUE indicates whether a transmit-underrun error occurred. Transmit-underrun errors can occur only when operating in the SPI slave mode or the I<sup>2</sup>C slave mode (when HCKFR is cleared). In master mode, transmission takes place on demand and no underrun can occur. The HTUE bit is set when both the shift register and the HTX register are empty and the external master begins reading the next word:

- When operating in I<sup>2</sup>C mode, the HTUE bit is set on the falling edge of the ACK bit, and the SHI re-transmits the previously transmitted word.
- When operating in SPI mode,
  - If CPHA = 1, then the HTUE bit is set at the first clock edge.
  - If CPHA = 0, then the HTUE bit is set at the assertion of the  $\overline{SS}$  line.

If a transmit interrupt occurs:

- When the HTUE bit is set, then the transmit-underrun interrupt vector is generated.
- When the HTUE bit cleared, then the regular transmit-data interrupt vector is generated.

The HTUE bit is cleared by reading the HCSR register and then writing to the HTX register. The HTUE bit is cleared by hardware, software, and SHI individual resets, and also during the stop state.

### 10.3.8.13 HCSR Host Transmit Data Empty (HTDE)—Bit 15

The read-only status bit HTDE indicates whether the HTX register is empty and can be written by the DSP core.

- The HTDE bit is set when the data word is transferred from the HTX register to the shift register, except in SPI master mode when CPHA bit = 0 (see HCKR register).
- When in SPI master mode with CPHA bit = 0, the HTDE bit is set after the end of the data word transmission.
- The HTDE bit is cleared when the DSP writes the HTX register either with write instructions or DMA transfers.

The HTDE bit is set by hardware, software, and SHI individual resets, and also during the stop state.

### 10.3.8.14 HCSR Reserved Bits—Bits 23, 18 and 16

These bits are reserved; they read as zero and should be written with zeroes for future compatibility.

### 10.3.8.15 Host Receive FIFO Not Empty (HRNE)—Bit 17

The read-only status bit HRNE indicates that the Host Receive FIFO (HRX) contains at least one data word.

- The HRNE bit is set when the HRX FIFO is not empty.
- The HRNE bit is cleared when the HRX FIFO is read by the DSP (using read instructions or DMA transfers), reducing the number of words in the FIFO to zero.

The HRNE bit is cleared during hardware, software, and SHI individual resets, and also during the stop state.

### 10.3.8.16 Host Receive FIFO Full (HRFF)—Bit 19

The read-only status bit HRFF indicates, when set, that the Host Receive FIFO (HRX) is full. The HRFF bit is cleared when the HRX FIFO is read by the DSP (using read instructions or DMA transfers) and at least one place is available in the HRX FIFO. The HRFF bit is cleared by hardware, software, and SHI individual resets, and also during the stop state.

### 10.3.8.17 Host Receive Overrun Error (HROE)—Bit 20

The read-only status bit HROE indicates, when set, that a data-receive overrun error has occurred. Receive-overrun errors cannot occur when operating in I<sup>2</sup>C master mode, because the clock is suspended if the receive FIFO is full; nor can receive-overrun errors occur in I<sup>2</sup>C slave mode when the HCKFR bit is set.

The HROE bit is set when the shift register (IOSR) is filled and ready to transfer the data word to the HRX FIFO and the FIFO is already full (HRFF bit is set). When a receive-overflow error occurs, the shift register is not transferred to the HRX FIFO.

- If a receive interrupt occurs when the HROE bit is set, the receive-overflow interrupt vector is generated.
- If a receive interrupt occurs when the HROE is cleared, the regular receive-data interrupt vector is generated.

The HROE bit is cleared by reading the HCSR register with the HROE bit set, followed by reading the HRX FIFO. The HROE bit is cleared by hardware, software, and SHI individual resets, and also during the stop state.

#### 10.3.8.18 Host Bus Error (HBER)—Bit 21

When the read-only status bit HBER is set, it indicates that an SHI bus error occurred when operating as a master (HMST bit is set).

- In I<sup>2</sup>C mode, the HBER bit is set if the transmitter does not receive an acknowledge after a byte is transferred; then a stop event is generated and transmission is suspended.
- In SPI mode, the HBER bit is set if the  $\overline{SS}$  line is asserted; then the transmission is suspended at the end of transmission of the current word.

The HBER bit is cleared only by hardware, software, and SHI individual resets, and also during the stop state.

#### 10.3.8.19 HCSR Host Busy (HBUSY)—Bit 22

The read-only status bit HBUSY indicates that the I<sup>2</sup>C bus is busy (when in the I<sup>2</sup>C mode) or that the SHI itself is busy (when in the SPI mode).

- When operating in I<sup>2</sup>C mode, the HBUSY bit is set after the SHI detects a start event, and the HBUSY bit remains set until a stop event is detected.
- When operating in the slave SPI mode, the HBUSY bit is set while the  $\overline{SS}$  line is asserted.
- When operating in the master SPI mode, the HBUSY bit is set if the HTX register is not empty or if the IOSR register is not empty.

The HBUSY bit is cleared otherwise. The HBUSY bit is cleared by hardware, software, and SHI individual resets, and also during the stop state.

### 10.4 Characteristics of the SPI Bus

The SPI bus consists of two serial data lines (MISO, MOSI), a clock line (SCK) and a Slave Select line ( $\overline{SS}$ ). During an SPI transfer, a byte is shifted *out* one data pin while a different byte is simultaneously shifted *in* through a second data pin. It can be viewed as two 8-bit shift registers connected together in a circular manner, with one shift register on the master side and the other shift register on the slave side. Thus the data bytes in the master device and slave device are exchanged.

The MISO and MOSI data pins are used for transmitting and receiving serial data.

## Serial Host Interface (SHI, SHI\_1)

- When the SPI is configured as a master, MISO is the master data input line, and MOSI is the master data output line.
- When the SPI is configured as a slave, MISO is the slave data output line, and MOSI is the slave data input line.

Clock control logic allows a selection of clock polarity and a choice of two fundamentally different clocking protocols to accommodate most available synchronous serial peripheral devices. When the SPI is configured as a master, the control bits in the HCKR register select the appropriate clock rate, as well as the desired clock polarity and phase format. (See [Figure 10-6](#).)

The  $\overline{SS}$  line allows selection of an individual slave SPI device; slave devices that are not selected do not interfere with SPI bus activity—they keep their MISO output pin in the high-impedance state. When the SHI is configured as an SPI master device, the  $\overline{SS}$  line should be held high. If the  $\overline{SS}$  line is driven low when the SHI is in SPI master mode, a bus error is generated (the HCSR HBER bit is set).

## 10.5 Characteristics of the I<sup>2</sup>C Bus

The I<sup>2</sup>C serial bus consists of two bidirectional lines, one for data signals (SDA) and one for clock signals (SCL). Both the SDA and SCL lines must be connected to a positive supply voltage via a pull-up resistor.

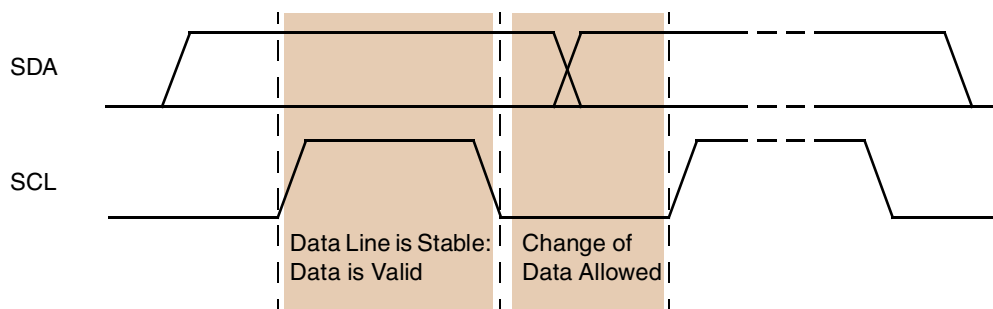
### NOTE

In the I<sup>2</sup>C bus specifications, the standard mode (100 kHz clock rate) and a fast mode (400 kHz clock rate) are defined. The SHI can operate in either standard or fast mode.

### 10.5.1 Overview of I<sup>2</sup>C Bus

The I<sup>2</sup>C bus protocol must conform to the following rules:

- Data transfers may be initiated only when the bus is not busy.
- During data transfers, the data line must remain stable whenever the clock line is high. Changes in the data line when the clock line is high are interpreted as control signals (see [Figure 10-7](#)).



AA0422

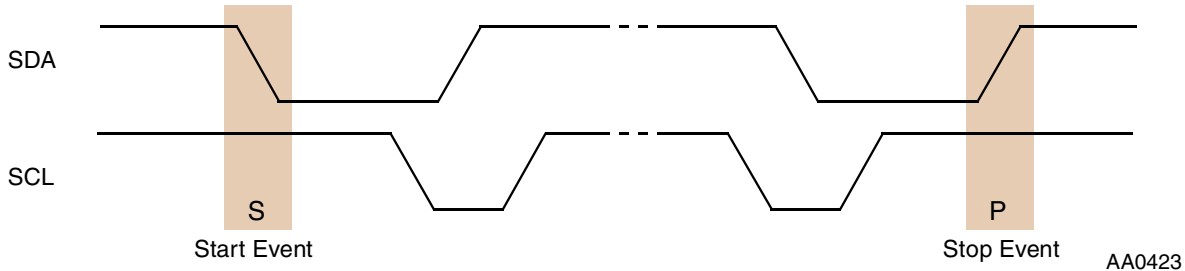
Figure 10-7. I<sup>2</sup>C Bit Transfer

The I<sup>2</sup>C bus protocol defines the following events:

- Bus not busy—Both data and clock lines remain high.
- Start data transfer—The start event is defined as a change in the state of the data line, from high to low, while the clock is high (see [Figure 10-8](#)).

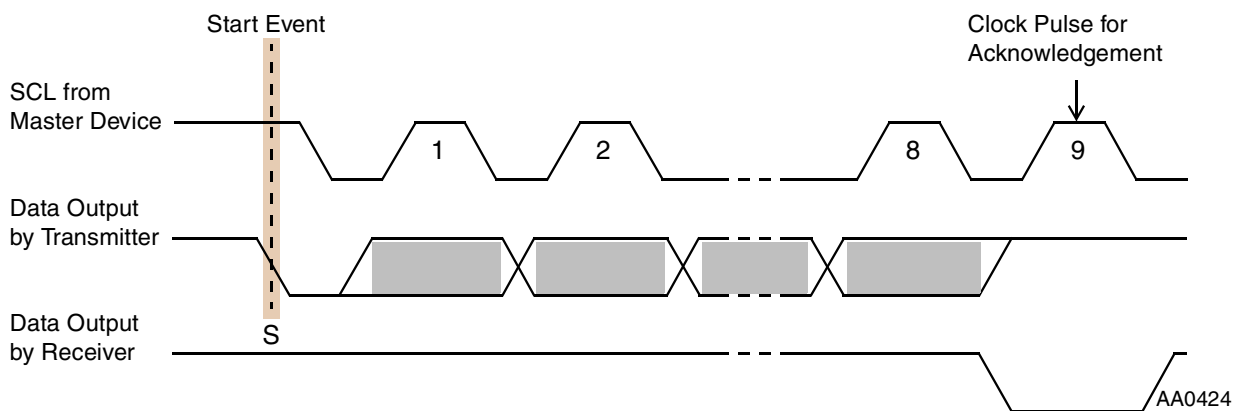


- Stop data transfer—The stop event is defined as a change in the state of the data line, from low to high, while the clock is high (see Figure 10-8).
- Data valid—The state of the data line represents valid data when, after a start event, the data line is stable for the duration of the high period of the clock signal. The data on the line may be changed during the low period of the clock signal. There is one clock pulse per bit of data.



**Figure 10-8. I<sup>2</sup>C Start and Stop Events**

Each 8-bit word is followed by one acknowledge bit. This acknowledge bit is a high level put on the bus by the transmitter when the master device generates an extra acknowledge-related clock pulse. A slave receiver that is addressed must generate an acknowledge after each byte is received. Also, a master receiver must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The acknowledging device must pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable low during the high period of the acknowledge-related clock pulse (see Figure 10-9).

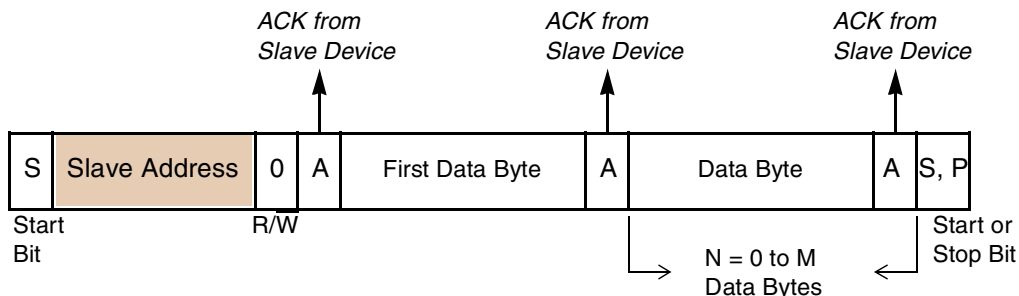


**Figure 10-9. Acknowledgment on the I<sup>2</sup>C Bus**

A device generating a signal is called a transmitter, and a device receiving a signal is called a receiver. A device controlling a signal is called a master, and devices controlled by the master are called slaves. A master receiver must signal an end-of-data to the slave transmitter by not generating an acknowledge on the last byte clocked out of the slave device. In this case the transmitter must leave the data line high to enable the master to generate the stop event. Handshaking may also be accomplished by using the clock synchronizing mechanism. Slave devices can hold the SCL line low, after receiving and acknowledging a byte, to force the master into a wait state until the slave device is ready for the next byte transfer. The SHI supports this feature when operating as a master device, and waits until the slave device releases the SCL line before proceeding with the data transfer.

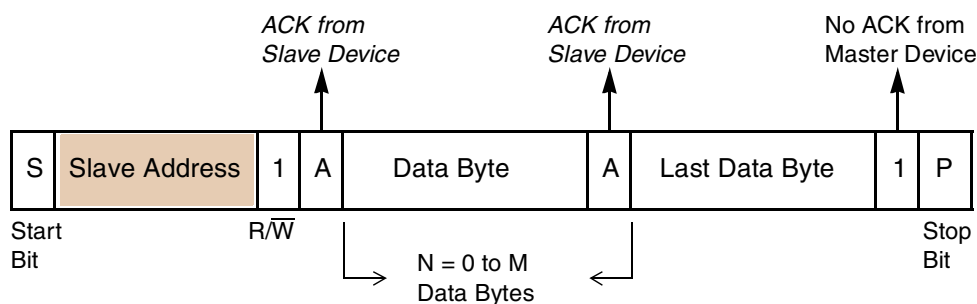
## 10.5.2 I<sup>2</sup>C Data Transfer Formats

I<sup>2</sup>C bus data transfers follow the following process: after the start event, a slave device address is sent. The address consists of 7 address bits and an 8th bit as a data direction bit (R/W). In the data direction bit, “0” indicates a transmission (write), and “1” indicates a request for data (read). A data transfer is always terminated by a stop event generated by the master device. However, if the master device still wishes to communicate on the bus, it can generate another start event and address another slave device without first generating a stop event. (The SHI does not support this feature when operating as an I<sup>2</sup>C master device.) This method is also used to provide indivisible data transfers. Various combinations of read/write formats are illustrated in Figure 10-10 and Figure 10-11.



AA0425

Figure 10-10. I<sup>2</sup>C Bus Protocol For Host Write Cycle



AA0426

Figure 10-11. I<sup>2</sup>C Bus Protocol For Host Read Cycle

### NOTE

The first data byte in a write-bus cycle can be used as a user-predefined control byte (for example, to determine the location where the forthcoming data bytes should be transferred to).

## 10.6 SHI Programming Considerations

The SHI implements both SPI and I<sup>2</sup>C bus protocols, and can be programmed to operate as a slave device or a single-master device. After the operating mode is selected, the SHI may communicate with an external device by receiving and/or transmitting data. Before changing the SHI operating mode, an SHI individual reset should be generated by clearing the HEN bit. The following sections describe programming considerations for each operating mode.

## 10.6.1 SPI Slave Mode

Initiate SPI slave mode by performing the following steps:

1. Set HEN = 1 to enable the SHI.
2. Set HI<sup>2</sup>C = 0 to select the SPI mode.
3. Set HMST = 0 to select the slave mode of operation.

The programmer should verify that the CPHA and CPOL bits (in the HCKR register) correspond to the external host clock phase and polarity. (Note that the other HCKR register bits are ignored.) When configured in the SPI slave mode, the SHI external pins operate as follows:

- SCK/SCL is the SCK serial clock input.
- MISO/SDA is the MISO serial data output.
- MOSI/HA0 is the MOSI serial data input.
- $\overline{SS}$ /HA2 is the  $\overline{SS}$  slave select input.
- $\overline{HREQ}$  is the Host Request output.

In the SPI slave mode, a receive, transmit, or full-duplex data transfer may be performed. Actually, the interface performs data receive and transmit simultaneously. The status bits of both receive and transmit paths are active; however, the programmer may disable undesired interrupts and ignore irrelevant status bits. Before beginning data reception, it is recommended that an SHI individual reset (HEN bit is cleared) be generated, to reset the HRX FIFO to its initial (empty) state (for example, when switching from transmitting data to receiving data).

If a write to the HTX register occurs, its contents are transferred to the IOSR register between data word transfers. The IOSR register data is shifted out (via MISO) and received data is shifted in (via MOSI). The DSP may write the HTX register using either DSP instructions or DMA transfers (if the HTDE status bit is set). If no writes to the HTX register occur, the contents of the HTX register are not transferred to the IOSR register, so the data shifted out when receiving is the data present in the IOSR register at that time. The HRX FIFO contains valid receive data, which the DSP can read using either DSP instructions or DMA transfers (if the HRNE status bit is set).

- If the  $\overline{HREQ}$  output pin is enabled for receive (HRQE[1:0] = 01), then the  $\overline{HREQ}$  output pin is asserted when the IOSR register is ready to receive and the HRX FIFO is not full; this operation guarantees that the next received data word is stored in the HRX FIFO.
- If the  $\overline{HREQ}$  output pin is enabled for transmit (HRQE[1:0] = 10), then the  $\overline{HREQ}$  output pin is asserted when the IOSR register is loaded from the HTX register with a new data word to transfer.
- If the  $\overline{HREQ}$  output pin is enabled for both transmit and receive (HRQE[1:0] = 11), then the  $\overline{HREQ}$  output pin is asserted when the receive and transmit conditions are both true.

The  $\overline{HREQ}$  is deasserted at the first clock pulse of the next data word transfer.

The  $\overline{HREQ}$  line can be used to interrupt the external master device. Connecting the  $\overline{HREQ}$  line between two SHI-equipped DSPs enables full hardware handshaking (if operating with CPHA = 1), with one DSP operating as an SPI master device and the other DSP as an SPI slave device.

The  $\overline{SS}$  line should be kept asserted during a data word transfer. If the  $\overline{SS}$  line is deasserted before the end of the data word transfer, the transfer is aborted and the received data word is lost.

## 10.6.2 SPI Master Mode

Initiate SPI master mode by performing the following steps:

1. Set HEN = 1 to enable the SHI.
2. Set HI<sup>2</sup>C = 0 to select the SPI mode.
3. Set HMST = 1 to select the master mode of operation.

Before enabling the SHI as an SPI master device, the programmer should program the proper clock rate, phase and polarity in the HCKR register. When configured in the SPI master mode, the SHI external pins operate as follows:

- SCK/SCL is the SCK serial clock output.
- MISO/SDA is the MISO serial data input.
- MOSI/HA0 is the MOSI serial data output.
- $\overline{SS}$ /HA2 is the  $\overline{SS}$  input. It should be kept deasserted (high) for proper operation.
- $\overline{HREQ}$  is the Host Request input.

The external slave device can be selected either by using external logic or by activating a GPIO pin connected to its  $\overline{SS}$  pin. However, the  $\overline{SS}$  input pin of the SPI master device should be held deasserted (high) for proper operation. If the SPI master device's  $\overline{SS}$  pin is asserted, the host bus error status bit (HBER) is set. If the HBIE bit is also set, the SHI issues a request to the DSP interrupt controller to service the SHI bus error interrupt.

In the SPI master mode, the DSP must write to the HTX register to receive, transmit or perform a full-duplex data transfer. Actually, the interface performs simultaneous data receive and transmit. The status bits of both receive and transmit paths are active; however, the programmer may disable undesired interrupts and ignore irrelevant status bits. In a data transfer, the HTX register is transferred to the IOSR register, clock pulses are generated, the IOSR register data is shifted out (via MOSI) and received data is shifted in (via MISO). The DSP programmer may write the HTX register (if the HTDE status bit is set) using either DSP instructions or DMA transfers to initiate the transfer of the next word. The HRX FIFO contains valid receive data, which the DSP can read using either DSP instructions or DMA transfers, if the HRNE status bit is set.

It is recommended that an SHI individual reset (HEN bit is cleared) be generated before beginning data reception to reset the receive FIFO to its initial (empty) state (for example, when switching from transmitting to receiving data).

If the HRQE[1:0] bits are cleared, the  $\overline{HREQ}$  input pin is ignored by the SPI master device; if any of the the HRQE[1:0] bits are set, the  $\overline{HREQ}$  input pin is considered by the SPI master device.

When asserted by the slave device, the  $\overline{HREQ}$  input pin indicates that the external slave device is ready for the next data transfer. As a result, the SPI master sends clock pulses for the full data word transfer. At the first clock pulse of the new data transfer, the external slave device deasserts  $\overline{HREQ}$ . When  $\overline{HREQ}$  is deasserted,  $\overline{HREQ}$  prevents the clock generation of the next data word transfer until  $\overline{HREQ}$  is asserted again.

Connecting the  $\overline{\text{HREQ}}$  line between two SHI-equipped DSPs enables full hardware handshaking (if  $\text{CPHA} = 1$ ), with one DSP operating as an SPI master device and the other DSP operating as an SPI slave device. For  $\text{CPHA} = 0$ ,  $\overline{\text{HREQ}}$  should be disabled by clearing the  $\text{HRQE}[1:0]$  bits.

### 10.6.3 I<sup>2</sup>C Slave Mode

Initiate I<sup>2</sup>C slave mode by performing the following steps:

1. Set  $\text{HEN} = 1$  to enable the SHI.
2. Set  $\text{HI}^2\text{C} = 1$  to select the I<sup>2</sup>C mode.
3. Set  $\text{HMST} = 0$  to select the slave mode of operation.

In I<sup>2</sup>C slave mode, the contents of the HCKR register are ignored, and the SHI external pins operate as follows:

- $\text{SCK/SCL}$  is the SCL serial clock input.
- $\text{MISO/SDA}$  is the SDA open drain serial data line.
- $\text{MOSI/HA0}$  is the HA0 slave device address input.
- $\overline{\text{SS}}/\text{HA2}$  is the HA2 slave device address input.
- $\overline{\text{HREQ}}$  is the Host Request output.

When the SHI is enabled and configured in I<sup>2</sup>C slave mode, the SHI controller inspects the SDA and SCL lines to detect a start event. After detecting a start event, the SHI receives the slave device address byte and enables the slave device address recognition unit:

- If the slave device address byte was not identified as its personal address, then the SHI controller fails to acknowledge this byte by not driving the SDA line low at the 9th clock pulse ( $\text{ACK} = 1$ ). However, the SHI controller continues to poll the SDA and SCL lines to detect a new start event.
- If the slave device address byte was identified as its personal address (if the personal slave device address was correctly identified), then the slave device address byte is acknowledged ( $\text{ACK} = 0$  is sent), and a receive/transmit session is initiated according to the 8th bit ( $\text{R}/\overline{\text{W}}$ ) of the received slave device address byte.

#### 10.6.3.1 Receive Data in I<sup>2</sup>C Slave Mode

A receive session is initiated when the personal slave device address has been correctly identified and the  $\text{R}/\overline{\text{W}}$  bit of the received slave device address byte has been cleared. Following a receive initiation, data in the SDA line is shifted (MSB first) into the IOSR register. Following each received byte, an acknowledge ( $\text{ACK} = 0$ ) is sent at the 9th clock pulse via the SDA line. Data is acknowledged byte-wise, as required by the I<sup>2</sup>C bus protocol, and is transferred to the HRX FIFO when the complete word (according to the  $\text{HM}[1:0]$  bits) is loaded into the IOSR register. It is the programmer's responsibility to select the correct number of bytes in an I<sup>2</sup>C frame so that they fit into a complete number of words. For this purpose, the slave device address byte does not count as part of the data; the slave device address byte is treated separately.

In a receive session, only the receive path is enabled and HTX-register-to-IOSR-register transfers are inhibited. The HRX FIFO contains valid data, which may be read by the DSP using either DSP instructions or DMA transfers (if the  $\text{HRNE}$  status bit is set).

If the HCKFR bit is cleared, the HRX FIFO is full, and the IOSR register is filled, an overrun error occurs and the HROE status bit is set. In this case, the last received byte is not acknowledged (ACK = 1 is sent) and the word in the IOSR register is not transferred to the HRX FIFO. This informs the external I<sup>2</sup>C master device of an overrun error on the slave side, and upon learning that, the I<sup>2</sup>C master device may terminate this session by generating a stop event.

If the HCKFR bit is set and the HRX FIFO is full, the SHI will hold the clock line to GND, thereby not letting the master device write to the IOSR register, which eliminates the possibility of reaching the overrun condition.

When the IOSR register is ready to receive and the HRX FIFO is not full, the  $\overline{\text{HREQ}}$  output pin (if enabled for receive (HRQE[1:0] = 01) is asserted; this operation guarantees that the next received data word is stored in the HRX FIFO.  $\overline{\text{HREQ}}$  is deasserted at the first clock pulse of the next received word.

The  $\overline{\text{HREQ}}$  line can be used to interrupt the external I<sup>2</sup>C master device. Connecting the  $\overline{\text{HREQ}}$  line between two SHI-equipped DSPs enables full hardware handshaking, with one DSP operating as an I<sup>2</sup>C master device and the other DSP operating as an I<sup>2</sup>C slave device.

### 10.6.3.2 Transmit Data In I<sup>2</sup>C Slave Mode

A transmit session is initiated when the personal slave device address has been correctly identified and the R/ $\overline{\text{W}}$  bit of the received slave device address byte has been set. Following a transmit initiation, the IOSR register is loaded from the HTX register (assuming the HTX register is not empty), and the HTX register's contents are shifted out, MSB first, onto the SDA line. Following each transmitted byte, the SHI controller samples the SDA line at the 9th clock pulse and inspects the ACK status. If the transmitted byte was acknowledged (ACK = 0), then the SHI controller continues and transmits the next byte. However, if the transmitted byte was not acknowledged (ACK = 1), then the transmit session is stopped and the SDA line is released. Consequently, the external master device may generate a stop event to terminate the session.

The HTX register contents are transferred to the IOSR register when the complete word (according to HM[1:0] bits) has been shifted out. It is the programmer's responsibility to select the correct number of bytes in an I<sup>2</sup>C frame, so that the bytes fit into a complete number of words. For this purpose, the slave device address byte does not count as part of the data; the slave device address byte is treated separately.

In a transmit session, only the transmit path is enabled and the IOSR-register-to-HRX FIFO transfers are inhibited. When the HTX register transfers its valid data word to the IOSR register, the HTDE status bit is set and the DSP may write a new data word to the HTX register using either DSP instructions or DMA transfers.

When the master device attempts a transmit session:

- If the HCKFR bit is cleared and if both IOSR and HTX registers are empty, an underrun condition occurs (thereby setting the HTUE status bit), and the previous word is re-transmitted.
- If the HCKFR bit is set and if both IOSR and HTX registers are empty, the SHI holds the clock line to GND to avoid an underrun condition.

When the HTX register is transferred to the IOSR register for transmission, the  $\overline{\text{HREQ}}$  output pin is asserted (if  $\overline{\text{HREQ}}$  is enabled for transmit (HRQE[1:0] = 10)). When asserted,  $\overline{\text{HREQ}}$  indicates that the

slave device is ready to transmit the next data word.  $\overline{\text{HREQ}}$  is deasserted at the first clock pulse of the next transmitted data word.

The  $\overline{\text{HREQ}}$  line can be used to interrupt the external I<sup>2</sup>C master device. Connecting the  $\overline{\text{HREQ}}$  line between two SHI-equipped DSPs enables full hardware handshaking, with one DSP operating as an I<sup>2</sup>C master device and the other DSP operating as an I<sup>2</sup>C slave device.

### 10.6.4 I<sup>2</sup>C Master Mode

Initiate I<sup>2</sup>C master mode by performing the following steps:

1. Set HEN = 1 to enable the SHI.
2. Set HI<sup>2</sup>C = 1 to select the I<sup>2</sup>C mode.
3. Set HMST = 1 to select the master mode of operation.

Before enabling the SHI as an I<sup>2</sup>C master, the programmer should program the appropriate clock rate in HCKR. When configured in the I<sup>2</sup>C master mode, the SHI external pins operate as follows:

- SCK/SCL is the SCL open drain serial clock output.
- MISO/SDA is the SDA open drain serial data line.
- MOSI/HA0 is the HA0 slave device address input.
- $\overline{\text{SS}}$ /HA2 is the HA2 slave device address input.
- $\overline{\text{HREQ}}$  is the Host Request input.

In the I<sup>2</sup>C master mode, a data transfer session is always initiated by the DSP by writing to the HTX register when the HIDLE bit is set. This condition ensures that the data byte written to the HTX register is interpreted as being a slave address byte. This data byte must specify the slave device address to be selected and the requested data transfer direction.

#### NOTE

The slave address byte should be located in the high byte of the data word, because the middle and low bytes of the data word are ignored. Only one byte (the slave address byte) is shifted out, independent of the word length defined by the HM[1:0] bits.

To the DSP to initiate a data transfer, the following actions should be performed:

- The DSP tests the HIDLE status bit.
- If the HIDLE status bit is set, the DSP writes the slave device address and the R/ $\overline{\text{W}}$  bit to the most significant byte of the HTX register.
- The SHI generates a start event.
- The SHI transmits one byte only, internally samples the R/ $\overline{\text{W}}$  direction bit (last bit) and accordingly initiates a receive or transmit session.
- The SHI inspects the SDA level at the 9th clock pulse to determine the ACK value:
  - If acknowledged (ACK = 0), the SHI starts its receive or transmit session according to the sampled R/ $\overline{\text{W}}$  value.

- If not acknowledged ( $ACK = 1$ ), the HBER status bit in the HCSR register is set, which causes an SHI Bus Error interrupt request (if the HBIE bit is set), and a stop event is generated.

If the HRQE[1:0] bits are cleared, then the I<sup>2</sup>C master device ignores the  $\overline{HREQ}$  input pin. If either of the HRQE[1:0] bits are set, then the I<sup>2</sup>C master device considers the  $\overline{HREQ}$  input pin.

When asserted,  $\overline{HREQ}$  indicates that the external slave device is ready for the next data transfer. As a result, the I<sup>2</sup>C master device sends clock pulses for the full data word transfer. At the first clock pulse of the next data transfer, the external slave device deasserts  $\overline{HREQ}$ . When deasserted,  $\overline{HREQ}$  prevents the clock generation of the next data word transfer, until  $\overline{HREQ}$  is asserted again.

Connecting the  $\overline{HREQ}$  line between two SHI-equipped DSPs enables full hardware handshaking, with one DSP operating as an I<sup>2</sup>C master device and the other DSP operating as an I<sup>2</sup>C slave device.

#### 10.6.4.1 Receive Data in I<sup>2</sup>C Master Mode

A receive session is initiated if the R/ $\overline{W}$  direction bit of the transmitted slave device address byte is set. Following a receive initiation, data in the SDA line is shifted (MSB first) into the IOSR register. Following each received byte, an acknowledge ( $ACK = 0$ ) is sent at the 9th clock pulse via the SDA line, if the HIDL control bit is cleared. Data is acknowledged byte-wise, as required by the I<sup>2</sup>C bus protocol, and is transferred to the HRX FIFO when the complete word (according to HM[1:0] bits) is loaded into the IOSR register. It is the programmer's responsibility to select the correct number of bytes in an I<sup>2</sup>C frame so that the bytes fit in a complete number of words. For this purpose, the slave device address byte does not count as part of the data; the slave device address byte is treated separately.

If the I<sup>2</sup>C slave transmitter is acknowledged, it should transmit the next data byte. To terminate the receive session, the programmer should set the HIDL bit at the last required data word. As a result, the last byte of the next received data word is not acknowledged, the slave transmitter releases the SDA line, and the SHI generates the stop event and terminates the session.

In a receive session, only the receive path is enabled and the HTX-register-to-IOSR-register transfers are inhibited. If the HRNE status bit is set, the HRX FIFO contains valid data, which may be read by the DSP using either DSP instructions or DMA transfers. When the HRX FIFO is full, the SHI suspends the serial clock prior to acknowledge. In this case, the clock is re-activated when the FIFO is read (the SHI gives an  $ACK = 0$  and starts receiving).

#### 10.6.4.2 Transmit Data In I<sup>2</sup>C Master Mode

A transmit session is initiated if the R/ $\overline{W}$  direction bit of the transmitted slave device address byte is cleared. Following a transmit initiation, the IOSR register is loaded from the HTX register (assuming that the HTX register is not empty), and the HTX register's contents are shifted out (MSB-first) onto the SDA line. Following each transmitted byte, the SHI controller samples the SDA line at the 9th clock pulse and inspects the ACK status:

- If the transmitted byte was acknowledged ( $ACK = 0$ ), the SHI controller continues transmitting the next byte.
- If the transmitted byte was not acknowledged ( $ACK = 1$ ), the HBER status bit is set to inform the DSP side that a bus error (or overrun, or any other exception in the slave device) has occurred. Consequently, the I<sup>2</sup>C master device generates a stop event and terminates the session.



The HTX register contents are transferred to the IOSR register when the complete word (according to HM[1:0] bits) has been shifted out. It is the programmer's responsibility to select the right number of bytes in an I<sup>2</sup>C frame so that the bytes fit in a complete number of words. Remember that for this purpose, the slave device address byte does not count as part of the data.

In a transmit session, only the transmit path is enabled and the IOSR-register-to-HRX FIFO transfers are inhibited. When the HTX register transfers its valid data word to the IOSR register, the HTDE status bit is set and the DSP may write a new data word to the HTX register using either DSP instructions or DMA transfers. If both IOSR and HTX registers are empty, the SHI suspends the serial clock until new data is written into the HTX register (when the SHI proceeds with the transmit session) or until the HIDL bit is set (the SHI re-activates the clock to generate the stop event and terminate the transmit session).

### 10.6.5 SHI Operation During DSP Stop

The SHI operation cannot continue when the DSP is in the stop state, because no DSP clocks are active. While the DSP is in the stop state, the SHI remains in the individual reset state.

While in the individual reset state the following is true:

- If the SHI was operating in I<sup>2</sup>C mode, the SHI signals are disabled (high impedance state).
- If the SHI was operating in SPI mode, the SHI signals are not affected.
- The HCSR status bits and the transmit/receive paths are reset to the same state produced by hardware or software resets.
- The HCSR and HCKR control bits are not affected.

#### NOTE

It is recommended that you disable the SHI before entering the stop state.

## 10.7 SHI Pin-Outs for Device Packages

### 10.7.1 SHI Pin-Outs for Small Pin Count Packages

The DSP56725 80-pin and DSP56724 144-pin packages still allow both DSP cores to be accessible to an external microcontroller.

Figure 10-12 shows the SHI/SHI\_1 pin-outs in small pin count packages.

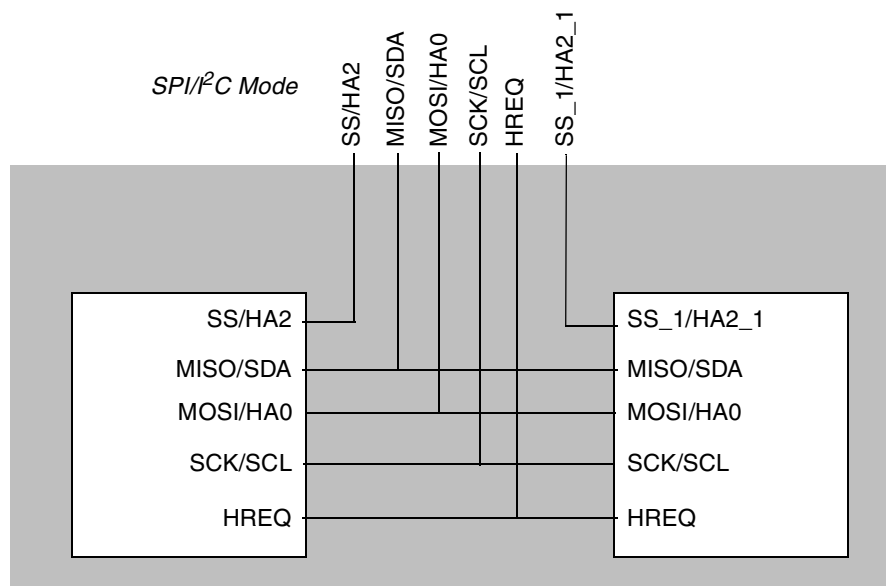


Figure 10-12. SHI/SHI\_1 Pin-Outs in Small Pin Count Packages

In the DSP56725 80-pin and DSP56724 144-pin packages, all the pins of both SHI and SHI\_1 are multiplexed, except for two signals:  $\overline{SS}$ /HA2 and  $\overline{SS}_1$ /HA2\_1. This multiplexing of pins sets some limits on operations:

- Both SHI and SHI\_1 are in I<sup>2</sup>C slave mode together.  $\overline{SS}$ /HA2 is the HA2 slave device address input of SHI;  $\overline{SS}_1$ /HA2\_1 is the device address input of SHI\_1. To obtain the slave address, the following bits are combined and formed into HA[6:0]: the HA[6:3], HA1 bits in the SHI/SHI\_1 configuration registers, and the HA0 and HA2 pins. The SHI can acknowledge the read/write request when the request slave device address is the same as its own address. So in small packages, the two SHIs can be configured in I<sup>2</sup>C slave mode together and set to different addresses by the setting the HA2/HA2\_1 pin.
- Otherwise, if one of the SHI/SHI\_1 blocks is in I<sup>2</sup>C slave mode, the other block must be in I<sup>2</sup>C master mode or disabled.
- If one of the SHI/SHI\_1 blocks is in I<sup>2</sup>C master mode, the other block must be disabled or in I<sup>2</sup>C slave mode.
- If both cores are in SPI slave mode:  
When SHI(SHI\_1) is in SPI slave mode,  $\overline{SS}$ /HA2( $\overline{SS}_1$ /HA2\_1) is the  $\overline{SS}$ ( $\overline{SS}_1$ ) input. In SPI slave mode, the  $\overline{SS}$  line should be kept asserted (low) during a data word transfer. If the  $\overline{SS}$  line is deasserted (high) before the end of the data word transfer, the transfer is aborted and the received data word is lost. So in small packages, the two SHI can be configured in SPI slave mode, and can transfer data (at different times) by setting the  $\overline{SS}$  or  $\overline{SS}_1$  pins active at different times. These pins can never be active simultaneously or contention occurs.
- Otherwise, if one of the SHI/SHI\_1 blocks is in SPI slave mode, then the other block must be disabled.

- If one of the SHI/SHI\_1 blocks is in SPI master mode, then the other block must be disabled. Otherwise, if the other SHI/SHI\_1 block is enabled, then the multiplexed pins conflict with each other.
- When both SHI and SHI\_1 are not in SPI mode,  $\overline{\text{HREQ}}$  can act as GPIO port H function or GPIO port H1 function, but only one of the SHI/SHI\_1 blocks at a time can control the output function of the  $\overline{\text{HREQ}}$  pin.

#### NOTE

Because both  $\overline{\text{HREQ}}$  and  $\overline{\text{HREQ}_1}$  are function-multiplexed with GPIO port H functions, write the appropriate, non-contending values to the GPIO control registers *before* enabling the SHI function.

These constraints on SHI operations also apply to the boot modes of each core. Care must be taken when selecting boot modes.



# Chapter 11

## Triple Timer Module (TEC, TEC\_1)

### 11.1 Introduction

In the DSP56724/DSP56725, there are two internal triple timer modules. Core-0 uses TEC and Core-1 uses TEC\_1. In this chapter, only the TEC block is described in detail.

#### 11.1.1 Overview

The timer module contains a common 21-bit prescaler and three independent and identical general-purpose, 24-bit timer/event counters, each with its own register set. Each of the timers has the following capabilities:

- Uses internal clocking.
- Interrupts the DSP Core after a specified number of events (clocks).
- Triggers DMA transfers after a specified number of events (clocks) occurs.

### 11.1.2 Triple Timer Module Block Diagram

Figure 11-1 shows a block diagram of the triple timer module. This module includes a 24-bit Timer Prescaler Load Register (TPLR), a 24-bit Timer Prescaler Count Register (TPCR), and three timers. Each timer can use the prescaler clock as its clock source.

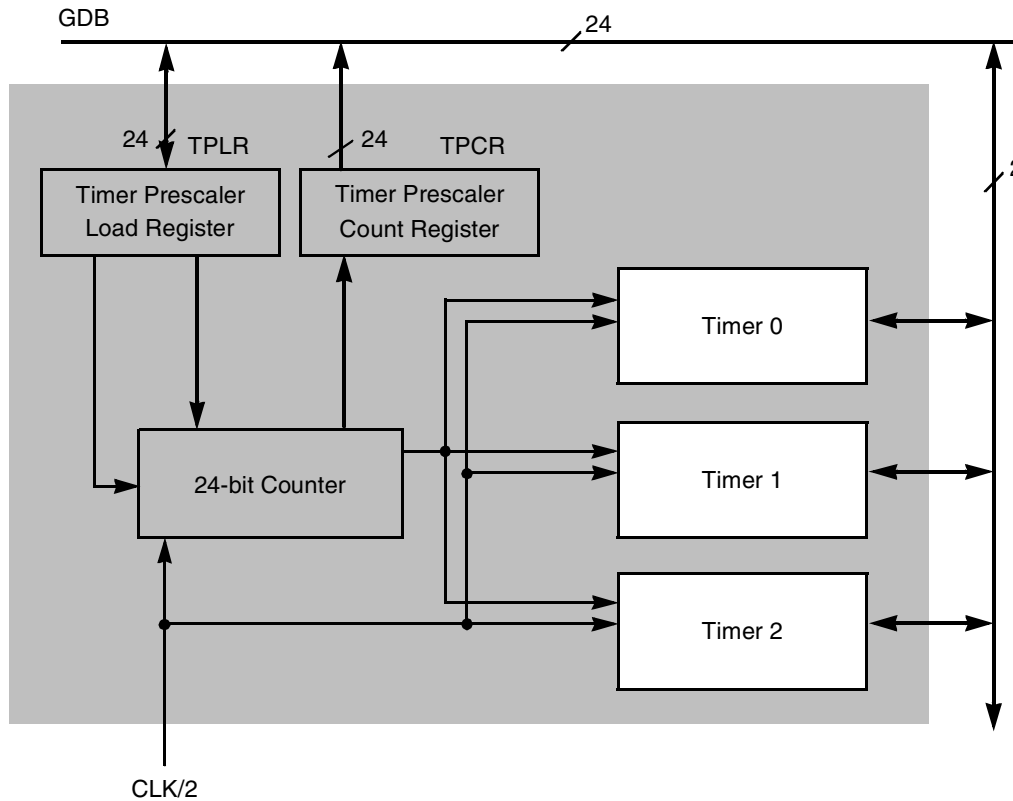


Figure 11-1. Triple Timer Block Diagram

### 11.2 Individual Timer Block Diagram

Figure 11-2 shows the structure of an individual timer block. The DSP56724/DSP56725 treats each timer as a memory-mapped peripheral with four registers occupying four 24-bit words in the X data memory space. The three timers are identical in structure and function. Either standard polled or interrupt programming techniques can be used to service the timers. A single, generic timer is discussed in this chapter. Each timer includes the following:

- 24-bit counter
- 24-bit read/write Timer Control and Status Register (TCSR)
- 24-bit read-only Timer Count Register (TCR)
- 24-bit write-only Timer Load Register (TLR)
- 24-bit read/write Timer Compare Register (TCPR)
- Logic for clock selection and interrupt/DMA trigger generation.

The timer mode is controlled by the TC[3–0] bits which are TCSR[7–4]. For a description of timer mode operations, see [Section 11.4, “Operating Mode.”](#)

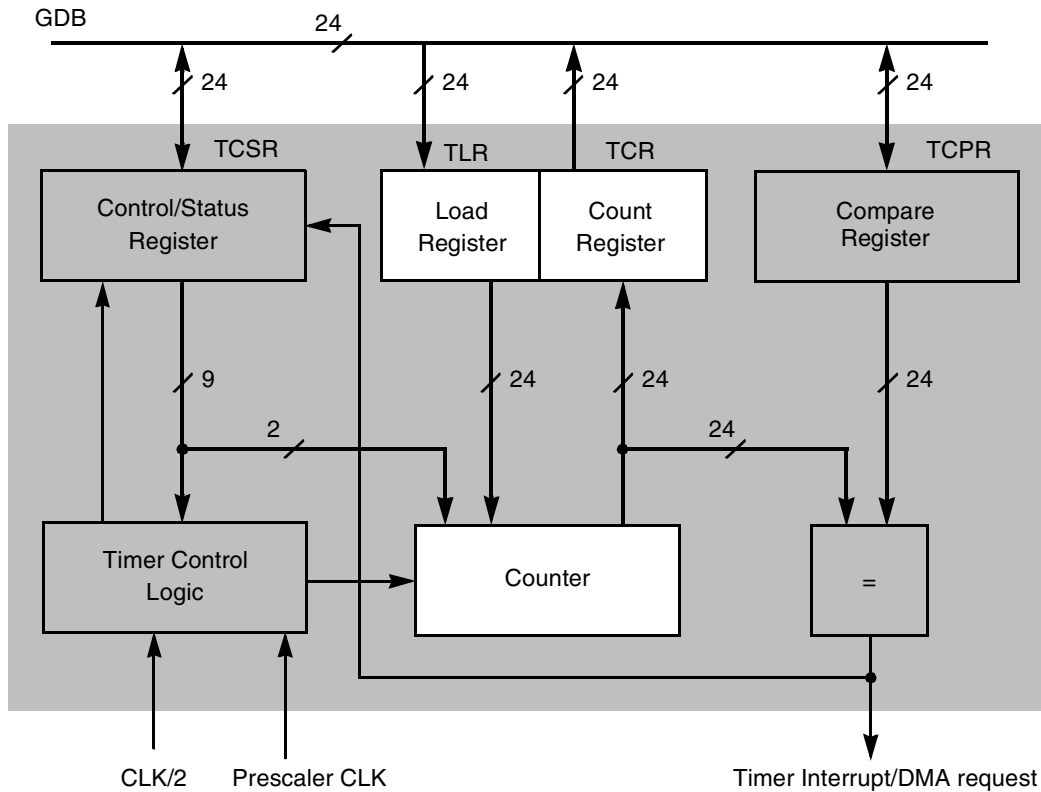


Figure 11-2. Individual Timer Block Diagram

## 11.3 Operation

This section discusses the following timer basics:

- Reset
- Initialization
- Exceptions

### 11.3.1 Timer After Reset

A hardware  $\overline{\text{RESET}}$  signal or software RESET instruction clears the Timer Control and Status Register for each timer. A timer is active only if the timer enable bit 0 (TCSR[TE]) in the specific timer TCSR register is set.

## 11.3.2 Timer Initialization

To initialize a timer, do the following:

1. Ensure that the timer is not active, either by sending a reset or by clearing the TCSR[TE] bit.
2. Configure the control register (TCSR) to set the timer operating mode. Set the interrupt enable bits as desired.
3. Configure the other registers: Timer Prescaler Load Register (TPLR), Timer Load Register (TLR), and Timer Compare Register (TCPR).
4. Enable the timer by setting the TCSR[TE] bit.

## 11.3.3 Timer Exceptions

Each timer can generate two different exceptions:

- **Timer Overflow (highest priority)** — Occurs when the timer counter reaches the overflow value. This exception sets the TOF bit. The TOF bit is cleared when “1” is written to it or when the timer overflow exception is serviced.
- **Timer Compare (lowest priority)** — Occurs when the timer counter reaches the value given in the Timer Compare Register (TCPR), for all modes except measurement modes. The Compare exception sets the TCF bit. The TCF bit is cleared when “1” is written to it or when the timer compare interrupt is serviced.

To configure a timer exception, perform the following steps. The text to the right of each step shows the register settings for configuring a Timer 0 compare interrupt. The order of the steps is optional except that the timer should not be enabled (step 2e) until all other exception configuration is complete:

1. Configure the interrupt service routine (ISR):
  - a) Load vector base address register. VBA (b23-8)
  - b) Define I\_VEC to be equal to the VBA value (if that is nonzero). If it is defined, I\_VEC must be defined for the assembler before the interrupt equate file is included.
  - c) Load the exception vector table entry: two-word fast interrupt, or jump/branch to subroutine (long interrupt). p:TIM0C
2. Configure the interrupt trigger:
  - a) Enable and prioritize overall peripheral interrupt functionality. IPRP (TOL[1-0])
  - b) Enable a specific peripheral interrupt. TCSR0 (TCIE)
  - c) Unmask interrupts at the global level. SR (I[1-0])
  - d) Configure a peripheral interrupt-generating function. TCSR0 (TC[7-4])
  - e) Enable peripheral and associated signals. TCSR0 (TE)



## 11.4 Operating Mode

The TEC timer has one mode, GPIO, mode 0: Internal timer interrupt generated by the internal clock.

- The TCSR[TE] bit is set to clear the counter and enable the timer. Clearing TCSR[TE] disables the timer.
- The value to which the timer is to count is loaded into the TCPR register. (This is true for all modes except the measurement modes (modes 4 through 6).)
- The counter is loaded with the TLR value on the first clock.
- If the counter overflows, TCSR[TOF] is set, and if TCSR[TOIE] is set, an overflow interrupt is generated.
- You can read the counter contents at any time from the Timer Count Register (TCR).

### NOTE

To ensure proper operation, the TCSR TC[3–0] bits should be changed only when the timer is disabled (that is, when TCSR[TE] is cleared).

### 11.4.1 Timer GPIO (Mode 0)

Table 11-1. Timer GPIO (Mode 0)

Bit Settings				Mode Characteristics			
TC3	TC2	TC1	TC0	Mode	Name	Function	Clock
0	0	0	0	0	GPIO	Timer	Internal

In Mode 0, the timer generates an internal interrupt when a counter value is reached, if the timer compare interrupt is enabled (see [Figure 11-3](#). and [Figure 11-4](#).). When the counter equals the TCPR register value, TCSR[TCF] is set and a compare interrupt is generated (if the TCSR[TCIE] bit is set). If the TCSR[TRM] bit is set, the counter is reloaded with the TLR register value at the next timer clock and the count is resumed. If TCSR[TRM] is cleared, the counter continues to increment on each timer clock signal. This process repeats until the timer is disabled.

**Mode 0 (Internal clock, no timer output): TRM = 1**

N = write preload  
M = write compare

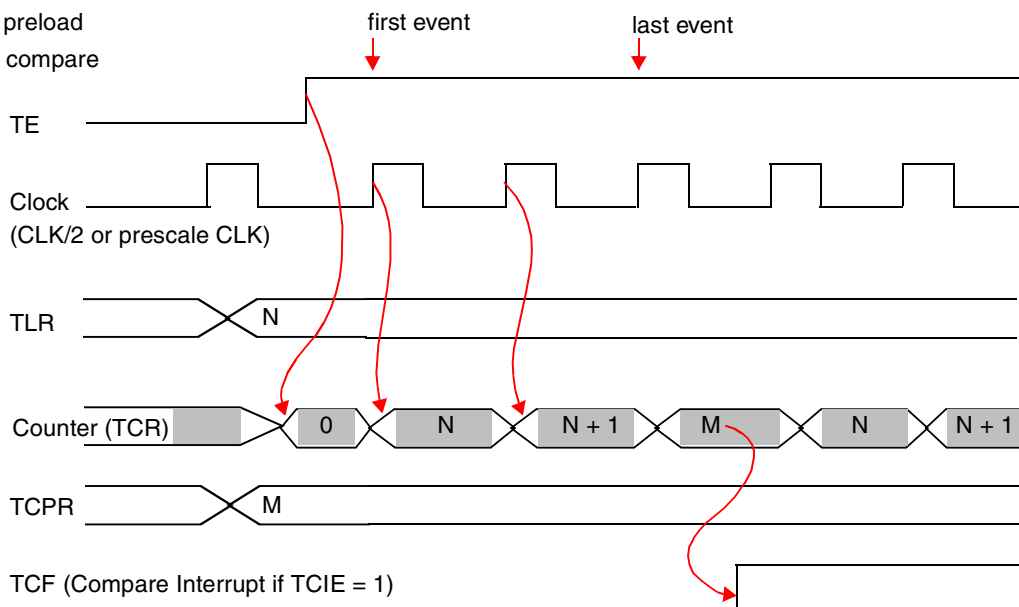


Figure 11-3. Timer Mode (TRM = 1)

**Mode 0 (Internal clock, no timer output): TRM = 0**

N = write preload  
M = write compare

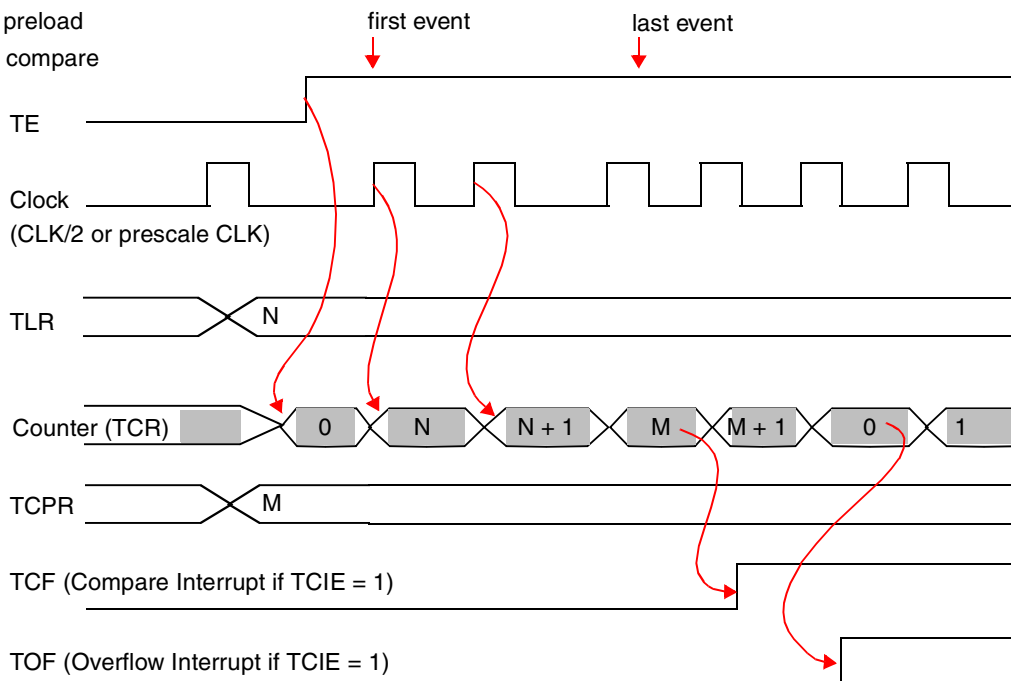


Figure 11-4. Timer Mode (TRM = 0)

## 11.4.2 Reserved Modes

Modes 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15 are reserved.

## 11.4.3 Special Cases

The following special cases apply during wait and stop state.

- Timer behavior during wait—Timer clocks are active during the execution of the WAIT instruction and timer activity is undisturbed. If a timer interrupt is generated, the DSP56725 leaves the wait state and services the interrupt.
- Timer behavior during stop—During execution of the STOP instruction, the timer clocks are disabled, timer activity stops, and the TIO signals are disconnected. Any external changes that happen to the TIO signals are ignored when the corresponding DSP Core is in stop state. To ensure correct operation, disable the timers before the corresponding DSP Core is placed into stop state.

## 11.4.4 DMA Trigger

Each timer can also trigger DMA transfers if a DMA channel is programmed to be triggered by a timer event. The timer issues a DMA trigger on every event in all modes of operation. To ensure that all DMA triggers are serviced, provide for the preceding DMA trigger to be serviced before the DMA channel receives the next trigger.

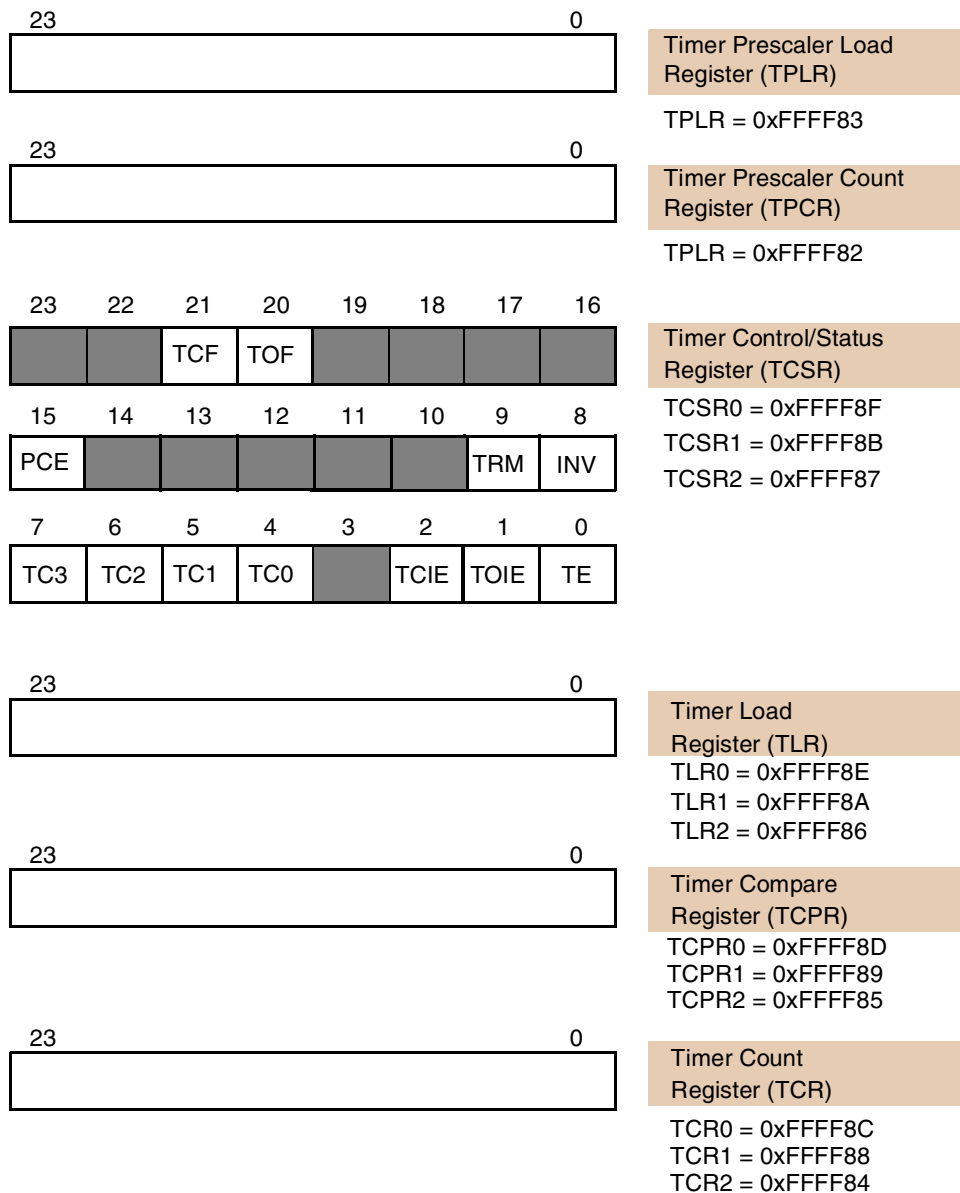
## 11.5 Triple Timer Module Programming Model

The timer programmer's model in [Figure 11-5](#) shows the structure of the timer registers.

### 11.5.1 Prescaler Counter

The prescaler counter is a 21-bit counter that decrements on the rising edge of the prescaler input clock. The counter is enabled when at least one of the three timers is enabled (that is, one or more of the timer enable bits are set) and is using the prescaler output as its source (that is, one or more of the PCE bits are set).

### Triple Timer Module (TEC, TEC\_1)



Reserved bit. Read as 0. Write with 0 for future compatibility

**Figure 11-5. Timer Module Programmer's Model**

## 11.5.2 Timer Prescaler Load Register (TPLR)

The TPLR is a read/write register that controls the prescaler divide factor (that is, the number that the prescaler counter loads and begins counting from) and the source for the prescaler input clock, shown in Figure 11-6 and Table 11-2.

Access: Read/Write

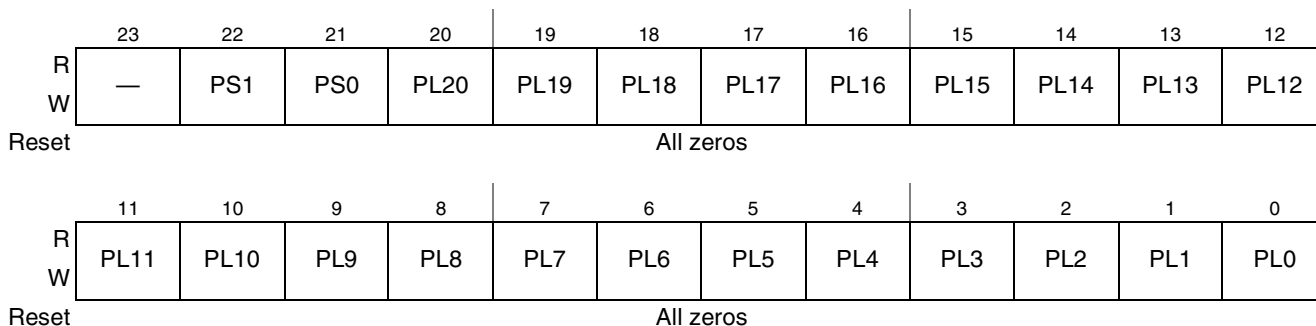


Figure 11-6. Timer Prescaler Load Register

Table 11-2. Timer Prescaler Load Register (TPLR) Bit Definitions

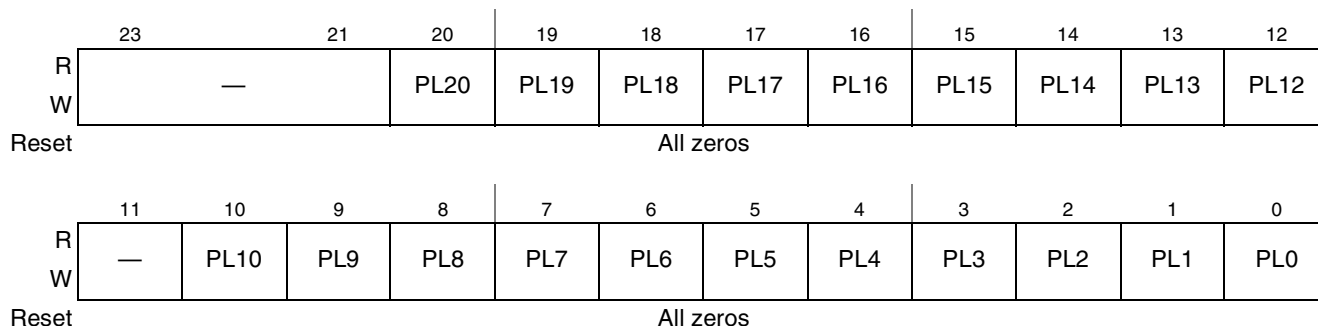
Bit	Name	Description															
23	—	Reserved. Write to zero for future compatibility.															
22–21	PS[1–0]	<p><b>Prescaler Source</b>                      Controls the source of the prescaler clock. The external clock is internally synchronized to the internal clock. The external clock frequency must be lower than the DSP56725 internal operating frequency divided by 4 (that is, CLK/4).                      NOTE: To ensure proper operation, change the PS[1–0] bits only when the prescaler counter is disabled. Disable the prescaler counter by clearing TCSR[TE] of each of three timers.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">PS1</th> <th style="width: 20%;">PS0</th> <th style="width: 60%;">Prescaler Clock Source</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Internal CLK/2</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Reserved</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Reserved</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Reserved</td> </tr> </tbody> </table>	PS1	PS0	Prescaler Clock Source	0	0	Internal CLK/2	0	1	Reserved	1	0	Reserved	1	1	Reserved
PS1	PS0	Prescaler Clock Source															
0	0	Internal CLK/2															
0	1	Reserved															
1	0	Reserved															
1	1	Reserved															
20–0	PL[20–0]	<p><b>Prescaler Preload Value</b>                      Contains the prescaler preload value, which is loaded into the prescaler counter when the counter value reaches 0, or when the counter switches state from disabled to enabled. If PL[20–0] = N, then the prescaler counts N+1 source clock cycles before generating a prescaler clock pulse. Therefore, the prescaler divide factor = (preload value) + 1.</p>															

### 11.5.3 Timer Prescaler Count Register (TPCR)

The TPCR is a read-only register that reflects the current value in the prescaler counter, shown in [Figure 11-7](#) and [Table 11-3](#).

Address

Access: Read/Write



**Figure 11-7. Timer Prescaler Count Register (TPCR)**

**Table 11-3. Timer Prescaler Count Register (TPCR) Bit Definitions**

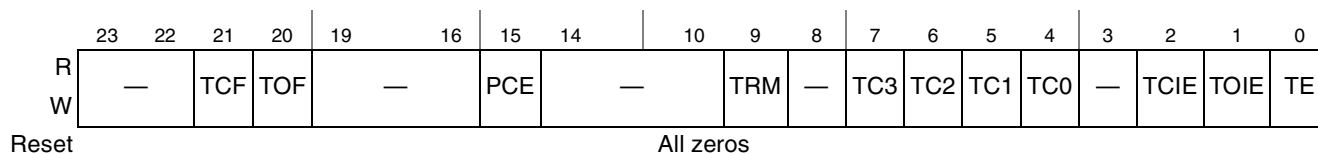
Bit	Name	Description
23–21	—	<i>Reserved.</i> Write to zero for future compatibility.
20–0	PC[20–0]	<b>Prescaler Counter Value</b> Contain the current value of the prescaler counter.

### 11.5.4 Timer Control/Status Register (TCSR)

The TCSR is a read/write register controlling the timer and reflecting its status, shown in [Figure 11-8](#) and [Table 11-4](#).

Address

Access: Read/Write



**Figure 11-8. Timer Control/Status Register (TCSR)**

**Table 11-4. Timer Control/Status Register (TCSR) Bit Definitions**

Bit	Name	Description
23–22	—	<i>Reserved. Write to zero for future compatibility.</i>
21	TCF	<p>Timer Compare Flag</p> <p>Indicate that the event count is complete.</p> <p>In timer modes, the TCF bit is set after <math>(M - N + 1)</math> events are counted. (<math>M</math> is the value in the compare register and <math>N</math> is the TLR value.) In measurement modes, the TCF bit is set when the measurement completes.</p> <p>Writing a one to the TCF bit clears it. A zero written to the TCF bit has no effect. The bit is also cleared when the timer compare interrupt is serviced. The TCF bit is cleared by a hardware <math>\overline{\text{RESET}}</math> signal, a software RESET instruction, the STOP instruction, or by clearing the TCSR[TE] bit to disable the timer.</p> <p>NOTE: The TOF and TCF bits are cleared by a 1 written to the specific bit. To ensure that only the target bit is cleared, do not use the BSET command. The proper way to clear these bits is to write 1 (using a MOVEP instruction) to the flag to be cleared and to write 0 to the other flag.</p>
20	TOF	<p>Timer Overflow Flag</p> <p>Indicates that a counter overflow has occurred. This bit is cleared by writing a one to the TOF bit. Writing a zero to the TOF bit has no effect. The TOF bit is also cleared when the timer overflow interrupt is serviced.</p> <p>The TOF bit is cleared by a hardware <math>\overline{\text{RESET}}</math> signal, a software RESET instruction, the STOP instruction, or by clearing the TCSR[TE] bit to disable the timer.</p>
19–16	—	<i>Reserved. Write to zero for future compatibility.</i>
15	PCE	<p>Prescaler Clock Enable</p> <p>Selects the prescaler clock as the timer source clock. When the PCE bit is cleared, the timer uses either an internal (CLK/2) signal or an external (TIO) signal as its source clock. When the PCE bit is set, the prescaler output is the timer source clock for the counter, regardless of the timer operating mode. To ensure proper operation, the PCE bit is changed only when the timer is disabled. The PS[1–0] bits of the TPLR determine which source clock is used for the prescaler. A timer can be clocked by a prescaler clock that is derived from the TIO of another timer.</p>
14–10	—	<i>Reserved. Write to zero for future compatibility.</i>
9	TRM	<p>Timer Reload Mode</p> <p>Controls the counter preload operation. In timer (0–3) and watchdog (9–10) modes, the counter is preloaded with the TLR register value after the TCSR[TE] bit is set and the first internal or external clock signal is received. If the TRM bit is set, the counter is reloaded each time after it reaches the value contained by the TCR register. In PWM mode (7), the counter is reloaded each time counter overflow occurs. In measurement (4–5) modes, if the TRM and the TCSR[TE] bits are set, the counter is preloaded with the TLR register value on each appropriate edge of the input signal. If the TRM bit is cleared, the counter operates as a free running counter and is incremented on each incoming event.</p>
8	—	<i>Reserved. Write to zero for future compatibility.</i>

**Table 11-4. Timer Control/Status Register (TCSR) Bit Definitions (Continued)**

Bit	Name	Description																																																																																																																																																	
7-4	TC[3-0]	<p>Timer Control</p> <p>Control the source of the timer clock and the Timer mode of operation. <a href="#">Section 11.4, “Operating Mode,”</a> describes the timer operating modes in detail.</p> <p><b>Note:</b> To ensure proper operation, the TC[3-0] bits should be changed only when the timer is disabled (that is, when the TCSR[TE] bit is cleared).</p>																																																																																																																																																	
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4">Bit Settings</th> <th colspan="4">Mode Characteristics</th> </tr> <tr> <th>TC3</th> <th>TC2</th> <th>TC1</th> <th>TC0</th> <th>Mode Number</th> <th>Mode Function</th> <th>TIO</th> <th>Clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Timer and GPIO</td> <td>GPIO<sup>1</sup></td> <td>Internal</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>2</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>3</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>4</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>5</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>6</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>7</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>8</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>9</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>10</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>11</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>12</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>13</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>14</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>15</td> <td><i>Reserved</i></td> <td>—</td> <td>—</td> </tr> </tbody> </table>		Bit Settings				Mode Characteristics				TC3	TC2	TC1	TC0	Mode Number	Mode Function	TIO	Clock	0	0	0	0	0	Timer and GPIO	GPIO <sup>1</sup>	Internal	0	0	0	1	1	<i>Reserved</i>	—	—	0	0	1	0	2	<i>Reserved</i>	—	—	0	0	1	1	3	<i>Reserved</i>	—	—	0	1	0	0	4	<i>Reserved</i>	—	—	0	1	0	1	5	<i>Reserved</i>	—	—	0	1	1	0	6	<i>Reserved</i>	—	—	0	1	1	1	7	<i>Reserved</i>	—	—	1	0	0	0	8	<i>Reserved</i>	—	—	1	0	0	1	9	<i>Reserved</i>	—	—	1	0	1	0	10	<i>Reserved</i>	—	—	1	0	1	1	11	<i>Reserved</i>	—	—	1	1	0	0	12	<i>Reserved</i>	—	—	1	1	0	1	13	<i>Reserved</i>	—	—	1	1	1	0	14	<i>Reserved</i>	—	—	1	1	1	1	15	<i>Reserved</i>	—	—
		Bit Settings				Mode Characteristics																																																																																																																																													
		TC3	TC2	TC1	TC0	Mode Number	Mode Function	TIO	Clock																																																																																																																																										
		0	0	0	0	0	Timer and GPIO	GPIO <sup>1</sup>	Internal																																																																																																																																										
		0	0	0	1	1	<i>Reserved</i>	—	—																																																																																																																																										
		0	0	1	0	2	<i>Reserved</i>	—	—																																																																																																																																										
		0	0	1	1	3	<i>Reserved</i>	—	—																																																																																																																																										
		0	1	0	0	4	<i>Reserved</i>	—	—																																																																																																																																										
		0	1	0	1	5	<i>Reserved</i>	—	—																																																																																																																																										
		0	1	1	0	6	<i>Reserved</i>	—	—																																																																																																																																										
		0	1	1	1	7	<i>Reserved</i>	—	—																																																																																																																																										
		1	0	0	0	8	<i>Reserved</i>	—	—																																																																																																																																										
		1	0	0	1	9	<i>Reserved</i>	—	—																																																																																																																																										
		1	0	1	0	10	<i>Reserved</i>	—	—																																																																																																																																										
		1	0	1	1	11	<i>Reserved</i>	—	—																																																																																																																																										
		1	1	0	0	12	<i>Reserved</i>	—	—																																																																																																																																										
1	1	0	1	13	<i>Reserved</i>	—	—																																																																																																																																												
1	1	1	0	14	<i>Reserved</i>	—	—																																																																																																																																												
1	1	1	1	15	<i>Reserved</i>	—	—																																																																																																																																												
<p><b>Note:</b> The GPIO function is enabled only if all of the TC[3-0] bits are 0.</p>																																																																																																																																																			
3	—	<i>Reserved.</i> Write to zero for future compatibility.																																																																																																																																																	
2	TCIE	<p>Timer Compare Interrupt Enable</p> <p>Enables/disables the timer compare interrupts. When set, the TCIE bit enables the compare interrupts. In the timer mode, a compare interrupt is generated after the counter value matches the value of the TCPR register. The counter starts counting up from the number loaded from the TLR register, and if the TCPR register value is M, an interrupt occurs after (M – N + 1) events, where N is the value of the TLR register. When cleared, the TCSR[TCIE] bit disables the compare interrupts.</p>																																																																																																																																																	



**Table 11-4. Timer Control/Status Register (TCSR) Bit Definitions (Continued)**

Bit	Name	Description
1	TOIE	Timer Overflow Interrupt Enable Enables timer overflow interrupts. When set, the TOIE bit enables overflow interrupt generation. The timer counter can hold a maximum value of 0xFFFFF. When the counter value is at the maximum value and a new event causes the counter to be incremented to 0x000000, the timer generates an overflow interrupt. When cleared, the TOIE bit disables overflow interrupt generation.
0	TE	Timer Enable Enables/disables the timer. When set, the TE bit enables the timer and clears the timer counter. The counter starts counting according to the mode selected by the timer control (TC[3–0]) bit values. When clear, the TE bit disables the timer.

### 11.5.5 Timer Load Register (TLR)

The TLR is a 24-bit write-only register. In all modes, the counter is preloaded with the TLR value after the TCSR[TE] bit is set and a first event occurs.

- In timer modes, if the TCSR[TRM] bit is set, the counter is reloaded each time after it reaches the value contained by the timer compare register (TCPR) and the new event occurs.
- In measurement modes, if TCSR[TRM] and TCSR[TE] are set, the counter is reloaded with the value in the TLR register on each appropriate edge of the input signal.
- In PWM modes, if TCSR[TRM] is set, the counter is reloaded each time after it overflows and the new event occurs.
- In watchdog modes, if TCSR[TRM] is set, the counter is reloaded each time after it reaches the value contained by the timer compare register (TCPR) and the new event occurs. In this mode, the counter is also reloaded whenever the TLR register is written with a new value while TCSR[TE] is set.
- In all modes, if TCSR[TRM] is cleared (TRM = 0), the counter operates as a free-running counter.

### 11.5.6 Timer Compare Register (TCPR)

The TCPR is a 24-bit read/write register that contains the value to be compared to the counter value. These two values are compared every timer clock after TCSR[TE] is set. When the values match, the timer compare flag bit is set and an interrupt is generated *if* interrupts are enabled (that is, the timer compare interrupt enable bit in the TCSR register is set). The TCPR register is ignored in measurement modes.

### 11.5.7 Timer Count Register (TCR)

The TCR is a 24-bit read-only register. In timer mode, the contents of the counter can be read at any time from the TCR register.



# Chapter 12

## Watchdog Timer (WDT, WDT\_1)

### 12.1 Introduction

The watchdog timer (WDT, WDT\_1) is a 16-bit timer used to help software recover from runaway code. The timer is a free-running down-counter used to assert the WDT pin on underflow. Software must periodically service the watchdog timer to restart the count down and prevent assertion of the WDT pin. [Figure 12-1](#) shows the watchdog timer block diagram.

The DSP56724/DSP56725 has two WDT modules: WDT, WDT\_1. The only differences between the WDT and WDT\_1 modules is that WDT is used by DSP Core-0 and WDT\_1 is used by DSP Core-1. Only the WDT is described in detail in this chapter.

### 12.2 WDT Pin-Outs for Different Device Packages

The watchdog timer has a dedicated output pin (WDT). WDT is held high during watchdog timer operation until the watchdog timer times out. When the watchdog timer times out, the WDT pin is asserted low after two EXTAL system clock cycle delays. Following a reset of the device, the WDT pin will de-assert high after two EXTAL system clock cycle delays.

In the DSP56725 (80-pin package) and the DSP56724 (144-pin package), the WDT and WDT\_1 pins are 'ORed' together, so that when either watchdog timer times out, the external pin is asserted.

The watchdog timer is driven by the DSP's main system clock (Fsys). Fsys is scaled by a fixed prescaler (divide by 4096) prior to driving the 16-bit counter. The time-out period can be selected by writing to the watchdog modulus register (WMR).

Time-out period =  $4096 \times (\text{WMR} + 1)$  clocks

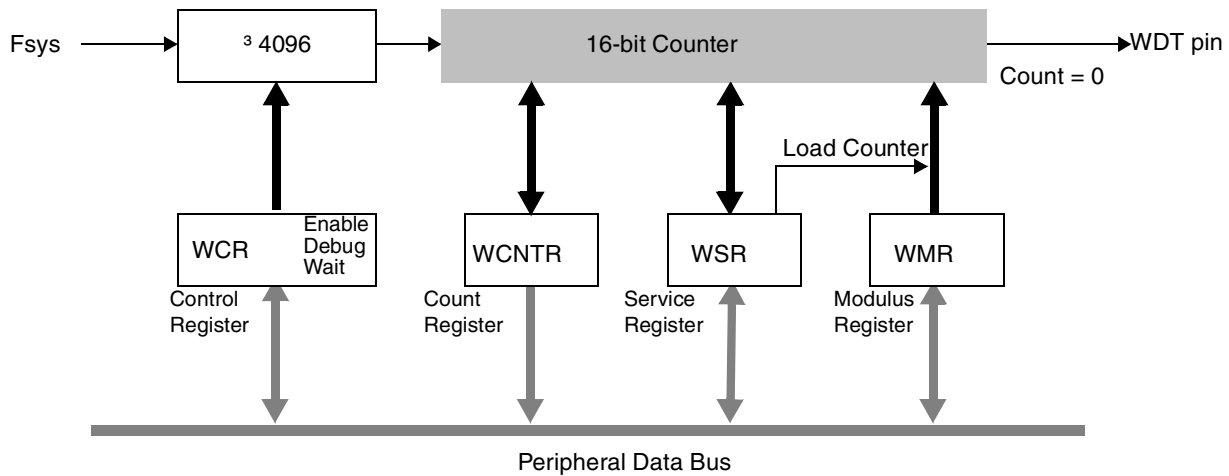
- *Example 1:* For Fsys = 200 MHz,  
 Time-out period =  $4096 \times (0x00FFFF+1) = 268,435,456$  clocks  
 Countdown time =  $(268,435,456 \text{ clocks} / 200,000,000 \text{ clocks per second})$   
 Countdown time = 1.3422 seconds
- *Example 2:* For Fsys = 150 MHz,  
 Time-out period =  $4096 \times (0x00FFFF+1) = 268,435,456$  clocks  
 Countdown time =  $(268,435,456 \text{ clocks} / 150,000,000 \text{ clocks per second})$   
 Countdown time = 1.7896 seconds
- *Example 3:* For Fsys = 100 MHz,  
 Time-out period =  $4096 \times (0x006234 + 1) = 25,141$  clocks  
 Countdown time =  $(25,141 \text{ clocks} / 100,000,000 \text{ clocks per second})$   
 Countdown time = 251.41 microseconds

## Watchdog Timer (WDT, WDT\_1)

When the counter reaches 0x000000, the WDT pin is asserted low. The WDT can be serviced by writing to the WSR register (as described in [Section 12.3.4, “Watchdog Service Register \(WSR\)”](#)). When serviced, the counter is loaded with the reload value stored in the WMR register, and then continues counting down from the new value.

There are the following registers in the WDT:

1. The watchdog control register (WCR) configures the watchdog’s operation.
2. The watchdog modulus register (WMR) determines the timer modulus reload value.
3. The watchdog count register (WCNTR) provides visibility to the counter value.
4. The watchdog service register (WSR) requires a service sequence to prevent assertion of the WDT pin.

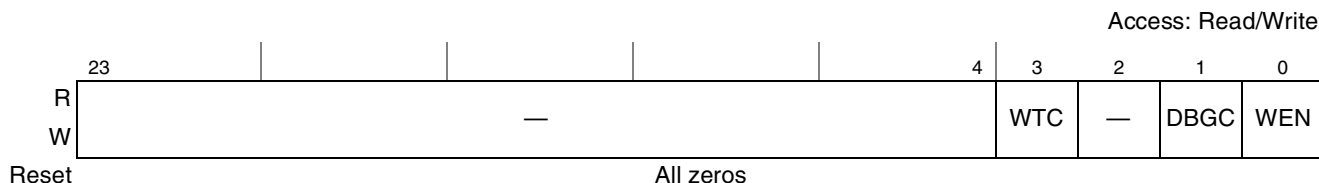


**Figure 12-1. Watchdog Timer Block Diagram**

## 12.3 WDT Registers

### 12.3.1 Watchdog Control Register (WCR)

The Watchdog Control Register is a 16-bit read/write register, shown in [Figure 12-2](#). It is a write-once register. Once written, the WCR register cannot be written again before a hardware reset (except in Debug mode). The WCR register uses only three bits—the rest of the bits are reserved.



**Figure 12-2. WCR Register**

1. Wait mode control bit (WTC)
  - 0 = Watchdog timer functions normally in Wait mode.
  - 1 = Watchdog timer is stopped in Wait mode.
2. Debug mode control bit (DBGC)
  - 0 = Watchdog timer functions normally in Debug mode.
  - 1 = Watchdog timer is stopped in Debug mode.
3. Watchdog Enable (WEN)
  - 0 = Watchdog timer is disabled.
  - 1 = Watchdog timer is enabled.

The WCR register is reset by a hardware reset only, and the reset value is 0x00000F. The WCR register can be updated in Debug mode, and the WCR register retains the changed value after Debug mode. If the WCR register has not been written before entering Debug mode, then writing in Debug mode does not affect the WCR register's capability to be written once in Normal mode. When Debug mode is exited, the timer operation continues from the state it was in (before entering Debug mode), but any updates made in Debug mode remain. If a write-once register is written to for the first time in Debug mode, the register is still writable when the Debug mode is exited.

#### NOTE

Changing the DBGC bit from 1 to 0 during Debug mode starts the watchdog timer. Changing the DBGC bit from 0 to 1 during Debug mode stops the watchdog timer.

### 12.3.2 Watchdog Counter and WCNTR Register

The watchdog count register (WCNTR) is a read-only register, shown in [Figure 12-3](#)—writing to the WCNTR register has no effect.

Address Y: 0xFFFFC2

Access: Read/Write

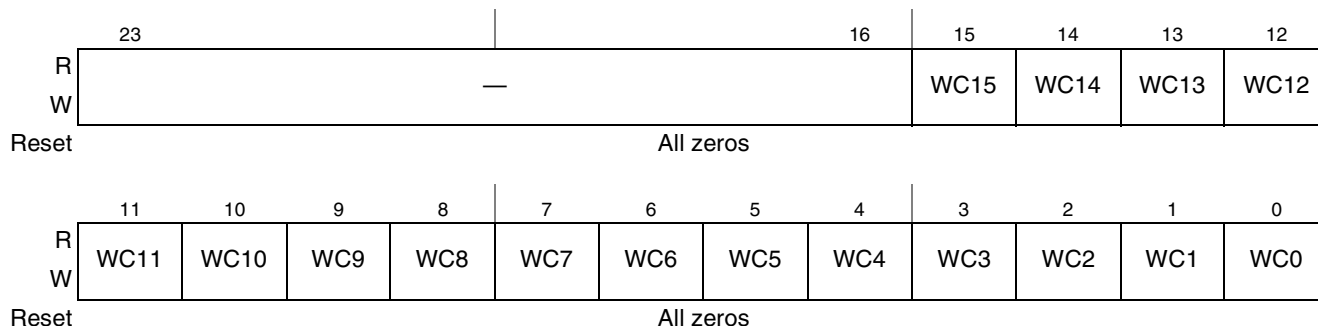


Figure 12-3. WCNTR Register

The WCNTR register reflects the current value of the 16-bit down-counter. The counter has a reset value of 0xFFFF. Counter operations include:

1. Counter Reset: The counter is reset asynchronously by a hardware reset. Reset value = 0xFFFF.
2. Load counter: The counter is synchronously loaded from the WMR register. A 16-bit load occurs if the watchdog is serviced. On a write to the WMR register, the corresponding value is updated in the counter.
3. Down-counting: The 16-bit counter is decremented every  $F_{sys}/4096$  clock cycles. When the counter value changes from 0x0000 to 0xFFFF, which means the WSR was not serviced properly, the WDT pin is asserted. The WDT pin assertion can only be cleared by hardware reset.

### 12.3.3 Watchdog Modulus Register (WMR)

The WMR register is a 16-bit read/write register, and is a write-once register, shown in [Figure 12-4](#).

Address Y: 0xFFFFC1

Access: Read/Write

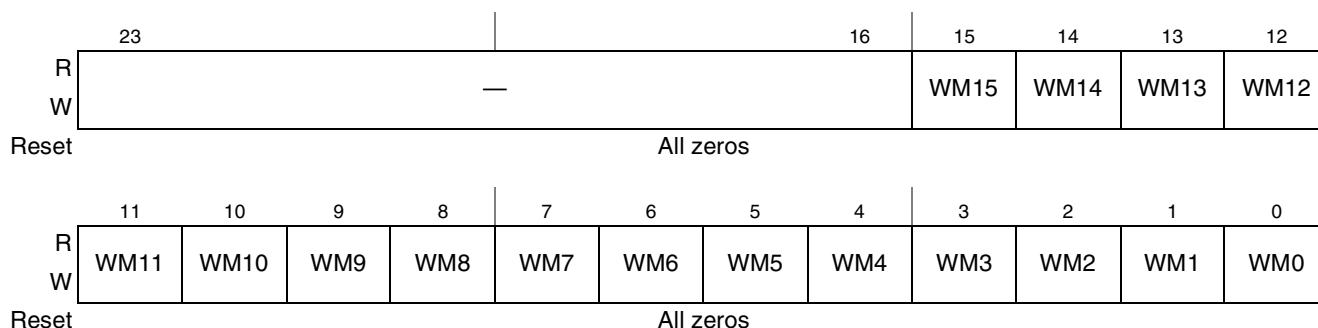


Figure 12-4. WMR Register

The WMR register contains the modulus value that is reloaded into the watchdog counter by a service sequence. After being written, the WMR register is not affected by further writes (except in Debug mode).

The WMR register can be written to in Debug mode (even if written to once earlier), and it retains the changed value after exiting the Debug mode. If the WMR register has not been written to before entering the Debug mode, writing to the WMR register in debug mode does not affect the WMR register’s capability to be written to once in Normal mode.

Writing to the WMR register immediately loads the new modulus value into the watchdog counter. The new value is also used at all subsequent reloads. Reading the WMR register returns the value in the modulus register. A hardware reset initializes the WMR register to 0x00FFFF.

### 12.3.4 Watchdog Service Register (WSR)

The WSR is a 16-bit write register, and is used to service the Watchdog timer, shown in [Figure 12-5](#).

Address Y: 0xFFFFC3

Access: Read/Write

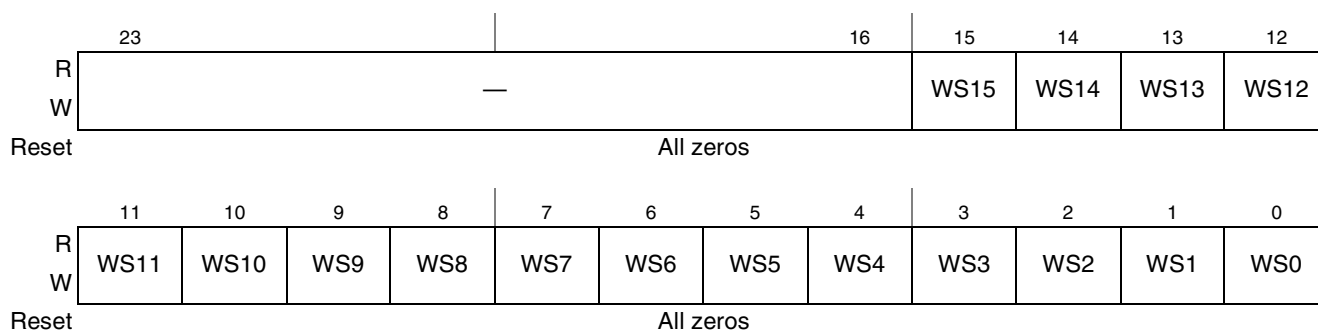


Figure 12-5. WSR Register

When the watchdog timer is enabled, the watchdog timer is serviced by writing 0x005555, and then writing 0x00AAAA to the watchdog service register (WSR). If the watchdog timer is not serviced before the time-out, the watchdog timer will assert the WDT pin.

Both writes must occur in the order listed before the time-out, but any number of instructions can be executed between the two writes. However, writing any value other than 0x005555 or 0x00AAAA to the WSR resets the servicing sequence, requiring both values to be written to keep the watchdog timer from asserting the WDT pin.

## 12.4 Watchdog Operating Modes

### 12.4.1 Wait Mode

If the WCR[3] bit is set, then the Watchdog timer function stops and enters Wait mode. The counter and the prescaler retain their values during Wait mode. If the WCR[3] bit is cleared, the timer function is unaffected in Wait mode. All register accesses function in the normal fashion, regardless of the value of the WCR[3] bit.

## 12.4.2 Debug Mode

If the WCR[1] bit is set, the Watchdog timer function stops and enters Debug mode. The counter and the prescaler retain their values during Debug mode. If the WCR[1] bit is cleared, the timer function is unaffected in Debug mode. In Debug mode, the WMR and WCR registers can be updated like a simple read/write register. The write-once property of these registers do not apply in Debug mode. All changes made in Debug mode are retained. A write-once register bit that has not previously been written is still writable when Debug mode is exited.

## 12.4.3 Stop Mode

The Fsys is assumed to be stopped in Stop mode. The watchdog timer does not function in Stop mode.



# Chapter 13

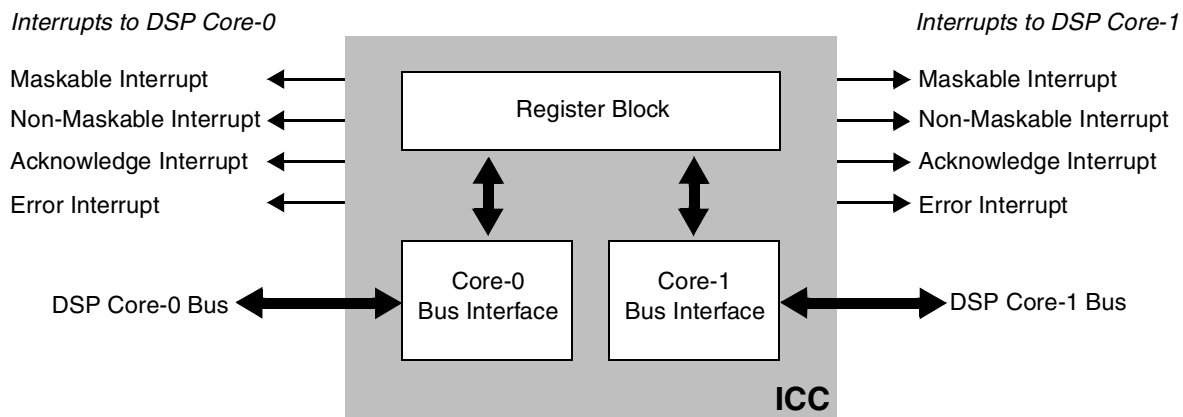
## Inter-Core Communication (ICC)

### 13.1 Introduction

Each DSP core can send out maskable or non-maskable interrupts to the other core via the ICC block. Each core has its own write data register, which is used to pass data to the other core when generating an interrupt to the other core. There are also polling registers for inter-core data exchange. [Figure 13-1](#) shows a block diagram of the ICC module.

#### 13.1.1 Overview

The ICC block is interfaced to the peripheral bus of both DSP cores; each core can access the registers that are dedicated to it. This section describes all of the registers in the register block.



**Figure 13-1. ICC Block Diagram**

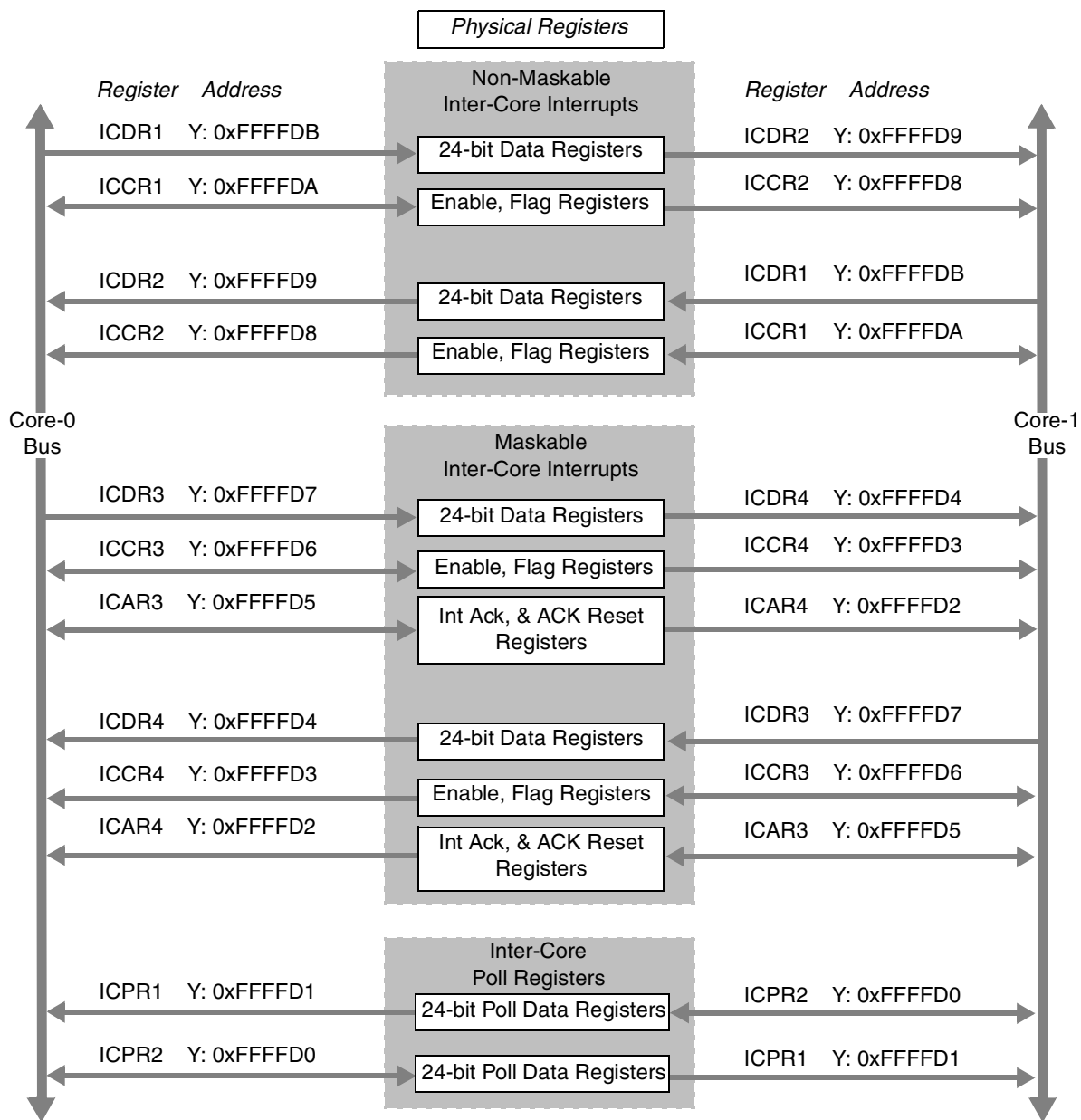


Figure 13-2. ICC Registers Block

There are a total of 12 registers in the ICC block. Each physical register is memory-mapped to the different addresses for each of the two DSP cores, but the registers that are mapped to the same address in the two cores have the same function.

## 13.1.2 Features

- Each DSP core can generate interrupts to the other core via the ICC, and each core can clear the interrupts from the other core by reading the corresponding data register.
  - The data is written into the *write data* register to generate an interrupt to the other core, if the IE bit of the core's control register is set. The core that services the interrupt can read the data from the other core by reading its own *read data* register, and this reading operation will clear the interrupt flag which produced the interrupt.
- An interrupt acknowledge function is available for the maskable interrupt. When the interrupted core acknowledges the maskable interrupt by reading the data register, an ACK register bit is set to inform the other core that the maskable interrupt has been serviced.
- Poll registers allow the DSP cores to exchange data and status. The contents of the polling register can be read by the other DSP core via its own read-only *polling* register.

## 13.2 Memory Map and Register Definition

### 13.2.1 Memory Map

Table 13-1 shows the memory tables for ICC registers. Each core has the same register address mapping.

**Table 13-1. ICC Block Memory Map**

Offset or Address	Register		Access	Reset Value	Section/Page
y: 0xFFFFDB	ICDR1	ICC Data Register 1	W	0x000000	<a href="#">13.2.2.1/13-5</a>
y: 0xFFFFDA	ICCR1	ICC Control Register 1	R/W	0x000004	<a href="#">13.2.2.2/13-6</a>
y: 0xFFFFD9	ICDR2	ICC Data Register 2	R	0x000000	<a href="#">13.2.2.3/13-6</a>
y: 0xFFFFD8	ICCR2	ICC Control Register 2	R	0x000004	<a href="#">13.2.2.4/13-7</a>
y: 0xFFFFD7	ICDR3	ICC Data Register 3	W	0x000000	<a href="#">13.2.2.5/13-7</a>
y: 0xFFFFD6	ICCR3	ICC Control Register 3	R/W	0x000000	<a href="#">13.2.2.6/13-8</a>
y: 0xFFFFD5	ICAR3	ICC Acknowledge Register 3	R/W	0x000000	<a href="#">13.2.2.7/13-8</a>
y: 0xFFFFD4	ICDR4	ICC Data Register 4	R	0x000000	<a href="#">13.2.2.8/13-9</a>
y: 0xFFFFD3	ICCR4	ICC Control Register 4	R	0x000000	<a href="#">13.2.2.9/13-9</a>
y: 0xFFFFD2	ICAR4	ICC Acknowledge Register 4	R/W	0x000000	<a href="#">13.2.2.10/13-10</a>
y: 0xFFFFD1	ICPR1	ICC Polling Register 1	R	0x000000	<a href="#">13.2.2.11/13-10</a>
y: 0xFFFFD0	ICPR2	ICC Polling Register 2	R/W	0x000000	<a href="#">13.2.2.12/13-11</a>

Table 13-2 shows the register summary table for ICC block.

**Table 13-2. ICC Registers Summary**

Register		23	22	21	20	19	18	17	16	15	14	13	12
		11	10	9	8	7	6	5	4	3	2	1	0
ICDR1 Y: FFFFDB	R												
	W	Data											
	R												
	W	Data											
ICCR1 Y: FFFFDA	R												
	W												
	R										DRE	IF	
	W												
ICDR2 Y: FFFF D9	R	Data of the other Core's ICDR1 register.											
	W												
	R	Data of the other Core's ICDR1 register.											
	W												
ICCR2 Y: FFFF D8	R												
	W												
	R										DRE	IF	
	W												
ICDR3 Y: FFFF D7	R												
	W	Data											
	R												
	W	Data											
ICCR3 Y: FFFF D6	R												
	W												
	R									EIE	EF	MIF	MIE
	W												
ICAR3 Y: FFFF D5	R												
	W												
	R											RACK	ACK
	W												
ICDR4 Y: FFFF D4	R	Data of the other Core's ICDR3 register.											
	W												
	R	Data of the other Core's ICDR3 register.											
	W												

**Table 13-2. ICC Registers Summary (Continued)**

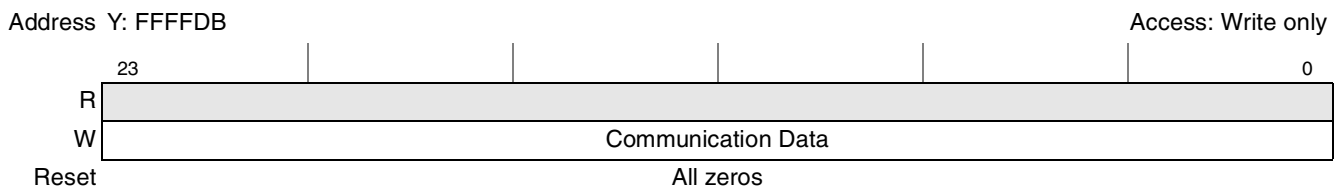
Register		23	22	21	20	19	18	17	16	15	14	13	12
		11	10	9	8	7	6	5	4	3	2	1	0
ICCR4 Y: FFFFD3	R												
	W												
	R									EIE	EF	MIF	MIE
	W												
ICAR4 Y: FFFFD2	R												
	W												
	R											RACK	ACK
	W												
ICPR1 Y: FFFFD1	R	Poll data from the other Core's ICPR2 register.											
	W												
	R	Poll data from the other Core's ICPR2 register.											
	W												
ICPR2 Y: FFFFD0	R	Poll Data to the other core (ICPR2 Data)											
	W	Poll Data to the other core (ICPR2 Data)											
	R	Poll Data to the other core (ICPR2 Data)											
	W	Poll Data to the other core (ICPR2 Data)											

For both DSP cores, ICPR1 is a read-only register, and ICPR2 is a R/W register.

## 13.2.2 Register Descriptions

### 13.2.2.1 ICDR1 (ICC Data Register 1)

The ICDR1 register is a 24-bit write-only data register. See [Figure 13-3](#).



**Figure 13-3. ICDR1 Write Data Register**

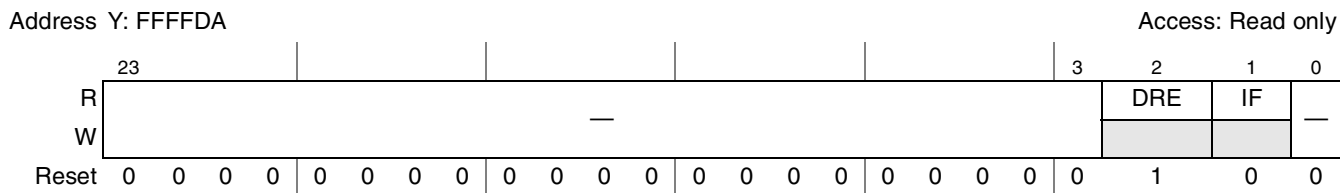
**Table 13-3. ICDR1 Field Descriptions**

Bit	Name	Description
23–0	Communication Data	Communication data that is written by one core and read only by the other core. The core that writes this register cannot read the data that it writes. One core writes this register to issue a non-maskable Interrupt to the other core.

Inter-Core Communication (ICC)

### 13.2.2.2 ICCR1 (ICC Control Register 1)

The ICCR1 control register is shown in [Figure 13-4](#).



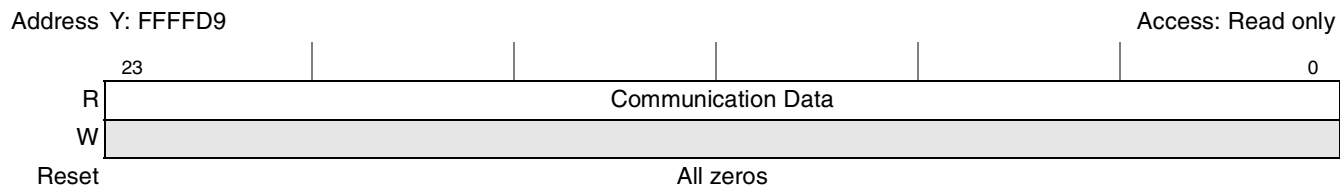
**Figure 13-4. ICCR1 Control Register 1**

**Table 13-4. ICCR1 Field Descriptions**

Bit	Name	Description
23–3	—	Reserved
2	DRE	Data Register Empty. Read-only; reset value is 1. 1: Data Register ICDR1 is Empty, new data can be written to ICDR1 register. 0: Data Register ICDR1 is Full, new data cannot be written to ICDR1 register.
1	IF	Read-only Interrupt Flag for non-maskable interrupt 1: Non-maskable interrupt flag is set, and an interrupt generated to the other core. 0: Flag cleared.
0	—	Reserved

### 13.2.2.3 ICDR2 (ICC Data Register 2)

The ICDR2 data register is shown in [Figure 13-5](#).



**Figure 13-5. ICDR2 Data Register**

**Table 13-5. ICDR2 Field Descriptions**

Bit	Name	Description
23–0	Communication Data	Read-only communication data reflecting the other core's ICDR1 data register.

### 13.2.2.4 ICCR2 (ICC Control Register 2)

The ICCR2 control register is show in Figure 13-6.

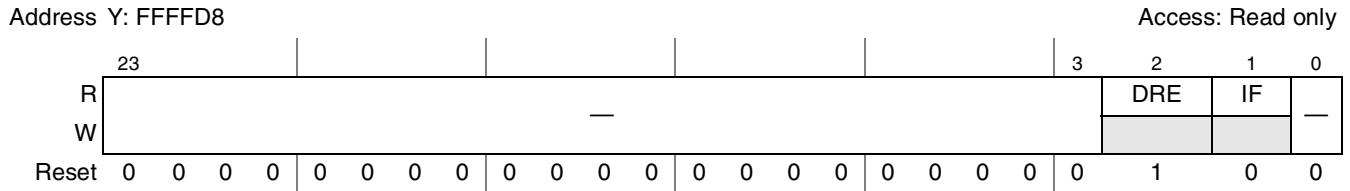


Figure 13-6. ICCR2 Control Register

Table 13-6. ICCR2 Field Descriptions

Bit	Name	Description
23–3	—	Reserved
2	DRE	Data Register Empty flag that reflects the status of the same bit of the other core's ICCR1 register.
1	IF	Interrupt Flag that reflects the status of the same bit of the other core's ICCR1 register. 1: Interrupt Flag is set valid, and interrupt is pending. 0: Interrupt Flag is cleared, no interrupt is pending.
0	—	Reserved

### 13.2.2.5 ICDR3 (ICC Data Register 3)

The ICDR3 data register is shown in Figure 13-7.

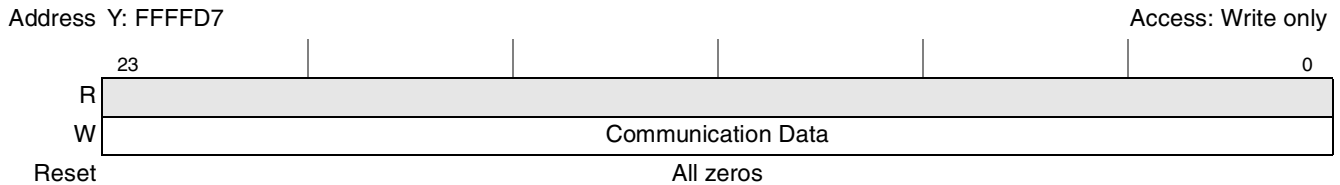


Figure 13-7. ICDR3 Data Register

Table 13-7. ICDR3 Field Descriptions

Bit	Name	Description
23–0	Communication Data	Communication data that written by one core to issue a Maskable Interrupt to the other core.



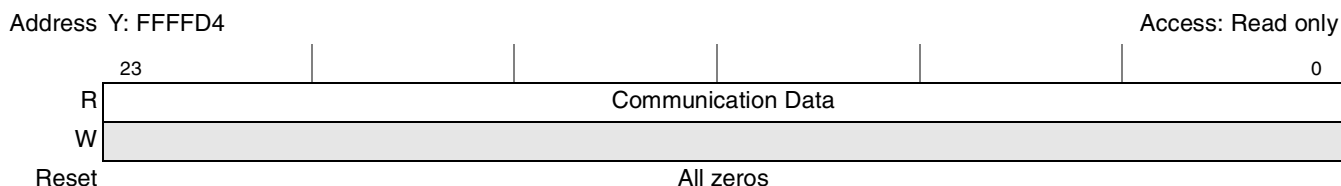


**Table 13-9. ICAR3 Field Descriptions (Continued)**

Bit	Name	Description
1	RACK	Reset ACK bit of this register. Writing a one (1) to this bit will clear the ACK bit.
0	ACK	Maskable interrupt acknowledge bit. 1: Asserted when the other core has serviced the maskable interrupt. 0: No interrupt serviced has been by the other core. To clear ACK bit, write a one (1) to RACK bit (which means that the ACK interrupt is serviced).

### 13.2.2.8 ICDR4 (ICC Data Register 4)

The ICDR4 Data Register is shown in [Figure 13-10](#).



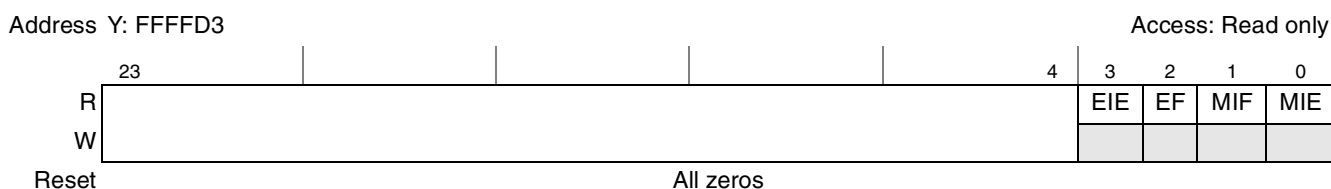
**Figure 13-10. ICDR4 Data Register**

**Table 13-10. ICDR4 Field Descriptions**

Bit	Name	Description
23–0	Communication Data	Read-only communication data that reflects the other core’s ICDR3 data register.

### 13.2.2.9 ICCR4 (ICC Control Register 4)

The ICCR4 Control Register is shown in [Figure 13-11](#).



**Figure 13-11. ICCR4 Control Register**

**Table 13-11. ICCR4 Field Descriptions**

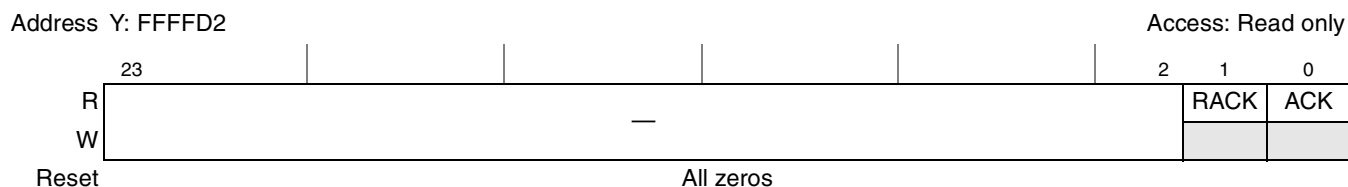
Bit	Name	Description
23–4	—	Reserved. Write 0 to ensure future compatibility.
3	EIE	Error Interrupt Enable bit that reflects the status of the same bit of the other core’s ICCR3 register.
2	EF	Error Interrupt Flag that reflects the status of the same bit of the other core’s ICCR3 register.

**Table 13-11. ICCR4 Field Descriptions (Continued)**

Bit	Name	Description
1	MIF	Maskable Interrupt Flag that reflects the status of the same bit of the other core's ICCR3 register. 1: Interrupt Flag is set 0: Interrupt Flag is cleared
0	MIE	Interrupt Enable bit that reflects the status of the same bit of the other core's ICCR3 register. 1: Interrupt is Enabled 0: Interrupt is Disabled

### 13.2.2.10 ICAR4 (ICC Acknowledge Register 4)

The ICAR4 Register is shown in [Figure 13-12](#).



**Figure 13-12. ICAR4 Control Register**

**Table 13-12. ICAR4 Field Descriptions**

Bit	Name	Description
23–2	—	Reserved. Write 0 to ensure future compatibility.
1	RACK	Reflects the value of the RACK bit in the other core's ICAR3 register.
0	ACK	Reflects the value of the ACK bit in the other core's ICAR3 register.

### 13.2.2.11 ICPR1 (ICC Poll Register 1)

The ICPR1 polling register is shown in [Figure 13-13](#).



**Figure 13-13. ICC\_ICPR1 Polling Register**

**Table 13-13. CPR1 Field Descriptions**

Bit	Name	Description
23–0	Poll Data	24-bit Poll data from the other core. Read-only.

### 13.2.2.12 ICPR2 (ICC Poll Register 2)

The ICPR2 polling register is shown in Figure 13-14.



**Figure 13-14. ICPR2 Polling Register**

**Table 13-14. CPR2 Field Descriptions**

Bit	Name	Description
23–0	Poll Data	The DSP Core writes this register to transfer poll data to the other core. The data can be read out by the other core by polling its own ICPR1 register.

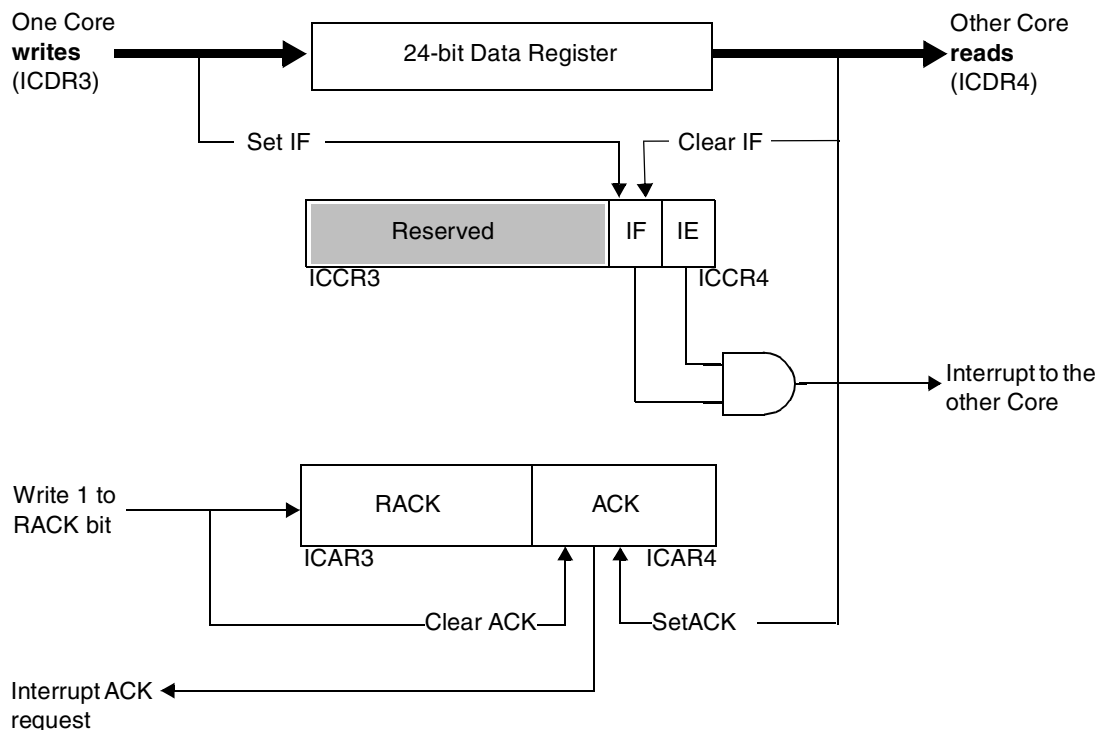
## 13.3 Programming Model

The ICC provides two means for each DSP core to interact with the other core:

- Poll registers are implemented to allow each core to write a poll data register (write ICPR2), which can be read by the other core. Both cores can only read the data from the other core by polling its own read-only register (read ICPR1).
- Interrupt-driven core interactions are supported. Two interrupts can be generated by each core to interrupt the other core. One interrupt is maskable, and the other interrupt is non-maskable. If the interrupt is enabled by the IE bit in the core’s control register, an interrupt is generated when the other DSP core writes its data register. The core that receives this interrupt will service it by running an interrupt service routine to clear the interrupt. The priority of the interrupt is determined by the corresponding bits in the PIC register (IPRP).

### 13.3.1 Inter-Core Maskable Interrupts

The maskable interrupt operation is shown in [Figure 13-15](#).



**Figure 13-15. Maskable Interrupts**

In [Figure 13-15](#), the 24-bit data register is used to store the maskable interrupt communication data that will be read out by the other core. The control register contains the IF (Interrupt Flag) and IE (Interrupt Enable) bits that are used to generate the interrupt to the other core. When interrupt is disabled or unused, the 24-bit data register can act as a polling data register as well.

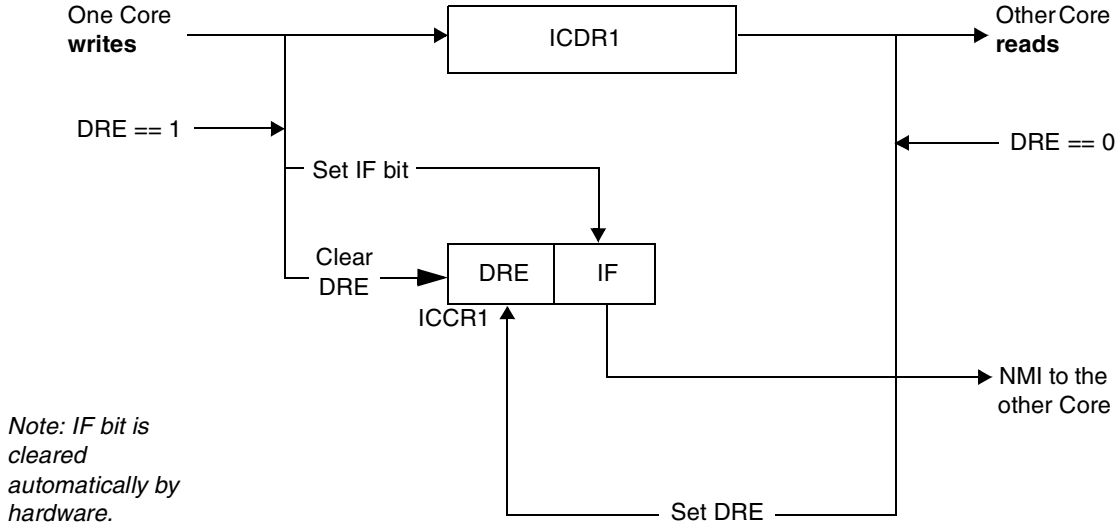
When a core writes any data to its communication data register (ICDR3), the IF bit of its control register (ICCR3) will be set by hardware. If the IE bit is logic 1 at the same time, an interrupt is generated for the other core. When the other core services this interrupt, it reads its own read-only communication data register (ICDR4) to obtain the data from the core that produces the interrupt. This reading operation will clear the IF bit and set the ACK bit of the other core’s ICAR3 register, and an acknowledge interrupt is generated to indicate the serviced core that the interrupt has been serviced. The ACK bit can be cleared by writing 1 to the RACK bit of ICAR3 register.

For the core that generates the interrupt, the IE bit is readable and writable, but the IF bit is read-only. For the core that services the interrupt, IF and IE are read-only register bits.

For the core that generates the maskable interrupt, the RACK bit is read-writable and the ACK bit is read-only. For the core that services the maskable interrupt, RACK and ACK are both read-only bits.

### 13.3.2 Inter-Core Non-Maskable Interrupts

The ICC non-maskable interrupt operation is shown in [Figure 13-16](#).



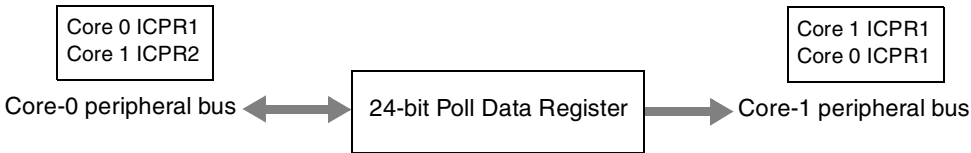
**Figure 13-16. Non-Maskable Interrupts**

When one DSP core needs to generate a non-maskable interrupt to the other core, that core writes a datum to the ICDR1 data register. However, the data can be written to this register and the IF bit can only be set when DRE is 1, which indicates that the data register is empty. So the DRE bit must be polled *before* writing a datum to the ICDR1 data register. When the core writes a datum successfully into the ICDR1 register, the DRE bit is cleared to indicate that the ICDR1 register is full, and the other core can now read this data correctly using its interrupt service routine. After the other core has read the communication data via its read-only ICDR2 register, the DRE bit will be set to indicate that the ICDR1 register is empty again.

The IF bit of the inter-core non-maskable interrupt control register (ICCR1) is not cleared by writing 1 to it or by the other core’s reading of the ICDR1 register. The IF bit of the ICCR1 register is automatically cleared by the hardware when the other core has serviced this non-maskable interrupt.

### 13.3.3 Polling

The poll register is a 24-bit R/W register for the two DSP cores to share data or software-defined status flags to the other core. The two cores can only read *their own* read-only poll registers to obtain the data from the other core. See [Figure 13-17](#).



**Figure 13-17. Poll Data Register**

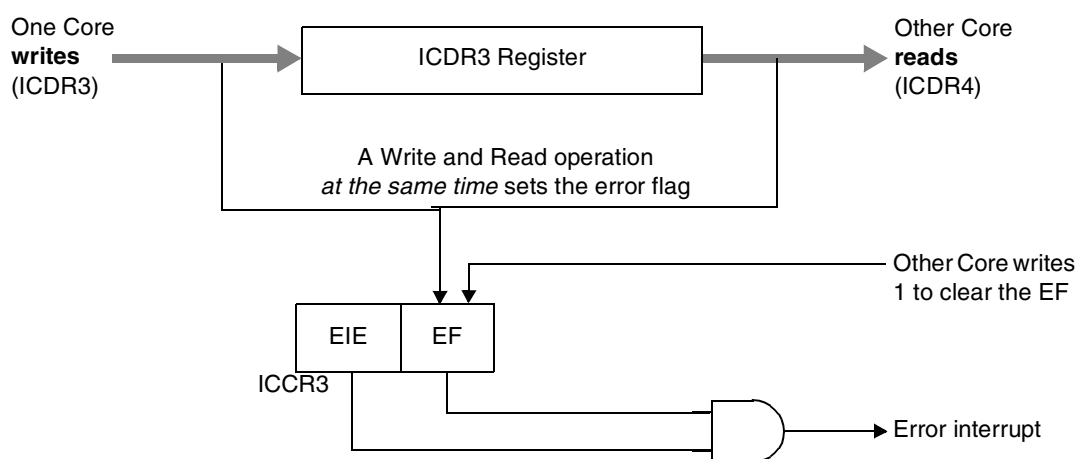
## Inter-Core Communication (ICC)

For the two DSP cores, poll register ICPR2 is read-writable, and poll register ICPR1 is read-only. One core writes data to its ICPR2 poll register, after which the other core reads its own ICPR1 to obtain the data (that was written to the ICPR2 register).

### 13.3.4 Error Interrupts

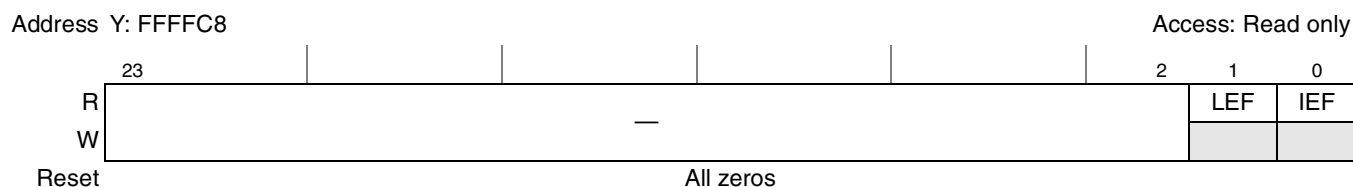
The maskable error interrupt for each core is generated when one core writes to the ICDR3 register and the other core reads the ICDR4 register *at the same time*. The core that reads the ICDR4 register will receive this error interrupt. This situation will not happen if reading the ICDR4 register is always executed in the maskable interrupt service routine. Generating this error interrupt condition is only for handling such an error in case that the wrong operation takes place.

For the communications via maskable interrupt, users should always read the ICDR4 register in the maskable interrupt service routine. [Figure 13-18](#) shows how error interrupts are generated.



**Figure 13-18. Error Interrupt Generation**

When this error condition occurs, the maskable interrupt will not be generated (even it is enabled), because the maskable interrupt flag is not set by the hardware when the error condition occurs. The ICC error interrupt is shared with the EMC error interrupt. For distinguishing the two interrupts (ICC error and EMC error), there is a read-only status register implemented, called the Error Interrupt Status Register. The address of this read-only status register is Y: 0xFFFFC8. See [Figure 13-19](#) for the bit definitions.



**Figure 13-19. Error Interrupt Status Register (EISR)**

**Table 13-15. EISR Field Description**

Bit	Name	Description
23–2	Reserved	
1	LEF	EMC Error Flag 1: EMC Error is Generated 0: EMC Error is Not Generated
0	IEF	ICC Error Flag 1: ICC Error Condition is Generated 0: ICC Error Condition is Not Generated

Each core has an EISR register. For Core-0's EISR register, the IEF indicates the flag of the error interrupt to Core-0. For Core-1's EISR register, the IEF bit indicates the flag of the error interrupt to Core-1.

To see which module produces this error interrupt, you must poll this ESIR register in the interrupt service routine. If the interrupt is caused by the ICC, then write a one (1) to the ICCR3[2] bit to clear this status flag.

### 13.3.5 Reset

Both hardware reset and software reset can put all of the ICC registers to a known state.





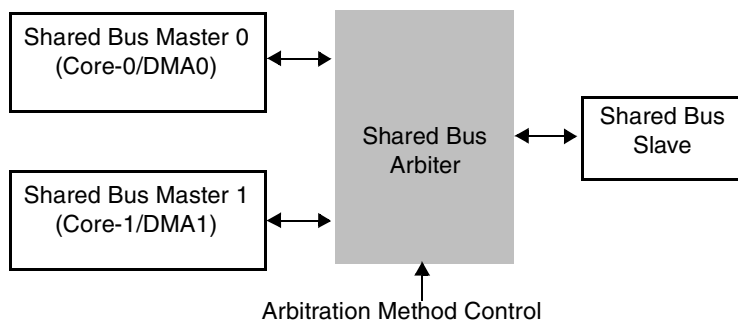
# Chapter 14

## Shared Bus Arbiter

### 14.1 Introduction

#### 14.1.1 Overview

The Shared Bus Arbiter provides coherent system bus arbitration between two Shared Bus masters (Core-0/DMA0, Core-1/DMA1) using round-robin, fixed, and other methods. The Shared Bus Arbiter is used when two masters access external memory, shared memory or shared peripherals, but only if both masters are accessing the same endpoint. No arbitration occurs if each Shared Bus master accesses different endpoints because there is no bus contention.



**Figure 14-1. Shared Bus Arbiter**

#### 14.1.2 Features

Each Shared Bus is a multiplexed bus of the DSP core's P, X and Y buses and the DMA bus. Arbitration on each Shared Bus between the DSP core and its DMA is configured by the Core-DMA Priority bits in the OMR register (see [Chapter 5, “Core Configuration,”](#) for more information).

This section is limited to describing the arbitration between the shared bus for Core 0/DMA 0 and the shared bus for Core 1/DMA 1.

The arbiter includes:

- Flexible arbitration policy, which supports:
  - Fixed arbitration method
  - Round-robin arbitration method
  - Specific arbitration for continuous transfer
- Arbitration happens only when the current access is enabled and in active address scope
- Zero-cycle delay in arbitration

## 14.2 Memory Map and Register Definition

There are no registers or memory in this block.

## 14.3 Functional Description

The following sections describe the arbiter functionality.

### 14.3.1 Shared Bus Arbitration

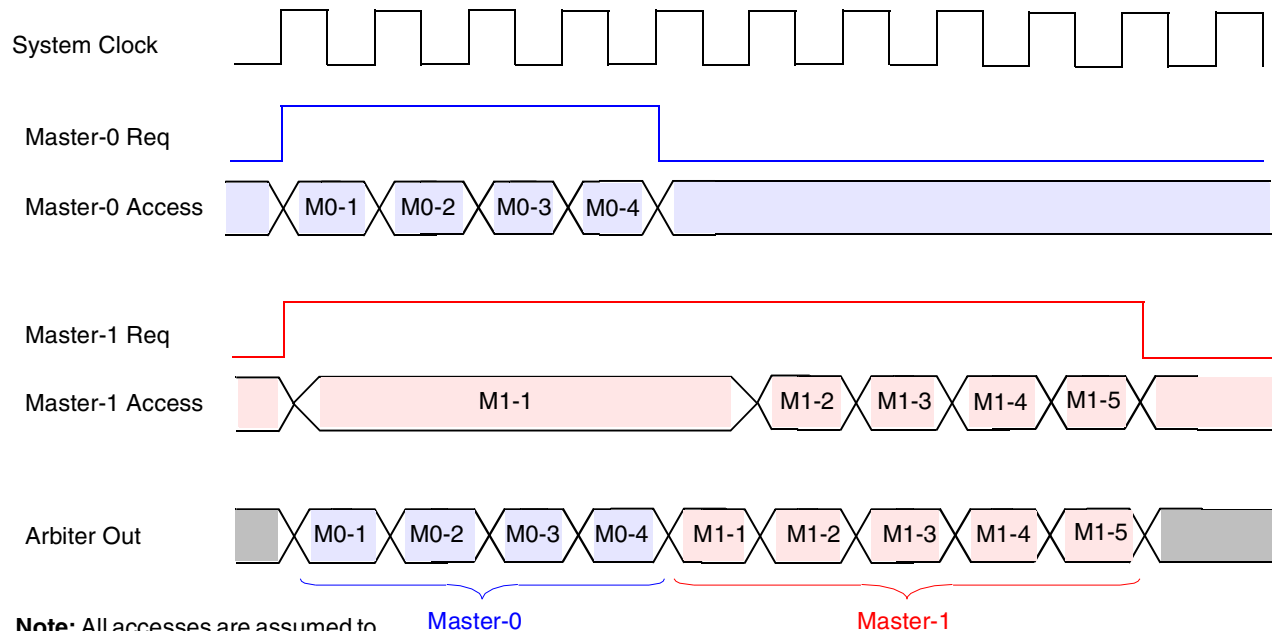
The arbitration process involves the masters and the arbiter. The masters arbitrate on the privilege to own the address tenure. For the data tenure, the arbiter uses the same order of transactions as address tenures.

#### 14.3.1.1 Configuring the Arbitration Method

When Shared Bus access contention occurs, the Arbiter Control Register (ARCR) in the chip configuration module determines the arbitration method. Three static arbitration methods are supported: Shared Bus Master 0 always has priority, Shared Bus Master 1 always has priority, and the round-robin method (the bus masters take turns getting bus access). See the [Chapter 20, “Chip Configuration Module,”](#) for more about the ARCR register.

### 14.3.1.2 Shared Bus Master 0/1 Always Has Priority Methods

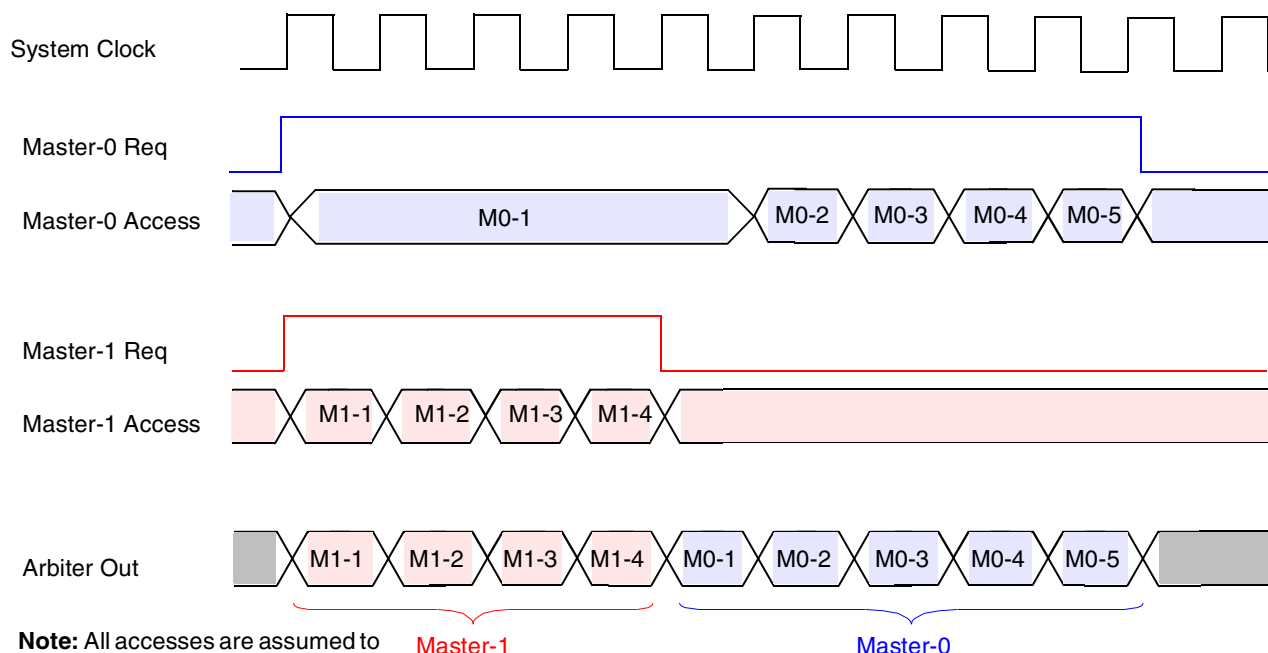
When either Shared Bus Master 0 or Shared Bus Master 1 has priority, a Shared Bus access request from the master with priority is always granted first. The access sequences for this type of static arbitration based on priority for Master 0 or Master 1 are shown in [Figure 14-2](#) and [Figure 14-3](#).



**Note:** All accesses are assumed to finish with 0 wait states after the arbitration has occurred.

**Figure 14-2. Shared Bus Master 0 Always has Priority**

## Shared Bus Arbiter



**Note:** All accesses are assumed to finish with 0 wait states after the arbitration has occurred.

**Figure 14-3. Shared Bust Master 1 Always has Priority**

### 14.3.1.3 Round-Robin Arbitration Method

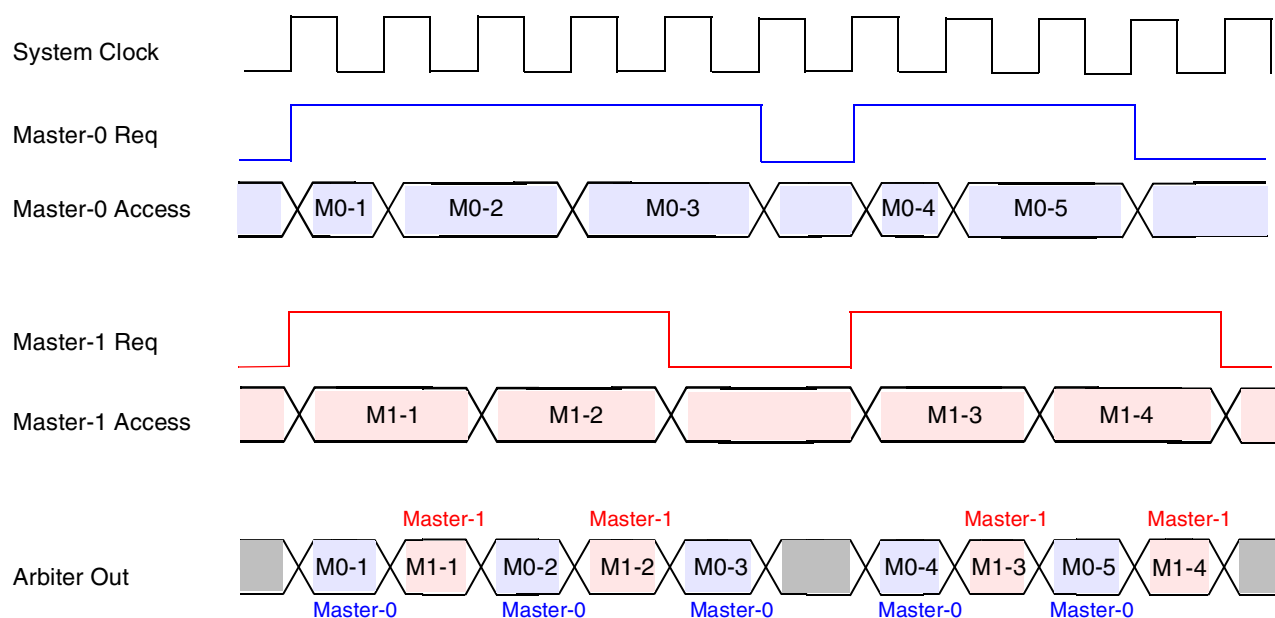
The Round-Robin arbitration method that decides the access delays is shown in [Table 14-1](#) and [Figure 14-4](#). In round-robin arbitration, Master 0 always has priority on the first contention before bus access starts alternating between the two bus masters. All it takes is one access from either Master 0 or Master 1 with no contention and bus access goes back to Master 0 first on the next bus contention.

**Table 14-1. Round-Robin Arbitration Method**

State	Input			Output		
	Flag	Master 0	Master 1	Master 0	Master 1	Flag
1	0	0	0	0	0	0
2	0	0	1	0	1	0
3	0	1	0	1	0	0
4	0	1	1	1	0	1
5	1	0	0	N/A		
6	1	0	1	0	1	0
7	1	1	0	N/A		

**Table 14-1. Round-Robin Arbitration Method (Continued)**

State	Input			Output		
	Flag	Master 0	Master 1	Master 0	Master 1	Flag
8	1	1	1	0	1	0
<b>Note:</b>	Flag: 1 – Set, 0 –Cleared			Flag: 1 – Set, 0 – Cleared		
	Master 0/Master 1: 1 – Access requested, 0 – No access requested			Master 0/Master 1: 1 – Access provided, 0 – No access provided		



**Note:** All accesses are assumed to finish with 0 wait states after the arbitration has occurred.

**Figure 14-4. Round-Robin Arbitration Method**

### 14.3.1.4 Cores/DMA Continuous Access Arbitration

The arbiter has specific arbitration behavior for continuous access from a DSP core or DMA.

A core access introduced by a core RMW (read-modify-write) instruction means that the next access must be by the same core, and will cause the Shared Bus Arbiter to reserve the next access for the same Shared Bus master. This behavior implies that one core RMW operation cannot be interrupted by the other Shared Bus masters (the other DSP core or DMA).

DMA continuous access can be set by configuring the corresponding DMA channel control register. If a DMA continuous access has priority over the other Shared Bus master, then the Shared Bus master must wait until one cycle after the DMA continuous access has completed, before it takes the access.

## Shared Bus Arbiter

For shared memory access, DMA continuous access is ignored (the core RMW instruction is not ignored). If DMA uses continuous access to the shared memory, the DMA continuous access is regarded as normal access by the shared memory.

# Chapter 15

## Shared Memory

### 15.1 Overview

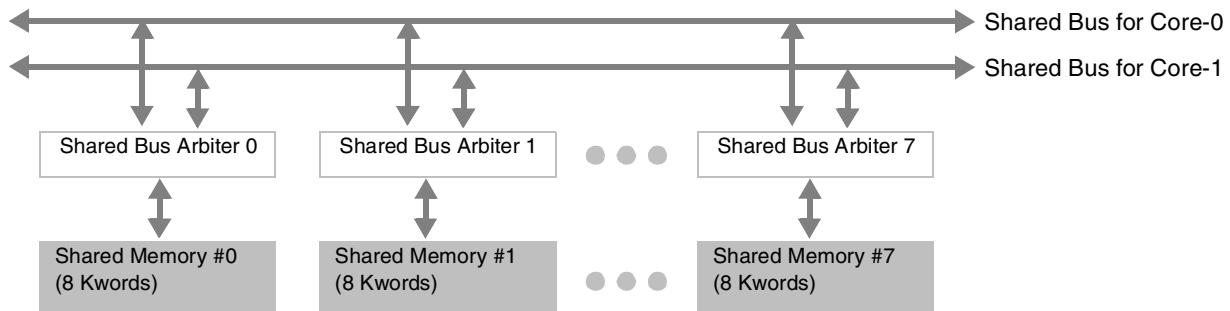
The shared memory is memory space accessible by either DSP Core-0 or Core-1, and consists of multiple 8-Kword blocks for a total of  $8 \times 4$  Kwords of memory space. The memory space is located at 0x030000–0x37FFFF, and can be accessed as X, Y or P memory.

- Three arbitration methods can be used. See the [Chapter 14, “Shared Bus Arbiter,”](#) for more information about arbitration.
- Consists of multiple 4-Kword memory blocks are used. Different blocks can be accessed at the same time by the two DSP cores with no contention.

### 15.2 Block Diagram

To reduce access conflict, the entire shared memory space is divided into several shared memory banks. Shared Bus 0 and Shared Bus 1 accesses are able to happen simultaneously as long as the two accesses do not hit the same memory bank. If both Cores or DMA attempt to access data within the same shared memory block, a delay of at least 1 cycle will occur, but no data will be lost. In general, memory accesses complete with zero wait states unless there is memory contention. All P memory accesses by a core via the Shared Bus always completes with one additional wait state.

Figure 15-1 shows the block diagram for shared memory.



**Figure 15-1. Shared Memory**

See the [Chapter 14, “Shared Bus Arbiter,”](#) for more information.





# Chapter 16

## Shared Peripheral Bus

### 16.1 Introduction

The Shared Peripheral Bus module performs arbitration between the two DSP cores, to convert transactions issued on the Shared Bus to Shared Peripheral Bus transactions, and to integrate other internal device circuits with the Shared Peripheral bus interfaces. Both DSP cores use the Shared Peripheral Bus to access the shared peripherals. [Figure 16-1](#) shows how the Shared Peripheral Bus fits into the DSP56724/DSP56725 device.

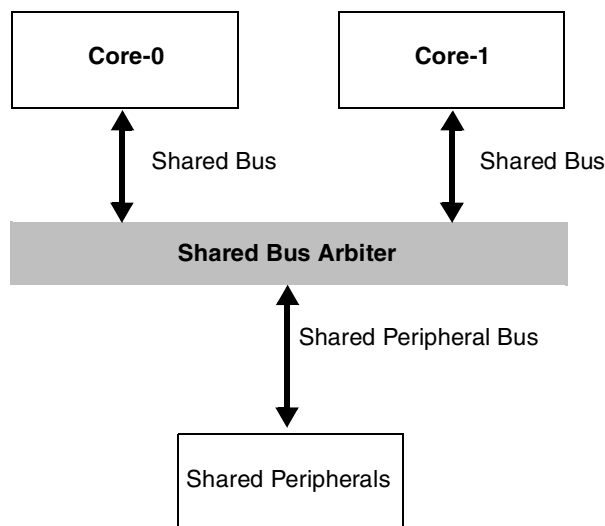


Figure 16-1. Shared Peripheral Bus in DSP56724/DSP56725

#### 16.1.1 Features

Shared Peripheral Bus features include:

- Two arbitration methods are used:
  - Fixed arbitration
  - Round-robin arbitration
- Specific arbitration methods for continuous transfers for DSP Cores and DMAs
- Fire and forget single write accesses
- No additional delays are introduced when converting Shared Bus access to Shared Peripheral Bus access. Most accesses compete with zero wait states (in addition to any delays added by the shared peripheral and arbitration).

## 16.2 Memory Map and Register Definition

There are no registers or memory in this block.

## 16.3 Functional Description

The Shared Peripheral Bus sits between a Shared Bus Arbiter and the shared peripherals. (For more details about the possible arbitration schemes, see [Chapter 14, “Shared Bus Arbiter.”](#))

Write accesses on the Shared Peripheral Bus complete with zero wait states, in addition to any wait states added by the Shared Bus Arbitration. Please note that when writing to a peripheral that requires wait states on accesses, the write will still complete with zero wait states, but the Shared Peripheral Bus will still take the required number of wait states to complete the bus access. This has the effect of adding wait states to any subsequent accesses during that time.

Read accesses on the Shared Peripheral Bus usually complete with zero wait states, in addition to any wait states added by the particular peripheral accessed and the Shared Bus arbitration.

However, any write access on the Shared Peripheral Bus that is immediately followed by a read access on the following system clock cycle will cause one wait state to be added to the read access. There must always be one empty cycle between a write access and a subsequent read access on the Shared Peripheral Bus.

In addition, any read or write access that follows a write access to a peripheral (that adds wait states to the access) will add that number of wait states to the current access.

## Chapter 17

# EMC Burst Buffer

### NOTE

The EMC is not available on the DSP56725 device.

## 17.1 Introduction

The EMC Burst Buffer converts single transactions on the Shared Bus to EMC burst transactions.

[Figure 17-1](#) shows how the EMC Burst Buffer is incorporated in the DSP56724 device. In the DSP56724, Core 0/DMA 0 is Shared Bus master 0, and Core 1/DMA 1 is Shared Bus master 1.

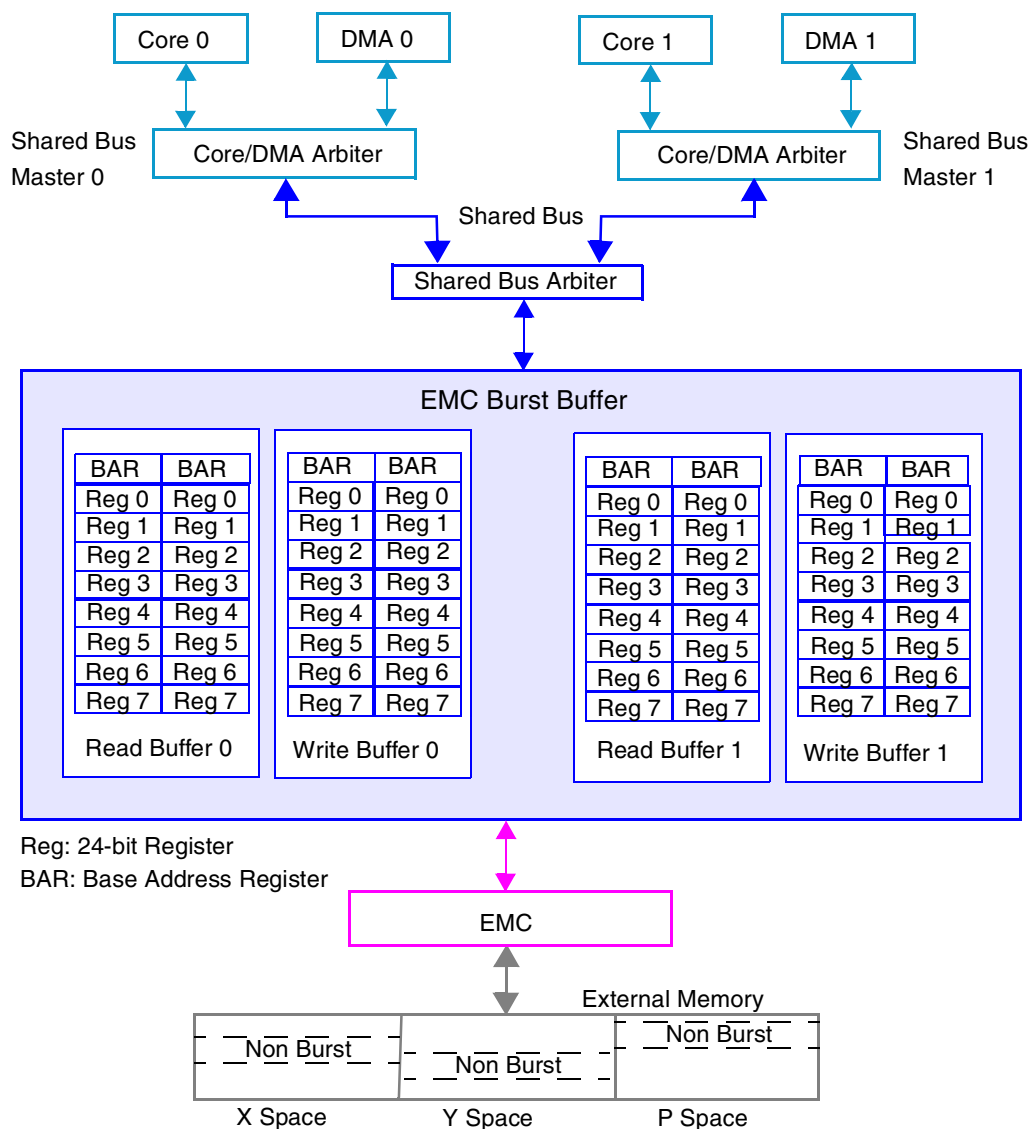


Figure 17-1. EMC Burst Buffer in DSP56724

### 17.1.1 Overview

The EMC Burst Buffer sits between the Shared Bus and the EMC. The EMC Burst Buffer transfers read/write transactions issued on the Shared Bus to the EMC. In addition, a small burst cache is introduced into this block to provide burst operations, to support successive accesses from Shared Bus masters for high data transfer performance.

Four burst buffers are included in the burst cache for Shared Bus master 0 read, Shared Bus master 0 write, Shared Bus master 1 read, and shared Bus master 1 write respectively. Each burst buffer has 16 24-bit words alternating (ping-pong style) between the two sets of 8 words.

## 17.1.2 Features

The EMC Burst Buffer includes the following features:

- A simple burst cache enables burst accesses, including four burst buffers.
- Executes burst from Shared Bus masters based solely on the address being accessed.
- Fire-and-forget single write access.
- Zero wait state access for writes and reads that hit within the burst buffer.
- Read data returned are always zero if no access happens.

## 17.2 Memory Map and Register Definition

No registers or memory are in this block.

## 17.3 Functional Description

The following sections describe the arbiter functionality: address scope, burst control, read access and write access. The module can execute burst read/write transfers because there is a small burst cache with two 8-word buffers for both burst read and burst write. All outputs will not be registered, so a one cycle delay is avoided in transfers. All reads to entries within the burst buffer finish with zero wait states, while writes to an entry within a burst buffer (if there is room) also finish with zero wait states.

Four burst buffers are included in the burst cache of the gasket for Shared Bus master 0 read, Shared Bus master 0 write, Shared Bus master 1 read, and Shared Bus master 1 write respectively. Each buffer has 16 24-bit words alternating (ping-pong operation) between the two sets of 8. Burst operation can be controlled by configuring the corresponding register bits in the Chip Configuration module in the DSP56724.

### 17.3.1 Burst Control

There are control bits in the EMBC (External Memory Burst Control) register for burst access control of X, Y, and P external memory space.

- Bits 3-0 of the EMBC register are named the “X Space Burst Boundary”, and these bits define the 1M burst region in the X external memories.
- Bits 11-8 of the EMBC register are named the “Y Space Burst Boundary”, and these bits define the 1M burst region in the Y external memories.
- Bits 19-16 of the EMBC register are named the “P Space Burst Boundary”, and these bits define the 1M burst region in the P external memories.

For more information about these bits or the EMBC register, see [Chapter 20, “Chip Configuration Module.”](#)

## EMC Burst Buffer

In the EMBC register, the two EXMBC bits, two EYMBC bits, and two EPMBC bits control the burst behavior of X, Y, and P external memories, as shown in [Table 17-1](#), [Table 17-2](#), and [Table 17-3](#).

**Table 17-1. External Memory X Space Burst Control**

EXMBC	Action
00	Module doesn't burst any external memory accesses inside the X address space.
01	Module only bursts external memory accesses outside of the non-burst region of X address space.
10	Module bursts any external memory accesses inside the X address space.
11	Reserved

**Table 17-2. External Memory Y Space Burst Control**

EYMBC	Action
00	Module does not burst any external memory access inside the Y address space.
01	Module only bursts external memory access outside the non-burst region of Y address space.
10	Module bursts any external memory accesses inside Y address space.
11	Reserved

**Table 17-3. External Memory P Space Burst Control**

EPMBC	Action
00	Module does not burst any external memory accesses inside the P address space.
01	Module only bursts external memory accesses outside the non-burst region of P address space.
10	Module bursts any external memory accesses inside the P address space.
11	Reserved

Any access which does not use a burst buffer, does not affect the current contents of that burst buffer.

Software can manually invalidate the read buffer or write buffer by setting the corresponding control register bit to assert the corresponding control signal to the gasket. Software is then totally responsible for maintaining coherency on the four buffers.

When software sets the IRB0 or IRB1 bit in the ODBC register, the module will set the data left in the read buffer of Shared Bus master [0/1] to “invalid,” and will stall any access requests from any Shared Bus master [0/1] until all of the data in the read buffer has been invalidated. After invalidation of the read buffer has finished, the hardware automatically clears the corresponding register bit (IRB0 or IRB1).

When software sets the IWB0 or IWB1 bit in the ODBC register, the module should store the data left in the write buffer of Shared Bus master [0/1] into external memory, and will stall access requests from any Shared Bus master [0/1] until all of the data in the burst buffer is written to external memory. After invalidation of the write buffer has finished, the hardware automatically clears the corresponding register bit (IWB0 or IWB1).

## 17.3.2 Read Access

When the current access is read access inside the available burst address scope, the EMC Burst Buffer Burst Buffer will execute a burst read operation; otherwise the access is taken as a single read access. The Figure 17-2 shows the read access flow.

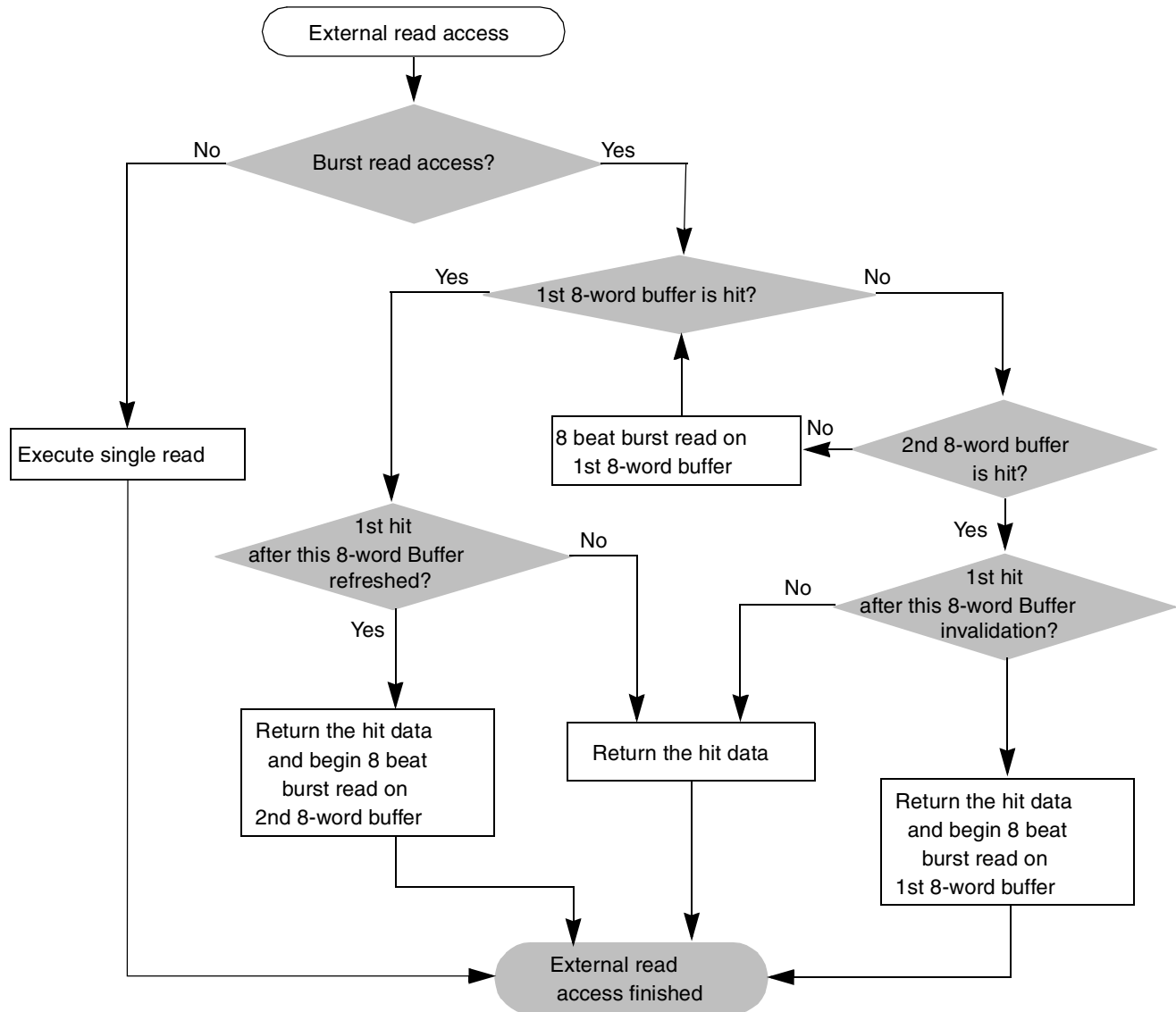


Figure 17-2. Read Access Flow

### 17.3.2.1 Single Read

For single read access, the EMC Burst Buffer passes the access straight through to the EMC, without any other functions being involved.

### 17.3.2.2 Burst Read

There are two burst read buffers: one burst read buffer for Shared Bus master 0, and one burst read buffer for Shared Bus master 1. Each Shared Bus master uses its own read buffer independently.

Each read buffer has 16 24-bit words alternating (ping-pong style operation) between the two sets of 8 words, and there are two 8-word buffers in each read buffer.

The burst read is always an 8-beat transfer to the EMC.

In one read buffer, each 8-word buffer has a 26-bit base address register, because the Shared Bus access address is 26-bits— actually consists of a 24-bit address and a 2-bit encoded address space.

Each 8-word buffer has one “valid” bit, which indicates that the read data in the 8-word read buffer is available for burst read access when the bit asserted.

During an burst read access, the module compares the access address with the base address register for the two 8-word buffers, respectively.

- If the address is inside the range from base address to (base address + 7) in either 8-word buffer, the data is in the buffer, and the buffer returns the read data to the Shared Bus directly with zero wait states.
- If no 8-word buffer is hit, the module sends an 8 word burst request to the EMC to load 8 words from external memory into the first 8-word buffer; at the same time, the access address is loaded into the base address register. After the burst operation has finished, the corresponding “valid” bit is set, and the read data that was loaded into the buffer is returned.
- If the data hit is the first time for one 8-word buffer (after the last burst operation has finished), or is the first data hit for one of the 8 word buffers, then the module triggers the other 8-word buffer to execute an 8-word burst operation after the end address of the last burst operation, and on the following 8 words in external memory, the module stores the burst access address into the base address register. The two 8-word buffers alternately execute bursts (ping-pong style).

When an 8-word buffer is executing a burst operation, the other 8-word buffer can return its data (stored in it) to the Shared Bus on a hit due to a read access.

### 17.3.3 Write Access

When the current access is a write access inside the available burst address scope, the EMC Burst Buffer executes a burst write operation; otherwise the access is taken as a single write access.



The Figure 17-3 shows the write transfer.

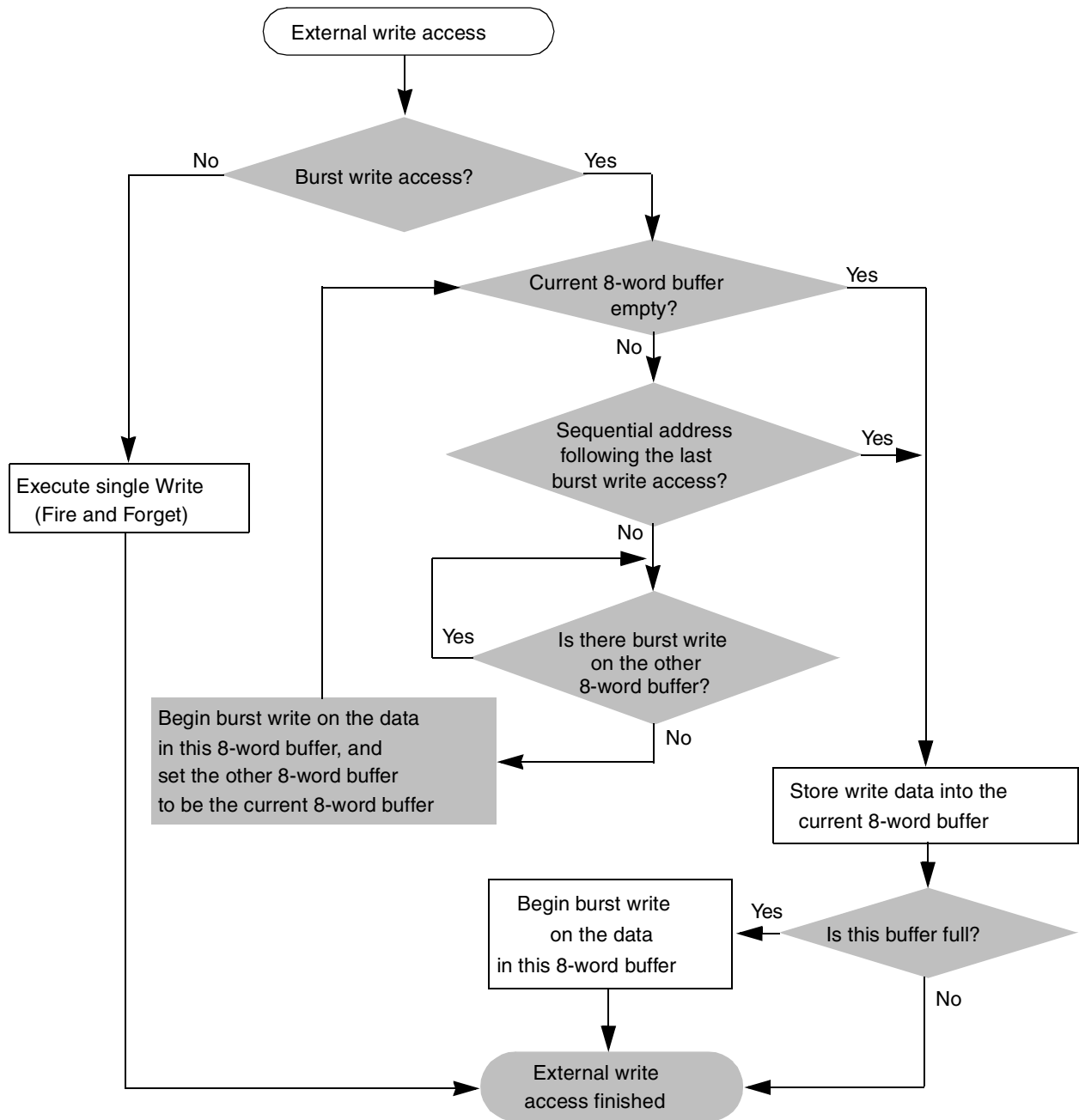


Figure 17-3. Write Access Flow

### 17.3.3.1 Single Write

When a single write happens, the EMC Burst Buffer will execute a *fire-and-forget* write. The EMC Burst Buffer terminates the Shared Bus access (with zero wait states) and registers the write data, allowing the Shared Bus master to continue operations while the EMC Burst Buffer performs the write operation to the

## EMC Burst Buffer

EMC. The EMC Burst Buffer will stall any following request acknowledge accesses from the Shared Bus until the EMC is ready for the next access.

### 17.3.3.2 Burst Write

There are two burst write buffers; one burst write buffer for Shared Bus master 0, and one burst write buffer for Shared Bus master 1. Each Shared Bus master use its own write buffer independently.

Each write buffer has 16 24-bit words alternating (ping-pong style operation) between the two sets of eight words, and there are two 8-word buffers in each write buffer.

The burst write length can be 1 to 8 words. In one write buffer, each 8-word buffer has a 26-bit base address register the same as the read buffer.

When a burst write access occurs, the EMC Burst Buffer checks the current 8-word buffer being used.

- If the 8-word buffer is empty, or the access address is a sequential address following the last burst write access and the buffer is not full, then the 8-word buffer stores the write data in it, terminates the access immediately without involving the EMC, and returns the address of the first write data is stored into the base address register.
- If the 8-word buffer is not empty and is not a sequential address, the sub-buffer stores the write data left in it into external memory through the EMC, by executing a burst operation (the burst length is the number of data words in the buffer). At the same time, the module sets the other 8-word buffer as the current buffer to accept the following write data, and the write access address is recorded in the corresponding base address register.

When a 8-word buffer is storing write data into external memory, the other 8-word buffer can continue to accept write data from the Shared Bus, unless the 8-word buffer is not empty, or if the write access address is not sequential with the last write access address, or if the 8-word buffer is full. If the other 8-word buffer needs to execute a burst operation to flush itself, it must wait until after the other 8-word buffer's burst operation has finished.

# Chapter 18

## S/PDIF—Sony/Philips Digital Interface

### 18.1 Introduction

The Sony/Philips Digital Interface (S/PDIF) audio module is a stereo transceiver that allows the DSP to receive and transmit digital audio over it. The S/PDIF transceiver allows the handling of both S/PDIF channel status (CS) and User (U) data and includes a frequency measurement block that allows the precise measurement of an incoming sampling frequency. A recovered clock is provided to drive both internal and external components in the system such as the on-chip S/PDIF transmitter, ESAI ports, as well as external A/Ds or D/As, with clocking control provided via related registers.

Figure 18-1 shows a block diagram of the S/PDIF transceiver data paths (receiver and transmitter) and its interface.

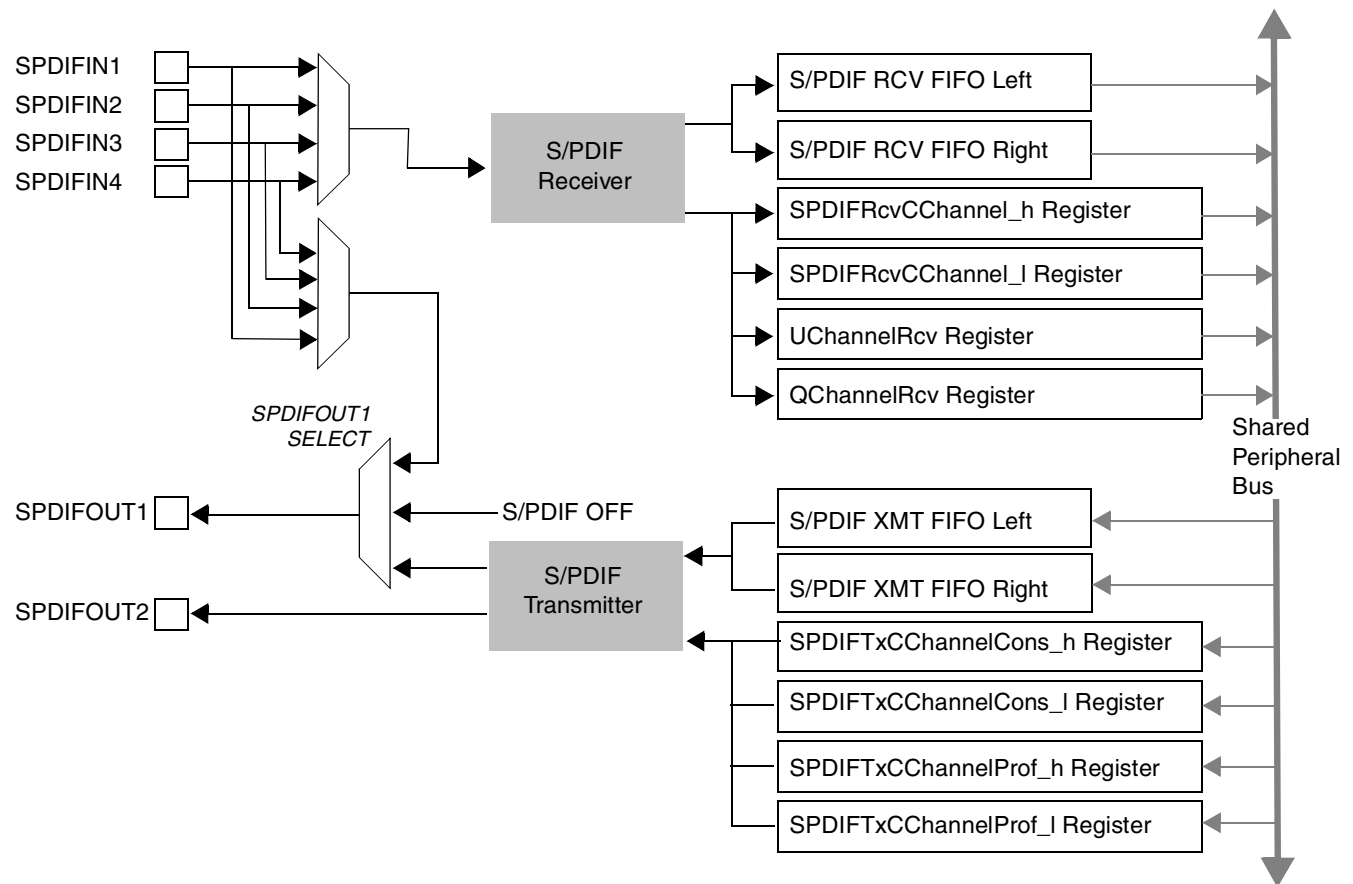


Figure 18-1. S/PDIF Transceiver Data Interface Block Diagram

As is shown in Figure 18-1, the S/PDIF is composed of two parts: S/PDIF receiver and S/PDIF transmitter.

The S/PDIF receiver extracts the audio data from each S/PDIF frame and places the data in a 6-word-deep FIFO. The channel status and user bits are also extracted from each frame and placed in the corresponding registers. The S/PDIF receiver also provides a bypass option for direct transfer of the S/PDIF input signal to the S/PDIF transmitter.

For the S/PDIF transmitter, the audio data is provided by the processor via the SPDIFTxLeft and SPDIFTxRight registers. The channel status bits are also provided via the corresponding registers. The S/PDIF transmitter generates a S/PDIF output bitstream in the biphasic mark format (IEC958), which consists of audio data, channel status, and user bits.

In the S/PDIF transmitter, the IEC958 biphasic bit stream is generated on both edges of the S/PDIF Transmit clock. The S/PDIF Transmit clock is generated by the S/PDIF internal clock generate module and the sources are from outside of the S/PDIF block. For the S/PDIF receiver, it can recover the S/PDIF rcv clock. Both the tx clock and rcv clock are sent to the ASRC. Figure 18-2 shows the clock structure of the S/PDIF transceiver.

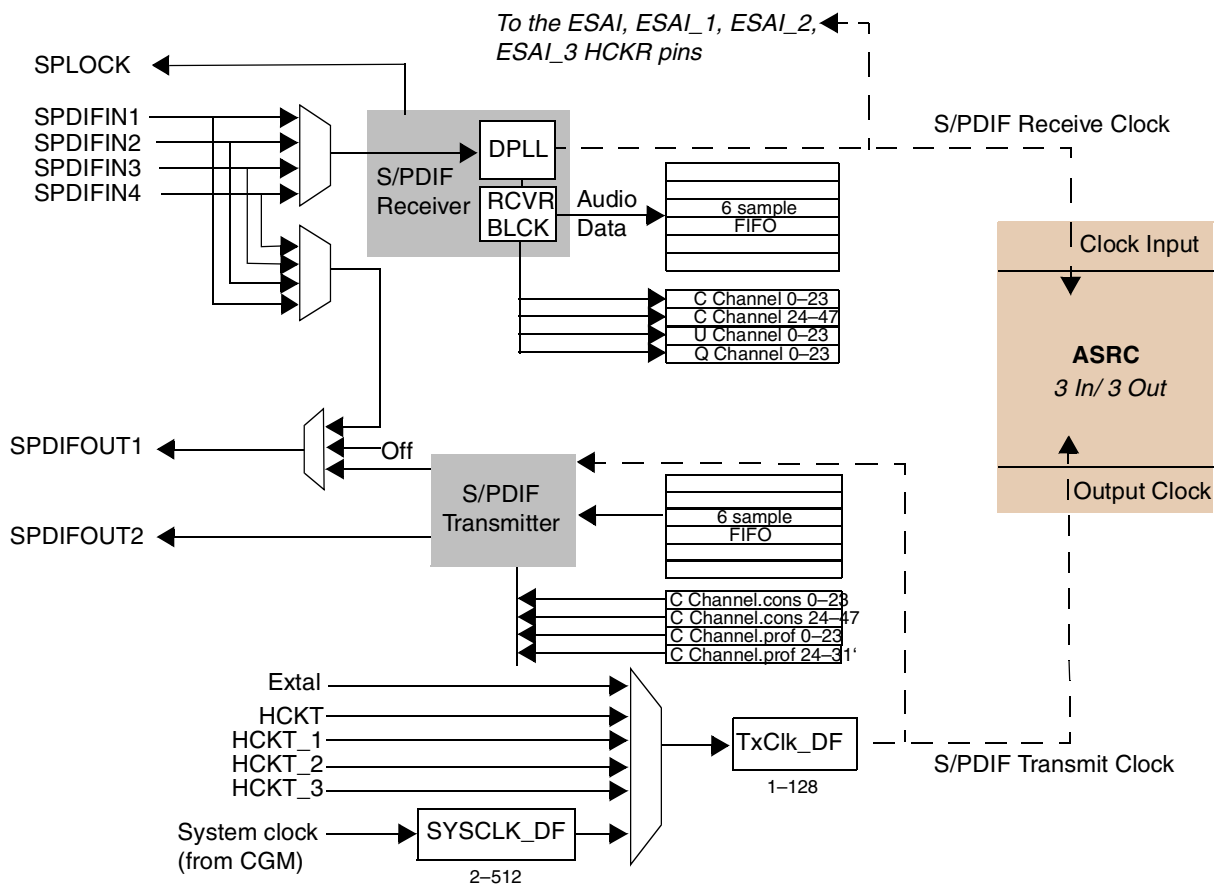


Figure 18-2. S/PDIF Transceiver Clock Diagram

### 18.1.1 Features

The S/PDIF features are as follows:

- The S/PDIF receiver features are as follows:
  - One S/PDIF Receiver
  - Four multiplexed S/PDIF inputs
  - Input sample rate recovery
  - Supports 32 kHz, 44.1 kHz, 48 kHz, 64 kHz, 88.2 kHz, and 96 kHz
  - CD Text Support
  - CS and U bit Recovery
- The S/PDIF transmitter features are as follows:
  - Single S/PDIF Transmitter
  - Two independent S/PDIF outputs.
  - CS bit Support
- Interrupt or DMA control of S/PDIF input and output data
- S/PDIF Receiver to S/PDIF Transmitter Bypass Mode

### 18.1.2 External Signal Description

Table 18-1 lists the signal properties.

**Table 18-1. Signal Properties**

Signal Name	Signal Type	State During Reset	Description
SPDIFIN1	Input	GPIO Disconnected	S/PDIF Input Line 1 IEC958 data in biphas mark format.
SPDIFIN2	Input	GPIO Disconnected	S/PDIF Input Line 2 IEC958 data in biphas mark format.
SPDIFIN3	Input	GPIO Disconnected	S/PDIF Input line 3 IEC958 data in biphas mark format.
SPDIFIN4	Input	GPIO Disconnected	S/PDIF Input Line 4 IEC958 data in biphas mark format.
SPDIFOUT1	Output	GPIO Disconnected	S/PDIF Output Line 1 IEC958 data in biphas mark format. (consumer C channel).
SPDIFOUT2	Output	GPIO Disconnected	S/PDIF Output Line 2 IEC958 data in biphas mark format (professional C channel).
SPLOCK	Output	GPIO Disconnected	S/PDIF Rx DPLL Lock Indicator

**Note:** When the S/PDIF is configured as GPIO, these signals are individually programmable as input, output, or internally disconnected.

### 18.1.3 Memory Map

Table 18-2 shows the S/PDIF memory map.

**Table 18-2. S/PDIF Memory Map**

Address	Access	Register Name	Description	Size Bits	Valid Bits	Reset Value
X: 0xFFFF60	R/W	S/PDIFConfig (SCR)	S/PDIF Configuration Register	24	[23:0]	0x000400
X: 0xFFFF61	R/W	CDTEXT_Control (SRCD)	CDText Configuration Register	24	[1]	0x000000
X: 0xFFFF62	R/W	PhaseConfig (SRPC)	FreqMeas Configuration Register	24	[5:0]	0x000000
X: 0xFFFF63	R/W	InterruptEn (SIE)	Interrupt Enable Register	24	[23:0]	0x000000
X: 0xFFFF64	R-Stat W-Clear	InterruptStat/Clear (SIS/SIC)	Interrupt Status/Clear Register	24	[23:0]	0x000002
X: 0xFFFF65	R	SPDIFRcvLeft (SRL)	S/PDIF Receive Data—left channel	24	[23:0]	0x000000
X: 0xFFFF66	R	SPDIFRcvRight (SRR)	S/PDIF Receive Data—right channel	24	[23:0]	0x000000
X: 0xFFFF67	R	SPDIFRcvCChannel_h (SRCSH)	S/PDIF Receive C channel, bits [47:24]	24	[23:0]	0x000000
X: 0xFFFF68	R	SPDIFRcvCChannel_l (SRCSL)	S/PDIF Receive C channel, bits [23:0]	24	[23:0]	0x000000
X: 0xFFFF69	R	UchannelRcv (SQU)	S/PDIF Receive U channel	24	[23:0]	0x000000
X: 0xFFFF6A	R	QchannelRcv (SRQ)	S/PDIF Receive Q channel	24	[23:0]	0x000000
X: 0xFFFF6B	W	SPDIFTxLeft (STL)	S/PDIF Transmit Left channel	24	[23:0]	0x000000
X: 0xFFFF6C	W	SPDIFTxRight (STR)	S/PDIF Transmit Right channel	24	[23:0]	0x000000
X: 0xFFFF6D	R/W	SPDIFTxCChannelCons_h (STCSCH)	S/PDIF Transmit Cons. C channel, bits [47:24]	24	[23:0]	0x000000
X: 0xFFFF6E	R/W	SPDIFTxCChannelCons_l (STCSCL)	S/PDIF Transmit Cons. C channel, bits [23:0]	24	[23:0]	0x000000
X: 0xFFFF6F	R/W	SPDIFTxCChannelProf_h (STCSPH)	S/PDIF Transmit Prof. C channel, bits [31:24]	24	[7:0]	0x000000
X: 0xFFFF70	R/W	SPDIFTxCChannelProf_l (STCSPL)	S/PDIF Transmit Prof. C channel, bits [23:0]	24	[23:0]	0x000000
X: 0xFFFF71	R	FreqMeas (SRFM)	FreqMeasurement	24	[23:0]	0x000000
X: 0xFFFF74	R/W	SPDIFTxCIk (STC)	Transmit Clock Control Register	24	[23:0]	0x020f00

## 18.2 Register Descriptions

### 18.2.1 S/PDIF Configuration Register (SCR)

Figure 18-3 shows the bit fields of the S/PDIF configuration register.

Address X: 0xFFFF60

Access: Read/Write

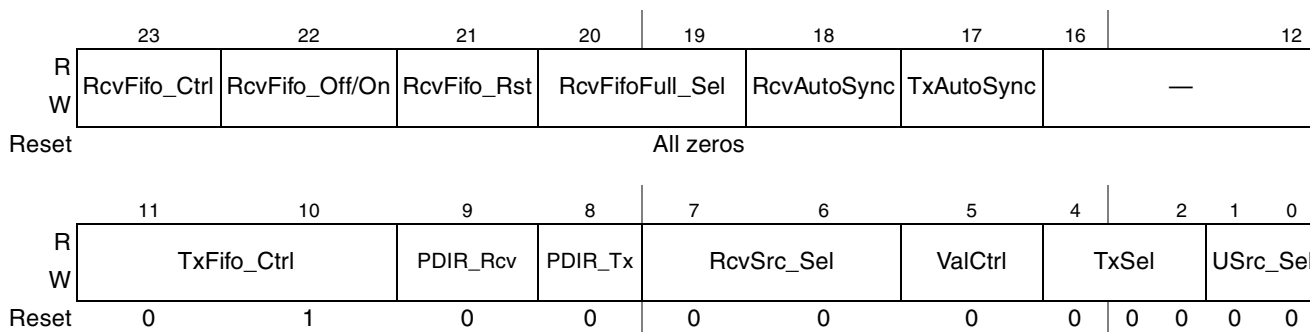


Figure 18-3. S/PDIF Configuration Register (SCR)

Table 18-3 defines the SCR fields.

Table 18-3. S/PDIF Configuration Register (SCR) Fields

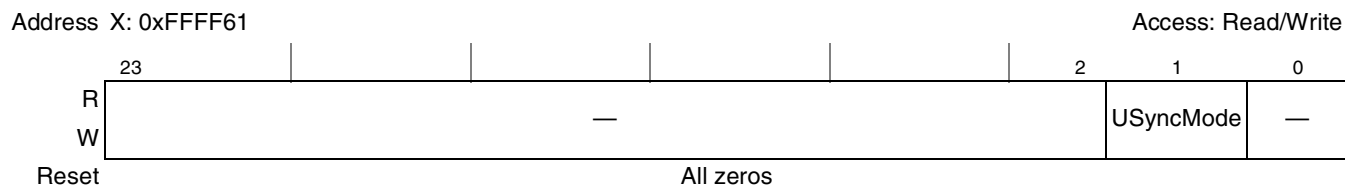
Bits	Name	Description
23	RcvFifo_Ctrl	0 Normal operation 1 Always read zero from rcv data register
22	RcvFifo_Off/On	0 S/PDIF Rcv FIFO is on 1 S/PDIF Rcv FIFO is off. Does not accept data from interface
21	RcvFifo_Rst	0 Normal operation 1 Reset register to 1 sample remaining
20–19	RcvFifoFull_Sel	00 Full interrupt if at least 1 sample in FIFO 01 Full interrupt if at least 2 sample in FIFO 10 Full interrupt if at least 3 sample in FIFO 11 Full interrupt if at least 6 sample in FIFO
18	RcvAutoSync	0 Rcv FIFO auto sync off 1 Rcv FIFO auto sync on
17	TxAutoSync	0 Tx FIFO auto sync off 1 Tx FIFO auto sync on
16–12	—	Reserved
11, 10	TxFifo_Ctrl	00 Send out digital zero on S/PDIF Tx 01 Normal operation 10 Reset to 1 sample remaining 11 Reserved
9	PDIR_Rcv	DMA Receive Request (PDIR1 FIFO full)
8	PDIR_TX	DMA Transmit Request (Transmit FIFO empty)

**Table 18-3. S/PDIF Configuration Register (SCR) Fields (Continued)**

Bits	Name	Description
7–6	RcvSrc_Sel	Rcv Source Select 00 S/PDIF in 1 01 Reserved 10 Reserved 11 Reserved
5	ValCtrl	0 Outgoing Validity always set 1 Outgoing Validity always clear
4–2	TxSel	000 Off and output 0 001 Feed-through SPDIFIN1 010 Reserved 011 Reserved 100 Reserved 101 Normal operation
1–0	USrc_Sel	U Channel Source Select 00 No embedded U channel 01 U channel from S/PDIF receive block (CD mode) 10 Reserved 11 U channel from on chip transmitter

## 18.2.2 CDText Control Register (SRCD)

Figure 18-4 shows the bit fields of the CDText control register.



**Figure 18-4. CDText Control Register (SRCD)**

Table 18-4 defines the SRCD fields.

**Table 18-4. CDText Control Register (SRCD) Field Descriptions**

Bit	Field	Description
23–2	—	Reserved. Bits 23–15 and 7–3 return zeros when read.
1	USyncMode	0 Non-CD data 1 CD user channel subcode
0	—	Reserved



### 18.2.3 PhaseConfig Register (SRPC)

Figure 18-4 shows the bit fields of the PhaseConfig register.

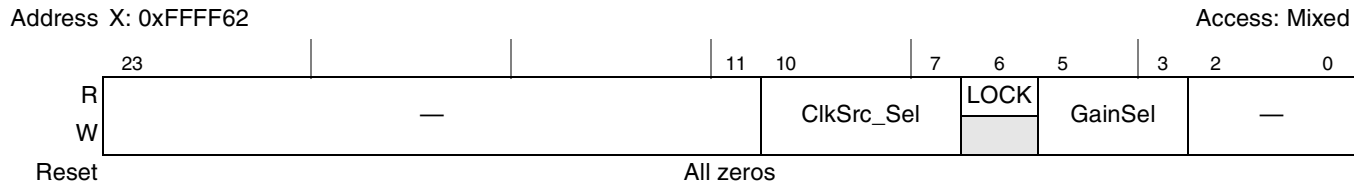


Figure 18-5. PhaseConfig Register (SRPC)

Table 18-5 defines the SRPC fields.

Table 18-5. PhaseConfig Register (SRPC) Field Descriptions

Bit	Field	Description
23–11	—	Reserved. Return zeros when read
10–7	ClkSrc_Sel	Clock source selection: 0000 if (DPLL Locked) SPDIF_RcvClk else EXTAL 0001 if (DPLL Locked) SPDIF_RcvClk else HCKT 0010 if (DPLL Locked) SPDIF_RcvClk else HCKT1 0011 if (DPLL Locked) SPDIF_RcvClk else HCKT2 0100 if (DPLL Locked) SPDIF_RcvClk else HCKT3 0101 EXTAL 0110 HCKT 0111 HCKT1 1000 HCKT2 1001 HCKT3 Others: Reserved
6	LOCK	LOCK bit to show that the internal DPLL is locked, read only
5–3	GainSel	Gain selection: 000 $24 \cdot 2^{10}$ 001 $16 \cdot 2^{10}$ 010 $12 \cdot 2^{10}$ 011 $8 \cdot 2^{10}$ 100 $6 \cdot 2^{10}$ 101 $4 \cdot 2^{10}$ 110 $3 \cdot 2^{10}$
2–0	—	Reserved. Return zeros when read.

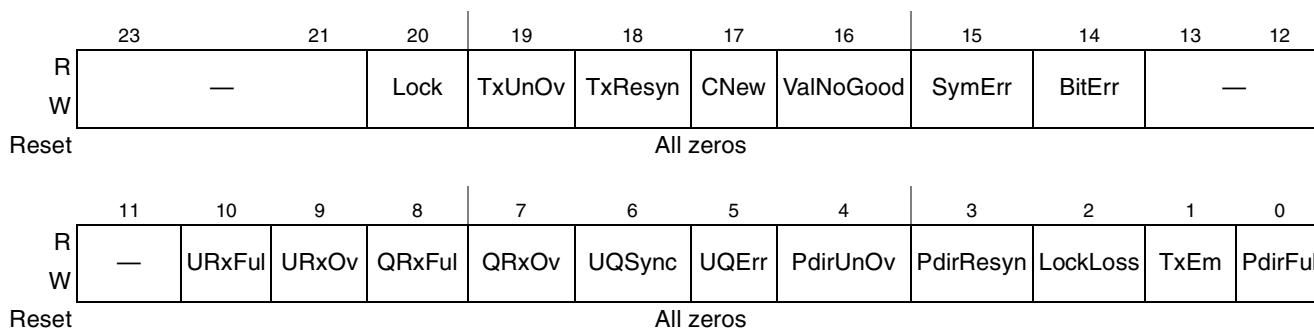
## 18.2.4 Interrupt Registers (SIE, SIS, SIC)

The interrupt registers include InterruptEn, InterruptStat, and InterruptClear.

The InterruptEn register (SIE) provides control over the enabling of interrupts, shown in [Figure 18-6](#).

Address X: 0xFFFF63

Access: Read/Write

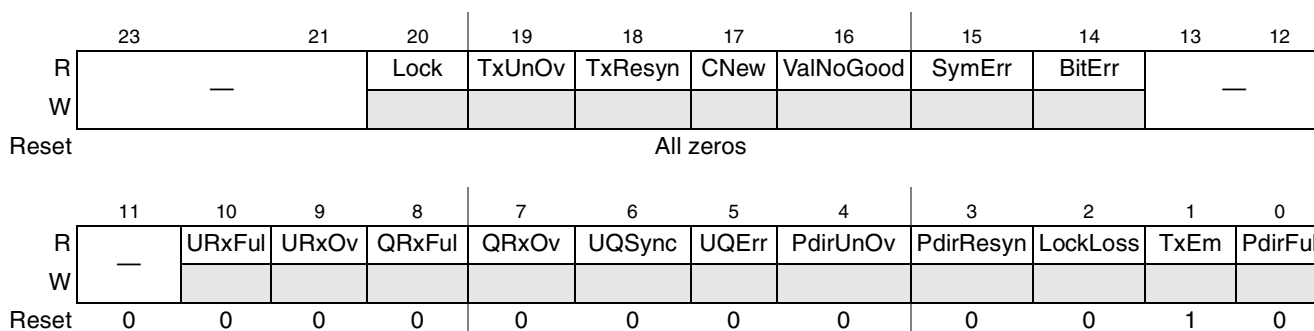


**Figure 18-6. InterruptEn Register (SIE)**

The InterruptStat register (SIS) is a read-only register that provides status on interrupt operations, shown in [Figure 18-7](#).

Address X: 0xFFFF64

Access: Read only

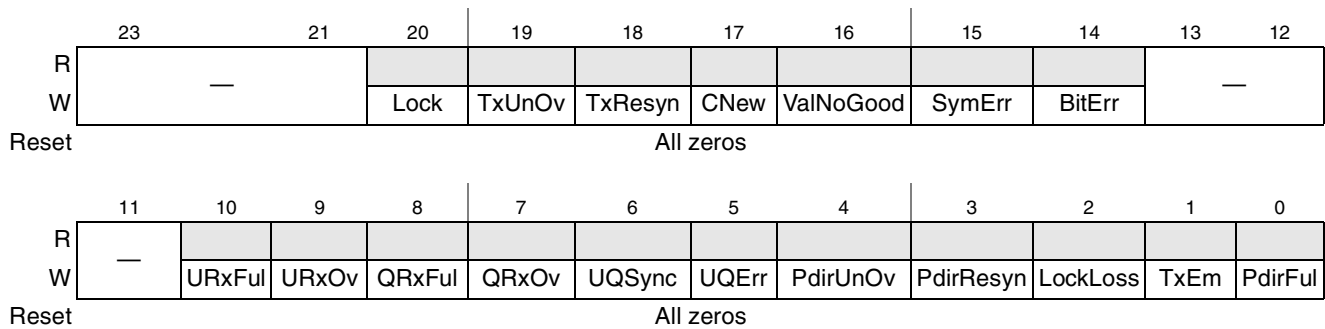


**Figure 18-7. InterruptStat Register (SIS)**

The InterruptClear register (SIC) is a write-only register that is used to clear interrupts, shown in Figure 18-8.

Address X: 0xFFFF64

Access: Write only



**Figure 18-8. InterruptClear Register (SIC)**

Table 18-6 defines the interrupt register fields.

**Table 18-6. Interrupt Register Field Descriptions**

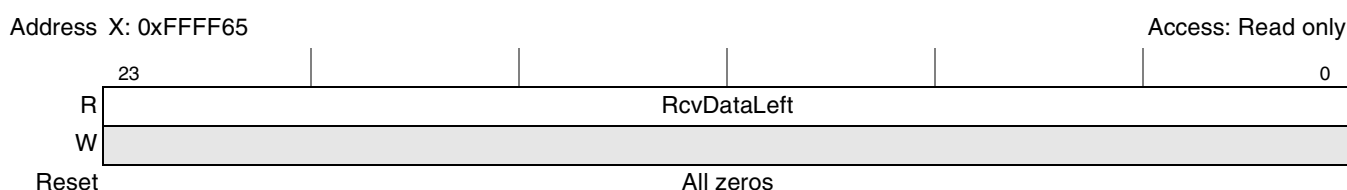
Bit	Field	Description
23–21	—	Reserved. For InterruptStat/Clear, return zeros when read. For InterruptEn, bit 23 also reads zero.
20	Lock	S/PDIF receiver’s DPLL is locked.
19	TxUnOv	S/PDIF transmit FIFO is under/overrun.
18	TxResyn	S/PDIF transmit FIFO resync.
17	CNew	S/PDIF receive change in value of control channel.
16	ValNoGood	S/PDIF validity flag is no good.
15	SymErr	S/PDIF receiver found illegal symbol.
14	BitErr	S/PDIF receiver found parity bit error.
13–11	—	Reserved. Return zeros when read.
10	URxFul	UChannel receive register is full. The URxFul bit cannot be cleared using the IntClear register. To clear the URxFul bit, read from U RCV register.
9	URxOv	UChannel receive register is overrun.
8	QRxFul	QChannel receive register is full. The QRxFul bit cannot be cleared using the IntClear register. To clear the QRxFul bit, read from the Q RCV register.
7	QRxOv	QChannel receive register is overrun.
6	UQSync	U/Q Channel sync is found.
5	UQErr	U/Q Channel framing error.
4	PdirUnOv	Processor data input underrun/overrun occurred.
3	PdirResyn	Processor data input resync.

**Table 18-6. Interrupt Register Field Descriptions (Continued)**

Bit	Field	Description
2	LockLoss	S/PDIF receiver loss of lock has occurred.
1	TxEm	S/PDIF transmit FIFO is empty. The TxEm bit cannot be cleared using the IntClear register. To clear the TxEm bit, write to the TX fifo.
0	PdirFul	Processor data input is full. The PdirFul bit cannot be cleared using the IntClear register. To clear the PdirFul bit, read from the PDIR FIFO.

## 18.2.5 S/PDIF Reception Registers

Figure 18-9 shows the bit fields of the SPDIFRcvLeft register.



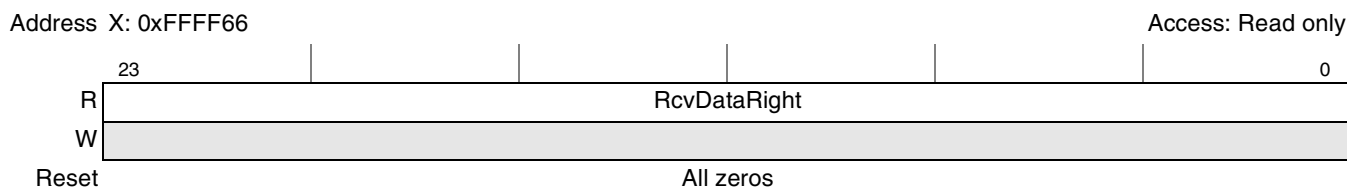
**Figure 18-9. SPDIFRcvLeft Register (SRL)**

Table 18-7 defines the SRL fields.

**Table 18-7. SPDIFRcvLeft Register (SRL) Fields**

Bit	Field	Description
23–0	RcvDataLeft	Processor receives S/PDIF data left.

Figure 18-10 shows the bit fields of the SPDIFRcvRight register.



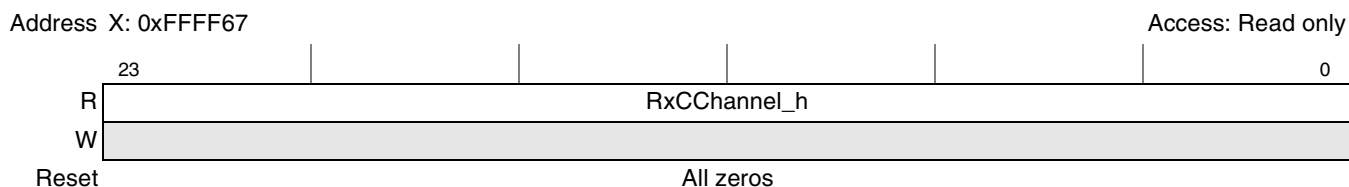
**Figure 18-10. SPDIFRcvRight Register (SRR)**

Table 18-8 defines the SRR fields.

**Table 18-8. SPDIFRcvRight Register (SRR) Fields**

Bit	Field	Description
23–0	RcvDataRight	Processor receives S/PDIF data right.

Figure 18-11 shows the bit fields of the SPDIFRxCCchannel\_h register.



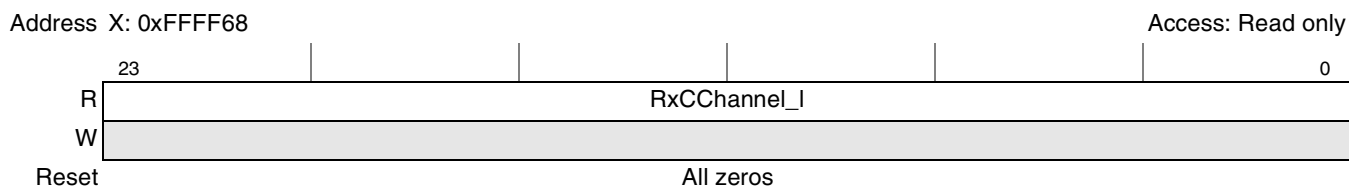
**Figure 18-11. SPDIFRxCCchannel\_h Register (SRCSH)**

Table 18-9 defines the SRCSH fields.

**Table 18-9. SPDIFRxCCchannel\_h Register (SRCSH) Fields**

Bit	Field	Description
23–0	RxCCchannel_h	S/PDIF receive C channel register Contains the first 24 bits of C channel without interpretation.

Figure 18-12 shows the bit fields of the SPDIFRxCCchannel\_l register.



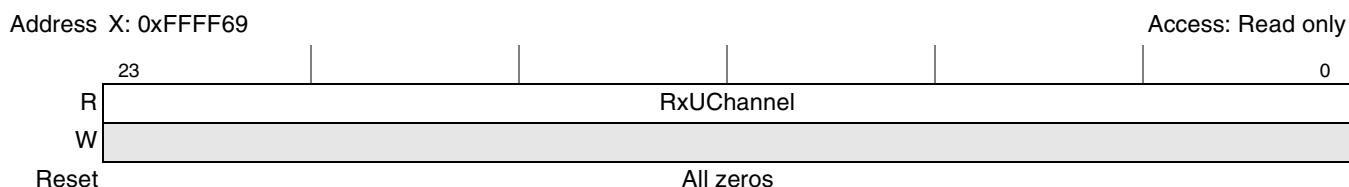
**Figure 18-12. SPDIFRxCCchannel\_l Register (SRCSL)**

Table 18-10 defines the SRCSL fields.

**Table 18-10. SPDIFRxCCchannel\_l Register (SRCSL) Fields**

Bit	Field	Description
23–0	RxCCchannel_l	S/PDIF receive C channel register Contains the next 24 bits of C channel without interpretation.

Figure 18-13 shows the bit fields of the S/PDIF QchannelRcv register.



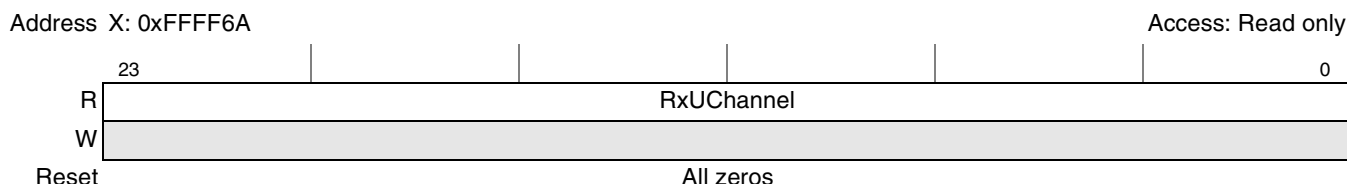
**Figure 18-13. S/PDIF QchannelRcv Register (SRQ)**

Table 18-11 defines the SRU fields.

**Table 18-11. S/PDIF UchannelRcv Register (SRU) Fields**

Bit	Field	Description
23–0	RxUChannel	S/PDIF receive U channel register Contains the next 3 U channel bytes.

Figure 18-14 shows the bit fields of the S/PDIF QchannelRcv register.



**Figure 18-14. S/PDIF QchannelRcv Register (SRQ)**

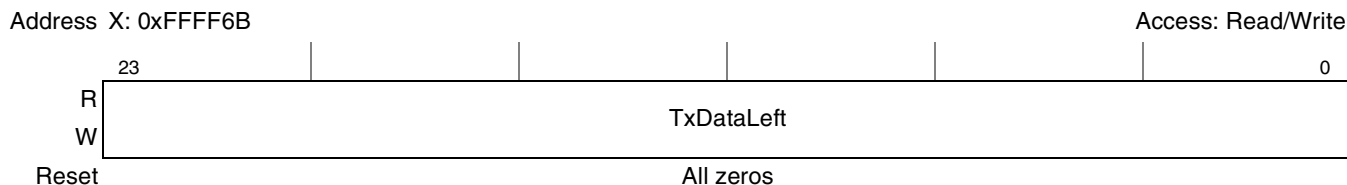
Table 18-12 defines the SRQ fields.

**Table 18-12. S/PDIF QchannelRcv Register (SRQ) Fields**

Bit	Field	Description
23–0	RxQChannel	S/PDIF receive Q channel register Contains the next 3 Q channel bytes.

## 18.2.6 S/PDIF Transmission Registers

Figure 18-15 shows the bit fields of the SPDIFTxLeft register.



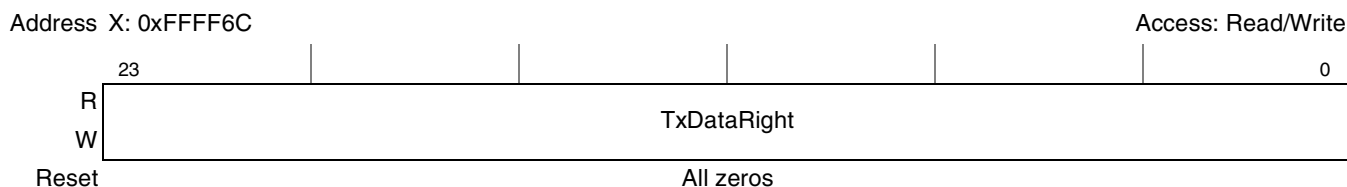
**Figure 18-15. SPDIFTxLeft Register (STL)**

Table 18-13 defines the STL fields.

**Table 18-13. SPDIFTxLeft Register (STL) Fields**

Bit	Field	Description
23–0	TxDataLeft	S/PDIF transmit left channel data It is write-only, and always returns zeros when read.

Figure 18-16 shows the bit fields of the SPDIFTxRight register.



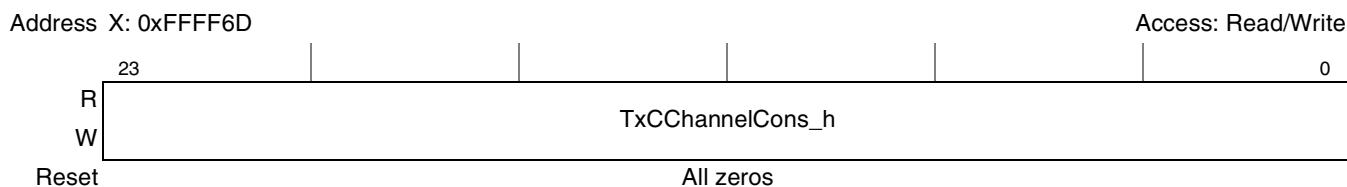
**Figure 18-16. SPDIFTxRight Register (STR)**

Table 18-14 defines the STR fields.

**Table 18-14. SPDIFTxRight Register (STR) Fields**

Bit	Field	Description
23–0	TxDataRight	S/PDIF transmit right channel data It is write-only, and always returns zeros when read.

Figure 18-17 shows the bit fields of the SPDIFTxCChannelCons\_h register.



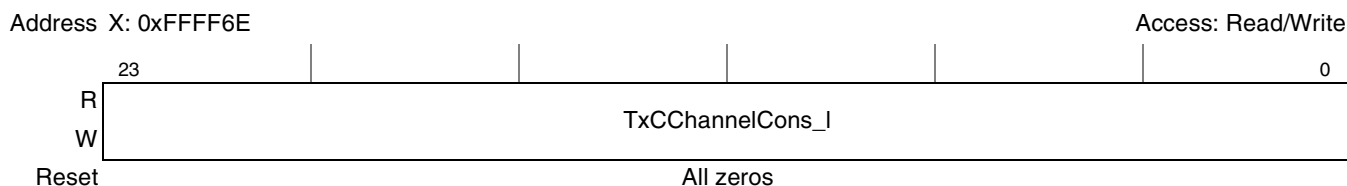
**Figure 18-17. SPDIFTxCChannelCons\_h Register (STCSCH)**

Table 18-15 defines the STCSCH fields.

**Table 18-15. SPDIFTxCChannelCons\_h Register (STCSCH) Fields**

Bit	Field	Description
23–0	TxCChannelCons_h	S/PDIF Transmit Cons. C channel data Contains first 24 bits without interpretation. When read, it returns the latest data written by the processor.

Figure 18-18 shows the bit fields of the SPDIFTxCChannelCons\_l register.



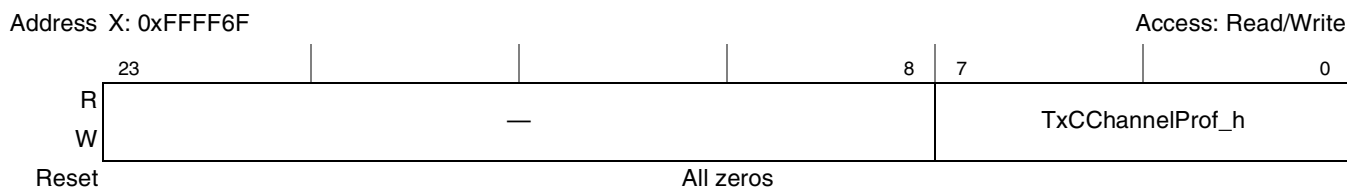
**Figure 18-18. SPDIFTxCChannelCons\_l Register (STCSCL) Register**

Table 18-16 defines the STCSCL fields.

**Table 18-16. SPDIFTxChannelCons\_I Register (STCSCL) Fields**

Bit	Field	Description
23–0	TxCChannelCons_I	S/PDIF transmit consumer C channel data Contains the next 24 bits without interpretation. When read, it returns the latest data written by the processor.

Figure 18-19 shows the bit fields of the SPDIFTxChannelProf\_h register.



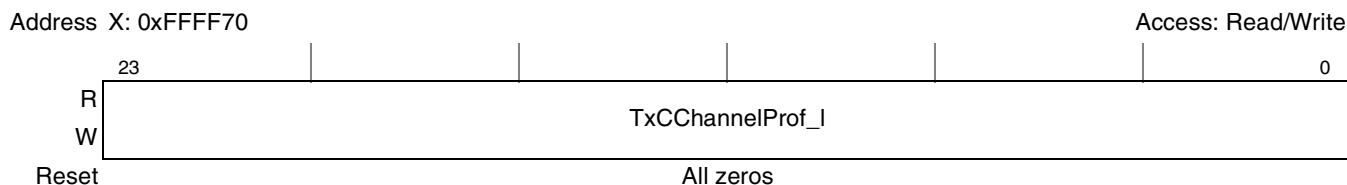
**Figure 18-19. SPDIFTxChannelProf\_h Register (STCSPH)**

Table 18-17 defines the STCSPH fields.

**Table 18-17. SPDIFTxChannelProf\_h Register (STCSPH) Fields**

Bit	Field	Description
23–8	—	Reserved. Returns zeros when read.
7–0	TxCChannelProf_h	S/PDIF transmit professional C channel data Contains the first 8 bits without interpretation. When read, it returns the latest data written by the processor.

Figure 18-20 shows the bit fields of the SPDIFTxChannelProf\_I register.



**Figure 18-20. SPDIFTxChannelProf\_I Register (STCSPL)**

Table 18-18 defines the STCSPL fields.

**Table 18-18. SPDIFTxChannelProf\_I Register (STCSPL) Fields**

Bit	Field	Description
23–0	TxCChannelProf_I	S/PDIF transmit Prof. C channel data Contains the next 24 bits without interpretation. When read, it returns the latest data written by the processor.



### 18.2.7 S/PDIF FreqMeas Register (SRFM)

Figure 18-21 shows the bit fields of the S/PDIF FreqMeas register.



Figure 18-21. S/PDIF FreqMeas Register (SRFM)

Table 18-19 defines the SRFM fields.

Table 18-19. S/PDIF FreqMeas Register (SRFM) Fields

Bit	Field	Description
23–0	FreqMeas	Frequency measurement

### 18.2.8 SPDIFTxClk Register (STC)

The SPDIFTxClk Control register includes the means to select the transmit clock and frequency division, shown in Figure 18-22.

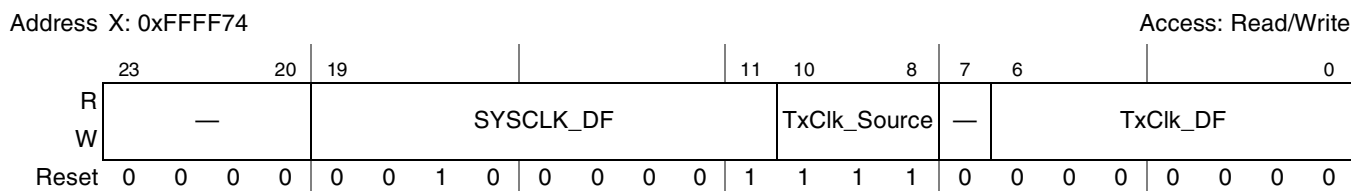


Figure 18-22. SPDIFTxClk Register (STC)

Table 18-20 defines the STC fields.

Table 18-20. SPDIFTxClk Register (STC) Fields

Bit	Field	Description
23–20	—	Reserved. Return zeros when read.
19–11	SYSCLK_DF	System Clock Divider Factor, 2–512. 9'd0 no clock signal 9'd1 divider factor is 2 ... 9'd511 divider factor is 512
10–8	TxClk_Source	000 EXTAL input, 001: HCKT input, 010 HCKT1 input 011 HCKT2 input 100 HCKT3 101 Frequency divided system clock input

**Table 18-20. SPDIFTxClk Register (STC) Fields (Continued)**

Bit	Field	Description
7	—	Reserved. Returns zero when read.
6–0	TxCIk_DF	Divider factor (1-128) 4'd0 divider factor is 1 4'd1 divider factor is 2 ... 4'd127 divider factor is 128

## 18.3 S/PDIF Receiver

The S/PDIF receiver extracts the audio data from each S/PDIF frame and places the data in a 6-deep FIFO. The channel status and user bits are also extracted from each frame and placed in corresponding registers. The S/PDIF receiver also provides a bypass option for direct transfer of the S/PDIF input signal to the S/PDIF transmitter.

The S/PDIF receiver handles the main data audio stream and recovers the bit clock from the S/PDIF input signal. The sample rate can be determined from the frequency measuring block. Additionally, the receiver supports the S/PDIF C and U channels. The S/PDIF C and U channel data is interfaced directly to memory-mapped registers. The input data is sent via a 6-deep FIFO to the memory-mapped data registers. All the data registers are controlled by the Interrupt Control Block and transferred to the memory-mapped IP bus.

The following functions are performed by the S/PDIF receiver:

- Audio Data Reception—see [Section 18.3.1, “Audio Data Reception”](#)
- Channel Status bits Reception—see [Section 18.3.2, “Channel Status Reception”](#)
- User Channel bits Reception—see [Section 18.3.3, “User Bit Reception”](#)
- Validity Flag Reception—see [Section 18.3.4, “Validity Flag Reception”](#)
- S/PDIF Receiver Exception support—see [Section 18.3.5, “S/PDIF Receiver Interrupt Exception Definition”](#)
- S/PDIF Lock Detection

### 18.3.1 Audio Data Reception

The S/PDIF Receiver block extracts the audio data from the IEC958 stream, and outputs this via a 6-deep FIFO to the memory-mapped registers SPDIFRcvLeft and SPDIFRcvRight. Data from the S/PDIF receiver is buffered in the receive FIFO, and can be read by any DSP Core from the memory-mapped registers.

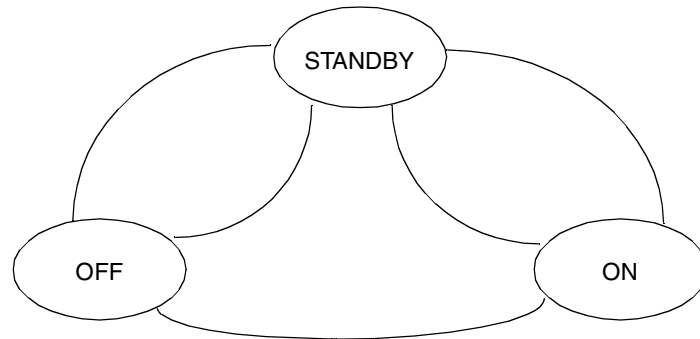
#### 1. S/PDIF Receiver Data Registers—Behavior on Overrun and Underrun

The S/PDIF Data Receive registers (SPDIFRcvLeft and SPDIFRcvRight) have different FIFOs for the left and right channels. As a result, there is always the possibility that the left and right FIFOs may go out of sync due to FIFO underruns and overruns that affect only one part (left or right) of any FIFO. To prevent this from happening, two mechanisms to prevent mismatch between the FIFOs are available:

- If a S/PDIF Data Receive FIFO overrun occurs on, for example, the right half of the FIFO, the sample that caused the overrun is not written to the right half of the FIFO (due to overrun). Special hardware will make sure the next sample is not written to the left half of the FIFO.
- If the overrun occurs on the left half of the FIFO, the next sample is not written to the right half of the FIFO.

**2. S/PDIF Receiver Data Registers—Automatic Resynchronization of FIFOs**

An automatic FIFO resynchronization feature is available, which can be enabled and disabled separately for every FIFO. If the automatic FIFO resynchronization feature is enabled, the hardware will check if the left and right FIFO are in sync— if the left and right FIFO are not in sync, it will set the filling pointer of the right FIFO to be equal to the filling pointer of the left FIFO.



**Figure 18-23. FIFO Auto-Resync Controller State Machine**

The operation can be explained from the state diagram in [Figure 18-23](#). Every FIFO auto-resync controller has a state machine with 3 states: OFF, STANDBY and ON. In the ON state, the filling of the left FIFO is compared with the filling of right FIFO, and if they are not equal, the right FIFO is made equal to the left FIFO, and an interrupt is generated.

The controller will stay in the OFF state when the automatic FIFO resynchronization feature is disabled. When the automatic FIFO resynchronization feature is not disabled, the state machine will go to the OFF state on any DSP Core read or write to the FIFO. The controller will go from ON or OFF to STANDBY on any left sample read from the S/PDIF Tx FIFOs, or on any left sample write to the S/PDIF Rcv FIFOs. The controller will go from STANDBY to ON on any right sample read from the S/PDIF Tx FIFO, or on any right sample write to the S/PDIF Rcv FIFO. There is a control bit in the SPDIFConfig register to enable/disable the automatic FIFO resynchronization feature for the S/PDIF Rcv FIFO and S/PDIF Tx FIFO.

**NOTE**

The automatic FIFO resynchronization feature can be switched on, and will avoid all mismatches between left and right FIFOs, if the software obeys the following rules:

- When the left data is read or written to the left FIFO, in the same place of the program, data must be read or written to the right FIFO. Maximum time difference between left and right operations is 1/2 sample clock. (For example, if the sample frequency is 44 kHz, the

maximum time difference between left and right operations is approximately 10 microseconds. For 88 kHz, the maximum time difference between left and right operations is approximately 5 microseconds.)

- Write/read data to FIFOs must be at least 2 samples. If there is a mismatch between Left and Right FIFOs, the resync logic may go on only one sample clock after the last data is read or written to the FIFO. Also acceptable is polling the FIFO, if at least part of the time 2 samples will be read/written to it.

### 3. S/PDIF Receiver—Additional Features

There are three exceptions associated with the S/PDIF Receivers FIFOs, as follows:

- Full
- Under/overflow
- Resync

When the “full” condition is set for DSP Core data input registers, the DSP Core should read data from the FIFO *before* overrun occurs. When “full” is set and the FIFO contains for example 6 samples, it is acceptable for the software:

- To read the first 6 samples from the LEFT address, followed by 6 samples from the RIGHT address,
- Or to read 6 samples from the RIGHT address, followed by 6 samples from the LEFT address,
- Or to read 1 sample LEFT, followed by 1 sample RIGHT repeated 6 times.

There is no order specified.

The implementation for SPDIFRcv is a double FIFO, with one FIFO for left and one FIFO for right. “Full” is set when both FIFOs are full. “Underrun” and “overflow” are set when one of the FIFOs are underrun or overflow. The resync interrupt occurs when the hardware has to take special action to resynchronize the left and right FIFOs.

The FIFO level (at which the “full” interrupt is generated) is programmable via the Full Select field in the SPDIFConfig register.

### 4. Rcv FIFO On and Rcv FIFO Reset

Two additional control fields of the S/PDIF Rcv Fifo are the On/Off Select and FIFO Reset fields. If On/Off Select is set to Off, all zero will be read from the FIFO, regardless of the data that was received over the S/PDIF interface.

If FIFO Reset is set, the FIFO is blocked at “1 sample in FIFO.” In this situation, the full interrupt will be On if FullSelect is set to “00”. If FullSelect is set to any other value, interrupt will be Off. The other interrupts are always Off.

### 18.3.2 Channel Status Reception

A total of 48 channel status bits are received in two registers. No interpretation is performed by the S/PDIF receiver module. Channel Status Bits are ordered first bit left. CS-channel MSB bit “0” is located in bit position 23 in the memory-mapped register SPDIFRxCChannel\_h. CS-channel bit “23” is considered bit 0 in the register. C-channel bits 24–47 is seen as MSB–LSB bits of register SPDIFRxCChannel\_l.

When the value of a new S/PDIF “CS” channel status frame is loaded in the register, a new interrupt is generated. The interrupt is cleared when the DSP Core writes the corresponding bit in the InterruptStat register.

### 18.3.3 User Bit Reception

There are two modes for User Channel reception, CD and non-CD, which is selected using the USyncMode bit (bit 1 of the CDText Control register).

#### 1. Behavior of User Channel receive interface on incoming CD User Channel Sub-code in S/PDIF receiver

This mode is selected if USyncMode, bit 1 in the CDText control register is set to “1”. The CD subcode stream embedded into the S/PDIF User channel consists of a sequence of packets. Every packet is made up of 98 symbols. The first two symbols of every packet are “sync symbols”, while the other 96 symbols are “data symbols.”

Any sequence found in the S/PDIF U-channel stream starting with a leading “1”, followed by 7 information bits, is recognized as a “data symbol.” Subsequent data symbols are separated by “pauses.” During the “pause,” “zero bits” are seen on the S/PDIF U-channel.

Data symbols come in MSB first. The MSB is the leading “1”.

When a “long pause” is seen between two subsequent “data symbols,” the S/PDIF receiver will assume the reception of one or more “sync symbols.” See [Table 18-21](#).

**Table 18-21. Sync Control Bits**

Number of U Channel Zero Bits	Corresponding Number of Sync Symbols
0–1	Unpredictable, not allowed
2–10	0
11–22	1
23–34	2
35–46	3
>45	Unpredictable, not allowed

The recognition of the number of sync symbols is derived from the fact that the U-channel transmitter in the CD channel decoder will transmit one symbol on average every 12 S/PDIF channel bits. At this average rate, there is a tolerance of maximum 5%.

The S/PDIF receiver is tolerant of symbol errors. Due to the physical nature of the transmission of the data over the CD disc, not more than 1 out of any 5 consecutive user channel symbols may be

in error. The error may cause a change in data value, which is not treated by this interface, or it may cause a data symbol to be seen as a sync symbol, or cause a sync symbol to be seen as a data symbol. However, not more than 1 out of any 5 consecutive user channel symbols should be affected in this way.

The S/PDIF user channel circuitry recognizes the 98-symbol packet structure, and sends the 96-symbol payload to the DSP Core application. The 96-symbol payload is transmitted to the DSP through two registers:

- **The UchannelRcv register**

In the S/PDIF UchannelRcv register, data is presented 3 symbols at a time to the DSP Core. Every time 3 new valid symbols are received on the S/PDIF U-Channel, the UChannelRcvFull interrupt is asserted. For one 98-symbol packet, 96 symbols are carried across the S/PDIF UchannelRcv register. To transfer all of this data, 32 UChannelRcvFull interrupts are generated.

- **The QChannelRcv register**

In the QChannelRcv register, only the Q-bit of the packet is accumulated. The operation of the QChannelRcv register is similar to the UchannelRcv register. Because only the Q-bit is transferred, only 96 Q-bits are transferred for any 98-symbol packet. To transfer this data, 4 QChannelRcvFull interrupts are generated. When a QChannelRcvFull interrupt occurs, it is coincident with a UChannelRcvFull interrupt. There is only one QChannelRcvFull interrupt for every 8 UChannelRcvFull interrupts. The convention is that the most significant data is transmitted first, and is left-aligned in the registers.

Timing regarding the packet boundary is extracted by hardware. The last UChannelRcvFull interrupt corresponding to a given packet should be coincident with the last QChannelRcvFull interrupt. In this last U, Q channel interrupt, symbols 95-98 and Q-channel bits 67-98 are received. The interrupts are coincident with UQSyncFound, flagging the last symbols of the current frame.

When the start of the new packet is found before the current packet is complete (less than 98 symbols in the packet), the UQFrameError interrupt is set. The application software should read the UchannelRcv and QchannelRcv registers, discard the values, and assume the start of a new packet.

As already said, packet sync extraction is tolerant for single-symbol errors. Packet sync detection is based on the recognition of the sequence *data-sync-sync-data* in the symbol stream, because this is the only syncing sequence that is not affected by single errors. If the sync symbols are not found 98 symbols after the previous occurrence, it is assumed to be destroyed by channel error, and a new sync symbols is interpolated.

Normally, only data bytes are passed to the application software. Every databyte will have its most significant bit set. If sync symbols are passed to the application software, they are seen as all-zero symbols. Sync symbols can only end up in the data stream due to channel error.

## 2. Behavior of the user channel receive interface on incoming non-CD data

This mode is selected if the UsyncMode bit (bit 1 in CD Text control register) is set to “0”.

In non-CD mode, the S/PDIF User channel stream is recognized as a sequence of “data symbols.” No packet recognition is done. Any sequence found in the S/PDIF U-channel stream starting with

a leading “1”, followed by 7 information bits, is recognized as a “data symbol.” Subsequent data symbols are separated by “pauses.” During the “pause,” “zero bits” are seen on the S/PDIF U-channel.

Three consecutive data symbols seen in the S/PDIF U-Channel stream are grouped together into the UchannelRcv register. The first symbol is left-aligned; the last symbol is right-aligned. When the UchannelRcv register contains 3 new data symbols, the UChannelRcvFull interrupt is asserted. In this mode, the operation of QchannelRcv and associated interrupt QchannelRcvFull is “Reserved, Undefined.” Also “Reserved, Undefined” is the operation of UQFrameError and UQSyncFound.

The U-channel is extracted, and output by the S/PDIF Rcv block on SPDIFRcvUChannel-Stream. When an incoming S/PDIF data parity error or bit error is detected, and if the next S/PDIF word for that channel is error-free, the S/PDIF word in error is replaced with the average of the previous word and next word. When an incoming S/PDIF data parity error or bit error is detected, and the next S/PDIF word is in error, the previous S/PDIF word is repeated.

### 18.3.4 Validity Flag Reception

An interrupt is associated with the Validity flag (interrupt 16—SPDIFValNoGood). This interrupt is set every time a frame is seen on the S/PDIF interface with the validity bit set to “invalid”.

### 18.3.5 S/PDIF Receiver Interrupt Exception Definition

There are several S/PDIF exceptions that will trigger an interrupt. See [Table 18-22](#).

**Table 18-22. S/PDIF Receiver Interrupt Exceptions**

S/PDIF Exception	Description
RcvChannelNew	Set when the SPDIFRcvCChannel_h and SPDIFRcvCChannel_l registers are updated. The SPDIFRcvCChannel_h and SPDIFRcvCChannel_l registers are updated for every new C-Channel received. The Control Status channel change exception is reset on a write to the InterruptClear register.
RcvIllegalSymbol	Set on reception of an illegal symbol during S/PDIF receive. Reset by writing to the InterruptClear register. <sup>1</sup>
RcvParityError	Set on reception of bit error. (Parity bit does not match). Reset on write to the InterruptClear register.
Rcv FIFO Full	Set when the S/PDIF receive data FIFO is full.
Rx Over/Under	Set when there is an overrun/underrun on the S/PDIF receive data FIFO.
Rx Resync	Set when a resynchronization event occurs on the S/PDIF receive data FIFO.
RxUChannelFull	Set when the next 24 bits of U channel code are available.
RxQChannelOver	Set when the Q channel buffer is overrun.
RxUChannelOver	Set when the U channel buffer is overrun.
RxQChannelFull	Set when the next 24 bits of Q channel code are available.
RxUQSyncFound	Set when the UQ channel sync is found.
RxUQFrameError	Set when a UQ frame error is found.

- <sup>1</sup> The S/PDIF input is a biphasic/mark modulated signal. The time between any two successive transitions of the S/PDIF signal is always 1, 2 or 3 S/PDIF symbol periods long. The S/PDIF receiver will parse the stream, and split it in so-called symbols. It recognizes s1, s2 and s3 symbols, depending on the length of the symbols. Not all sequences of these symbols are allowed. For example, a sequence s2-s1-s1-s1-s2 cannot occur in a no-error S/PDIF signal. If the receiver finds such an illegal sequence, the illegal symbol interrupt is set. No corrective action is undertaken. When the interrupt occurs, this means that
- (a) The S/PDIF signal is destroyed by noise.
  - (b) The S/PDIF frequency changed.

### 18.3.6 Standards Compliance

The S/PDIF interface is compatible with the Tech 3250-E standard of the European Broadcasting Union, except for clause 6.3.3 and the IEC958-3 Ed2 for relevant topics. The supported input frequency range is 12 kHz up to 96 kHz (fully compliant) and 96 kHz up to 176 kHz. (It can interface with a compliant S/PDIF transmitter within the same cabinet, making reasonable assumptions on jitter added due to interconnecting wire.)

Tolerated jitter on S/PDIF input signals are 0.25 bit peak-to-peak for high frequencies. There is no jitter limit for low frequencies. The user channel extraction in CD mode can cope with single-symbol errors, and still retrieve U-channel frames on correct boundaries. This capability is required for reliable reception of CD-Text from some Philips CD channel decoders. This capability was deemed more important than compliance with the IEC958 annex A.3 standard, and for this reason user channel reception is not compliant with IEC958 annex A.3. However, the S/PDIF interface can receive a U channel inserted by a typical CD channel decoder. Also, in this case, it is more robust and tolerant for channel errors than what is required by IEC958 annex A.3.

### 18.3.7 S/PDIF PLOCK Detection and Rxclk Output

Using the high speed system clock, the internal DPLL can extract the bit clock (advanced pulse) from the input bitstream. When this internal DPLL is locked, the LOCK bit of PhaseConfig Register will be set, and the S/PDIF Lock output pin “SPLOCK” will be asserted.

After DPLL has locked, the pulses are generated, and the average pulse rate is  $128 \times$  “the sampling frequency.” (For a 44.1 kHz input sampling frequency, the average pulse rate =  $128 \times 44.1$  kHz.) The pulse signal is used in the FreqMeas circuit to generate the frequency measurement result.

### 18.3.8 Measuring the Frequency of SPDIF\_RcvClk

The internal DPLL can extract the bit clock (advanced plus) from the input bitstream. To do that, it is necessary to measure the frequency of the incoming signal in relationship with the system clock (BUS\_CLK).

Associated with it are two registers, PhaseConfig and FreqMeas. The circuit will measure the frequency of the incoming clock as a function of the BUS\_CLK. The circuit is a second-order filter. The output is a value represented by an unsigned number stored in the 24-bit FreqMeas register, giving the frequency of the source as a function of the BUS\_CLK.

$$\text{FreqMeas}[23:0] = \text{FreqMeas\_CLK} / \text{BUS\_CLK} * 2^{10} * \text{GAIN}.$$



For example, if the GAIN is selected as  $8 \times 2^{10}$  (PhaseConfig[5:3] = 3'b011), the actual result  $\text{FreqMeas\_CLK} / \text{BUS\_CLK}$  is equal to  $\text{FreqMeas}[23:0] / 2^{23}$ .

## 18.4 S/PDIF Transmitter

Audio data for the S/PDIF transmitter is provided by the DSP Core via the SPDIFTxLeft and SPDIFTxRight registers. Clocking for S/PDIF transmitter is from either the Extal pin, HCTK and HCKT1\_3 pin of ESAI0\_3, or system clock. A multiplexer is used to choose the clock source. The S/PDIF transmitter clock source can be divided down as needed using Txclk\_DF. The S/PDIF transmitter output can be chosen from either the S/PDIF transmitter block, directly from the S/PDIF receiver (via the output multiplexer), or disabled.

The S/PDIF transmitter generates a S/PDIF output bitstream in IEC958 biphasic mark format, consisting of audio data, channel status and user bits.

### 18.4.1 Audio Data Transmission

Audio data for the S/PDIF transmitter is provided by the processor via the SPDIFTxLeft and SPDIFTxRight registers. They send audio data to the left and right TX FIFOs. The Tx FIFOs are 6-deep and 24-bits wide (equal to the audio data width).

#### 1. S/PDIF Transmitter Data Registers—Behavior on Overrun, Underrun

The S/PDIF Data Transmit registers (SPDIFTxLeft and SPDIFTxRight) have different FIFOs for the left and right channels. As a result, there is always the possibility that the left and right FIFOs may go out of sync due to FIFO underruns and overruns that affect only one part (left or right) of any FIFO. To prevent this from happening, there are two mechanisms to prevent mismatch between the FIFOs.

If the S/PDIF Tx FIFO underruns, for example on the right half of the FIFO, no sample leaves that FIFO (because it was already empty). Special hardware will make sure that the next sample read from the left FIFO will not leave the FIFO (no read strobe will be generated). If the underrun occurs on the left half of the FIFO, then the next read strobe to the right FIFO is blocked.

#### 2. S/PDIF Transmitter Data Registers—Automatic Resynchronization of FIFOs

See 2 in Section 18.3.1, “Audio Data Reception.”

#### 3. SPDIFTxLeft and SPDIFTxRight Details

Three exceptions are associated with the S/PDIF Tx FIFOs:

- Empty
- Under/Overrun
- Resync

When the Empty condition is set for the DSP Core data output registers, the DSP Core should write data to the FIFO, before underrun occurs. For example, when Empty is set and 6 samples need to be written, it is acceptable for the software

- to write the first 6 samples from the LEFT address, followed by 6 samples from the RIGHT address,

- or to write 1 sample from the LEFT address, followed by 1 sample from the RIGHT address repeated 6 times.

The LEFT address should be written before the RIGHT address. The implementation of all data out FIFOs is a double FIFO, one FIFO for the left and one FIFO for the right. Empty is set when both FIFOs are empty. Underrun and Overrun are set when one of the FIFOs are underrun or overrun. Resync is set when the hardware resynchronizes left and right FIFOs.

On receiving an Underrun or Overrun interrupt, the synchronization between Left and Right words in the FIFOs may be lost. Synchronization will not be lost when the Underrun or Overrun comes from the IEC958 side of the FIFO. If the processor reads or writes more data from the left than from the right (for example), synchronization will be lost. If automatic resynchronization is enabled, and if the software obeys the rules to let this work, resynchronization will be automatic.

## 18.4.2 Channel Status Transmission

A total of 48 Consumer channel status bits are transmitted from two registers. Channel Status Bits are ordered first bit left. CS-channel MSB bit “0” is located in bit position 23 in the memory-mapped register SPDIFTxChannelCons\_h. CS-channel bit “23” is considered bit 0 in the register. C-channel bits 24–47 are seen as MSB–LSB bits of register SPDIFTxChannelCons\_l.

A total of 32 Professional channel status bits are transmitted from one register. Channel Status Bits are ordered first bit left. CS-channel MSB bit “0” is located in bit position 7 in the memory-mapped register SPDIFTxChannelProf\_h. CS-channel bit “7” is considered bit 0 in the register. C-channel bits 8–31 are seen as MSB–LSB bits of register SPDIFTxChannelProf\_l.

## 18.4.3 Validity Flag Transmission

The validity bit setting is selected using bit 5 of the SPDIFConfig register.

# Chapter 19

## Asynchronous Sample Rate Converter

### 19.1 Introduction

Figure 19-1 shows how the ASRC module connects to other modules.

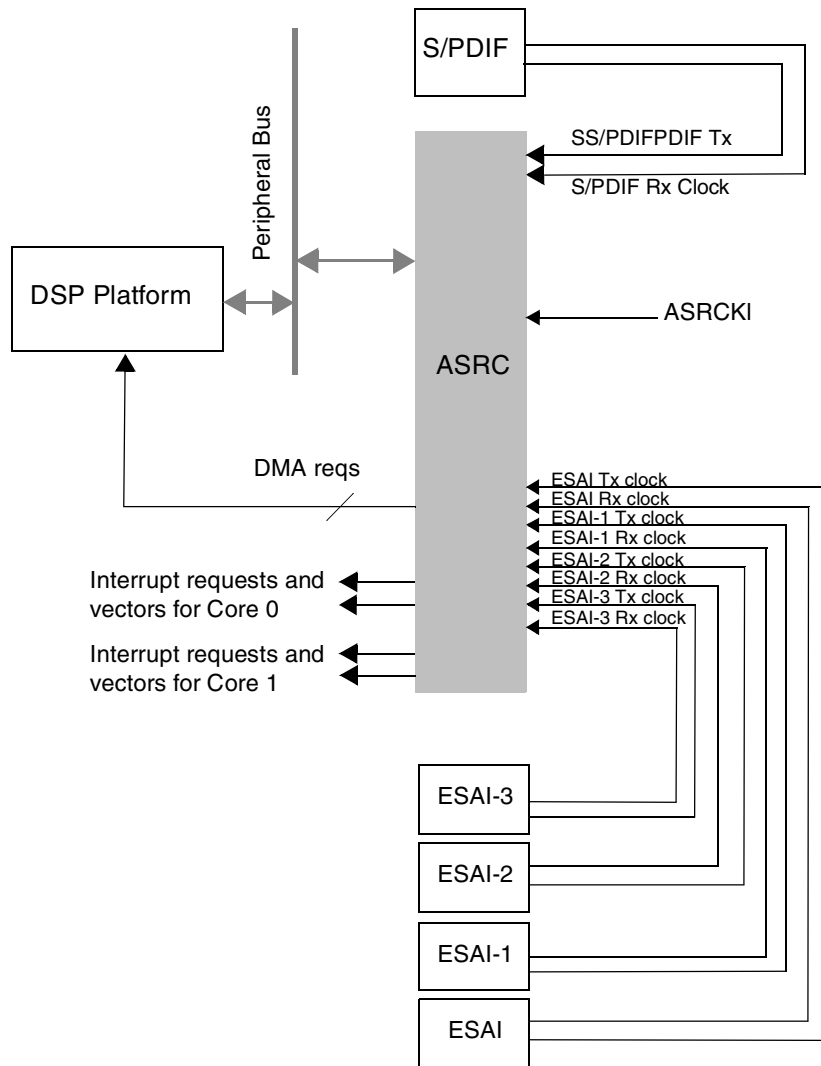


Figure 19-1. ASRC Connections

Figure 19-2 shows the ASRC block diagram.

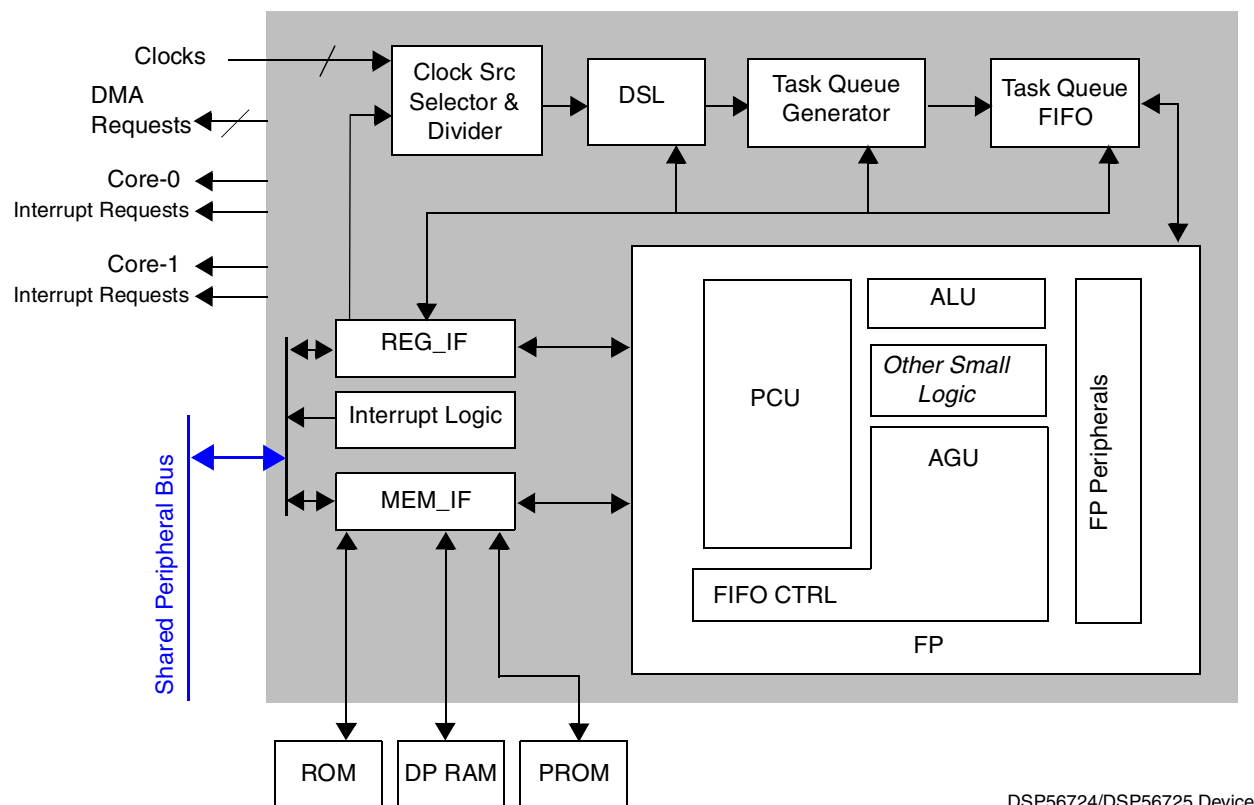


Figure 19-2. ASRC Block Diagram

### 19.1.1 Overview

The incoming audio data to the DSP56724/DSP56725 device may be received from various sources at different sampling rates. Also, the outgoing audio data of DSP56724/DSP56725 may have different sampling rates, and the outgoing audio data can also be associated with output clocks that are asynchronous to the input clocks.

The Asynchronous Sample Rate Converter (ASRC) converts the sampling rate of a signal (associated with an input clock) into a signal associated to a different output clock. The ASRC supports concurrent sample rate conversion of up to 10 channels at about -120 dB THD+N. The sample rate conversion of each channel is associated to a pair of incoming and outgoing sampling rates. The ASRC supports up to 3 sampling rate pairs.

The ASRC is hard-coded implemented, as a co-processor, and requires minimal CPU intervention.

## 19.1.2 Features

**Table 19-1. ASRC Specifications**

Parameters	Test Conditions	Min	Typical	Max	Unit
Channels Supported	—	2	$2 \times n$	10	—
Pairs of Rate Conversion	—	1	—	3	—
THD+N	$120 \text{ MHz} < F_{\text{master}} < 160 \text{ MHz}$	—	-120	—	dB
Dynamic Range	—	—	—	144	dB
Settling Time	—	—	$40^1$	—	ms

**Note:**

<sup>1</sup> The settling time is defined to be the time from audio input to the conversion performance reaching -115 dB THD + N.

Other features include the following:

- Each pair supports 2, 4, 6, 8, or 10 channels
- Designed for rate conversion between 32 kHz, 44.1 kHz, 48 kHz, 64 kHz, 88.2 kHz, 96 kHz, 128 kHz and 192 kHz. The useful signal bandwidth is below 24 kHz
- Other input sampling rates in the range of 8 kHz–200 kHz are also supported, but with less performance.
- Output sampling rates in the range of 8 kHz–200 kHz are also supported but with less performance.
- Automatic accommodation to slow variations in the incoming and outgoing sampling rates
- Linear phase
- Designed for real-time streaming audio usage: the output sampling clock must always be physically available in the system.
- Tolerant to sample clock jitter

Clock/Data Connections are as follows:

- The physical sampling clocks are directly connected to the ASRC module and the ratio estimation of the input clocks with output clocks are done in ASRC hardware when both the input and output sampling clocks are physically available. When this is the case, the rate conversion can be done by configuring the physical clocks.

When the input clock is not physically available, the rate conversion can still be done by setting ideal ratio values in the ASRC interface registers.

- The clock signals come from the following modules:
  - ESAI, receiving clock and transmitting clock
  - ESAI-1, receiving clock and transmitting clock
  - ESAI-2, receiving clock and transmitting clock
  - ESAI-3, receiving clock and transmitting clock
  - S/PDIF, receiving clock and transmitting clock

- Core master clock derivative (default is 5.644 MHz, programmable through external module). The exchange of audio data is done by the DSP Core accessing the ASRC module through registers defined on the Shared Peripheral bus.

### 19.1.3 Modes of Operation

In the description, many registers are used. The definition of the registers and parameters are described in [Section 19.2, “Memory Map and Register Definitions.”](#)

#### 19.1.3.1 Data Transfer Schemes

##### 19.1.3.1.1 Data Input Modes

Three transfer modes are supported by ASRC: Mode 1 (Polling mode), Mode 2 (Interrupt mode), and Mode 3 (DMA mode).

##### Data Input Mode 1 (Polling Mode)

After power on, reset (individually) or DSP clears ASRIER\_ADIE<sub>x</sub> (where x = A, B or C), and ASRC operates in Mode 1. In Polling mode, data input interrupts are disabled (ASRIER<sub>n</sub>\_ADIE<sub>x</sub>=0, where x = A, B or C). The DSP can poll ASRSTR\_AIDEx to monitor input service requests.

The ASRC consumes data from each enabled input FIFO sample by sample, continually after each rising edge of the input sampling clock. When the number of data words in every input FIFO is less than the threshold, ASRC will set ASRSTR\_AIDEx (x:A, B or C). After DSP checks the ASRSTR and finds these input requests, the DSP will write enough data into the ASRDIA, ASRDIB or ASRDIC input register accordingly, before the ASRC fetches the next data. When the ASRC fetches the next data, and if the FIFOs are empty, an error will happen and ASRSTR\_AOLE will be set.

The threshold of the input FIFOs is 32 samples (by default); the FIFO size of each channel is 64 samples.

All data should be written in a predefined sequence. If ASRDIA needs to be written, the sequence should be: asrdi\_0, asrdi\_1, asrdi\_2, asrdi\_3, asrdi\_0,asrdi\_1, asrdi\_2,..., asrdi\_0, asrdi\_1, asrdi\_2, asrdi\_3. Here asrdi\_n stands for the data intended for the nth channel. The hardware will re-allocate each data to its corresponding channel FIFO. The channel being re-allocated is shown by ASRCCR\_ACIA.

After the number of data in every input FIFO is greater than the threshold, the data-needed status bit will be cleared.

##### Data Input Mode 2 (Interrupt Mode)

After DSP sets ASRIER\_ADIE<sub>x</sub> (x=A, B or C), the input interrupt is enabled.

The ASRC consumes data from the input FIFO continually when it is working. When the number of data words in the input FIFO is less than the threshold:

- The ASRC generates an interrupt request.
- A corresponding interrupt vector is transferred to the DSP.
- The status register’s corresponding bit ASRSTR\_AIDEx (x:A, B or C) will be set.

To serve this request, the DSP should write enough data into the input FIFOs. When the ASRC fetches the next data, and if the buffer is empty, an error will happen and ASRSTR\_AOLE will be set. If the overload interrupt is enabled, an overload interrupt will happen.

All data should be written in a predefined sequence too.

### Data Input Mode 3 (DMA Mode)

ASRSTR\_AIDEx (x = A, B or C) bits can also be used as DMA request source.

In DMA mode, when ASRSTR\_AIDEx (x = A, B or C) is active, they will cause a DMA transfer. The DMA-transferred data will feed into the ASRDIA, ASRDIB or ASRDIC input FIFOs. The other requirements and behaviors of the DMA mode are the same as those in the polling mode.

#### 19.1.3.1.2 Data Output Modes

Three transfer modes are supported by the Interface block.

#### Data Output Mode 1 (Polling Mode)

Output mode 1 is also a polling mode. It is almost the same as input mode 1, except that the direction of data being transferred and the involved register bits are different. The threshold of the output FIFOs is 32 samples; the FIFO size of each channel is 64 samples.

#### Data Output Mode 2 (Interrupt Mode)

Output mode 2 is also an interrupt mode. It is almost the same as input mode 2, except that again the direction of data being transferred and the involved register bits are different.

#### Data Output Mode 3 (DMA Mode)

Output mode 3 is also a DMA mode. It is almost the same as input mode 3, except that the direction of data being transferred and the involved register bits are different.

## 19.2 Memory Map and Register Definitions

### 19.2.1 Memory Map

Table 19-2 shows the ASRC memory map.

Table 19-2. Block Memory Map

Base Address Offset (Register Abbreviation)	Register	Access	Reset Value	Section/Page
0x0 (ASRCTR)	ASRC Control Register	R/W	0x000000	<a href="#">19.2.2.1/19-7</a>
0x1	ASRIER Interrupt Enable Register	R/W	0x000000	<a href="#">19.2.2.2/19-9</a>

**Table 19-2. Block Memory Map (Continued)**

Base Address Offset (Register Abbreviation)	Register		Access	Reset Value	Section/Page
0x2	ASRIEM	Interrupt Enable Mask Register for both DSP cores	R/W	0x000000	<a href="#">19.2.2.2/19-9</a>
0x3	ASRCNCR	Channel Number Configuration Register	R/W	0x000000	<a href="#">19.2.2.3/19-11</a>
0x4	ASRCFG	Filter Configuration Status Register	R/W	0x000000	<a href="#">19.2.2.4/19-12</a>
0x5 ASRCSR	ASRC Clock Source Register		R/W	0x000000	<a href="#">19.2.2.5/19-14</a>
0x6	ASRCDR1	ASRC Clock Divider Register 1	R/W	0x000000	<a href="#">19.2.2.6/19-16</a>
0x7	ASRCDR2	ASRC Clock Divider Register 2	R/W	0x000000	<a href="#">19.2.2.6/19-16</a>
0x8	ASRSTR	ASRC Status Register	R	0x000000	<a href="#">19.2.2.7/19-18</a>
0x9–0xB	Reserved		—	—	—
0xC	Reserved		—	—	—
0xD	Reserved		—	—	—
0xE	ASRDCR	ASRC Debug Control Register	R/W	0x02_0000	<a href="#">19.2.2.8/19-21</a>
0xF	Reserved		—	—	—
0x10	ASRPM1	Parameter Register 1	R/W	0x000000	<a href="#">19.2.2.9/19-23</a>
0x11	ASRPM2	Parameter Register 2	R/W	0x000000	<a href="#">19.2.2.9/19-23</a>
0x12	ASRPM3	Parameter Register 3	R/W	0x000000	<a href="#">19.2.2.9/19-23</a>
0x13	ASRPM4	Parameter Register 4	R/W	0x000000	<a href="#">19.2.2.9/19-23</a>
0x14	ASRPM5	Parameter Register 5	R/W	0x000000	<a href="#">19.2.2.9/19-23</a>
0x15	ASRTRF1	ASRC Task queue FIFO Register 1	R/W <sup>1</sup>	0x000000	<a href="#">19.2.2.10/19-23</a>
0x16	Reserved		—	—	—
0x17	ASRCCR	Channel Counter Register	R/W	0x000000	<a href="#">19.2.2.11/19-24</a>
0x18	ASRDIA	ASRC Data Input Register for Pair A	W	NA <sup>1</sup>	<a href="#">19.2.2.12/19-24</a>
0x19	ASRDOA	ASRC Data Output Register for Pair A	R	NA <sup>1</sup>	<a href="#">19.2.2.12/19-24</a>
0x1A	ASRDIB	ASRC Data Input Register for Pair B	W	NA <sup>1</sup>	<a href="#">19.2.2.12/19-24</a>
0x1B	ASRDOB	ASRC Data Output Register for Pair B	R	NA <sup>1</sup>	<a href="#">19.2.2.12/19-24</a>
0x1C	ASRDIC	ASRC Data Input Register for Pair C	W	NA <sup>1</sup>	<a href="#">19.2.2.12/19-24</a>
0x1D	ASRDOC	ASRC Data Output Register for Pair C	R	NA <sup>1</sup>	<a href="#">19.2.2.12/19-24</a>
0x20	ASRIDRHA	ASRC Ideal Ratio for Pair A (High Part)	R/W	0x000000	<a href="#">19.2.2.12.1/19-24</a>
0x21	ASRIDRLA	ASRC Ideal Ratio for Pair A (Low Part)	R/W	0x000000	<a href="#">19.2.2.12.1/19-24</a>
0x22	ASRIDRHB	ASRC Ideal Ratio for Pair B (High Part)	R/W	0x000000	<a href="#">19.2.2.12.2/19-25</a>
0x23	ASRIDRLB	ASRC Ideal Ratio for Pair B (Low Part)	R/W	0x000000	<a href="#">19.2.2.12.2/19-25</a>



**Table 19-2. Block Memory Map (Continued)**

Base Address Offset (Register Abbreviation)	Register	Access	Reset Value	Section/Page
0x24	ASRIDRHC ASRC Ideal Ratio for Pair C (High Part)	R/W	0x000000	19.2.2.12.3/19-25
0x25	ASRIDRLC ASRC Ideal Ratio for Pair C (Low Part)	R/W	0x000000	19.2.2.12.3/19-25
0x26	ASR76K ASRC 76kHz Period in terms of master clock	R/W	0x000A47	19.2.2.12.4/19-26
0x27	ASR56K ASRC 56kHz Period in terms of master clock	R/W	0x000DF3	19.2.2.12.5/19-26

**Notes:**

<sup>1</sup> Note that R/W registers may contain some read-only or write-only bits.

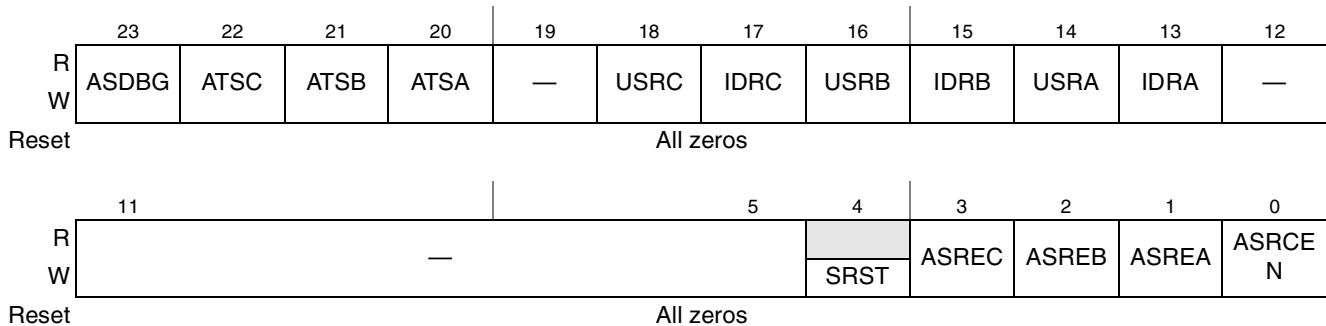
## 19.2.2 Register Descriptions

### 19.2.2.1 ASRC Control Register (ASRCTR)

The control register (ASRCTR) is a 24-bit read/write register that controls the ASRC operations, shown in [Figure 19-3](#).

Offset 0x0

Access: Mixed



**Figure 19-3. ASRC Control Register (ASRCTR)**

[Table 19-3](#) defines the ASRCTR.

**Table 19-3. ASRC Control Register Bits (ASRCTR)**

Bit	Field	Description
23	ASDBG	ASRC Debug Control Enable ASRC to enter debug mode.
22	ATSC	ASRC Pair C Automatic Selection For Processing Options When the ATSC bit is 1, pair C will automatic update its pre-processing and post-processing options (ASRCFG: PREMODC, ASRCFG:POSTMODC; see <a href="#">Section 19.2.2.4, “Filter Configuration Status Register (ASRCFG)”</a> ) based on the frequencies it detected. To use this option, the two parameter registers (TS76KHZ and TS56KHZ) should be set correctly (see <a href="#">Section 19.2.2.12.4, “ASRC 76 kHz Period Register in Terms of Master Clock (ASR76K)”</a> ). This bit should be disabled when {USRC, IDRC}={1,1}.

**Table 19-3. ASRC Control Register Bits (ASRCTR) (Continued)**

Bit	Field	Description
21	ATSB	ASRC Pair B Automatic Selection For Processing Options When the ATSB bit is 1, pair B will automatic update its pre-processing and post-processing options (ASRCFG:PREMODB, ASRCFG:POSTMODB; see <a href="#">Section 19.2.2.4, “Filter Configuration Status Register (ASRCFG)”</a> ) based on the frequencies it detected. To use this option, the two parameter registers (ASR76K and ASR56K) should be set correctly (see <a href="#">Section 19.2.2.12.4, “ASRC 76 kHz Period Register in Terms of Master Clock (ASR76K)”</a> ). When the ATSB bit is 0, the user is responsible for choosing the proper processing options for pair B. This bit should be disabled when {USRB, IDRB}={1,1}.
20	ATSA	ASRC Pair A Automatic Selection For Processing Options When the ATSA bit is 1, pair A will automatic update its pre-processing and post-processing options (ASRCFG:PREMODA, ASRCFG:POSTMODA; see <a href="#">Section 19.2.2.4, “Filter Configuration Status Register (ASRCFG)”</a> ) based on the frequencies it detected. To use this option, the two parameter registers (ASR76K and ASR56K) should be set correctly (see <a href="#">Section 19.2.2.12.4, “ASRC 76 kHz Period Register in Terms of Master Clock (ASR76K)”</a> ). When the ATSA bit is 0, the user is responsible for choosing the proper processing options for pair A. This bit should be disabled when {USRA, IDRA}={1,1}.
19	—	Reserved
18	USRC	Use Ratio for Pair C Use ratio as the input to ASRC. This bit is used in conjunction with IDRC control bit.
17	IDRC	Use Ideal Ratio for Pair C When USRC=0, this bit has no usage. When USRC=1 and IDRC=0, ASRC internal measured ratio will be used. When USRC=1 and IDRC=1, idea ratio from the interface register ASRIDRHC, ASRIDRLC will be used. It issuggested to manually set ASRCFG:POSTMODC, ASRCFG:PREMODC
16	USRB	Use Ratio for Pair B Use ratio as the input to ASRC. This bit is used in conjunction with IDRB control bit.
15	IDRB	Use Ideal Ratio for Pair B When USRB=0, this bit has no usage. When USRB=1 and IDRB=0, ASRC internal measured ratio will be used. When USRB=1 and IDRB=1, idea ratio from the interface register ASRIDRHB, ASRIDRLB will be used.It is suggested to manually set ASRCFG:POSTMODB, ASRCFG:PREMODB.
14	USRA	Use Ratio for Pair A Use ratio as the input to ASRC. This bit is used in conjunction with IDRA control bit.
13	IDRA	Use Ideal Ratio for Pair A When USRA=0, this bit has no usage. When USRA=1 and IDRA=0, ASRC internal measured ratio will be used. When USRA=1 and IDRA=1, idea ratio from the interface register ASRIDRHA, ASRIDRLA will be used. It is suggested to manuallymanually set ASRCFG:POSTMODA, ASRCFG:PREMODA.
12–5	—	Reserved
4	SRST	Software Reset This bit is self-clear bit. Once it is been written as 1, it will generate a software reset signal inside ASRC. After 9 cycles of the master clock, this reset process will stop, and this bit will be cleared automatically.
3	ASREC	ASRC Enable C Enables the conversion of pair C of the ASRC. When ASREC is cleared, conversion of pair C is disabled.

**Table 19-3. ASRC Control Register Bits (ASRCTR) (Continued)**

Bit	Field	Description
2	ASREB	ASRC Enable B Enables the conversion of pair B of the ASRC. When ASREB is cleared, conversion of pair B is disabled.
1	ASREA	ASRC Enable A Enables the conversion of pair A of the ASRC. When ASREA is cleared, conversion of pair A is disabled.
0	ASRCEN	ASRC Enable Enables the operation of the ASRC.

### 19.2.2.2 Interrupt Enable Register and Mask Register (ASRIER, ASRIEM)

The Interrupt Enable and Mask registers are read/write registers, and they support the two interrupt enable lines connected to the two different DSP cores. Use the ASRIER register (Figure 19-4) to enable interrupts, and use the ASRIEM register (Figure 19-5) to determine which DSP core the interrupt will request to.



**Figure 19-4. Interrupt Enable Register (ASRIER)**

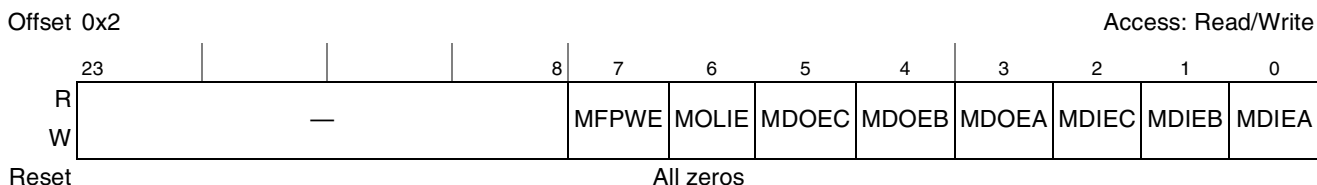
Table 19-4 defines the ASRIER.

**Table 19-4. Interrupt Enable Register (ASRIER)**

Bit	Field	Description
23–8	—	Reserved. Should be written as zero for compatibility.
7	AFPWE	FP in Wait State Interrupt Enable Enables the FP for the wait state interrupt.
6	AOLIE	Overload Interrupt Enable Enables the overload interrupt.
5	ADOEC	Data Output C Interrupt Enable Enables the data output C interrupt.
4	ADOEB	Data Output B Interrupt Enable Enables the data output B interrupt.
3	ADOEA	Data Output A Interrupt Enable Enables the data output A interrupt.
2	ADIEC	Data Input C Interrupt Enable Enables the data input C interrupt.

**Table 19-4. Interrupt Enable Register (ASRIER) (Continued)**

Bit	Field	Description
1	ADIEB	Data Input B Interrupt Enable Enables the data input B interrupt.
0	ADIEA	Data Input A Interrupt Enable Enables the data input A Interrupt.



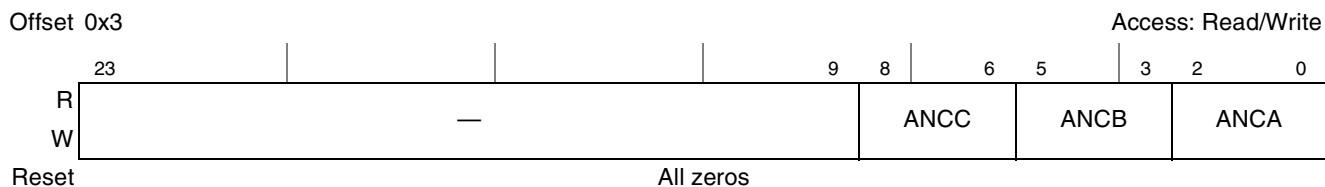
**Figure 19-5. Interrupt Enable Mask Register (ASRIEM)**

**Table 19-5. Interrupt Enable Mask Register (ASRIEM)**

Bit	Field	Description
23–8	—	Reserved. Should be written as zero for compatibility.
7	MFPWE	Mask of FP in Wait State Interrupt 0 Enables the wait state interrupt to Core 1. 1 Enables the wait state interrupt to Core 2.
6	MOLIE	Mask of Overload Interrupt Enable 0 Enables the overload interrupt to Core 1. 1 Enables the overload interrupt to Core 2.
5	MDOEC	Mask of Data Output C Interrupt Enable 0 Enables the data output C interrupt to Core 1. 1 Enables the data output C interrupt to Core 2.
4	MDOEB	Mask of Data Output B Interrupt Enable 0 Enables the data output B interrupt to Core 1. 1 Enables the data output B interrupt to Core 2.
3	MDOEA	Mask of Data Output A Interrupt Enable 0 Enables the data output A interrupt to Core 1. 1 Enables the data output A interrupt to Core 2.
2	MDIEC	Mask of Data Input C Interrupt Enable 0 Enables the data input C interrupt to Core 1. 1 Enables the data input C interrupt to Core 2.
1	MDIEB	Mask of Data Input B Interrupt Enable 0 Enables the data input B interrupt to Core 1. 1 Enables the data input B interrupt to Core 2.
0	MDIEA	Mask of Data Input A Interrupt Enable 0 Enables the data input A Interrupt to Core 1. 1 Enables the data input A Interrupt to Core 2.

### 19.2.2.3 Channel Number Configuration Register (ASRCNCR)

The Channel Number Configuration register (ASRCNCR) is a 24-bit read/write register that sets the number of channels used by each ASRC conversion pair, shown in [Figure 19-6](#).



**Figure 19-6. Channel Number Configuration Register (ASRCNCR)**

[Table 19-6](#) defines the ASRCNCR.

**Table 19-6. Channel Number Configuration Register (ASRCNCR)**

Bit	Field	Description
23–9	—	Reserved. Should be written as zero for compatibility.
8–6	ANCC <sup>1</sup>	Number of C Channels 000 0 channels in C (Pair C is disabled) 001 2 channel in C 010 4 channels in C 011 6 channels in C 100 8 channels in C 101 10 channels in C
5–3	ANCB	Number of B Channels 000 0 channels in B (Pair B is disabled) 001 2 channel in B 010 4 channels in B 011 6 channels in B 100 8 channels in B 101 10 channels in B
2–0	ANCA	Number of A Channels 000 0 channels in A (Pair A is disabled) 001 2 channel in A 010 4 channels in A 011 6 channels in A 100 8 channels in A 101 10 channels in A

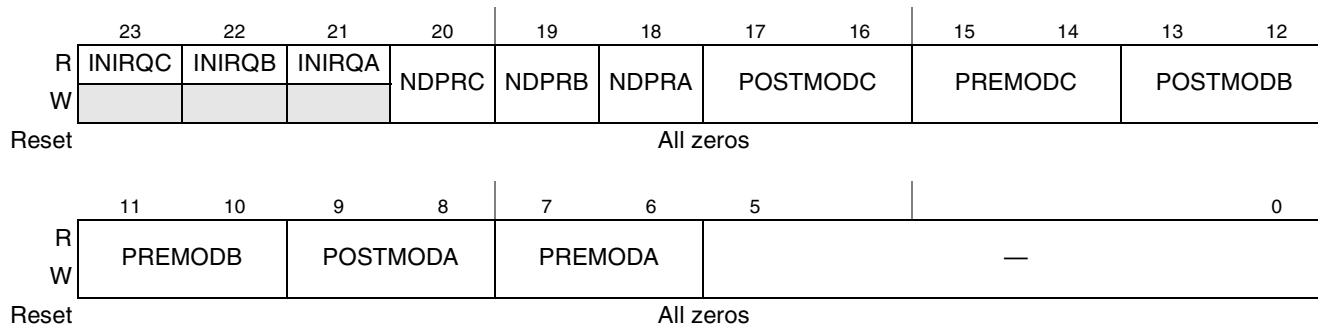
<sup>1</sup> ANCC+ANCB+ANCA<=10.

### 19.2.2.4 Filter Configuration Status Register (ASRCFG)

The Filter Configuration Status register (ASRCFG) is a 24-bit read/write register that sets and/or automatically senses the ASRC operations, shown in [Figure 19-7](#).

Offset 0x4

Access: Mixed



**Figure 19-7. Filter Configuration Status Register (ASRCFG)**

[Table 19-7](#) defines the ASRCFG.

**Table 19-7. Filter Configuration Status Register (ASRCFG)**

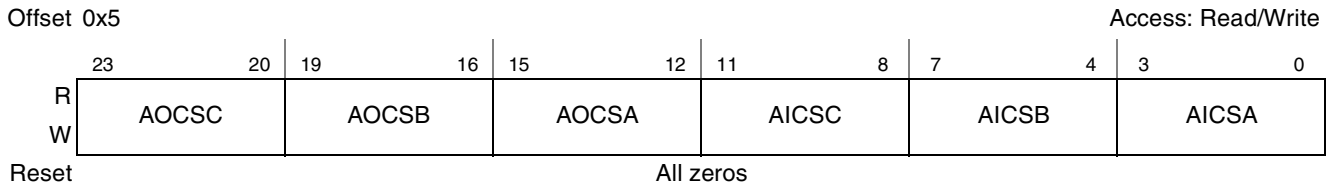
Bit	Field	Description
23	INIRQC	Initialization for Conversion Pair C is served When the INIRQC bit is 1, the initialization for conversion pair C is served. The INIRQC bit is cleared by disabling the ASRC conversion pair (ASRCTR:ASREC = 0 or ASRCTR:ASRCEN = 0).
22	INIRQB	Initialization for Conversion Pair B is served When the INIRQB bit is 1, the initialization for conversion pair B is served. The INIRQB bit is cleared by disabling the ASRC conversion pair (ASRCTR:ASREB = 0 or ASRCTR:ASRCEN = 0).
21	INIRQA	Initialization for Conversion Pair A is served When the INIRQA bit is 1, the initialization for conversion pair A is served. The INIRQA bit is cleared by disabling the ASRC conversion pair (ASRCTR:ASREA = 0 or ASRCTR:ASRCEN = 0).
20	NDPRC	Not Use Default Parameters for RAM-stored Parameters For Conversion Pair C 0 Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 Do not use default parameters for RAM-stored parameters. Use the parameters already stored in RAM.
19	NDPRB	Do Not Use Default Parameters for RAM-stored Parameters For Conversion Pair B 0 Use the default parameters for RAM-stored parameters, and override any parameters already in RAM. 1 Do not use the default parameters for RAM-stored parameters; use the parameters already stored in RAM.
18	NDPRA	Do Not Use Default Parameters for RAM-stored Parameters For Conversion Pair A 0 Use the default parameters for RAM-stored parameters; override any parameters already in RAM. 1 Do not use the default parameters for RAM-stored parameters; use the parameters already stored in RAM.

**Table 19-7. Filter Configuration Status Register (ASRCFG) (Continued)**

Bit	Field	Description
17–16	POSTMODC [1-0]	Post-Processing Configuration for Conversion Pair C Use these bits to set the selection of the post-processing configuration for Pair C. 00 Select Upsampling-by-2, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 01 Select Direct-Connection, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 10 Select Downsampling-by-2, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> These bits can be read/written by the user if ASRCTR:ATSC = 0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSC = 1 (see <a href="#">Table 19-3</a> ).
15–14	PREMODC [1-0]	Pre-Processing Configuration for Conversion Pair C Use these bits to set the selection of the post-processing configuration for Pair C. 00 Select Upsampling-by-2, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 01 Select Direct-Connection, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 10 Select Downsampling-by-2, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 11 Select passthrough mode. In this case, POSTMODC[1-0] will not apply. These bits can be read/written by the user if ASRCTR:ATSC = 0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSC = 1 (see <a href="#">Table 19-3</a> ).
13–12	POSTMODB [1-0]	Post-Processing Configuration for Conversion Pair B Use these bits to set the selection of the post-processing configuration for Pair B. 00 Select Upsampling-by-2, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 01 Select Direct-Connection, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 10 Select Downsampling-by-2, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> These bits can be read/written by the user if ASRCTR:ATSB = 0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSB = 1 (see <a href="#">Table 19-3</a> ).
11–10	PREMODB [1-0]	Pre-Processing Configuration for Conversion Pair B Use these bits to set the selection of the post-processing configuration for Pair B. 00 Select Upsampling-by-2, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 01 Select Direct-Connection, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 10 Select Downsampling-by-2, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 11 Select passthrough mode. In this case, POSTMODB[1-0] will not apply. These bits can be read/written by the user if ASRCTR:ATSB = 0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSB = 1 (see <a href="#">Table 19-3</a> ).
9–8	POSTMODA [1-0]	Post-Processing Configuration for Conversion Pair A Use these bits to set the selection of the post-processing configuration for Pair A. 00 Select Upsampling-by-2, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 01 Select Direct-Connection, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 10 Select Downsampling-by-2, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> These bits can be read/written by the user if ASRCTR:ATSA = 0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSA = 1 (see <a href="#">Table 19-3</a> ).
7–6	PREMODA [1-0]	Pre-Processing Configuration for Conversion Pair A Use these bits to set the selection of the post-processing configuration for Pair A. 00 Select Upsampling-by-2, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 01 Select Direct-Connection, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 10 Select Downsampling-by-2, as defined in <a href="#">Section 19.5.1.1, “Signal Processing Flow.”</a> 11 Select passthrough mode. In this case, POSTMODA[1-0] will not apply. These bits can be read/written by the user if ASRCTR:ATSA = 0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSA = 1 (see <a href="#">Table 19-3</a> ). These bits set the selection of the pre-processing configuration.
5–0	—	Reserved. Should be written as zero for compatibility.

### 19.2.2.5 ASRC Clock Source Register (ASRCSR)

The clock source register (ASRCSR) is a 24-bit read/write register that controls the sources of the input and output clocks of the ASRC, shown in [Figure 19-8](#).



**Figure 19-8. Clock Source Register (ASRCSR)**

[Table 19-8](#) defines the ASRCSR.

**Table 19-8. Clock Source Register (ASRCSR)**

Bit	Field	Description
23–20	AOCSC	Output Clock Source C 0000 ESAI Tx clock 0001 ESAI-1 Tx clock 0010 ESAI-2 Tx clock 0011 ESAI-3 Tx clock 0100 S/PDIF Tx clock 0101 Reserved 0110 Reserved 0111 Reserved 1000 ESAI Rx clock 1001 ESAI-1 Rx clock 1010 ESAI-2 Rx clock 1011 ESAI-3 Rx clock 1100 SPDIF Rx clock Any other value—ASRCK1 (In DSP56724/DSP56725, this signal is derived from the PLL, and can be controlled by the ASCDR register in the CGM module.)
19–16	AOCSB	Output Clock Source B 0000 ESAI Tx clock 0001 ESAI-1 Tx clock 0010 ESAI-2 Tx clock 0011 ESAI-3 Tx clock 0100 S/PDIF Tx clock 0101 Reserved 0110 Reserved 0111 Reserved 1000 ESAI Rx clock 1001 ESAI-1 Rx clock 1010 ESAI-2 Rx clock 1011 ESAI-3 Rx clock 1100 SPDIF Rx clock Any other value—ASRCK1 (In DSP56724/DSP56725, this signal is derived from the PLL, and can be controlled by the ASCDR register in the CGM module.)



**Table 19-8. Clock Source Register (ASRCSR) (Continued)**

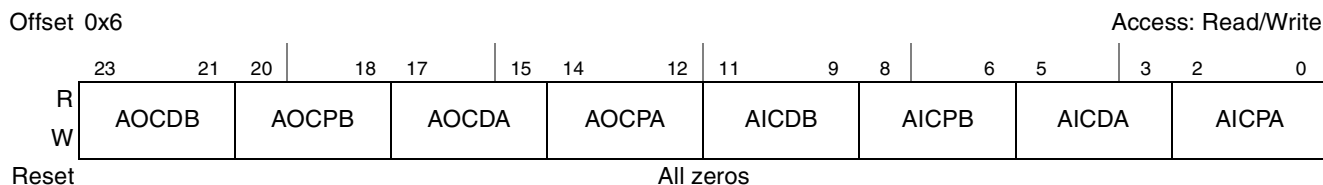
Bit	Field	Description
15–12	AOCSA	Output Clock Source A 0000 ESAI Tx clock 0001 ESAI-1 Tx clock 0010 ESAI-2 Tx clock 0011 ESAI-3 Tx clock 0100 S/PDIF Tx clock 0101 Reserved 0110 Reserved 0111 Reserved 1000 ESAI Rx clock 1001 ESAI-1 Rx clock 1010 ESAI-2 Rx clock 1011 ESAI-3 Rx clock 1100 SPDIF Rx clock Any other value—ASRCK1 (In DSP56724/DSP56725, this signal is derived from the PLL, and can be controlled by the ASCDR register in the CGM module.)
11–8	AICSC	Input Clock Source C 0000 ESAI Rx clock 0001 ESAI-1 Rx clock 0010 ESAI-2 Rx clock 0011 ESAI-3 Rx clock 0100 S/PDIF Rx clock 0101 Reserved 0110 Reserved 1000 ESAI Tx clock 1001 ESAI-1 Tx clock 1010 ESAI-2 Tx clock 1011 ESAI-3 Tx clock 1100 SPDIF Tx clock Any other value—ASRCK1 (In DSP56724/DSP56725, this signal is derived from the PLL, and can be controlled by the ASCDR register in the CGM module.)

**Table 19-8. Clock Source Register (ASRCSR) (Continued)**

Bit	Field	Description
7–4	AICSB	Input Clock Source B 0000 ESAI Rx clock 0001 ESAI-1 Rx clock 0010 ESAI-2 Rx clock 0011 ESAI-3 Rx clock 0100 S/PDIF Rx clock 0101 Reserved 0110 Reserved 1000 ESAI Tx clock 1001 ESAI-1 Tx clock 1010 ESAI-2 Tx clock 1011 ESAI-3 Tx clock 1100 SPDIF Tx clock Any other value—ASRCK1 (In DSP56724/DSP56725, this signal is derived from the PLL, and can be controlled by the ASCDR register in the CGM module.)
3–0	AICSA	Input Clock Source A 0000 ESAI Rx clock 0001 ESAI-1 Rx clock 0010 ESAI-2 Rx clock 0011 ESAI-3 Rx clock 0100 S/PDIF Rx clock 0101 Reserved 0110 Reserved 1000 ESAI Tx clock 1001 ESAI-1 Tx clock 1010 ESAI-2 Tx clock 1011 ESAI-3 Tx clock 1100 SPDIF Tx clock Any other value—ASRCK1 (In DSP56724/DSP56725, this signal is derived from the PLL, and can be controlled by the ASCDR register in the CGM module.)

### 19.2.2.6 ASRC Clock Divider Registers (ASRCDR1, ASRCDR2)

The Clock Divider registers (ASRCDR1, ASRCDR2) are two 24-bit read/write registers that control the division factors of the ASRC input and output clock sources, shown in [Figure 19-9](#) and [Figure 19-10](#).



**Figure 19-9. Clock Divider Register-1 (ASRCDR1)**

Table 19-9 defines the ASRCDR1.

**Table 19-9. Clock Divider Register-1 (ASRCDR1)**

Bit	Field	Description
23–21	AOCDB	Output Clock Divider B Specify the divide ratio of the output clock divider B. The divide ratio can be from 1 to 8 (AOCDB[2:0] = 000 to 111).
20–18	AOCPB	Output Clock Prescaler B Specify the prescaling factor of the output prescaler B. The prescaling ratio may be any power of 2 from 1 to 128.
17–15	AOCDA	Output Clock Divider A Specify the divide ratio of the output clock divider A. The divide ratio can be from 1 to 8 (AOCDA[2:0] = 000 to 111).
14–12	AOCPA	Output Clock Prescaler A Specify the prescaling factor of the output prescaler A. The prescaling ratio may be any power of 2 from 1 to 128.
11–9	AICDB	Input Clock Divider B Specify the divide ratio of the input clock divider B. The divide ratio can be from 1 to 8 (AICDB[2:0] = 000 to 111).
8–6	AICPB	Input Clock Prescaler B Specify the prescaling factor of the input prescaler B. The prescaling ratio may be any power of 2 from 1 to 128.
5–3	AICDA	Input Clock Divider A Specify the divide ratio of the input clock divider A. The divide ratio can be from 1 to 8 (AICDA[2:0] = 000 to 111).
2–0	AICPA	Input Clock Prescaler A Specify the prescaling factor of the input prescaler A. The prescaling ratio may be any power of 2 from 1 to 128.



**Figure 19-10. Clock Divider Register-2 (ASRCDR2)**

Table 19-10 defines the ASRCDR2.

**Table 19-10. Clock Divider Register-2 (ASRCDR2)**

Bit	Field	Description
23–12	—	Reserved. Should be written as zero for compatibility.
11–9	AOCDC	Output Clock Divider C Specify the divide ratio of the output clock divider C. The divide ratio is from 1 to 8 (AOCDC[2:0] = 000 to 111).

**Table 19-10. Clock Divider Register-2 (ASRCDR2) (Continued)**

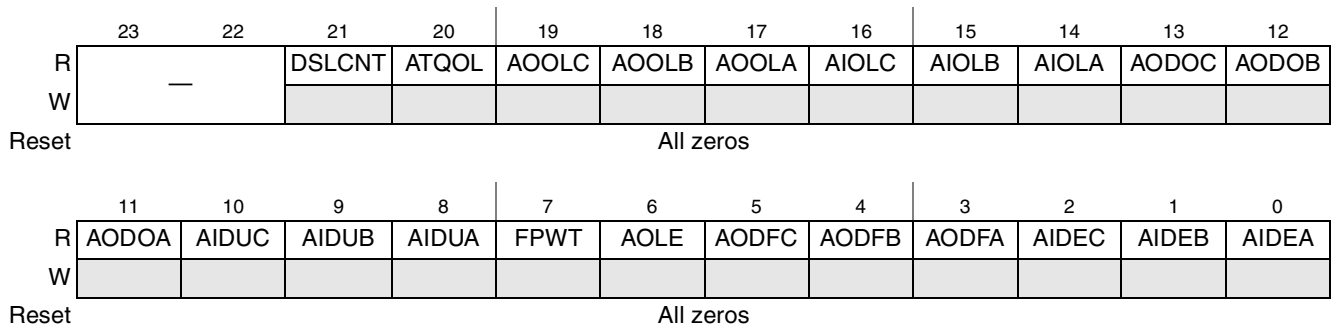
Bit	Field	Description
8–6	AOCPC	Output Clock Prescaler C Specify the prescaling factor of the output prescaler C. The prescaling ratio may be any power of 2 from 1 to 128.
5–3	AICDC	Input Clock Divider C Specify the divide ratio of the input clock divider C. The divide ratio is from 1 to 8 (AICDC[2:0] = 000 to 111).
2–0	AICPC	Input Clock Prescaler C Specify the prescaling factor of the input prescaler C. The prescaling ratio may be any power of 2 from 1 to 128.

### 19.2.2.7 ASRC Status Register (ASRSTR)

The status register (ASRSTR) is a 24-bit read-only register used by the DSP core to examine the status of the ASRC module and clear the overload interrupt request and AOLE flag bit, shown in [Figure 19-11](#). Reading the status register will return the current state of ASRC.

Offset 0x8

Access: Read only



**Figure 19-11. Status Register (ASRSTR)**

[Table 19-11](#) defines the ASRSTR.

**Table 19-11. Status Register (ASRSTR)**

Bit	Field	Description
23–22	—	Reserved. Should be written as zero for compatibility.
21	DSL CNT	DSL Counter Input to FIFO ready 1 Indicates when the new DSL counter information is stored in the internal ASRC FIFO. 0 Indicates that new DSL counter information is in the process of being stored into the internal ASRC FIFO. When ASRIER:AFPWE = 1, the rising edge of this signal will propose an interrupt request. When the DSLCNT bit is set, writing any value will clear the interrupt request proposed by the rising edge of this bit.
20	ATQOL	Taskque FIFO overload 1 Indicates when the FIFO logic is overloaded. This may help to check the reason why overload interrupt happens. The ATQOL bit is cleared when writing 1 to ASRCTR:AOLIE.

**Table 19-11. Status Register (ASRSTR) (Continued)**

Bit	Field	Description
19	AOOLC	Pair C Output Task Overload 1 Indicates when the pair C output task is overloaded. This may help to check the reason why overload interrupt happens. The AOOLC bit is cleared when writing 1 to ASRCTR:AOLIE.
18	AOOLB	Pair B Output Task Overload 1 Indicates that the pair B output task is overloaded. This may help to check the reason why overload interrupt happens. The AOOLB bit is cleared when writing 1 to ASRCTR:AOLIE.
17	AOOLA	Pair A Output Task Overload 1 Indicates that the pair A output task is overloaded. This may help to check the reason why overload interrupt happens. The AOOLA bit is cleared when writing 1 to ASRCTR:AOLIE.
16	AIOLC	Pair C Input Task Overload 1 Indicates that the pair C input task is overloaded. This may help to check the reason why overload interrupt happens. The AIOLC bit is cleared when writing 1 to ASRCTR:AOLIE.
15	AIOLB	Pair B Input Task Overload 1 Indicates that the pair B input task is overloaded. This may help to check the reason why overload interrupt happens. The AIOLB bit is cleared when writing 1 to ASRCTR:AOLIE.
14	AIOLA	Pair A Input Task Overload 1 Indicates that the pair A input task is overloaded. This may help to check the reason why overload interrupt happens. The AIOLA bit is cleared when writing 1 to ASRCTR:AOLIE.
13	AODOC	Output Data Buffer C is overflowed 1 Indicates that output data buffer C is overflowed. 0 Indicates that output data buffer C is not overflowed.
12	AODOB	Output Data Buffer B is overflowed 1 Indicates that output data buffer B is overflowed. 0 Indicates that output data buffer B is not overflowed.
11	AODOA	Output Data Buffer A is overflowed 1 Indicates that output data buffer A is overflowed. 0 Indicates that output data buffer A is not overflowed.
10	AIDUC	Input Data Buffer C is underflowed 1 Indicates that input data buffer C is underflowed. 0 Indicates that input data buffer C is not underflowed.
9	AIDUB	Input Data Buffer B is underflowed 1 Indicates that input data buffer B is underflowed. 0 Indicates that input data buffer B is not underflowed.
8	AIDUA	Input Data Buffer A is underflowed 1 Indicates that input data buffer A is underflowed. 0 Indicates that input data buffer A is not underflowed.

**Table 19-11. Status Register (ASRSTR) (Continued)**

Bit	Field	Description
7	FPWT	<p>FP is in wait state  <i>This bit is for debug only.</i>                      1 Indicates that filter processor is in wait states.                      0 Indicates that filter processor is not in wait states.                      If ASRCTR:AFPWE=1 and ASRCTR:ASDBG=1, an interrupt will be proposed when this bit is set.</p>
6	AOLE	<p>Overload Error Flag                      1 Indicates that the task rate is too high for the ASRC to handle.                      The reasons for overload may be:</p> <ul style="list-style-type: none"> <li>• too high input clock frequency</li> <li>• too high output clock frequency</li> <li>• incorrect selection of the pre-filter</li> <li>• low DSP system clock</li> <li>• too many channels</li> <li>• underrun</li> <li>• or any combination of the reasons above.</li> </ul> <p>Because the ASRC uses the same hardware resources to perform various tasks, the real reason for the overload is not straightforward, and it should be carefully analyzed by the programmer.                      If ASRCTR:AOLIE=1, an interrupt will be proposed when this bit is set.                      With this bit set = 1, writing any value into the status register will clear this bit and can clear the interrupt request proposed by this bit.</p>
5	AODFC	<p>Number of data in Output Data Buffer C is greater than threshold                      1 Indicates that the number of data words already existing in ASRDORC is greater than the threshold, and the DSP can read data from ASRDORC. When AODFC is set, the ASRC generates a data output C interrupt request to the DSP core, if enabled (that is, ASRCTR:ADOEC = 1).                      A DMA request is always generated when the AODFC bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.</p>
4	AODFB	<p>Number of data in Output Data Buffer B is greater than threshold                      1 Indicates that the number of data words already existing in ASRDORB is greater than the threshold, and the DSP can read data from ASRDORB. When AODFB is set, the ASRC generates a data output B interrupt request to the DSP core, if enabled (that is, ASRCTR:ADOEB = 1).                      A DMA request is always generated when the AODFB bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.</p>
3	AODFA	<p>Number of data in Output Data Buffer A is greater than threshold                      1 Indicates that the number of data words already existing in ASRDORA is greater than the threshold, and the DSP can read data from ASRDORA. When AODFA is set, the ASRC generates a data output A interrupt request to the DSP core, if enabled (that is, ASRCTR:ADOEA = 1).                      A DMA request is always generated when the AODFA bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.</p>
2	AIDEC	<p>Number of data in Input Data Buffer C is less than threshold                      1 Indicates that the number of data words still available in ASRDIRC is less than the threshold, and the DSP can write data to ASRDIRC. When AIDEC is set, the ASRC generates a data input C interrupt request to the DSP core, if enabled (that is, ASRCTR:ADIEC = 1).                      A DMA request is always generated when the AIDEC bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.</p>

**Table 19-11. Status Register (ASRSTR) (Continued)**

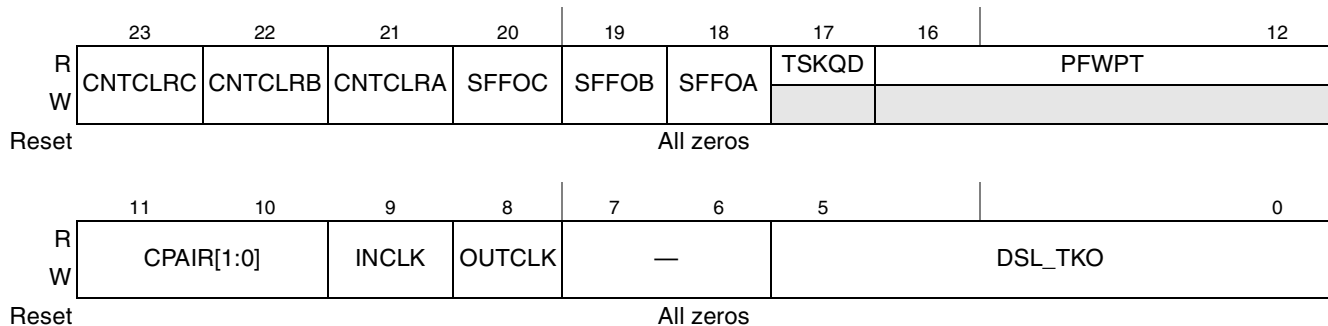
Bit	Field	Description
1	AIDEB	Number of data in Input Data Buffer B is less than threshold 1 Indicates that the number of data words still available in ASRDIRB is less than the threshold, and the DSP can write data to ASRDIRB. When AIDEB is set, the ASRC generates a data input B interrupt request to the DSP core, if enabled (that is, ASRCTR:ADIEB = 1). A DMA request is always generated when the AIDEB bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
0	AIDEA	Number of data in Input Data Buffer A is less than threshold 1 Indicates that the number of data words still available in ASRDIRA is less than the threshold, and the DSP can write data to ASRDIRA. When AIDEA is set, the ASRC generates a data input A interrupt request to the DSP core, if enabled (that is, ASRCTR:ADIEA = 1). A DMA request is always generated when the AIDEA bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.

### 19.2.2.8 ASRC Debug Control Register (ASRDCR)

The ASRC Debug Control register is used to clear FIFOs and set full output FIFOs, shown in [Figure 19-12](#). The bits CNTCLRA, CNTCLRB, CNTCLRC, SFFOC, SFFOB and SFFOA are used for testing and are not recommended to be used by you. There is no guarantee for future expansion of these bits.

Offset 0xE

Access: Mixed



**Figure 19-12. Debug Control Register (ASRDCR)**

[Table 19-12](#) defines the ASRDCR.

**Table 19-12. Debug Control Register (ASRDCR)**

Bit	Field	Description
23	CNTCLRC	Input/Output FIFO Pointers Clear for conversion pair C 0 Enable FIFO C counters 1 Clear FIFO C counters
22	CNTCLRB	Input/Output FIFO Pointers Clear for conversion pair B 0 Enable FIFO B counters 1 Clear FIFO B counters
21	CNTCLRA	Input/Output FIFO Pointers Clear for conversion pair A 0 Enable FIFO A counters 1 Clear FIFO A counters

**Table 19-12. Debug Control Register (ASRDCR) (Continued)**

Bit	Field	Description
20	SFFOC	Force Output FIFO Full or Not for conversion pair C 0 Do not set output FIFO C full 1 Force output FIFO C full
19	SFFOB	Force Output FIFO Full or Not for conversion pair B 0 Do not set output FIFO B full 1 Force output FIFO B full
18	SFFOA	Force Output FIFO Full or Not for conversion pair A 0 Do not set output FIFO A full 1 Force output FIFO A full
17	TSKQE	Task Queue is Empty 1 Indicates that the task queue is empty. 0 Indicates that the task queue is not empty. In debug mode, it is not recommended to generate new tasks when this bit is 1'b0.
16–12	PFWPT [4:0]	Write Pointer for Prefilter Output Buffer This is the write pointer for the prefilter output buffer of the conversion pair under debugging (selected by CPAIR[1:0]). It is read-only and for debug purpose only.
11–10	CPAIR[1:0]	Current Pair under Debugging These two bits select the conversion pair number under debugging. 00 Conversion pair A is under debugging. 01 Conversion pair B is under debugging. 10 Conversion pair C is under debugging.
9	INCLK	The Input Clock for the Debugging conversion pair Writing this bit as 1 will generate an ASRC input processing task for the debugging pair.
8	OUTCLK	The Output Clock for the Debugging conversion pair Writing this bit as 1 will generate an ASRC output processing task for the debugging pair.
7–6	—	Reserved. Should be written as zero for compatibility.
5–0	DSL_TKO[29:24]	The Simulated DSL Track Out for the Debugging pair These are the 6 MSBs of the simulated DSL track out for the debugging pair. The filter processor will use these values to calculate the ASRC output as it detects the rising edge of ASRDCR[7] (OUTCLK).

When ASRMAA:MEMOPT[1:0] = 11; it means the accessing ASRMAD will actually be accessing the small internal register bank. Currently this small register bank includes only two 14-bit registers (Register 0, Register 1):

- Register 0: TS76KHZ[13:0]: Read/Write. This value should be equal to Fsys (frequency of the system clock, in Hz)/76000. The reset value is 0x0A47, which assumes that Fsys = 200 MHz.
- Register 1: TS56KHZ[13:0]: Read/Write. This value should be equal to Fsys (frequency of the system clock, in Hz)/56000. The reset value is 0x0DF3, which assumes that Fsys = 200 MHz.

These two small internal register bank registers help the ASRC internal logic decide the pre-processing and the post-processing options automatically (see [Table 19-3](#) and [Table 19-7](#)). To access these two registers, first assign 0xC00000h to ASRMAA, then:

1. Read the ASRMAD register, which will give the value of TS76KHZ[13:0];  
then read the ASRMAD register again, which will give the value of TS56KHZ[13:0].



- Write the ASRMAD register, to assign a value to TS76KHZ[13:0];  
then write the ASRMAD register again, to assign a value to TS56KHZ[13:0].

### 19.2.2.9 Parameter Registers (ASRPM1–ASRPM5)

The Parameter registers determine the performance, and these registers are readable and writable. The Parameter registers should be set before enabling the ASRC. Recommended values for the Parameter registers are [Table 19-13](#).

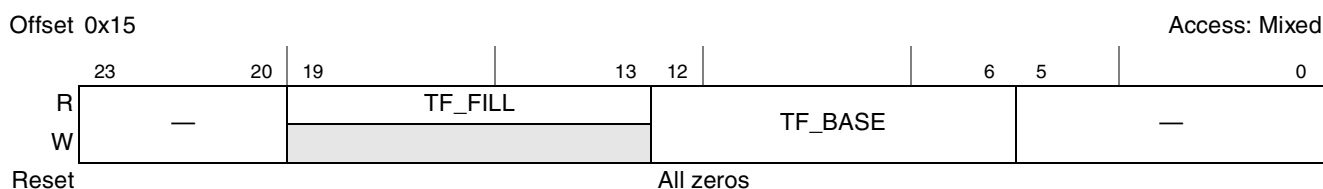
**Table 19-13. Recommended Values for Parameter Registers**

Register	Offset	Access	Reset Value	Recommended Value
asrcpm1	0x10	R/W	0x000000	0x7fffff
asrcpm2	0x11	R/W	0x000000	0x255555
asrcpm3	0x12	R/W	0x000000	0xff7280
asrcpm4	0x13	R/W	0x000000	0xff7280
asrcpm5	0x14	R/W	0x000000	0xff7280

### 19.2.2.10 ASRC Task Queue FIFO Register (ASRTFR1)

These registers define and show the parameters for the ASRC inner task queue FIFOs.

[Figure 19-13](#) shows the bit fields of the ASRC task queue FIFO register 1.



**Figure 19-13. ASRC Task Queue FIFO Register 1 (ASRTFR1)**

[Table 19-14](#) defines the ASRTFR1.

**Table 19-14. ASRC Task Queue FIFO Register 1 (ASRTFR1)**

Bit	Field	Description
23–20	—	Reserved. Should be written as zero for compatibility.
19–13	TF_FILL	Display the entries of the task queue FIFO.
12–6	TF_BASE	Set and display the base address for the task queue FIFO. Recommended Value is: 0x7C.
5–0	—	Reserved. Should be written as zero for compatibility.

### 19.2.2.11 Channel Counter Register (ASRCCR)

The channel counter register (ASRCCR) is a 24-bit read/write register that sets and reflects the current specific input/output FIFO being accessed through the Shared bus for each ASRC conversion pair, shown in Figure 19-14.

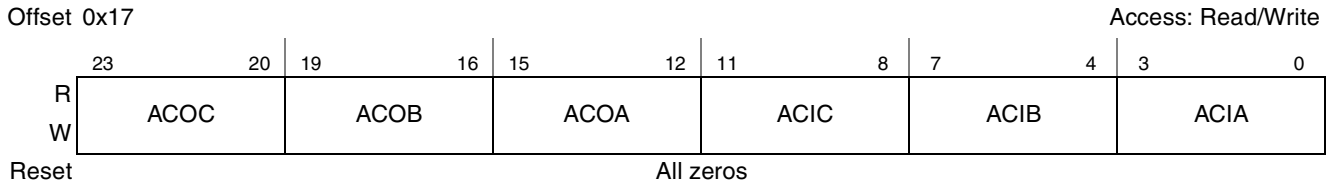


Figure 19-14. Channel Counter Register (ASRCCR)

Table 19-15 defines the ASRCCR.

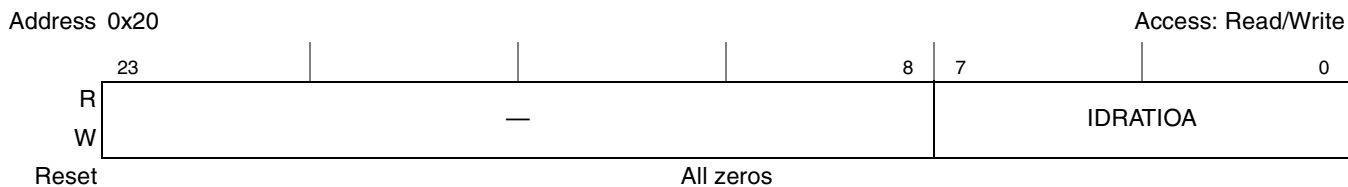
Table 19-15. Channel Counter Register (ASRCCR)

Bit	Field	Description
23–20	ACOC	The channel counter for Pair C’s output FIFO These bits indicate the current channel being accessed through the Shared bus for Pair C’s output FIFO’s usage. The value can be any value between [0, ANCC-1]
19–16	ACOB	The channel counter for Pair B’s output FIFO These bits indicate the current channel being accessed through the Shared bus for Pair B’s output FIFO’s usage. The value can be any value between [0, ANCB-1]
15–12	ACOA	The channel counter for Pair A’s output FIFO These bits indicate the current channel being accessed through the Shared bus for Pair A’s output FIFO’s usage. The value can be any value between [0, ANCA-1]
11–8	ACIC	The channel counter for Pair C’s input FIFO These bits indicate the current channel being accessed through the Shared bus for Pair C’s input FIFO’s usage. The value can be any value between [0, ANCC-1]
7–4	ACIB	The channel counter for Pair B’s input FIFO These bits indicate the current channel being accessed through the Shared bus for Pair B’s input FIFO’s usage. The value can be any value between [0, ANCB-1]
3–0	ACIA	The channel counter for Pair A’s input FIFO These bits indicate the current channel being accessed through the Shared bus for Pair A’s input FIFO’s usage. The value can be any value between [0, ANCA-1]

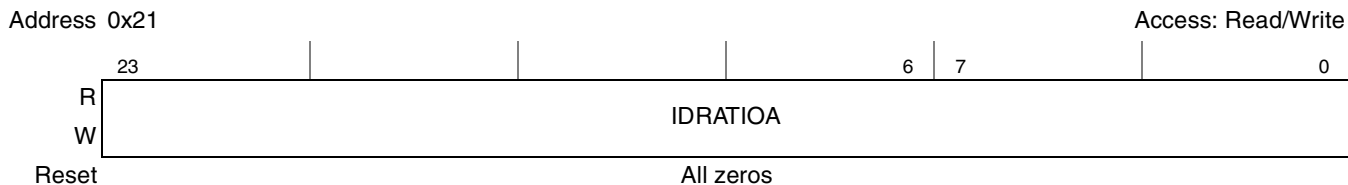
### 19.2.2.12 ASRC Data Input and Output Registers

#### 19.2.2.12.1 Ideal Ratio Registers for Pair A, High/Low Part (ASRIDRHA, ASRIDRLA)

These are three 24-bit wide register for reading data from the output data FIFOs Ideal Ratio Registers for Pair A, High/Low Part (ASRIDRHA, ASRIDRLA) The ideal ratio registers (ASRIDRHA, ASRIDRLA) hold the ratio value IDRATIOA.  $IDRATIOA = F_{SinA}/F_{SoutA} = T_{SoutA}/T_{SinA}$  is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRA, IDRA}=2’b11.



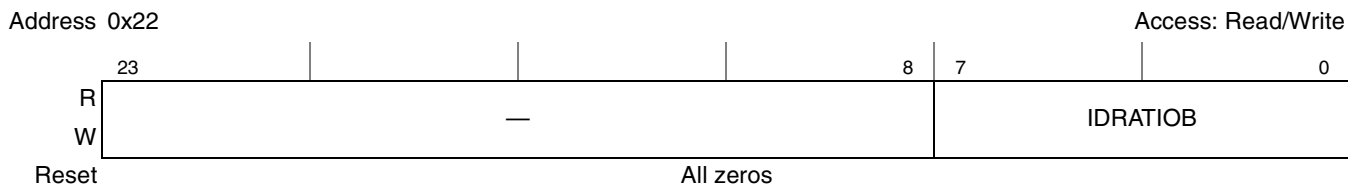
**Figure 19-15. Ideal Ratio Register for Pair A, High Part (ASRIDRHA)**



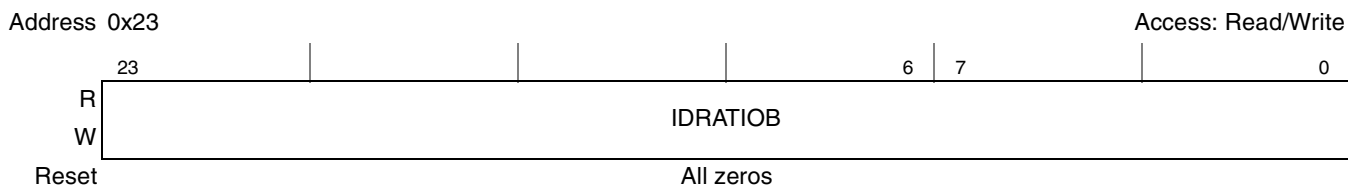
**Figure 19-16. Ideal Ratio Register for Pair A, Low Part (ASRIDRLA)**

**19.2.2.12.2 Ideal Ratio Registers for Pair B, High/Low Part (ASRIDRHB, ASRIDRLB)**

The ideal ratio registers (ASRIDRHB, ASRIDRLB) hold the ratio value IDRATIOB.  $IDRATIOB = F_{SinB}/F_{SoutB} = T_{SoutB}/T_{SinB}$  is a 32-bit fixed point value with 26 fractional bits. This value is only useful when  $ASRCTR:\{USRB, IDRB\}=2'b11$ .



**Figure 19-17. Ideal Ratio Register for Pair B, High Part (ASRIDRHB)**



**Figure 19-18. Ideal Ratio Register for Pair B, Low Part (ASRIDRLB)**

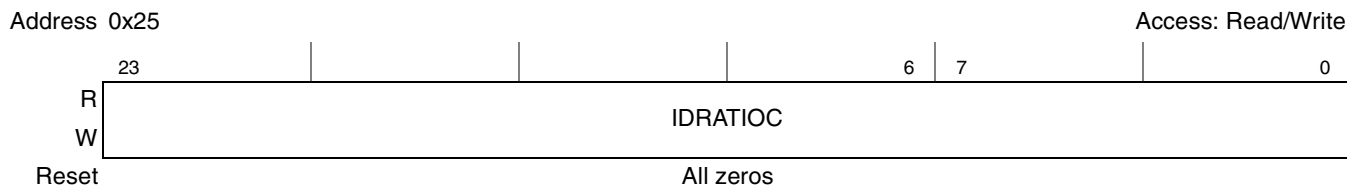
**19.2.2.12.3 Ideal Ratio Registers for Pair C, High/Low Part (ASRIDRHC, ASRIDRLC)**

The ideal ratio registers (ASRIDRHC, ASRIDRLC) hold the ratio value IDRATIOC.  $IDRATIOC = F_{SinC}/F_{SoutC} = T_{SoutC}/T_{SinC}$  is a 32-bit fixed point value with 26 fractional bits. This value is only useful when  $ASRCTR:\{USRC, IDRC\}=2'b11$ .



**Figure 19-19. Ideal Ratio Register for Pair C, High Part (ASRIDRHC)**

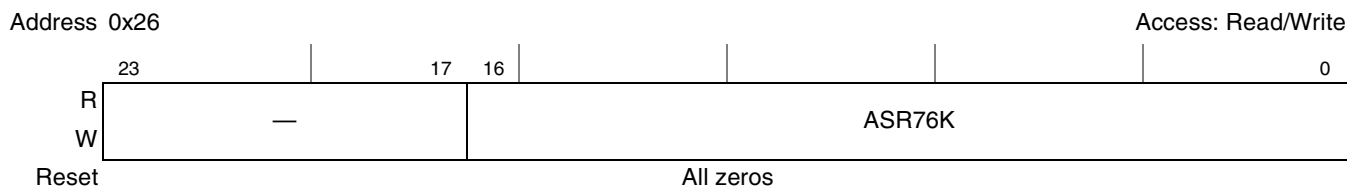
## Asynchronous Sample Rate Converter



**Figure 19-20. Ideal Ratio Register for Pair C, Low Part (ASRIDRLC)**

### 19.2.2.12.4 ASRC 76 kHz Period Register in Terms of Master Clock (ASR76K)

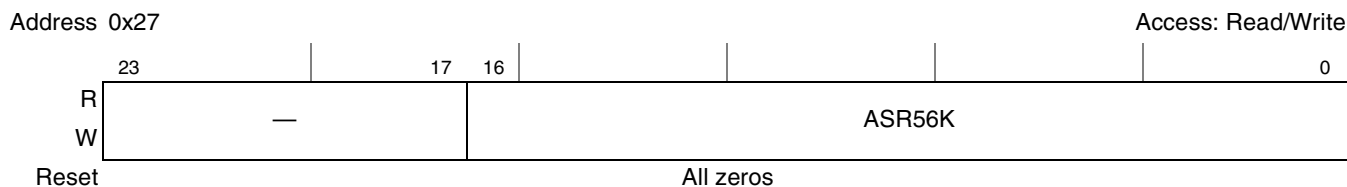
The register (ASR76K) holds the period of the 76 kHz sampling clock in terms of the master clock.  $ASR76K = F_{S_{master}}/F_{S_{76k}}$ . Reset value is 0x0A47 which assumes that  $F_{S_{master}}=200$  MHz. This register is used to help the ASRC internal logic to decide the pre-processing and the post-processing options automatically (see [Section 19.2.2.1, “ASRC Control Register \(ASRCTR\),”](#) and [Section 19.2.2.4, “Filter Configuration Status Register \(ASRCFG\)”](#)).



**Figure 19-21. ASRC 76 kHz Period Register (ASR76K)**

### 19.2.2.12.5 ASRC 56 kHz Period Register in Terms of Master Clock (ASR56K)

The register (ASR56K) holds the period of the 56 kHz sampling clock in terms of the master clock.  $ASR56K = F_{S_{master}}/F_{S_{56k}}$ . Reset value is 0x0DF3 which assumes that  $F_{S_{master}}=200$  MHz. This register is used to help the ASRC internal logic to decide the pre-processing and the post-processing options automatically (see [Section 19.2.2.1, “ASRC Control Register \(ASRCTR\),”](#) and [Section 19.2.2.4, “Filter Configuration Status Register \(ASRCFG\)”](#)).



**Figure 19-22. ASRC 56kHz Period Register (ASR56K)**

## 19.3 Interrupts

The ASRC has several interrupt events. When the interrupt request is active, the interrupt vector has the choices, shown in [Table 19-16](#).

**Table 19-16. Interrupt Vectors**

Offset	Description
0x0	ASRC Pair A input data is needed.
0x2	ASRC Pair B input data is needed.

**Table 19-16. Interrupt Vectors (Continued)**

Offset	Description
0x4	ASRC Pair C input data is needed.
0x6	ASRC Pair A output data is ready.
0x8	ASRC Pair B output data is ready.
0xA	ASRC Pair C output data is ready.
0xC	ASRC Overload
0xE	ASRC FP Wait State

## 19.4 DMA Requests

ASRC has six DMA requests. The six DMA requests are directly connected to the lowest six status bits in the ASRSTR register, shown in [Table 19-17](#).

**Table 19-17. DMA Requests**

Type	Description
0	ASRC Pair A input data is needed.
1	ASRC Pair B input data is needed.
2	ASRC Pair C input data is needed.
3	ASRC Pair A output data is ready.
4	ASRC Pair B output data is ready.
5	ASRC Pair C output data is ready.

## 19.5 Functional Description

### 19.5.1 Algorithm Description

#### 19.5.1.1 Signal Processing Flow

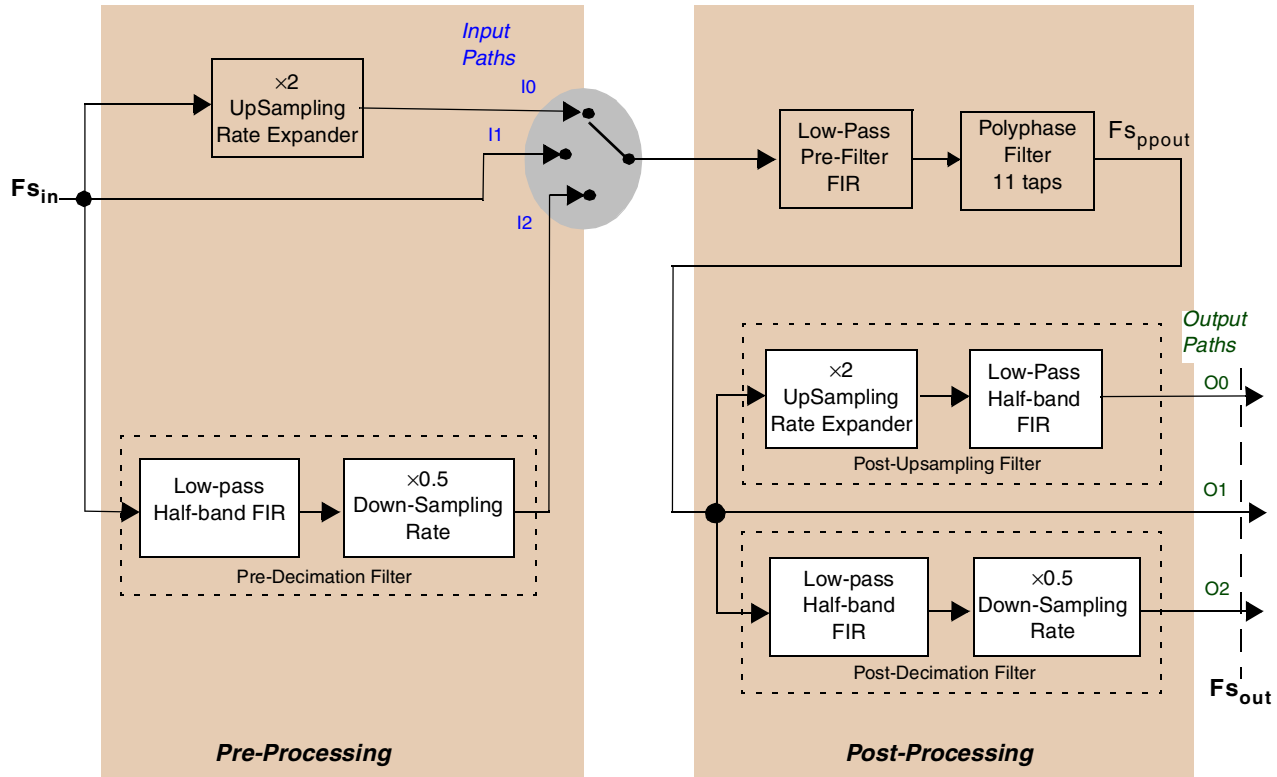


Figure 19-23. Signal Processing Configurations for ASRC

Figure 19-23 shows the possible signal processing configurations for the ASRC. Each configuration consists of 2 to 4 stages, as follows:

- $\times 2$  up-sampling rate expander (zero insertion only) (Input path I0),  
or direct connection (Input path I1),  
or low-pass pre-decimation filter (consisting of a low-pass half-band FIR filter with  $\times 0.5$  downsampling rate decimator) (Input path I2)
- Low-pass pre-filter, the low-pass bandwidth is at most  $0.25 \times F_s$ , where  $F_s$  is the sampling rate of the input signal to this low-pass pre-filter,
- Polyphase filter
- $\times 2$  post-upsampling filter (consisting of a  $\times 2$  up-sampling rate expander (zero insertion only) with low-pass half-band FIR filter) (Output path O0),  
or direct connection (Output path O1),

or low-pass post-decimation filter (consisting of a low-pass half-band FIR filter with x0.5 downsampling rate decimator) (Output path O2).

By using different input and output paths, and different set-ups of the pre-filter, this ASRC scheme can be used to handle different rate conversion requirements, shown in Table 19-18.

**Table 19-18. Possible ASRC Configurations**

Config	Input Path	Output Path	Description
a	I0	O0	The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/2$ . The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}/2$ .
b	I0	O1	The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/2$ . The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}$ .
c	I0	O2	The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/2$ . The signal sampling rate of the polyphase filter output is $Fs_{ppout} = 2Fs_{out}$ .
d	I1	O0	The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/4$ . The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}/2$ .
e	I1	O1	The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/4$ . The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}$ .
f	I1	O2	The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/4$ . The signal sampling rate of the polyphase filter output is $Fs_{ppout} = 2Fs_{out}$ .
g	I2	O0	The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/8$ . The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}/2$ .
h	I2	O1	The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/8$ . The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}$ .
i	I2	O2	The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/8$ . The signal sampling rate of the polyphase filter output is $Fs_{ppout} = 2Fs_{out}$ .

The suggested paths of the pre-processing and post-processing operations w.r.t the standard sampling rates shown in [Table 19-19](#).

**Table 19-19. Pre-Processing, Post-Processing Options**

{Pre_Proc, Post_Proc}		Fsout (kHz)								
		8	32	44.1	48	64	88.2	96	128	192
Fsin (kHz)	8	{0, 1}	{0, 0}	{0, 0}	{0, 0}	{0, 0}	{0, 0}	{0, 0}	{0, 0}	{0, 0}
	12	{0, 2}	{0, 1}	{0, 0}	{0, 0}	{0, 0}	{0, 0}	{0, 0}	{0, 0}	{0, 0}
	16	{1, 2}	{0, 1}	{0, 1}	{0, 1}	{0, 0}	{0, 0}	{0, 0}	{0, 0}	{0, 0}
	24	{1, 2}	{0, 1}	{0, 1}	{0, 1}	{0, 1}	{0, 0}	{0, 0}	{0, 0}	{0, 0}
	32	{1, 2}	{0, 1}	{0, 1}	{0, 1}	{0, 1}	{0, 1}	{0, 1}	{0, 1}	{0, 0}
	44.1	{2, 2}	{0, 2}	{0, 1}	{0, 1}	{0, 1}	{0, 1}	{0, 1}	{0, 1}	{0, 0}
	48	{2, 2}	{0, 2}	{0, 2}	{0, 1}	{0, 1}	{0, 1}	{0, 1}	{0, 1}	{0, 0}
	64	{2, 2}	{1, 2}	{0, 2}	{0, 2}	{0, 1}	{0, 1}	{0, 1}	{0, 1}	{0, 0}
	88.2	N/A	{1, 2}	{1, 2}	{1, 2}	{1, 1}	{1, 1}	{1, 1}	{1, 1}	{1, 0}
	96	N/A	{1, 2}	{1, 2}	{1, 2}	{1, 1}	{1, 1}	{1, 1}	{1, 1}	{1, 1}
	128	N/A	{1, 2}	{1, 2}	{1, 2}	{1, 1}	{1, 1}	{1, 1}	{1, 1}	{1, 1}
192	N/A	{2, 2}	{2, 2}	{2, 2}	{2, 1}	{2, 1}	{2, 1}	{2, 1}	{2, 1}	

**Comments:**

In the {Pre\_Proc, Post\_Proc} pair, the meaning of the values {x, y} are:

Pre\_Proc:

- 0 Pre-processing input path I0 as shown in [Figure 19-23](#)
- 1 Pre-processing input path I1 as shown in [Figure 19-23](#)
- 2 Pre-processing input path I2 as shown in [Figure 19-23](#)

Post\_Proc:

- 0 Post-processing output path O0 as shown in [Figure 19-23](#)
- 1 Post-processing output path O1 as shown in [Figure 19-23](#)
- 2 Post-processing output path O2 as shown in [Figure 19-23](#)

## 19.5.1.2 Miscellaneous Topics

### 19.5.1.2.1 Support of Physical Clocks

The device supports only physical sampling clocks. The clocks can be the clocks from SPDIF, ESAI, or the PLL.

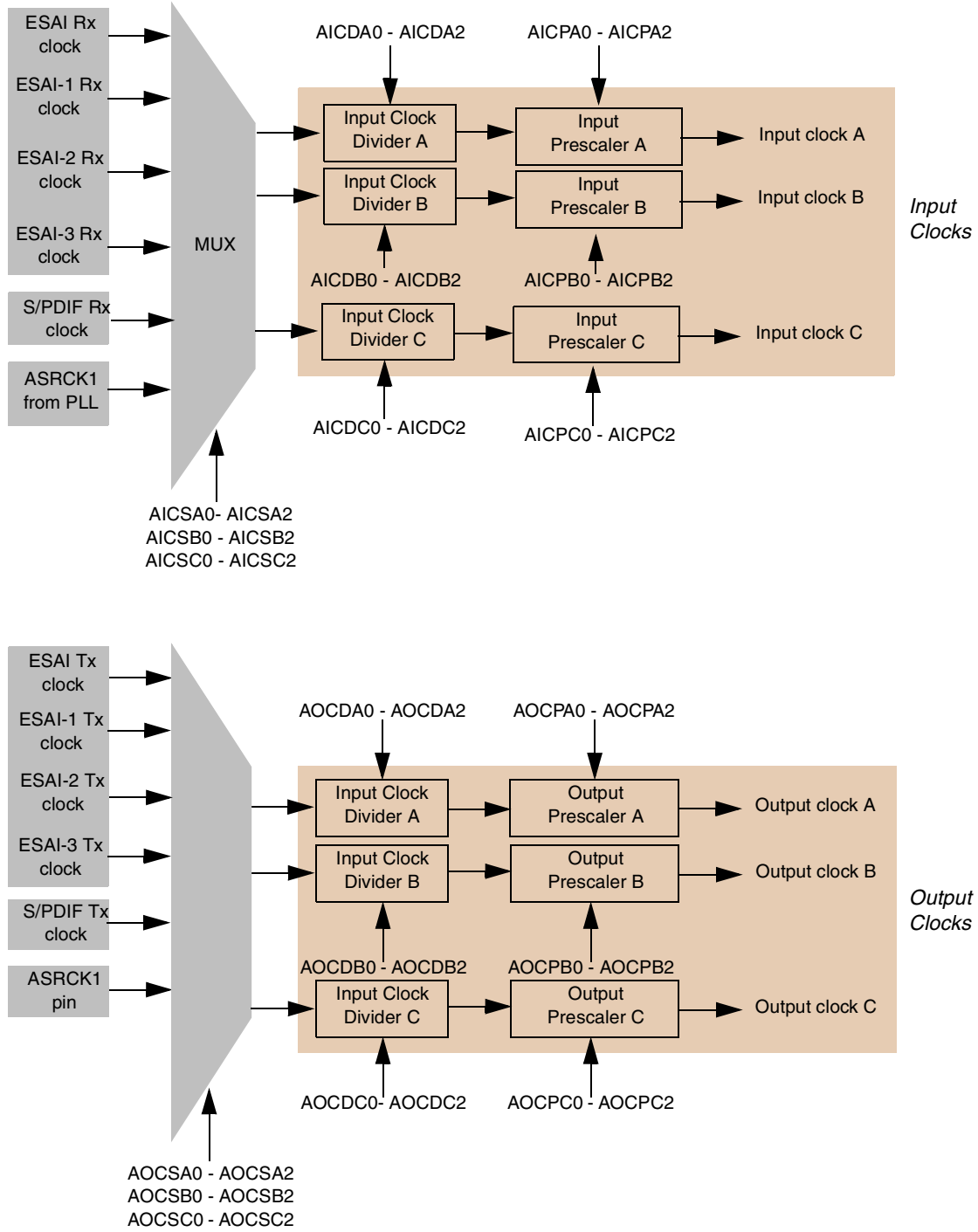


### 19.5.1.2.2 Physical Clock Source Selector and Divider

The DSP program can set the Clock Source Register (ASRCSR) and Clock Divider Registers (ASRCDR1, ASRCDR2) to choose the clock source and divide them to sample rate clock for internal use. See [Figure 19-24](#).

There is a restriction with the clocks: if the prescaler is set to 1, the Clock Divider can only be set to 1 and the clock source duty cycle must be 50%.

# Asynchronous Sample Rate Converter



**Figure 19-24. Clock Source Selector and Divider**

### 19.5.1.2.3 Configure ASRCDR1/ASRCDR2

The ASRC processes the audio data at the same frequency as the sampling clock of the audio stream, that is: with one audio sample input, there will be one iteration of rate conversion taking place inside the ASRC logic. If the clock source selected from either ESAI, SPDIF, and so on is a multiple of the audio sampling clock, it must be divided by the correct selection of ASRCDR factors.

For example, if a 3.072 MHz clock as ASRCK1 (from PLL) is connected to ASRC pair A and its desired audio sampling rate is 48 kHz, then the divide factor must be 64 ( $48 * 64 = 3.072$  K). Next, the ASRC Clock Divide register can be configured with the following:

```
ASRCDR1:AICDA =0 ; ASRCDR1:AIDPA = 0x6; ASRCDR2 = 0;
```

For SPDIF:

- If S/PDIF is used as a receiver, the divide factor should be 128.
- If S/PDIF is used as transmitter, the divide factor should be 64.

For ESAI and PLL clock, the divide factor should be calculated according to the corresponding configurations in the ESAI and PLL module, and the value can be obtained following the process of the example shown previously. If the ESAI clock is from off-chip, then the ESAI Tx (Rx) clock in [Figure 19-24](#) is connected with the SCKT (SCKR) in ESAI module ([Figure 9-3](#)).



# Chapter 20

## Chip Configuration Module

### 20.1 Introduction

The Chip Configuration module contains several registers which are used to select the features of the chip and to control some peripheral devices' pin switching or pin mux. Chip Configuration module features includes:

- Controls Shared Bus Arbitration and EMC Burst Mode and EMC Burst Mode
- Pin Mux control of ESAI and S/PDIF Rx Clock output mux on ESAI HCKR pins
- Shared Peripherals soft reset triggering and auto de-assertion
- EMC phase-locked loop (PLL) control and status

#### 20.1.1 Modes of Operation

The chip configuration registers can be accessed by both DSP cores.

### 20.2 Memory Map and Register Definition

[Table 20-1](#) shows the memory map for all the chip configuration registers.

#### 20.2.1 Memory Map

**Table 20-1. Chip Configuration Module Memory Map**

Offset or Address	Register	Access	Reset Value	Section/Page
y: 0xFFFFE7	Reserved	R	0x000000	<a href="#">20.2.2.1/20-3</a>
y: 0xFFFFE6	External Memory Burst Control	R/W	0x000000	<a href="#">20.2.2.3/20-3</a>
y: 0xFFFFE5	EMC PLL Status and Control	R/W	0x000002	<a href="#">20.2.2.4/20-5</a>
y: 0xFFFFE4	Pin Mux Control	R/W	0x000000	<a href="#">20.2.2.5/20-7</a>
y: 0xFFFFE3	ESAI Pin Switch Control	R/W	0x000000	<a href="#">20.2.2.6/20-8</a>
y: 0xFFFFE2	Once Debug and Burst Control	R/W	0x000000	<a href="#">20.2.2.7/20-10</a>
y: 0xFFFFE1	Shared Peripherals Soft Reset Control	rwm	0x000000	<a href="#">20.2.2.8/20-11</a>
y: 0xFFFFE0	Shared Bus Arbitration Mode Configuration	R/W	0x000000	<a href="#">20.2.2.9/20-12</a>

Table 20-2 shows the register summary table for Chip-configuration module.

**Table 20-2. CFG Register Summary**

Name		23	22	21	20	19	18	17	16	15	14	13	12
		11	10	9	8	7	6	5	4	3	2	1	0
Reserved Y: FFFFE7	R												
	W												
	R												
	W												
EMBC Y: FFFFE6	R	EPMBC			P Space Burst Boundary				EYMBC				
	W												
	R	Y Space Burst Boundary				EXMBC		X Space Burst Boundary					
	W												
LPSC Y: FFFFE5	R	lplock											
	W												
	R										EMC PLL Power Down and clock frequency divide control		
	W												
PMC Y: FFFFE4	R	PKG			ESAI, HCKR Pin Mux Select				spdout1_en	spdin1_en			
	W												
	R	SPDIF Pin Mux Control						Timer Pin Mux Control					
	W												
EPSC Y: FFFFE3	R	ESAI Pin Switch Control.1											
	W												
	R	ESAI Pin Switch Control.0											
	W												
ODBC Y: FFFFE2	R	ONCE											
	W	Debug Enable											
	R									Burst Buffer Invalidate			
	W												
PSRC Y: FFFFE1	R												
	W												
	R										Soft Reset Control Bits		
	W												

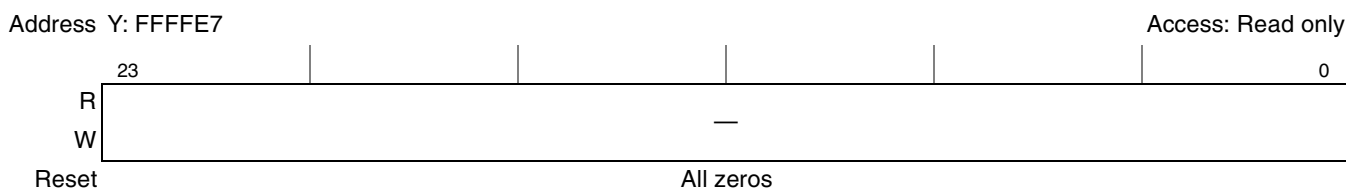
**Table 20-2. CFG Register Summary (Continued)**

Name		23	22	21	20	19	18	17	16	15	14	13	12
		11	10	9	8	7	6	5	4	3	2	1	0
<b>ARCR</b> Y: FFFFE0	R					Shared Bus Arbiter Control							
	W												
	R	Shared Bus Arbiter Control											
	W												

## 20.2.2 Register Descriptions

### 20.2.2.1 Reserved Register

This register is a 24-bit read-only register, shown in [Figure 20-1](#).



**Figure 20-1. Reserved Control Register**

[Table 20-3](#) defines the reserved control register fields.

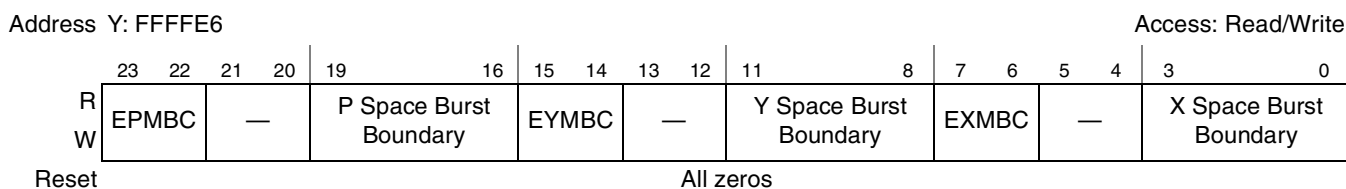
**Table 20-3. Reserved Control Register Field Description**

Bit	Name	Description
23–0	—	Reserved

### 20.2.2.2 EMBC (External Memory Burst Control) Register

#### 20.2.2.3 EMBC Register

The external memory burst buffer control register is shown in [Figure 20-2](#).



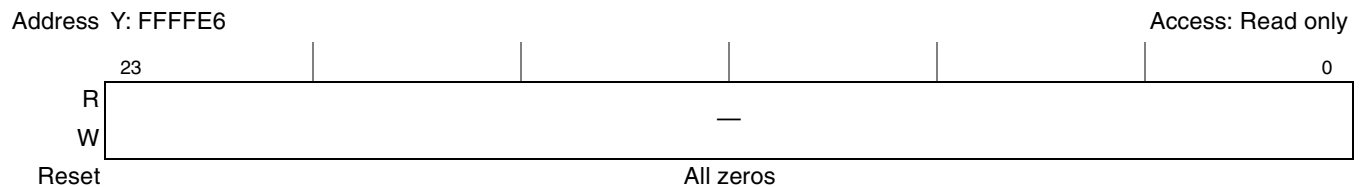
**Figure 20-2. External Memory Burst Buffer Control Register—DSP56724**

Table 20-4 defines the EMBC register fields for DSP56724.

**Table 20-4. EMBC Register Field Description—DSP56724**

Bits	Name	Descriptions
23–22	EPMBC	Burst control for external P memory space 00 Burst buffer is disabled for the access to external P address space 01 Burst buffer is enabled for the access to external P address space, except for external P address space determined by P Space Burst Boundary 10 Burst buffer is enabled for any access to external P address space 11 Reserved
21–20	—	Reserved. Write 0 for future compatibility.
19–16	P Space Burst Boundary	External peripherals P space base address. The external peripheral space is used to connect external I/O devices for which burst access is not needed. The external peripheral space is determined by the address mapping shown in <a href="#">Table 20-6</a>
15–14	EYMBC	Burst control for external Y memory space 00 Burst buffer is disabled for the access to external Y address space 01 Burst buffer is enabled for the access to external Y address space, except for external Y address space determined by Y Space Burst Boundary 10 Burst buffer is enabled for any access to external Y address space 11 Reserved
13–12	—	Reserved. Write 0 for future compatibility.
11–8	Y Space Burst Boundary	External peripherals Y space base address. The external peripheral space is used to connect external I/O devices for which burst access is not needed. The external peripheral space is determined by the boundary bits shown in <a href="#">Table 20-6</a>
7–6	EXMBC	Burst control for external X memory space 00 Burst buffer is disabled for the access to external X address space 01 Burst buffer is enabled for the access to external X address space, except for X external address space determined by X Space Burst Boundary 10 Burst buffer is enabled for any access to external X address space 11 Reserved
5–4	—	Reserved. Write 0 for future compatibility.
3–0	X Space Burst Boundary	External peripherals X space base address. The external peripheral space is used to connect external I/O devices for which burst access is not needed. The external peripheral space is determined by the address mapping shown in <a href="#">Table 20-6</a>

Figure 20-3 shows the bit fields for the EMBC register for DSP56725.



**Figure 20-3. EMBC Register, DSP56725**



Table 20-5 defines the EMBC register fields for DSP56725.

**Table 20-5. Field Descriptions, DSP56725**

Bits	Name	Description
23–0	—	Reserved. Write 0 for future compatibility.

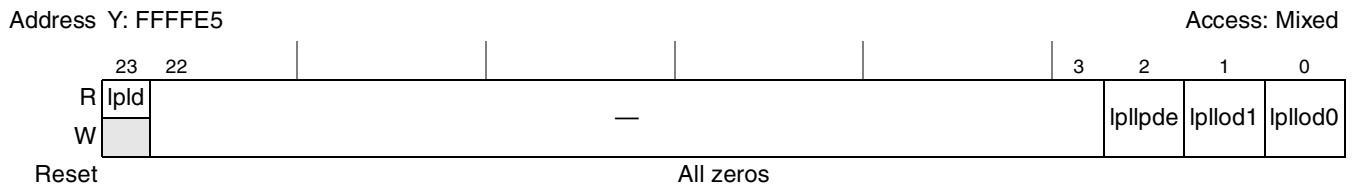
Table 20-6 lists the external peripherals.

**Table 20-6. External Peripherals X,Y,P Space Set by Burst Boundary**

External Device X,Y,P Burst Boundary	Address Range Not Bursted
0000	0x040000–0x0FFFFFF
0001	0x100000–0x1FFFFFF
0010	0x200000–0x2FFFFFF
0011	0x300000–0x3FFFFFF
0100	0x400000–0x4FFFFFF
0101	0x500000–0x5FFFFFF
0110	0x600000–0x6FFFFFF
0111	0x700000–0x7FFFFFF
1000	0x800000–0x8FFFFFF
1001	0x900000–0x9FFFFFF
1010	0xA00000–0xAFFFFFF
1011	0xB00000–0xBFFFFFF
1100	0xC00000–0xCFFFFFF
1101	0xD00000–0xDFFFFFF
1110	0xE00000–0xEFFFFFF
1111	0xF00000–0xF7FFFF

### 20.2.2.4 LPSC (EMC PLL Status and Control) Register

The LPSC control register is shown in Figure 20-4 and Figure 20-5.

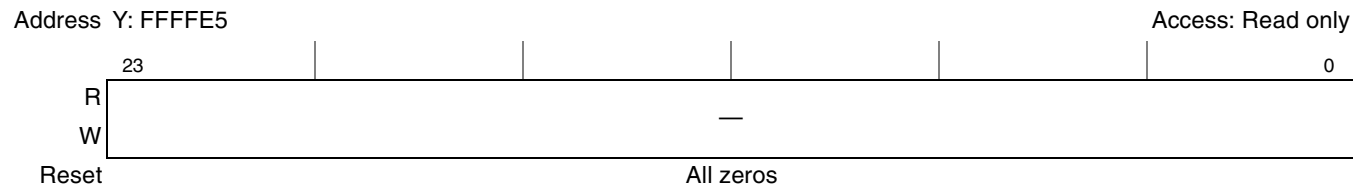


**Figure 20-4. EMC PLL Status and Control Register (PSC)—DSP56724**

Table 20-7 defines the LPSC register fields for DSP56724.

**Table 20-7. LPSC Register Field Description—DSP56724**

Bits	Name	Descriptions
23	lpld	EMC PLL Lock Detection Status bit. 1 lock detected 0 lock undetected
22–3	—	Reserved. Write 0 for future compatibility
2	lplpde	EMC PLL Power Down Enable 1 Power Down Enabled 0 Power Down Disabled
1–0	lpllod[1:0]	EMC PLL output clock frequency Divide Ratio. 00 output clock not divided 01 output clock divided by 2 10 output clock divided by 4(default reset value) 11 output clock divided by 8



**Figure 20-5. LPSC Register—DSP56725**

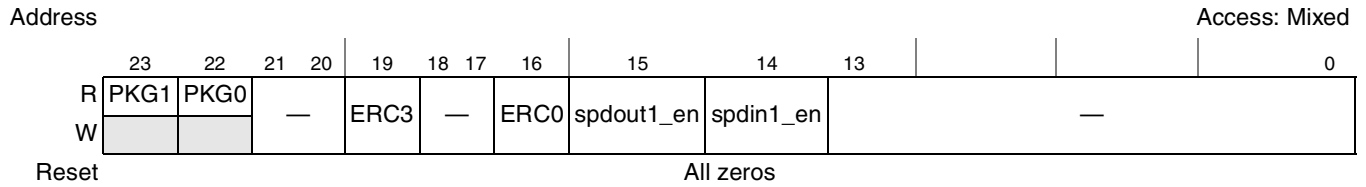
Table 20-7 defines the LPSC register fields for DSP56725.

**Table 20-8. Field Descriptions—DSP56725**

Bit	Name	Description
23–0	—	Reserved. Write 0 for future compatibility.

### 20.2.2.5 Pin Mux Control Register (PMC)

The PMC control register is shown in [Figure 20-6](#).



**Figure 20-6. Pin Mux Control Register**

[Table 20-9](#) defines the PMC register fields.

**Table 20-9. PMC Field Descriptions**

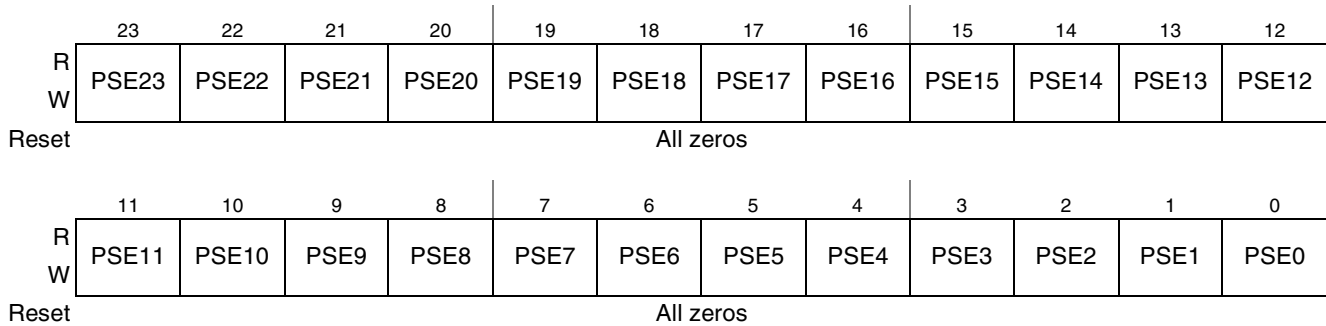
Bit	Name	Description
23–22	PKG	Read-only Chip Package Info 2'b10 DSP56725 80-pin QFP package 2'b01 Reserved 2'b11 DSP56724 144-pin QFP package The reset value of these 2 bits is determined by the product's package.
21–20	—	Reserved. Write 0 for future compatibility.
19	ERC3	S/PDIF Rx Clock Output via ESAI_3 HCKR Pin Select 0 No S/PDIF clock output via HCKR_3 pin 1 Select the S/PDIF Rx clock output via the HCKR_3 pin other than external clock when the corresponding ESAI External clock control bit is set. Note: HCKR_3 pin is not available on DSP56725 80-pin packages.
18–17	—	Reserved
16	ERC0	S/PDIF Rx Clock Output via ESAI HCKR Pin Select 0 No S/PDIF clock output via HCKR pin 1 Select the S/PDIF Rx Clock output via the HCKR pin other than external clock when the corresponding ESAI External clock control bit is set.
15	spdifout1_en	SPDIFOUT1 Pin Mux Control 1 ESAI_2's data pin SDO3/SDI2 is set to the SPDIFOUT1 function. 0 ESAI_2's data pin SDO3/SDI2 is set to the ESAI_2 SDO3/SDI2 function. Note: This bit only applies to DSP56725 80-pin packages.
14	spdifin1_en	SPDIFIN1 Pin Mux Control 1 ESAI_2's data pin SDO2/SDI3 is set to the SPDIFIN1 function. 0 ESAI_2's data pin SDO2/SDI3 is set to the ESAI_2 SDO2/SDI3 function. Note: This bit only applies to the DSP56725 80-pin package.
13–0	—	Reserved

### 20.2.2.6 ESAI Pin Switch Control Register (EPSC)

The EPSC register is shown in [Figure 20-7](#).

Address Y: FFFFE3

Access: Read/Write



**Figure 20-7. ESAI Pin Switch Control Register (EPSC)**

[Table 20-10](#) defines the EPSC register fields.

**Table 20-10. EPSC Field Descriptions**

Bits	Name	Description
23	PSE[23]	Pin Switch Control bits for ESAI_1 Pin SDO0_1 and ESAI_3 Pin SDO0_3 0 Pin Switch Disabled 1 Pin Switch Enabled
22	PSE[22]	Pin Switch Control bits for ESAI_1 Pin SDO1_1 and ESAI_3 Pin SDO1_3 0 Pin Switch Disabled 1 Pin Switch Enabled
21	PSE[21]	Pin Switch Control bits for ESAI_1 Pin SDI3_1/SDO2_1 and ESAI_3 Pin SDI3_3/SDO2_3 0 Pin Switch Disabled 1 Pin Switch Enabled
20	PSE[20]	Pin Switch Control bits for ESAI_1 Pin SDI2_1/SDO3_1 and ESAI_3 Pin SDI2_3/SDO3_3 0 Pin Switch Disabled 1 Pin Switch Enabled
19	PSE[19]	Pin Switch Control bits for ESAI_1 Pin SDI1_1/SDO4_1 and ESAI_3 Pin SDI1_3/SDO4_3 0 Pin Switch Disabled 1 Pin Switch Enabled
18	PSE[18]	Pin Switch Control bits for ESAI_1 Pin SDI0_1/SDO5_1 and ESAI_3 Pin SDI0_3/SDO5_3 0 Pin Switch Disabled 1 Pin Switch Enabled
17	PSE[17]	Pin Switch Control bits for ESAI_1 Pin HCKT_1 and ESAI_3 Pin HCKT_3 0 Pin Switch Disabled 1 Pin Switch Enabled
16	PSE[16]	Pin Switch Control bits for ESAI_1 Pin FST_1 and ESAI_3 Pin FST_3 0 Pin Switch Disabled 1 Pin Switch Enabled

**Table 20-10. EPSC Field Descriptions (Continued)**

Bits	Name	Description
15	PSE[15]	Pin Switch Control bits for ESAI_1 Pin SCKT_1 and ESAI_3 Pin SCKT_3 0 Pin Switch Disabled 1 Pin Switch Enabled
14	PSE[14]	Pin Switch Control bits for ESAI_1 Pin HCKR_1 and ESAI_3 Pin HCKR_3 0 Pin Switch Disabled 1 Pin Switch Enabled
13	PSE[13]	Pin Switch Control bits for ESAI_1 Pin FSR_1 and ESAI_3 Pin FSR_3 0 Pin Switch Disabled 1 Pin Switch Enabled
12	PSE[12]	Pin Switch Control bits for ESAI_1 Pin SCKR_1 and ESAI_3 Pin SCKR_3 0 Pin Switch Disabled 1 Pin Switch Enabled
11	PSE[11]	Pin Switch Control bits for ESAI Pin SDO0 and ESAI_2 Pin SDO0_2 0 Pin Switch Disabled 1 Pin Switch Enabled
10	PSE[10]	Pin Switch Control bits for ESAI Pin SDO1 and ESAI_2 Pin SDO1_2 0 Pin Switch Disabled 1 Pin Switch Enabled
9	PSE[9]	Pin Switch Control bits for ESAI Pin SDI3/SDO2 and ESAI_2 Pin SDI3_2/SDO2_2 0 Pin Switch Disabled 1 Pin Switch Enabled
8	PSE[8]	Pin Switch Control bits for ESAI Pin SDI2/SDO3 and ESAI_2 Pin SDI2_2/SDO3_2 0 Pin Switch Disabled 1 Pin Switch Enabled
7	PSE[7]	Pin Switch Control bits for ESAI Pin SDI1/SDO4 and ESAI_2 Pin SDI1_2/SDO4_2 0 Pin Switch Disabled 1 Pin Switch Enabled
6	PSE[6]	Pin Switch Control bits for ESAI Pin SDI0/SDO5 and ESAI_2 Pin SDI0_2/SDO5_2 0 Pin Switch Disabled 1 Pin Switch Enabled
5	PSE[5]	Pin Switch Control bits for ESAI Pin HCKT and ESAI_2 Pin HCKT 0 Pin Switch Disabled 1 Pin Switch Enabled
4	PSE[4]	Pin Switch Control bits for ESAI Pin FST and ESAI_2 Pin FST 0 Pin Switch Disabled 1 Pin Switch Enabled
3	PSE[3]	Pin Switch Control bits for ESAI Pin SCKT and ESAI_2 Pin SCKT 0 Pin Switch Disabled 1 Pin Switch Enabled
2	PSE[2]	Pin Switch Control bits for ESAI Pin HCKR and ESAI_2 Pin HCKR 0 Pin Switch Disabled 1 Pin Switch Enabled

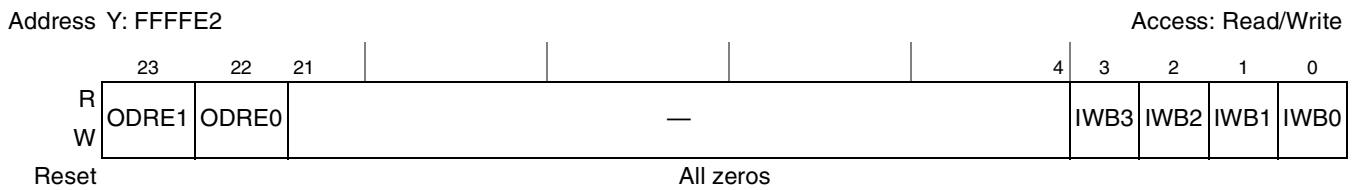
**Table 20-10. EPSC Field Descriptions (Continued)**

Bits	Name	Description
1	PSE[1]	Pin Switch Control bits for ESAI Pin FSR and ESAI_2 Pin FSR 0 Pin Switch Disabled 1 Pin Switch Enabled
0	PSE[0]	Pin Switch Control bits for ESAI Pin SCKR and ESAI_2 Pin SCKR 0 Pin Switch Disabled 1 Pin Switch Enabled

**Note:** See the product pin-out information for using these bits to switch the available pins on the DSP56724 or DSP56725.

### 20.2.2.7 ONCE Debug and Burst Control Register (ODBC)

The ODBC Control Register is shown in [Figure 20-8](#).



**Figure 20-8. ONCE Debug and Burst Control Register (ODBC)**

[Table 20-11](#) defines the ODBC register fields.

**Table 20-11. ODBC Field Descriptions**

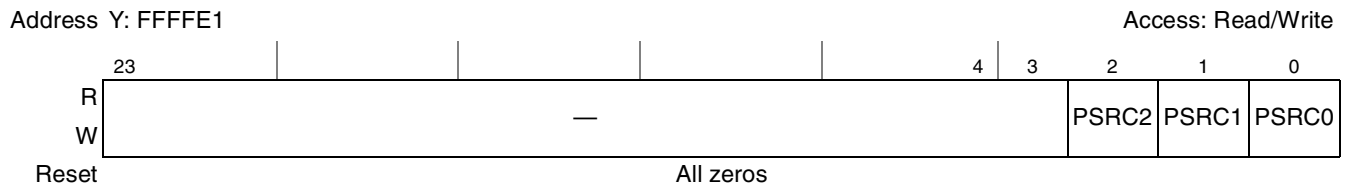
Bit	Name	Description
23	ODRE1	ONCE Debug Request from Core-1 to Core-0 0 Disable ONCE Debug Request from Core-1 to Core-0 1 Enable ONCE Debug Request from Core-1 to Core-0
22	ODRE0	ONCE Debug Request from Core-0 to Core-1 0 Disable ONCE Debug Request from Core-0 to Core-1 1 Enable ONCE Debug Request from Core-0 to Core-1
21-4	—	Reserved. Write 0 for future compatibility.
3	IWB1	Invalidating the write buffer of Core-1(DMA 1) 0 Not invalidating the write buffer of Core-1(DMA 1) 1 invalidating the write buffer of Core-1(DMA 1) Writing 1 to this bit will invalidate the write-buffer of Core-1(DMA 1), and it is automatically cleared by hardware when Core-1(DMA 1) write-buffer invalidation acknowledge asserted
2	IRB1	Invalidating the read buffer of Core-1(DMA 1) 0 Not invalidating the read buffer of Core-1(DMA 1) 1 Invalidating the read buffer of Core-1(DMA 1) Writing 1 to this bit will invalidate the read-buffer of Core-1(DMA 1), and it is automatically cleared by hardware when Core-1(DMA 1) read-buffer invalidation acknowledge asserted

**Table 20-11. ODBC Field Descriptions (Continued)**

Bit	Name	Description
1	IWB0	Invalidating the write buffer of Core-0(DMA 0) 0 Not invalidating the write buffer of Core-0(DMA 0) 1 invalidating the write buffer of Core-0(DMA 0) Writing 1 to this bit will invalidate the write-buffer of Core-0(DMA 1), and it is automatically cleared by hardware when Core-1(DMA 1) write-buffer invalidation acknowledge asserted.
0	IRB0	Invalidating the read buffer of Core-0(DMA 0) 0 Not invalidating the read buffer of Core-0(DMA 0) 1 Invalidating the read buffer of Core-0(DMA 0) Writing 1 to this bit will invalidate the read-buffer of Core-0(DMA 0), and it is automatically cleared by hardware when Core-0(DMA 0) read-buffer invalidation acknowledge asserted.

### 20.2.2.8 Peripheral Soft Reset Control Register (PSRC)

The PSRC Control Register is shown in [Figure 20-9](#).



**Figure 20-9. Peripheral Soft Reset Control Register (PSRC)**

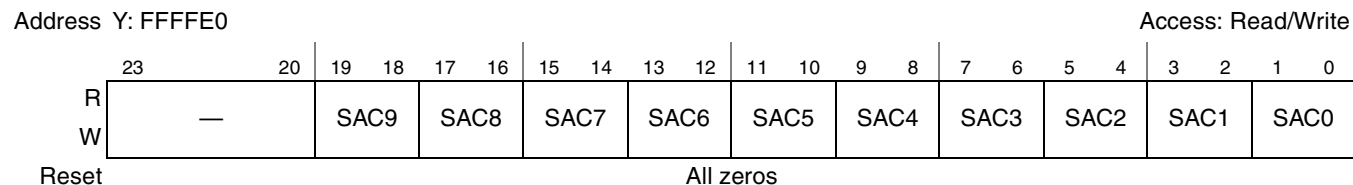
[Table 20-12](#) defines the PSRC register fields.

**Table 20-12. PSRC Field Description**

Bit	Name	Description
23–3	—	Reserved. Write 0 for future compatibility.
2	PSRC2	ASRC Soft Reset Trigger Bit Writing 1 to this bit will cause a soft reset of the ASRC Block; the reset period is 6 system clock cycles. This bit is cleared by hardware after the reset period reaches 6 system clock cycles.
1	PSRC1	EMC Soft Reset Trigger Bit Writing 1 to this bit will cause a soft reset of the EMC module; the reset period is 6 system clock cycles. This bit is cleared by hardware after the reset period reaches 6 system clock cycles.
0	PSRC0	S/PDIF Soft Reset Trigger Bit Writing 1 to this bit will cause a soft reset of the S/PDIF Block; the reset period is 6 system clock cycles. This bit is cleared by hardware after the reset period reaches 6 system clock cycles.

### 20.2.2.9 Arbiter Control Register (ARCR)

The ARCR Control Register is shown in [Figure 20-10](#).



**Figure 20-10. Arbiter Control Register (ARCR)**

[Table 20-13](#) defines the ARCR register fields.

**Table 20-13. ARCR Field Descriptions**

Bit	Name	Description
23–20	—	Reserved. Write 0 for future compatibility.
19–18	SAC9	Shared bus arbitration method select for accessing external memories.
17–16	SAC8	Shared Bus Arbiter arbitration method select for peripherals on the shared peripherals bus.
15–14	SAC7	Shared Bus Arbiter arbitration method select for shared memory block 7
13–12	SAC6	Shared Bus Arbiter arbitration method select for shared memory block 6
11–10	SAC5	Shared Bus Arbiter arbitration method select for shared memory block 5
9–8	SAC4	Shared Bus Arbiter arbitration method select for shared memory block 4
7–6	SAC3	Shared Bus Arbiter arbitration method select for shared memory block 3
5–4	SAC2	Shared Bus Arbiter arbitration method select for shared memory block 2
3–2	SAC1	Shared Bus Arbiter arbitration method select for shared memory block 1
1–0	SAC0	Shared Bus Arbiter arbitration method select for shared memory block 0

SACn[1:0] controls the arbitration method for different peripherals:

- 2'bx0 Round-robin method is always used.
- 2'b11 DSP-0 (Core-0) always gets the priority.
- 2'b01 DSP-1 (Core-1) always gets the priority.



## 20.3 Programming Model

The Chip Configuration module provides R/W registers used to configure some features of the chip, such as peripherals pin-switching, Shared Bus arbitration mode selection, and peripherals pin mux control. Some of the peripherals' soft reset can also be triggered by writing to the PSRC register.

This section describes the pin-switching and muxing functions for ESAIs, SHIs, S/PDIF, and TECs.

### 20.3.1 ESAI/ESAI\_1/ESAI\_2/ESAI\_3 Pin-Switching and Internal Connections

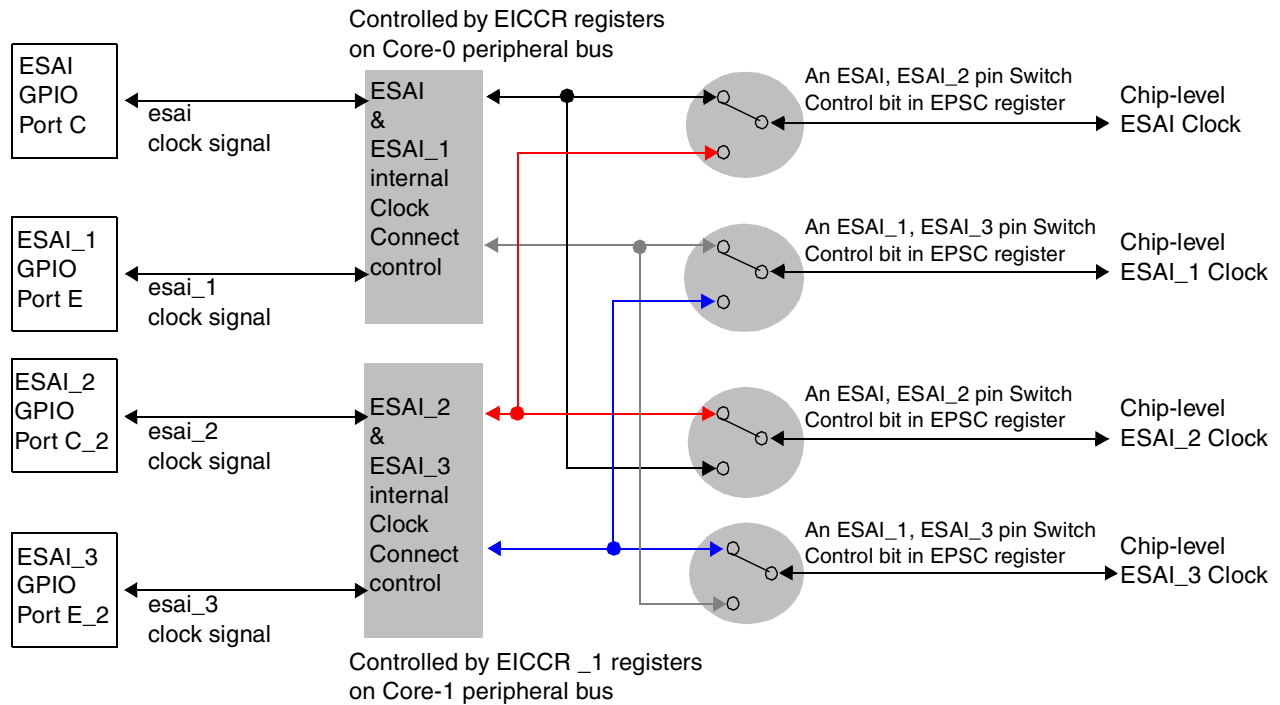
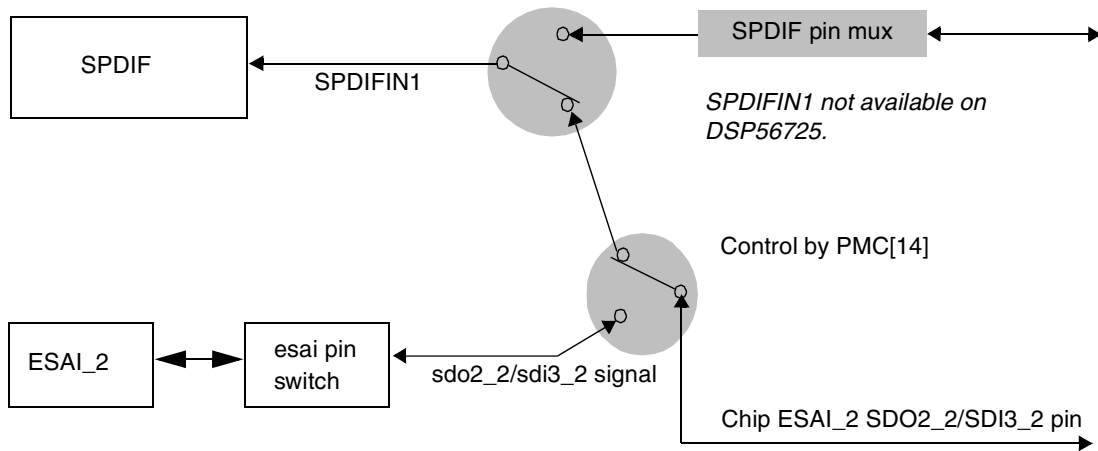


Figure 20-11. ESAI Internal Clock Connections and Pin-Switching

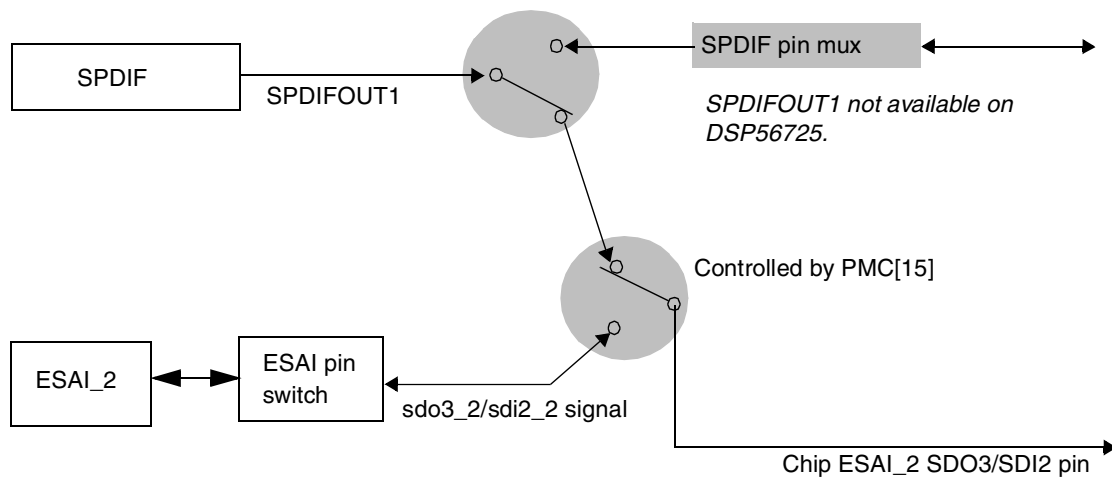
### 20.3.2 ESAI\_2 Data and SPDIF Data Pin Mux

In the DSP56725, the SPDIFIN1 input is multiplexed with ESAI\_2's SDO2\_SDI3 pin, and SPDIFOUT1 is multiplexed with ESAI\_2's SDO3\_SDI2 pin. [Figure 20-12](#) and [Figure 20-13](#) show the connection.



**Figure 20-12. SPDIFIN1 and ESAI Pin Mux**

In the DSP56725, the PMC[14] bit enables/disables the pin mux of the SPDIFIN1 and ESAI\_2 SDO2\_2/SDI3\_2 pins. When the PMC[14] bit is set (1), the shared ESAI\_2 SDO2\_2/SDI3\_2 pin is used as the SPDIFIN1 input and the ESAI\_2 output on this pin is disabled. When the PMC[14] bit is cleared (0), this pin is controlled by ESAI\_2.



**Figure 20-13. SPDIFOUT1 and ESAI\_2 Pin Mux**

In the DSP56725, the PMC[15] bit enables/disables the pin mux of the SPDIFOUT1 and ESAI\_2 SDO3\_2/SDI2\_2 pins. When the PMC[15] bit is set (1), this shared ESAI\_2 pin is used as the SPDIFOUT1 output and the ESAI\_2 input on this pin is disabled. When the PMC[15] bit is cleared, this pin is controlled by ESAI\_2.

### 20.3.3 Soft Reset

The chip configuration module contains a soft reset control register used for triggering soft reset to different shared peripherals. The bit definitions of the PSRC register describes the functionality of each bit that will trigger a soft reset to its corresponding peripheral. Writing “1” to a specific bit of the PSRC register will cause a soft reset of the corresponding peripheral. The PSRC register bit will be cleared automatically by hardware after the reset period reaches the predefined number of reset cycles, which is six system clock cycles.

### 20.3.4 Reset

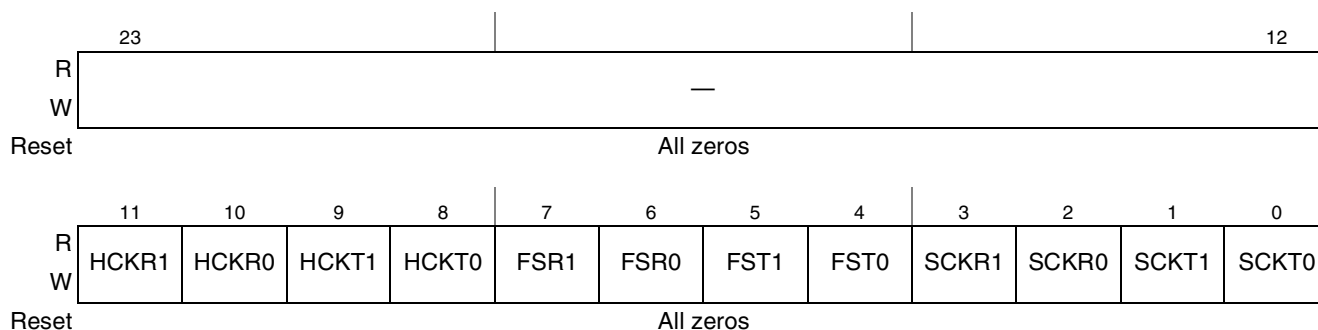
Hardware reset can put all the registers to a known state. All registers are reset to their default value asynchronously.

### 20.3.5 ESAI Pin Switch and Internal Clock Connections

In the DSP56724/DSP56725 devices, new features for ESAI clock and data input/output have been added. The ESAI internal clock connection control bits are implemented in an ESAI Internal Clock Control Register, shown in [Figure 20-14](#). Both DSP cores have their own ESAI Internal Clock Control Register (EICCR and EICCR\_1, respectively).

Address Y: FFFFCA

Access: Read/Write



**Figure 20-14. ESAI Internal Clock Control Register (EICCR for Core-0, EICCR\_1 for Core-1)**

[Table 20-14](#) shows the EICCR register fields.

**Table 20-14. EICCR Field Descriptions**

Bit	Name	Description
23–12	—	Reserved
11–10	HCKR[1:0]	ESAI HCKR clock internal connect control Controls the HCKR clock direction between ESAI and ESAI_1. For Core-1, it controls the HCKR clock direction between ESAI_2 and ESAI_3. Core-0 and Core -1 have these bits respectively.
9–8	HCKT[1:0]	ESAI HCKT clock internal connect control Controls the HCKT clock direction between ESAI and ESAI_1. For Core-1, it controls the HCKT clock direction between ESAI_2 and ESAI_3. Core-0 and Core-1 have these bits respectively.

**Table 20-14. EICCR Field Descriptions (Continued)**

Bit	Name	Description
7–6	FSR[1:0]	ESAI FSR clock internal connect control Controls the FSR clock direction between ESAI and ESAI_1. For Core-1, it controls the FSR clock direction between ESAI_2 and ESAI_3. Core-0 and Core-1 have these bits respectively.
5–4	FST[1:0]	ESAI FST clock internal connect control Controls the FST clock direction between ESAI and ESAI_1. For Core-1, it controls the FST clock direction between ESAI_2 and ESAI_3. Core-0 and Core-1 have these bits respectively.
3–2	SCKR[1:0]	ESAI SCKR clock internal connect control Controls the SCKR clock direction between ESAI and ESAI_1. For Core -1, it controls the SCKR clock direction between ESAI_2 and ESAI_3. Core-0 and Core-1 have these bits respectively.
1–0	SCKT[1:0]	ESAI SCKT clock internal connect control Controls the SCKT clock direction between ESAI and ESAI_1. For Core-1, it controls the SCKT clock direction between ESAI_2 and ESAI_3. Core-0 and Core-1 have these bits respectively.
<p><i>Functional Description of the two control bits for ESAI and ESAI_1 Clocks:</i></p> <p>00 Default value, internal clock connection is disabled.</p> <p>01 Internal clock connection is disabled.</p> <p>10 ESAI clock is connected with ESAI_1; ESAI controls (in/out) the clock pin; ESAI_1's corresponding clock pins should be set as input pins.</p> <p>11 ESAI clock is connected with ESAI_1; ESAI_1 controls (in/out) the clock pin; ESAI's corresponding clock pins should be set as input pins.</p> <p>Note: In all available packages (DSP56725 80-pin, DSP56724 144-pin), ESAI_1's Clock pins will not be available by default.</p>		
<p><i>Functional Description of the two control bits for ESAI_2 and ESAI_3 Clocks:</i></p> <p>00 Default value, internal clock connection is disabled.</p> <p>01 Internal clock connection is disabled.</p> <p>10 ESAI_2 clock is connected with ESAI_3; ESAI_2 controls (in/out) the clock pins; ESAI_3's corresponding clock pins should be set as input pins.</p> <p>11 ESAI_2 clock is connected with ESAI_3; ESAI_3 controls (in/out) the clock pin; ESAI_2's corresponding clock pins should be set as input pins.</p> <p>Note: In all available packages (DSP56725 80-pin, DSP56724 144-pin), ESAI_2's Clock pins will not be available by default.</p>		

### 20.3.5.1 ESAI and ESAI\_1 Internal Clocks Connections

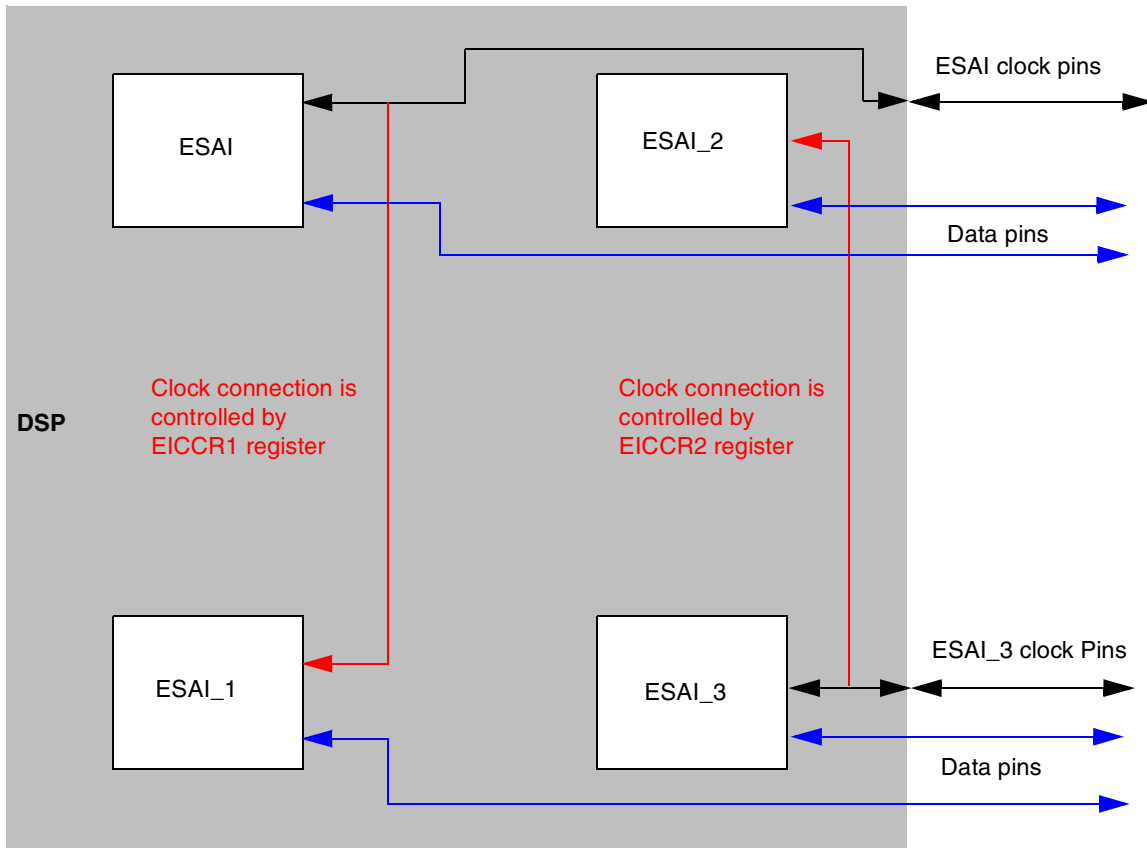
In DSP56725 80-pin and DSP56724 144-pin packages, the following pins are available for transferring data:

- 6 ESAI clocks and 4 data pins
- 6 ESAI\_3 clocks and 4 data pins (DSP56725 also does not have HCKR\_3)
- 4 data pins of ESAI\_1 (SDI0\_1/SDO5\_1, SDI1\_1/SDO4\_1, SDI2\_1/SDO3\_1, SDI3\_1/SDO2\_1)
- 4 data pins of ESAI\_2 (SDI0\_2/SDO5\_2, SDI1\_2/SDO4\_2, SDI2\_2/SDO3\_2, SDI3\_2/SDO2\_2)

The ESAI\_1's clock pins and ESAI\_2's clock signals can be internally connected as shown in [Figure 20-15](#). In DSP56725, the HCKR\_1, HCKT\_1, HCKR\_2, and HCKT\_2 pins are not available; but these signals can be internally connected to the HCKR, HCKT, HCKR\_3 and HCKT\_3 pins. In DSP56725 80-pin package and DSP56724 144-pin packages, 6 ESAI clocks and 4 data pins, 6 ESAI\_3 clocks and 4 data pins, and only 4 data pins of ESAI\_1 (SDI0\_1/SDO5\_1, SDI1\_1/SDO4\_1, SDI2\_1/SDO3\_1, SDI3\_1/SDO2\_1), 4 data pins of ESAI\_2 (SDI0\_2/SDO5\_2, SDI1\_2/SDO4\_2, SDI2\_2/SDO3\_2, SDI3\_2/SDO2\_2) are available for data

transferring. The ESAI\_1's clock pins and ESAI\_2's clock signals can be internally connected as shown in [Figure 20-15](#).

When using ESAI's internal clock connection functions, do not enable the ESAI pin switch function.



**Figure 20-15. ESAI Pin-Out and Internal Clock Connection**

Each clock internal connection is controlled by the EICCR register, and DSP Core-0 and Core-1 both have an EICCR register. DSP Core-0's EICCR register controls the ESAI and ESAI\_1 clocks internal connection. DSP Core-1's EICCR register controls the ESAI\_2 and ESAI\_3 clocks internal connection. There are two bits for each clock internal connection control, described in [Table 20-15](#) and [Table 20-16](#).

**Table 20-15. Internal Clock Connections Between ESAI and ESAI\_1**

EICCR: <i>CLOCK_NAME</i> [1:0]		Description	See
0	0	This is the default value after chip reset; there is no internal clock connection between ESAI and ESAI_1.	Figure 20-16
0	1	No internal clock connection is between ESAI and ESAI_1.	

**Table 20-15. Internal Clock Connections Between ESAI and ESAI\_1 (Continued)**

EICCR: CLOCK_NAME[1:0]		Description	See
1	0	ESAI_1 clock is connected with ESAI; ESAI controls (in/out) the clock pin; the clock signal is also input to ESAI_1 block; ESAI_1's corresponding clock pins should be set as input.	<a href="#">Figure 20-17</a>
1	1	ESAI clock is connected with ESAI_1; ESAI_1 controls (in/out) the clock pin; the clock signal is also input to ESAI block; ESAI's corresponding clock pins should be set as input.	<a href="#">Figure 20-18</a>

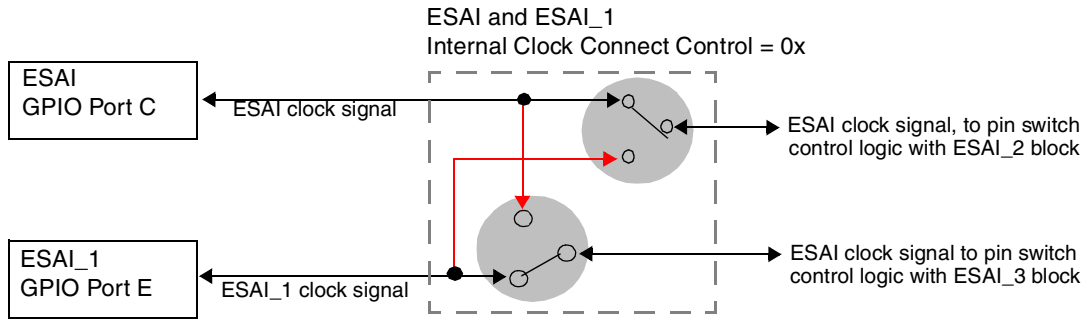
**Table 20-16. Internal Clock Connections Between ESAI\_2 and ESAI\_3**

EICCR: CLOCK_NAME[1:0]		Description	See
0	0	This is the default value after chip reset; there is no internal clock connection between ESAI_3 and ESAI_2.	<a href="#">Figure 20-19</a>
0	1	No internal clock connection is between ESAI_3 and ESAI_2.	
1	0	ESAI_3 clock is connected with ESAI_2; ESAI_2 controls (in/out) the clock pin; the clock signal is also input to ESAI_3 block; ESAI_3's corresponding clock pins should be set as input.	<a href="#">Figure 20-20</a>
1	1	ESAI_2 clock is connected with ESAI_3; ESAI_3 controls (in/out) the clock pin; the clock signal is also input to ESAI_2 block; ESAI_2's corresponding clock pins should be set as input.	<a href="#">Figure 20-21</a>

The figures listed in [Table 20-17](#) show the internal clock connection between ESAI and ESAI\_1 under different clock connection control bits settings. These figures are also applicable to each of the 6 ESAI/ESAI\_1 clocks.

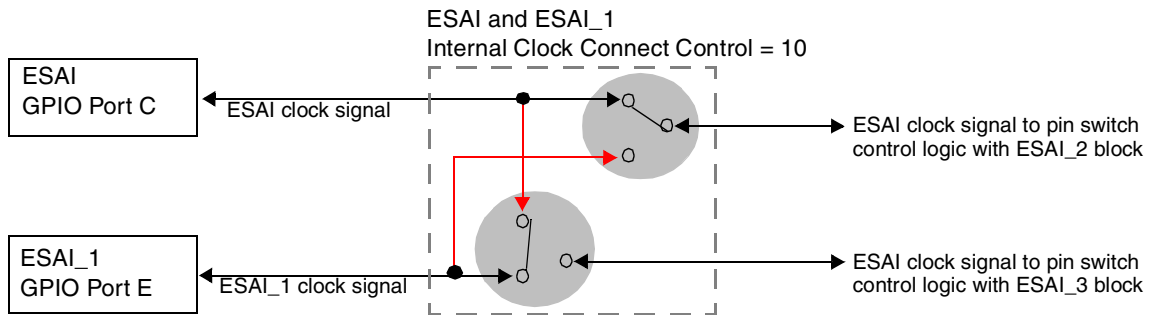
**Table 20-17. ESAI and ESAI\_1 Internal Clock Connections**

Internal Clock Connection	See
No Internal Clock Is Connected Between ESAI and ESAI_1: 0x	<a href="#">Figure 20-16</a>
An Internal Clock Is Connected Between ESAI and ESAI_1: 10	<a href="#">Figure 20-17</a>
An Internal Clock Is Connected Between ESAI and ESAI_1: 11	<a href="#">Figure 20-18</a>



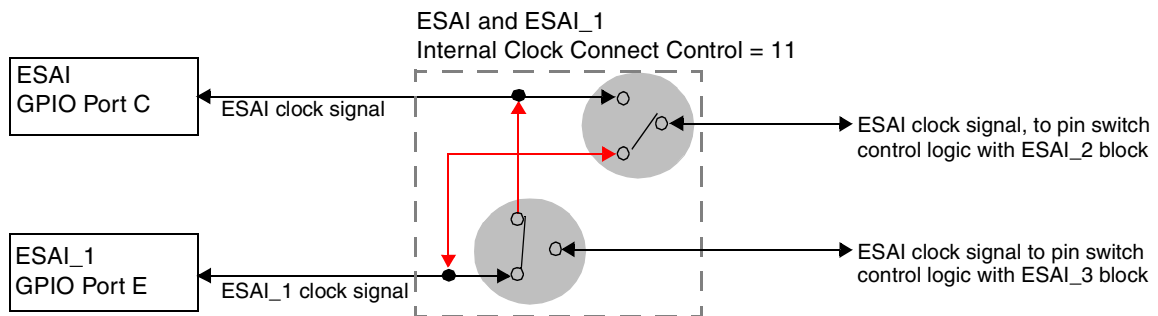
This is the default internal clock connection for ESAI and ESAI\_1. Under this setting, ESAI and ESAI\_1 clock signals are connected to their own clock pins before switching.

**Figure 20-16. No Internal Clock Is Connected Between ESAI and ESAI\_1: 0x**



When the clock connect control bits are set to “10”, the ESAI\_1 clock should be set as input. ESAI\_1’s clock comes from the ESAI clock pin, regardless of whether the ESAI clock pin is set to input or output.

**Figure 20-17. An Internal Clock Is Connected Between ESAI and ESAI\_1: 10**



When the clock connect control bits are set to “11”, ESAI\_1 controls the ESAI clock pin, to input or output the ESAI\_1’s clock signal; the ESAI clock should be set as input, because signals come from this pin.

When these clock pins are used as GPIO ports, the clock connection control bits should be set as 0.

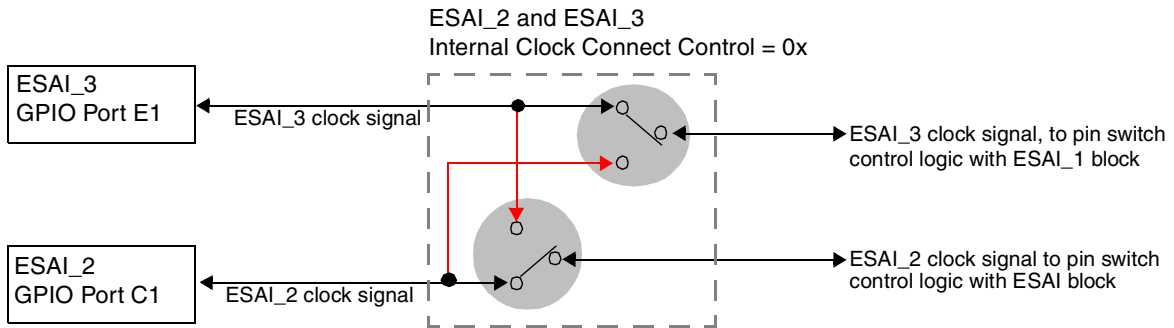
**Figure 20-18. An Internal Clock Is Connected Between ESAI and ESAI\_1: 11**

### 20.3.5.2 ESAI\_2 and ESAI\_3 Internal Clock Connections

The figures listed in [Table 20-18](#) show the internal clock connections between ESAI\_2 and ESAI\_3 under different clock connection control bits settings. These figures are also applicable to each of the 6 ESAI\_2/ESAI\_3 clocks.

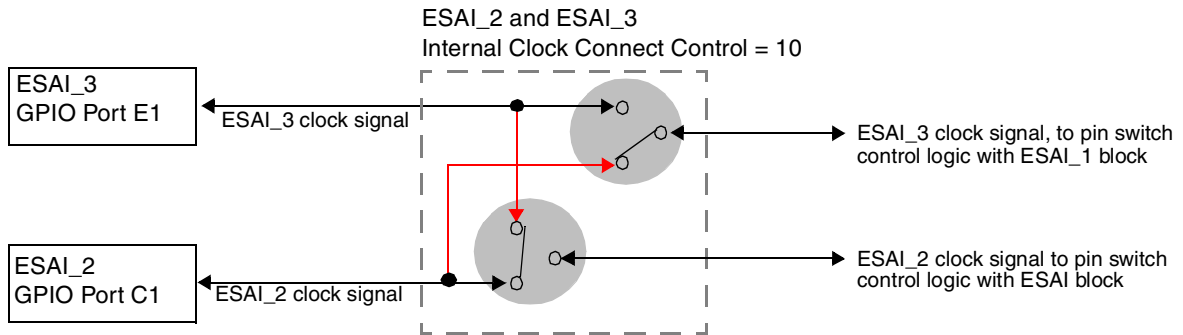
**Table 20-18. ESAI\_2 and ESAI\_3 Internal Clock Connections**

Internal Clock Connection	See
ESAI_2/ESAI_3 Internal Clock Connection Is Disabled: 0x	<a href="#">Figure 20-19</a>
An Internal Clock Is Connected Between ESAI_2 and ESAI_3: 10	<a href="#">Figure 20-20</a>
An Internal Clock Is Connected Between ESAI_2 and ESAI_3: 11	<a href="#">Figure 20-21</a>



As shown here, the internal clock connection is disabled. ESAI\_2 and ESAI\_3 control their own clock signals before switching.

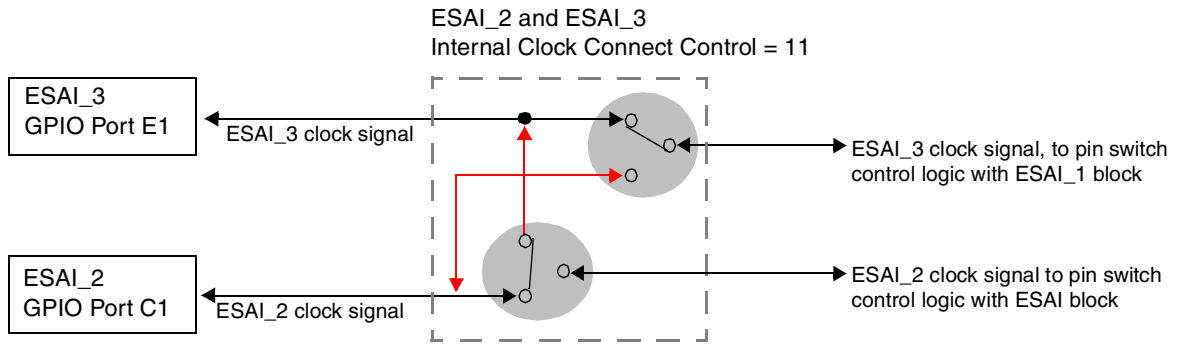
**Figure 20-19. ESAI\_2/ESAI\_3 Internal Clock Connection Is Disabled: 0x**



When the clock connect control bits are set to “10”, ESAI\_3 clock should be set as input. ESAI\_3’s clock comes from the ESAI\_3 clock pin, but input or output state of this pin is controlled by ESAI\_2.

**Figure 20-20. An Internal Clock Is Connected Between ESAI\_2 and ESAI\_3: 10**





When the clock connect control bits are set to “11”, ESAI\_3 controls the ESAI\_3 clock pin, to input or output the ESAI\_3’s clock signal. ESAI\_2 clock should be set as input to get the clock that comes from either pin or from the ESAI\_3 block.

When these clock pins are used as GPIO ports, the clock connection control bits should be set as 0.

**Figure 20-21. An Internal Clock Is Connected Between ESAI\_2 and ESAI\_3: 11**



# Chapter 21

## External Memory Controller (EMC)

### 21.1 Introduction

#### NOTE

The EMC is not available on the DSP56725 device.

The External Memory Controller (EMC) provides a seamless interface to many types of memory devices and peripherals over a shared address and data bus and dedicated control signals. The memory controller in the EMC is responsible for controlling a parameterized number of memory banks shared by a high performance SDRAM machine, a general-purpose chip-select machine (GPCM), and up to three user-programmable machines (UPMs).

As such, the EMC supports a glueless interface to synchronous DRAM (SDRAM), SRAM, EPROM, flash EPROM, burstable RAM, regular DRAM devices, extended data output DRAM devices, and other peripherals. Support signals for external address latch (LALE) allows the multiplexing of address with data lines in devices with limited pin counts.

The EMC includes write protection features, and also a bus monitor to ensure that each internal transaction is terminated within a user-specified period.

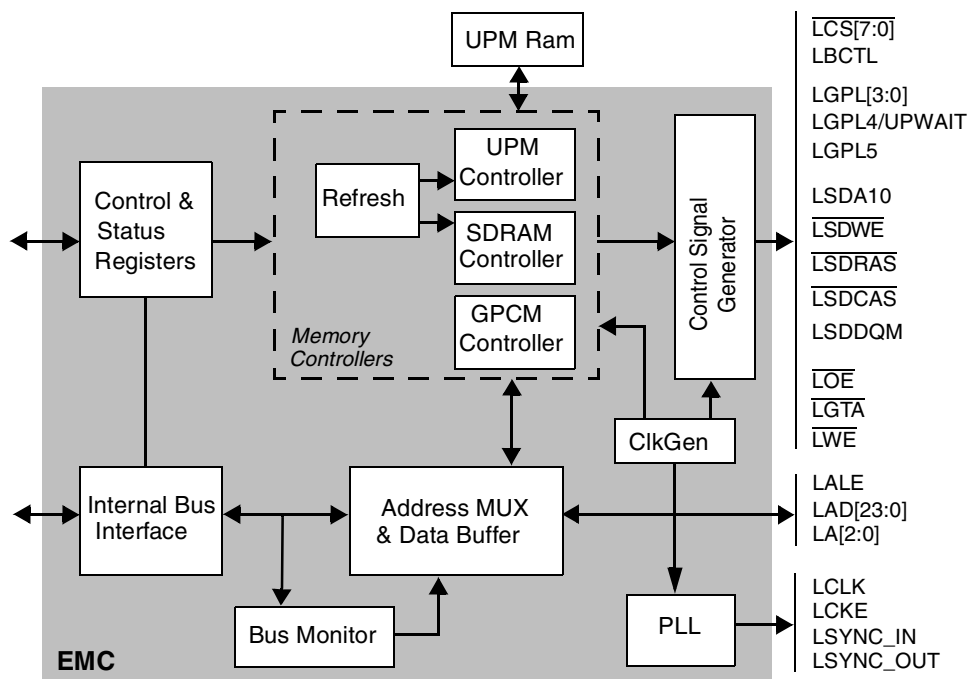


Figure 21-1. EMC Block Diagram

## 21.1.1 Features

Major features of the External Memory Controller (EMC) include the following:

- Generic features
  - Eight memory chip select pins (supports eight memory banks)
  - Accessible from the entire X, Y and P address space
  - Variable memory block sizes (8 K words to 16 M words)
  - Address and data share the LAD[23:0]
  - Selection of the memory controller among SDRAM, GPCM and UPM is on per-bank basis
  - Data buffer controls are activated on a per-bank basis
  - Write protection error detection
  - Chip select error detection
  - Bus timeout error detection
- Synchronous DRAM (SDRAM) machine
  - Provides the control functions and signals to any JEDEC-compliant SDRAM device
  - Up to 100 SDRAM clock
  - Four caches for active pages
  - Supports 4-way bank interleaving
  - Supports 24-bit SDRAM port size
- General-purpose chip-select machine (GPCM)
  - Compatible with SRAM, EPROM, FEPRM, and other peripherals
  - Supports boot from external memory connected at LCS0 (at system reset)
  - Supports boot from 8-bit or 24-bit devices
  - External access termination signal (GTA)
- Three User-Programmable Machines (UPMs)
  - User-specified control-signal patterns can be initiated by software
  - User-specified control-signal patterns can be run when an internal master requests a single-beat or burst read or write access.
  - UPM refresh timer runs a user-specified control signal pattern to support refresh
  - Each UPM can be defined to support DRAM devices with depths of 64 Kw, 128 Kw, 256 Kw, 512 Kw, 1 Mw, 2 Mw, 4 Mw, 8 Mw and 16 Mw.
  - Supports 24-bit memory port size
  - External UPM handshake input—asynchronous freeze/restart, or synchronous transfer acknowledge
- De-skew Phase-locked Loop (PLL)
  - Uses de-skew PLL to automatically compensate for PCB wire delay

- User-programmable PLL bypass enable
- User-programmable PLL power down

## 21.2 External Signal Descriptions

Table 21-1. External Signal Descriptions

Signal	Description	I/O	Reset
LALE	External address latch enable	O	Low
$\overline{\text{LCS}}[7:0]$	Chip selects	O	All High
$\overline{\text{LWE}}/$ LSDDQM	GPCM mode: write enable SDRAM mode: data mask	O O	All High
LSDA10/ LGPL0	SDRAM mode: row address bit/command bit UPM mode: general purpose line 0	O O	High
$\overline{\text{LSDWE}}/$ LGPL1	SDRAM mode: write enable UPM mode: general purpose line 1	O O	High
$\overline{\text{LOE}}/$ $\overline{\text{LSDRAS}}/$ LGPL2	GPCM mode: output enable SDRAM mode: row address strobe UPM mode: general purpose line 2	O O O	High
$\overline{\text{LSDCAS}}/$ LGPL3	SDRAM mode: column address strobe UPM mode: general purpose line 3	O O	High
$\overline{\text{LGTA}}/$ LGPL4/ UPWAIT/	GPCM mode: transaction termination UPM mode: general purpose line 4 UPM mode: external device wait	I O I	High-Z
LGPL5	UPM mode: general purpose line 5	O	High
LBCTL	Data buffer control	O	High
LA[2:0]	External memory non-multiplexed address LSBs	O	All Low
LAD[23:0]	Multiplexed address and data bus	I/O	High-Z
LCKE	External memory clock enable	O	High
LCLK	External memory clocks	O	Driven
LSYNC_IN	PLL synchronize input	I	High-Z
LSYNC_OUT	PLL synchronize output	O	Driven

## 21.2.1 Detailed Signal Descriptions

Table 21-2. Detailed Signal Descriptions

Signal	I/O	Description
LALE	O	External address latch enable. The EMC provides control for an external address latch, which allows address and data to be multiplexed on the device pins.
		<b>State Meaning</b> <i>Asserted</i> —LALE is asserted for at least 1/2 bus clock cycle for each memory-controller transaction. If ORx[EAD] = 1, LALE is asserted for (N + 1/2) bus clock cycles, where N is the number of bus clock cycles defined by the EADC field in the CRR. Note that no other control signals are asserted during the assertion of LALE. <i>Negated</i> —LALE is negated at the negative edge of bus clock during address phase. LALE is negated 1/2 bus clock cycle earlier than the next positive edge of bus clock, to obtain hold time for external latch devices.
		<b>Timing</b> N/A
$\overline{\text{LCS}}[7:0]$	O	Chip selects. Eight mutually-exclusive chip selects are provided.
		<b>State Meaning</b> <i>Asserted/Negated</i> —Used to enable specific memory devices or peripherals connected to the EMC. $\overline{\text{LCS}}[7:0]$ are provided on a per-bank basis. For example, $\overline{\text{LCS}}0$ is the chip select for memory bank 0, which has the memory type and attributes defined by BR0 and OR0.
		<b>Timing</b> N/A
$\overline{\text{LWE}}$ / LSDDQM/	O	GPCM write enable/SDRAM data mask
		<b>State Meaning</b> <i>Asserted/Negated</i> —For GPCM operation, $\overline{\text{LWE}}$ is asserted for writing. For SDRAM operation, LSDDQM functions as the DQM or data mask signals provided by JEDEC-compliant SDRAM devices. LSDDQM is driven high when the EMC wishes to mask a write or disable read data output from the SDRAM.
		<b>Timing</b> N/A
LSDA10/ LGPL0	O	SDRAM A10/General purpose line 0
		<b>State Meaning</b> <i>Asserted/Negated</i> —For SDRAM accesses, this signal represents address bit 10. When the row address is driven, this signal drives the value of address bit 10. When the column address is driven, this signal forms part of the SDRAM command. This signal is one of six general purpose signals when in UPM mode and drives a value programmed in the UPM array
		<b>Timing</b> N/A

**Table 21-2. Detailed Signal Descriptions (Continued)**

Signal	I/O	Description	
$\overline{\text{LSDWE}}$ / LGPL1	O	SDRAM write enable/General-purpose line 1	
		<b>State Meaning</b>	<i>Asserted/Negated</i> —This signal is connected to the SDRAM device WE input and acts as the SDRAM write enable when accessing SDRAM. This signal is one of six general purpose signals when in UPM mode, and drives a value programmed in the UPM array
		<b>Timing</b>	N/A
$\overline{\text{LOE}}$ / $\overline{\text{LSDRAS}}$ / LGPL2	O	GPCM output enable/SDRAM RAS/General-purpose line 2	
		<b>State Meaning</b>	<i>Asserted/Negated</i> —This signal controls the output buffer of memory when accessing memory/devices in GPCM mode. For SDRAM accesses, this signal is the row address strobe (RAS). This signal is one of six general purpose lines when in UPM mode, and drives a value programmed in the UPM array.
		<b>Timing</b>	N/A
$\overline{\text{LSDCAS}}$ / LGPL3	O	SDRAM CAS/General-purpose line 3	
		<b>State Meaning</b>	<i>Asserted/Negated</i> —In SDRAM mode, this signal drives the column address strobe (CAS). This signal is one of six general purpose signals when in UPM mode, and drives a value programmed in the UPM array.
		<b>Timing</b>	N/A
$\overline{\text{LGTA}}$ / LGPL4/ UPWAIT	I/O	GPCM terminate access/General-purpose line 4/UPM wait	
		<b>State Meaning</b>	<i>Asserted/Negated</i> —This signal is an input in GPCM mode and is used for transaction termination. This signal may also be configured as one of six general purpose output signals when in UPM mode, or as an input to force the UPM controller to wait for the memory/device.
		<b>Timing</b>	N/A
LGPL5	O	General-purpose line 5	
		<b>State Meaning</b>	<i>Asserted/Negated</i> —This signal is one of six general purpose signals when in UPM mode, and drives a value programmed in the UPM array.
		<b>Timing</b>	N/A
LBCTL	O	Data buffer control. The memory controller activates a data buffer control signal (BCTL) for the external memory when a GPCM- or UPM-controlled bank is accessed. Access to an SDRAM machine-controlled bank does not activate the buffer control. The buffer control can be disabled by setting ORx[BCTLD].	
		<b>State Meaning</b>	<i>Asserted/Negated</i> —The LBCTL pin normally functions as a write/read control for a bus transceiver connected to the LAD lines. Note that an external data buffer must not drive the LAD lines in conflict with the EMC when LBCTL is high, because LBCTL remains high after reset and during address phases.
		<b>Timing</b>	N/A

**Table 21-2. Detailed Signal Descriptions (Continued)**

Signal	I/O	Description	
LA[2:0]	O	External memory non-multiplexed address LSBs	
		<b>State Meaning</b>	<i>Asserted/Negated</i> —Even though the EMC shares an address and data bus, up to three least significant bits of the RAM address always appear on the dedicated address pins, LA[2:0]. These bits may be used, unlatched, in place of LAD[2:0] to connect the three least significant bits of the address for address phases. For some RAM devices, such as fast-page DRAM, LA[2:0] serves as the column address offset during a burst access.
		<b>Timing</b>	N/A
LAD[23:0]	I/O	Multiplexed address and data bus	
		<b>State Meaning</b>	<i>Asserted/Negated</i> —LAD[23:0] is the shared 24-bit address and data bus through which external RAM devices transfer data and receive addresses.
		<b>Timing</b>	<i>Assertion/Negation</i> —During assertion of LALE, LAD[23:0] are driven with the RAM address for the access to follow. External logic should propagate the address on LAD[23:0] while LALE is asserted, and latch the address upon negation of LALE. After LALE is negated, LAD[23:0] are either driven by write data or are made high-impedance by the EMC to sample read data driven by an external device. Following the last data transfer of a write access, LAD[23:0] are again taken into a high-impedance state.
LCKE	O	External memory clock enable	
		<b>State Meaning</b>	<i>Asserted/Negated</i> —LCKE is the bus clock enable signal (CKE) for JEDEC-standard SDRAM devices. This signal is asserted during normal SDRAM operation.
LCLK	O	External memory clocks	
		<b>State Meaning</b>	<i>Asserted/Negated</i> —LCLK drive external memory clock signal. If the EMC Phase-Locked Loop (PPLL) is enabled (see CRR[DBYP], <a href="#">Table 21-33</a> ), the bus clock phase is shifted earlier than transitions on other EMC signals (such as LAD[23:0] and LCSx) by a time delay matching the delay of the PPLL timing loop set up between LSYNC_OUT and LSYNC_IN.
LSYNC_OUT	O	PPLL synchronization out	
		<b>State Meaning</b>	<i>Asserted/Negated</i> —A replica of the bus clock (appearing on LSYNC_OUT) that should be propagated through a passive timing loop and returned to LSYNC_IN for achieving correct PPLL lock.
		<b>Timing</b>	<i>Assertion/Negation</i> —The time delay of the timing loop should be such that it compensates for the round-trip flight time of LCLK and clocked drivers in the system. No load other than a timing loop should be placed on LSYNC_OUT.
LSYNC_IN	I	PPLL synchronization in	
		<b>State Meaning</b>	<i>Asserted/Negated</i> —See the description of LSYNC_OUT (previous item in this table).



## 21.3 Memory Map and Register Definition

### 21.3.1 Memory Map

Table 21-3 shows the EMC memory map.

**Table 21-3. EMC Registers Memory Map**

Address Offset	Register		Access	Reset Value
0xFFFE00	BRL0	Base register 0 low part	R/W	0x001801
0xFFFE01	BRH0	Base register 0 high part	R/W	0x000000
0xFFFE02	ORL0	Options register 0 low part	R/W	0x000FF7
0xFFFE03	ORH0	Options register 0 high part	R/W	0x000000
0xFFFE04	BRL1	Base register 1 low part	R/W	0x000000
0xFFFE05	BRH1	Base register 1 high part	R/W	0x000000
0xFFFE06	ORL1	Options register 1 low part	R/W	0x000000
0xFFFE07	ORH1	Options register 1 high part	R/W	0x000000
0xFFFE08	BRL2	Base register 2 low part	R/W	0x000000
0xFFFE09	BRH2	Base register 2 high part	R/W	0x000000
0xFFFE0A	ORL2	Options register 2 low part	R/W	0x000000
0xFFFE0B	ORH2	Options register 2 high part	R/W	0x000000
0xFFFE0C	BRL3	Base register 3 low part	R/W	0x000000
0xFFFE0D	BRH3	Base register 3 high part	R/W	0x000000
0xFFFE0E	ORL3	Options register 3 low part	R/W	0x000000
0xFFFE0F	ORH3	Options register 3 high part	R/W	0x000000
0xFFFE10	BRL4	Base register 4 low part	R/W	0x000000
0xFFFE11	BRH4	Base register 4 high part	R/W	0x000000
0xFFFE12	ORL4	Options register 4 low part	R/W	0x000000
0xFFFE13	ORH4	Options register 4 high part	R/W	0x000000
0xFFFE14	BRL5	Base register 5 low part	R/W	0x000000
0xFFFE15	BRH5	Base register 5 high part	R/W	0x000000
0xFFFE16	ORL5	Options register 5 low part	R/W	0x000000
0xFFFE17	ORH5	Options register 5 high part	R/W	0x000000
0xFFFE18	BRL6	Base register 6 low part	R/W	0x000000
0xFFFE19	BRH6	Base register 6 high part	R/W	0x000000
0xFFFE1A	ORL6	Options register 6 low part	R/W	0x000000
0xFFFE1B	ORH6	Options register 6 high part	R/W	0x000000

**Table 21-3. EMC Registers Memory Map (Continued)**

Address Offset	Register		Access	Reset Value
0xFFFE1C	BRL7	Base register 7 low part	R/W	0x000000
0xFFFE1D	BRH7	Base register 7 high part	R/W	0x000000
0xFFFE1E	ORL7	Options register 7 low part	R/W	0x000000
0xFFFE1F	ORH7	Options register 7 high part	R/W	0x000000
0xFFFE20–0xFFFE33	—	Reserved	—	—
0xFFFE34	MARL	UPM address register low part	R/W	0x000000
0xFFFE35	MARH	UPM address register high part	R/W	0x000000
0xFFFE36–0xFFFE37	—	Reserved	—	—
0xFFFE38	MAMRL	UPMA mode register low part	R/W	0x000000
0xFFFE39	MAMRH	UPMA mode register high part	R/W	0x000000
0xFFFE3A	MBMRL	UPMB mode register low part	R/W	0x000000
0xFFFE3B	MBMRH	UPMB mode register high part	R/W	0x000000
0xFFFE3C	MCMRL	UPMC mode register low part	R/W	0x000000
0xFFFE3D	MCMRH	UPMC mode register high part	R/W	0x000000
0xFFFE3E–0xFFFE41	—	Reserved	—	—
0xFFFE42	—	Reserved. Memory refresh timer prescaler register has no low part.	—	—
0xFFFE43	MRTPR	Memory refresh timer prescaler register	R/W	0x000000
0xFFFE44	MDRL	UPM data register low part	R/W	0x000000
0xFFFE45	MDRH	UPM data register high part	R/W	0x000000
0xFFFE46–0xFFFE49	—	Reserved	—	—
0xFFFE4A	SDMRL	SDRAM mode register low part	R/W	0x000000
0xFFFE4B	SDMRH	SDRAM mode register high part	R/W	0x000000
0xFFFE4C–0xFFFE4F	—	Reserved	—	—
0xFFFE50	—	Reserved. UPM refresh timer register has no low part.	—	—
0xFFFE51	URT	UPM refresh timer register	R/W	0x000000
0xFFFE52	LALE	Reserved. SDRAM refresh timer has no low part	—	—
0xFFFE53	SRT	SDRAM refresh timer	R/W	0x000000
0xFFFE54–0xFFFE47	—	Reserved	—	—
0xFFFE58	—	Reserved. Transfer error status register has no low part.	—	—
0xFFFE59	TESR	Transfer error status register	Read/ Write Clear	0x000000
0xFFFE5A	—	Reserved. Transfer error disable register has no low part.	—	—

**Table 21-3. EMC Registers Memory Map (Continued)**

Address Offset	Register		Access	Reset Value
0xFFFE5B	TEDR	Transfer error disable register	R/W	0x000000
0xFFFE5C	—	Reserved. Transfer error interrupt register has no low part.	—	—
0xFFFE5D	TEIR	Transfer error interrupt register	R/W	0x000000
0xFFFE5E	TEATRL	Transfer error attributes register low part	R/W	0x000000
0xFFFE5F	TEATRH	Transfer error attributes register high part	R/W	0x000000
0xFFFE60	TEARL	Transfer error address register low part	R/W	0x000000
0xFFFE61	TEARH	Transfer error address register high part	R/W	0x000000
0xFFFE62–0xFFFE67	—	Reserved	—	—
0xFFFE68	BCRL	Configuration register low part	R/W	0x000000
0xFFFE69	BCRH	Configuration register high part	R/W	0x000000
0xFFFE6A	CRRL	Clock ratio register low part	R/W	0x000008
0xFFFE6B	CRRH	Clock ratio register high part	R/W	0x008000

### 21.3.2 Register Descriptions

This section provides a detailed description of the EMC configuration, status, and control registers, with detailed bit and field descriptions.

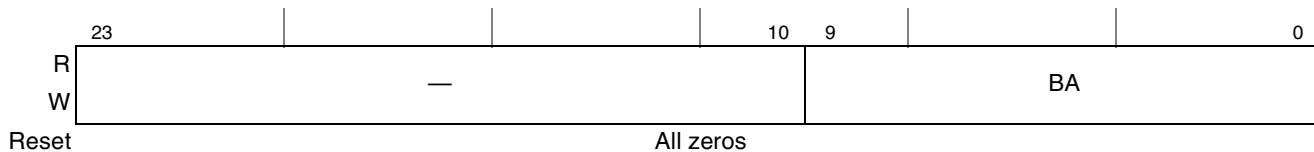
Address offsets in the EMC address range that are not defined in [Table 21-3](#) should not be accessed for reading or writing. Similarly, only zero should be written to reserved bits of defined registers, as writing ones can have unpredictable results in some cases.

#### 21.3.2.1 Base Registers (BR0–BR7)

The base registers (BR<sub>x</sub>) contain the base address and address types for each memory bank, shown in [Figure 21-2](#) and [Figure 21-3](#). The memory controller uses this information to compare the address bus value with the current address accessed. Each BR<sub>x</sub> also includes a memory attribute and selects the machine for memory operation handling.

### External Memory Controller (EMC)

Address BRH0 X: 0xFFFE01 Access: Read/Write  
 BRH1 X: 0xFFFE05  
 BRH2 X: 0xFFFE09  
 BRH3 X: 0xFFFE0D  
 BRH4 X: 0xFFFE11  
 BRH5 X: 0xFFFE15  
 BRH6 X: 0xFFFE19  
 BRH7 X: 0xFFFE1D



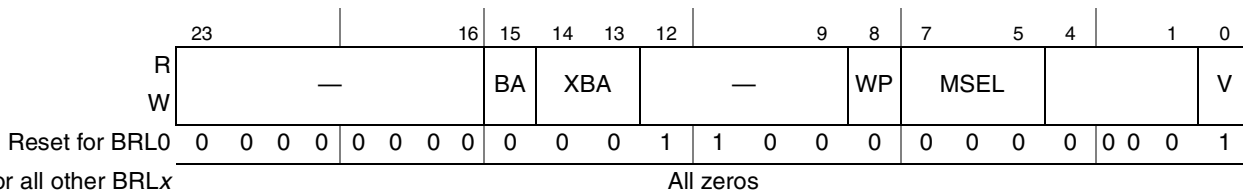
**Figure 21-2. Base Register High Part (x)**

Table 21-4 defines the fields for BRHx.

**Table 21-4. BRHx field Descriptions**

Bits	Name	Description
23–10	—	Reserved
9–0	BA	Base address bits 23–14. The upper 11 bits of each base address register are compared to the address on the address bus to determine if the bus master is accessing a memory bank controlled by the memory controller. This is used with the address mask bits ORx[AM].

Address BRH0 X: 0xFFFE00 Access: Read/Write  
 BRH1 X: 0xFFFE04  
 BRH2 X: 0xFFFE08  
 BRH3 X: 0xFFFE0C  
 BRH4 X: 0xFFFE10  
 BRH5 X: 0xFFFE14  
 BRH6 X: 0xFFFE18  
 BRH7 X: 0xFFFE1C



<sup>1</sup> BRL0 has this value set during reset (GPCM is the default control machine for all banks coming out of reset). All other option registers have all bits cleared.

**Figure 21-3. Base Register Low Part (x)**

Table 21-5 defines the fields for BRL<sub>x</sub>.

**Table 21-5. BRL<sub>x</sub> Field Descriptions**

Bits	Name	Description
23–16	—	Reserved
15	BA	Base address bit 13. The upper 11 bits of each base address register are compared to the address on the address bus to determine if the bus master is accessing a memory bank controlled by the memory controller. This is used with the address mask bits OR <sub>x</sub> [AM].
14–13	XBA	Extended base address. BA are compared with the address bus only for a given type of access: X access, Y access, P access (read/write) or P access (instruction fetch). This is used with the extended address mask bits OR <sub>x</sub> [XAM]. 00 Response X access 01 Response Y access 10 Response P access (read/write) 11 Response P access (instruction fetch)
12–9	—	Reserved
8	WP	Write protect 0 Read and write accesses are allowed. 1 Only read accesses are allowed. The memory controller does not assert $\overline{\text{LCSx}}$ on write cycles to this memory bank. TESR[WP] is set (if enabled) if a write to this memory bank is attempted, and a external memory error interrupt is generated (if enabled), terminating the cycle.
7–5	MSEL	Machine select. Specifies which machine to use for the memory operations handling. Reset value selects GPCM. 000 GPCM 001 Reserved 010 Reserved 011 SDRAM 100 UPMA 101 UPMB 110 UPMC 111 Reserved
4–1	—	Reserved
0	V	Valid bit. Indicates that the contents of the BR <sub>x</sub> and OR <sub>x</sub> pair are valid. The $\overline{\text{LCSx}}$ signal does not assert unless V is set (an access to a region that has no valid bit set may cause a bus time-out). After a system reset, only BR0[V] is set. 0 This bank is invalid. 1 This bank is valid.

### 21.3.2.2 Option Registers (OR0–OR7)

The OR<sub>x</sub> registers define the sizes of the memory banks and access attributes. The OR<sub>x</sub> attribute bits support the following three modes of operation as defined by BR<sub>x</sub>[MSEL]:

- GPCM mode
- UPM mode
- SDRAM mode

The OR<sub>x</sub> registers are interpreted differently depending on which of the three machine types is selected for that bank.

### 21.3.2.2.1 Address Mask

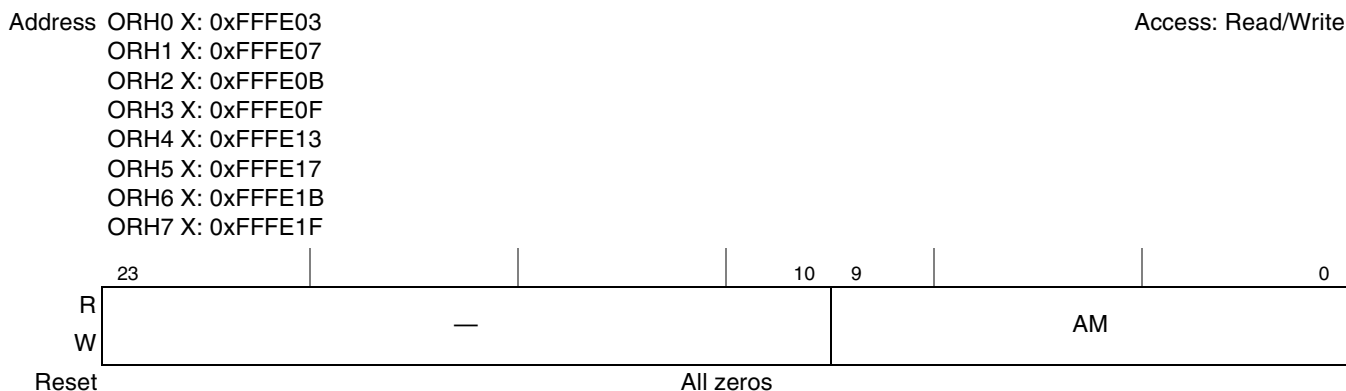
The address mask fields of the option registers (OR<sub>x</sub>[XAM,AM]) mask up to 13 corresponding BR<sub>x</sub>[XBA,BA]. The 13 LSBs of the 24-bit internal address do not participate in bank address matching in selecting a bank for access. Masking address bits independently allows external devices of different size address ranges to be used. Address mask bits can be set or cleared in any order in the field, allowing a resource to reside in more than one area of the address map. Table 21-6 shows memory bank sizes from 8 Kwords up to 16 Mwords.

**Table 21-6. Memory Bank Sizes in Relation to Address Mask**

XAM	Address Mask	Memory Bank Size in Words
11	0000000000	16 Mwords
11	1000000000	8 Mwords
11	1100000000	4 Mwords
11	1110000000	2 Mwords
11	1111000000	1 Mword
11	1111100000	512 Kwords
11	1111110000	256 Kwords
11	1111111000	128 Kwords
11	1111111100	64 Kwords
11	1111111110	32 Kwords
11	11111111110	16 Kwords
11	11111111111	8 Kwords

### 21.3.2.2.2 Option Registers (OR<sub>x</sub>)—GPCM Mode

Figure 21-4 and Figure 21-5 show the bit fields for OR<sub>x</sub> when the corresponding BR<sub>x</sub>[MSEL] selects the GPCM machine.



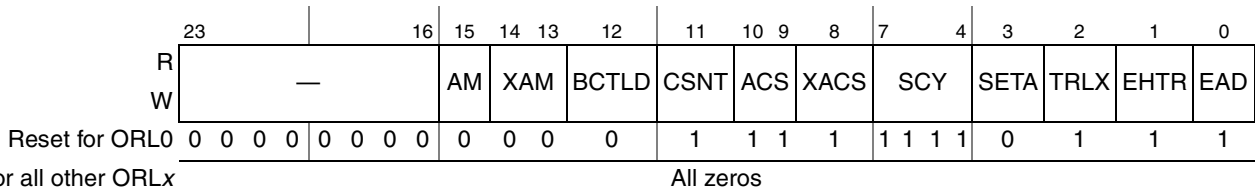
**Figure 21-4. Option Register High Part—GPCM Mode(x)**

Table 21-7 defines the fields for ORH<sub>x</sub>-GPCM.

**Table 21-7. ORH<sub>x</sub>—GPCM Field Descriptions**

Bits	Name	Description
23–10	—	Reserved
9–0	AM	GPCM address mask bits 23–14. Masks correspond to BR <sub>x</sub> bits. Masking address bits independently allows external devices of different size address ranges to be used. Address mask bits can be set or cleared in any order in the field, allowing a resource to reside in more than one area of the address map. 0 Corresponding address bits are masked. 1 Corresponding address bits are used in the comparison between base and transaction addresses.

Address ORL0 X: 0xFFFE02 Access: Read/Write  
 ORL1 X: 0xFFFE06  
 ORL2 X: 0xFFFE0A  
 ORL3 X: 0xFFFE0E  
 ORL4 X: 0xFFFE12  
 ORL5 X: 0xFFFE16  
 ORL6 X: 0xFFFE1A  
 ORL7 X: 0xFFFE1E



<sup>1</sup> ORH0 has this value set during reset (GPCM is the default control machine for all banks coming out of reset). All other option registers have all bits cleared.

**Figure 21-5. Option Register Low Part—GPCM Mode(x)**

Table 21-8 defines the fields for ORL<sub>x</sub>-GPCM.

**Table 21-8. ORL<sub>x</sub>—GPCM Field Descriptions**

Bits	Name	Description
23–16	—	Reserved
15	AM	GPCM address mask bit 13. Masks corresponding BR <sub>x</sub> bits. Masking address bits independently allows external devices of different size address ranges to be used. Address mask bits can be set or cleared in any order in the field, allowing a resource to reside in more than one area of the address map. 0 Corresponding address bits are masked. 1 Corresponding address bits are used in the comparison between base and transaction addresses.
14–13	XAM	Extended address mask. Masks the corresponding XBA bits in the BR <sub>x</sub> register. For example, if BR <sub>x</sub> [XBA] = 00 or 01, and OR <sub>x</sub> [XAM] = 10, then the settings permit access from both X and Y.
12	BCTLD	Buffer control disable. Disables assertion of LBCTL during access to the current memory bank. 0 LBCTL is asserted upon access to the current memory bank. 1 LBCTL is not asserted upon access to the current memory bank.

**Table 21-8. ORLx—GPCM Field Descriptions (Continued)**

Bits	Name	Description																					
11	CSNT	<p>Chip select negation time. Determines when <math>\overline{\text{LCSx}}</math> and <math>\overline{\text{LWE}}</math> are negated during an external memory write access handled by the GPCM, provided that ACS is not set to 00 (when ACS = 00, only WE is affected by the setting of CSNT). This helps meet address and data hold times for slow memories and peripherals.</p> <p>0 <math>\overline{\text{LCSx}}</math> and <math>\overline{\text{LWE}}</math> are negated normally.</p> <p>1 <math>\overline{\text{LCSx}}</math> and <math>\overline{\text{LWE}}</math> are negated earlier depending on the value of CRR[CLKDIV] and external memory personality mode.</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>LCRR[CLKDIV]</th> <th>CSNT</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>x</td> <td>0</td> <td><math>\overline{\text{LCSx}}</math> and <math>\overline{\text{LWE}}</math> are negated normally.</td> </tr> <tr> <td>2</td> <td>1</td> <td><math>\overline{\text{LCSx}}</math> and <math>\overline{\text{LWE}}</math> are negated normally.</td> </tr> <tr> <td>4 or 8</td> <td>1</td> <td><math>\overline{\text{LCSx}}</math> and <math>\overline{\text{LWE}}</math> are negated quarter of a bus clock cycle earlier.</td> </tr> </tbody> </table>	LCRR[CLKDIV]	CSNT	Meaning	x	0	$\overline{\text{LCSx}}$ and $\overline{\text{LWE}}$ are negated normally.	2	1	$\overline{\text{LCSx}}$ and $\overline{\text{LWE}}$ are negated normally.	4 or 8	1	$\overline{\text{LCSx}}$ and $\overline{\text{LWE}}$ are negated quarter of a bus clock cycle earlier.									
LCRR[CLKDIV]	CSNT	Meaning																					
x	0	$\overline{\text{LCSx}}$ and $\overline{\text{LWE}}$ are negated normally.																					
2	1	$\overline{\text{LCSx}}$ and $\overline{\text{LWE}}$ are negated normally.																					
4 or 8	1	$\overline{\text{LCSx}}$ and $\overline{\text{LWE}}$ are negated quarter of a bus clock cycle earlier.																					
10–9	ACS	<p>Address to chip-select setup. Determines the delay of the <math>\overline{\text{LCSx}}</math> assertion relative to the address change when the external memory access is handled by the GPCM. At system reset, OR0[ACS] = 11.</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>LCRR[CLKDIV]</th> <th>ACS</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>x</td> <td>00</td> <td><math>\overline{\text{LCSx}}</math> is output at the same time as the address lines. Note that this overrides the value of CSNT such that CSNT = 0.</td> </tr> <tr> <td>x</td> <td>01</td> <td>Reserved.</td> </tr> <tr> <td>2</td> <td>10</td> <td><math>\overline{\text{LCSx}}</math> is output a half bus clock cycle after the address lines.</td> </tr> <tr> <td>2</td> <td>11</td> <td><math>\overline{\text{LCSx}}</math> is output a half bus clock cycle after the address lines.</td> </tr> <tr> <td>4 or 8</td> <td>10</td> <td><math>\overline{\text{LCSx}}</math> is output a quarter bus clock cycle after the address lines.</td> </tr> <tr> <td>4 or 8</td> <td>11</td> <td><math>\overline{\text{LCSx}}</math> is output a half bus clock cycle after the address lines.</td> </tr> </tbody> </table>	LCRR[CLKDIV]	ACS	Meaning	x	00	$\overline{\text{LCSx}}$ is output at the same time as the address lines. Note that this overrides the value of CSNT such that CSNT = 0.	x	01	Reserved.	2	10	$\overline{\text{LCSx}}$ is output a half bus clock cycle after the address lines.	2	11	$\overline{\text{LCSx}}$ is output a half bus clock cycle after the address lines.	4 or 8	10	$\overline{\text{LCSx}}$ is output a quarter bus clock cycle after the address lines.	4 or 8	11	$\overline{\text{LCSx}}$ is output a half bus clock cycle after the address lines.
LCRR[CLKDIV]	ACS	Meaning																					
x	00	$\overline{\text{LCSx}}$ is output at the same time as the address lines. Note that this overrides the value of CSNT such that CSNT = 0.																					
x	01	Reserved.																					
2	10	$\overline{\text{LCSx}}$ is output a half bus clock cycle after the address lines.																					
2	11	$\overline{\text{LCSx}}$ is output a half bus clock cycle after the address lines.																					
4 or 8	10	$\overline{\text{LCSx}}$ is output a quarter bus clock cycle after the address lines.																					
4 or 8	11	$\overline{\text{LCSx}}$ is output a half bus clock cycle after the address lines.																					
8	XACS	<p>Extra address to chip-select setup. Setting this bit increases the delay of the <math>\overline{\text{LCSx}}</math> assertion relative to the address change when the external memory access is handled by the GPCM. After a system reset, OR0[XACS] = 0.</p> <p>0 Address to chip-select setup is determined by ORx[ACS] and CRR[CLKDIV].</p> <p>1 Address to chip-select setup is extended. (See <a href="#">Table 21-36</a> and <a href="#">Table 21-36</a> for CRR[CLKDIV] = 4 or 8, <a href="#">Table 21-37</a> and <a href="#">Table 21-38</a> for CRR[CLKDIV] = 2)</p>																					
7–4	SCY	<p>Cycle length in bus clocks. Determines the number of wait states inserted in the bus cycle, when the GPCM handles the external memory access. Therefore, it is the main parameter for determining cycle length. The total cycle length depends on other timing attribute settings. After a system reset, OR0[SCY] = 1111.</p> <p>0000 No wait states</p> <p>0001 1-bus clock cycle wait state</p> <p>...</p> <p>1111 15-bus clock cycle wait states</p>																					

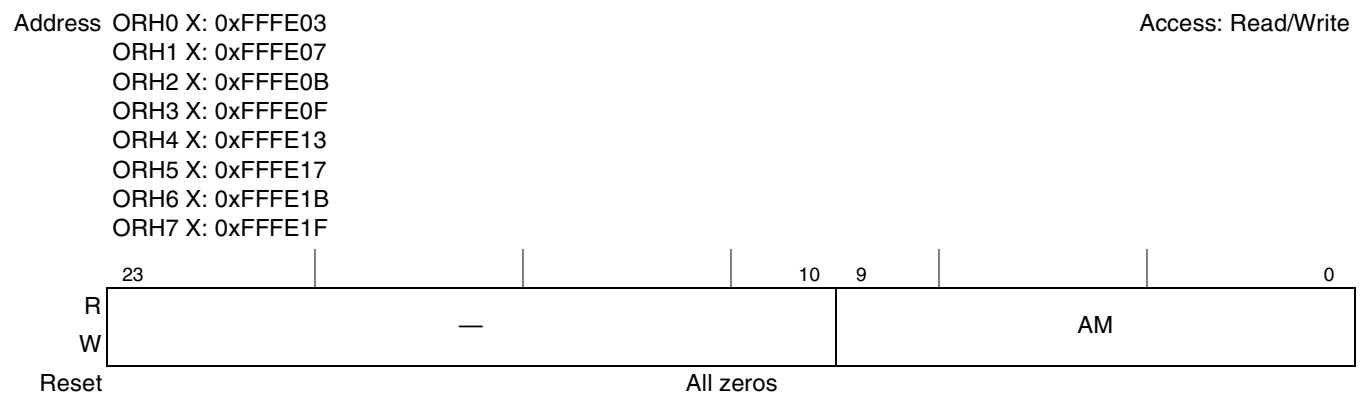


**Table 21-8. ORLx—GPCM Field Descriptions (Continued)**

Bits	Name	Description															
3	SETA	External address termination. 0 Access is terminated internally by the memory controller unless the external device asserts $\overline{LGTA}$ earlier to terminate the access. 1 Access is terminated externally by asserting the $\overline{LGTA}$ external pin. (Only $\overline{LGTA}$ can terminate the access).															
2	TRLX	Timing relaxed. Modifies the settings of timing parameters for slow memories or peripherals. 0 Normal timing is generated by the GPCM. 1 Relaxed timing on the following parameters: <ul style="list-style-type: none"> <li>• Adds an additional cycle between the address and control signals (only if ACS is not equal to 00).</li> <li>• Doubles the number of wait states specified by SCY, providing up to 30 wait states.</li> <li>• Works in conjunction with EHTR to extend hold time on read accesses.</li> <li>• <math>\overline{LCSx}</math> (only if ACS is not equal to 00) and <math>\overline{LWE}</math> signals are negated one cycle earlier during writes.</li> </ul>															
1	EHTR	Extended hold time on read accesses. Indicates with TRLX how many cycles are inserted between a read access from the current bank and the next access. <table border="1" style="margin: 10px auto;"> <thead> <tr> <th>TRLX</th> <th>EHTR</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>The memory controller generates normal timing. No additional cycles are inserted.</td> </tr> <tr> <td>0</td> <td>1</td> <td>1 idle clock cycle is inserted.</td> </tr> <tr> <td>1</td> <td>0</td> <td>4 idle clock cycles are inserted.</td> </tr> <tr> <td>1</td> <td>1</td> <td>8 idle clock cycles are inserted.</td> </tr> </tbody> </table>	TRLX	EHTR	Meaning	0	0	The memory controller generates normal timing. No additional cycles are inserted.	0	1	1 idle clock cycle is inserted.	1	0	4 idle clock cycles are inserted.	1	1	8 idle clock cycles are inserted.
TRLX	EHTR	Meaning															
0	0	The memory controller generates normal timing. No additional cycles are inserted.															
0	1	1 idle clock cycle is inserted.															
1	0	4 idle clock cycles are inserted.															
1	1	8 idle clock cycles are inserted.															
0	EAD	External address latch delay. Allows extra bus clock cycles when using external address latch (LALE). 0 No additional bus clock cycles (LALE asserted for one bus clock cycle only) 1 Extra bus clock cycles are added (LALE is asserted for the number of bus clock cycles specified by CRR[EADC]).															

### 21.3.2.2.3 Option Registers (ORx)—UPM Mode

Figure 21-6 and Figure 21-7 show the bit fields for ORx when the corresponding BRx[MSEL] selects a UPM machine.



**Figure 21-6. Option Register High Part—UPM Mode**

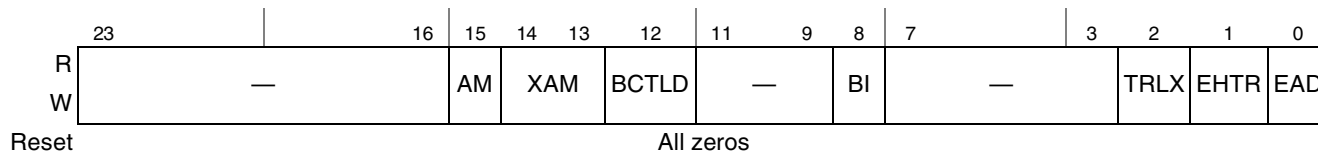
Table 21-9 define ORHx fields for UPM mode.

**Table 21-9. ORHx—UPM Field Descriptions**

Bits	Name	Description
23–10	—	Reserved
9–0	AM	UPM address mask bits 23–14. Masks correspond to BRx bits. Masking address bits independently allows external devices of different size address ranges to be used. Address mask bits can be set or cleared in any order in the field, allowing a resource to reside in more than one area of the address map. 0 Corresponding address bits are masked. 1 The corresponding address bits are used in the comparison with address pins.

Address ORL0 X: 0xFFFE02  
 ORL1 X: 0xFFFE06  
 ORL2 X: 0xFFFE0A  
 ORL3 X: 0xFFFE0E  
 ORL4 X: 0xFFFE12  
 ORL5 X: 0xFFFE16  
 ORL6 X: 0xFFFE1A  
 ORL7 X: 0xFFFE1E

Access: Read/Write



**Figure 21-7. Option Register Low Part (x)—UPM Mode**

Table 21-10 defines ORLx fields for UPM mode.

**Table 21-10. ORLx—UPM Field Descriptions**

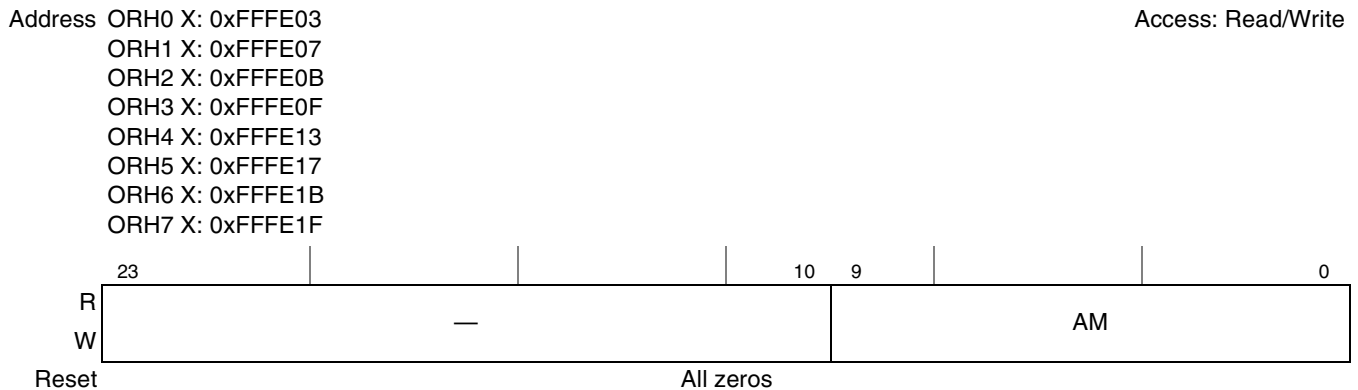
Bits	Name	Description
23–16	—	Reserved
15	AM	UPM address mask bit 13. Masks correspond to BRx bits. Masking address bits independently allows external devices of different size address ranges to be used. Address mask bits can be set or cleared in any order in the field, allowing a resource to reside in more than one area of the address map. 0 Corresponding address bits are masked. 1 The corresponding address bits are used in the comparison with address pins.
14–13	XAM	Extended address mask. Masks the corresponding XBA bits in the BRx register. For example, if BRx[XBA] = 00 or 01, and ORx[XAM] = 10, then the settings permit access from both X and Y.
12	BCTLD	Buffer control disable. Disables assertion of LBCTL during access to the current memory bank. 0 LBCTL is asserted upon access to the current memory bank. 1 LBCTL is not asserted upon access to the current memory bank.
11–9	—	Reserved
8	BI	Burst inhibit. Indicates if this memory bank supports burst accesses. 0 The bank supports burst accesses. 1 The bank does not support burst accesses. The selected UPM executes burst accesses as a series of single accesses.
7–3	—	Reserved

**Table 21-10. ORLx—UPM Field Descriptions (Continued)**

Bits	Name	Description																
2	TRLX	Timing relaxed. Works in conjunction with EHTR to extend hold time on read accesses.																
1	EHTR	Extended hold time on read accesses. Indicates with TRLX how many cycles are inserted between a read access from the current bank and the next access.																
			<table border="1"> <thead> <tr> <th>TRLX</th> <th>EHTR</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Normal timing is generated by the memory controller. No additional cycles are inserted.</td> </tr> <tr> <td>0</td> <td>1</td> <td>1 idle clock cycle is inserted.</td> </tr> <tr> <td>1</td> <td>0</td> <td>4 idle clock cycles are inserted.</td> </tr> <tr> <td>1</td> <td>1</td> <td>8 idle clock cycles are inserted.</td> </tr> </tbody> </table>	TRLX	EHTR	Meaning	0	0	Normal timing is generated by the memory controller. No additional cycles are inserted.	0	1	1 idle clock cycle is inserted.	1	0	4 idle clock cycles are inserted.	1	1	8 idle clock cycles are inserted.
TRLX			EHTR	Meaning														
0			0	Normal timing is generated by the memory controller. No additional cycles are inserted.														
0			1	1 idle clock cycle is inserted.														
1	0	4 idle clock cycles are inserted.																
1	1	8 idle clock cycles are inserted.																
0	EAD	External address latch delay. Allow extra bus clock cycles when using external address latch (LALE). 0 No additional bus clock cycles. (LALE is asserted for one bus clock cycle only.) 1 Extra bus clock cycles are added. (LALE is asserted for the number of bus clock cycles specified by CRR[EADC].)																

### 21.3.2.2.4 Option Registers (ORx)—SDRAM Mode

Figure 21-8 and Figure 21-9 show the bit fields for ORx when the corresponding BRx[MSEL] selects the SDRAM machine.



**Figure 21-8. Option Register High Part (x)—SDRAM Mode**

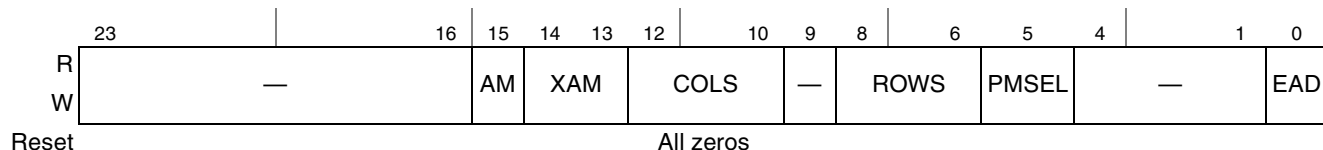
**External Memory Controller (EMC)**

Table 21-11 defines the ORHx fields for SDRAM mode.

**Table 21-11. ORHx—SDRAM Mode**

Bits	Name	Description
23–10	—	Reserved
9–0	AM	SDRAM address mask bits 23–14. Masks correspond to BRx bits. Masking address bits independently allows external devices of different size address ranges to be used. Address mask bits can be set or cleared in any order in the field, allowing a resource to reside in more than one area of the address map. AM can be read or written at any time. 0— Corresponding address bits are masked. 1—The corresponding address bits are used in the comparison with address pins.

Address ORL0 X: 0xFFFE02 Access: Read/Write  
 ORL1 X: 0xFFFE06  
 ORL2 X: 0xFFFE0A  
 ORL3 X: 0xFFFE0E  
 ORL4 X: 0xFFFE12  
 ORL5 X: 0xFFFE16  
 ORL6 X: 0xFFFE1A  
 ORL7 X: 0xFFFE1E



**Figure 21-9. Option Register Low Part (x)—SDRAM Mode**

Table 21-12 defines the ORLx fields for SDRAM mode.

**Table 21-12. ORLx—SDRAM Field Descriptions**

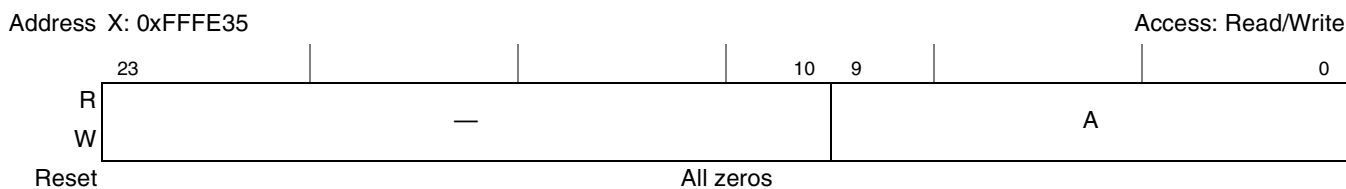
Bits	Name	Description
23–16	—	Reserved
15	AM	SDRAM address mask bit 13. Masks correspond to BRx bits. Masking address bits independently allows external devices of different size address ranges to be used. Address mask bits can be set or cleared in any order in the field, allowing a resource to reside in more than one area of the address map. AM can be read or written at any time. 0 Corresponding address bits are masked. 1 The corresponding address bits are used in the comparison with address pins.
14–13	XAM	Extended address mask. Masks the corresponding XBA bits in the BRx register. For example, if BRx[XBA] = 00 or 01, and ORx[XAM] = 10, then the settings permit access from both X and Y.

**Table 21-12. ORLx—SDRAM Field Descriptions (Continued)**

Bits	Name	Description
12–10	COLS	Number of column address lines. Sets the number of column address lines in the SDRAM device. 000 7 001 8 010 9 011 10 100 11 101 12 110 13 111 14
9	—	Reserved
8–6	ROWS	Number of row address lines. Sets the number of row address lines in the SDRAM device. 000 9 001 10 010 11 011 12 100 13 101 14 110 15 111 Reserved
5	PMSEL	Page mode select. Selects page mode for the SDRAM connected to the memory controller bank. 0 Back-to-back page mode (normal operation). Page is closed when the bus becomes idle. 1 Page is kept open until a page miss or refresh occurs.
4–1	—	Reserved
0	EAD	External address latch delay. Allow extra bus clock cycles when using external address latch (LALE). 0 No additional bus clock cycles (LALE asserted for one bus clock cycle only) 1 Extra bus clock cycles are added (LALE is asserted for the number of bus clock cycles specified by CRR[EADC]).

### 21.3.2.3 UPM Memory Address Register (MAR)

Figure 21-10 and Figure 21-11 show the bit fields for the UPM memory address register.



**Figure 21-10. UPM Memory Address High Part**

## External Memory Controller (EMC)

Table 21-13 defines the MARH fields.

**Table 21-13. MARH Field Descriptions**

Bits	Name	Description
23–10	—	Reserved
9–0	A	The memory address register is used to store A23-A14 of the address which can be output to LAD[23:14] under control of the AMX bits in the UPM RAM word. (AMX = 11)



**Figure 21-11. UPM Memory Address Low Part**

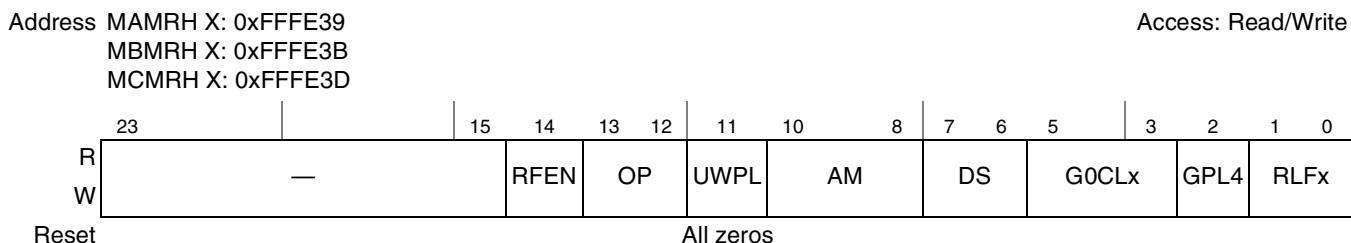
Table 21-14 defines the MARL fields.

**Table 21-14. MARL Field Descriptions**

Bits	Name	Description
23–16	—	Reserved
15–2	A	The memory address register stores A13-A0 of the address, which can be output to LAD[13:0] under control of the AMX bits in the UPM RAM word. (AMX = 11)
1-0	—	Reserved

### 21.3.2.4 UPM Mode Registers (MxMR)

The UPM machine mode registers (MAMR, MBMR and MCMR) contain the configuration for the three UPMs, shown in Figure 21-12 and Figure 21-13.



**Figure 21-12. UPM Mode Registers High Part**

Table 21-15 defines the MxMRH fields.

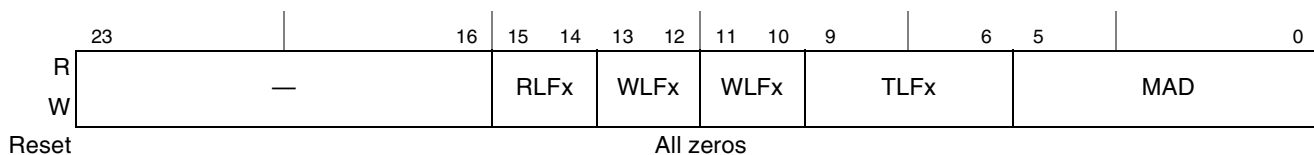
**Table 21-15. MxMR High Part Field Descriptions**

Bits	Name	Description																																																						
23–15	—	Reserved																																																						
14	RFEN	Refresh enable. Indicates that the UPM needs refresh services. This bit must be set for UPMA (refresh executor) if refresh services are required on any UPM assigned chip selects. If MAMR[RFEN] = 0, no refresh services will be provided even if UPMB and/or UPMC have their RFEN bit set. 0 Refresh services are not required. 1 Refresh services are required.																																																						
13–12	OP	Command opcode. Determines the command executed by the UPMx when a memory access hits a UPM assigned bank. 00 Normal operation 01 Write to UPM array. On the next memory access that hits a UPM-assigned bank, write the contents of the MDR into the RAM location pointed to by MAD. After the access, the MAD field is automatically incremented. 10 Read from UPM array. On the next memory access that hits a UPM-assigned bank, read the contents of the RAM location pointed to by MAD into the MDR. After the access, the MAD field is automatically incremented. 11 Run pattern. On the next memory access that hits a UPM-assigned bank, run the pattern written in the RAM array. The pattern run starts at the location pointed to by MAD and continues until the LAST bit is set in the RAM word.																																																						
11	UWPL	UPWAIT polarity active low. Sets the polarity of the UPWAIT pin when in UPM mode 0 UPWAIT is active high 1 UPWAIT is active low																																																						
10–8	AM	Address multiplex size. Determines how the address of the current memory cycle can be output on the address pins. This field is needed when interfacing with devices requiring row and column addresses multiplexed on the same pins.  <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Value</th> <th>LA23–LA14</th> <th>LA13</th> <th>LA12</th> <th>LA11</th> <th>LA10–LA0</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>0</td> <td>A21</td> <td>A20</td> <td>A19</td> <td>A18–A8</td> </tr> <tr> <td>001</td> <td>0</td> <td>A22</td> <td>A21</td> <td>A20</td> <td>A19–A9</td> </tr> <tr> <td>010</td> <td>0</td> <td>A23</td> <td>A22</td> <td>A21</td> <td>A20–A10</td> </tr> <tr> <td>011</td> <td>0</td> <td>0</td> <td>A23</td> <td>A22</td> <td>A21–A11</td> </tr> <tr> <td>100</td> <td>0</td> <td>0</td> <td>0</td> <td>A23</td> <td>A22–A12</td> </tr> <tr> <td>101</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>A23–A13</td> </tr> <tr> <td>110</td> <td colspan="5" style="text-align: center;">Reserved</td> </tr> <tr> <td>111</td> <td colspan="5" style="text-align: center;">Reserved</td> </tr> </tbody> </table>	Value	LA23–LA14	LA13	LA12	LA11	LA10–LA0	000	0	A21	A20	A19	A18–A8	001	0	A22	A21	A20	A19–A9	010	0	A23	A22	A21	A20–A10	011	0	0	A23	A22	A21–A11	100	0	0	0	A23	A22–A12	101	0	0	0	0	A23–A13	110	Reserved					111	Reserved				
Value	LA23–LA14	LA13	LA12	LA11	LA10–LA0																																																			
000	0	A21	A20	A19	A18–A8																																																			
001	0	A22	A21	A20	A19–A9																																																			
010	0	A23	A22	A21	A20–A10																																																			
011	0	0	A23	A22	A21–A11																																																			
100	0	0	0	A23	A22–A12																																																			
101	0	0	0	0	A23–A13																																																			
110	Reserved																																																							
111	Reserved																																																							

**Table 21-15. MxMR High Part Field Descriptions (Continued)**

Bits	Name	Description														
7–6	DS	<p>Disable timer period. Guarantees a minimum time between accesses to the same memory bank controlled by UPMx. The disable timer is turned on by the TODT bit in the RAM array word, and after it expires, the UPMx allows the machine access to handle a memory pattern to the same bank. Accesses to a different bank by the same UPMx is also allowed.</p> <p>To avoid conflicts between successive accesses to different banks, the minimum pattern in the RAM array for a request serviced, should not be shorter than the period established by DS.</p> <p>00 1-bus clock cycle disable period            01 2-bus clock cycle disable period            10 3-bus clock cycle disable period            11 4-bus clock cycle disable period</p>														
5–3	G0CLx	<p>General line 0 control. Determines which logical address line can be output to the LGPL0 pin when the UPMx is selected to control the memory access.</p> <p>000 A17            001 A18            010 A19            011 A20            100 A21            101 A22            110 A23            111 Reserved</p>														
2	GPL4	<p>LGPL4 output line disable. Determines how the LGPL4/UPWAIT pin is controlled by the corresponding bits in the UPMx array. See <a href="#">Table 21-42</a>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th rowspan="2">Value</th> <th rowspan="2">LGPL4/UPWAIT pin function</th> <th colspan="2">Interpretation of UPM word bits</th> </tr> <tr> <th>G4T1/DLT3</th> <th>G4T3/WAEN</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LGPL4 (output)</td> <td>G4T1</td> <td>G4T3</td> </tr> <tr> <td>1</td> <td>UPWAIT (input)</td> <td>DLT3</td> <td>WAEN</td> </tr> </tbody> </table>	Value	LGPL4/UPWAIT pin function	Interpretation of UPM word bits		G4T1/DLT3	G4T3/WAEN	0	LGPL4 (output)	G4T1	G4T3	1	UPWAIT (input)	DLT3	WAEN
Value	LGPL4/UPWAIT pin function	Interpretation of UPM word bits														
		G4T1/DLT3	G4T3/WAEN													
0	LGPL4 (output)	G4T1	G4T3													
1	UPWAIT (input)	DLT3	WAEN													
1–0	RLFx	<p>Read loop field. Determines the number of times a loop defined in the UPMx will be executed for a burst- or single-beat read pattern or when MxMR[OP] = 11 (RUN command)</p> <p>0000 16            0001 1            0010 2            0011 3            ...            1110 14            1111 15</p>														

Address MAMRL X: 0xFFFE38 Access: Read/Write  
 MBMRL X: 0xFFFE3A  
 MCMRL X: 0xFFFE3C



**Figure 21-13. UPM Mode Registers Low Part**



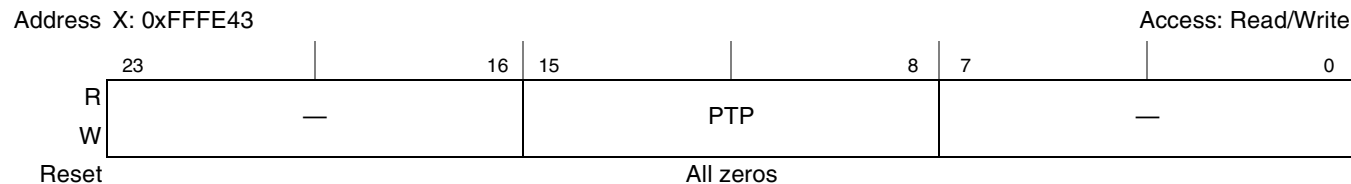
Table 21-16 defines the MxMRL fields.

**Table 21-16. MxMRL Field Descriptions**

Bits	Name	Description
23–16	—	Reserved
15–14	RLFx	Read loop field. Determines the number of times a loop defined in the UPMx will be executed for a burst- or single-beat read pattern or when MxMR[OP] = 11 (RUN command). 0000 16 0001 1 0010 2 0011 3 ... 1110 14 1111 15
13–10	WLFx	Write loop field. Determines the number of times a loop defined in the UPMx will be executed for a burst- or single-beat write pattern. 0000 16 0001 1 0010 2 0011 3 ... 1110 14 1111 15
9–6	TLFx	Refresh loop field. Determines the number of times a loop defined in the UPMx will be executed for a refresh service pattern. 0000 16 0001 1 0010 2 0011 3 ... 1110 14 1111 15
5–0	MAD	Machine address. RAM address pointer for the command executed. Each time that the UPM is accessed and the OP field is set to WRITE or READ, this field is incremented by 1. Address range is 64 words per UPMx.

### 21.3.2.5 Memory Refresh Timer Prescaler Register (MRTPR)

The refresh timer prescaler register is used to divide the system clock to provide the SDRAM and UPM refresh timers clock, shown in [Figure 21-14](#).



**Figure 21-14. Memory Refresh Timer Prescaler Register**

**Table 21-17. MRTPR Field Descriptions**

Bits	Name	Description
2–16	—	Reserved
15–8	PTP	Refresh timers prescaler. Determines the period of the refresh timers input clock. The system clock is divided by PTP except when the value is 000000000, which represents the maximum divider of 256.
7–0	—	Reserved

**NOTE**

The MRTPR register has no low part. The low part is reserved and the corresponding address should not be used.

**21.3.2.6 UPM Data Register (MDR)**

The memory data register (MDR) contains data written to or read from the RAM array for UPM read or write commands, shown in [Figure 21-15](#) and [Figure 21-16](#). Before issuing a write command to the UPM, the MDR must be set up.

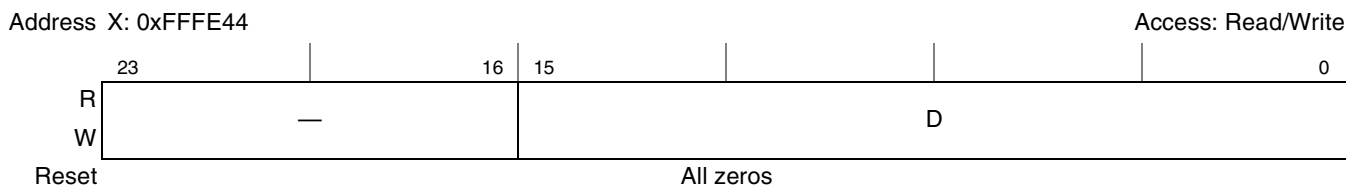


**Figure 21-15. UPM Data Register High Part**

[Table 21-18](#) defines the MDRH fields.

**Table 21-18. MDRH Field Descriptions**

Bits	Name	Description
23–16	—	Reserved
15–0	D	Bits 31–16 of the data to be read or written into the RAM array, when a write or read command is supplied to the UPM (MxMR[OP] = 01 or MxMR[OP] = 10).



**Figure 21-16. UPM Data Register Low Part**

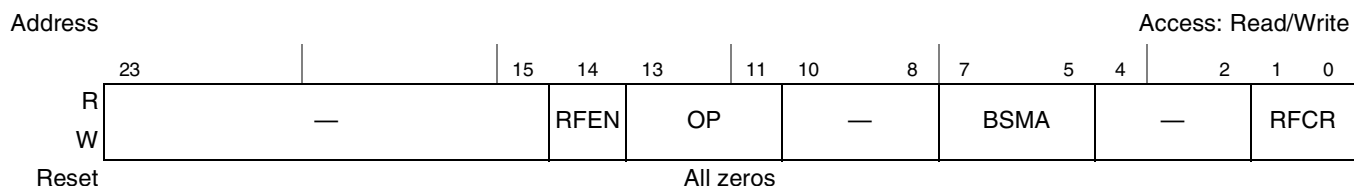
[Table 21-19](#) defines the MDRL fields.

**Table 21-19. MDRL Field Descriptions**

Bits	Name	Description
23–16	—	Reserved
15–0	D	Bits 15–0 of the data to be read or written into the RAM array when a write or read command is supplied to the UPM (MxMR[OP] = 01 or MxMR[OP] = 10).

### 21.3.2.7 SDRAM Machine Mode Register (SDMR)

The external memory SDRAM mode register (SDMR) is used to configure operations pertaining to SDRAM, shown in [Figure 21-17](#) and [Figure 21-18](#).



**Figure 21-17. SDRAM Machine Mode Register High Part**

[Table 21-20](#) defines the SDMRH fields.

**Table 21-20. SDMRH Field Descriptions**

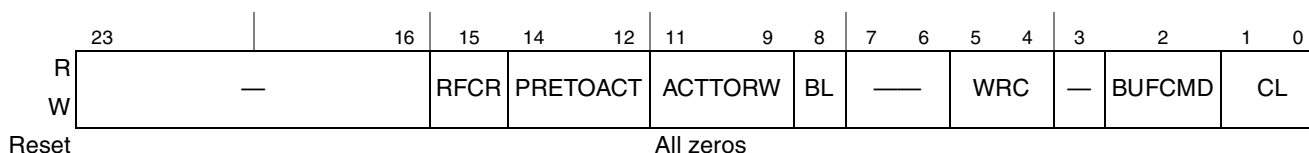
Bits	Name	Description																											
23–15	—	Reserved																											
14	RFEN	Refresh enable. Indicates that the SDRAM requires refresh services. 0 Refresh services are not required. 1 Refresh services are required.																											
13–11	OP	SDRAM operation. Selects the operation that occurs when the SDRAM device is accessed. <table border="1" style="margin: 10px auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Value</th> <th>Meaning</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Normal operation</td> <td>Normal operation</td> </tr> <tr> <td>001</td> <td>Auto refresh</td> <td>Initialization</td> </tr> <tr> <td>010</td> <td>Self refresh</td> <td>Power down mode or Debug</td> </tr> <tr> <td>011</td> <td>Mode Register write</td> <td>Initialization</td> </tr> <tr> <td>100</td> <td>Precharge bank</td> <td>Debug</td> </tr> <tr> <td>101</td> <td>Precharge all banks</td> <td>Initialization</td> </tr> <tr> <td>110</td> <td>Activate bank</td> <td>Debug</td> </tr> <tr> <td>111</td> <td>Read/write without valid data transfer</td> <td>Debug</td> </tr> </tbody> </table>	Value	Meaning	Use	000	Normal operation	Normal operation	001	Auto refresh	Initialization	010	Self refresh	Power down mode or Debug	011	Mode Register write	Initialization	100	Precharge bank	Debug	101	Precharge all banks	Initialization	110	Activate bank	Debug	111	Read/write without valid data transfer	Debug
Value	Meaning	Use																											
000	Normal operation	Normal operation																											
001	Auto refresh	Initialization																											
010	Self refresh	Power down mode or Debug																											
011	Mode Register write	Initialization																											
100	Precharge bank	Debug																											
101	Precharge all banks	Initialization																											
110	Activate bank	Debug																											
111	Read/write without valid data transfer	Debug																											
10–8	—	Reserved																											

**Table 21-20. SDMRH Field Descriptions (Continued)**

Bits	Name	Description
7–5	BSMA	Bank select multiplexed address line. Selects which address pins serve as the 2-bit bank-select address for SDRAM. Note that only 4-bank SDRAMs are supported. 000 LA17:LA16 001 LA16:LA15 010 LA15:LA14 011 LA14:LA13 100 LA13:LA12 101 LA12:LA11 110 LA11:LA10 111 LA10:LA9
4–2	—	Reserved
1–0	RFCR	Refresh recovery. Sets the refresh recovery interval in bus clock cycles. Defines the earliest timing for an ACTIVATE or REFRESH command after a REFRESH command. 000 Reserved 001 3 clocks 010 4 clocks 011 5 clocks 100 6 clocks 101 7 clocks 110 8 clocks 111 16 clocks

Address X: 0xFFFE4A

Access: Read/Write



**Figure 21-18. SDRAM Machine Mode Register Low Part**

Table 21-21 defines the SDML fields.

**Table 21-21. SDML Field Descriptions**

Bits	Name	Description
23–16	—	Reserved
15	RFCR	Refresh recovery. Sets the refresh recovery interval in bus clock cycles. Defines the earliest timing for an ACTIVATE or REFRESH command after a REFRESH command. 000 Reserved 001 3 clocks 010 4 clocks 011 5 clocks 100 6 clocks 101 7 clocks 110 8 clocks 111 16 clocks

**Table 21-21. SDMRL Field Descriptions (Continued)**

Bits	Name	Description
14–12	PRETOACT	Defines the earliest timing for an ACTIVATE or REFRESH command after a PRECHARGE command (number of bus clock cycle wait states). 000 8 001 1 010 2 011 3 100 4 101 5 110 6 111 7
11–9	ACTTORW	Defines the earliest timing for a READ/WRITE command after an ACTIVATE command (number of bus clock cycle wait states). 000 8 001 Reserved 010 2 011 3 100 4 101 5 110 6 111 7
8	BL	Sets the burst length for SDRAM accesses. 0 Reserved 1 SDRAM burst length is 8.
7–6	—	Reserved
5–4	WRC	Write recovery time. Defines the earliest timing for a PRECHARGE command after the last data is written to the SDRAM. 00 4 01 Reserved 10 2 11 3
3	—	Reserved
2	BUFCMD	Control line assertion timing. If external buffers are placed on the control lines going to both the SDRAM and address lines, setting BUFCMD causes all SDRAM control lines (except $\overline{\text{LCSx}}$ , LCKE, LALE and $\overline{\text{LSDDQM}}$ ) to be asserted for CRR[BUFCMDC] cycles, instead of being asserted for one cycle. 0 Normal timing for the control lines 1 All control lines except $\overline{\text{LCSx}}$ are asserted for the number of cycles specified by CRR[BUFCMDC].
1–0	CL	CAS latency. Defines the timing for first read data after a column address is sampled by the SDRAM. 00 Extended CAS latency. According to CRR[ECL]. See <a href="#">Table 21-33 on page 21-35</a> . 01 1 10 2 11 3

### 21.3.2.8 UPM Refresh Timer (URT)

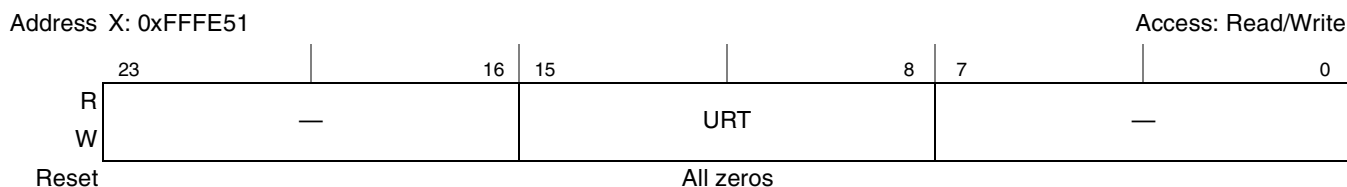
The UPM refresh timer (URT), shown in [Figure 21-19](#), generates a refresh request for all valid banks that selected a UPM machine and are refresh-enabled ( $\text{MxMR}[\text{RFEN}] = 1$ ). Each time the timer expires, a

## External Memory Controller (EMC)

qualified bank generates a refresh request using the selected UPM. The qualified banks rotate their requests.

### NOTE

The URT register has no low part. The low part is reserved and the corresponding address should not be used.



**Figure 21-19. UPM Refresh Timer**

Table 21-22 defines the URT fields.

**Table 21-22. URT Field Descriptions**

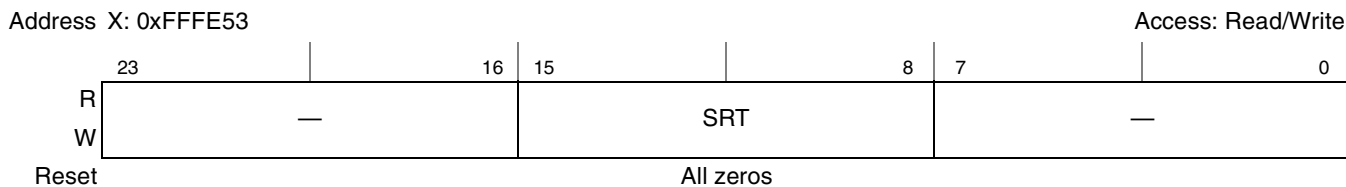
Bits	Name	Description
23–16	—	Reserved
15–8	URT	<p>UPM refresh timer period. Determines, along with the timer prescaler (MRTPR), the timer period according to the following equation:</p> $\text{TimerPeriod} = \frac{\text{URT}}{\left(\frac{\text{Fsystemclock}}{\text{MRTPR[PTP]}}\right)}$ <p>Note that a value of 0x00 (reset value) sets the maximum period of 256 x MRTPR[PTP] system clock cycles.  <i>Example:</i> For a 200 MHz system clock and a required refresh rate of 16 μs, and given MRTPR[PTP] = 32, the URT value should be 100 (decimal). 100/(200 MHz/32) = 16 μs, which meets the demand for the required refresh period of 16 μs.</p>
7-0	—	Reserved

### 21.3.2.9 SDRAM Refresh Timer (SRT)

The SDRAM refresh timer (SRT), shown in Figure 21-20, generates a refresh request for all valid banks that selected a SDRAM machine and are refresh-enabled (SDMR[RFEN] = 1). Each time the timer expires, all banks that qualify generate a staggered bank auto-refresh request using the SDRAM machine.

### NOTE

The SDRAM refresh timer register has no low part. The low part is reserved and the corresponding address should not be used.



**Figure 21-20. SDRAM Refresh Timer**

Table 21-23 defines the SRT fields.

**Table 21-23. SRT Field Descriptions**

Bits	Name	Description
23–16	—	Reserved
15–8	SRT	<p>SDRAM refresh timer period. Determines, along with the timer prescaler (MRTPR), the timer period according to the following equation:</p> $\text{TimerPeriod} = \frac{\text{SRT}}{\left(\frac{\text{Fsystemclock}}{\text{MRTPR[PTP]}}\right)}$ <p>Note that a value of 0x00 (reset value) sets the maximum period of 256 x MRTPR[PTP] system clock cycles.  <i>Example:</i> For a 200 MHz system clock and a required refresh rate of 16 μs, and given MRTPR[PTP] = 32, the SRT value should be 100 (decimal). 100/(200 MHz/32) = 16 μs, which meets the demand for required refresh period of 16 μs.</p>
7-0	—	Reserved

### 21.3.2.10 Transfer Error Status Register (TESR)

The EMC has the following registers for error management:

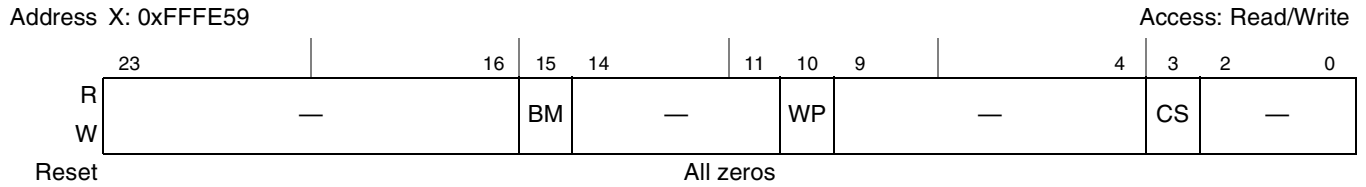
- TESR indicates the cause of an error.
- TEDR enables/disables error checking.
- TEIR enables/disables reporting of errors through an interrupt.
- TEATR captures the source attributes of an error.
- TEAR captures the address of a transaction that caused an error.

TESR, shown in Figure 21-21, is a write-clear type of register. Reading from the TESR register occurs normally; however, write operations can clear but not set bits. A bit is cleared whenever the register is written and the data in the corresponding bit location is a 1. For example, to clear the write protect error bit (TESR[WP]) and not affect any other bits in the register, the value 0x000400 should be written to the register.

#### NOTE

The TESR register has no low part. The low part is reserved and the corresponding address should not be used.

## External Memory Controller (EMC)



**Figure 21-21. Transfer Error Status Register**

Table 21-24 defines the TESR fields.

**Table 21-24. TESR Field Descriptions**

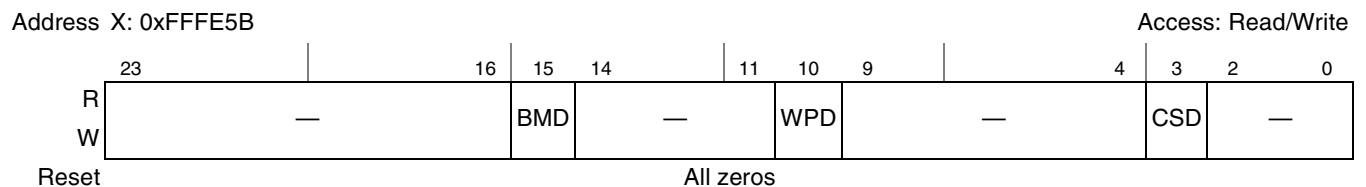
Bits	Name	Description
23–16	—	Reserved
15	BM	Bus monitor time-out 0 No external memory monitor time-out has occurred. 1 External memory monitor time-out has occurred. No data beat was acknowledged on the bus within BCR[BMT] x 8 bus clock cycles from the start of a transaction.
14–11	—	Reserved
10	WP	Write protect error 0 No write-protect error has occurred. 1 A write was attempted to a external memory region that was defined as read-only in the memory controller. Usually, in this case, a bus monitor time-out will occur.
9–4	—	Reserved
3	CS	Chip select error 0 No chip select error has occurred. 1 A transaction was sent to the EMC that did not hit any memory bank.
2–0	—	Reserved

### 21.3.2.11 Transfer Error Check Disable Register (TEDR)

The TEDR register is used to disable error checking, shown in Figure 21-22. Note that control of error checking is independent of control of reporting of errors (TEIR) via the interrupt mechanism.

#### NOTE

The TEDR register has no low part. The low part is reserved and the corresponding address should not be used.



**Figure 21-22. Transfer Error Check Disable Register**

Table 21-25 defines the TEDR fields.



**Table 21-25. TEDR Field Descriptions**

Bits	Name	Description
23–16	—	Reserved
15	BMD	Bus monitor disable 0 Bus monitor is enabled. 1 Bus monitor is disabled.
14–11	—	Reserved
10	WPD	Write-protect error checking disable 0 Write protect error checking is enabled. 1 Write protect error checking is disabled.
9–4	—	Reserved
3	CSD	Chip select error checking disable 0 Chip select error checking is enabled. 1 Chip select error checking is disabled.
2–0	—	Reserved

### 21.3.2.12 Transfer Error Interrupt Enable Register (TEIR)

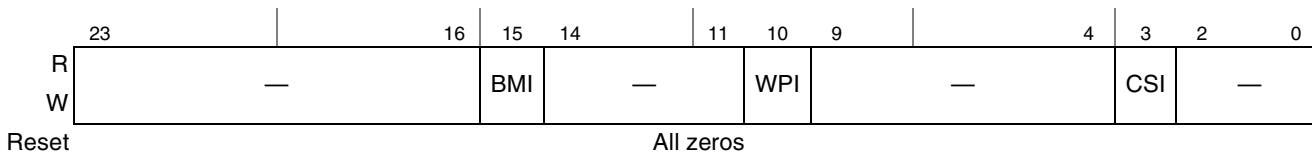
The TEIR register is used to send or block error reporting through the EMC internal interrupt mechanism, shown in [Figure 21-23](#). Software should clear pending errors in the TCSR register before enabling interrupts. After an interrupt has occurred, clearing relevant TCSR error bits negates the interrupt.

#### NOTE

The TEIR register has no low part. The low part is reserved and the corresponding address should not be used.

Address X:X: 0xFFFE5D

Access: Read/Write



**Figure 21-23. Transfer Error Interrupt Enable Register**

[Table 21-26](#) defines the TEIR fields.

**Table 21-26. TEIR Field Descriptions**

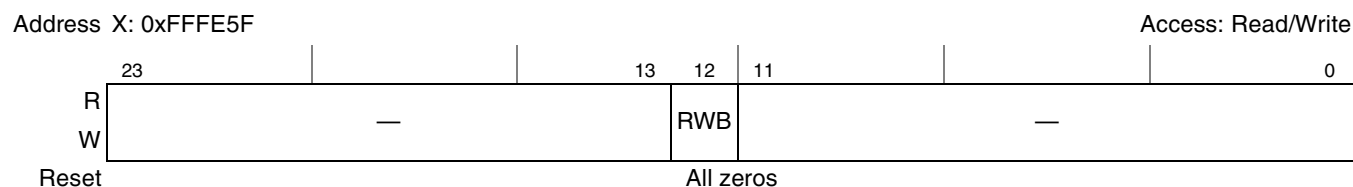
Bits	Name	Description
23–16	—	Reserved
15	BMI	Bus monitor error interrupt enable 0 Bus monitor error reporting is disabled. 1 Bus monitor error reporting is enabled.
14–11	—	Reserved

**Table 21-26. TEIR Field Descriptions (Continued)**

Bits	Name	Description
10	WPI	Write-protect error interrupt enable 0 Write-protect error reporting is disabled. 1 Write-protect error reporting is enabled.
9–4	—	Reserved
3	CSI	Chip select error interrupt enable 0 Chip select error reporting is disabled. 1 Chip select error reporting is enabled.
2–0	—	Reserved

### 21.3.2.13 Transfer Error Attributes Register (TEATR)

Figure 21-24 and Figure 21-25 show the TEATR register. After the TEATR[V] bit has been set (1), software must clear this bit to allow the TEATR and TEAR registers to update any subsequent errors.

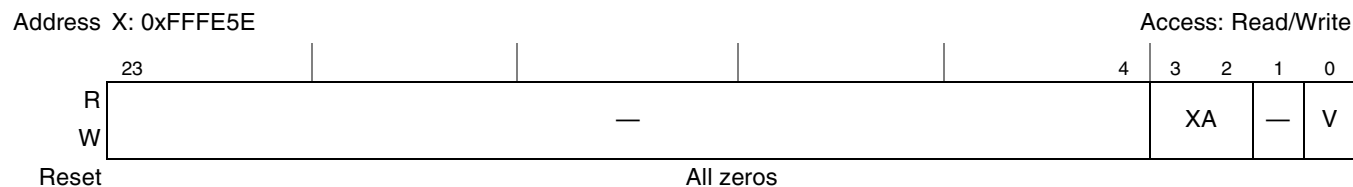


**Figure 21-24. Transfer Error Attributes Register High Part**

Table 21-27 defines the TEATR<sub>H</sub> fields.

**Table 21-27. TEATR<sub>H</sub> Field Descriptions**

Bits	Name	Description
23–13	—	Reserved
12	RWB	Transaction type for the error: 0 The transaction for the error was a write transaction. 1 The transaction for the error was a read transaction.
11–0	—	Reserved



**Figure 21-25. Transfer Error Attributes Register Low Part**

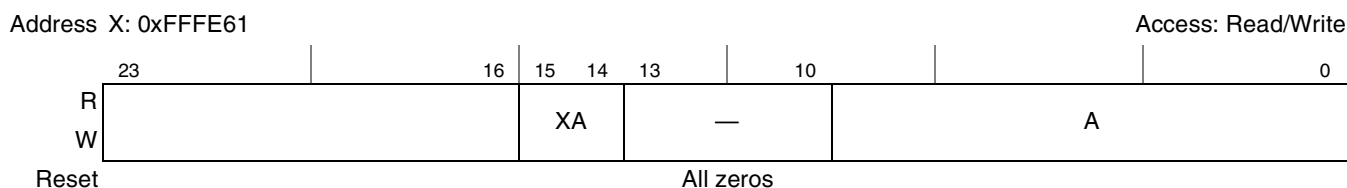
Table 21-28 defines the TEATRL fields.

**Table 21-28. TEATRL Field Descriptions**

Bits	Name	Description
23–4	—	Reserved
3–2	XA	Extended address for the error. The address indicates which type of access (X, Y or P) makes the error happen. 00 X access 01 Y access 10 P access (read/write) 11 P access (instruction fetch)
1	—	Reserved
0	V	Error attribute capture is valid. Indicates that the captured error information is valid. 0 Captured error attributes and address are not valid. 1 Captured error attributes and address are valid.

### 21.3.2.14 Transfer Error Address Register (TEAR)

Figure 21-26 and Figure 21-27 show the TEAR.



**Figure 21-26. Transfer Error Address Register High Part**

Table 21-29 defines the TEARH fields.

**Table 21-29. TEARH Field Descriptions**

Bits	Name	Description
23–10	—	Reserved
15–14	XA	Extended transaction address for the error. The address indicates the access is from X, Y, or P memory space.
13–10	—	Reserved
9–0	A	Transaction address A[23:14] for the error.



**Figure 21-27. Transfer Error Address Register Low Part**

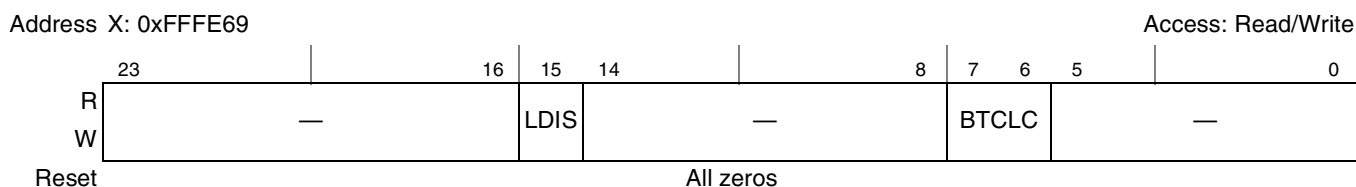
Table 21-30 defines the TEARL fields.

**Table 21-30. TEARL Field Descriptions**

Bits	Name	Description
23–16	—	Reserved
15–2	A	Transaction address A[13:0] for the error.
1–0	—	Reserved

### 21.3.2.15 Local Bus Configuration Register (LBCR)

Figure 21-28 and Figure 21-29 show the LBCR.

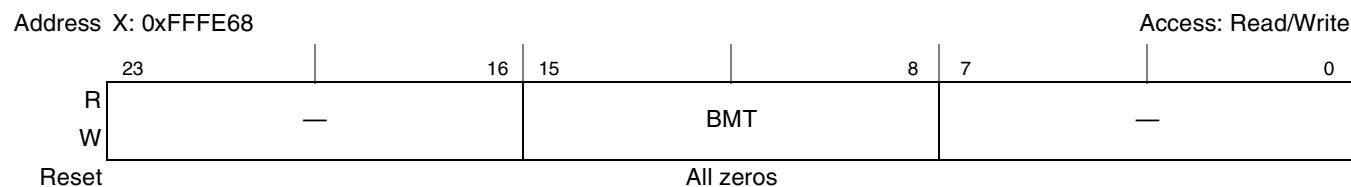


**Figure 21-28. Local Bus Configuration Register High Part**

Table 21-31 defines the LBCRH fields.

**Table 21-31. LBCRH Field Description**

Bits	Name	Description
23–16	—	Reserved
15	LDIS	Local bus disable 0 Local bus is enabled. 1 Local bus is disabled. No internal transactions will be acknowledged.
14–8	—	Reserved
7–6	BCTLC	Defines the use of LBCTL. 00 LBCTL is used as $W/\overline{R}$ control for GPCM or UPM accesses (buffer control). 01 LBCTL is used as $\overline{LOE}$ for GPCM accesses only. 10 LBCTL is used as $\overline{LWE}$ for GPCM accesses only. 11 Reserved
5–0	—	Reserved



**Figure 21-29. Local Bus Configuration Register Low Part**

Table 21-32 defines the LBCRL fields.

**Table 21-32. LBCRL Field Descriptions**

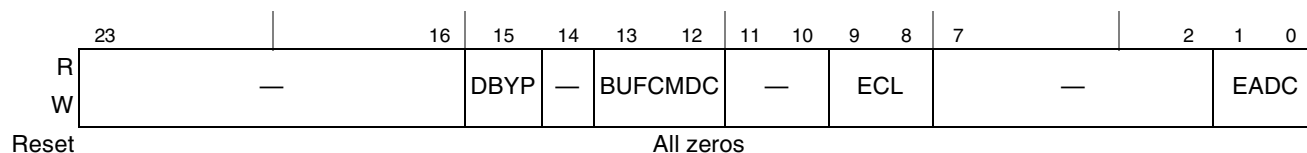
Bits	Name	Description
23–16	—	Reserved
15–8	BMT	Bus monitor timing. Defines the bus monitor time-out period. Clearing BMT (reset value) selects the maximum count of 2048 bus clock cycles. For non-zero values of BMT, the number of LCLK clock cycles to count down (before a time-out error is generated) is given by: bus cycles = BMT x 8. Apart from BMT = 0x00, the minimum value of BMT is 5, corresponding to 40 bus cycles. Shorter time-outs may result in spurious errors during SDRAM operation.
7–0	—	Reserved

### 21.3.2.16 Clock Ratio Register (LCRR)

The clock ratio register sets the system clock to the EMC bus frequency ratio, shown in Figure 21-30 and Figure 21-31. The clock ratio register also provides configuration bits for extra delay cycles for address and control signals.

Address X: 0xFFFE6B

Access: Read/Write



**Figure 21-30. Clock Ratio Register High Part**

Table 21-33 defines the LCRRH fields.

**Table 21-33. LCRRH Field Description**

Bits	Name	Description
23–16	—	Reserved
15	DBYP	PLL Bypass. This bit should be set when using low bus clock frequencies less than 50 . When in PLL bypass mode, incoming data is captured in the middle of the bus clock cycle. 0 The PLL is enabled. 1 The PLL is bypassed.
14	—	Reserved
13–12	BUFCMDC	Additional delay cycles for SDRAM control signals. Defines the number of cycles to be added for each SDRAM command when SDMR[BUFCMD] = 1. 00 4 01 1 10 2 11 3
11–10	—	Reserved

**Table 21-33. LCRRH Field Description (Continued)**

Bits	Name	Description
9–8	ECL	Extended CAS latency. Determines the extended CAS latency for SDRAM accesses when SDMR[CL] = 00. 00 4 01 5 10 6 11 7
7–2	—	Reserved
1–0	EADC	External address delay cycles. Defines the number of cycles for the assertion of LALE. 00 4 01 1 10 2 11 3



**Figure 21-31. Clock Ratio Register Low Part**

Table 21-34 defines the LCRRL fields.

**Table 21-34. LCRRL Field Descriptions**

Bits	Name	Description
23–4	—	Reserved
3–0	CLKDIV	System (input) clock divider. Sets the frequency ratio between the (input) system clock and the memory bus clock. Only the values shown in the list below are allowed. 0000 Reserved 0001 Reserved 0010 2 0011 Reserved 0100 4 0101 Reserved 0110 Reserved 0111 Reserved 1000 8 1001–1111 Reserved

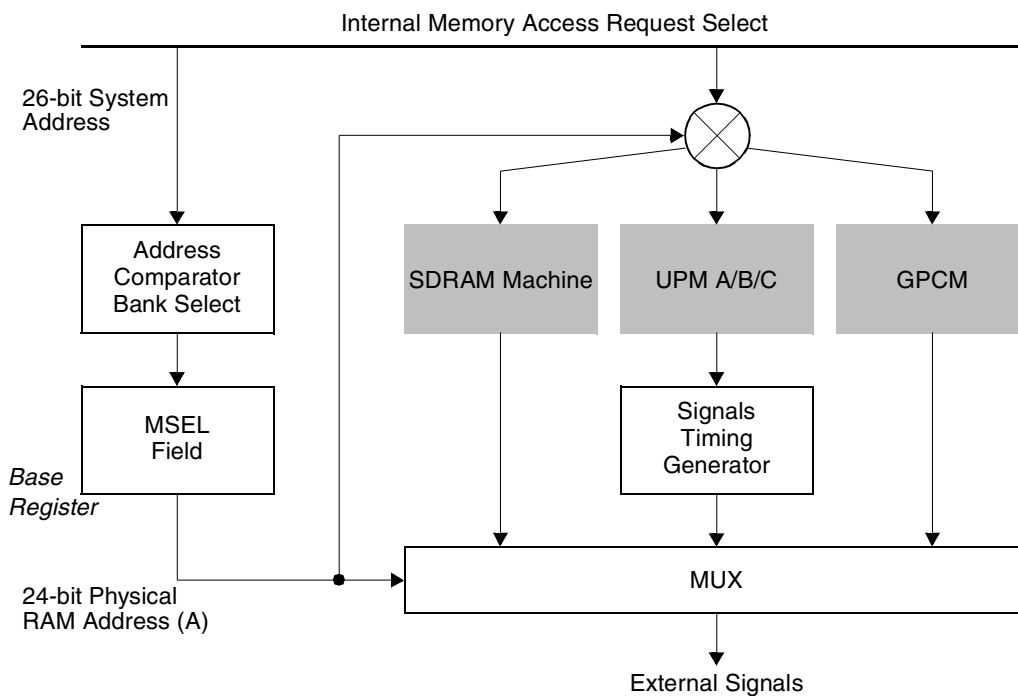
## 21.4 Functional Description

The EMC allows the implementation of memory systems with specific timing requirements. In this section, the SDRAM, GPCM, and UPM machines will be described using diagrams and waveforms. Unless it is noted otherwise, the waveforms shown are observed in the de-skew PLL enable mode.

- To achieve high performance through a multiplexed address and data bus, the SDRAM machine provides an interface to synchronous DRAMs using bank interleaving and back-to-back page

mode. An internal phase-locked loop (PLL) for bus clock generation ensures improved data set-up margins for board designs.

- The GPCM machine provides interfacing for simpler, lower-performance memories and memory-mapped devices. The GPCM has inherently lower performance because it does not support bursting. For this reason, GPCM-controlled banks are used primarily for boot-loading and access to low-performance memory-mapped peripherals.
- The UPM machine supports refresh timers, address multiplexing of the external bus, and the generation of programmable control signals for row address and column address strobes, to allow for a minimal glue logic interface to DRAMs, burstable SRAMs, and almost any other type of peripheral. The UPM can be used to generate flexible, user-defined timing patterns for control signals that govern a memory device. These patterns define how the external control signals behave during a read, write, burst-read, or burst-write access. Refresh timers are also available to periodically initiate user-defined refresh patterns.



**Figure 21-32. Basic Operation of Memory Controllers in EMC**

Each memory bank (chip select) can be assigned to any one of these three type of machines via the machine select bits of the base register for that bank (BRx[MSEL]), as illustrated in [Figure 21-32](#). If a bank match occurs, the corresponding machine (GPCM, SDRAM or UPM) then takes ownership of the external signals that control the access and maintains control until the transaction ends.

## 21.4.1 Basic Architecture

### 21.4.1.1 Address and Address Space Checking

The defined base addresses are written to the BRx registers, while the corresponding address masks are written to the ORx registers. Each time an external memory access is requested, the internal transaction address is compared with each bank. Addresses are decoded by comparing the 11 MSBs of the address, masked by ORx[XAM] and ORx[AM], with the base address for each bank (BRx[XBA] and BRx[BA]). If a match is found on a memory controller bank, the attributes defined in the BRx and ORx for that bank are used to control the memory access. If a match is found in more than one bank, the lowest-numbered bank handles the memory access (that is, bank 0 has priority over bank 1).

### 21.4.1.2 External Address Latch Enable Signal (LALE)

The external memory bus uses a multiplexed address and data bus. Therefore, the EMC must distinguish between address and data phases, which take place on the same bus (LAD[23:0] pins). The LALE signal, when asserted, signifies an address phase during which the EMC drives the memory address on the LAD[23:0] pins. An external address latch uses the LALE signal to capture the address and provide it to the address pins of the memory or peripheral device. When LALE is negated, LAD[23:0] then serves as the (bidirectional) data bus for the access. Any address phase initiates the assertion of LALE, which has a programmable duration of 1–4 bus clock cycles.

The frequency of LALE assertion varies across the three memory controllers. In the case of GPCM, every assertion of  $\overline{\text{LCSx}}$  is considered an independent access, and accordingly, LALE will assert prior to each such access. For example, GPCM would assert LALE and  $\overline{\text{LCSx}}$  8 times to satisfy an 8-words cache line transfer. The SDRAM controller asserts LALE only to initiate a burst transfer with a starting address, therefore no more than one assertion of LALE may be required for SDRAM to transfer a 8-word cache line.

In the case of UPM, the frequency of LALE assertion depends on how the UPM RAM is programmed. UPM single accesses typically assert LALE once, upon commencement, but it is possible to program UPM to assert LALE several times, and to change the values of LA[2:0] with and without LALE being involved. In general, when using the GPCM and SDRAM controllers it is not necessary to use LA[2:0] if a sufficiently wide latch is used to capture the entire address during LALE phases. UPM may require LA[2:0] if EMC is generating its own burst address sequence.

### 21.4.1.3 Data Transfer Acknowledge (TA)

The three memory controllers in the EMC generate an internal transfer acknowledge signal (TA), to allow data on LAD[23:0] to be either sampled (for reads) or changed (on writes). The data sampling/data change always occurs at the end of the bus cycle in which TA is asserted internally by the EMC. GPCM and SDRAM controllers automatically generate TA according to the timing parameters programmed for them in option and mode registers; UPM generates TA only when a UPM pattern has the UTA RAM word bit set. An illustration of LALE, TA (internal), and  $\overline{\text{LCSx}}$  is shown in [Figure 21-33](#). Note that TA and LALE are never asserted together, and that for the duration of LALE,  $\overline{\text{LCSx}}$  (or any other control signal) remains negated or frozen.



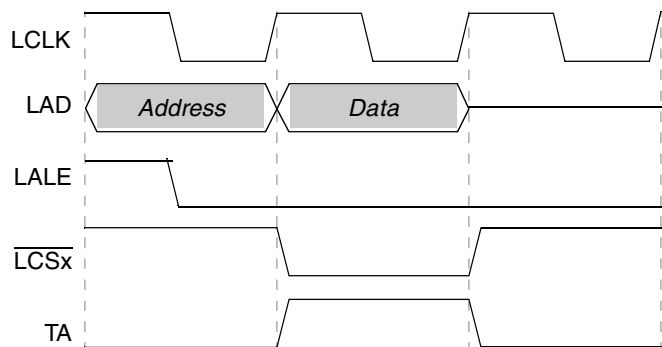


Figure 21-33. Basic EMC Bus Cycle with LALE, TA, and  $\overline{\text{LCSx}}$

### 21.4.1.4 Data Buffer Control (LBCTL)

The memory controller provides a data buffer control signal for the EMC data buffer control (LBCTL). LBCTL is activated when a GPCM or UPM controlled bank is accessed. LBCTL can be disabled by setting  $\text{ORx}[\text{BCTL D}]$ . Access to an SDRAM machine controlled bank does not activate the LBCTL control. The LBCTL control can be further configured by  $\text{BCR}[\text{BCTL C}]$  to act as an extra  $\overline{\text{LWE}}$  or an extra  $\overline{\text{LOE}}$  signal when in GPCM mode.

If LBCTL is configured as a data buffer control ( $\text{BCR}[\text{LBCTL C}] = 00$ ), LBCTL is asserted (high) on the rising edge of the bus clock on the first cycle of the memory controller operation, coincident with LALE. If the access is a write, LBCTL remains high for the whole duration. However, if the access is a read, LBCTL is negated (low) with the negation of LALE so that the memory device is able to drive the bus. If back-to-back read accesses are pending, LBCTL is asserted (high) one bus clock cycle before the next transaction starts (that is, one bus clock cycle before LALE) to allow a whole bus cycle for the bus to turn around before the next address is driven.

If an external bus transceiver is used, LBCTL should be used to signify the write direction when high. Note that the default (reset and bus idle) value of LBCTL is also high.

### 21.4.1.5 Bus Monitor

A bus monitor is provided to ensure that each bus cycle is terminated within a reasonable period (user-defined). When a transaction starts, the bus monitor starts counting down from the time-out value ( $\text{LBCR}[\text{BMT}]$ ) until a data beat is acknowledged on the bus. The bus monitor then reloads the time-out value and resumes the count down until the data tenure is completed, and then idles if there is no pending transaction. Bus monitor error reporting through  $\text{TESR}[\text{BM}]$  can be disabled by setting the  $\text{TEDR}[\text{BMD}]$  bit. However, even if bus monitor error reporting is disabled, the bus monitor is still active and can generate a UPM exception, or even terminate a GPCM access.

It is important to ensure that the value of  $\text{BCR}[\text{BMT}]$  is not set too low; otherwise spurious bus time-outs may occur during normal operation—particularly for SDRAMs—resulting in incomplete data transfers. Accordingly, apart from the reset value of 0x00 (corresponding with the maximum time-out of 2048 bus cycles),  $\text{BCR}[\text{BMT}]$  must not be set below 0x05 (or 40 bus cycles for time-out) under any circumstances.

## 21.4.2 General-Purpose Chip-Select Machine (GPCM)

The GPCM allows a minimal glue logic and flexible interface to SRAM, EPROM, FEPRM, ROM devices, and external peripherals. The GPCM contains two basic configuration register groups—BR<sub>x</sub> and OR<sub>x</sub>.

Figure 21-34 shows a simple connection between an 24-bit port size SRAM device and the EMC in GPCM mode. Write-enable signals ( $\overline{LWE}$ ) are available for entire 24-bit word written to memory. Also, the output enable signal ( $\overline{LOE}$ ) is provided to minimize external glue logic. On system reset, a global (boot) chip-select is available that provides a boot ROM chip-select ( $\overline{LCS0}$ ) prior to the system being fully configured.

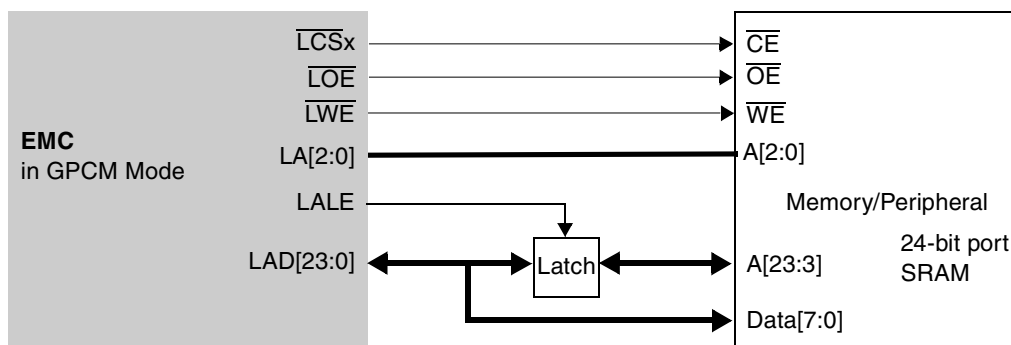


Figure 21-34. Local Bus to GPCM Device Interface

Figure 21-35 shows  $\overline{LCS}$  as defined by the setup time required between the address lines and  $\overline{CE}$ . You can configure OR<sub>x</sub>[ACS] to specify  $\overline{LCS}$  to meet this requirement.

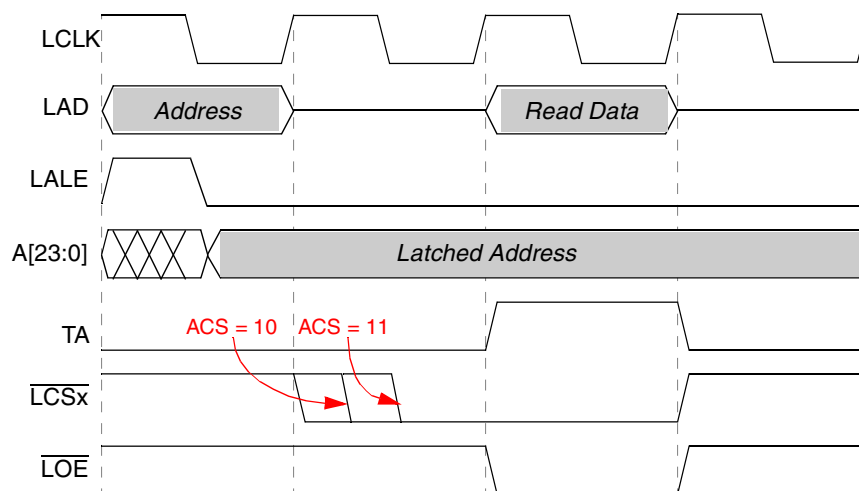


Figure 21-35. GPCM Basic Read Timing  
(XACS = 0, ACS = 1x, TRLX = 0, CLKDIV = 4,8)

### 21.4.2.1 Timing Configuration

If BR<sub>x</sub>[MSEL] selects the GPCM, the attributes for the memory cycle are taken from OR<sub>x</sub>. These attributes include the CSNT, ACS, XACS, SCY, TRLX, EHTR and SETA fields. [Table 21-35](#) shows signal behavior and system response for a write access with CRR[CLKDIV] = 4 or CRR[CLKDIV] = 8.

[Table 21-36](#) shows the signal behavior and system response for a read access with CRR[CLKDIV] = 4 or CRR[CLKDIV] = 8 for both personalities. [Table 21-37](#) and [Table 21-38](#) show the write and read signal behavior respectively, when CRR[CLKDIV] = 2.

**Table 21-35. GPCM Write Control Signal Timing for CRR[CLKDIV] = 4 or 8**

Option Register Attributes				Signal Behavior (Bus Clock Cycles)			
TRLX	XACS	ACS	CSNT	Address to LCS <sub>x</sub> Asserted	$\overline{\text{LCS}}_x$ Negated to Address Change	$\overline{\text{LWE}}$ Negated to Address and Data Invalid	Total Cycles <sup>1</sup>
0	0	00	0	0	0	0	3 + SCY
0	0	10	0	1/4	0	0	3 + SCY
0	0	11	0	1/2	0	0	3 + SCY
0	1	00	0	0	0	0	3 + SCY
0	1	10	0	1	0	0	3 + SCY
0	1	11	0	2	0	0	4 + SCY
0	0	00	1	0	0	-1/4	3 + SCY
0	0	10	1	1/4	-1/4	-1/4	3 + SCY
0	0	11	1	1/2	-1/4	-1/4	3 + SCY
0	1	00	1	0	0	-1/4	3 + SCY
0	1	10	1	1	-1/4	-1/4	3 + SCY
0	1	11	1	2	-1/4	-1/4	4 + SCY
1	0	00	0	0	0	0	3 + 2 × SCY
1	0	10	0	1 + 1/4	0	0	4 + 2 × SCY
1	0	11	0	1 + 1/2	0	0	4 + 2 × SCY
1	1	00	0	0	0	0	3 + 2 × SCY
1	1	10	0	2	0	0	4 + 2 × SCY
1	1	11	0	3	0	0	5 + 2 × SCY
1	0	00	1	0	0	-1 - 1/4	4 + 2 × SCY
1	0	10	1	1 + 1/4	-1 - 1/4	-1 - 1/4	5 + 2 × SCY
1	0	11	1	1 + 1/2	-1 - 1/4	-1 - 1/4	5 + 2 × SCY
1	1	00	1	0	0	-1 - 1/4	4 + 2 × SCY

**Table 21-35. GPCM Write Control Signal Timing for CRR[CLKDIV] = 4 or 8 (Continued)**

Option Register Attributes				Signal Behavior (Bus Clock Cycles)			
TRLX	XACS	ACS	CSNT	Address to $\overline{\text{LCSx}}$ Asserted	$\overline{\text{LCSx}}$ Negated to Address Change	$\overline{\text{LWE}}$ Negated to Address and Data Invalid	Total Cycles <sup>1</sup>
1	1	10	1	2	-1 - 1/4	-1 - 1/4	5 + 2 × SCY
1	1	11	1	3	-1 - 1/4	-1 - 1/4	6 + 2 × SCY

<sup>1</sup> Total cycles when LALE is asserted for one cycle only (ORx[EAD] = 0; ORx[EAD] = 1 and CRR[EADC] = 01). Asserting LALE for more than one cycle increases the total cycle count accordingly.

**Table 21-36. GPCM Read Control Signal Timing for CRR[CLKDIV] = 4 or 8**

Option Register Attributes				Signal Behavior (Bus Clock Cycles)		
TRLX	EHTR	XACS	ACS	Address to $\overline{\text{LCSx}}$ Asserted	$\overline{\text{LCSx}}$ Negated to Address Change	Total Cycles <sup>1</sup>
0	0	0	00	0	1	4 + SCY
0	0	0	10	1/4	1	4 + SCY
0	0	0	11	1/2	1	4 + SCY
0	0	1	00	0	1	4 + SCY
0	0	1	10	1	1	4 + SCY
0	0	1	11	2	1	5 + SCY
0	1	0	00	0	2	5 + SCY
0	1	0	10	1/4	2	5 + SCY
0	1	0	11	1/2	2	5 + SCY
0	1	1	00	0	2	5 + SCY
0	1	1	10	1	2	5 + SCY
0	1	1	11	2	2	6 + SCY
1	0	0	00	0	5	8 + 2 × SCY
1	0	0	10	1 + 1/4	5	9 + 2 × SCY
1	0	0	11	1 + 1/2	5	9 + 2 × SCY
1	0	1	00	0	5	8 + 2 × SCY
1	0	1	10	2	5	9 + 2 × SCY
1	0	1	11	3	5	10 + 2 × SCY
1	1	0	00	0	9	12 + 2 × SCY
1	1	0	10	1 + 1/4	9	13 + 2 × SCY

**Table 21-36. GPCM Read Control Signal Timing for CRR[CLKDIV] = 4 or 8 (Continued)**

Option Register Attributes				Signal Behavior (Bus Clock Cycles)		
TRLX	EHTR	XACS	ACS	Address to $\overline{\text{LCSx}}$ Asserted	$\overline{\text{LCSx}}$ Negated to Address Change	Total Cycles <sup>1</sup>
1	1	0	11	1 + 1/2	9	13 + 2 × SCY
1	1	1	00	0	9	12 + 2 × SCY
1	1	1	10	2	9	13 + 2 × SCY
1	1	1	11	3	9	14 + 2 × SCY

<sup>1</sup> Total cycles when LALE is asserted for one cycle only (ORx[EAD] = 0; ORx[EAD] = 1 and CRR[EADC] = 01). Asserting LALE for more than one cycle increases the total cycle count accordingly.

**Table 21-37. GPCM Write Control Signal Timing for CRR[CLKDIV] = 2**

Option Register Attributes				Signal Behavior (Bus Clock Cycles)			
TRLX	XACS	ACS	CSNT	Address to $\overline{\text{LCSx}}$ Asserted	$\overline{\text{LCSx}}$ Negated to Address Change	$\overline{\text{LWE}}$ Negated to Address and Data Invalid	Total Cycles <sup>1</sup>
0	0	00	0	0	0	0	3 + SCY
0	0	10	0	1/2	0	0	3 + SCY
0	0	11	0	1/2	0	0	3 + SCY
0	1	00	0	0	0	0	3 + SCY
0	1	10	0	1	0	0	3 + SCY
0	1	11	0	2	0	0	4 + SCY
0	0	00	1	0	0	0	3 + SCY
0	0	10	1	1/2	0	0	3 + SCY
0	0	11	1	1/2	0	0	3 + SCY
0	1	00	1	0	0	0	3 + SCY
0	1	10	1	1	0	0	3 + SCY
0	1	11	1	2	0	0	4 + SCY
1	0	00	0	0	0	0	3 + 2 × SCY
1	0	10	0	1 + 1/2	0	0	4 + 2 × SCY
1	0	11	0	1 + 1/2	0	0	4 + 2 × SCY
1	1	00	0	0	0	0	3 + 2 × SCY
1	1	10	0	2	0	0	4 + 2 × SCY
1	1	11	0	3	0	0	5 + 2 × SCY
1	0	00	1	0	0	-1	4 + 2 × SCY

**Table 21-37. GPCM Write Control Signal Timing for CRR[CLKDIV] = 2 (Continued)**

Option Register Attributes				Signal Behavior (Bus Clock Cycles)			
TRLX	XACS	ACS	CSNT	Address to $\overline{\text{LCSx}}$ Asserted	$\overline{\text{LCSx}}$ Negated to Address Change	$\overline{\text{LWE}}$ Negated to Address and Data Invalid	Total Cycles <sup>1</sup>
1	0	10	1	1 + 1/2	-1	-1	5 + 2 × SCY
1	0	11	1	1 + 1/2	-1	-1	5 + 2 × SCY
1	1	00	1	0	0	-1	4 + 2 × SCY
1	1	10	1	2	-1	-1	5 + 2 × SCY
1	1	11	1	3	-1	-1	6 + 2 × SCY

<sup>1</sup> Total cycles when LALE is asserted for one cycle only (ORx[EAD]=0; ORx[EAD]=1 and CRR[EADC]=01). Asserting LALE for more than one cycle increases the total cycle count accordingly.

**Table 21-38. GPCM Read Control Signal Timing for CRR[CLKDIV] = 2**

Option Register Attributes				Signal Behavior (Bus Clock Cycles)		
TRLX	EHTR	XACS	ACS	Address to $\overline{\text{LCSx}}$ Asserted	$\overline{\text{LCSx}}$ Negated to Address Change	Total Cycles <sup>1</sup>
0	0	0	00	0	1	4 + SCY
0	0	0	10	1/2	1	4 + SCY
0	0	0	11	1/2	1	4 + SCY
0	0	1	00	0	1	4 + SCY
0	0	1	10	1	1	4 + SCY
0	0	1	11	2	1	5 + SCY
0	1	0	00	0	2	5 + SCY
0	1	0	10	1/2	2	5 + SCY
0	1	0	11	1/2	2	5 + SCY
0	1	1	00	0	2	5 + SCY
0	1	1	10	1	2	5 + SCY
0	1	1	11	2	2	6 + SCY
1	0	0	00	0	5	8 + 2 × SCY
1	0	0	10	1 + 1/2	5	9 + 2 × SCY
1	0	0	11	1 + 1/2	5	9 + 2 × SCY
1	0	1	00	0	5	8 + 2 × SCY
1	0	1	10	2	5	9 + 2 × SCY
1	0	1	11	3	5	10 + 2 × SCY
1	1	0	00	0	9	12 + 2 × SCY

**Table 21-38. GPCM Read Control Signal Timing for CRR[CLKDIV] = 2 (Continued)**

Option Register Attributes				Signal Behavior (Bus Clock Cycles)		
TRLX	EHTR	XACS	ACS	Address to $\overline{\text{LCSx}}$ Asserted	$\overline{\text{LCSx}}$ Negated to Address Change	Total Cycles <sup>1</sup>
1	1	0	10	1 + 1/2	9	13 + 2 × SCY
1	1	0	11	1 + 1/2	9	13 + 2 × SCY
1	1	1	00	0	9	12 + 2 × SCY
1	1	1	10	2	9	13 + 2 × SCY
1	1	1	11	3	9	14 + 2 × SCY

<sup>1</sup> Total cycles when LALE is asserted for one cycle only (ORx[EAD]=0; ORx[EAD]=1 and CRR[EADC]=01). Asserting LALE for more than one cycle increases the total cycle count accordingly.

### 21.4.2.2 Chip-Select Assertion Timing

The banks (selected to work with the GPCM) support an option to drive the  $\overline{\text{LCSx}}$  signal with different timings (with respect to the external address and data bus).  $\overline{\text{LCSx}}$  can be driven in any of the following ways:

- Simultaneously with the latched memory address. (Refers to the externally latched address and not the address timing on LAD[23:0]. In other words, the chip select does not assert during LALE).
- One quarter of a clock cycle later (for CRR[CLKDIV] = 4, 8).
- One half of a clock cycle later (for CRR[CLKDIV] = 2, 4, or 8).
- One clock cycle later (for CRR[CLKDIV] = 4), when ORx[XACS] = 1.
- One clock cycle later (for CRR[CLKDIV] = 2), when ORx[XACS] = 1
- Two clock cycles later (for CRR[CLKDIV] = 2, 4, or 8), when ORx[XACS] = 1.
- Three clock cycles later (for CRR[CLKDIV] = 2, 4, or 8), when ORx[XACS] = 1 and ORx[TRLX] = 1.

The timing diagram in [Figure 21-35](#) shows two chip-select assertion timings for the case CRR[CLKDIV] = 4 or 8. If CRR[CLKDIV] = 2,  $\overline{\text{LCSx}}$  asserts identically for ORx[ACS] = 10 or 11.

#### 21.4.2.2.1 Programmable Wait State Configuration

The GPCM supports internal generation of transfer acknowledge. It allows between zero and 30 wait states to be added to an access by programming ORx[SCY] and ORx[TRLX]. Internal generation of transfer acknowledge is enabled if ORx[SETA] = 0. If  $\overline{\text{LGTA}}$  is asserted externally two bus clock cycles or more before the wait state counter has expired (to allow for synchronization latency), the current memory cycle is terminated by  $\overline{\text{LGTA}}$ ; otherwise the current memory cycle is terminated by the expiration of the wait state counter. Regardless of the setting of ORx[SETA], wait states prolong the assertion duration of both  $\overline{\text{LOE}}$  and  $\overline{\text{LWEx}}$  in the same manner. When TRLX = 1, the number of wait states inserted by the memory controller is doubled from ORx[SCY] cycles to 2 × ORx[SCY] cycles, allowing a maximum of 30 wait states.

### 21.4.2.2.2 Chip-Select and Write Enable Negation Timing

Figure 21-34 shows a basic connection between the EMC and a static memory device. In this case,  $\overline{\text{LCSx}}$  is connected directly to the  $\overline{\text{CE}}$  of the memory device. The  $\overline{\text{LWE}}$  signal is connected to the  $\overline{\text{WE}}$  signals on the memory device.

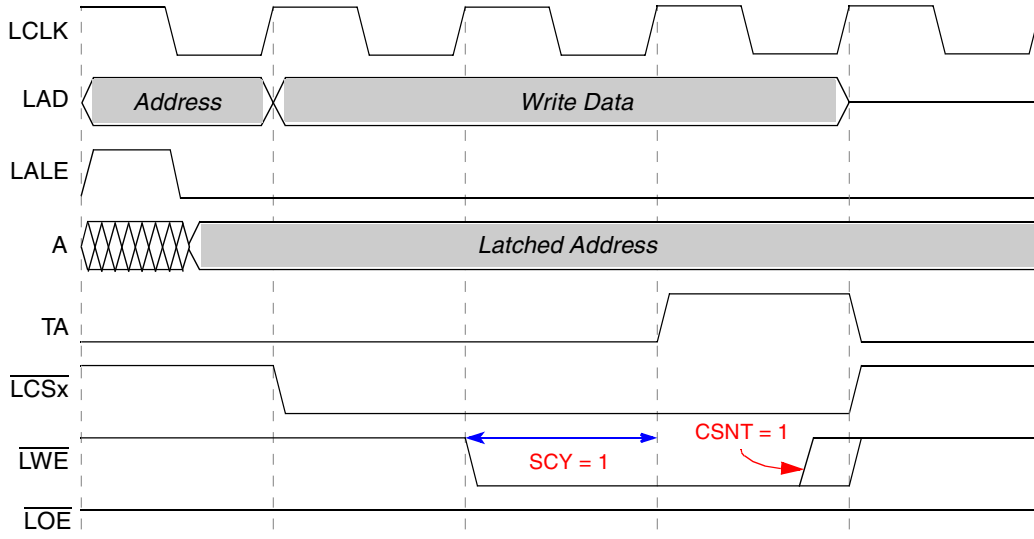


Figure 21-36. GPCM Basic Write Timing (XACS = 0, ACS = 00, CSNT = 1, SCY = 1, TRLX = 0, CLKDIV = 4, 8)

As Figure 21-36 shows, the timing for  $\overline{\text{LCSx}}$  is the same as for the latched address. The strobes for the transaction are supplied by  $\overline{\text{LOE}}$  or  $\overline{\text{LWE}}$ , depending on the transaction direction—read or write (the write case shown in Figure 21-36).  $\text{ORx}[\text{CSNT}]$  controls the timing for the appropriate strobe negation in write cycles. When this attribute is asserted, the strobe is negated one quarter of a clock before the normal case provided that  $\text{CRR}[\text{CLDIV}] = 4$  or  $8$ . For example, when  $\text{ACS} = 00$  and  $\text{CSNT} = 1$ ,  $\overline{\text{LWE}}$  is negated one quarter of a clock earlier, as shown in Figure 21-36. If  $\text{CRR}[\text{CLDIV}] = 2$ ,  $\overline{\text{LWE}}$  is negated either coincident with  $\overline{\text{LCSx}}$  or one cycle earlier.

### 21.4.2.2.3 Relaxed Timing

$\text{ORx}[\text{TRLX}]$  is provided for memory systems that require more relaxed timing between signals. Setting  $\text{TRLX} = 1$  has the following effect on timing:

- An additional bus cycle is added between the address and control signals (but only if ACS is not equal to 00).
- The number of wait states specified by SCY is doubled, providing up to 30 wait states.
- The extended hold time on read accesses (EHTR) is extended further.
- $\overline{\text{LCSx}}$  signals are negated one cycle earlier during writes (but only if ACS is not equal to 00).
- $\overline{\text{LWE}}$  signals are negated one cycle earlier during writes.



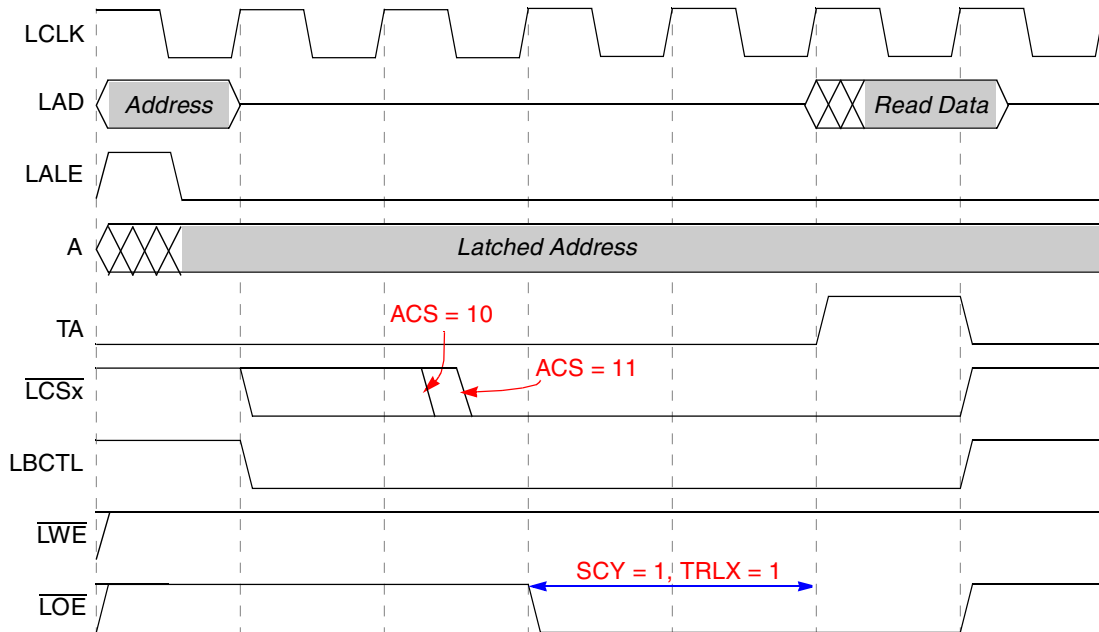


Figure 21-37. GPCM Relaxed Timing Read (XACS = 0, ACS = 1x, SCY = 1, CSNT = 0, TRLX = 1, CLKDIV = 4, 8)

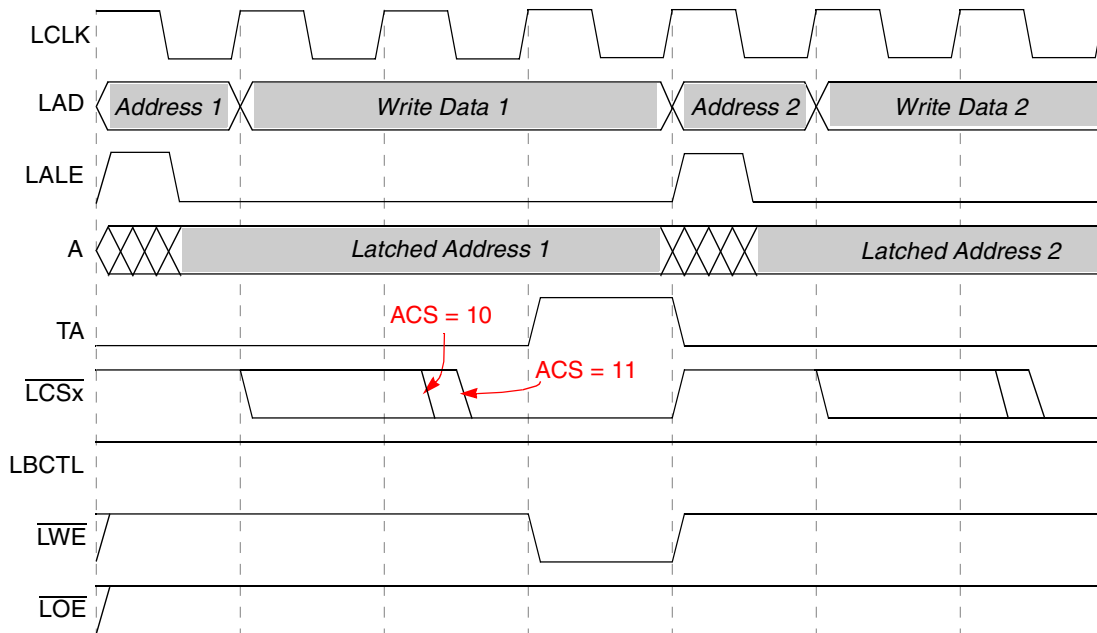


Figure 21-38. GPCM Relaxed Timing Back-to-Back Writes (XACS = 0, ACS = 1x, SCY = 0, CSNT = 0, TRLX = 1, CLKDIV = 4, 8)

Relaxed timing read and write transactions are illustrated in [Figure 21-37](#) and [Figure 21-38](#). The effect of CLKDIV = 2 for these examples is only to delay the assertion of  $\overline{\text{LCSx}}$  in the ACS = 10 case to the ACS = 11 case. The example in [Figure 21-38](#) also shows address and data multiplexing on LAD[23:0] for a pair of writes issued consecutively.

## External Memory Controller (EMC)

When TRLX and CSNT are set in a write access, the  $\overline{\text{LWE}}$  strobe signals are negated one clock earlier than in the normal case, as shown in Figure 21-39 and Figure 21-40. If  $\text{ACS} \neq 00$ ,  $\overline{\text{LCSx}}$  is also negated one clock earlier.

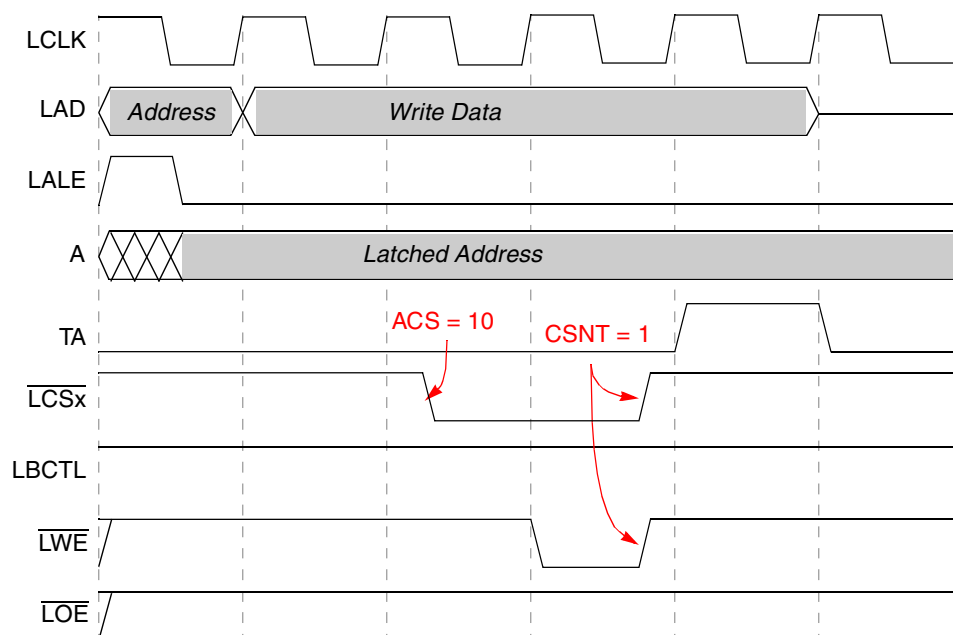


Figure 21-39. GPCM Relaxed Timing Write (XACS = 0, ACS = 10, SCY = 0, CSNT = 1, TRLX = 1, CLKDIV = 4, 8)

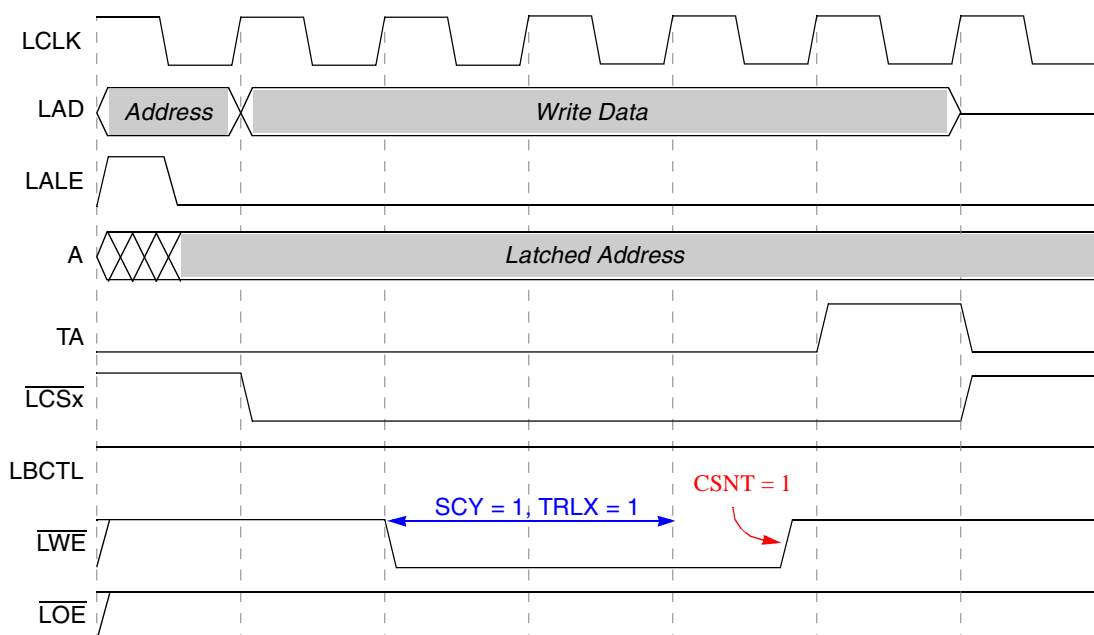


Figure 21-40. GPCM Relaxed Timing Write (XACS = 0, ACS = 00, SCY = 1, CSNT = 1, TRLX = 1, CLKDIV = 4, 8)

#### 21.4.2.2.4 Output Enable ( $\overline{\text{LOE}}$ ) Timing

The timing of the  $\overline{\text{LOE}}$  is affected only by  $\text{TRLX}$ . It always asserts and negates on the rising edge of the bus clock.  $\overline{\text{LOE}}$  asserts either on the rising edge of the bus clock after  $\overline{\text{LCSx}}$  is asserted or coinciding with  $\overline{\text{LCSx}}$  (if  $\text{XACS} = 1$  and  $\text{ACS} = 10$  or  $\text{ACS} = 11$ ). Accordingly, assertion of  $\overline{\text{LOE}}$  can be delayed (along with the assertion of  $\overline{\text{LCSx}}$ ) by programming  $\text{TRLX} = 1$ .  $\overline{\text{LOE}}$  negates on the rising clock edge coinciding with  $\overline{\text{LCSx}}$  negation.

#### 21.4.2.2.5 Extended Hold Time on Read Accesses

Slow memory devices that take a long time to disable their data bus drivers on read accesses should choose some combination of  $\text{ORx}[\text{TRLX}]$  and  $\text{ORx}[\text{EHTR}]$ . Any access following a read access to the slower memory bank is delayed by the number of clock cycles specified in [Table 21-5](#) and [Table 21-8](#), in addition to any existing bus turnaround cycle. The final bus turnaround cycle is automatically inserted by EMC for reads, regardless of the setting of  $\text{ORx}[\text{EHTR}]$ .

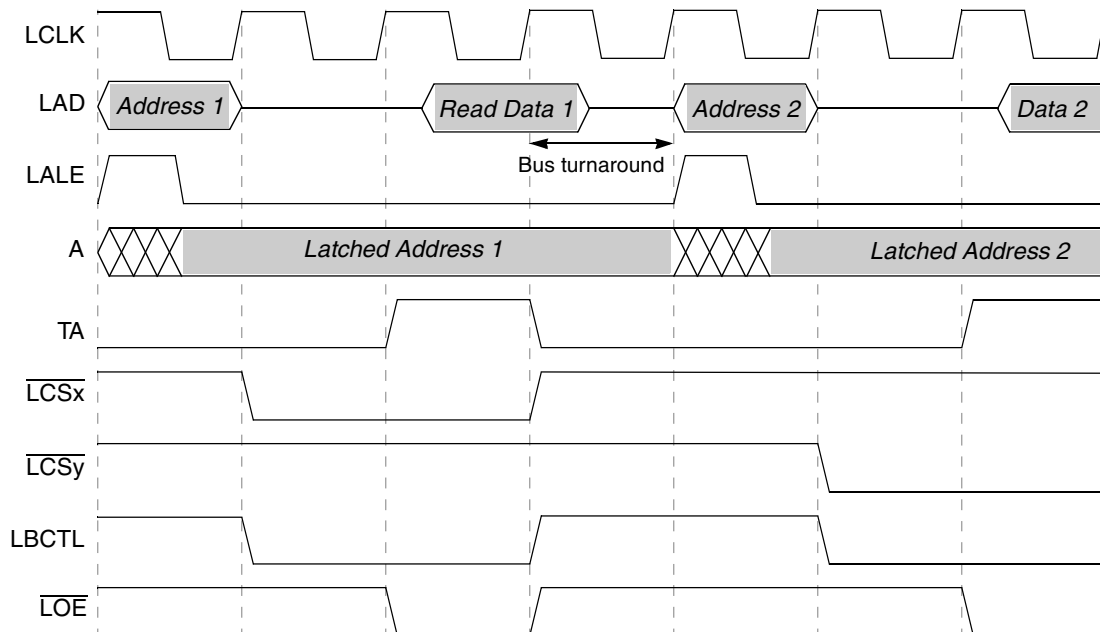


Figure 21-41. GPCM Read Followed by Read ( $\text{TRLX} = 0$ ,  $\text{EHTR} = 0$ , Fastest Timing)

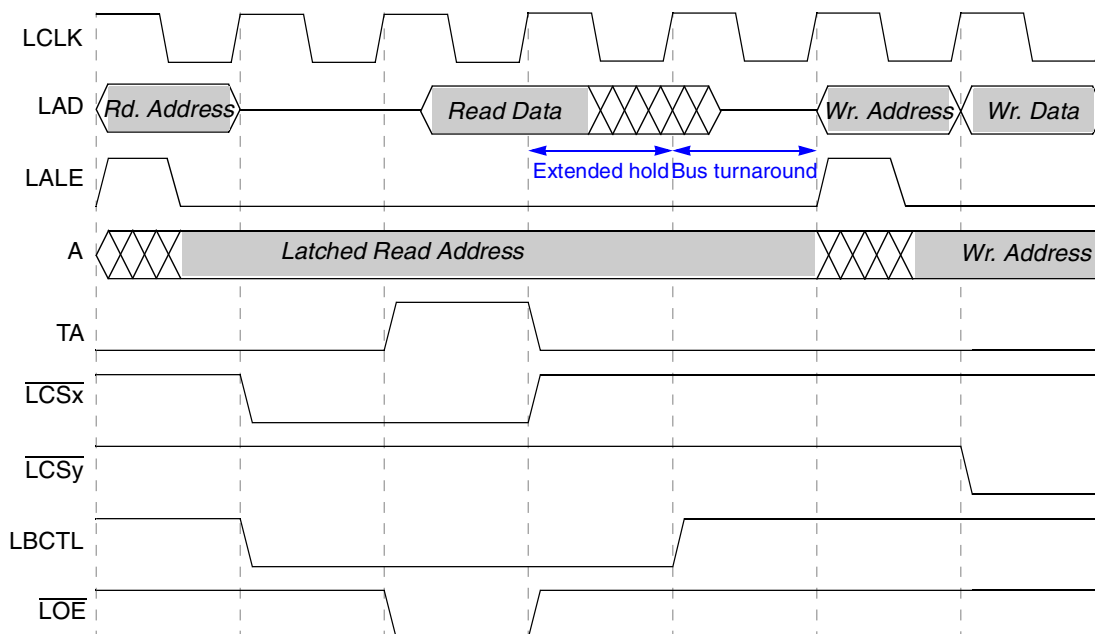


Figure 21-42. GPCM Read Followed by Write (TRLX = 0, EHTR = 1, One-Cycle Extended Hold Time on Reads)

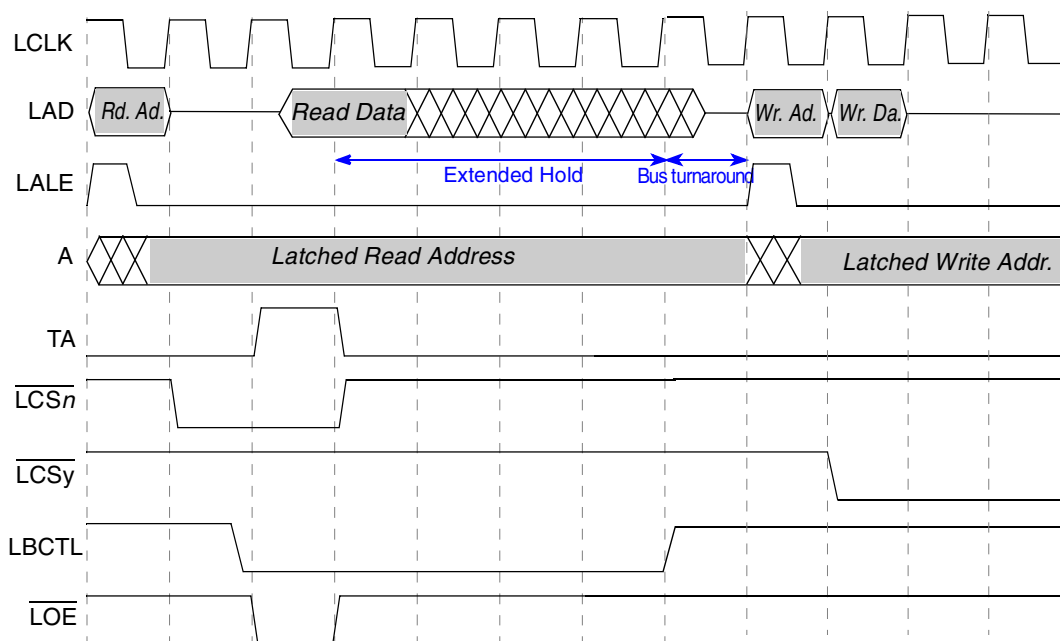


Figure 21-43. GPCM Read Followed by Write (TRLX = 1, EHTR = 0, Four-Cycle Extended Hold Time on Reads)

### 21.4.2.3 External Access Termination (LGTA)

External access termination is supported by the GPCM using the asynchronous  $\overline{\text{LGTA}}$  input signal, which is synchronized and sampled internally by the external memory controller. If, during assertion of  $\overline{\text{LCSx}}$ ,

the sampled  $\overline{\text{LGTA}}$  signal is asserted, it is converted to an internal generation of transfer acknowledge, which terminates the current GPCM access (regardless of the setting of  $\text{OR}_x[\text{SETA}]$ ).  $\overline{\text{LGTA}}$  should be asserted for at least one bus cycle to be effective. Note that because  $\overline{\text{LGTA}}$  is synchronized, bus termination occurs two cycles after  $\overline{\text{LGTA}}$  assertion, so in case of read cycle, the device still must drive data as long as  $\overline{\text{LOE}}$  is asserted.

The user selects whether transfer acknowledge is generated internally or externally ( $\overline{\text{LGTA}}$ ) by programming  $\text{OR}_x[\text{SETA}]$ . Asserting  $\overline{\text{LGTA}}$  always terminates an access, even if  $\text{OR}_x[\text{SETA}] = 0$  (internal transfer acknowledge generation), but it is the only means by which an access can be terminated if  $\text{OR}_x[\text{SETA}] = 1$ . The timing of  $\overline{\text{LGTA}}$  is illustrated by the example in [Figure 21-44](#).

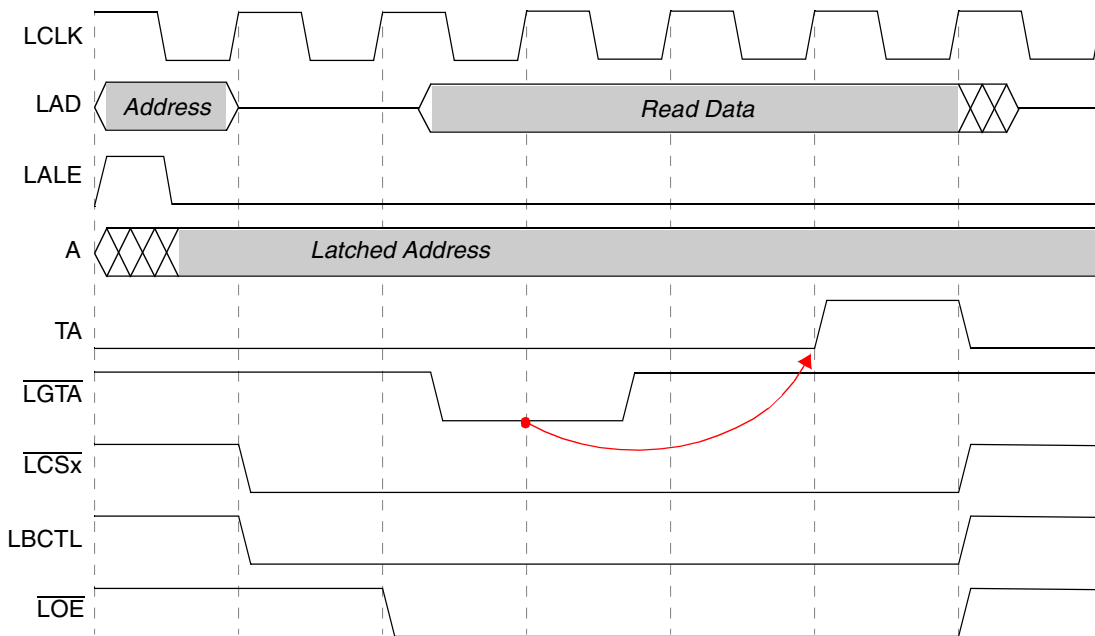


Figure 21-44. External Termination of GPCM Access

#### 21.4.2.4 Boot Chip-Select Operation

Boot chip-select operation allows address decoding for a boot ROM before system initialization. The  $\overline{\text{LCS0}}$  signal is the boot *chip select* output; its operation differs from the other external chip select outputs after a system reset. When the core begins accessing memory after system reset,  $\overline{\text{LCS0}}$  is asserted for every EMC access until  $\text{BR0}$  or  $\text{OR0}$  is reconfigured.

The boot chip select also provides a programmable port size (byte boot or word boot), which is configured by boot mode pins during reset. The boot chip-select does not provide write protection.  $\overline{\text{LCS0}}$  operates this way until the first write to  $\text{OR0}$ , and it can be used as any other chip-select register after the preferred address range is loaded into  $\text{BR0}$ . [Table 21-39](#) describes the initial values of the boot bank in the memory controller. Note that if you want to use  $\text{LCS1-7}$ ,  $\text{BR0}[\text{V}]$  should be written with 0 to disable the memory bank 0.

**Table 21-39. Boot Bank Field Values After Reset**

Register	Field	Setting
BR0	BA	00000000000
	XBA	00
	WP	0
	MSEL	000
	V	1
OR0	AM	00000000000
	XAM	00
	BCTLD	0
	CSNT	1
	ACS	11
	XACS	1
	SCY	1111
	SETA	0
	TRLX	1
	EHTR	1
	EAD	1

### 21.4.3 SDRAM Machine

The EMC provides an SDRAM interface (machine) for the external memory. The SDRAM machine provides the control functions and signals for JEDEC-compliant SDRAM devices. Each bank can control a SDRAM device on the EMC.

#### 21.4.3.1 Supported SDRAM Configurations

The memory controller supports any SDRAM configuration with the restriction that all SDRAM devices that reside on the bus should have the same timing parameters (as defined in SDMR). [Figure 21-45](#) shows an example connection between the EMC and a 24-bit SDRAM device with 12 address lines. Address signals A[2:0] of the SDRAM connect directly to LA[2:0], address pin A10 connects to the EMC's dedicated LSDA10 signal, while the remaining address bits (except A10) are latched from LAD[11:3].

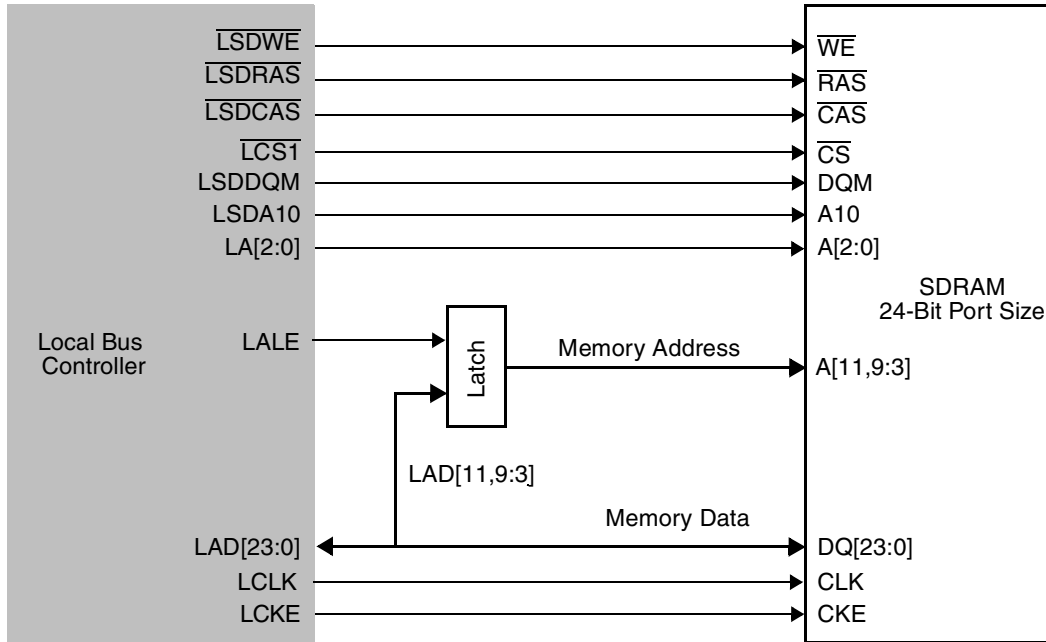


Figure 21-45. Connection to a 24-Bit SDRAM with 12 Address Lines

### 21.4.3.2 SDRAM Power-On Initialization

Following a system reset, initialization software must set up the programmable parameters in the memory controller banks registers (OR<sub>x</sub>, BR<sub>x</sub>, SDMR). After all memory parameters are configured, system software should execute the following initialization sequence for each SDRAM device according to its data sheet. A typical sequence is as follows:

- Issue a PRECHARGE-ALL-BANKS command
- Issue eight AUTO-REFRESH commands
- Issue a MODE-SET command to initialize the mode register

The initial commands are executed by setting SDMR[OP] and accessing the SDRAM with any read or write that hits the relevant bank.

Note that software should ensure that no memory operations begin until this process completes.

### 21.4.3.3 JEDEC-Standard SDRAM Interface Commands

The SDRAM machine performs all accesses to SDRAM by using JEDEC-standard SDRAM interface commands. The SDRAM device samples the command and data inputs on the rising edge of the bus clock. Data at the output of the SDRAM device is sampled on the rising edge of the bus clock.

## External Memory Controller (EMC)

The SDRAM interface commands shown in [Table 21-40](#) are provided by setting `SDMR[OP]` to a nonzero value (`SDMR[OP] = 000` sets normal read/write operation).

**Table 21-40. SDRAM Interface Commands**

(SDMR[OP])	Command	Description
110	ACTIVATE	Latches the row address and initiates a memory read of that row. Row data is latched in SDRAM sense amplifiers and must be restored with a PRECHARGE command before another ACTIVATE is issued.
011	MODE-SET	Allows setting of SDRAM options—CAS latency and burst length. CAS latency depends on the SDRAM device used. Although some SDRAMs provide burst lengths of 1, 2, 4, 8, or a page, the EMC supports only 8-beat bursts. The EMC does not support burst lengths of 1, 2 and a page for SDRAMs. The mode register data (CAS latency and burst length) is programmed into the SDMR register by initialization software after reset. After the SDMR is set, the EMC transfers the information to the SDRAM device by issuing a MODE-SET command.
100: single bank 101: all-banks	PRECHARGE	Restores data from the sense amplifiers to the appropriate row in the SDRAM device array. Also initializes the sense amplifiers to prepare for activating another row in the SDRAM device. Note that the EMC uses LSDA10 to distinguish between PRECHARGE-ALL-BANKS (LSDA10 is high) and PRECHARGE-SINGLE-BANK (LSDA10 is low). The SDRAMs must be compatible with this format.
111	READ	Latches the column address and transfers data from the selected sense amplifier on the SDRAM device, to the output buffer as determined by the column address. During each successive clock, additional data is driven without additional read commands. At the end of the burst, the page remains open. Burst length is the one set for this bank. Read data is discarded by the EMC.
111	WRITE	Latches the column address and transfers data from the data signals to the selected sense amplifier on the SDRAM device, as determined by the column address. During each successive clock, additional data is transferred to the sense amplifiers from the data signals without additional write commands. At the end of the burst, the page remains open. Burst length is the one set for this bank. LSDDQM are inactive and write data is undefined.
001	AUTO-REFRESH	Causes a row to be read in all memory banks as determined by the refresh row address counter. The refresh row address counter is internal to the SDRAM device. After being read, a row is automatically rewritten into the memory array. All banks must be in a precharged state before executing refresh.
010	SELF-REFRESH	Allows data to be retained in the SDRAM device, even when the rest of the EMC is in a power-saving mode with the clocks turned off. When placed in this mode, the SDRAM device is capable of issuing its own refresh commands, without external clocking from the EMC, and the LCKE pin from EMC is deasserted. This command can be issued at any time. Normal operation can be resumed only by setting <code>SDMR[OP] = 000</code> , and waiting a minimum of 200 bus cycles before issuing reads or writes to EMC.

### 21.4.3.4 Page Hit Checking

The SDRAM machine supports page-mode operation. Each time a page is activated on the SDRAM device, the SDRAM machine stores its address in a page register. The page information, which the user writes to the ORx register, is used along with the bank size to compare page bits of the address to the page register each time a bus-cycle access is requested. If a match is found, together with a bank match, the bus



cycle is defined as a page hit. An open page is automatically closed by the SDRAM machine if the bus becomes idle, unless  $ORx[PMSEL] = 1$ .

### 21.4.3.5 Page Management

The EMC is capable of managing a maximum of 4 open pages (one page per SDRAM bank) for a single SDRAM device. After a page is opened, the page remains open unless:

- the next access is to a page in a different SDRAM device, in which case all open pages on the current device are closed with a PRECHARGE-ALL-BANKS command.
- the next access is to a page in an SDRAM bank which has a different page open on it, in which case the old page is closed with a PRECHARGE-SINGLE-BANK command.
- the current SDRAM device requires refresh services, in which case all open pages on the current device are closed with a PRECHARGE-ALL-BANKS command.
- the bus becomes idle and  $ORx[PMSEL] = 0$ , in which case all open pages in the current device are closed with a PRECHARGE-ALL-BANKS command.

### 21.4.3.6 SDRAM Address Multiplexing

The lower bits of the address bus are connected to the memory device’s address port, with the memory controller multiplexing the row/column and the internal bank select lines. The position of the bank select lines are set according to  $SDMR[BSMA]$ . Figure 21-46 shows how the SDRAM controller shifts the row address down to the lower output address signals during activate and shifts the bank select bits up to the address pins specified by  $SDMR[BSMA]$ .

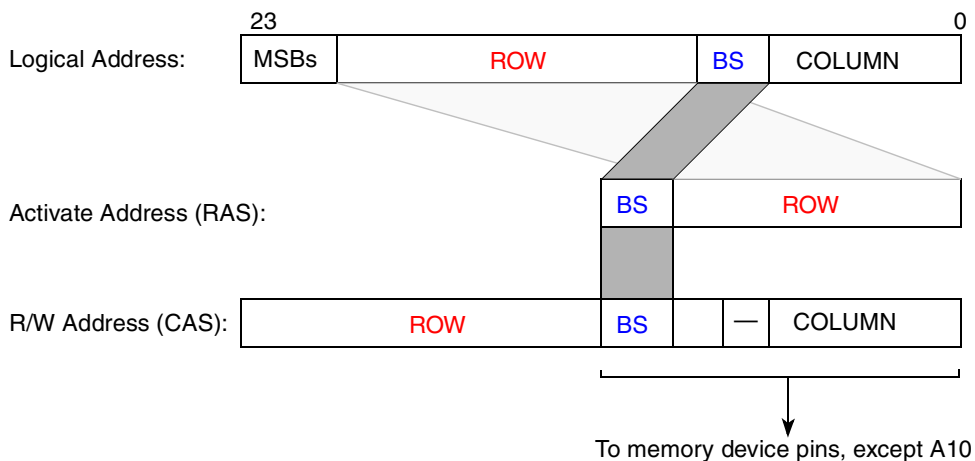


Figure 21-46. SDRAM Address Multiplexing

Note that during normal operation (read/write), a full 24-bit address which includes row and column, is generated on the  $LAD[23:0]$  signals. However, address and data signal multiplexing implies that the address must be latched by an external latch that is controlled by  $LALE$ . All SDRAM device address signals must be connected to the latched address bits and burst address bits ( $LA[2:0]$ ) of the EMC, with the exception of  $A10$ , which has a dedicated connection on  $LSDA10$ .  $LSDA10$  is driven with the appropriate row address bit for SDRAM commands that require  $A10$  to be an address.

### 21.4.3.7 SDRAM Device-Specific Parameters

The software is responsible for setting correct values for device-specific parameters that can be extracted from the device’s data sheet. The values are stored in the ORx and SDMR registers. These parameters include the following:

- Precharge-to-activate interval (SDMR[PRETOACT])
- Activate-to-read/write interval (SDMR[ACTTORW])
- CAS latency, column address to first-data-out (SDMR[CL] and CRR[ECL])
- Write recovery, last-data-in to precharge (SDMR[WRC])
- Refresh recovery interval (SDMR[RFRC])
- External buffers on the control lines present (SDMR[BUFCMD] and CRR[BUFCMDC])

In addition, the EMC hardware ensures a default activate to precharge interval of 10 bus cycles. The following sections describe SDRAM parameters programmed in SDMR.

#### 21.4.3.7.1 Precharge-to-Activate Interval

The *Precharge-to-Activate Interval* parameter, controlled by SDMR[PRETOACT] defines the earliest timing for an ACTIVATE or REFRESH command after a PRECHARGE command to the same SDRAM bank.

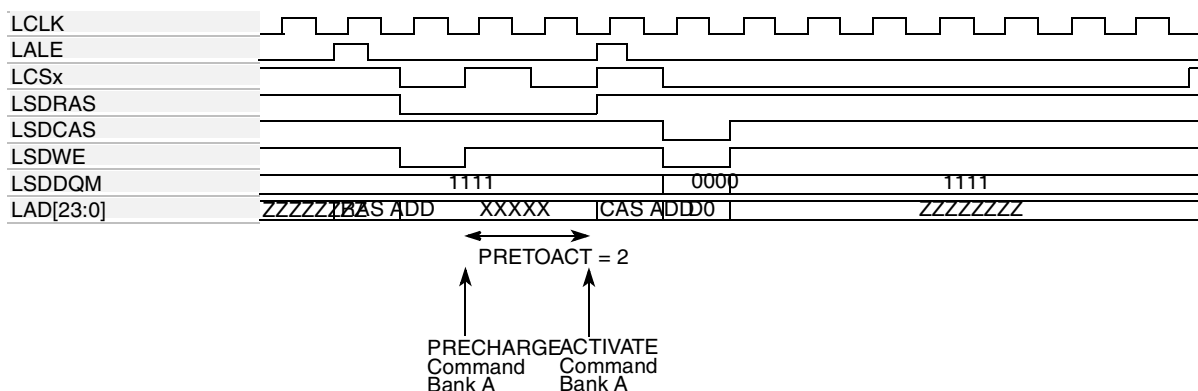


Figure 21-47. PRETOACT = 2 (2 Clock Cycles)

#### 21.4.3.7.2 Activate-to-Read/Write Interval

The *Activate-to-Read/Write Interval* parameter, controlled by SDMR[ACTTORW], defines the earliest timing for a READ/WRITE command after an ACTIVATE command to the same SDRAM bank.

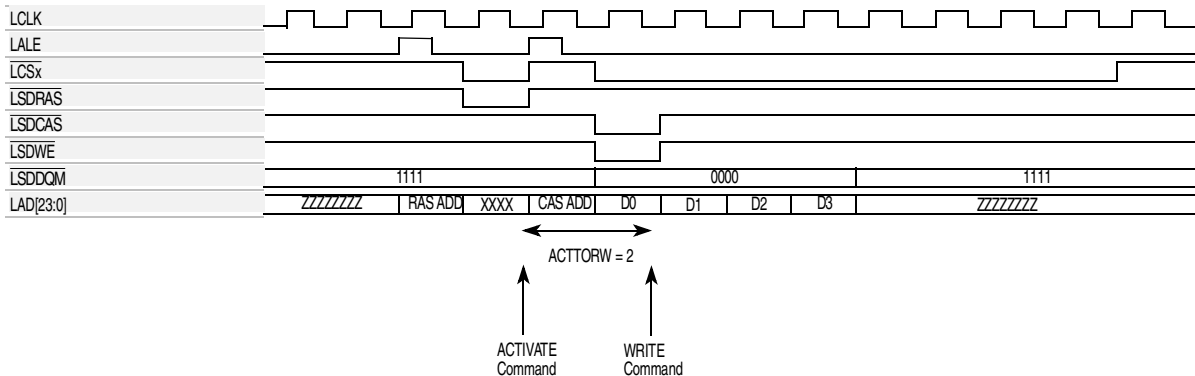


Figure 21-48. ACTTORW = 2 (2 Clock Cycles)

### 21.4.3.7.3 Column Address to First-Data-Out—CAS Latency

The *First-Data-Out* parameter, controlled by SDMR[CL] for latencies of 1, 2, or 3 and by CRR[ECL] for a latency of more than 3, defines the timing for first read data after a column address is sampled by the SDRAM.

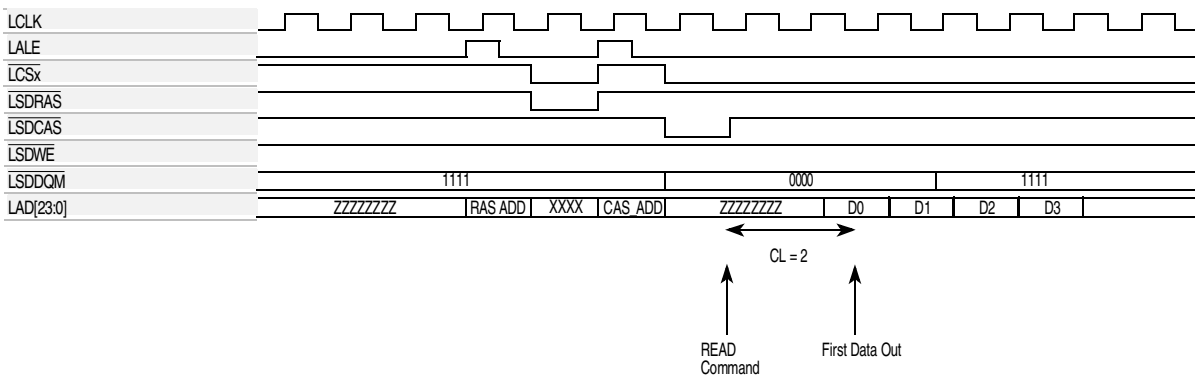


Figure 21-49. CL = 2 (2 Clock Cycles)

### 21.4.3.7.4 Last-Data-In-to-Precharge—Write Recovery

The *Last-Data-In-to-Precharge* parameter, controlled by SDMR[WRC], defines the earliest timing for a PRECHARGE command after the last data was written to the SDRAM.

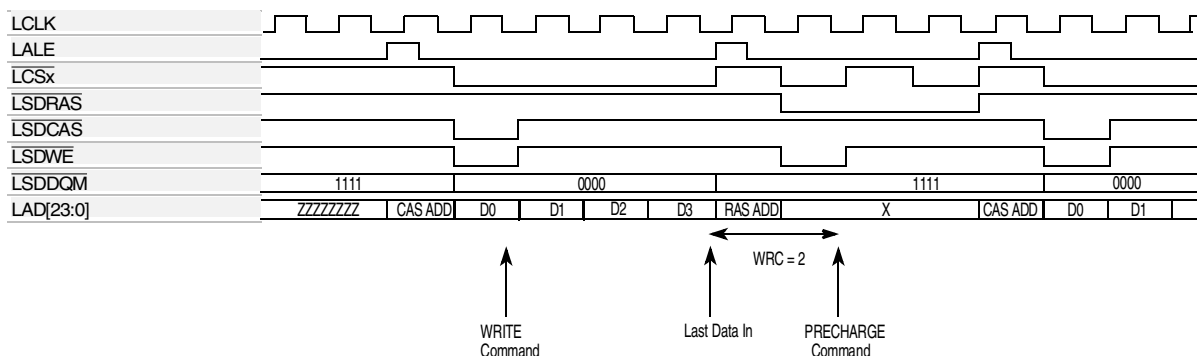


Figure 21-50. WRC = 2 (2 Clock Cycles)

### 21.4.3.7.5 Refresh Recovery Interval (RFRC)

The *Refresh Recovery Interval* parameter, controlled by `SDMR[RFRC]`, defines the earliest timing for an `ACTIVATE` or `REFRESH` command after a `REFRESH` command to the same SDRAM device.

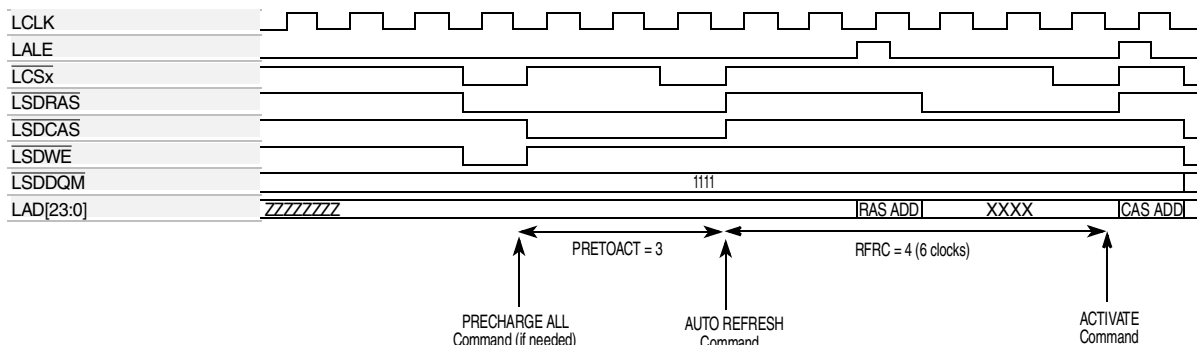


Figure 21-51. RFRC = 4 (6 Clock Cycles)

### 21.4.3.7.6 External Address and Command Buffers (BUFCMD)

If the additional delay of any buffers placed on the command strobes (`LSDRAS`, `LSDCAS`, `LSDWE`, and `LSDA10`), is endangering the device set-up time, `SDMR[BUFCMD]` should be set. Setting the `SDMR[BUFCMD]` bit causes the memory controller to add `CRR[BUFCMDC]` extra bus cycles to the assertion of SDRAM control signals (`LSDRAS`, `LSDCAS`, `LSDWE` and `LSDA10`) for each SDRAM command.

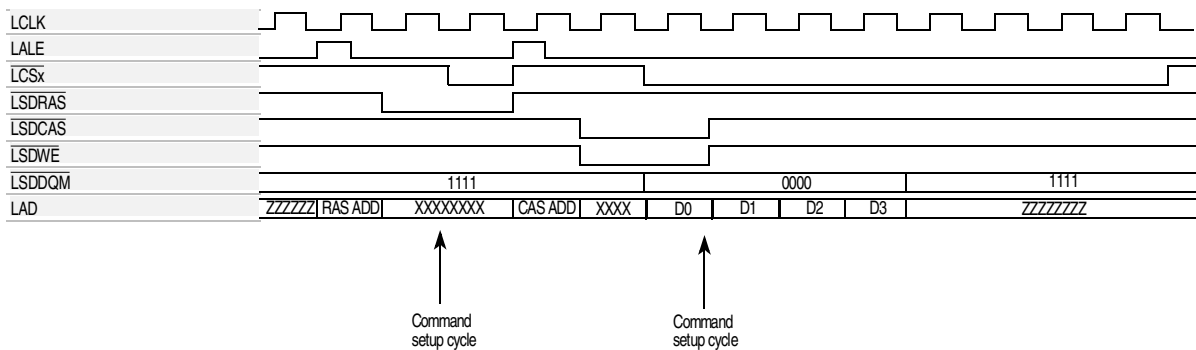


Figure 21-52. BUFCMD = 1, CRR[BUFCMDC] = 2

### 21.4.3.8 SDRAM Interface Timing

The following figures show SDRAM timing for various types of accesses.

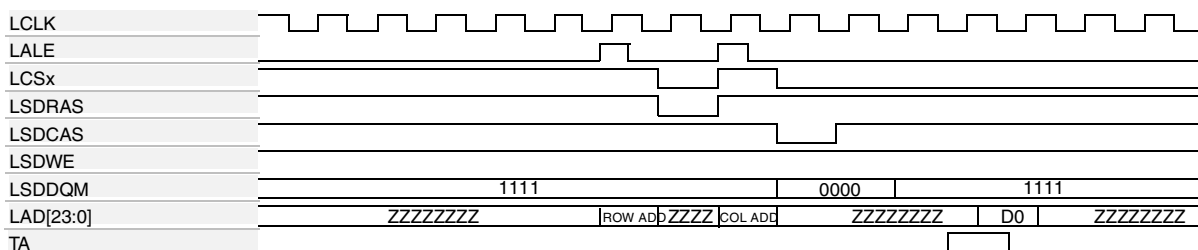


Figure 21-53. SDRAM Single-Beat Read, Page Closed, CL = 3

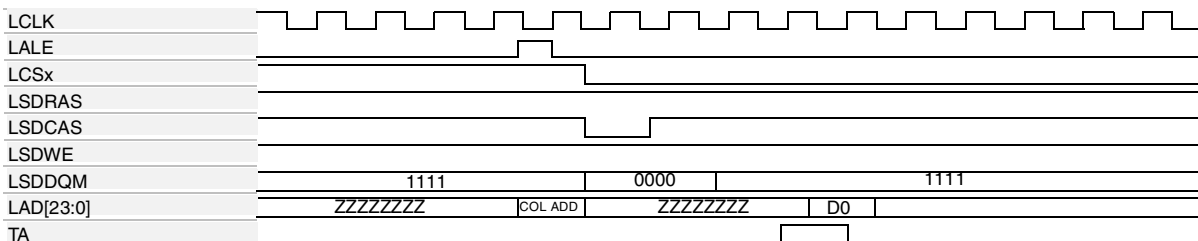


Figure 21-54. SDRAM Single-Beat Read, Page Hit, CL = 3

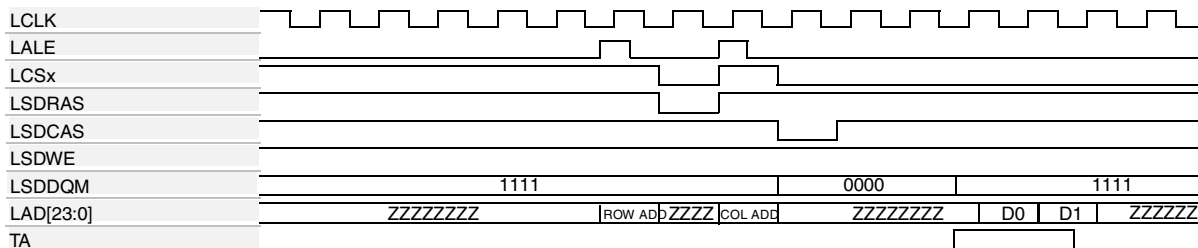


Figure 21-55. SDRAM Two-Beat Burst Read, Page Closed, CL = 3

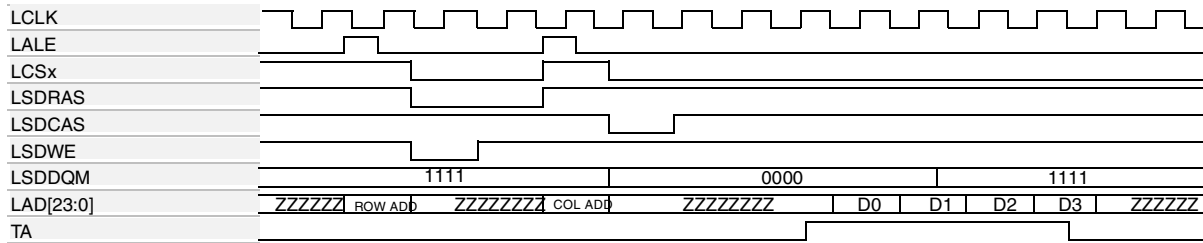


Figure 21-56. SDRAM Four-Beat Burst Read, Page Miss, CL = 3

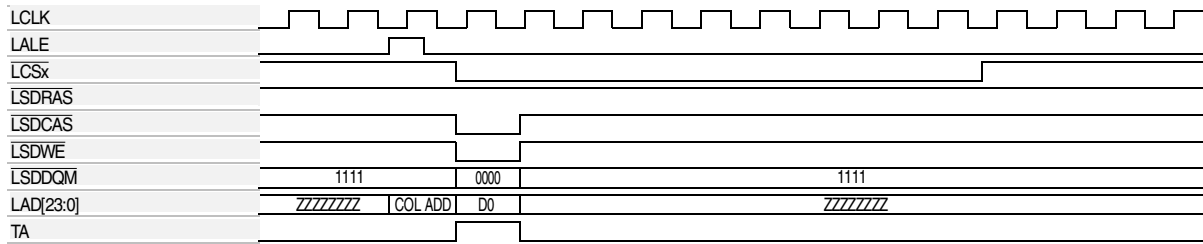


Figure 21-57. SDRAM Single Beat Write, Page Hit

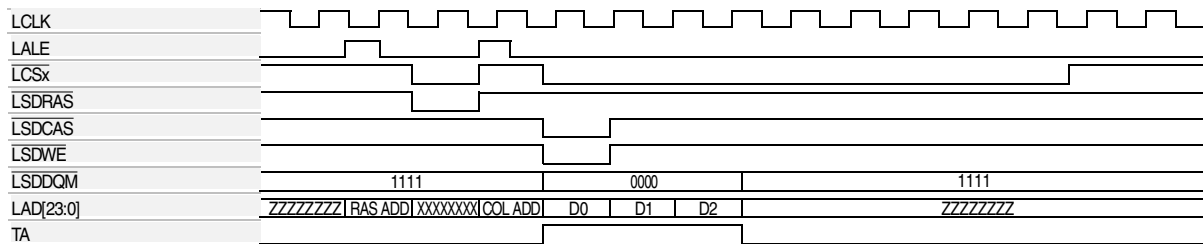


Figure 21-58. SDRAM Three-Beat Write, Page Closed

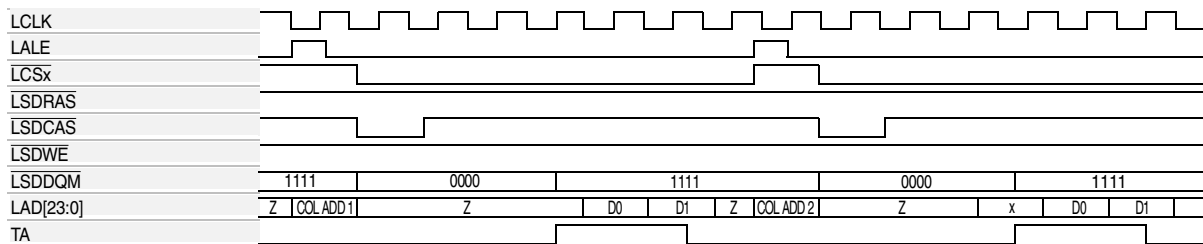


Figure 21-59. SDRAM Read-after-Read Pipelined, Page Hit, CL = 3

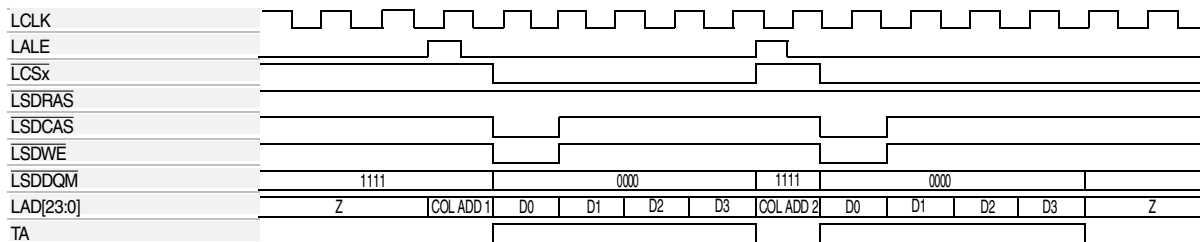


Figure 21-60. SDRAM Write-after-Write Pipelined, Page Hit

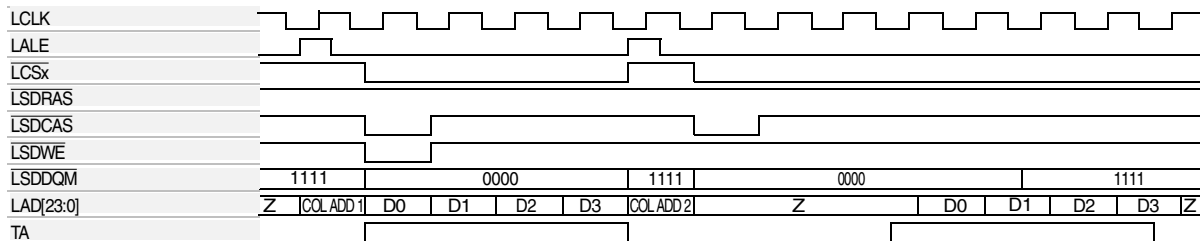


Figure 21-61. SDRAM Read-after-Write Pipelined, Page Hit

### 21.4.3.9 SDRAM Read/Write Transactions

The SDRAM interface supports read and write transactions of between 1 and 8 data beats for transaction sizes ranging from 1 to 8 words. For reads that require less than the full burst length, extraneous data in the burst is ignored and suppressed by the assertion of LSDDQM.

For writes that require less than the full burst length, the non-targeted addresses are protected by driving corresponding LSDDQM bits high (inactive) on the irrelevant cycles of the burst. However, system performance is not compromised because, if a new transaction is pending, the SDRAM controller begins executing it immediately, effectively terminating the burst early.

### 21.4.3.10 SDRAM MODE-SET Command Timing

The EMC transfers mode register data (CAS latency and burst length) stored in the SDMR register to the SDRAM device by issuing the MODE-SET command, as shown in Figure 21-62. In this case, the latched address carries the mode bits for the command.

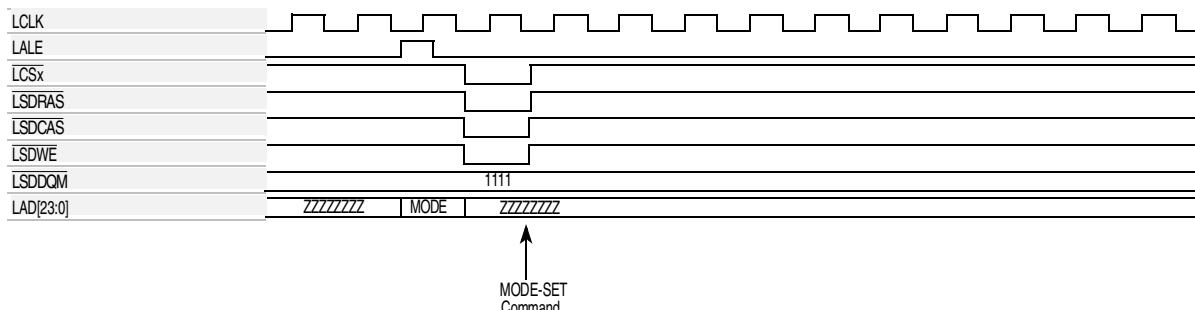


Figure 21-62. SDRAM MODE-SET Command

### 21.4.3.11 SDRAM Refresh

The memory controller supplies AUTO-REFRESH commands to any connected SDRAM device according to the interval specified in SRT (and prescaled by MRTPR[PTP]). This represents the time period required between refreshes. The values of SRT and MRTPR depend on the specific SDRAM devices used and the system clock frequency of the EMC. These values should allow for a potential collision between memory accesses and refresh cycles. The period of the refresh interval must be greater than the access time to ensure that read and write operations complete successfully.

There are two levels of refresh request priority—low and high. The low priority request is generated as soon as the refresh timer expires; this request is granted only if no other requests to the memory controller are pending. If the request is not granted (because the memory controller is busy) and the refresh timer expires two more times, the request becomes high priority and is served when the current memory controller operation finishes.

#### 21.4.3.11.1 SDRAM Refresh Timing

The SDRAM memory controller implements *bank staggering* for the auto refresh function. This reduces instantaneous current consumption for memory refresh operations.

After a refresh request is granted, the memory controller begins issuing an AUTO-REFRESH command to each device associated with the refresh timer. After a refresh command is issued to an SDRAM device, the memory controller waits for the number of bus clock cycles programmed in the SDRAM machine's mode register (SDMR[RFCR]) before issuing any subsequent ACTIVATE command to the same device. To avoid violating SDRAM device timing constraints, you should ensure that the refresh request interval (defined by SRT and MRTPR) is greater than the refresh recovery interval (defined by SDMR[RFCR]).



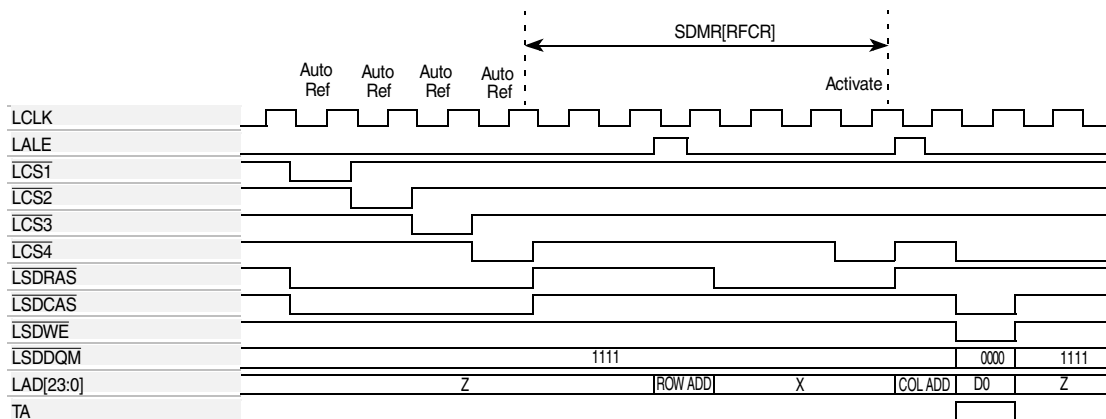


Figure 21-63. SDRAM Bank-Staggered Auto Refresh Timing

### 21.4.4 User-Programmable Machines (UPMs)

The user-programmable machines (UPMs) are flexible interfaces that connect to a wide range of memory devices. At the heart of each UPM is an internal RAM array that specifies the logical value driven on the external memory control signals ( $\overline{LCSx}$ , and  $\overline{LGPL}[5:0]$ ) for a given clock cycle. Each word in the RAM array provides bits that allow a memory access to be controlled with a resolution of up to one quarter of the external bus clock period on the chip-select lines. Figure 21-64 shows the basic operation of each UPM.

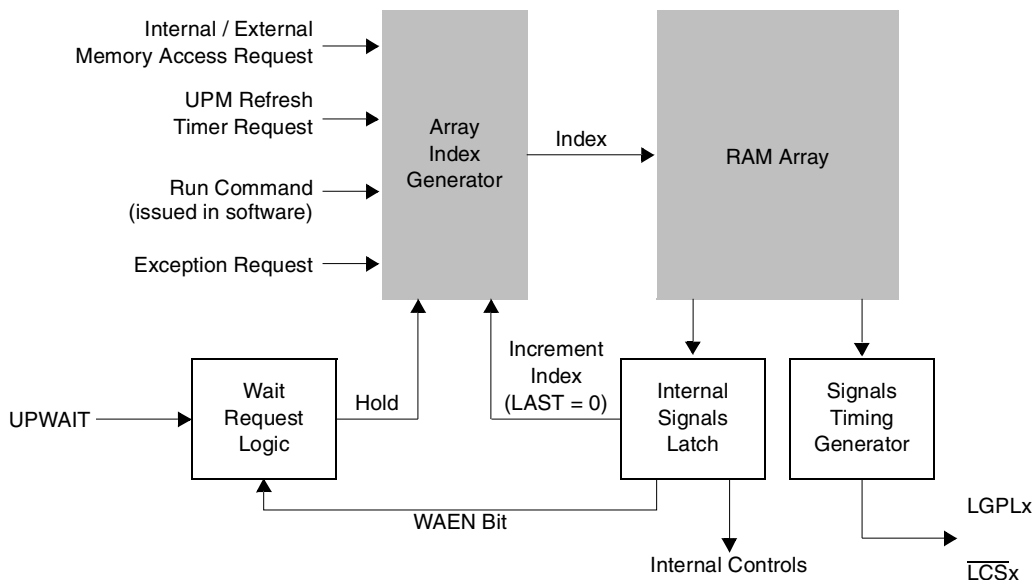


Figure 21-64. User-Programmable Machine Functional Block Diagram

The following events initiate a UPM cycle:

- Any internal device requests an external memory access to an address space mapped to a chip-select serviced by the UPM.
- A UPM refresh timer expires and requests a transaction, such as a DRAM refresh.

## External Memory Controller (EMC)

- A bus monitor time-out error during a normal UPM cycle redirects the UPM to execute an exception sequence.

The RAM array contains 64 words of 32 bits each. The signal timing generator loads the RAM word from the RAM array to drive the general-purpose lines and chip-selects. If the UPM reads a RAM word with WAEN set, the external UPWAIT signal is sampled and synchronized by the memory controller and the current request is frozen.

### 21.4.4.1 UPM Requests

A special pattern location in the RAM array is associated with each of the possible UPM requests. An internal device's request for a memory access initiates one of the following patterns ( $MxMR[OP] = 00$ ):

- Read single-beat pattern (RSS)
- Read burst cycle pattern (RBS)
- Write single-beat pattern (WSS)
- Write burst cycle pattern (WBS)

A UPM refresh timer request pattern initiates a refresh timer pattern (RTS).

An exception (caused by a bus monitor time-out error) occurs while another UPM pattern is running initiates an exception condition pattern (EXS).

Figure 21-65 and Table 21-41 show the start addresses of these patterns in the UPM RAM, according to cycle type. RUN commands ( $MxMR[OP] = 11$ ), however, can initiate patterns starting at any of the 64 UPM RAM words.

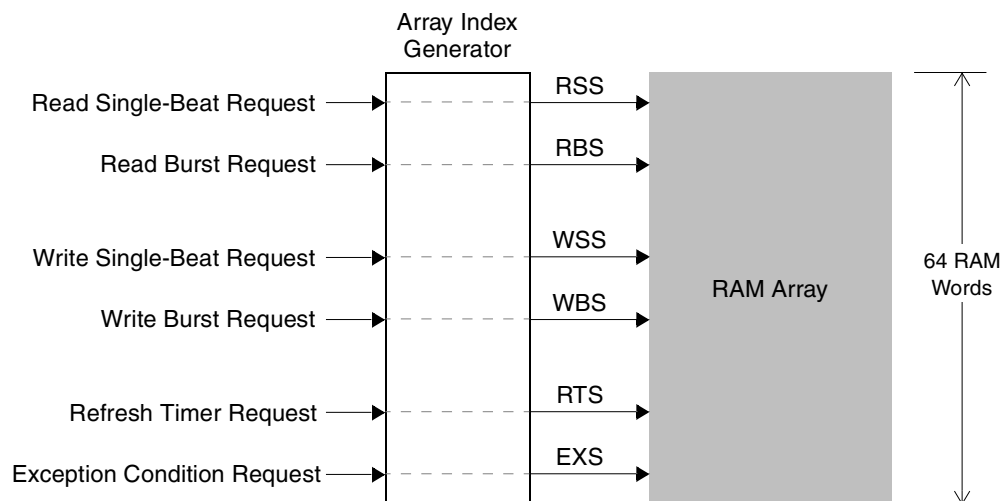


Figure 21-65. RAM Array Indexing

**Table 21-41. UPM Routines Start Addresses**

UPM Routine	Routine Start Address
Read single-beat (RSS)	0x00
Read burst (RBS)	0x08
Write single-beat (WSS)	0x18
Write burst (WBS)	0x20
Refresh timer (RTS)	0x30
Exception condition (EXS)	0x3C

### 21.4.4.1.1 Memory Access Requests

The user must ensure that the UPM is appropriately initialized before a request occurs.

The UPM supports two types of memory reads and writes:

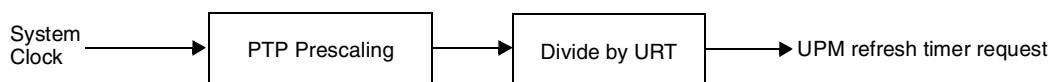
- A single-beat transfer transfers one operand consisting of up to a single word. A single-beat cycle starts with one transfer start and ends with one transfer acknowledge.
- A burst transfer transfers exactly four double words. The burst cycle starts with one transfer start but ends after eight transfer acknowledges.

Ensure that patterns for single-beat transfers contain one and only one transfer acknowledge (UTA bit in RAM word set high). For a burst transfer, the patterns must contain the exact number of transfer acknowledges required.

Any transfers that do not naturally fit single or burst transfers are synthesized as a series of single transfers. These accesses are treated by the UPM as back-to-back, single-beat transfers. Burst transfers can also be inhibited by setting ORx[BI]. Burst performance can be achieved by ensuring that UPM transactions are 8-word aligned with a transaction size being some multiple of 8-word, which is a natural fit for a cache-line transfer.

### 21.4.4.1.2 UPM Refresh Timer Requests

Each UPM contains a refresh timer that can be programmed to generate refresh service requests of a particular pattern in the RAM array. [Figure 21-66](#) shows the clock division hardware associated with memory refresh timer request generation. The UPM refresh timer register (URT) defines the period for the timers associated with all three UPMs.


**Figure 21-66. Memory Refresh Timer Request Block Diagram**

By default, all external memory refreshes are performed using the refresh pattern of UPMA. This means that if refresh is required, MAMR[RFEN] must be set (1). It also means that only one refresh routine should be programmed and be placed in UPMA, which serves as the refresh executor. Any banks assigned

## External Memory Controller (EMC)

to a UPM are provided with the refresh pattern, if the RFEN bit of the corresponding UPM is set. UPMA-assigned banks, therefore, always receive refresh services when MAMR[RFEN] is set, while UPMB- and UPMC-assigned banks also receive (the same) refresh services if the corresponding MxMR[RFEN] bits are set.

Note that the UPM refresh timer request should not be used in a system with SDRAM refresh enabled. The system designer must choose to use either SDRAM refresh or UPM refresh. Using both may result in missing refresh periods to memory.

### 21.4.4.1.3 Software Requests—RUN Command

Software can start a request to the UPM by issuing a RUN command to the UPM. Some memory devices have their own signal handshaking protocol to put them into special modes, such as self-refresh mode. Other memory devices require special commands to be issued on their control signals, such as for SDRAM initialization.

For these special cycles, you will create a special RAM pattern that can be stored in any unused areas in the UPM RAM; then the RUN command is used to run the cycle. The UPM runs the pattern beginning at the specified RAM location until it encounters a RAM word with its LAST bit set. The run command is issued by setting MxMR[OP] = 11 and accessing the UPMx memory region with any write transaction that hits the corresponding UPM machine. The starting address in the RAM array for the pattern is set in the MxMR[MAD] field.

Note that transfer acknowledges (UTA bit in the RAM word) are ignored for software (RUN command) requests, and afterwards the LAD signals remain high-impedance unless the normal initial LALE occurs or the RUN pattern causes assertion of LALE to occur on changes to the RAM word AMX field.

### 21.4.4.1.4 Exception Requests

When the EMC (under UPM control) initiates an access to a memory device and an exception occurs (bus monitor time-out), the UPM provides a mechanism by which memory control signals can meet the timing requirements of the device, without losing data. The mechanism is the exception pattern that defines how the UPM negates its signals in a controlled manner.

### 21.4.4.2 Programming the UPMs

The UPM is a microsequencer that requires micro-instructions or RAM words to generate signal timings for different memory cycles. To program the UPMs, follow these steps:

1. Set up the BRx and ORx registers.
2. Write patterns into the RAM array.
3. If refresh is required, program MRTPR, URT and MAMR[RFEN].
4. Program the machine mode register (MxMR).

Patterns are written to the RAM array by setting MxMR[OP] = 01 and accessing the UPM with any write transaction that hits the relevant chip select. The entire array is thus programmed by an alternating series of writes: to MDR (RAM word to be written) each time, followed by a read from MDR and then a (dummy) write transaction to the relevant UPM assigned bank. A read from MDR is required to ensure

that the MDR has been updated with the desired value prior to the dummy write transaction. The following example further illustrates the steps required to perform *two* writes to the RAM array at *non-sequential* addresses, assuming that the relevant BRx and ORx registers have been previously set up:

1. Program the MxMR register for the first write (with the desired RAM array address).
2. Write pattern/data to MDR register (to ensure that the MxMR register has already been updated with the desired configuration).
3. Read the MDR register (to ensure that the MDR register has already been updated with the desired pattern).
4. Perform a dummy write transaction (so that a write transaction can now be performed).
5. Read/check the MxMR[MAD] bit. If the MAD bit is incremented then the previous dummy write transaction has completed and then proceed to step 6 or remain at step 5 otherwise. This step is required to ensure that the previous dummy transaction progresses until it finishes with the correct configuration.
6. Program the MxMR register for the second write (with the desired RAM array address).
7. Write pattern/data to MDR register (to ensure that the MxMR register has already been updated with the desired configuration).
8. Read the MDR register (to ensure that the MDR register has already been updated with the desired pattern).
9. Perform a dummy write transaction (so that a write transaction can now be performed).
10. Read/check the MxMR[MAD] bit. If the MAD bit is incremented then the previous dummy write transaction has finished.

Note that if step 1 (or step 6) and step 2 (or step 7) are reversed then step 3 (or step 8) is replaced by the following:

- Read the MxMR register (to ensure that the MxMR register has already been updated with the desired configuration).

RAM array contents may also be read for debug purposes, for example, by alternating dummy read transactions, each time followed by reads of the MDR register (when MxMR[OP] = 10). The following example further illustrates the steps required to perform *two* reads from the RAM array at *non-sequential* addresses assuming that the relevant BRx and ORx registers have been previously set up:

1. Program the MxMR register for the first read (with the desired RAM array address).
2. Read the MxMR register (to ensure that the MxMR register has already been updated with the desired configuration, such as RAM array address).
3. Perform a dummy read transaction (so that a read transaction can now be performed).
4. Read/check the MxMR[MAD] bit. If the MAD bit is incremented then the previous dummy read transaction is completed and proceed to step 5, or remains at step 4 otherwise. This step is required to ensure that the previous dummy transaction progresses until it finishes with the correct configuration.
5. Read the MDR register.
6. Program the MxMR register for the second read (with the desired RAM array address).

## External Memory Controller (EMC)

7. Read the MxMR register (to ensure that the MxMR register has already been updated with the desired configuration, such as RAM array address).
8. Perform a dummy read transaction (so that a read transaction can now be performed).
9. Read/check the MxMR[MAD] bit. If the MAD bit is incremented then the previous dummy read transaction is completed and proceed to step 10, or remain at step 9 otherwise.
10. Read the MDR register.

### 21.4.4.3 UPM Signal Timing

RAM word fields specify the value of the various external signals at a granularity of up to four values for each bus clock cycle. The signal timing generator causes external signals to behave according to the timing specified in the current RAM word.

For  $CRR[CLKDIV] = 4$  or  $8$ , each bit in the RAM word relating to  $\overline{LCSx}$  timing specifies the value of the corresponding external signal at each quarter phase of the bus clock.

If  $CRR[CLKDIV] = 2$ , the external signal can change value only on each half phase of the bus clock. If the RAM word in this case ( $LCRR[CLKDIV] = 2$ ) specifies a quarter phase signal change, the signal timing generator interprets this as a half cycle change.

The division of UPM bus cycles into phases is shown in [Figure 21-67](#) and [Figure 21-68](#). If  $CRR[CLKDIV] = 2$ , the bus cycle comprises only two active phases (T1, T3), which correspond with the first and second halves of the bus clock cycle, respectively.

However, if  $CRR[CLKDIV] = 4$  or  $8$ , four phases, T1–T4, define four quarters of the bus clock cycle. Because T2 and T4 are inactive when  $CRR[CLKDIV] = 2$ , UPM ignores the signal timing programmed for assertion in either of these phases in the case  $CRR[CLKDIV] = 2$ .

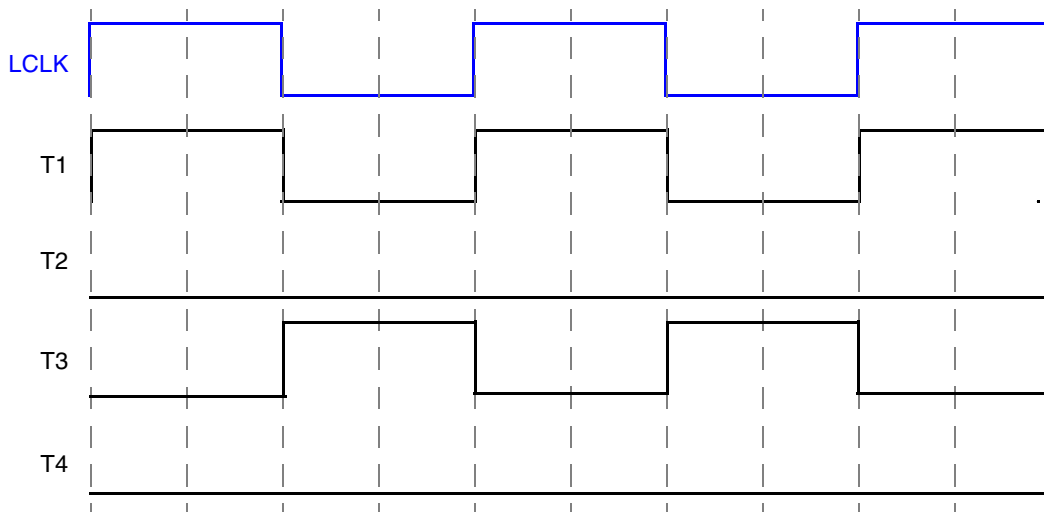


Figure 21-67. UPM Clock Scheme for  $CRR[CLKDIV] = 2$

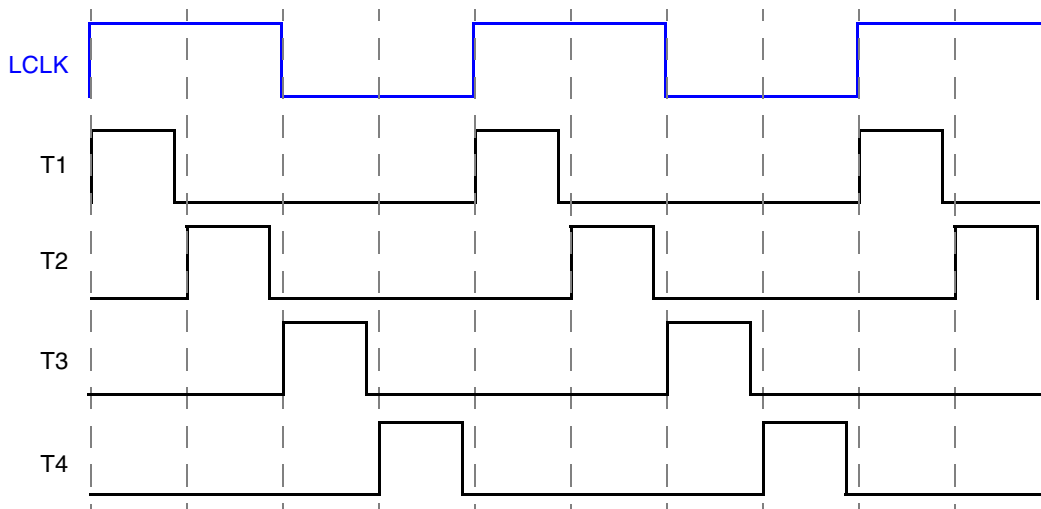


Figure 21-68. UPM Clock Scheme for  $CRR[CLKDIV] = 4$  (or 8) or  $CRR[CLKDIV] = 2$  (or 4)

### 21.4.4.4 The RAM Array

The RAM array for each UPM is 64 locations deep and 32 bits wide, as shown in Figure 21-69. The signals at the bottom of the figure are UPM outputs. The selected  $\overline{LCSx}$  is for the bank that matches the current address.

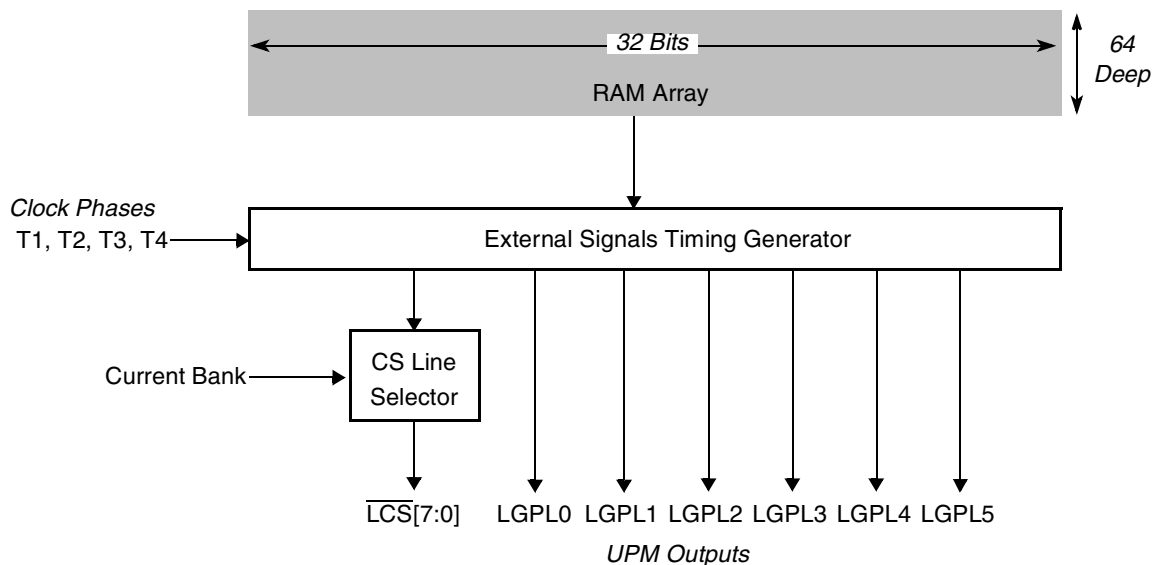


Figure 21-69. RAM Array and Signal Generation

#### 21.4.4.4.1 RAM Words

The RAM word is a 32-bit micro-instruction stored in one of 64 locations in the RAM array. It specifies the timing for the external signals controlled by the UPM. Figure 21-70 shows the RAM word fields.

## External Memory Controller (EMC)

When  $CRR[CLKDIV] = 4$  or  $8$ , the  $CSTx$  bits determine the state of UPM signals  $\overline{LCSx}$  at each quarter phase of the bus clock.

When  $CRR[CLKDIV] = 2$ ,  $CST2$  and  $CST4$  are ignored and the external signal has the values defined by  $CST1$  and  $CST3$  but extended to half the clock cycle in duration.

		Access: Read/Write															
		31	30	29	28	27	24	23	22	21	20	19	18	17	16		
R	W	CST1	CST2	CST3	CST4	—		G0L		G0H		G1T1	G1T3	G2T1	G2T3		
Reset		All zeros															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	W	G3T1	G3T3	G4T1DLT3	G4T3WAEN	G5T1	G5T3	REDO	LOOP	EXEN	AMX		NA	UTA	TODT	LAST	
Reset		All zeros															

**Figure 21-70. RAM Word Field Descriptions**

Table 21-42 defines RAM word fields.

**Table 21-42. RAM Word Field Descriptions**

Bits	Name	Description
31	CST1	Chip select timing 1. Defines the state (0 or 1) of $\overline{LCSx}$ during bus clock quarter phase 1 if $CRR[CLKDIV] = 4$ or $8$ . Defines the state (0 or 1) of $LCSx$ during bus clock half phase 1 if $CRR[CLKDIV] = 2$ .
30	CST2	Chip select timing 2. Defines the state (0 or 1) of $\overline{LCSx}$ during bus clock quarter phase 2 if $CRR[CLKDIV] = 4$ or $8$ . Ignored when $CRR[CLKDIV] = 2$ .
29	CST3	Chip select timing 3. Defines the state (0 or 1) of $\overline{LCSx}$ during bus clock quarter phase 3 if $CRR[CLKDIV] = 4$ or $8$ . Defines the state (0 or 1) of $LCSx$ during bus clock half phase 2 if $CRR[CLKDIV] = 2$ .
28	CST4	Chip select timing 4. Defines the state (0 or 1) of $\overline{LCSx}$ during bus clock quarter phase 4 if $CRR[CLKDIV] = 4$ or $8$ . Ignored when $CRR[CLKDIV] = 2$ .
27–24	—	Reserved
23–22	G0L	General purpose line 0 lower. Defines the state of $LGPL0$ during the bus clock quarter phases 1 and 2 (first half phase). 00 Value defined by $MxMR[G0CL]$ 01 Reserved 10 0 11 1
21–20	G0H	General purpose line 0 higher. Defines the state of $LGPL0$ during the bus clock quarter phases 3 and 4 (second half phase). 00 Value defined by $MxMR[G0CL]$ 01 Reserved 10 0 11 1



**Table 21-42. RAM Word Field Descriptions (Continued)**

Bits	Name	Description
19	G1T1	General purpose line 1 timing 1. Defines the state (0 or 1) of LGPL1 during bus clock quarter phases 1 and 2 (first half phase).
18	G1T3	General purpose line 1 timing 3. Defines the state (0 or 1) of LGPL1 during bus clock quarter phases 3 and 4 (second half phase)
17	G2T1	General purpose line 2 timing 1. Defines state (0 or 1) of LGPL2 during bus clock quarter phases 1 and 2 (first half phase).
16	G2T3	General purpose line 2 timing 3. Defines the state (0 or 1) of LGPL2 during bus clock quarter phases 3 and 4 (second half phase).
15	G3T1	General purpose line 3 timing 1. Defines the state (0 or 1) of LGPL3 during bus clock quarter phases 1 and 2 (first half phase).
14	G3T3	General purpose line 3 timing 3. Defines the state (0 or 1) of LGPL3 during bus clock quarter phases 3 and 4 (second half phase).
13	G4T1/DLT3	General purpose line 4 timing 1/delay time 3. The function of this bit is determined by MxMR[GPL4].  If MxMR[GPL4] = 0 and LGPL4/UPWAIT pin functions as an output (LGPL4), then G4T1/DLT3 defines the state (0 or 1) of LGPL4 during bus clock quarter phases 1 and 2 (first half phase).  If MxMR[GPL4] = 1 and LGPL4/UPWAIT functions as an input (UPWAIT), and if a read burst or single read is executed, then G4T1/DLT3 defines the sampling of the data bus as follows: 0 In the current word, the data bus should be sampled at the start of bus clock quarter phase 1 of the next bus clock cycle. 1 In the current word, the data bus should be sampled at the start of bus clock quarter phase 3 of the current bus clock cycle.
12	G4T3/WAEN	General purpose line 4 timing 3/wait enable. The function of this bit is determined by MxMR[GPL4].  If MxMR[GPL4] = 0 and LGPL4/UPWAIT pin functions as an output (LGPL4), then G4T3/WAEN defines the state (0 or 1) of LGPL4 during bus clock quarter phases 3 and 4 (second half phase).  If MxMR[GPL4] = 1 and LGPL4/UPWAIT functions as an input (UPWAIT), then G4T3/WAEN is used to enable the wait mechanism: 0 UPWAIT detection is disabled. 1 UPWAIT is enabled. If UPWAIT is detected as being asserted, a freeze in the external signals logical values occurs until UPWAIT is detected as being negated.
11	G5T1	General purpose line 5 timing 1. Defines the state (0 or 1) of LGPL5 during bus clock quarter phases 1 and 2 (first half phase).
10	G5T3	General purpose line 5 timing 3. Defines the state (0 or 1) of LGPL5 during bus clock quarter phases 3 and 4 (second half phase).
9–8	REDO	Redo current RAM word. Defines the number of times to execute the current RAM word. 00 once (normal operation) 01 twice 10 three times 11 four times

**Table 21-42. RAM Word Field Descriptions (Continued)**

Bits	Name	Description
7	LOOP	<p>Loop start/end.</p> <p>The first RAM word in the RAM array where LOOP = 1 is recognized as the loop start word. The next RAM word where LOOP = 1 is the loop end word. RAM words between, and including the start and end words, are defined as part of the loop.</p> <p>The number of times the UPM executes this loop is defined in the corresponding loop fields of the MxMR.</p> <p>0 The current RAM word is not the loop start word or loop end word.</p> <p>1 The current RAM word is the start or end of a loop.</p>
6	EXEN	<p>Exception enable. Allows branching to an exception pattern at the exception start address (EXS). When an internal bus monitor timeout exception is recognized and EXEN in the RAM word is set, the UPM branches to the special exception start address (EXS) and begins operating as the pattern defined there specifies.</p> <p>You should provide an exception pattern to deassert signals controlled by the UPM in a controlled fashion. For DRAM control, a handler should negate RAS and CAS to prevent data corruption. If EXEN = 0, exceptions are ignored by UPM (but not by the Local Bus) and execution continues. After the UPM branches to the exception start address, it continues reading until the LAST bit is set in the RAM word.</p> <p>0 The UPM continues executing the remaining RAM words, ignoring any internal bus monitor timeout.</p> <p>1 The current RAM word allows a branch to the exception pattern after the current cycle if an exception condition is detected.</p>
5–4	AMX	<p>Address multiplexing. Determines the source of LAD[23:0] during an LALE phase. Any change in the AMX field initiates a new LALE (address) phase.</p> <p>00 LAD[23:0] is the non-multiplexed address. For example, a column address.</p> <p>01 Reserved</p> <p>10 LAD[23:0] is the address multiplexed according to MxMR[AMx]. For example, a row address.</p> <p>11 LAD[23:0] is the contents of MAR. Used, for example, to initialize a mode.</p>
3	NA	<p>Next burst address: Determines when the address is incremented during a burst access.</p> <p>0 The address increment function is disabled.</p> <p>1 The address is incremented in the next cycle. The increment value of LA[2:0] is 1.</p>
2	UTA	<p>UPM transfer acknowledge. Indicates assertion of transfer acknowledge in the current cycle.</p> <p>0 Transfer acknowledge is not asserted in the current cycle.</p> <p>1 Transfer acknowledge is asserted in the current cycle.</p>
1	TODT	<p>Turn-on disable timer. The disable timer associated with each UPM allows a minimum time to be guaranteed between two successive accesses to the same memory bank. This feature is critical when DRAM requires a RAS precharge time. TODT turns the timer on to prevent another UPM access to the same bank until the timer expires.</p> <p>The disable timer period is determined in MxMR[DSx]. The disable timer does not affect memory accesses to different banks. Note that TODT must be set together with LAST, otherwise it is ignored.</p> <p>0 The disable timer is turned off.</p> <p>1 The disable timer for the current bank is activated, preventing a new access to the same bank (when controlled by the UPMs) until the disable timer expires. For example, a precharge time.</p>
0	LAST	<p>Last word. When LAST is read in a RAM word, the current UPM pattern terminates and the control signal timing set in the RAM word is applied to the current (and last) cycle. However, if the disable timer is activated and the next access is to the same bank, the execution of the next UPM pattern is held off and the control signal values specified in the last word are extended in duration for the number of clock cycles specified in MxMR[DSx].</p> <p>0 The UPM continues executing RAM words.</p> <p>1 Indicates the last RAM word in the program. The service to the UPM request is done after this cycle concludes.</p>

The following sections provide additional information about some of the RAM word fields.

### 21.4.4.4.2 Chip-Select Signal Timing (CST<sub>n</sub>)

If BR<sub>x</sub>[MSEL] of the accessed bank selects a UPM on the currently requested cycle, the UPM manipulates the  $\overline{\text{LCS}}_x$  signal for that bank with the timing specified in the UPM RAM word CST<sub>n</sub> fields. The selected UPM affects only the assertion and negation of the appropriate  $\overline{\text{LCS}}_x$  signal.

The state of the selected  $\overline{\text{LCS}}_x$  signal of the corresponding bank depends on the value of each CST<sub>n</sub> bit. Figure 21-71 shows how UPMs control  $\overline{\text{LCS}}_x$  signals.

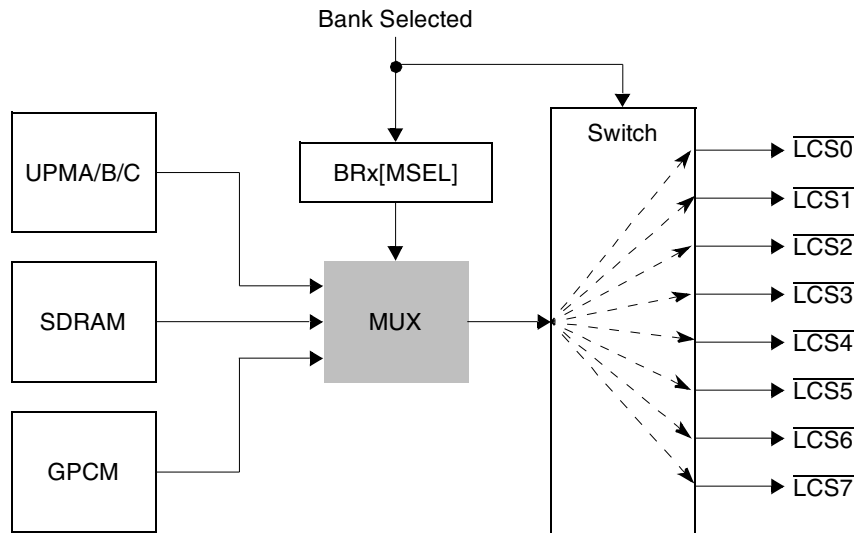


Figure 21-71.  $\overline{\text{LCS}}_x$  Signal Selection

### 21.4.4.4.3 General-Purpose Signals (GxTx, GOx)

The general-purpose signals (LGPL[5:0]) each have two bits in the RAM word that define the logical value of the signal to be changed at the rising edge of the bus clock and/or at the falling edge of the bus clock. LGPL0 offers enhancements beyond the other LGPL<sub>x</sub> lines.

GPL0 can be controlled by an address line specified in MxMR[G0CL<sub>x</sub>]. To use this feature, G0H and G0L should be set in the RAM word. For example, for a SIMM with multiple banks, this address line can be used to switch between internal memory device banks.

### 21.4.4.4.4 Loop Control (LOOP)

The LOOP bit in the RAM word specifies the beginning and end of a set of UPM RAM words that are to be repeated. The first time LOOP = 1, the memory controller recognizes it as a loop start word and loads the memory loop counter with the corresponding contents of the loop field (shown in Table 21-43). The next RAM word for which LOOP = 1 is recognized as a loop end word. When it is reached, the loop counter is decremented by one.

Continued loop execution depends on the loop counter. If the counter is not zero, the next RAM word executed is the loop start word. Otherwise, the next RAM word executed is the one after the loop end word. Loops can be executed sequentially but cannot be nested. Also, special care must be taken in the following case: LAST and LOOP must not be set together.

Table 21-43 shows the MxMR loop field use.

**Table 21-43. MxMR Loop Field Use**

Request Serviced	Loop Field
Read single-beat cycle	RLFx
Read burst cycle	RLFx
Write single-beat cycle	WLFx
Write burst cycle	WLFx
Refresh timer expired	TLFx
RUN command	RLFx

#### 21.4.4.4.5 Repeat Execution of Current RAM Word (REDO)

The REDO function is useful for wait-state insertion in a long UPM routine that would otherwise need too many RAM words. Setting the REDO bits of the RAM word to a nonzero value causes the UPM to re-execute the current RAM word up to three more times, as defined in the REDO field of the current RAM word.

Special care must be taken in the following cases:

- When UTA and REDO are set together, TA is asserted the number of times specified by the REDO function.
- When NA and REDO are set together, the address is incremented the number of times specified by the REDO function.
- When LOOP and REDO are set together, the loop mechanism works as usual and the line is repeated according to the REDO function.
- LAST and REDO must not be set together.
- REDO should not be used within the exception routine.

#### 21.4.4.4.6 Address Multiplexing (AMX)

The address lines can be controlled by the pattern the user provides in the UPM. The address multiplex bits can choose between driving the transaction address, driving it according to the multiplexing specified by the MxMR[AM] field, or driving the MAR contents on the address signals. In all cases, the LA[2:0] signals of the EMC are driven by the 3 LSBs of the address selected by AMX, regardless of whether the NA bit of the RAM word is used to increment the current address. The effect of NA = 1 is visible only when AMX = 00 chooses the column address.

Note that any change to the AMX field from one RAM word to the next RAM word executed results in an address phase on the LAD[23:0] bus with the assertion of LALE for the number of cycles set for LALE in the ORx and CRR registers. The LGPL[5:0] signals maintain the value specified in the RAM word during the LALE phase.

### 21.4.4.4.7 Data Valid and Data Sample Control (UTA)

When a read access is handled by the UPM, and the UTA bit = 1 (data is to be sampled by EMC), the value of the DLT3 bit in the same RAM word, in conjunction with MxMR[GPLx4DIS], determines when the data input is sampled by EMC, as follows:

- If MxMR[GPLx4DIS] = 1 (G4T4/DLT3 functions as DLT3) and DLT3 = 1 in the RAM word, data is latched on the falling edge of the bus clock instead of the rising edge. The data is sampled by EMC on the next falling edge of the bus clock, which is during the middle of the current bus cycle. This feature should only be used in systems without external synchronous bus devices that require mid-cycle sampling.
- If MxMR[GPLx4DIS] = 0 (G4T4/DLT3 functions as G4T4), or if MxMR[GPLx4DIS] = 1 but DLT3 = 0, data is latched on the rising edge of the bus clock, which occurs at the end of the current bus clock cycle (normal operation).

Figure 21-72 shows how data sampling is controlled by the UPM.

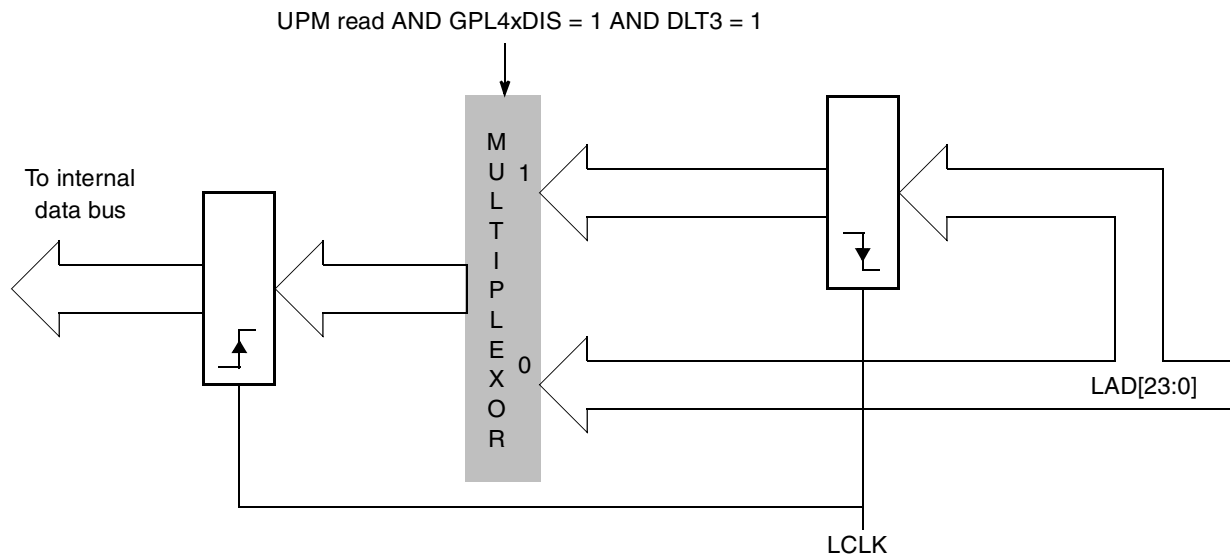


Figure 21-72. UPM Read Access Data Sampling

### 21.4.4.4.8 $\overline{\text{LGPL}}[5:0]$ Signal Negation (LAST)

When the LAST bit is read in a RAM word, the current UPM pattern is terminated at the end of the current cycle. On the next cycle (following LAST), all of the UPM signals are negated unconditionally (driven to logic 1), unless there is a back-to-back UPM request pending. In this case, the signal values for the cycle (following the cycle in which the LAST bit was set) are taken from the first RAM word of the pending UPM routine.

### 21.4.4.4.9 The Wait Mechanism (WAEN)

The WAEN bit in the RAM array word can be used to enable the UPM wait mechanism in selected UPM RAM words. If the UPM reads a RAM word with WAEN set, the external UPWAIT signal is sampled and

## External Memory Controller (EMC)

synchronized by the memory controller as if it were an asynchronous signal. The WAEN bit is ignored if  $LAST = 1$  in the same RAM word.

Synchronization of UPWAIT starts at the rising edge of the bus clock and takes at least 1 bus cycle to complete. If UPWAIT is asserted and  $WAEN = 1$  in the current UPM word, the UPM is frozen until UPWAIT is negated. The value of external signals driven by the UPM remains as indicated in the previous RAM word. When UPWAIT is negated, the UPM continues normal functions. Note that during WAIT cycles, the UPM does not handle data.

Figure 21-73 shows how the WAEN bit (in the word read by the UPM) and the UPWAIT signal are used to hold the UPM in a particular state until UPWAIT is negated. As the example shows, the  $\overline{LCSx}$  and  $\overline{LGPL1}$  states and the WAEN value are frozen until UPWAIT is recognized as negated. WAEN is typically set before the line that contain  $UTA = 1$ . Note that if WAEN and NA are both set in the same RAM word, NA causes the burst address to increment once as normal regardless of whether UPM freezes or not.

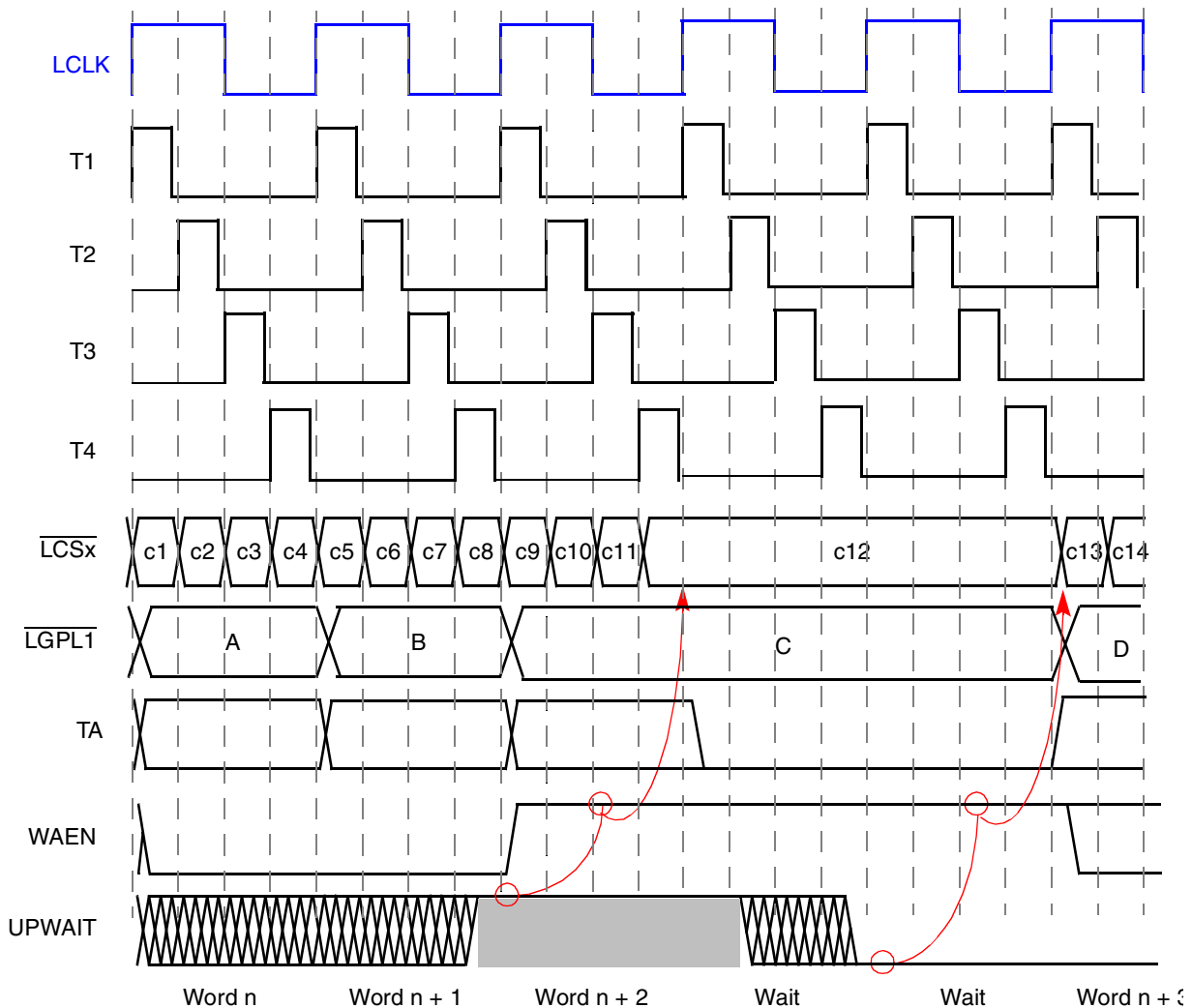


Figure 21-73. Effect of UPWAIT Signal

#### 21.4.4.5 Synchronous Sampling of UPWAIT for Early Transfer Acknowledge

If UPWAIT is to be considered an asynchronous signal (which can be asserted/negated at any time), no UPM RAM word must contain both WAEN = 1 and UTA = 1 simultaneously.

However, programming WAEN = 1 and UTA = 1 in the same RAM word allows UPM to treat UPWAIT as a synchronous signal, which must meet set-up and hold times in relation to the rising edge of the bus clock. In this case, as soon as UPM samples UPWAIT negated on the rising edge of the bus clock, it immediately generates an internal transfer acknowledge, which allows a data transfer one bus clock cycle later. The generation of transfer acknowledge is early because UPWAIT is not re-synchronized, and the acknowledge occurs regardless of whether UPM was already frozen in WAIT cycles or not. This feature allows the synchronous negation of UPWAIT to affect a data transfer.

#### 21.4.4.6 Extended Hold Time on Read Accesses

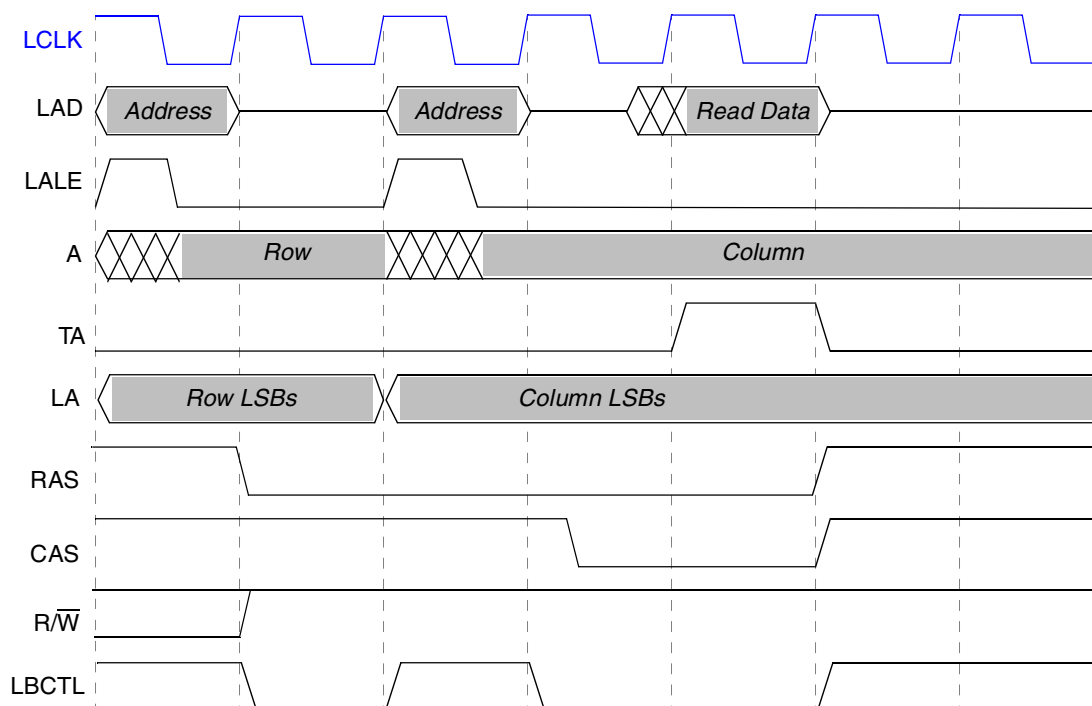
Slow memory devices that take a long time to turn off their data bus drivers on read accesses should choose some non-zero combination of ORx[TRLX] and ORx[EHTR]. The next accesses after a read access to a slow memory device is delayed by the number of clock cycles (specified in the ORx register), in addition to any existing bus turnaround cycle.

#### 21.4.4.7 Memory System Interface Example Using UPM

Connecting the external memory UPM controller to a DRAM device requires a detailed examination of the timing diagrams representing the possible memory cycles that must be performed when accessing this device. This section describes timing diagrams for various UPM configurations, using fast-page mode DRAM as an example, with CRR[CLKDIV] = 4 or 8.

The examples shown are for illustrative purposes only, and may not represent the timing necessary for any specific device used with the EMC. In the examples, LGPL1 is programmed to drive  $R/\overline{W}$  of the DRAM, although any LGPLx signal may be used for this purpose.

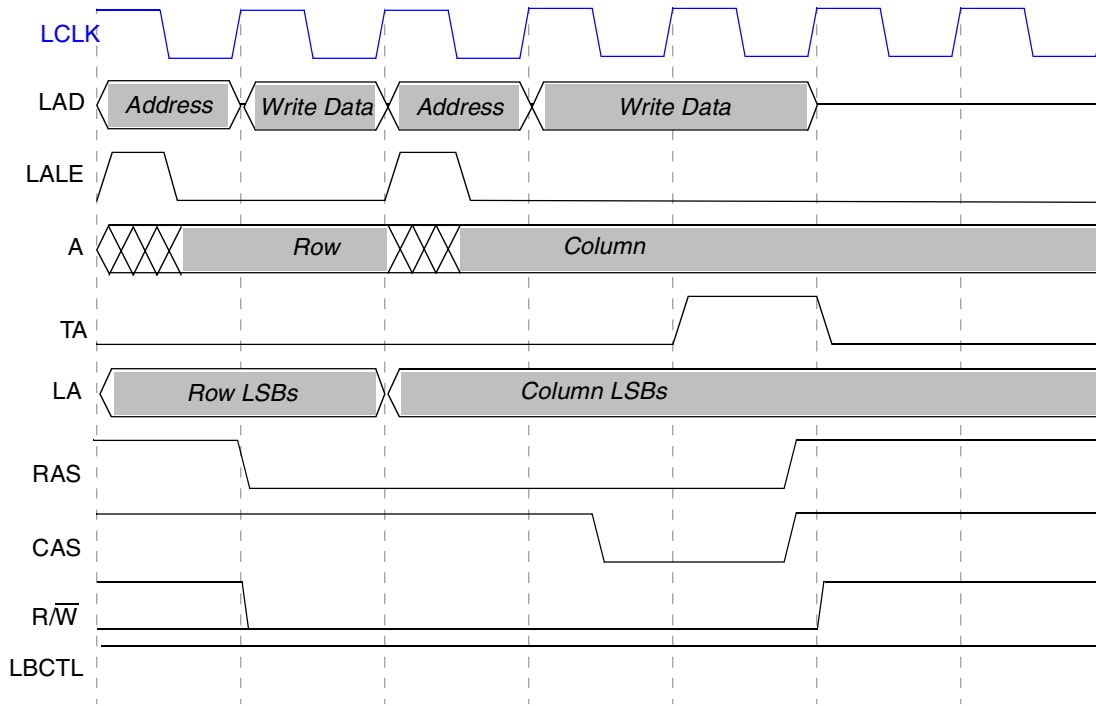
### External Memory Controller (EMC)



cst1	Bit 0	0	LALE pause (due to change in AMX)	0	0
cst2	Bit 1	0		0	0
cst3	Bit 2	0		0	0
cst4	Bit 3	0		0	0
Reserved	Bit 4	1		1	0
Reserved	Bit 5	1		0	0
Reserved	Bit 6	1		0	0
Reserved	Bit 7	1		0	0
g0l0	Bit 8				
g0l1	Bit 9				
g0h0	Bit 10				
g0h1	Bit 11				
g1t1	Bit 12	1		1	1
g1t3	Bit 13	1		1	1
g2t1	Bit 14				
g2t3	Bit 15				
g3t1	Bit 16				
g3t3	Bit 17				
g4t1	Bit 18				
g4t3	Bit 19				
g5t1	Bit 20				
g5t3	Bit 21				
redo[0]	Bit 22				
redo[1]	Bit 23				
loop	Bit 24	0		0	0
exen	Bit 25	0		0	0
amx0	Bit 26	1		0	0
amx1	Bit 27	0		0	0
na	Bit 28	0		0	0
uta	Bit 29	0		0	1
todt	Bit 30	0		0	1
last	Bit 31	0		0	1
	RSS	RSS	RSS+1	RSS+1	RSS+2

Figure 21-74. Single-Beat Read Access to FPM DRAM

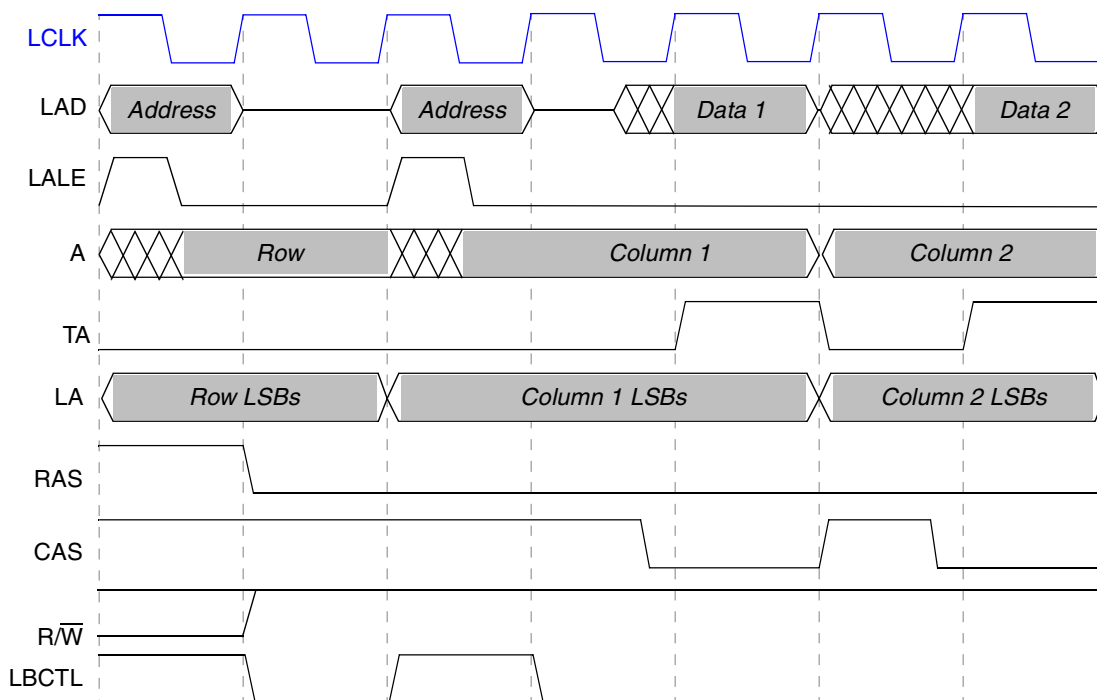




cst1	Bit 0	0	LALE pause (due to change in AMX)	0	0
cst2	Bit 1	0		0	0
cst3	Bit 2	0		0	0
cst4	Bit 3	0		0	1
Reserved	Bit 4	1		1	0
Reserved	Bit 5	1		1	0
Reserved	Bit 6	1		0	0
Reserved	Bit 7	1		0	1
g0i0 g0i1 g0h0 g0h1	Bit 8 Bit 9 Bit 10 Bit 11				
g1t1 g1t3	Bit 12 Bit 13	0 0		0 0	0 0
g2t1 g2t3	Bit 14 Bit 15				
g3t1 g3t3	Bit 16 Bit 17				
g4t1 g4t3	Bit 18 Bit 19				
g5t1 g5t3	Bit 20 Bit 21				
redo[0] redo[1]	Bit 22 Bit 23				
loop exen	Bit 24 Bit 25	0 0	0 0	0 0	
amx0 amx1	Bit 26 Bit 27	1 0	0 0	0 0	
na uta	Bit 28 Bit 29	0 0	0 0	0 1	
todt last	Bit 30 Bit 31	0 0	0 0	1 1	
	<b>WSS</b>	<b>WSS</b>	<b>WSS+1</b>	<b>WSS+1</b>	<b>WSS+2</b>

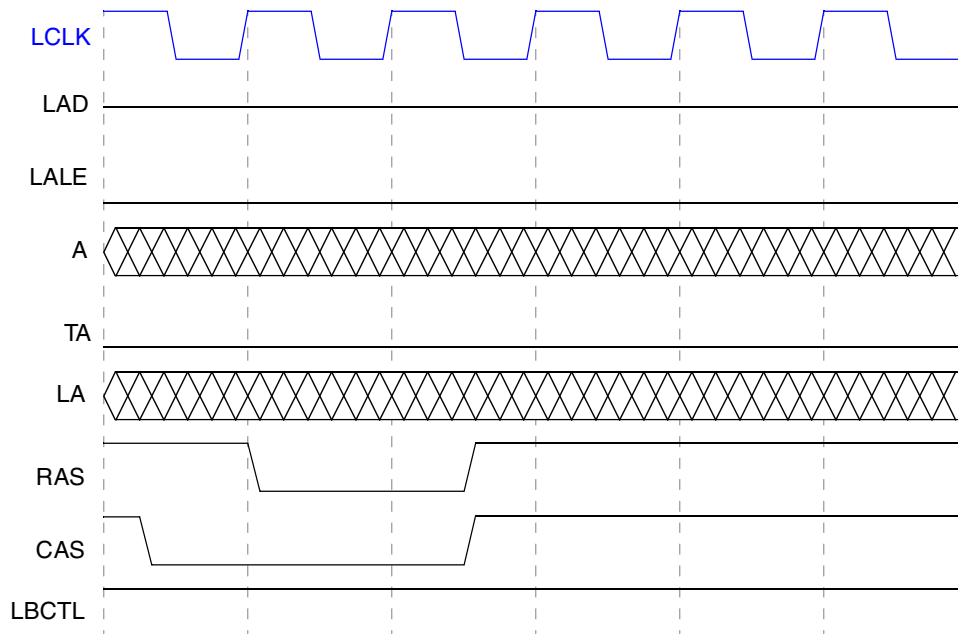
Figure 21-75. Single-Beat Write Access to FPM DRAM

### External Memory Controller (EMC)



cst1	Bit 0	0	LALE pause (due to change in AMX)	0	0	1
cst2	Bit 1	0		0	0	1
cst3	Bit 2	0		0	0	1
cst4	Bit 3	0		0	0	1
Reserved	Bit 4	1		1	0	1
Reserved	Bit 5	1		1	0	1
Reserved	Bit 6	1		1	0	1
Reserved	Bit 7	1		0	0	1
g0I0	Bit 8					
g0I1	Bit 9					
g0h0	Bit 10					
g0h1	Bit 11					
g1t1	Bit 12	1	1	1	1	
g1t3	Bit 13	1	1	1	1	
g2t1	Bit 14					
g2t3	Bit 15					
g3t1	Bit 16					
g3t3	Bit 17					
g4t1	Bit 18					
g4t3	Bit 19					
g5t1	Bit 20					
g5t3	Bit 21					
redo[0]	Bit 22					
redo[1]	Bit 23					
loop	Bit 24	0	1	1	0	
exen	Bit 25	0	0	1	0	
amx0	Bit 26	1	0	0	0	
amx1	Bit 27	0	0	0	0	
na	Bit 28	0	0	1	0	
uta	Bit 29	0	0	1	0	
todt	Bit 30	0	0	0	1	
last	Bit 31	0	0	0	1	
	<b>RBS</b>	<b>RBS</b>		<b>RBS+1</b>	<b>RBS+2</b>	<b>RBS+3</b>

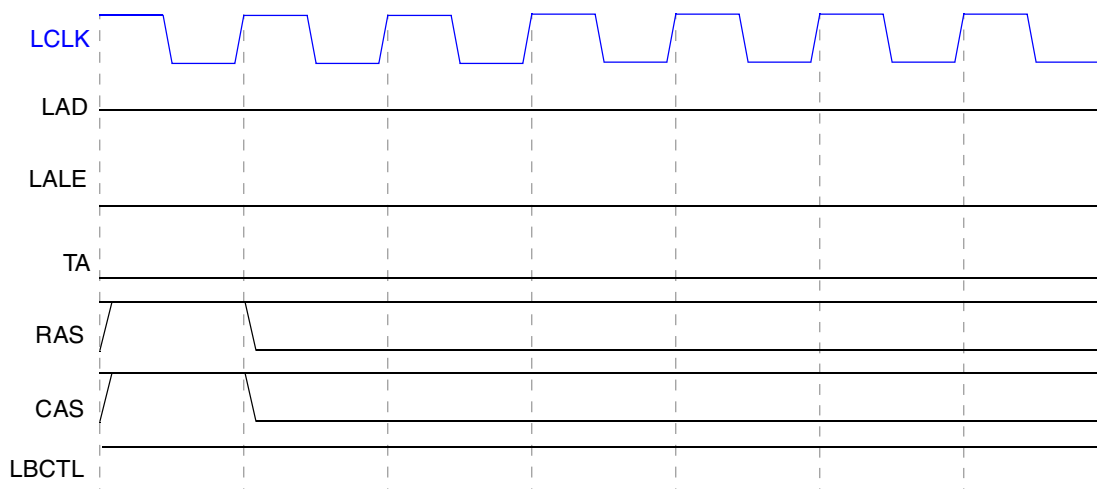
Figure 21-76. Burst Read Access to FPM DRAM Using LOOP (2 Beats Shown)



cst1	1	0	0	Bit 0
cst2	1	0	0	Bit 1
cst3	1	0	1	Bit 2
cst4	1	0	1	Bit 3
Reserved	1	0	0	Bit 4
Reserved	0	0	0	Bit 5
Reserved	0	0	1	Bit 6
Reserved	0	0	1	Bit 7
g0l0				Bit 8
g0l1				Bit 9
g0h0				Bit 10
g0h1				Bit 11
g1t1				Bit 12
g1t3				Bit 13
g2t1				Bit 14
g2t3				Bit 15
g3t1				Bit 16
g3t3				Bit 17
g4t1				Bit 18
g4t3				Bit 19
g5t1				Bit 20
g5t3				Bit 21
redo[0]				Bit 22
redo[1]				Bit 23
loop	0	0	0	Bit 24
exen	0	0	0	Bit 25
amx0	0	0	0	Bit 26
amx1	0	0	0	Bit 27
na	0	0	0	Bit 28
uta	0	0	0	Bit 29
todt	0	0	1	Bit 30
last	0	0	1	Bit 31
	<b>PTS</b>	<b>PTS+ 1</b>	<b>PTS+ 2</b>	

Figure 21-77. Refresh Cycle to FPM DRAM

### External Memory Controller (EMC)



cst1	1	Bit 0
cst2	1	Bit 1
cst3	1	Bit 2
cst4	1	Bit 3
Reserved	1	Bit 4
Reserved	1	Bit 5
Reserved	1	Bit 6
Reserved	1	Bit 7
g0l0		Bit 8
g0l1		Bit 9
g0h0		Bit 10
g0h1		Bit 11
g1t1		Bit 12
g1t3		Bit 13
g2t1		Bit 14
g2t3		Bit 15
g3t1		Bit 16
g3t3		Bit 17
g4t1		Bit 18
g4t3		Bit 19
g5t1		Bit 20
g5t3		Bit 21
redo[0]		Bit 22
redo[1]		Bit 23
loop	0	Bit 24
exen	0	Bit 25
amx0	0	Bit 26
amx1	0	Bit 27
na	0	Bit 28
uta	0	Bit 29
todt	1	Bit 30
last	1	Bit 31
	<b>EXS</b>	

Figure 21-78. Exception Cycle

## 21.5 Application Information

### 21.5.1 Interfacing to Peripherals

#### 21.5.1.1 Multiplexed Address and Data Bus and Unmultiplexed Address Signals

To save pins on the EMC, the address and data are multiplexed onto the same 24-bit bus. An external latch is needed to unmultiplex and reconstruct the original address. No external intelligence is needed, because the LALE signal provides the correct timing to control a standard logic latch. The LAD pins can be directly connected to the data signals of the memory/peripheral.

Transactions on the EMC start with an address phase, where the EMC drives the transaction address on the LAD signals and asserts the LALE signal. This can be used to latch the address, and then the EMC can continue with the data phase.

In addition, the EMC supports burst transfers (not in the GPCM machine). LA[2:0] are the burst addresses within a natural 8-word burst. To minimize the amount of address phases needed on the EMC and to optimize the throughput, those signals are driven separately and should be used whenever a device requires the three least significant addresses. Those should not be used from LAD[2:0]. All other addresses, A[23:3], must be reconstructed through the latch.

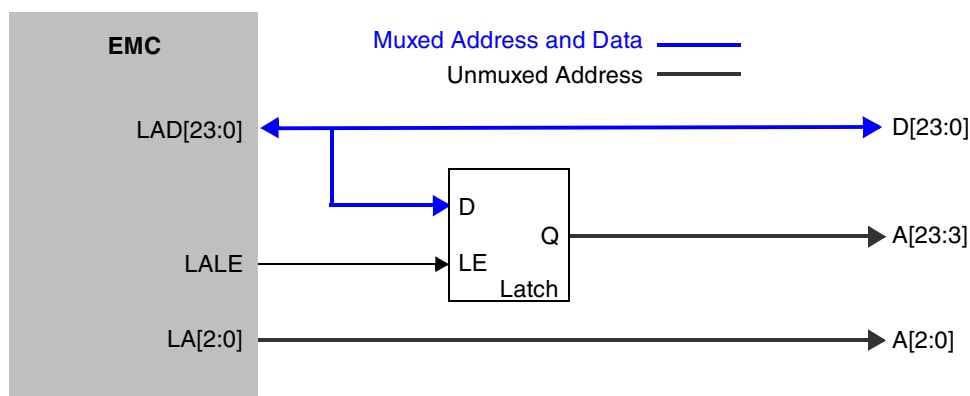
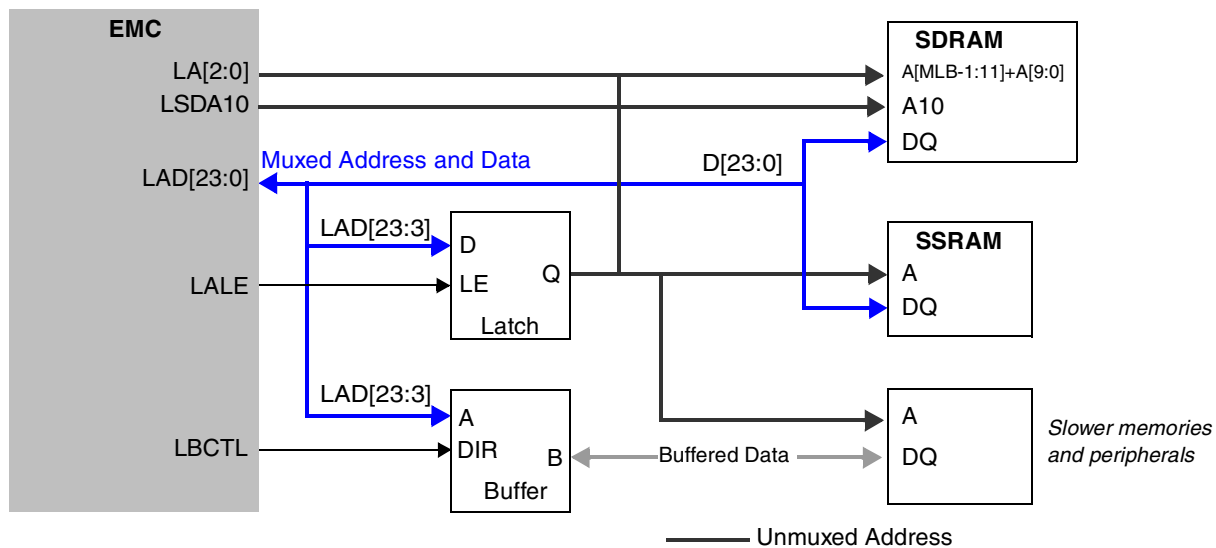


Figure 21-79. Multiplexed Address and Data Bus

### 21.5.1.2 Peripheral Hierarchy on the External Memory Controller

To achieve high bus speed interfaces for synchronous SRAMs or SDRAMs, a hierarchy of the memories/peripherals connected to the EMC is suggested. Figure 21-80 shows an example of such a hierarchy.



**Figure 21-80. External Memory Controller Peripheral Hierarchy**

The multiplexed address and data bus sees the capacitive loading of the data pins of the fast SDRAMs or synchronous SRAMs, plus one load for an address latch, plus one load for a buffer to the slow memories. The loadings of all other memories and peripherals are hidden behind the buffer and the latch. The system designer needs to investigate the loading scenario and ensure that I/O timings can be met with the loading determined by the connected components.

### 21.5.1.3 Peripheral Hierarchy on External Memory Controller for Very High Bus Speeds

To achieve the highest possible bus speeds on the EMC, it is recommended to reduce even further the number of devices connected directly to the EMC. For those cases, probably only one bank of synchronous SRAMs or SDRAMs should be used, and instead of using a separate latch and a separate bus transceiver, a bus demultiplexer combining those two functions into one device should be used. Figure 21-81 shows an example of such a hierarchy. This section is only a guideline, and the board designer must simulate the electric characteristics of his scenario to determine the maximum operating frequency.

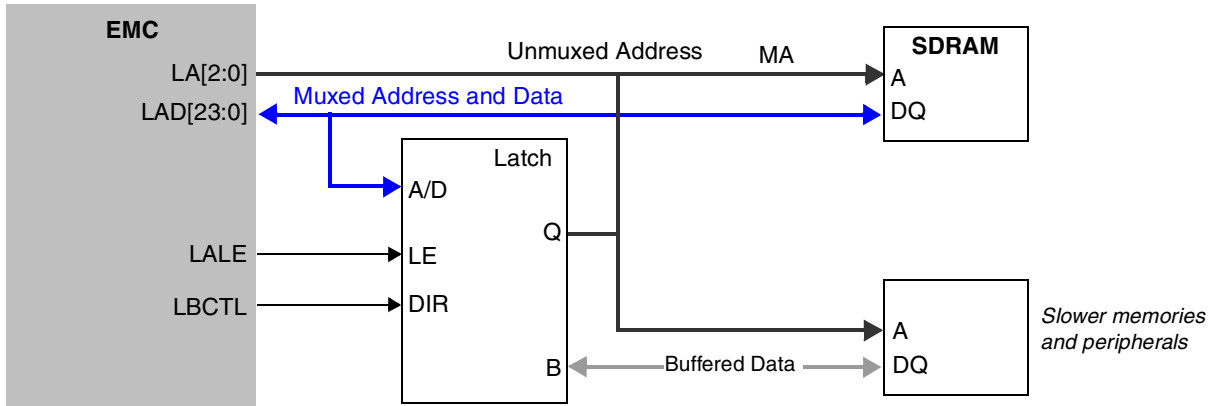


Figure 21-81. EMC Peripheral Hierarchy for Very High Bus Speeds

### 21.5.1.4 GPCM Timing

In the case where a system contains a memory hierarchy with high speed synchronous memories (SDRAM, synchronous SRAM) and lower speed asynchronous memories (like flash EPROM, peripherals, and others), then the GPCM controlled memories should be decoupled by buffers to reduce capacitive loading on the bus. Those buffers have to be taken into account for the timing calculations.

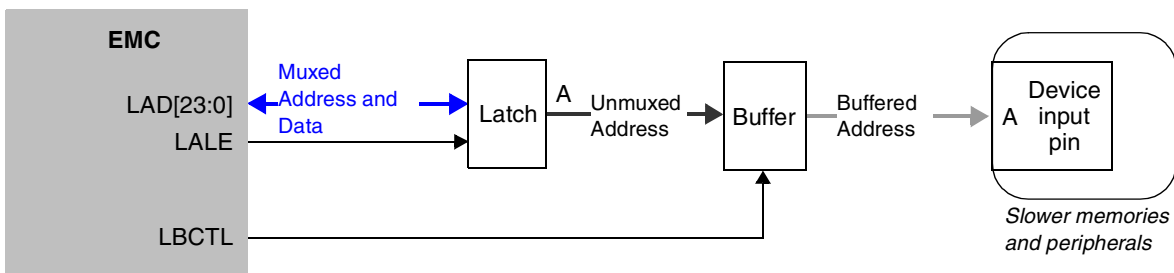


Figure 21-82. GPCM Address Timing

To calculate address set-up timing for a slower peripheral/memory, several parameters have to be added:

- Propagation delay for the address latch
- Propagation delay for the buffer
- Address set-up time for the actual peripheral

Typical values for the two propagation delays are in the order of 3–6 ns.

For data timings only the propagation delay of one buffer plus the actual data set-up time has to be considered.

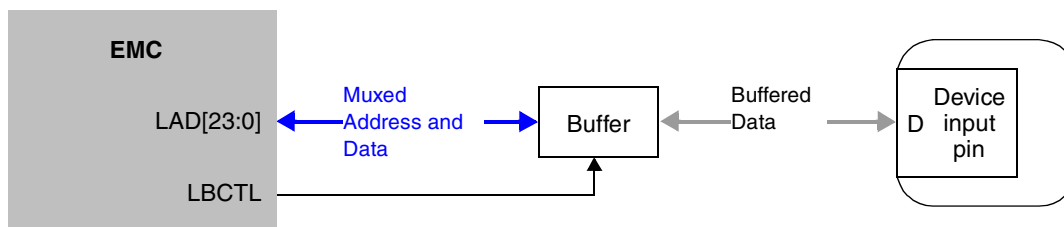


Figure 21-83. GPCM Data Timing

## 21.5.2 Bus Turnaround

Because the EMC uses multiplexed address and data, give special consideration to avoid bus contention at bus turnaround. The following cases must be examined:

- Address phase after previous read
- Read data phase after address phase
- UPM cycles with additional address phases

Because the bus does not change direction, the following cases do not require special attention:

- Continued burst after the first beat
- Write data phase after address phase
- Address phase after previous write

### 21.5.2.1 Address Phase after Previous Read

During a read cycle, the memory/peripheral drives the bus and the bus transceiver drives LAD. After the data has been sampled, the output drivers of the external device must be disabled. This can take some time; for slow devices the EHTR feature of the GPCM or the programmability of the UPM should be used to guarantee that those devices have stopped driving the bus when the EMC memory controller ends the bus cycle.

In this case, after the previous cycle ends, LBCTL goes high and changes the direction of the bus transceiver. The EMC then inserts a bus turnaround cycle to avoid contention. The external device has now already placed its data signals in high impedance and no bus contention will occur.

### 21.5.2.2 Read Data Phase after Address Phase

During the address phase, LAD actively drives the address and LBCTL is high, driving bus transceivers in the same direction as during a write. After the end of the address phase, LBCTL goes low and turns around the direction of the bus transceiver. The EMC places the LAD signals in high impedance after its  $t_{dis}(LB)$ . The LBCTL will have its new state after  $t_{en}(LB)$ , and because this is an asynchronous input, the transceiver starts to drive those signals after its  $t_{en}(transceiver)$  time. To avoid bus contention, you have to ensure that  $[t_{en}(LB) + t_{en}(transceiver)]$  is larger than  $t_{dis}(LB)$ .



### 21.5.2.3 UPM Cycles with Additional Address Phases

The flexibility of the UPM allows you to insert additional address phases during read cycles by changing the AMX field, thereby turning around the bus during one pattern. The EMC automatically inserts a single bus turnaround cycle if the bus (LAD) was previously high impedance for any reason, such as a read, before LALE is driven and if LAD is driven with the new address. The turnaround cycle is not inserted on a write, because the bus was already driven to begin with.

Bus contention could potentially still occur on the far side of a bus transceiver. It is the responsibility of the designer of the UPM pattern to guarantee that enough idle cycles are inserted in the UPM pattern to avoid this.

## 21.5.3 Interfacing to SDRAM

### 21.5.3.1 Basic SDRAM Capabilities of the External Memory Controller

The EMC provides one SDRAM machine for the external memory. Although there is only one SDRAM machine, multiple chip selects ( $\overline{\text{LCS}}_x$ ) can be programmed to support multiple SDRAM devices. Note that no limitation exists on the number of chip selects that can be programmed for SDRAM. This means that  $\overline{\text{LCS}}[7:1]$  can be programmed to support SDRAM, assuming  $\overline{\text{LCS}}_0$  is reserved for the general-purpose chip-select machine (GPCM) to connect to Flash memory.

If multiple chip selects are configured to support SDRAM on the EMC, each SDRAM device should have the same timing parameters. This means that all option registers ( $\text{OR}_n$ ) for the SDRAM chip selects should be programmed exactly the same.

#### NOTE

Although in principal it is possible to mix timing parameters, combinations are limited and this operation is not recommended.

All the chip selects share the same external memory SDRAM mode register (SDMR) for initialization, and also share the EMC-assigned SDRAM refresh timer register (SRT) and the memory refresh timer prescaler register (MPTPR), for refresh purposes.

For refresh, the memory controller supplies auto refresh to SDRAM according to the time interval specified in SRT and MPTPR, which is:

$$\text{Refresh Period} = \frac{\text{SRT}_x (\text{MPTPR}[\text{PTP}]}{\text{System Frequency}}$$

This represents the time period required between refreshes. When the refresh timer expires, the memory controller issues a refresh command to each chip select. Each refresh command is separated by one clock. A refresh timing diagram for multiple chip selects is shown in [Figure 21-63](#).

During a memory transaction dispatched to the EMC, the memory controller compares the memory address with the address information of each chip select (programmed with  $\text{BR}_n$  and  $\text{OR}_n$ ). If the comparison matches a chip select that is controlled by SDRAM, the memory controller requests service to the EMC SDRAM machine, depending on the information in  $\text{BR}_n$ . Although multiple chip selects may be

## External Memory Controller (EMC)

programmed for SDRAM, only one chip select is active at any given time; thus, multiple chip selects can share the same SDRAM machine.

### 21.5.3.2 Maximum Amount of SDRAM Supported

Table 21-44 summarizes information based on SDRAM data sheets supplied by Micron.

**Table 21-44. Micron SDRAM Devices**

	SDRAM Devices											
	64 Mbit				128 Mbit				256 Mbit			
I/O Port	x4	x8	x16	x32	x4	x8	x16	x32	x4	x8	x16	x32
Bank	4	4	4	4	4	4	4	4	4	4	4	4
Row	12	12	12	11	12	12	12	12	13	13	13	13
Column	10	9	8	8	11	10	9	8	11	10	9	8

The following examples use all 24 bits of the EMC. The 24-bit port size requires two SDRAM devices (with 16-bit I/O ports) or three SDRAM devices (with 8-bit I/O port), all connected in parallel to a single chip select.

### 21.5.3.3 Example of SDRAM Usage

This section shows examples of the use of EMC SDRAM machine.

#### 21.5.3.3.1 Maximum Row Number Due to Bank Select MUX

SDMR[BSMA] is used to multiplex the bank select address. The BSMA field and corresponding multiplexed address are shown below:

```

000 LA17–LA16
001 LA16–LA15
...
111 LA10–LA9

```

Note that LA17 is the latched value of LAD17.

The highest address pins that the bank selects can be multiplexed with, are LA[17:16], which limits the pins for the row address to LA[15:0]. The EMC SDRAM machine supports 15 rows, which is sufficient for all devices.

### 21.5.3.3.2 Bank Select Signals

Page-based interleaving allows bank signals to be multiplexed to the higher-order address pins to leave room for future upgrades. For example, you could multiplex the bank select signals to LA[15:14], leaving LA13 to connect to the address pin for a larger memory size.

This allows you to design a board that can be used with a current generation of SDRAM devices today, and in the future upgrade to the next generation of SDRAM devices without requiring a new board layout.

### 21.5.3.3.3 SDRAM of 256-Mbit

Figure 21-84 shows a SDRAM of 256-Mbit. Note that all of the circuit diagrams mainly show the connections, and do not guarantee signal integrity.

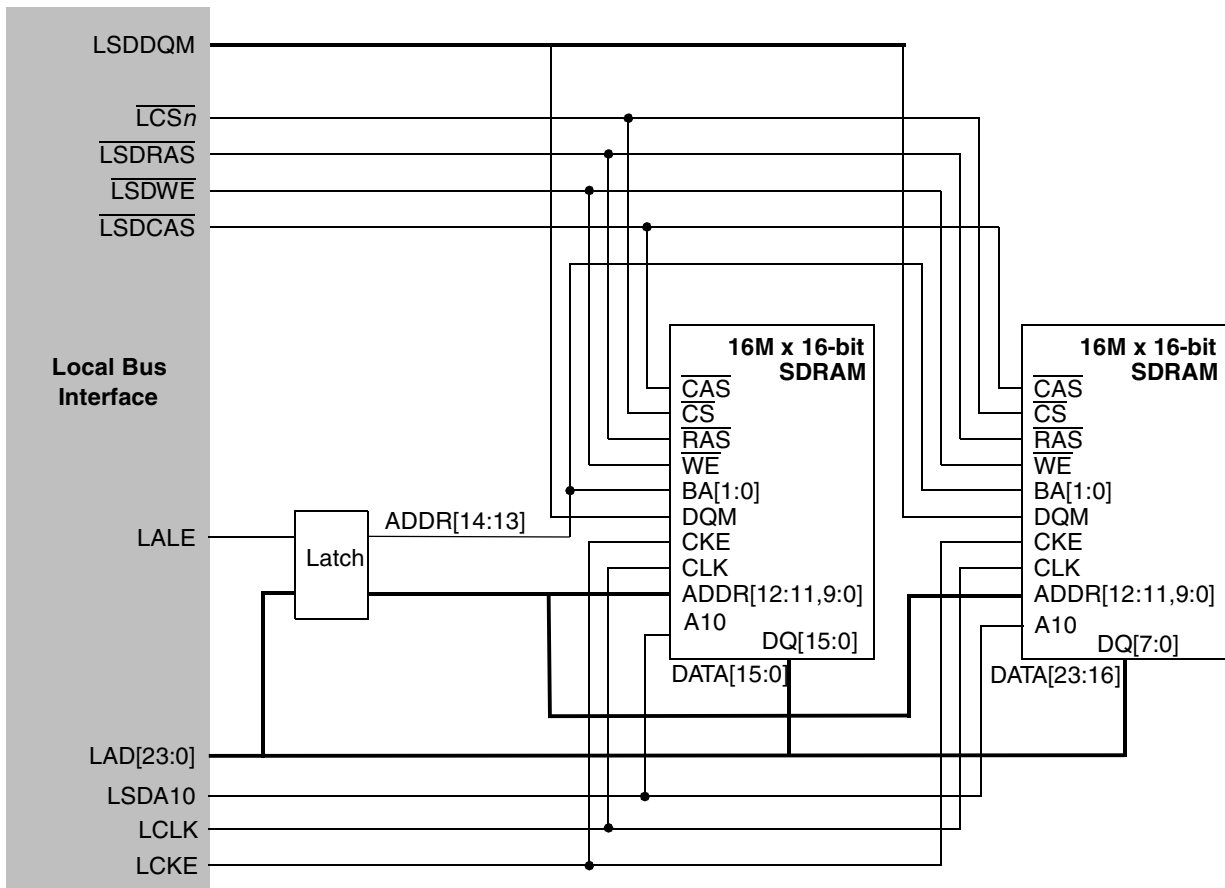


Figure 21-84. 256-Mbit SDRAM Diagram

Consider the following SDRAM organization:

- The 24-bit port size is combined with two 16-bit devices. All 16 bits of the first device and 8 bits of the LSBs of the second device are used.
- Each device has four internal banks, 13 row address lines, and 9 column address lines.

The logical address is partitioned as shown in [Table 21-45](#).

**Table 21-45. Logical Address Bus Partitioning**

A[23:11]	A[10:9]	A[8:0]
Row	Bank select	Column

The following parameters are extracted:

- COLS = 010, 9 column lines
- ROWS = 100, 13 row lines

During the row address phase, the SDRAM address port is set as shown in [Table 21-46](#).

**Table 21-46. SDRAM Device Address Port During Row Address Phase**

LA[23:15]	LA[14:13]	LA[12:0]
—	Bank	Row

Because the internal bank selects are multiplexed over LA[14:13], SDMR[BSMA] must be set to 011.

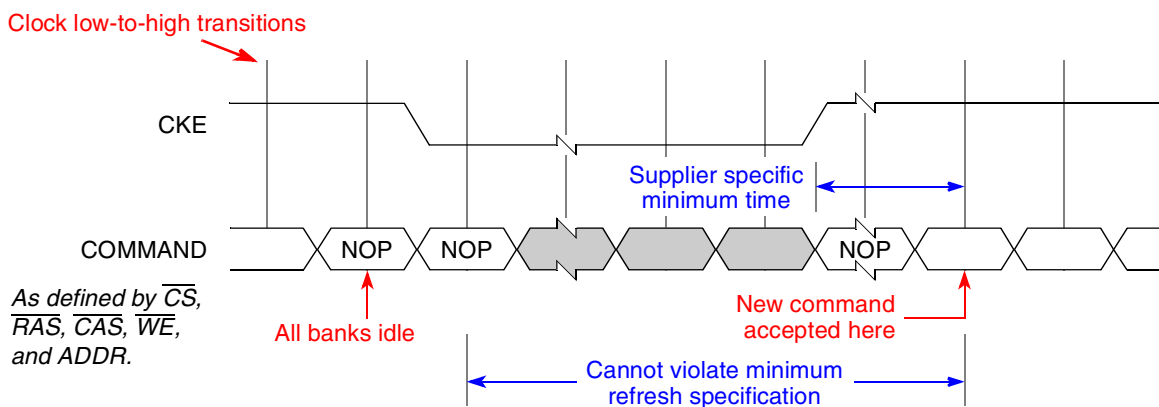
[Table 21-47](#) shows the address port configuration during a READ/WRITE command.

**Table 21-47. SDRAM Device Address Port During READ/WRITE Command**

LA[23:15]	LA[14:13]	LA[12, 11]	LA[10]	LA[9]	LA[8:0]
MSB of Start Address	Bank	<i>Don't Care</i>	AP	<i>Don't Care</i>	Column

### 21.5.3.3.4 Power-Down Mode

SDRAMs offer a power-down mode, during which the device is not refreshed, and therefore data is not maintained. This power-down mode is invoked by driving CKE low while all internal banks are idle; note that the banks must be precharged first. [Figure 21-85](#) shows the timing.



**Figure 21-85. SDRAM Power-Down Mode Timing**

CKE remains low, as long as the device is powered down. After CKE transitions to high, the SDRAM exits the power-down mode.

### 21.5.3.3.5 Self-Refresh

To enable stop activity on the EMC (for power save or debug), and also maintain the content of the SDRAM, the self-refresh mode is supported. The self-refresh mode is invoked by issuing a self-refresh command to the SDRAM. The EMC applies the same timing as for the auto-refresh, but also pulls SDRAM CKE (LCKE) low in the same cycle. This can only be done if all banks are idle; the SDRAM machine must precharge them ahead of this. As long as CKE stays low, the device refreshes itself and does not need to see any refreshes from the EMC. To exit self-refresh, CKE simply has to be pulled high. Note that after returning from self-refresh mode the SDRAM needs a supplier-specific time before it can accept new commands, and the auto-refresh mechanism has to be started again. Figure 21-86 shows this timing. The SDRAM controller always uses 200 EMC clocks, which should satisfy any SDRAM requirements.

See Section 21.4.3.3, “JEDEC-Standard SDRAM Interface Commands,” for SDRAM interface commands and information on the self-refresh command.

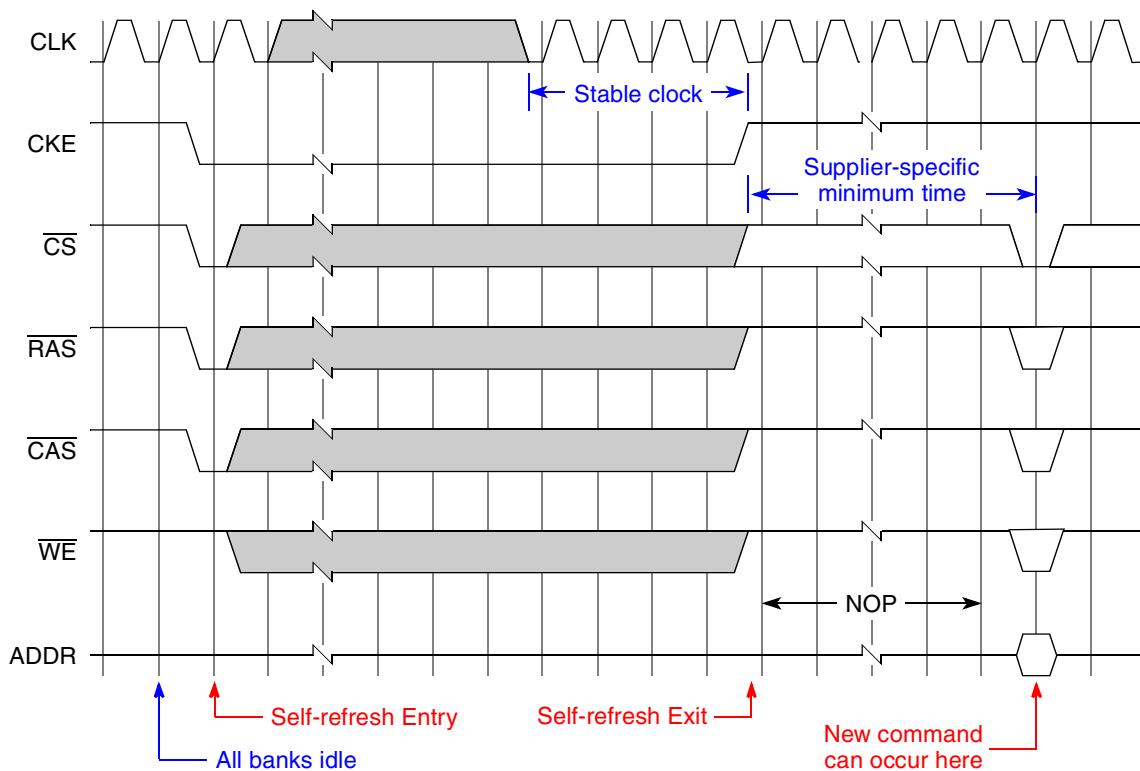


Figure 21-86. SDRAM Self-Refresh Mode Timing

### 21.5.3.3.6 SDRAM Timing

To allow for very high speeds on the memory bus, the capacitive loading on the EMC must be taken into consideration.

**Table 21-48. SDRAM Capacitance**

Pin	Min	Max	Unit
CLK	2.0	4.0	pf
$\overline{\text{RAS}}$ , $\overline{\text{CAS}}$ , $\overline{\text{WE}}$ , $\overline{\text{CS}}$ , CKE, DQM	2.0	5.0	pf
Address	2.0	5.0	pf
DQ	3.5	6.5	pf

**Note:** Capacitance values were compiled from worst case numbers from various datasheets from Samsung and Micron.

To implement a system (using the hierarchy described earlier) for two synchronous memory banks, one address latch and one buffer loading the multiplexed address and data bus sees a loading of four loads of about 6.5 pF maximum. 30 pF can be used as a nominal load.

**Table 21-49. SDRAM AC Characteristics**

Parameter	Device Speed				Unit	Notes Ta 0~65C, Vcc 3.0v~3.6v
	66 MHz		100 MHz			
	Min	Max	Min	Max		
CLK cycle time	15	—	10	—	ns	—
CLK to valid output delay	—	9	—	6	ns	—
Output data hold time	2.5	—	2.5	—	ns	—
Input setup time	3	—	2	—	ns	—
input hold time	1.5	—	1	—	ns	—
CLK to output in Hi-Z	2.5	12	2.5	9	ns	—

**Note:** AC Characteristics compiled from worst case numbers from various datasheets from Samsung and Micron

To improve the timing margins a PLL is used to generate external clocks, which minimizes the skew between the EMC and the memory clock. [Figure 21-87](#) shows the internal and external connection of the PLL.

The purpose of the PLL in the EMC is to move the edges of the LCLK signal such that the bus clock arrives at external RAM devices *synchronously*, with respect to the EMC after pad delay and PCB flight time have been accounted for. In practice, this requires rising edges of LCLK to emerge from 72x early with respect to received data sample points, so that set-up margins for EMC reads are increased. The read path performance is typically the most critical, and hence the PLL is necessary at frequencies of 100 MHz and above. For less critical EMC writes and control signals, set-up margin at the external RAM is degraded by the PLL, but hold time is improved accordingly.

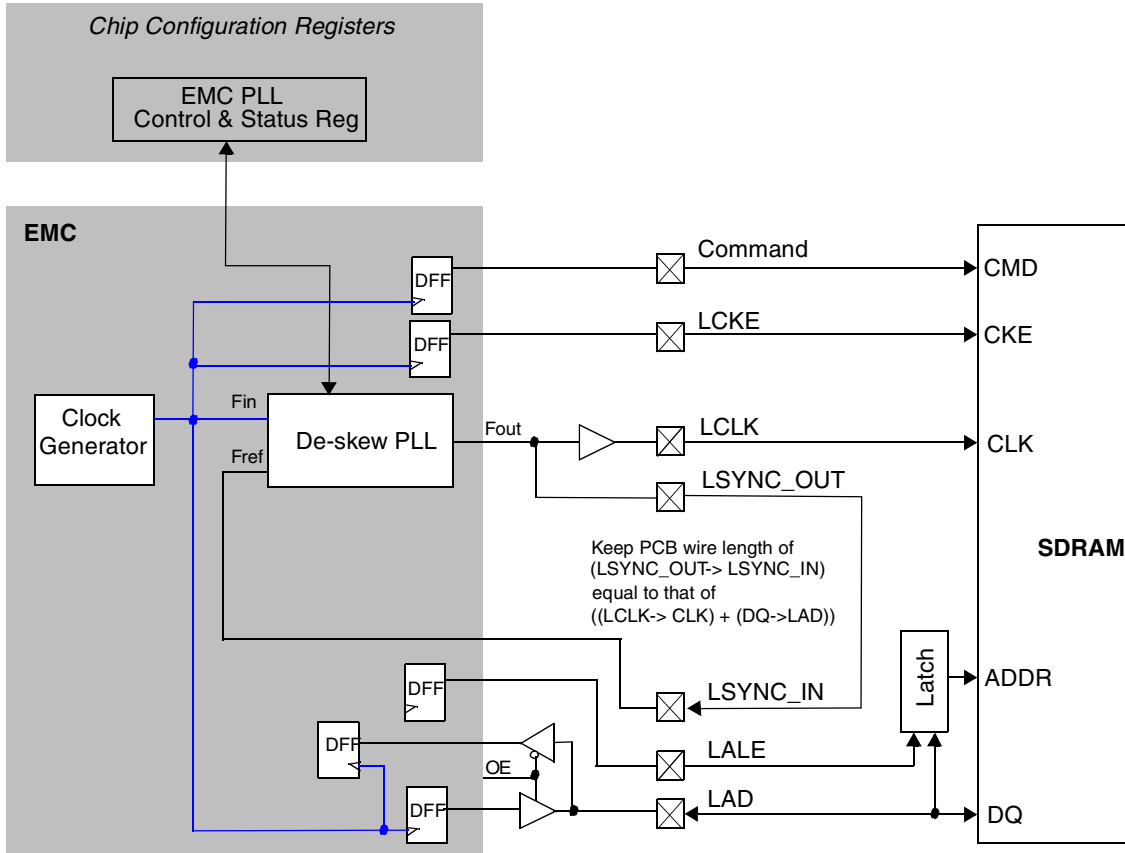


Figure 21-87. EMC De-Skew PLL





## Chapter 22

# JTAG Controller

### 22.1 Overview

In the DSP56724 and DSP56725 devices, there are two JTAG modules (one for each core), connected together serially to support multi-core debug OnCEs. When one DSP core enters the debug state, you can also define whether the other DSP core enters the debug state (or not). When multiple break-points are defined on different DSPs in a multi-DSP design, you are able to determine which DSP core initiated the entering of the debug state. To support this feature, two JTAG modules are daisy-chained, and it looks like two single core devices to the outside world.

The instruction length of each JTAG is 4 bits; for a correct Update-IR operation, an 8-bit shift from TDI is required. JTAG-0 only includes BYPASS and ONCE related instructions. JTAG-1 includes the OnCE-related instructions and all of the JTAG standard test instructions like BYPASS, IDCODE, EXTEST, HIZ, and so on.

Figure 22-1 shows the JTAG block diagram.

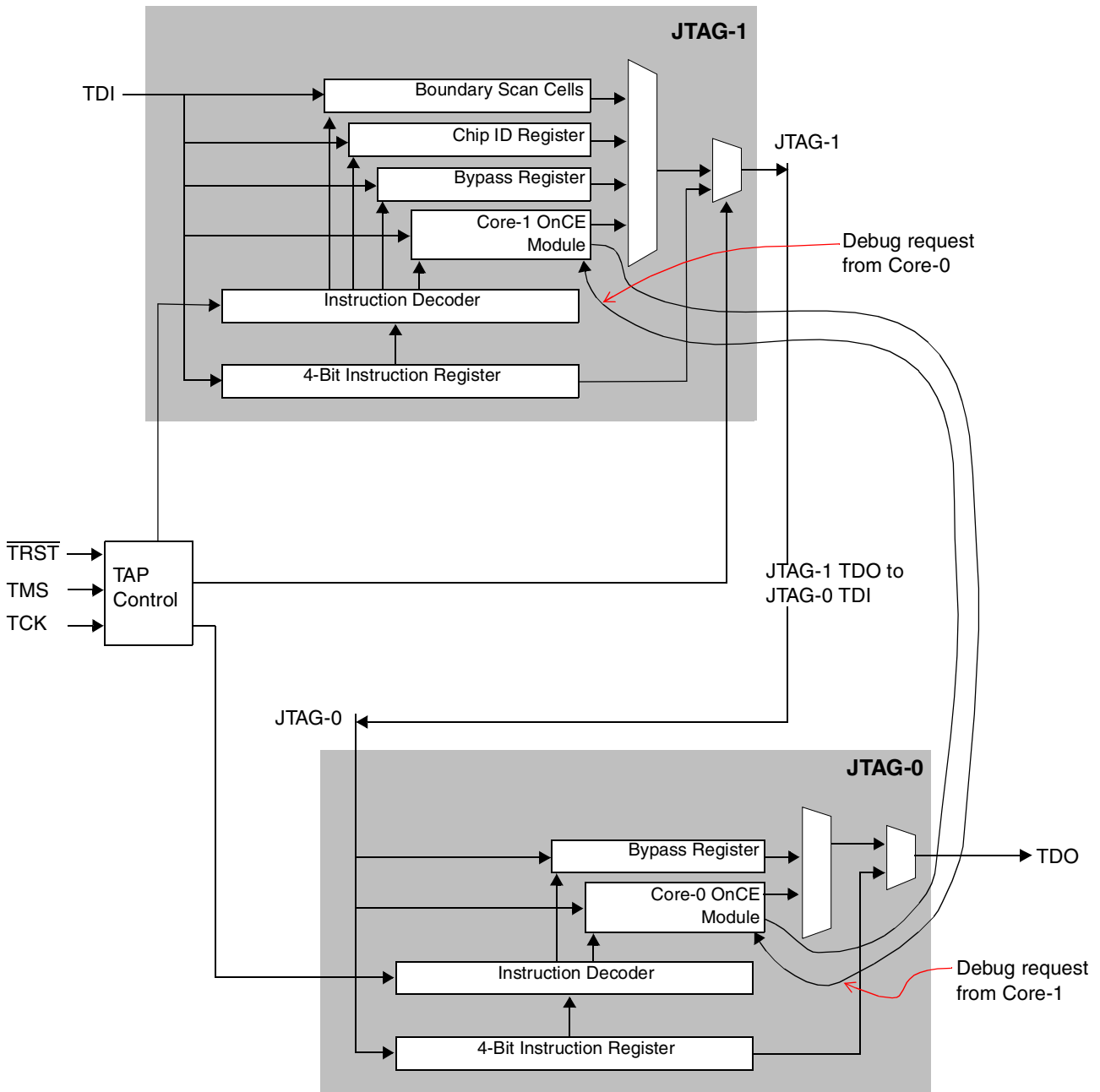


Figure 22-1. DSP56724/DSP56725 JTAG Block Diagram

## 22.2 Features

The DSP56724/DSP56725 JTAG:

- Performs boundary scan operations to check circuit-board electrical connectivity
- Bypasses the DSP56xxx for a given circuit-board test, by effectively reducing the boundary scan register to two bypass cells

- Disables the output drive to pins during circuit board testing
- Gives entry to debug mode
- Queries identification information (such as manufacturer, part number and version) from a DSP56xxx-based device
- Accesses the OnCE controller and circuits to control a target system
- Supports multiple core OnCE control and interconnection

**Table 22-1. JTAG-1 Supported Instructions**

jtag_ins[3:0]	Description	jtag_ins[3:0]	Description
4'b0000	EXTEST	4'b1000	<i>Reserved</i>
4'b0001	SAMPLE_RELOAD	4'b1001	Customized test instruction
4'b0010	IDCODE	4'b1010	Customized test instruction
4'b0011	CLAMP	4'b1011	<i>Reserved</i>
4'b0100	Hi-Z	4'b1100	Customized test instruction
4'b0101	<i>Reserved</i>	4'b1101	Customized test instruction
4'b0110	ENABLE_ONCE	4'b1110	Customized test instruction
4'b0111	DEBUG_REQUEST	4'b1111	BYPASS

**Table 22-2. JTAG-0 Supported Instructions**

jtag_ins[3:0]	Description	jtag_ins[3:0]	Description
4'b0000	<i>Reserved</i>	4'b1000	<i>Reserved</i>
4'b0001		4'b1001	
4'b0010		4'b1010	
4'b0011		4'b1011	
4'b0100		4'b1100	
4'b0101		4'b1101	
4'b0110		ENABLE_ONCE	
4'b0111	DEBUG_REQUEST	4'b1111	BYPASS

**Table 22-3. JTAG Identification Register Configuration**

31	28 27	22 21	12 11	1 0
<b>Version Information</b>	<b>Customer Part Number</b>	<b>Sequence Number</b>	<b>Manufacturer Identity</b>	<b>1</b>
0000	000111	1011010000	00000001110	1

**Table 22-4. JTAG Instruction Descriptions**

Type	Instruction	Description
JTAG Standard Instructions	EXTEST, SAMPLE_RELOAD, BYPASS, IDCODE, CLAMP	Standard JTAG instructions
	JTAG ID: 0x01ED001D	Standard JTAG instructions
OnCe Debug	ENABLE_ONCE	Selects the data register in the OnCe module connected between TDI and TDO.
	DEBUG_REQUEST	Issues a DEBUG_REQUEST to the related OnCE block.
Chip-Level Test	Hi_Z	Disables all of the outputs.

## 22.3 External Signal Descriptions

**Table 22-5. External Signal Descriptions**

Name	Function	I/O	Reset	Pull-Up
TCK	The external clock that synchronizes the test logic.	I	–	Pull-Up
TDI	Receives serial test instruction and data, which is sampled on the rising edge of TCK and has an internal pull-up resistor. Register values are shifted in Least Significant Bit (LSB) first.	I	–	Pull-Up
$\overline{\text{TRST}}$	Initializes the test controller asynchronously. TRST has an internal pull-up resistor.	I	From on-chip power on reset block.	–
TDO	The serial output for test instructions and data. TDO is tri-state and is actively driven in the shift-IR and shift-DR controller states. TDO changes on the falling edge of TCK. Register values are shifted out LSB first.	O	–	–
TMS	Sequences the JTAG controller state machine. TMS is sampled on the rising edge of TCK and has an internal pull-up resistor.	I	–	Pull-Up

# Appendix A

## Revision History

This appendix provides a list of the major differences between the *Symphony™ DSP56724/DSP56725 Multi-Core Audio Processors Reference Manual*, Revision 0 through Revision 1.

### A.1 Changes From Revision 0 to Revision 1

Major changes to the *Symphony™ DSP56724/DSP56725 Multi-Core Audio Processors Reference Manual*, from Revision 0 to Revision 1 are as follows:

Section, Page	Changes
Throughout	Remove all references to HD.
<a href="#">About This Book</a>	Modify the second paragraph, as follows: “The DSP56724/DSP56725 Multi-Core Audio Processors are devices of the DSP5672x family of programmable CMOS DSPs, designed using dual DSP56300 24-bit cores. The DSP56724/DSP56725 are intended for automotive, consumer, and professional audio applications that require high performance for audio processing. Potential applications include A/V receivers, car audio/amplifiers, and professional audio equipment.”
<a href="#">1.1/1-1</a>	In the list of features, change “400 MIPs (200 MIPs/core) with a 200 MHz clock” to say “up to 500 MIPs (up to 250 MIPs/core) with up to 250-MHz clock...”
<a href="#">1.4.2/1-5</a>	Modify the second sentence in the first paragraph, as follows: “The DSP56724/DSP56725 shared memory has four 8-Kword × 24-word memory blocks for a total of 32-Kword shared words and is located starting from 0x030000.”
<a href="#">1.4.5/1-6</a>	Revise section.
<a href="#">2.2.9/2-14</a>	In <a href="#">Table 2-15</a> , “Enhanced Serial Audio Interface Signals (ESAI_2),” change all instances of the following sentence: “Uses internal pull-up resistor in DSP56725 80-pin and 144-pin packages” to: “Uses internal pull-up resistor in DSP56725 80-pin package.”
<a href="#">2.2.12/2-29</a>	In <a href="#">Table 2-19</a> , “Digital Audio Interface: S/PDIF Signals,” remove both instances of the following sentence: “This pin is only available in the DSP56724 and DSP56725 144-pin package.”
<a href="#">3.1/3-1</a>	Change the sentence “Four blocks of 8 Kbytes” to say “Eight blocks of 4 Kwords.”

- 3.2/3-2 Switch [Table 3-1](#) and [Table 3-2](#), change the title of [Table 3-1](#) to “Core-0 Configuration,” and change the title of [Table 3-2](#) to “Core-1 Configuration.”  
In addition, switch [Table 3-3](#) and [Table 3-4](#), change the title of [Table 3-3](#) to “DSP Core-0 Memory Map Locations,” and change the title of [Table 3-4](#) to “DSP Core-1 Memory Map Locations.”
- 3.2/3-2 In [Table 3-4](#), “DSP Core-1 Memory Map Locations,” for MSW = NA, MS = 0, change program RAM range from “0x000000–0x007FFF” to “0x000000–0x0007FF.”
- 4.2/4-2 In list of features, change second bullet to: “Up to 250 million instructions per second (MIPS) with up to 250-MHz clock with 1.2-V internal logic supply”
- 7.1.1/7-1 Change 200 MHz to 250 MHz, and 400 MHz to 500 MHz.
- 7.3.3.2/7-8 Add new row to the top of [Table 7-9](#), “PLL Programming Examples,” as follows:

**Table 7-9. PLL Programming Examples**

Extal (MHz)	NR (= R + 1)	Fref (= Fosc/NR) (2~8 MHz)	NF (= F + 1)	Fvco (= Fref*NF) (200~400 MHz)	NO (=2^OD)	PLL Output (= Fvco/NO) (MHz)	PLL Setting (0x)
24.576	12	2.048	122	249.856	1	249.856	2B2079

- 8.2.6/8-10 Remove the following sentence: “When not used as timer signals, the two sets of timer event counter signals (TIO0, TIO1, TIO2, TIO0\_1, TIO1\_1, TIO2\_1) can be configured as GPIO signals.”
- 11.1.1/11-1 Modify the first and second bullet, as follows:
  - Uses internal clocking.
  - Interrupts the DSP Core after a specified number of events (clocks)
- Chapter 12 Remove Chapter 12, “Host Data Interface (HDI24, HDI24\_1).”
- 12.2/12-1 Change beginning of first sentence of second paragraph to: “In the DSP56725...”
- 15.1/15-1 Change memory space address to “0x030000–0x37ffff.”  
In addition, change “4 × 8 Kbytes” to “8 × 4 Kwords,” and change “Multiples of 8K memory blocks...” to “Consists of multiple 4-Kword memory blocks...”
- 19.1.2/19-3 In [Table 19-1](#), “ASRC Specifications,” add the following note to 40 ms: “The settling time is defined to be the time from audio input to the conversion performance reaching –115 dB THD + N.”
- 19.1.2/19-3 Add two new bullets, as follows:
  - Output sampling rates in the range of 8 kHz to 200 kHz are also supported but with less performance.
  - Designed for real-time streaming audio usage i.e. the output sampling clock must always be physically available in the system.
- 19.1.2/19-3 Significantly revise the first bullet under “Clock/Data Connections...”

19.1.2/19-3	Modify last sub-bullet, as follows: “Core master clock derivative (default is 5.644 MHz, programmable through external module)”
19.1.3.1/19-4	Change headings of modes 1–3 to be more specific, as follows: <ul style="list-style-type: none"> <li>• “Data Input Mode 1 (Polling Mode)”</li> <li>• “Data Input Mode 2 (Interrupt Mode)”</li> <li>• “Data Input Mode 3 (DMA Mode)”</li> <li>• “Data Output Mode 1 (Polling Mode)”</li> <li>• “Data Output Mode 2 (Interrupt Mode)”</li> <li>• “Data Output Mode 3 (DMA Mode)”</li> </ul>
Data Input Mode 1 (Polling Mode)/19-4	Add the following text: “the FIFO size of each channel is 64 samples.”
Data Output Mode 1 (Polling Mode)/19-5	Add the following text: “the FIFO size of each channel is 64 samples.”
19.2/19-5	In <a href="#">Table 19-2</a> , “Block Memory Map,” changed rows for 0xC, 0xD, and 0xF to reserved and added new rows for new registers. <a href="#">19.2.2.1/19-7</a> Modify <a href="#">Figure 19-3</a> , “ASRC Control Register (ASRCTR),” and <a href="#">Table 19-3</a> , “ASRC Control Register Bits (ASRCTR).”
19.2.2/19-7	Remove section, “Memory Access Registers (ASRMAA, ASRMAD).”
19.2.2/19-7	Add the following sections: <ul style="list-style-type: none"> <li>• <a href="#">Section 19.2.2.12.1</a>, “Ideal Ratio Registers for Pair A, High/Low Part (ASRIDRHA, ASRIDRLA)”</li> <li>• <a href="#">Section 19.2.2.12.2</a>, “Ideal Ratio Registers for Pair B, High/Low Part (ASRIDRHB, ASRIDRLB)”</li> <li>• <a href="#">Section 19.2.2.12.3</a>, “Ideal Ratio Registers for Pair C, High/Low Part (ASRIDRHC, ASRIDRLC)”</li> <li>• <a href="#">Section 19.2.2.12.4</a>, “ASRC 76 kHz Period Register in Terms of Master Clock (ASR76K)”</li> <li>• <a href="#">Section 19.2.2.12.5</a>, “ASRC 56 kHz Period Register in Terms of Master Clock (ASR56K)”</li> </ul>
19.2.2.4/19-12	Change bit 20 from reserved to “NDPRC” in <a href="#">Figure 19-7</a> , “Filter Configuration Status Register (ASRCFG),” and <a href="#">Table 19-7</a> , “Filter Configuration Status Register (ASRCFG).” <a href="#">19.2.2.5/19-14</a> Add values 1000–1100 to each bit field description in <a href="#">Table 19-8</a> , “Clock Source Register (ASRCSSR).”
19.2.2.8/19-21	Change title of section to “ASRC Debug Control Register (ASRCDR).” In addition, remove <a href="#">Figure 19-30</a> , “Debug Control Register-1 (ASRDCCR1),” and <a href="#">Table 19-19</a> , “Debug Control Register-1 (ASRDCCR1).”
19.5.1.1/19-28	Add new column “8” and new rows “8,” “12,” “16,” and “24,” to <a href="#">Table 19-19</a> , “Pre-Processing, Post-Processing Options.”

## Revision History

19.5.1.2.1/19-30      Change the sentence in this section to say, “The device supports only physical sampling clocks. The clocks can be the clocks from SPDIF, ESAI, or the PLL.”