M•CORE™

MMC2001
Reference Manual
Revision 1.1

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola 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 Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola 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 Motorola was negligent regarding the design or manufacture of the part. The Motorola name and logotype are trademarks of Motorola, Inc.

The M•CORE name and logotype and the OnCE name are trademarks of Motorola, Inc.

© Motorola, Inc. 1998

Freescale Semiconductor, Inc.

Conventions

This document uses the following notational conventions:

mnemonics	Instruction mnemonics are shown in lowercase bold
0x0F	Hexadecimal numbers
0b0011	Binary numbers

Nomenclature

Logic level one is the voltage that corresponds to a Boolean true (1) state.

Logic level zero is the voltage that corresponds to a Boolean false (0) state.

To **set** a bit or bits means to establish logic level one on the bit or bits.

To **clear** a bit or bits means to establish logic level zero on the bit or bits.

LSB means least significant bit or bits. **MSB** means most significant bit or bits. References to low and high bytes are spelled out.

A signal is **asserted** when it is in its active or true state, regardless of whether that state is represented by a high or low voltage. A signal is **negated** when it is in its inactive or false state.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

TABLE OF CONTENTS

Paragraph	Title	Page

SECTION 1
INTRODUCTION

SECTION 2
INTEGER CPU

2.1 M•CORE Overview 2-1
2.2 Features 2-2
2.3 Microarchitecture Summary 2-2
2.4 Programming Model 2-3
2.5 Data Format Summary 2-5
2.6 Operand Addressing Capabilities 2-6
2.7 Instruction Set Overview 2-6
2.8 M•CORE Bus Interface 2-8
2.8.1 Bus Characteristics 2-8
2.8.2 Bus Signals 2-9
2.8.3 Signal Descriptions 2-10
2.8.4 Bus Operation 2-11
2.8.5 Processor Instruction/Data Transfers 2-13
2.8.6 Bus Exception Cycles 2-14

SECTION 3
SYSTEM MEMORY MAP

3.1 Overview 3-1
3.2 Peripheral Module Address Allocation 3-1
3.3 Peripheral Module Interface Operation 3-2
3.4 Peripheral Module Address Assignment 3-2

SECTION 4
SIGNAL DESCRIPTIONS

4.1 Overview 4-1
4.2 Signal Index 4-2
4.3 Bus Signals 4-4
4.3.1 Address Bus (ADDR[19:0]) 4-4
4.3.2 Data Bus (DATA[15:0]) 4-4
4.3.3 Output Enable (OE) 4-4
4.3.4 Read/Write Enable (R/W) 4-4
4.3.5 Enable Byte 1 (EB1) 4-4
4.3.6 Enable Byte 0 (EB0) 4-4
4.3.7 Chip Selects (CS3, CS[2:0]) 4-4
4.3.8 Internal ROM Disable (MOD) 4-4
4.4 Exception Control Signals 4-4
4.4.1 Reset (RSTIN) 4-4
4.4.2 Low Voltage Reset (LVRSTIN) 4-5
4.4.3 Reset Out (RSTOUT) 4-5

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

TABLE OF CONTENTS

Paragraph	Title	Page
4.5	Clock Signals	4-5
4.5.1	Crystal Oscillator (XOSC, EXOSC)	4-5
4.5.2	Clock Input (CLKIN)	4-5
4.5.3	Clock Output (CLKOUT)	4-5
4.6	Debug and Emulation Support Signals	4-5
4.6.1	Test Clock (TCK)	4-5
4.6.2	Test Data Input (TDI)	4-5
4.6.3	Test Data Output (TDO)	4-6
4.6.4	Test Mode Select (TMS)	4-6
4.6.5	Test Reset ($\overline{\text{TRST}}$)	4-6
4.6.6	Debug Event ($\overline{\text{DE}}$)	4-6
4.6.7	Factory Test Mode (TEST)	4-6
4.7	External Interrupts/GPIO Signals	4-6
4.7.1	External Interrupts 7 – 0 (INT[7:0])	4-6
4.8	Keypad Signals	4-7
4.8.1	Column Strobes (COL[7:0])	4-7
4.8.2	Row Senses (ROW[7:0])	4-7
4.9	UART Module Signals	4-7
4.9.1	Receive Data (RxD0, RxD1)	4-7
4.9.2	Transmit Data ($\overline{\text{TxD0}}$, TxD1)	4-7
4.9.3	Clear to Send ($\overline{\text{CTS0}}$)	4-7
4.9.4	Request to Send ($\overline{\text{RTS0}}$)	4-8
4.10	Serial Peripheral Interface Module Signals	4-8
4.10.1	SPI Data Master Out/Slave In (SPI_MOSI)	4-8
4.10.2	SPI Data Master In/Slave Out (SPI_MISO)	4-8
4.10.3	SPI Serial Clock (SPI_CLK)	4-8
4.10.4	SPI Enable (SPI _EN)	4-8
4.10.5	SPI General-Purpose Output (SPI _GP)	4-8
4.11	Pulse Width Modulator Signals	4-8
4.11.1	PWM[5:0]	4-8
4.12	Power and Ground Pins	4-9
4.12.1	Positive Supply (V_{DD})	4-9
4.12.2	Ground (GND)	4-9
4.12.3	Standby Battery Power (V_{BATT})	4-9
4.12.4	Standby Power Filter (V_{STBY})	4-9

SECTION 5
ROM MODULE

5.1	Overview	5-1
5.2	Functional Description	5-1
5.3	Applications	5-2

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

TABLE OF CONTENTS

Paragraph	Title	Page

SECTION 6
STATIC RAM MODULE

6.1 Overview 6-1
6.2 Functional Description 6-1

SECTION 7
EXTERNAL INTERFACE MODULE

7.1 Overview 7-1
7.2 Signals 7-1
7.2.1 Address Bus 7-1
7.2.2 Data Bus 7-2
7.2.3 Read/Write 7-2
7.2.4 Control Signals 7-2
7.2.5 Boot Mode 7-2
7.2.6 Chip Select Outputs 7-2
7.3 Chip-Select Address Range 7-3
7.4 EIM Interface Example 7-3
7.5 EIM Functionality 7-4
7.5.1 Configurable Bus Sizing 7-4
7.5.2 External Boot ROM Control 7-6
7.5.3 Programmable Output Generation 7-6
7.5.4 Bus Watchdog Operation 7-6
7.5.5 Error Conditions 7-6
7.5.6 Show Cycles 7-7
7.6 EIM Programming Model 7-7
7.6.1 Chip-Select Control Registers 7-7
7.7 EIM Configuration Register 7-11
7.8 External Bus Timing Diagrams 7-13

SECTION 8
CLOCK MODULE AND LOW-POWER MODES

8.1 Overview 8-1
8.2 Low-Power Modes 8-4
8.2.1 CPU Core Low-Power Modes 8-4
8.2.2 Peripheral Behavior in Low-Power Modes 8-5
8.2.3 General Low-Power Features 8-7

SECTION 9
TIMER/RESET MODULE

9.1 Overview 9-1
9.2 Timer/Reset Programming Model 9-1
9.3 Reset Operation 9-2
9.3.1 Reset Pins 9-2

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

TABLE OF CONTENTS

Paragraph	Title	Page
9.3.2	Reset Sources	9-2
9.3.3	Reset Sequence	9-3
9.3.4	Reset Source/Chip Configuration Register (RSCR)	9-3
9.4	Time-of-Day Timer	9-4
9.4.1	TOD Operation	9-5
9.4.2	TOD in Low-Power Modes	9-5
9.4.3	Time-of-Day Control/Status Register (TODCSR)	9-5
9.4.4	TOD Seconds Register (TODSR)	9-6
9.4.5	TOD Fraction Register (TODFR)	9-6
9.4.6	TOD Seconds Alarm Register (TODSAR)	9-7
9.4.7	TOD Fraction Alarm Register (TODFAR)	9-7
9.5	Watchdog Timer	9-8
9.5.1	Watchdog Timing Specifications	9-9
9.5.2	Watchdog Timer after Reset	9-9
9.5.3	Watchdog Timer Service Operation	9-9
9.5.4	Watchdog Timer in Wait Mode	9-9
9.5.5	Watchdog Timer in Doze Mode	9-9
9.5.6	Watchdog Timer in Stop Mode	9-9
9.5.7	Watchdog Timer in Debug Mode	9-10
9.5.8	Watchdog Timer Programming Model	9-10
9.6	Interval Timer (PIT)	9-11
9.6.1	PIT Operation	9-12
9.6.2	PIT as a “Set-and-Forget” Timer	9-12
9.6.3	PIT as a “Free-Running” Timer	9-13
9.6.4	Interval Timer Registers	9-13
9.6.5	PIT Control/Status Register (ITCSR)	9-14
9.6.6	PIT Data Register (ITDR)	9-15
9.6.7	PIT Alternate Data Register (ITADR)	9-16
9.6.8	PIT in Low-Power Modes	9-16
9.6.9	PIT in Debug Mode	9-16

SECTION 10
INTERRUPT CONTROLLER

10.1	Overview	10-1
10.2	Interrupt Controller Programming Model	10-2
10.2.1	Interrupt Source Register (INTSRC)	10-2
10.2.2	Normal Interrupt Enable Register (NIER)	10-3
10.2.3	Fast Interrupt Enable Register (FIER)	10-3
10.2.4	Normal Interrupt Pending Register (NIPND)	10-4
10.2.5	Fast Interrupt Pending Register (FIPND)	10-5
10.2.6	Interrupt Request Input Assignments	10-5

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

TABLE OF CONTENTS

Paragraph **Title** **Page**

SECTION 11
UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER MODULE

11.1 Overview 11-1
11.2 UART Signals 11-2
11.2.1 RTS — Request to Send (UART0) 11-2
11.2.2 CTS — Clear to Send (UART0) 11-2
11.2.3 TXD — UART Transmit 11-3
11.2.4 RXD — UART Receive 11-3
11.3 Sub-Block Description 11-3
11.3.1 Transmitter 11-3
11.3.2 Receiver 11-3
11.3.3 Infrared Interface 11-4
11.3.4 16x Bit Clock Generator 11-4
11.3.5 General UART Definitions 11-4
11.4 UART Programming Model 11-5
11.4.1 UART Receive Register (URX) 11-7
11.4.2 UART Transmitter Register (UTX) 11-8
11.4.3 UART Control Register 1 (UCR1) 11-9
11.4.4 UART Control Register 2 (UCR2) 11-11
11.4.5 UART BRG Register (UBRGR) 11-13
11.4.6 UART Status Register (USR) 11-14
11.4.7 UART Test Register (UTS) 11-15
11.5 GPIO Pins and Registers 11-16
11.5.1 UART Port Control Register (UPCR) 11-16
11.5.2 UART Data Direction Register (UDDR) 11-16
11.5.3 UART Port Data Register (UPDR) 11-17
11.6 Data Sampling Technique on the Receiver 11-17
11.7 UART Operation in Low-Power System Modes 11-23
11.8 UART Operation in System Debug Mode 11-24

SECTION 12
INTERVAL MODE SERIAL PERIPHERAL INTERFACE

12.1 Overview 12-1
12.2 Operation 12-1
12.2.1 Manual (Master) Mode 12-2
12.2.2 Interval (Master) Mode 12-3
12.2.3 Slave Mode 12-3
12.3 Signal Descriptions 12-3
12.3.1 SPI_MISO (Master In, Slave Out) 12-3
12.3.2 SPI_MOSI (Master Out, Slave In) 12-4
12.3.3 SPI_EN 12-4
12.3.4 SPI_CLK 12-4
12.3.5 SPI_GP 12-4

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

TABLE OF CONTENTS

Paragraph	Title	Page
12.4	ISPI Programming Model	12-4
12.4.1	ISPI Data Register	12-5
12.4.2	ISPI Control Register	12-5
12.4.3	ISPI Interval Control Register	12-8
12.4.4	ISPI Status Register	12-8
12.5	ISPI Programming Examples	12-9
12.5.1	Manual Mode Example	12-9
12.5.2	Slave Mode Example	12-10
12.5.3	Interval Model Example	12-10
12.6	ISPI Operation in Low-Power System Modes	12-11
12.7	ISPI Operation in System Debug Mode	12-11

SECTION 13
EXTERNAL INTERRUPTS/GPIO (EDGE PORT)

13.1	Overview	13-1
13.2	Interrupt/General-Purpose I/O Pin Descriptions (INT[0:7])	13-1
13.3	Edge Port Programming Model	13-2
13.3.1	Edge Port Pin Assignment Register (EPPAR)	13-2
13.3.2	Edge Port Data Direction Register (EPDDR)	13-3
13.3.3	Edge Port Data Register (EPDR)	13-3
13.3.4	Edge Port Flag Register (EPFR)	13-4

SECTION 14
KEYPAD PORT

14.1	Overview	14-1
14.2	KPP Pin Description	14-2
14.2.1	Input Pins	14-2
14.2.2	Output Pins	14-2
14.3	KPP Programming Model	14-2
14.3.1	Keypad Control Register (KPCR)	14-2
14.3.2	Keypad Status Register (KPSR)	14-3
14.3.3	Keypad Data Direction Register (KDDR)	14-5
14.3.4	Keypad Data Register (KPDR)	14-5
14.4	Keypad Operation	14-6
14.4.1	Keypad Matrix Construction	14-6
14.4.2	Keypad Port Configuration	14-6
14.4.3	Keypad Matrix Scanning	14-6
14.4.4	Keypad Standby	14-7
14.4.5	Glitch Suppression on Keypad Inputs	14-7
14.4.6	Multiple Key Closures	14-8
14.4.7	Typical Keypad Configuration and Scanning Sequence	14-9

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

TABLE OF CONTENTS

Paragraph **Title** **Page**

SECTION 15
PULSE WIDTH MODULATOR

15.1 Overview 15-1
15.2 PWM Programming Model 15-2
15.2.1 PWM Control Register 15-4
15.2.2 PWM Period Register 15-6
15.2.3 PWM Width Register 15-7
15.2.4 PWM Counter Register 15-7
15.3 PWM Operating Range 15-8
15.4 PWM Operation in Low-Power System Modes 15-8

SECTION 16
OnCE™ DEBUG MODULE

16.1 Overview 16-1
16.2 Operation 16-1
16.3 OnCE Pins 16-3
16.3.1 Debug Serial Input (TDI) 16-3
16.3.2 Debug Serial Clock (TCK) 16-3
16.3.3 Debug Serial Output (TDO) 16-3
16.3.4 Debug Mode Select (TMS) 16-3
16.3.5 Test Reset ($\overline{\text{TRST}}$) 16-4
16.3.6 Debug Event ($\overline{\text{DE}}$) 16-4
16.4 OnCE Controller and Serial Interface 16-4
16.5 OnCE Interface Signals 16-5
16.5.1 Internal Debug Request Input ($\overline{\text{IDR}}$) 16-5
16.5.2 CPU Debug Request ($\overline{\text{DBGRQ}}$) 16-5
16.5.3 CPU Debug Acknowledge ($\overline{\text{DBGACK}}$) 16-5
16.5.4 CPU Breakpoint Request (BRKRQ) 16-5
16.5.5 CPU Address, Attributes (ADDR, ATTR) 16-5
16.5.6 CPU Status (PSTAT) 16-5
16.5.7 OnCE Debug Output ($\overline{\text{DEBUG}}$) 16-6
16.6 OnCE Controller Registers 16-6
16.6.1 OnCE Command Register (OCMR) 16-6
16.6.2 OnCE Control Register (OCR) 16-8
16.6.3 OnCE Status Register (OSR) 16-11
16.7 OnCE Decoder (ODEC) 16-12
16.8 Memory Breakpoint Logic 16-12
16.8.1 Memory Address Latch (MAL) 16-13
16.8.2 Breakpoint Address Base Registers (BABA, BABB) 16-14
16.8.3 Breakpoint Address Mask Registers (BAMA, BAMB) 16-14
16.8.4 Breakpoint Address Comparators 16-14
16.8.5 Memory Breakpoint Counters (MBCA, MBCB) 16-14
16.9 OnCE Trace Logic 16-14

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

TABLE OF CONTENTS

Paragraph	Title	Page
16.9.1	Trace Counter (OTC)	16-15
16.9.2	Trace Operation	16-15
16.10	Methods of Entering Debug Mode	16-16
16.10.1	Debug Request During RESET	16-16
16.10.2	Debug Request During Normal Activity	16-16
16.10.3	Debug Request During Stop, Doze, or Wait Mode	16-16
16.10.4	Software Request During Normal Activity	16-16
16.10.5	Enabling OnCE Trace Mode	16-16
16.10.6	Enabling OnCE Memory Breakpoints	16-17
16.11	Pipeline Information and Write-Back Bus Register	16-17
16.11.1	Program Counter Register (PC)	16-18
16.11.2	Instruction Register (IR)	16-18
16.11.3	Control State Register (CTL)	16-18
16.11.4	Write-Back Bus Register (WBBR)	16-19
16.11.5	Processor Status Register (PSR)	16-19
16.12	Instruction Address FIFO Buffer (PC FIFO)	16-20
16.12.1	Reserved Test Control Registers (Reserved, MEM_BIST, FTCR, LSRL)	16-21
16.13	Serial Protocol Description	16-21
16.13.1	OnCE Commands	16-21
16.14	Target Site Debug System Requirements	16-21
16.15	Interface Connector For JTAG/OnCE Serial Port	16-22

APPENDIX A
ELECTRICAL CHARACTERISTICS

A.1	Maximum Ratings	A-1
A.2	DC Electrical Specifications	A-1
A.3	Clock Input Specifications	A-2
A.4	AC Electrical Specifications	A-2
A.4.1	Reset, MOD Timing Specifications	A-2
A.4.2	External Interrupt Timing Specifications	A-3
A.4.3	EIM Timing Specifications	A-4
A.4.4	ISPI Timing Specifications	A-6
A.4.5	OnCE Timing Specifications	A-9

APPENDIX B
PACKAGING AND PIN ASSIGNMENTS

B.1	Overview	B-1

APPENDIX C
PROGRAMMING REFERENCE

C.1	Peripheral Module Address Assignment	C-1
C.2	Interrupt Controller Programming Model	C-2
C.2.1	Interrupt Source Register (INTSRC)	C-2
C.2.2	Normal Interrupt Enable Register (NIER)	C-2

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

TABLE OF CONTENTS

Paragraph	Title	Page
C.2.3	Fast Interrupt Enable Register (FIER)	C-3
C.2.4	Normal Interrupt Pending Register (NIPND)	C-4
C.2.5	Fast Interrupt Pending Register (FIPND)	C-4
C.3	Timer/Reset Programming Model	C-5
C.3.1	Reset Source/Chip Configuration Register (RSCR)	C-5
C.3.2	Time-of-Day Control/Status Register (TODCSR)	C-7
C.3.3	TOD Seconds Register (TODSR)	C-7
C.3.4	TOD Fraction Register (TODFR)	C-8
C.3.5	TOD Seconds Alarm Register (TODSAR)	C-8
C.3.6	TOD Fraction Alarm Register (TODFAR)	C-9
C.3.7	Watchdog Control Register (WCR)	C-9
C.3.8	Watchdog Service Register (WSR)	C-10
C.3.9	PIT Control/Status Register (ITCSR)	C-11
C.3.10	PIT Data Register (ITDR)	C-12
C.3.11	PIT Alternate Data Register (ITADR)	C-13
C.4	KPP Programming Model	C-13
C.4.1	Keypad Control Register (KPCR)	C-14
C.4.2	Keypad Status Register (KPSR)	C-14
C.4.3	Keypad Data Direction Register (KDDR)	C-15
C.4.4	Keypad Data Register (KPDR)	C-15
C.5	EIM Programming Model	C-16
C.5.1	Chip-Select Control Registers	C-16
C.5.2	EIM Configuration Register	C-20
C.6	PWM Module	C-22
C.6.1	PWM Control Register	C-23
C.6.2	PWM Period Register	C-25
C.6.3	PWM Width Register	C-26
C.6.4	PWM Counter Register	C-27
C.7	Edge Port Programming Model	C-27
C.7.1	Edge Port Pin Assignment Register (EPPAR)	C-27
C.7.2	Edge Port Data Direction Register (EPDDR)	C-28
C.7.3	Edge Port Data Register (EPDR)	C-29
C.7.4	Edge Port Flag Register (EPFR)	C-29
C.8	ISPI Programming Model	C-30
C.8.1	ISPI Send/Receive Data Register	C-30
C.8.2	ISPI Control Register	C-31
C.8.3	ISPI Interval Control Register	C-33
C.8.4	ISPI Status Register	C-34
C.9	UART Programming Model	C-34
C.9.1	UART Receive Register (URX)	C-36
C.9.2	UART Transmit Register (UTX)	C-37
C.9.3	UART Control Register 1 (UCR1)	C-38
C.9.4	UART Control Register 2 (UCR2)	C-40

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

TABLE OF CONTENTS

Paragraph	Title	Page
C.9.5	UART BRG Register (UBRGR)	C-42
C.9.6	UART Status Register (USR)	C-42
C.9.7	UART Test Register (UTSR)	C-43
C.9.8	UART Port Control Register (UPCR)	C-44
C.9.9	UART Data Direction Register (UDDR)	C-45
C.9.10	UART Port Data Register (UPDR)	C-45
C.10	OnCE Registers	C-46
C.10.1	OnCE Command Register (OCMR)	C-46
C.10.2	OnCE Control Register (OCR)	C-47
C.10.3	OnCE Status Register (OSR)	C-50
C.10.4	Memory Address Latch (MAL)	C-51
C.10.5	Breakpoint Address Base Registers (BABA, BABB)	C-51
C.10.6	Breakpoint Address Mask Registers (BAMA, BAMB)	C-51
C.10.7	Breakpoint Address Comparators	C-51
C.10.8	Memory Breakpoint Counters (MBCA, MBCB)	C-51
C.10.9	Program Counter Register (PC)	C-51
C.10.10	Instruction Register (IR)	C-52
C.10.11	Control State Register (CTL)	C-52
C.10.12	Write-Back Bus Register (WBBR)	C-53
C.10.13	Processor Status Register (PSR)	C-53
C.10.14	Reserved Test Control Registers (Reserved, MEM_BIST, FTCR, LSRL)	C-53

INDEX

RECORD OF CHANGES

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

LIST OF ILLUSTRATIONS

Paragraph	Title	Page
1-1	MMC2001 Block Diagram	1-2
2-1	Programming Model	2-4
2-2	Data Organization in Memory	2-5
2-3	Data Organization in Registers	2-5
2-4	Signal Relationships to Clocks	2-9
2-5	M•CORE Bus Signals	2-10
2-6	External Multiplexer Connections	2-13
4-1	Functional Signal Groups	4-1
7-1	EIM Block Diagram	7-1
7-2	EIM Interface to Memory and Peripherals	7-4
7-3	$\overline{CS0}$ Control Register	7-7
7-4	$\overline{CS1}$, $\overline{CS2}$, CS3 Control Registers	7-8
7-5	EIM Configuration Register	7-11
7-6	Read Memory Access (CSA = 0, WSC = 1)	7-14
7-7	Write Memory Access (CSA = 0, WSC = 1, WWS = 0)	7-15
7-8	Word Read Access from Halfword Width Memory	7-16
7-9	Word Write Access to Halfword Width Memory	7-17
7-10	Write after Read Memory Access (CSA = 0, WSC = 2, EDC = 0)	7-18
7-11	Write after Read Memory Access (CSA = 0, WSC = 1, EDC = 1)	7-19
7-12	Peripheral Read Access (CSA = 1, WSC = 5)	7-20
7-13	Peripheral Write Access (CSA = 1, WSC = 5)	7-21
7-14	Read and Write Fast Memory Access (CSA = 0, WSC = 0, WWS = 0)	7-22
8-1	MMC2001 Clock Module	8-3
9-1	Reset Functional Block Diagram	9-2
9-2	Reset Source Register	9-3
9-3	TOD Block Diagram	9-4
9-4	TOD Control/Status Register	9-5
9-5	TOD Seconds Register	9-6
9-6	TOD Fraction Register	9-7
9-7	TOD Seconds Alarm Register	9-7
9-8	TOD Fraction Alarm Register	9-8
9-9	Watchdog Timer Block Diagram	9-8
9-10	Watchdog Control Register	9-10
9-11	Watchdog Service Register	9-11
9-12	PIT Block Diagram	9-12
9-13	Starting a Count from an Off State	9-12
9-14	Counter Reloading from the Modulus Latch	9-13
9-15	Counter in Free-Running Mode	9-13
9-16	PIT Control and Status Register	9-14
9-17	PIT Data Register	9-15
9-18	PIT Alternate Data Register	9-16
10-1	Interrupt Source Register	10-2
10-2	Normal Interrupt Enable Register	10-3

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

LIST OF ILLUSTRATIONS

Paragraph	Title	Page
10-3	Fast Interrupt Enable Register	10-3
10-4	Normal Interrupt Pending Register	10-4
10-5	Fast Interrupt Pending Register	10-5
11-1	UART Channel Block Diagram	11-2
11-2	UART Receive Register	11-7
11-3	UART Transmitter Register	11-9
11-4	UART Control Register 1	11-9
11-5	UART Control Register 2	11-12
11-6	UART BRG Register	11-13
11-7	UART Status Register	11-14
11-8	UART Test Register	11-15
11-9	UART Port Control Register	11-16
11-10	UART Data Direction Register	11-16
11-11	UART Port Data Register	11-17
11-12	Start Bit — Ideal Case	11-19
11-13	Start Bit — Noise Case One	11-20
11-14	Start Bit — Noise Case Two	11-21
11-15	Start Bit — Noise Case Three	11-22
11-16	Start Bit — Noise Case Four	11-23
12-1	ISPI Channel Block Diagram	12-1
12-2	Timing Diagram of ISPI 8-Bit Operation	12-2
12-3	ISPI Data Register	12-5
12-4	ISPI Control Register	12-5
12-5	ISPI Interval Control Register	12-8
12-6	ISPI Status Register	12-8
13-1	External Interrupt/GPIO Block Diagram	13-1
13-2	Edge Port Pin Assignment Register	13-2
13-3	Edge Port Data Direction Register	13-3
13-4	Edge Port Data Register	13-3
13-5	Edge Port Flag Register	13-4
14-1	KPP Block Diagram	14-1
14-2	Keypad Control Register	14-3
14-3	Keypad Status Register	14-4
14-4	Keypad Data Direction Register	14-5
14-5	Keypad Data Register	14-5
14-6	Keypad Synchronizer Functional Diagram	14-8
14-7	Decoding Wrong Three Key Presses	14-9
15-1	PWM Block Diagram	15-1
15-2	PWM Generating Audio	15-1
15-3	PWM Prescaler	15-2
15-4	PWM Control Registers	15-4
15-5	PWM Period Registers	15-6
15-6	PWM Width Registers	15-7

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

LIST OF ILLUSTRATIONS

Paragraph	Title	Page
15-7	PWM Count Registers	15-7
16-1	OnCE Block Diagram	16-1
16-2	OnCE Controller	16-2
16-3	OnCE Controller and Serial Interface	16-4
16-4	OnCE Command Register	16-7
16-5	OnCE Control Register	16-8
16-6	OnCE Status Register	16-11
16-7	OnCE Memory Breakpoint Logic	16-13
16-8	OnCE Trace Logic Block Diagram	16-15
16-9	CPU Scan Chain Register (CPUSCR)	16-17
16-10	Control State Register	16-18
16-11	OnCE PC FIFO	16-20
16-12	Recommended Connector Interface to JTAG/OnCE Port	16-22
A-1	CLKIN Timing (for Square Wave Input)	A-2
A-2	Reset Timing	A-3
A-3	MOD Timing	A-3
A-4	External Interrupt Timing	A-4
A-5	EIM Read/Write Timing	A-5
A-6	SPI Slave Timing (PHA = 0)	A-7
A-7	SPI Slave Timing (PHA = 1)	A-7
A-8	SPI Manual/Interval Mode Timing (PHA = 0)	A-8
A-9	SPI Manual/Interval Mode Timing (PHA = 1)	A-8
A-10	Test Clock Input Timing	A-9
A-11	TRST Timing	A-9
A-12	Test Access Port Timing	A-10
B-1	144-Lead Plastic Thin Quad Flat Pack Pin Assignment	B-1
C-1	Interrupt Source Register	C-2
C-2	Normal Interrupt Enable Register	C-3
C-3	Fast Interrupt Enable Register	C-3
C-4	Normal Interrupt Pending Register	C-4
C-5	Fast Interrupt Pending Register	C-4
C-6	Reset Source Register	C-6
C-7	TOD Control/Status Register	C-7
C-8	TOD Seconds Register	C-8
C-9	TOD Fraction Register	C-8
C-10	TOD Seconds Alarm Register	C-9
C-11	TOD Fraction Alarm Register	C-9
C-12	Watchdog Control Register	C-10
C-13	Watchdog Service Register	C-11
C-14	PIT Control and Status Register	C-11
C-15	PIT Data Register	C-13
C-16	PIT Alternate Data Register	C-13
C-17	Keypad Control Register	C-14

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

LIST OF ILLUSTRATIONS

Paragraph	Title	Page
C-18	Keypad Status Register	C-14
C-19	Keypad Data Direction Register	C-15
C-20	Keypad Data Register	C-16
C-21	$\overline{CS0}$ Control Register	C-17
C-22	$\overline{CS1}$, $\overline{CS2}$, CS3 Control Registers	C-17
C-23	EIM Configuration Register	C-21
C-24	PWM Control Registers	C-23
C-25	PWM Period Registers	C-26
C-26	PWM Width Registers	C-26
C-27	PWM Count Registers	C-27
C-28	Edge Port Pin Assignment Register	C-28
C-29	Edge Port Data Direction Register	C-28
C-30	Edge Port Data Register	C-29
C-31	Edge Port Flag Register	C-29
C-32	ISPI Data Register	C-30
C-33	ISPI Control Register	C-31
C-34	ISPI Interval Control Register	C-33
C-35	ISPI Status Register	C-34
C-36	UART Receive Register	C-36
C-37	UART Transmit Register	C-37
C-38	UART Control Register 1	C-38
C-39	UART Control Register 2	C-40
C-40	UART BRG Register	C-42
C-41	UART Status Register	C-42
C-42	UART Test Register	C-44
C-43	UART Port Control Register	C-44
C-44	UART Data Direction Register	C-45
C-45	UART Port Data Register	C-45
C-46	OnCE Command Register	C-46
C-47	OnCE Control Register	C-47
C-48	OnCE Status Register	C-50
C-49	Control State Register	C-52

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

LIST OF TABLES

Paragraph	Title	Page
2-1	M•CORE Instruction Set	2-6
2-2	M•CORE Bus Signals	2-11
2-3	Interface Requirements for Read and Write Cycles	2-12
2-4	Termination Result Summary	2-14
3-1	MMC2001 Module Address Map	3-1
3-2	MMC2001 Address Map	3-2
4-1	Pin Requirements in 144-Pin Package	4-2
5-1	ROM Module Address Map	5-1
6-1	Static RAM Module Address Map	6-1
7-1	Chip Select Address Range	7-3
7-2	Interface Requirements for Read and Write Cycles	7-5
7-3	EIM Memory Map	7-7
7-4	Wait State Control Field Settings	7-9
7-5	Data Port Size Field Settings	7-10
7-6	Show Cycle Enable Field Settings	7-13
8-1	CPU Core and Peripherals Clock Source	8-1
8-2	CPU Core and Peripherals in Low-Power Modes	8-7
9-1	Timer/Reset Module Address Map	9-1
10-1	Interrupt Controller Address Map	10-2
10-2	Interrupt Source Assignment	10-6
11-1	UART Module Address Map	11-6
11-2	TxFL Field Settings	11-9
11-3	RxFL Field Settings	11-10
11-4	UART Pins GPIO Assignment	11-16
11-5	UART Low-Power Mode Operation	11-23
12-1	ISPI Module Address Map	12-4
12-2	BAUD RATE Field Settings	12-7
12-3	CLOCK COUNT Field Settings	12-7
12-4	ISPI Low-Power Mode Operation	12-11
13-1	GPIO Edge Port Address Map	13-2
13-2	EPPAx Field Settings	13-3
14-1	Keypad Port Column Modes	14-2
14-2	Keypad Port Address Map	14-2
15-1	PWM Address Map	15-3
15-2	CLK SEL Field Settings	15-6
15-3	PWM Range at 16 MHz	15-8
15-4	PWM Low-Power Mode Operation	15-8
16-1	OnCE Register Addressing	16-8
16-2	Sequential Control Field Settings	16-9
16-3	Memory Breakpoint Control Field Settings	16-10
16-4	Processor Mode Field Settings	16-12
A-1	Maximum Ratings	A-1
A-2	DC Electrical Specifications	A-1

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

LIST OF TABLES

Paragraph	Title	Page
A-3	Clock Input Specifications	A-2
A-4	Reset, MOD Timing Specifications	A-2
A-5	External Interrupt Timing Specifications	A-3
A-6	EIM Timing Specifications	A-4
A-7	ISPI Timing Specifications	A-6
A-8	OnCE Timing Specifications	A-9
C-1	MMC2001 Address Map	C-1
C-2	Interrupt Controller Address Map	C-2
C-3	Timer/Reset Module Address Map	C-5
C-4	Keypad Port Address Map	C-13
C-5	EIM Address Map	C-16
C-6	Wait State Control Field Settings	C-18
C-7	Data Port Size Field Settings	C-19
C-8	Chip-Select Address Range	C-20
C-9	Show Cycle Enable Field Settings	C-22
C-10	PWM Address Map	C-22
C-11	Clock Select Field Values	C-25
C-12	GPIO Edge Port Address Map	C-27
C-13	EPPAx Field Settings	C-28
C-14	Interval Mode Serial Peripheral Interface Address Map	C-30
C-15	BAUD RATE Values	C-32
C-16	CLOCK COUNT Values	C-33
C-17	UART Module Address Map	C-35
C-18	TxFL Field Settings	C-38
C-19	RxFL Field Settings	C-39
C-20	OnCE Register Addressing	C-47
C-21	Sequential Control Field Definition	C-48
C-22	Memory Breakpoint Control Field Definition	C-49
C-23	Processor Mode Field Definition	C-51

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 1
INTRODUCTION

The MMC2001 integrated processor incorporates the following functional units:

- M•CORE™ Integer Processor
  - — 32-bit RISC architecture
  - — Low power, high performance
- On-chip, 256-Kbyte ROM
- On-chip, 32-Kbyte SRAM with battery backup supply support
- Interrupt Controller
  - — Support for up to 32 interrupt sources
- External Interface Module (EIM)
  - — Transfers information between the MMC2001 and external memory or peripherals
  - — 22 address lines
  - — 16 data lines
  - — Chip select and wait state generation
  - — Bus watchdog timer
- Timer/Reset Module
  - — Crystal oscillator: generates the master clock signal for the time-of-day timer from a 32.768-kHz external crystal
  - — Time-of-day timer: provides time-of-day information as well as an alarm clock function
  - — Watchdog timer: resets the chip to recover from system failure
  - — Reset unit: provides low voltage detection input and backup power switching for SRAM and the time-of-day timer
  - — Periodic interrupt timer
- Universal Asynchronous Receiver/Transmitter Module (UART)
  - — Two independent UART channels
  - — Asynchronous operation
  - — Baud rate generation
  - — Infrared (IR) interface support
- 16-bit general-purpose I/O port with support for keyboard scan/encode
- 8-bit general-purpose I/O port with support for edge/level sensitive external interrupts
- Pulse-Width Modulation Module (PWM)
  - — Six independent PWM channels
  - — Programmable period
  - — Programmable duty cycle
  - — Periodic interrupt capability
  - — Pins can be configured as general-purpose I/O
- Interval Mode Serial Peripheral Interface (ISPI)
  - — Efficient communication with slower serial peripherals

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

— Designed for master/slave SPI operation
— Interval-mode SPI operation
• OnCE™ Debug Module

As a low-voltage part, the MMC2001 operates at voltages between 2.0 and 3.3 volts. It is particularly suited for use in battery-powered applications.

The internal logic and external I/O buffers are provided with independent power supply connections to allow 3.3-V I/O levels while operating internal logic at 2.0 V for lower power consumption.

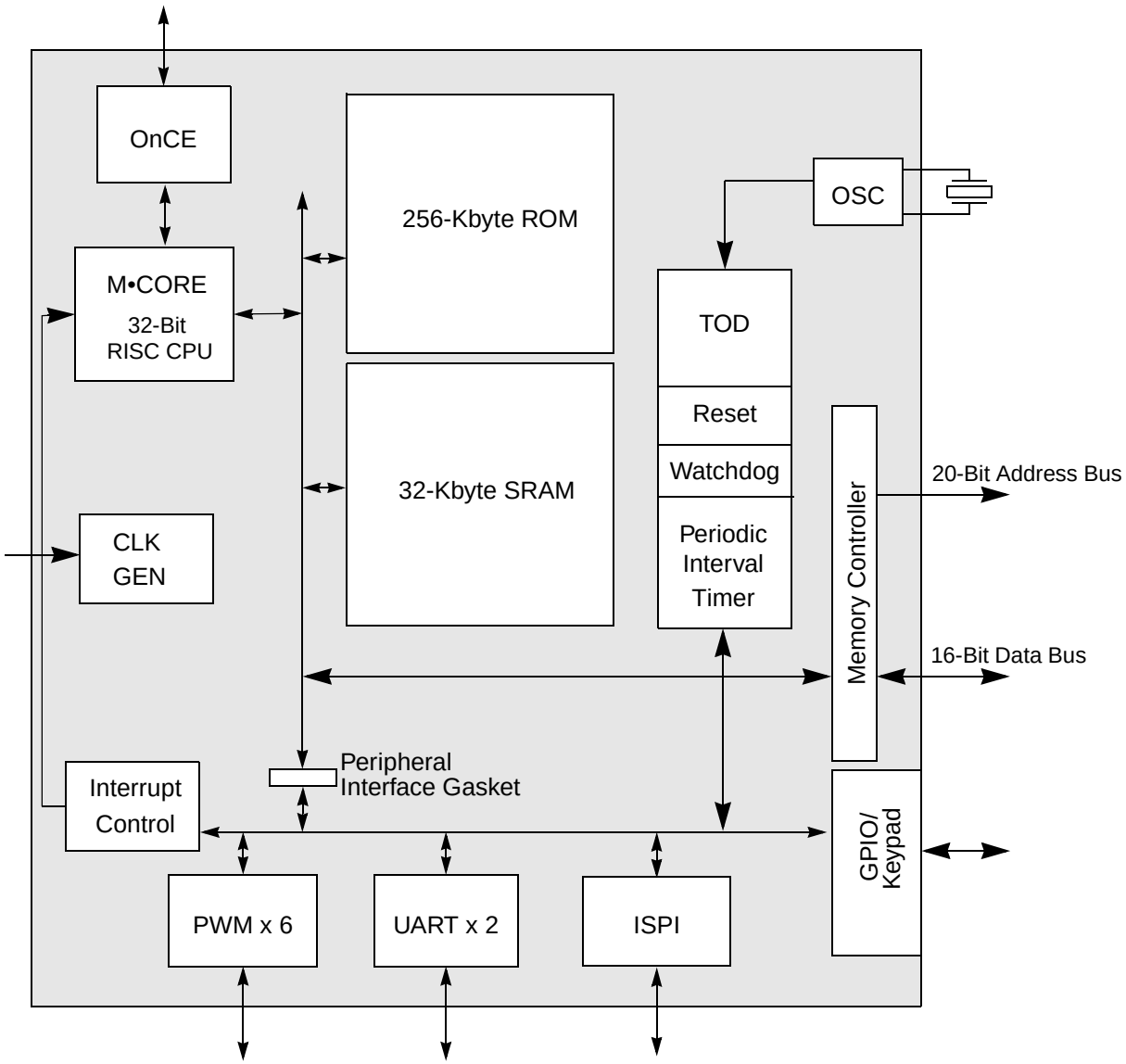


Figure 1-1 MMC2001 Block Diagram

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

SECTION 2
INTEGER CPU

This section gives a short description of the M•CORE CPU features and some basic bus interface information.

2.1 M•CORE Overview

The 32-bit M•CORE microRISC engine represents a new family of Motorola microprocessor core products. The processor architecture has been designed for high-performance and cost-sensitive embedded control applications, with particular emphasis on reduced system power consumption. This makes the M•CORE suitable for battery-operated, portable products, as well as for highly integrated parts designed for a high temperature environment.

Total system power consumption is dictated by various components in addition to the processor core. In particular, memory power consumption (both on-chip and external) is expected to dominate overall power consumption of the core-plus-memory subsystem. With this factor in mind, the instruction set architecture (ISA) for M•CORE makes the trade-off of absolute performance capability versus total energy consumption in favor of reducing the overall energy consumption, while maintaining an acceptably high level of performance at a given clock frequency.

M•CORE is a streamlined execution engine that provides many of the same performance enhancements as mainstream reduced instruction set computer (RISC) designs. Fixed length instruction encoding and a strict load/store architecture minimize control complexity and overhead. The goal of minimizing the overhead of memory system energy consumption is achieved by adopting a (relatively) short 16-bit instruction encoding. This choice significantly lowers the memory bandwidth needed to sustain a high rate of instruction execution.

Code density statistics for a number of applications show relative code density competitive in comparison to complex instruction set computer (CISC) designs, and implementation statistics show a large reduction in complexity and overhead relative to a CISC approach.

In addition to substantial cost and performance benefits, M•CORE also offers advantages in power consumption and power management. M•CORE minimizes power dissipation by using a fully static design, dynamic power management, and low-voltage operation. The M•CORE automatically powers-down internal functional blocks that are not needed on a clock-by-clock basis. Power conservation modes are also provided for absolute power conservation on a coarser granularity.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

2.2 Features

The main features of the M•CORE are as follows:

- 32-bit load/store RISC architecture
- Fixed 16-bit instruction length
- 16-entry, 32-bit general-purpose register file
- Efficient 4-stage execution pipeline, hidden from application software
- Single-cycle instruction execution for many instructions
- Two cycles for taken branches and memory access instructions
- Support for byte, halfword, and word memory accesses
- Fast interrupt support with 16-entry dedicated alternate register file
- Vectored and autovectored interrupt support

2.3 Microarchitecture Summary

The M•CORE instruction execution pipeline consists of the following stages:

- Instruction fetch
- Instruction decode/register file read
- Execute
- Register writeback

These stages operate in an overlapped fashion, allowing single-clock instruction execution for most instructions.

Sixteen general-purpose registers are provided for source operands and instruction results. Register R15 is used as the link register to hold the return address for subroutine calls, and register R0 is associated with the current stack pointer value by convention.

The execution unit consists of a 32-bit arithmetic/logic unit (ALU), a 32-bit barrel shifter, a find-first-one unit (FFO), result feed-forward hardware, and miscellaneous support hardware for multiplication and multiple register loads and stores. Arithmetic and logical operations are executed in a single cycle with the exception of the multiply, signed divide, and unsigned divide instructions. The multiply instruction is implemented with a 2-bit per clock, overlapped-scan, modified Booth algorithm with early-out capability to reduce execution time for operations with small multiplier values. The signed divide and unsigned divide instructions also have data-dependent timing. A find-first-one unit operates in a single clock cycle.

The program counter unit has a PC incrementer and a dedicated branch address adder to minimize delays during change of flow operations. Branch target addresses are calculated in parallel with branch instruction decode, with a single pipeline bubble for taken branches and jumps. This results in an execution time of two clocks. Conditional branches that are not taken execute in a single clock.

Memory load and store operations are provided for byte, halfword, and word (32-bit) data with automatic zero extension of byte and halfword load data. These instructions can execute in two clock cycles. Load and store multiple register instructions allow low overhead context save and restore operations. These instructions can execute in (N+1) clock cycles, where N is the numbers of registers to transfer.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

A single condition code/carry (C) bit is provided for condition testing and for use in implementing arithmetic and logical operations greater than 32 bits. Typically, the C-bit is set only by explicit test/comparison operations, not as a side-effect of normal instruction operation. Exceptions to this rule occur for specialized operations for which it is desirable to combine condition setting with actual computation.

A 16-entry alternate register file is provided to support low overhead interrupt exception processing. The CPU supports both vectored and autovectored interrupts.

2.4 Programming Model

The M•CORE programming model is defined separately for two privilege modes: supervisor and user. Certain operations are not available in user mode.

User programs can only access registers specific to the user mode; system software executing in the supervisor mode can access all registers, using the control registers to perform supervisory functions. User programs are thus restricted from accessing privileged information. The operating system performs management and service tasks for the user programs by coordinating their activities.

Most instructions execute in either mode, but some instructions that have important system effects are privileged and can only execute in the supervisor mode. For instance, user programs cannot execute the **stop**, **doze**, or **wait** instructions. To prevent a user program from entering the supervisor mode except in a controlled manner, instructions that can alter the S bit in the program status register (PSR) are privileged. The **trap #n** instructions provide controlled access to operating system services for user programs. Access to special control registers is also precluded in user mode.

When the S bit in the PSR is set, the processor executes instructions in the supervisor mode. Bus cycles associated with an instruction indicate either supervisor or user access depending on the mode.

The processor uses the user programming model during normal user mode processing. During exception processing, the processor changes from user to supervisor mode. Exception processing saves the current value of the PSR in the EPSR or FPSR shadow control register and then sets the S bit in the PSR, forcing the processor into the supervisor mode. To return to the previous operating mode, a system routine may execute the **rte** (return from exception) or **rfi** (return from fast interrupt) instruction as appropriate, causing the instruction pipeline to be flushed and refilled from the appropriate address space.

The registers depicted in the programming model (see **Figure 2-1**) provide operand storage and control. The registers are partitioned into two levels of privilege: user and supervisor. The user programming model consists of 16 general-purpose 32-bit registers, the 32-bit program counter (PC) and the condition/carry (C) bit. The C bit is implemented as bit 0 of the PSR. This is the only portion of the PSR accessible by the user. The supervisor programming model consists of 16 additional 32-bit general-purpose registers (the alternate file), as well as a set of status/control registers and scratch registers. By convention, register R15 serves as the link register for subroutine calls, and register R0 is typically used as the current stack pointer.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

The alternate file is selected for use via a control bit in the PSR. The status, control, and scratch registers are accessed via the move from control register (**mfcr**) and move to control register (**mtcr**) instructions. When the alternate file is selected via the AF bit in the PSR, general-purpose operands are accessed from it. When the AF bit is cleared, operands are accessed from the normal file. This alternate file is provided to allow very low overhead context switching capability for real-time event handling.

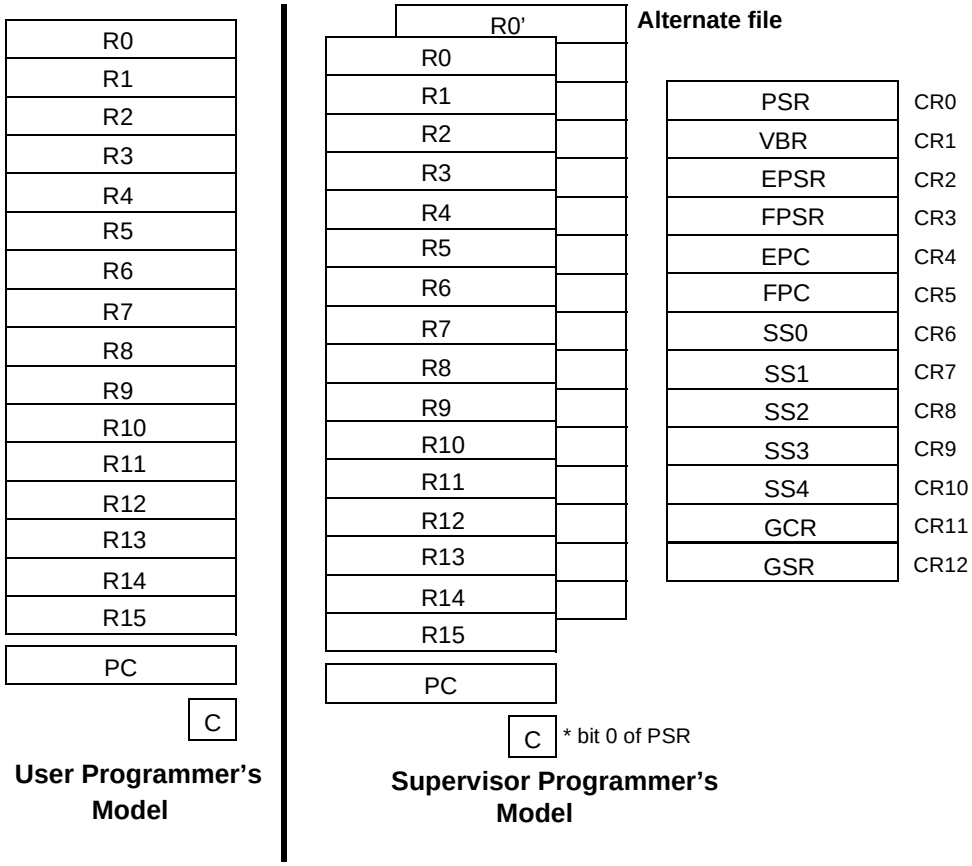


Figure 2-1 Programming Model

The supervisor programming model includes the PSR, which contains operation control and status information. In addition, a set of exception shadow registers is provided to save the state of the PSR and the program counter at the time an exception occurs. A separate set of shadow registers is provided for fast interrupt support to minimize context saving overhead.

Five scratch registers are provided for supervisor software use in handling exception events. A single register is provided to alter the base address of the exception vector table. Two registers are provided for global control and status.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

2.5 Data Format Summary

The operand data formats supported by the integer unit are standard two's-complement data formats. The operand size for each instruction is either explicitly encoded in the instruction (load/store instructions) or implicitly defined by the instruction operation (index operations, byte extraction). Typically, instructions operate on all 32 bits of the source operand(s) and generate a 32-bit result.

Memory is viewed from a big-endian byte ordering perspective. The most significant byte (byte 0) of word 0 is located at address 0. Bits are numbered within a word starting with bit 31 as the most significant bit.

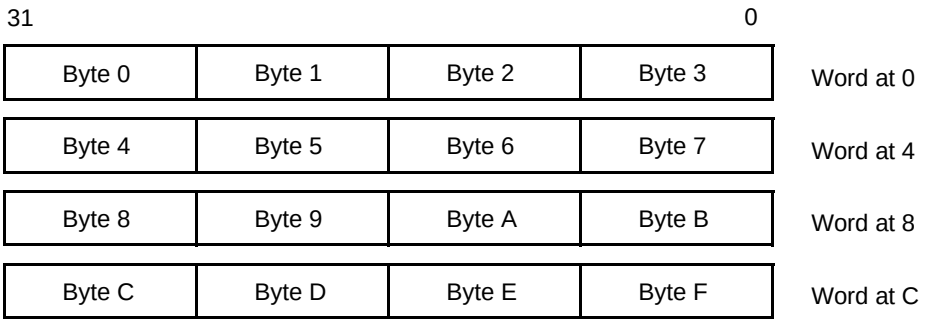


Figure 2-2 Data Organization in Memory

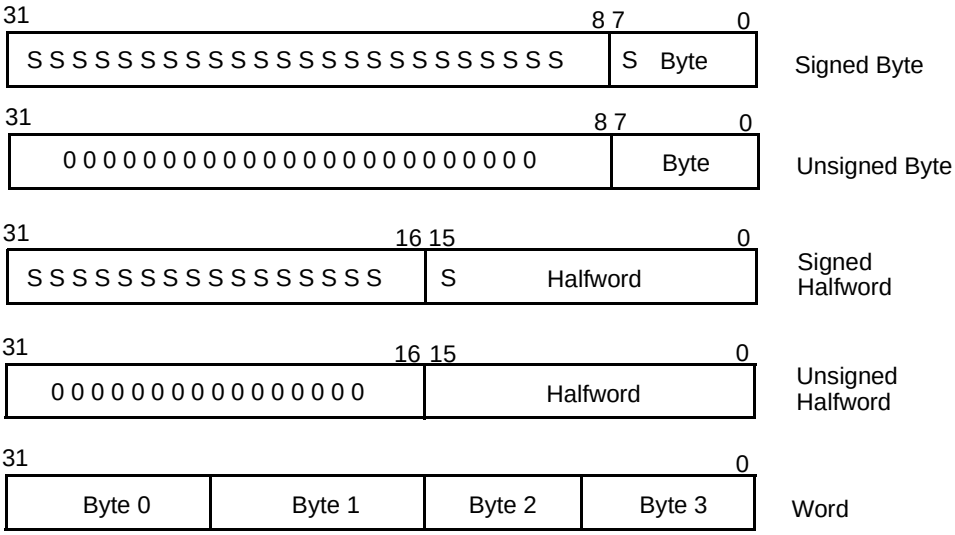


Figure 2-3 Data Organization in Registers

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

2.6 Operand Addressing Capabilities

M•CORE accesses all memory operands through load and store instructions, transferring data between the general-purpose registers and memory. Register-plus-four-bit scaled displacement addressing mode is used for the load and store instructions to address byte, halfword, or word (32-bit) data.

Load and store multiple instructions allow a subset of the 16 GPRs to be transferred to or from a base address pointed to by register R0 (the default stack pointer by convention).

Load and store register quadrant instructions use register indirect addressing to transfer a register quadrant to or from memory.

2.7 Instruction Set Overview

The instruction set is tailored to support high-level languages and is optimized for those instructions most commonly executed. A standard set of arithmetic and logical instructions is provided, as well as instruction support for bit operations, byte extraction, data movement, control flow modification, and a small set of conditionally executed instructions which can be useful in eliminating short conditional branches.

Table 2-1 is an alphabetized listing of the M•CORE instruction set. Refer to the M•CORE Reference Manual (MCORERM/AD) for more details on instruction operation.

Table 2-1 M•CORE Instruction Set

Mnemonic	Description
ABS	Absolute Value
ADDC	Add with C Bit
ADDI	Add Immediate
ADDU	Add Unsigned
AND	Logical AND
ANDI	Logical AND Immediate
ANDN	AND NOT
ASR	Arithmetic Shift Right
ASRC	Arithmetic Shift Right, Update C Bit
BCLRI	Bit Clear Immediate
BF	Branch on Condition False
BGENI	Bit Generate Immediate
BGENR	Bit Generate Register
BKPT	Breakpoint
BMASKI	Bit Mask Immediate
BR	Branch
BREV	Bit Reverse
BSETI	Bit Set Immediate
BSR	Branch to Subroutine
BT	Branch on Condition True
BTSTI	Bit Test Immediate

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 2-1 M•CORE Instruction Set (Continued)

Mnemonic	Description
CLRF CLRT CMPHS CMPLT CMPLTI CMPNE CMPNEI	Clear Register on Condition False Clear Register on Condition True Compare Higher or Same Compare Less Than Compare Less Than Immediate Compare Not Equal Compare Not Equal Immediate
DECF DECGT DECLT DECNE DECT DIVS DIVU DOZE	Decrement on Condition False Decrement Register and Set Condition if Result Greater Than Zero Decrement Register and Set Condition if Result Less Than Zero Decrement Register and Set Condition if Result Not Equal to Zero Decrement on Condition True Divide Signed Integer Divide Unsigned Integer Doze
FF1	Find First One
INCF INCT IXH IXW	Increment on Condition False Increment on Condition True Index Halfword Index Word
JMP JMPI JSR JSRI	Jump Jump Indirect Jump to Subroutine Jump to Subroutine Indirect
LD.[BHW] LDM LDQ LOOPT LRW LSL, LSR LSLC, LSRC LSLI, LSRI	Load Load Multiple Registers Load Register Quadrant Decrement with C-Bit Update and Branch if Condition True Load Relative Word Logical Shift Left and Right Logical Shift Left and Right, Update C Bit Logical Shift Left and Right by Immediate
MFCR MOV MOVI MOVF MOVT MTCR MULT MVC MVCV	Move from Control Register Move Move Immediate Move on Condition False Move on Condition True Move to Control Register Multiply Move C Bit to Register Move Inverted C Bit to Register
NOT	Logical Complement
OR	Logical Inclusive-OR
ROTLI RSUB RSUBI RTE RFI	Rotate Left by Immediate Reverse Subtract Reverse Subtract Immediate Return from Exception Return from Interrupt

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 2-1 M•CORE Instruction Set (Continued)

Mnemonic	Description
SEXTB SEXTH ST.[BHW] STM STQ STOP SUBC SUBU SUBI SYNC	Sign-Extend Byte Sign-Extend Halfword Store Store Multiple Registers Store Register Quadrant Stop Subtract with C Bit Subtract Subtract Immediate Synchronize
TRAP TST TSTNBZ	Trap Test Operands Test for No Byte Equal Zero
WAIT	Wait
XOR XSR XTRB0 XTRB1 XTRB2 XTRB3	Exclusive OR Extended Shift Right Extract Byte 0 Extract Byte 1 Extract Byte 2 Extract Byte 3
ZEXTB ZEXTH	Zero-Extend Byte Zero-Extend Halfword

2.8 M•CORE Bus Interface

The M•CORE bus is a synchronous pipelined interface. Signals driven on this bus are required to meet the set up and hold time relative to the falling and rising edges of the bus clock.

The M•CORE architecture supports byte, half-word, and word operands, allowing access to 8-,16-, and 32-bit data ports through the use of synchronous cycles controlled by the size outputs (TSIZ0, TSIZ1).

M•CORE bus interface features are summarized below.

- 32-bit address bus with transfer size indication
- 32-bit data bus
- Signals referenced to both the rising and falling edges of the bus clock
- Only aligned transfers allowed
- M•CORE is the only bus master; no arbitration support
- 32-bit fixed port size

2.8.1 Bus Characteristics

The bus transfers information between the M•CORE and external memory or a peripheral device via the external and internal bus interfaces. The M•CORE port size is fixed at 32 bits. External devices can accept or provide eight or 16 bits in parallel and must follow the handshake protocol described in this section. The number of bits accepted or provided during a bus transfer is defined as the transfer size. The M•CORE uses the address bus to specify the address for the transfer and the data bus to transfer the data. Control and attribute signals indicate the beginning and type

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

of the cycle as well as the address space and size of the transfer. The selected device then controls the length of the cycle with the signal(s) used to terminate the cycle. Access requests are generated in an overlapped fashion in order to support sustained single-cycle transfers.

Inputs to the M•CORE are sampled synchronously and must be stable during the sample windows defined in **Figure 2-4**. If an input makes a transition during the window time period, the level recognized by the M•CORE is not predictable.

Outputs from the M•CORE change on one of the two clock edges, depending on the signal class.

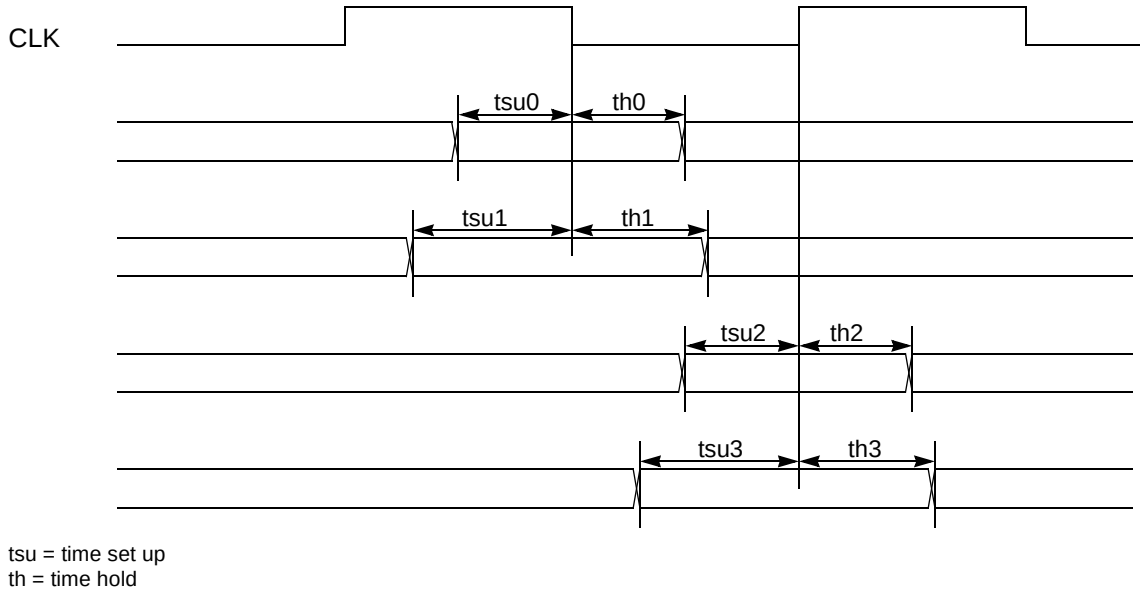


Figure 2-4 Signal Relationships to Clocks

2.8.2 Bus Signals

Figure 2-5 shows the M•CORE bus signals arranged by functional group.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

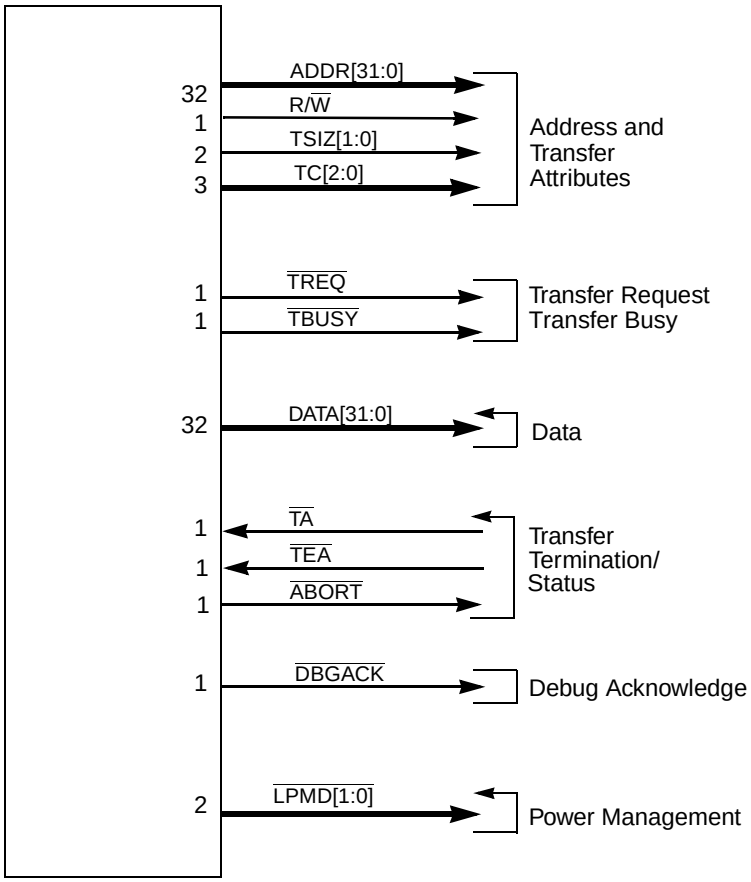


Figure 2-5 M•CORE Bus Signals

2.8.3 Signal Descriptions

Table 2-2 lists and describes the bus interface signals. More detailed descriptions can be found in subsequent sections. Signal direction is relative to the M•CORE.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 2-2 M•CORE Bus Signals

Signal Name	Pins	Active	I/O	Description
Address and Transfer Attributes				
ADDR[31:0] Address Bus	32	High	O	Driven by the M•CORE to specify the physical address of the bus transaction.
R/$\overline{W}$ Read/Write	1	High	O	Driven by the M•CORE along with the address. Driven high indicates that a read access is in progress. Driven low indicates that a write access is in progress.
TSIZ[1:0] Transfer Size	2	High	O	Driven by the M•CORE along with the address. Specifies the data transfer size for the transaction.
TC[2:0] Transfer Code	3	High	O	Driven by the M•CORE along with the address. Indicates the type of access for the current bus cycle.
Transfer Request/Transfer Busy				
$\overline{TREQ}$ Transfer Request	1	Low	O	Driven by the M•CORE along with the address and transfer attributes to indicate that a new access has been requested.
TBUSY Transfer Busy	1	Low	O	Driven by the M•CORE to indicate that an access is in progress. This signal is driven for the duration of a cycle and may be held asserted for multiple transfers.
Data				
DATA[31:0] Data Bus	32	High	O	Driven by the M•CORE when it “owns” the bus and it initiated a write transaction to a slave device. Eight (byte), 16 (halfword), or 32 (word) bits of data can be transferred per access.
			I	Driven by the slave in a read transaction. Eight (byte), 16 (halfword), or 32 (word) bits of data can be transferred per access.
Transfer Cycle Termination and Status				
$\overline{TA}$ Transfer Acknowledge	1	Low	I	Driven by the slave device to which the current transaction was addressed. Indicates that the slave has received the data on the write cycle or returned data on the read cycle.
$\overline{TEA}$ Transfer Error Acknowledge	1	Low	I	Driven by the slave device to which the current transaction was addressed. Indicates that an error condition has occurred during the bus cycle.
$\overline{ABORT}$ Abort	1	Low	O	Driven by the M•CORE to indicate that the transfer is to be aborted immediately.
Power Management				
$\overline{LPMD[1:0]}$ Low-Power Modes	2	Low	O	Driven by the M•CORE to indicate whether the core is running in normal mode or has just executed a low power mode instruction.
Debug				
$\overline{DBGACK}$ Debug Mode	1	Low	O	Driven by the M•CORE to indicate that debug mode has been entered.

2.8.4 Bus Operation

The following sections provide a functional description of the system bus, the signals that control it, and the bus cycles provided for data transfer operations. They also describe the error conditions and reset operation.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

The M•CORE receives a clock input (CLK) from an external clock source and generates two internal clocks (C1 and C2). The CLK input sets the frequency of operation for the bus interface directly. The clock source monitors the M•CORE low power mode outputs ($\overline{\text{LPMD}}$[1:0]) and controls the clock input to the M•CORE accordingly by forcing the clocks low for low-power operation.

Data transfers occur between an internal register and the external bus. The internal register connects to the external data bus through the internal data bus and a data multiplexer. The data multiplexer establishes the necessary connections for different combinations of address and data sizes. This multiplexer is physically positioned in the overall system to minimize power consumption by minimizing loading and reducing unnecessary signal transitions. Logically, however, it is considered part of the M•CORE.

The M•CORE does not support dynamic bus sizing and expects the referenced device to accept the requested access width. Peripherals with an interface width of N bits should not define internal registers greater than N bits wide.

Misaligned transfers are not supported. The M•CORE interface may drive the ADDR[1:0] address lines to a value which is not representative of an aligned transfer, but expects aligned data to be transferred. ADDR[1:0] should be selectively ignored by external logic, depending on the size of the transfer.

The data multiplexer takes the four bytes of the core interface data bus and routes them to their required positions to interface properly to memory or peripherals. The external multiplexer connections to memory are controlled on a byte granularity and are referred to as MB0–MB3, where MB0 resides at byte address 0 (mod 4) and MB3 resides at byte address 3 (mod 4). For example, MB0 would normally be routed to DATA[31:24] on a word transfer, but it can also be routed to DATA[7:0] for supporting a byte data transfer. The same is true for any of the other operand bytes.

Figure 2-6 shows the connection requirements for the multiplexer. The transfer size (TSIZ[1:0]) and byte offset (ADDR1 and ADDR0) signals determine the positioning of the bytes (see **Table 2-3**).

Table 2-3 Interface Requirements for Read and Write Cycles

Transfer Size					Active Interface Bus Sections				Mux Connections
	TSIZ1	TSIZ0	ADDR1	ADDR0	DATA[31:24]	DATA[23:16]	DATA[15:8]	DATA[7:0]	
Byte	0	1	0	0	—	—	—	MB0	a
	0	1	0	1	—	—	—	MB1	b
	0	1	1	0	—	—	—	MB2	c
	0	1	1	1	—	—	—	MB3	d
Halfword	1	0	0	X	—	—	MB0	MB1	e
	1	0	1	X	—	—	MB2	MB3	f
Word	0	0	X	X	MB0	MB1	MB2	MB3	g

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

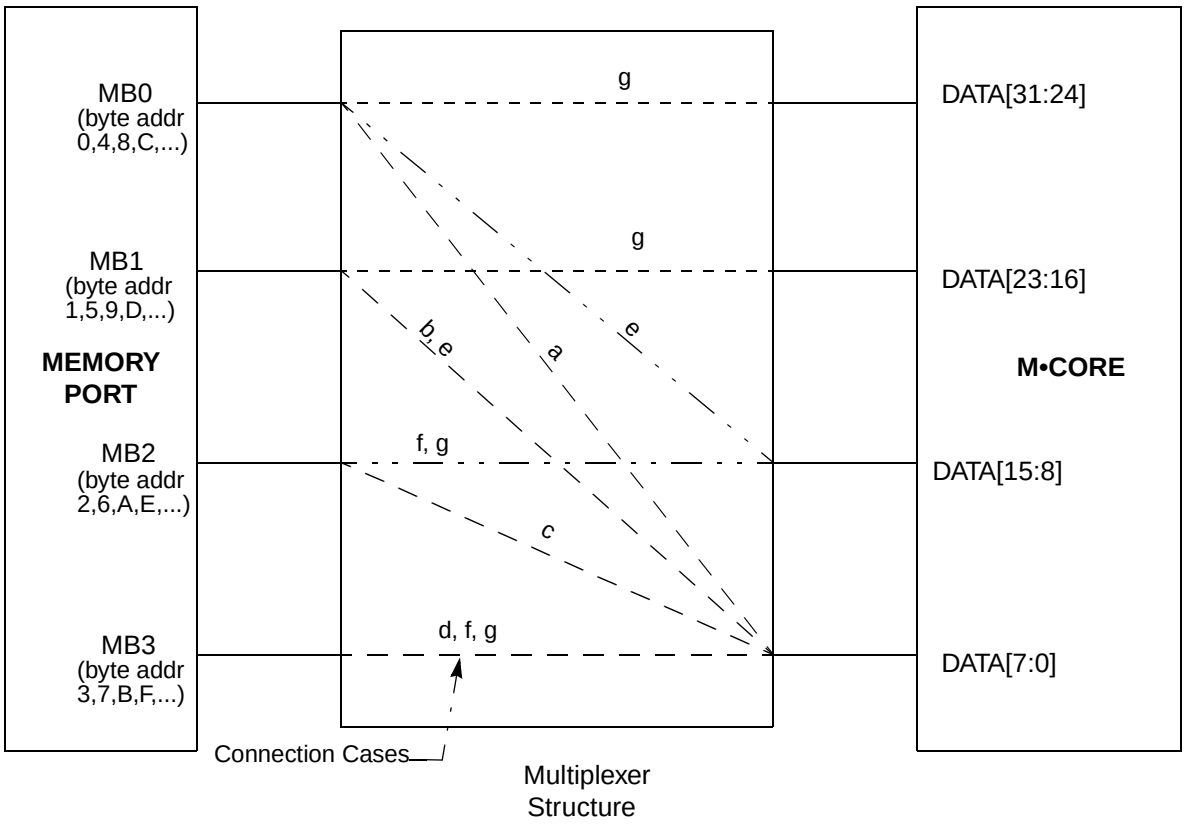


Figure 2-6 External Multiplexer Connections

Table 2-3 lists the combinations of the TSIZx, ADDR1, and ADDR0 signals that are used for each possible transfer size and alignment. In **Table 2-3**, MB0–MB3 indicate the portion of the requested operand that is read or written during that bus transfer. For word transfers, all bytes are valid as listed and correspond to portions of the requested operand. The bytes labeled with a dash are not required; they are ignored on read transfers and driven with undefined data on write transfers.

2.8.5 Processor Instruction/Data Transfers

The transfer of data between the processor and other devices involves the address bus, data bus, transfer attributes, and control signals. The address and data buses are parallel, non-multiplexed buses that support aligned byte, halfword, and word transfers. All bus input and output signals are sampled or driven with respect to one of the edges of the CLK signal. The M•CORE moves data on the bus by issuing control signals and using a handshake protocol to ensure correct data movement.

Access requests are generated in an overlapped fashion in order to support sustained single-cycle transfers. Once an access has been accepted, the processor is free to change the current request. Access information must therefore be latched by a slave device.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

2.8.6 Bus Exception Cycles

The M•CORE bus interface requires assertion of $\overline{TA}$ from an external device to signal that a bus cycle is complete. External circuitry can provide $\overline{TEA}$ when no device responds to indicate that an error condition is associated with an access. This allows the cycle to terminate and the processor to enter exception processing for the error condition if appropriate.

To control termination of a bus cycle for a bus error condition properly, $\overline{TA}$ and $\overline{TEA}$ must be asserted and negated about the same rising edge of CLK.

Table 2-4 Termination Result Summary

$\overline{TA}$	$\overline{TEA}$	Result
Don't Care	Low	Bus Error — terminate and take bus error exception if appropriate
Low	High	Normal cycle terminate and continue
High	High	Insert wait states

The system hardware can use the $\overline{TEA}$ signal to abort the current bus cycle when a fault is detected. When the processor recognizes a bus error condition for an access, the access is terminated immediately.

When a bus cycle is terminated with a bus error, the M•CORE can enter access error exception processing immediately following the bus cycle, or it can defer processing the exception. The instruction pre-fetch mechanism requests instruction words from the instruction memory unit before it is ready to execute them. If a bus error occurs on an instruction fetch, the processor does not take the exception until it attempts to use the instruction. If an intervening instruction causes a branch or if a task switch occurs, the access error exception for the unused access does not occur.

A bus error termination for any write or read access that references data specifically requested by the execution unit causes the processor to begin exception processing immediately.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 3
SYSTEM MEMORY MAP

3.1 Overview

This section describes the address allocation conventions for the MMC2001 system.

The general address map shown in **Table 3-1** is recommended for all members of the MMC2001 architecture. Accesses to unimplemented regions of the map may result in TEA termination to the processor.

Table 3-1 MMC2001 Module Address Map

Address Range	Use	Supervisor Access	User Access
00000000 – 00000003	Off-chip boot ROM vector fetch (MOD asserted)	Full	—
00000004 – 0FFFFFFF	On-chip ROM		Selective
10000000 – 1FFFFFFF	On-chip peripherals		None
20000000 – 2FFFFFFF	Off-chip devices		Selective
30000000 – 3FFFFFFF	On-chip RAM		Selective
40000000 – FFFFFFFF	Reserved		—

For the initial implementations of MMC2001, the address range of 0x4000 0000 – 0xFFFF FFFF is reserved for future use. Accesses to this range result in a transfer error termination to the CPU.

Note that in the current MMC2001 implementation, accesses to on-chip devices other than RAM and ROM are considered privileged; no user mode access is allowed to on-chip peripherals.

3.2 Peripheral Module Address Allocation

The register blocks for all on-chip peripheral devices are located on 4096-byte boundaries. Peripherals that require additional address space (e.g., for buffers) are assigned additional 4-Kbyte blocks. In this case, peripherals are located on a 2^n address boundary corresponding to the size of the block. Within a 4-Kbyte block, peripheral registers may be incompletely decoded, such that the register map repeats throughout the entire block; or the peripheral may return undefined values for unimplemented register addresses. The description of the peripherals in this document provide information regarding the result of accesses to unimplemented registers.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

3.3 Peripheral Module Interface Operation

Interface requirements for peripherals are defined to simplify the hardware interface implementation while providing a reasonable and extendable software model. The following requirements are currently defined (others may be added in the future):

- A given peripheral device appears only in the 4-Kbyte region(s) allocated to it.
- For on-chip devices, registers are defined to be 16 or 32 bits wide. For registers that do not implement all 32 bits, the unimplemented bits return zero when read, and writes to unimplemented bits have no effect. In general, unimplemented bits should be written to zero to ensure future compatibility.
- All peripherals define the exact results for 32-bit, 16-bit, and 8-bit accesses. These may vary according to individual peripheral definitions. In any event, misaligned accesses are not supported, nor is bus sizing performed for accesses to registers smaller than the access size.

3.4 Peripheral Module Address Assignment

The register maps of all peripheral devices for MMC2001 are located on 4096-byte boundaries. **Table 3-2** defines the address assignment for the on-chip components.

Table 3-2 MMC2001 Address Map

Address Range (Hex)	Use	Access
00000000 – 0003FFFF	On-Chip ROM Array	Supervisor, Selective User
00040000 – 000FFFFF	ROM Echoes	Supervisor, Selective User
00100000 – 0FFFFFFF	Not Used (Access causes transfer error)	—
10000000 – 10000FFF	Interrupt Controller	Supervisor Only
10001000 – 10001FFF	Timer/Reset Unit	Supervisor Only
10002000 – 10002FFF	Not Used (Access causes transfer error)	—
10003000 – 10003FFF	Keypad Port	Supervisor Only
10004000 – 10004FFF	External Interface Module	Supervisor Only
10005000 – 10005FFF	Pulse-width Modulator	Supervisor Only
10006000 – 10006FFF	Not Used (Access causes transfer error)	—
10007000 – 10007FFF	GPIO Edge Port	Supervisor Only
10008000 – 10008FFF	Interval SPI	Supervisor Only
10009000 – 10009FFF	UART 0	Supervisor Only
1000A000 – 1000AFFF	UART 1	Supervisor Only
1000B000 – 1FFFFFFF	Not Used (Access causes transfer error)	—
20000000 – 2FFFFFFF	External Devices	Supervisor, Selective User
30000000 – 30007FFF	On-Chip RAM Array	Supervisor, Selective User
30008000 – 3000FFFF	RAM Echoes	Supervisor, Selective User
30100000 – 40000000	Not Used (Access causes transfer error)	—

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 4
SIGNAL DESCRIPTIONS

4.1 Overview

This section contains brief descriptions of MMC2001 signal and power connections. Specific details of signal operation are provided in the individual module descriptions. **Figure 4-1** shows MMC2001 signals by functional group.

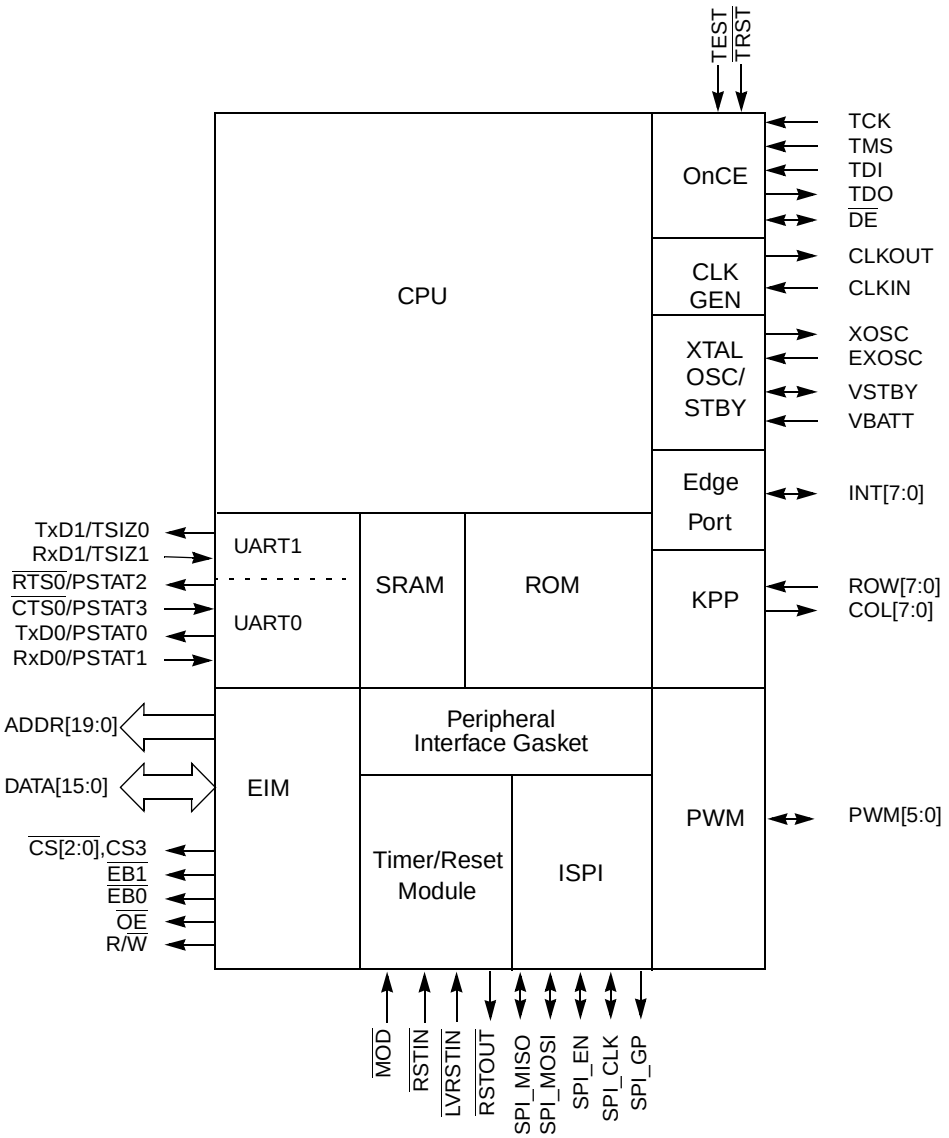


Figure 4-1 Functional Signal Groups

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

4.2 Signal Index

Table 4-1 Pin Requirements in 144-Pin Package

Name	Alt	Qty	Dir	Pull-Up[1]	Reset[2]	Use
External Bus						
ADDR[19:0]	—	20	O	—	OL	Address bus
DATA[15:0]	—	16	I/O	Keeper	I	Data bus
R/$\overline{W}$	—	1	O	—	H	Read/write enable
$\overline{EB0}$	—	1	O	—	H	Enable data bus byte 0 (DATA[15:8])
$\overline{EB1}$	—	1	O	—	H	Enable data bus byte 1 (DATA[7:0])
$\overline{OE}$	—	1	O	—	H	Output enable
Chip Selects						
$\overline{CS0}$	—	1	O	—	CS0	Chip select for external Flash
$\overline{CS1}$	GPO	1	O	—	GPOH	Chip select for external RAM
$\overline{CS2}$	GPO	1	O	—	GPOH	Chip select (spare)
CS3	GPO	1	O	—	GPOL	Chip select for external LCD
Clock, Reset, and Miscellaneous						
CLKIN	—	1	I	—	—	Programmable clock input
CLKOUT	—	1	O	—	OL	Programmable clock output
XOSC	—	1	O	—	—	32.768-kHz XTAL output
EXOSC	—	1	I	—	—	32.768-kHz XTAL input
$\overline{MOD}$	—	1	I	—	—	Boot ROM control
$\overline{RSTOUT}$	—	1	O	—	—	Resets external components
$\overline{RSTIN}$	—	1	I	—	—	Initiates system reset
$\overline{LVRSTIN}$	—	1	I	—	—	Low voltage supply switching control
Debug and Test Port Control						
TMS	—	1	I	47-K pull-up	—	Test mode select
TDI	—	1	I	47-K pull-up	—	Test data input
TDO	—	1	O	—	—	Test data output
TCK	—	1	I	47-K pull-up	—	Test clock input
$\overline{TRST}$	—	1	I	47-K pull-up	—	Initiates test controller reset
$\overline{DE}$	—	1	I/O	47-K pull-up	—	Initiates or acknowledges debug mode entry
TEST	—	1	I	100-K pulldown	—	Factory test mode
Keypad and Edge Port						
COL[7:0]	GPIO	8	I/O	—	GPI	Column 7 – 0
ROW[7:0]	GPIO	8	I/O	47-K pull-up	GPI	Row 7 – 0
INT[7:0]	GPIO	8	I/O	—	GPI	External interrupts 7 – 0

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 4-1 Pin Requirements in 144-Pin Package (Continued)

Name	Alt	Qty	Dir	Pull-Up[1]	Reset[2]	Use
UART Port						
TXD0	GPIO/PSTAT0	1	I/O	—	GPI	RS-232 TX for CH0 or PSTAT0 output
RXD0	GPIO/PSTAT1	1	I/O	—	GPI	RS-232 RX for CH0 or PSTAT1output
$\overline{\text{RTS0}}$	GPIO/PSTAT2	1	I/O	—	GPI	RS-232 RTS for CH0 or PSTAT2 output
$\overline{\text{CTS0}}$	GPIO/PSTAT3	1	I/O	—	GPI	RS-232 CTS for CH0 or PSTAT3 output
TXD1	GPIO/TSIZ0	1	I/O	—	GPI	RS-232 TX for CH1 or TSIZ0 output
RXD1	GPIO/TSIZ1	1	I/O	—	GPI	RS-232 RX for CH1 or TSIZ1 output
ISPI Port						
SPI_MOSI	—	1	I/O	—	—	Master output, slave input
SPI_MISO	—	1	I/O	—	—	Master input, slave output
SPI_EN	—	1	I/O	—	—	SPI enable
SPI_CLK	—	1	I/O	—	—	SPI clock
SPI_GP	—	1	O	—	—	General-purpose control signal
PWM Port						
PWM[5:0]	GPIO	6	I/O	—	GPI	PWM inputs
Power Supplies						
AVDD	—	3	I	—	—	External memory address supply
AGND	—	3	I	—	—	External memory address GND
CVDD	—	1	I	—	—	External memory control/Debug/JTAG supply
CGND	—	1	I	—	—	External memory control/Debug/JTAG GND
DVDD	—	2	I	—	—	External memory data supply
DGND	—	2	I	—	—	External memory data GND
FVDD	—	1	I	—	—	Clock/Clock outputs supply
FGND	—	1	I	—	—	Clock/Clock outputs GND
GVDD	—	2	I	—	—	Keypad port/Interrupts supply
GGND	—	2	I	—	—	Keypad port/Interrupts GND
HVDD	—	1	I	—	—	UART/ISPI supply
HGND	—	1	I	—	—	UART/ISPI GND
JVDD	—	1	I	—	—	PWM supply
JGND	—	1	I	—	—	PWM GND
XVDD	—	1	I	—	—	Oscillator V_{DD}
XGND	—	1	I	—	—	Oscillator GND
VBATT	—	1	I	—	—	Standby battery supply
VSTBY	—	1	I	—	—	Standby filter cap
QVCC	—	4	I	—	—	Internal supply
QVCCH	—	4	I	—	—	Quiet supply high for I/O internal logic
QGND	—	4	I	—	—	Quiet GND

NOTES:
1. All pull-ups and pulldowns are disconnected when the pin is programmed as an output.
2. GPIO = “General-Purpose Input/Output”, GPO = “General-Purpose Output”, GPI = “General-Purpose Input”.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

4.3 Bus Signals

This section describes the interface signals for external memory and peripherals.

4.3.1 Address Bus (ADDR[19:0])

The output address bus pins are used to address external devices. To minimize power dissipation, they do not change state unless external memory is being accessed.

4.3.2 Data Bus (DATA[15:0])

These pins provide the bidirectional data bus for external memory accesses. DATA[15:0] are held in their previous logic state when there is no external bus activity. This is accomplished with weak “keepers” inside the I/O buffers. They are also kept in their previous state during hardware reset.

4.3.3 Output Enable ($\overline{OE}$)

This active-low output signal indicates the bus access is a read and enables slave devices to drive the data bus.

4.3.4 Read/$\overline{Write}$ Enable (R/$\overline{W}$)

This active-low output signal indicates whether the current bus access is a read or write.

4.3.5 Enable Byte 1 ($\overline{EB1}$)

This active-low output pin is active during an operation to data bits DATA[7:0]. It may be configured to assert for both read and write cycles, or for write cycles only.

4.3.6 Enable Byte 0 ($\overline{EB0}$)

This active-low output pin is active during an operation to data bits DATA[15:8]. It may be configured to assert for both read and write cycles, or for write cycles only.

4.3.7 Chip Selects (CS3, $\overline{CS[2:0]}$)

These output pins provide chip selects to external devices.

4.3.8 Internal ROM Disable ($\overline{MOD}$)

This active-low input pin provides the capability of disabling the on-chip ROM and forcing $\overline{CS0}$ to be used to select an external boot ROM.

4.4 Exception Control Signals

4.4.1 Reset ($\overline{RSTIN}$)

This active-low input signal is used to initiate a system reset. An external reset signal resets the MMC2001 and most internal peripherals. The debug module is unaffected by $\overline{RSTIN}$; this function is provided through the $\overline{TRST}$ pin.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

4.4.2 Low Voltage Reset ($\overline{\text{LVRSTIN}}$)

This active-low input signal is used to initiate a system reset and to cause the backup power supply source to be selected for the RAM array and the OSC/time-of-day timer. This external reset signal resets the MMC2001 and most internal peripherals. The debug module is unaffected by $\overline{\text{LVRSTIN}}$; this function is provided through the $\overline{\text{TRST}}$ pin.

4.4.3 Reset Out ($\overline{\text{RSTOUT}}$)

This active-low output signal is used to reset external components. This pin is asserted when any internal reset source is active.

4.5 Clock Signals

These signals are used by the MMC2001 for controlling or generating the system clocks. See **SECTION 8 CLOCK MODULE AND LOW-POWER MODES** for more information on the various clocking methods and frequencies.

4.5.1 Crystal Oscillator (XOSC, EXOSC)

These pins are the connections for an external crystal to the internal oscillator circuit for generation of the internal LOW_REFCLK. EXOSC is the input pin, and XOSC is the output.

4.5.2 Clock Input (CLKIN)

This input pin provides the HI_REFCLK clock source to the CPU and internal peripherals.

4.5.3 Clock Output (CLKOUT)

This output pin provides an external clock source, either the LO_REFCLK or the HI_REFCLK.

4.6 Debug and Emulation Support Signals

These signals are used for testing or serial debugging. See **SECTION 16 OnCE™ DEBUG MODULE** for more information on these signals and debug mode.

4.6.1 Test Clock (TCK)

The test clock input (TCK) pin is the test clock used to synchronize the JTAG test logic. The TCK pin has an internal 47-Kbyte pull-up resistor.

4.6.2 Test Data Input (TDI)

The test data input (TDI) pin is the serial input for test instructions and data. TDI is sampled on the rising edge of TCK and it has an internal 47-Kbyte pull-up resistor.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

4.6.3 Test Data Output (TDO)

The test data output (TDO) pin is the serial output for test instructions and data. TDO is three-stateable and is actively driven in the shift-IR and shift-DR controller states. TDO changes on the falling edge of TCK.

4.6.4 Test Mode Select (TMS)

The test mode select input (TMS) pin is used to sequence the test controller's state machine. The TMS is sampled on the rising edge of TCK. It has an internal 47-Kbyte pull-up resistor.

4.6.5 Test Reset ($\overline{\text{TRST}}$)

The active-low Schmitt trigger input pin $\overline{\text{TRST}}$ is used to initialize the test controller asynchronously. The $\overline{\text{TRST}}$ pin has an internal 47-Kbyte pull-up resistor.

4.6.6 Debug Event ($\overline{\text{DE}}$)

$\overline{\text{DE}}$ is an open drain, bidirectional, active low pin. As an input, it provides a means of entering debug mode from an external command controller. As an output, it provides a means of acknowledging that the CPU has entered debug mode.

This pin, when asserted as an input, causes the CPU to finish the current instruction being executed, save the instruction pipeline information, enter debug mode, and wait for commands to be entered from the serial debug input line. This pin is asserted as an output for several clock cycles when the CPU enters debug mode as a result of a debug request or as a result of meeting a breakpoint condition.

If used to enter debug mode, $\overline{\text{DE}}$ must be negated after the OnCE responds with an acknowledge and before sending the first OnCE command. The $\overline{\text{DE}}$ pin has an internal 47-Kbyte pull-up resistor.

4.6.7 Factory Test Mode (TEST)

This input is used to select factory test mode. A description of the functionality of this pin is detailed in the factory test document. The TEST pin has an internal 100-Kbyte pulldown resistor.

4.7 External Interrupts/GPIO Signals

These pins provide the software with general-purpose access external to the chip. See **SECTION 13 EXTERNAL INTERRUPTS/GPIO (EDGE PORT)** for more information on these signals.

4.7.1 External Interrupts 7 – 0 (INT[7:0])

These bidirectional pins comprise the external interface to the GPIO module.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

4.8 Keypad Signals

4.8.1 Column Strobes (COL[7:0])

These are general-purpose I/O pins used as keypad column strobes. They are open-drain selectable in software. The default state at reset is general-purpose input.

4.8.2 Row Senses (ROW[7:0])

These are general-purpose I/O pins used as keypad row senses. The default state at reset is general-purpose input. On-chip 47-Kbyte pull-up resistors are connected to these pins.

4.9 UART Module Signals

The following signals are used by the UART module for data and clock signals. They are also shared for provision of internal processor status. See **SECTION 11 UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER MODULE** for more information on these signals. See **7.7 EIM Configuration Register** for provision of internal status signal operation.

4.9.1 Receive Data (RxD0, RxD1)

These signals are used for receiver data input for UART channels 0 and 1, respectively. Data is sampled on the rising edge of the clock source, with the least significant bit received first.

RxD0 is used to provide the PSTAT1 internal status signal when the PSTEN bit in the EIM configuration register is set. In this case it operates as an active-high PSTAT1 output.

RxD1 is used to provide the TSIZ1 internal status signal when the SZEN bit in the EIM configuration register is set. In this case it operates as an active-high TSIZ1 output.

4.9.2 Transmit Data (TxD0, TxD1)

These signals are used for transmit data output for UART channels 0 and 1, respectively. The output is held high (“mark” condition) when the transmitter is disabled, idle, or operating in the local loopback mode. Data is shifted out at the falling edge of the clock source with the least significant bit transmitted first.

TxD0 is used to provide the PSTAT0 internal status signal when the PSTEN bit in the EIM configuration register is set. In this case it operates as an active-high PSTAT0 output.

TxD1 is used to provide the TSIZ0 internal status signal when the SZEN bit in the EIM configuration register is set. In this case it operates as an active-high TSIZ0 output.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

4.9.3 Clear to Send ($\overline{\text{CTS0}}$)

This active-low signal can be programmed as the clear-to-send output for UART channel 0.

$\overline{\text{CTS0}}$ is used to provide the PSTAT3 internal status signal when the PSTEN bit in the EIM configuration register is set. In this case it operates as an active-high PSTAT3 output.

4.9.4 Request to Send ($\overline{\text{RTS0}}$)

This active-low signal can be programmed as the request-to-send input for UART channel 0.

$\overline{\text{RTS0}}$ is used to provide the PSTAT2 internal status signal when the PSTEN bit in the EIM configuration register is set. In this case it operates as an active-high PSTAT2 output.

4.10 Serial Peripheral Interface Module Signals

The following signals are used by the serial peripheral module. See **SECTION 12 INTERVAL MODE SERIAL PERIPHERAL INTERFACE** for more information on these signals.

4.10.1 SPI Data Master Out/Slave In (SPI_MOSI)

This signal is the serial data output from the SPI in master mode and the serial data input in slave mode.

4.10.2 SPI Data Master In/Slave Out (SPI_MISO)

This signal is the serial data input to the ISPI in master mode and the serial data output in slave mode.

4.10.3 SPI Serial Clock (SPI_CLK)

This I/O signal is the serial shift clock for the SPI.

4.10.4 SPI Enable (SPI _EN)

This I/O signal is the peripheral chip select pin in master mode and is a slave enable in slave mode.

4.10.5 SPI General-Purpose Output (SPI _GP)

This output signal is used as a general-purpose control signal for external logic or devices.

4.11 Pulse Width Modulator Signals

See **SECTION 15 PULSE WIDTH MODULATOR** for more information on these signals.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

4.11.1 PWM[5:0]

These pins provide the external interface to the PWM block. They may be configured as general-purpose I/O if the PWM output function is not needed. The default state at reset is general-purpose input.

4.12 Power and Ground Pins

These pins provide system power and ground to the MMC2001. Multiple pins are provided for adequate current capability. All power supply pins must have adequate bypass capacitance for high-frequency noise suppression.

4.12.1 Positive Supply (V_{DD})

This pin supplies positive power to the chip.

4.12.2 Ground (GND)

This pin is the negative supply (ground) to the chip.

4.12.3 Standby Battery Power (V_{BATT})

This pin supplies positive battery standby power to the chip.

4.12.4 Standby Power Filter (V_{STBY})

This pin is used to provide an external filter capacitor connection for the standby voltage switching logic.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 5
ROM MODULE

5.1 Overview

The ROM module provides 256 Kbytes of general-purpose code and data storage.

5.2 Functional Description

The 256-Kbyte ROM module supports byte, halfword, and word read accesses with a 32-bit data interface to the CPU. Only the requested bytes are guaranteed to be valid on accesses. Write cycles that are attempted to the ROM address space are terminated with a $\overline{TEA}$ response to the CPU.

Table 5-1 ROM Module Address Map

Address	Use	Access
00000000 to 0003FFFF	ROM Array 00000001 to 00000003 are used as off-chip boot vectors when the $\overline{MOD}$ signal is asserted	Supervisor, Selective User
00040000 to 000FFFFF	ROM echoes on 256-Kbyte boundaries	Supervisor, Selective User
00100000 to 0FFFFFFF	Not Used (Access causes transfer error)	Not Applicable

The ROM module base address is located at address 0x0000 0000 when the module is enabled. Echoing of the ROM block occurs throughout the region 0x0004 0000 – 0x000F FFFF; no attempt is made to detect this condition. Accesses in the range of 0x0010 0000 – 0x0FFF FFFF result in $\overline{TEA}$ termination to the CPU.

Software designers should be aware that the echoing characteristics of this implementation may change for future versions of the M•CORE family.

External control is provided to disable the on-chip ROM for system debugging purposes via a control input to the chip. When asserted, the $\overline{MOD}$ input signal causes the on-chip ROM to be disabled for the initial program counter fetch out of reset, and the chip select module to dedicate the $\overline{CS0}$ output for an external boot ROM. Refer to **SECTION 4 SIGNAL DESCRIPTIONS** and **SECTION 7 EXTERNAL INTERFACE MODULE** for details of the interaction of the $\overline{MOD}$ input signal with the $\overline{CS0}$ signal.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

The SPROM control bit in the EIM configuration register allows selective access protection to be applied to the ROM. (See **7.7 EIM Configuration Register**.) This bit may be used to control read access to the ROM based on the state of the M•CORE PSR(S) bit. Attempted cycles to a protected ROM address space are terminated with a $\overline{\text{TEA}}$ response to the CPU.

5.3 Applications

The ROM module can be used to store:

- Reset boot code
- Frequently accessed code
- Table of constants
- Revision and identification registers for all MMC2001 peripherals
- Self-test diagnostic code

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 6
STATIC RAM MODULE

6.1 Overview

The static RAM (SRAM) module provides 32 Kbytes of general-purpose code and data storage.

6.2 Functional Description

The 32-Kbyte SRAM module supports byte, halfword, and word accesses with a 32-bit data interface to the CPU. Only the requested bytes are guaranteed to be valid on read accesses. The SRAM acknowledges all accesses to its memory space.

Table 6-1 Static RAM Module Address Map

Address	Use	Access
30000000 to 30007FFF	RAM Array	Supervisor, Selective User
30008000 to 3000FFFF	RAM Echoes on 32-Kbyte Boundaries	Supervisor, Selective User
30100000 to 40000000	Not Used (Access causes transfer error)	Not Applicable

The SRAM module occupies physical addresses 0x3000 0000 – 0x3000 7FFF. Echoing of the SRAM block occurs throughout the region 0x3000 8000 – 0x300F FFFF; no attempt is made to detect this condition. Accesses in the range of 0x3010 0000 – 0x3FFF FFFF result in $\overline{\text{TEA}}$ termination to the CPU.

Software designers should be aware that the echoing characteristics of this implementation may change for future versions of the M•CORE family.

The SPRAM control bit in the EIM configuration register allows selective access protection to be applied to the RAM. (See **7.7 EIM Configuration Register**.) This bit may be used to control access to the RAM based on the state of the M•CORE PSR(S) bit. Attempted cycles to a protected RAM address space are terminated with a $\overline{\text{TEA}}$ response to the CPU.

The SRAM module is partitioned into two independent blocks for the purposes of battery backup. An external standby power pin is provided to power both sections of the SRAM for data retention.

The SRAM contents are undefined immediately following a power-on reset.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 7
EXTERNAL INTERFACE MODULE

7.1 Overview

The external interface module (EIM) handles the interface to devices external to the MMC2001, including generation of chip selects for external peripherals and memory. It provides the following features:

- Four chip selects for external devices, each covering a range of 16 Mbytes
- Programmable wait-state generator for each chip select
- Selectable protection for each chip select
- Programmable data port size for each chip select
- Control for external/internal boot ROM device selection
- Bus watchdog counter for all bus cycles
- Programmable general output capability for unused chip select outputs
- Show cycles to allow internal bus cycles to be monitored externally

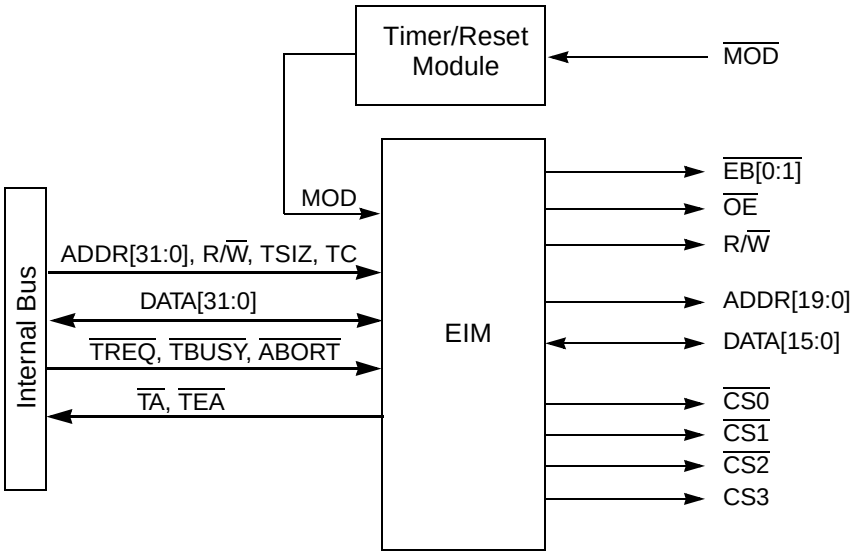


Figure 7-1 EIM Block Diagram

7.2 Signals

7.2.1 Address Bus

The ADDR[19:0] signals are address bus outputs used to address external devices.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

7.2.2 Data Bus

The DATA[15:0] signals are bidirectional data bus pins used to transfer data between the chip and an external device.

7.2.3 Read/Write

The R/$\overline{W}$ output signal indicates whether the current bus access is a read or write cycle. A high (logic one) level indicates a read cycle, and a low (logic zero) level indicates a write cycle.

7.2.4 Control Signals

The $\overline{OE}$ and $\overline{EB}$[0:1] signals are used to control the interface to the data bus.

7.2.4.1 Output Enable ($\overline{OE}$)

This active-low output signal indicates the bus access is a read and enables slave devices to drive the data bus with read data.

7.2.4.2 Enable Byte 0–1 ($\overline{EB[0:1]}$)

These active-low output pins indicate active data bytes for the current access. They can be programmed in the chip-select control registers to assert for read and write cycles or for write cycles only. $\overline{EB0}$ corresponds to DATA[15:8], and $\overline{EB1}$ corresponds to DATA[7:0].

7.2.5 Boot Mode

The $\overline{MOD}$ input pin selects the initial CPU boot mode during hardware reset.

If this pin is driven to a logic-low level four LOW_REFCLK clock cycles before $\overline{RSTOUT}$ negation, then the internal ROM will be disabled and the CPU will fetch the first word from offset 0x0 of the external Flash memory, which is located at the absolute address 0x2D00000 in the CPU address space. The internal ROM is disabled for the first CPU cycle only and is available for subsequent accesses.

If this pin is driven to a logic-high level four LOW_REFCLK clock cycles before $\overline{RSTOUT}$ negation, then the internal ROM is enabled and the CPU fetches the first word from absolute address 0x0, which is the starting address of the internal ROM. This signal is latched in the timer/reset module, and an internal version is supplied to the EIM.

7.2.6 Chip Select Outputs

7.2.6.1 Chip Select 0 ($\overline{CS0}$)

This active-low output signal is asserted based on a decode of bits ADDR[31:24] of the access address, and at reset based on the value of the $\overline{MOD}$ input.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

If $\overline{MOD}$ is driven to a logic-low level four LOW_REFCLK clock cycles before $\overline{RSTOUT}$ negation, and the CSEN0 bit is enabled (the default state on reset), then the internal ROM is disabled, and $\overline{CS0}$ is asserted for the first CPU access. The internal ROM is disabled for the first CPU access only and is available for subsequent accesses. The $\overline{CS0}$ access uses default values of 15 wait states and a 16-bit port size.

7.2.6.2 Chip Select 1–2 ($\overline{CS[1:2]}$)

These active-low output signals are asserted based on a decode of bits ADDR[31:24] of the access address. When disabled, these pins can be used as programmable general-purpose outputs.

7.2.6.3 Chip Select 3 (CS3)

This active-high output signal is asserted based on a decode of the internal address bus bits ADDR[31:24] of the access address. When disabled, this pin can be used as a programmable general-purpose output.

7.3 Chip-Select Address Range

Table 7-1 specifies the address range for each chip select output.

Table 7-1 Chip Select Address Range

CSENx	ADDR[31:24]	Chip Select	Typical Use
Cleared	—	Inactive	—
Set	00101101	$\overline{CS0}$	Flash
Set	00101111	$\overline{CS1}$	SRAM
Set	00101110	$\overline{CS2}$	Spare
Set	00101100	CS3	LCD

7.4 EIM Interface Example

Figure 7-2 shows an example of an EIM interface to memory and peripherals.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

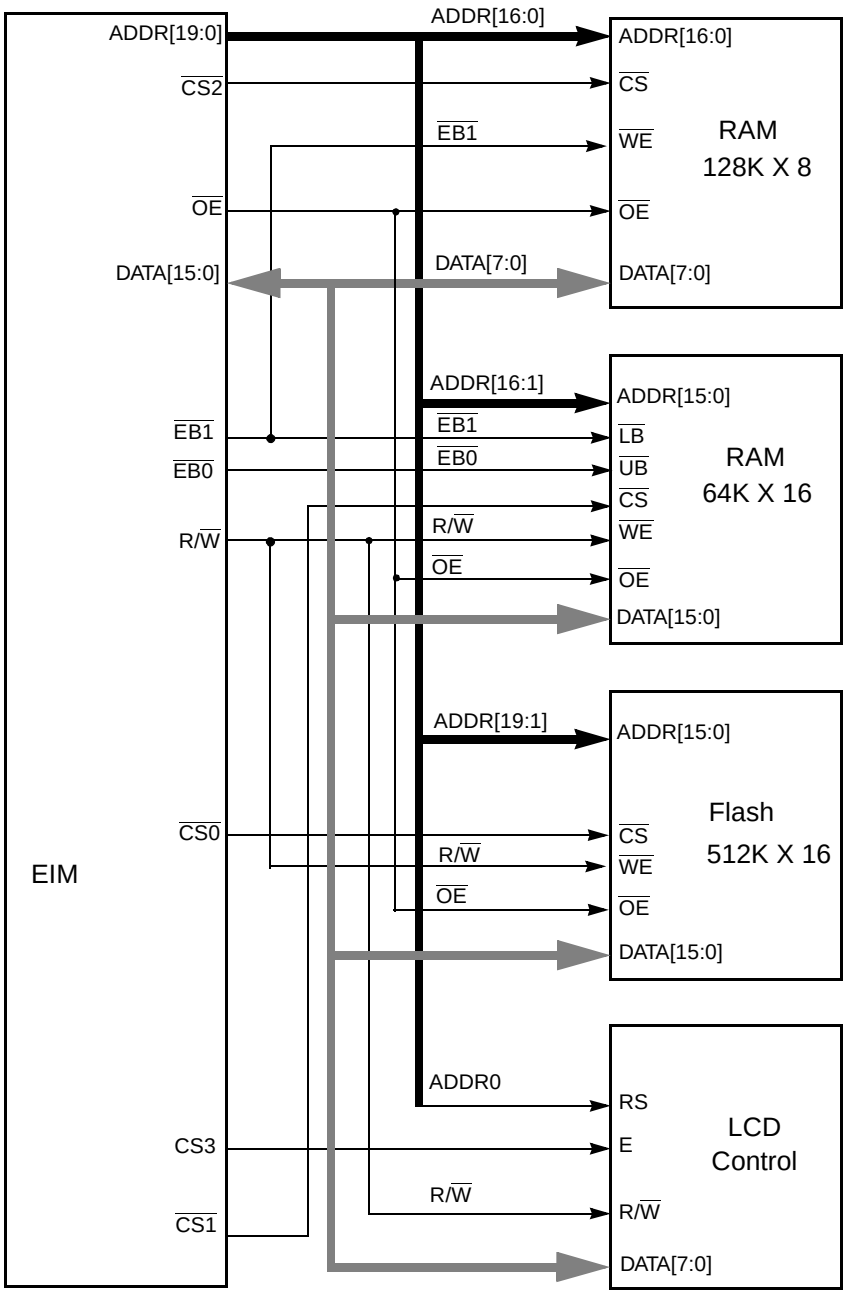


Figure 7-2 EIM Interface to Memory and Peripherals

7.5 EIM Functionality

7.5.1 Configurable Bus Sizing

The EIM supports byte, halfword, and word operands, allowing access to 8- and 16-bit ports. It does not support misaligned transfers.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

The port size is programmable via the DSZ bits in the corresponding CS control register. In addition, the portion of the data bus used for transfer to or from an 8-bit port is programmable via the same bits. An 8-bit port can reside on external data bus bits DATA[15:8] or DATA[7:0].

A word access to or from an 8-bit port requires four bus cycles to complete the transfer. A word access to or from a 16-bit port requires two bus cycles to complete the transfer. A halfword access to or from an 8-bit port requires two bus cycles to complete the transfer. In the case of a multi-cycle transfer, the lower two address bits, ADDR[1:0], are incremented appropriately.

The EIM data multiplexer takes the four bytes of the CPU interface data bus and routes them to their required positions to interface properly to memory and peripherals.

Table 7-2 lists the combination of TSIZ, ADDR[1:0] signals and DSZ bits that are used for each possible transfer size, alignment, and port width. The bytes labeled with a dash are not required; they are ignored on read transfers and driven with undefined data on write transfers.

Table 7-2 Interface Requirements for Read and Write Cycles

Transfer Size	Signal Encoding				Port Width	Active Interface Bus Sections (Internal)			
	TSIZ1	TSIZ0	ADDR1	ADDR0	DSZ[1:0]	DATA[31:24]	DATA[23:16]	DATA[15:8]	DATA[7:0]
Byte	0	1	0	0	00	DATA[15:8]	—	—	—
					01	DATA[7:0]	—	—	—
					10	DATA[15:8]	—	—	—
			0	1	00	—	DATA[15:8]	—	—
					01	—	DATA[7:0]	—	—
					10	—	DATA[7:0]	—	—
			1	0	00	—	—	DATA[15:8]	—
					01	—	—	DATA[7:0]	—
					10	—	—	DATA[15:8]	—
			1	1	00	—	—	—	DATA[15:8]
					01	—	—	—	DATA[7:0]
					10	—	—	—	DATA[7:0]
Halfword	1	0	0	x	00	DATA[15:8]	DATA[15:8]	—	—
					01	DATA[7:0]	DATA[7:0]	—	—
					10	DATA[15:8]	DATA[7:0]	—	—
			1	x	00	—	—	DATA[15:8]	DATA[15:8]
					01	—	—	DATA[7:0]	DATA[7:0]
					10	—	—	DATA[15:8]	DATA[7:0]
Word	0	0	x	x	00	DATA[15:8]	DATA[15:8]	DATA[15:8]	DATA[15:8]
					01	DATA[7:0]	DATA[7:0]	DATA[7:0]	DATA[7:0]
					10	DATA[15:8]	DATA[7:0]	DATA[15:8]	DATA[7:0]

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

7.5.2 External Boot ROM Control

The $\overline{\text{MOD}}$ input signal is used to determine the location of the boot ROM device during hardware reset. If an external boot ROM is used instead of the internal ROM, the $\overline{\text{CS0}}$ output can select the external ROM coming out of reset.

If $\overline{\text{MOD}}$ is driven to a logic-low level four LOW_REFCLK clock cycles before $\overline{\text{RSTOUT}}$ negation, and the CSEN0 bit is enabled (the default state on reset), then the internal ROM is disabled, and $\overline{\text{CS0}}$ is asserted for the first CPU cycle. The internal ROM is disabled for the first CPU access only and is available for subsequent accesses. The $\overline{\text{CS0}}$ access uses default values of 15 wait states and a 16-bit port size.

If $\overline{\text{MOD}}$ is driven to a logic-high level four LOW_REFCLK clock cycles before $\overline{\text{RSTOUT}}$ negation, then the internal ROM is enabled, and the CPU fetches the first word from internal ROM.

7.5.3 Programmable Output Generation

Unused chip select outputs can be configured to provide a programmable output signal. (This functionality is not provided for the $\overline{\text{CS0}}$ output signal. When the CSEN0 bit is cleared, $\overline{\text{CS0}}$ is always inactive.) To operate as a programmable output pin, the corresponding CSENx control bit must be cleared.

7.5.4 Bus Watchdog Operation

The EIM contains a bus watchdog timer that monitors the length of all requested accesses from the CPU. If an access does not terminate (i.e., the bus watchdog timer does not receive an internal $\overline{\text{TA}}$ or $\overline{\text{TEA}}$) within 128 clocks of being initiated, the watchdog timer expires and forces the access to be terminated by asserting a $\overline{\text{TEA}}$ signal to the CPU. The bus watchdog timer is automatically reset to a count of zero after the termination of each access. If an internal CPU peripheral does not terminate its access to the CPU or if the CPU accesses an unmapped location, the bus watchdog times out to prevent the CPU from locking up.

7.5.5 Error Conditions

The following conditions cause $\overline{\text{TEA}}$ to be asserted to the CPU:

- An access to a disabled chip select (i.e., an access to a mapped chip-select address space when the CSEN bit in the corresponding CS control register is cleared).
- A write access to a write-protected chip-select address space (i.e., the WP bit in the corresponding CS control register is set).
- A user access to a supervisor-protected chip-select address space (i.e., the SP bit in the corresponding CS control register is set).
- Bus watchdog time out when an access does not terminate within 128 clocks of being initiated. See **7.5.4 Bus Watchdog Operation** for a description of the bus watchdog operation.
- A user access to a supervisor-protected internal ROM or RAM (i.e., the corresponding SP bit in the EIM configuration register is set), or user access to peripheral space.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

7.5.6 Show Cycles

The CPU can perform data transfers to or from internal modules without using the external bus. For debugging purposes, however, it may be desirable to have the internal address and data bus appear on the external bus. These external bus cycles, called show cycles, are enabled by the SHEN bits in the EIM configuration register.

When show cycles are enabled, the EIM drives the internal address bus ADDR[19:0] onto the external address bus pins ADDR[19:0]. In addition, the internal data bus signals DATA[31:16] or DATA[15:0] are driven onto the external data bus pins DATA[15:0] according to the HDB bit in the EIM configuration register.

7.6 EIM Programming Model

Table 7-3 lists the registers in the EIM.

Table 7-3 EIM Memory Map

Address	Use	Access
10004000	$\overline{CS0}$ Control Register (CS0CR)	Supervisor Only
10004004	$\overline{CS1}$ Control Register (CS1CR)	Supervisor Only
10004008	$\overline{CS2}$ Control Register (CS2CR)	Supervisor Only
1000400C	CS3 Control Register (CS3CR)	Supervisor Only
10004010 to 10004014	Reserved	Supervisor Only
10004018	EIM Configuration Register (EIMCR)	Supervisor Only
10004020 to 10004FFF	Reserved	Supervisor Only

7.6.1 Chip-Select Control Registers

Each of the external chip selects has an enable bit as well as other control attributes. The layout of the control register is slightly different for the $\overline{CS0}$ output, which does not support the programmable output function. For CS1–CS3 control registers, bits two to 15 (i.e., bits other than the PA and CSEN bits) are undefined at reset.

Access these registers with 32-bit loads and stores only.

CS0CR — $\overline{CS0}$ Control Register **10004000**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WSC				WWS	EDC	CSA	OEA	WEN	EBC	DSZ		SP	WP	0	CSEN
W																
RESET:	1	1	1	1	1	0	0	0	0	1	1	0	0	0		1

Figure 7-3 $\overline{CS0}$ Control Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

CS1CR — $\overline{\text{CS1}}$ Control Register **10004004**
CS2CR — $\overline{\text{CS2}}$ Control Register **10004008**
CS3CR — CS3 Control Register **1000400C**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WSC				WWS	EDC	CSA	OEA	WEN	EBC	DSZ		SP	WP	PA	CSEN
RESET:	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0/1*	0

X = Undefined
* PA reset value equals zero for CS3 and one for $\overline{\text{CS}}$[1:2]

Figure 7-4 $\overline{\text{CS1}}$, $\overline{\text{CS2}}$, CS3 Control Registers

WSC — Wait-State Control

These four bits program the number of wait states for an access to the external device connected to the chip select. **Table 7-4** shows the encoding of this field. When WWS is cleared, setting WSC=0000 results in 1-clock transfers, WSC=0001 results in 2-clock transfers, and WSC=1111 results in 16-clock transfers. When WSC=0000, the WEN and CSA bits are ignored.

Set WSC=0000 and WWS=0 for access to fast SRAM devices (one-clock read and write access), CSA=0, WSC=0001 and WWS=0 for access to normal SRAM (two-clock read and write access), CSA=0, WSC=0001 and WWS=1 for access to Flash memory (two-clock read access and three-clock write access), EDC, CSA and WSC to the appropriate number for access to an LCD controller.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 7-4 Wait State Control Field Settings

WSC[3:0]	Number of Wait States			
	WWS = 0		WWS = 1	
	Read Access	Write Access	Read Access	Write Access
0000	0	0	0	1
0001	1	1	1	2
0010	2	2	2	3
0011	3	3	3	4
0100	4	4	4	5
0101	5	5	5	6
0110	6	6	6	7
0111	7	7	7	8
1000	8	8	8	9
1001	9	9	9	10
1010	10	10	10	11
1011	11	11	11	12
1100	12	12	12	13
1101	13	13	13	14
1110	14	14	14	15
1111	15	15	15	15

WWS — Write Wait State

This bit is used to determine if an additional wait state is required for write cycles. This is useful for writing to Flash memories that require additional data setup time.

0 = Reads and writes are the same length.

1 = An additional wait state is inserted for write cycles unless WSC is set to 1111. Setting WSC to 1111 results in 16-clock transfers regardless of the WWS bit. Read cycles are not affected.

EDC — Extra Dead Cycle

This bit determines whether an idle cycle is inserted after a read cycle for back-to-back external transfers to eliminate data bus contention. This is useful for slow memory and peripherals.

0 = Back-to-back external transfers occur normally, i.e., no idle cycle is inserted after a read cycle.

1 = An idle cycle is inserted after a read cycle for back-to-back external transfers, unless the next cycle is a read cycle to the same $\overline{CS}$ bank.

CSA — Chip Select Assert

This bit is used for devices that require additional address setup time and additional address/data hold times. It determines when the chip select is asserted and whether an idle cycle is inserted between back-to-back external transfers. If WSC=0000, this bit is ignored.

0 = Chip select is asserted normally, i.e., as early as possible. No idle cycle is inserted between back-to-back external transfers.

1 = Chip select is asserted a clock later during both read and write cycles. In addition, an idle cycle is inserted between back-to-back external transfers.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

OEA — $\overline{OE}$ Assert

This bit determines when $\overline{OE}$ is asserted during a read cycle. If WSC=0000, this bit is ignored and $\overline{OE}$ is asserted for one half of a clock cycle only. If EBC in the corresponding register is cleared, then the $\overline{EB}$[0:1] outputs are similarly affected.

0 = $\overline{OE}$ is asserted normally, i.e., as early as possible.
1 = $\overline{OE}$ is asserted one half of a clock cycle later during a read cycle to this chip select address space. The cycle length and write cycles are not affected.

WEN — $\overline{EB}$ Negate

This bit is used to determine when $\overline{EB}$[0:1] outputs are negated during a write cycle. This is useful to meet data hold time requirements for slow memories. If WSC=0000, this bit is ignored and $\overline{EB}$[0:1] outputs are asserted for one half of a clock cycle only.

0 = $\overline{EB}$[0:1] are negated normally, i.e., as late as possible.
1 = $\overline{EB}$[0:1] are negated one half of a clock cycle earlier during a write cycle to this chip select address space. The cycle length and read cycles are not affected.

EBC — Enable Byte Control

This bit is used to indicate which access types are allowed to assert the enable byte outputs ($\overline{EB}$[0:1]).

0 = Read and write accesses are both allowed to assert the $\overline{EB}$[0:1] outputs, thus configuring them as byte enables.
1 = Only write accesses are allowed to assert the $\overline{EB}$[0:1] outputs, thus configuring them as byte write enables. The $\overline{EB}$[0:1] outputs must be configured as byte write enables for accesses to dual x8 memories.

DSZ — Data Port Size

This field defines the width of the device data port.

Table 7-5 Data Port Size Field Settings

Value	Meaning
00	8-bit port, resides on DATA[15:8] pins
01	8-bit port, resides on DATA[7:0] pins
10	16-bit port
11	Reserved

SP — Supervisor Protect

This bit is used to restrict accesses to the address range defined by the corresponding chip select if the access is attempted in the user mode of CPU operation.

0 = User mode accesses are allowed in this chip select address range.
1 = User mode accesses are prohibited. An attempted access to an address mapped by this chip select in user mode will result in a $\overline{TEA}$ to the CPU and no assertion of the chip select output.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

WP — Write Protect

This bit is used to restrict writes to the address range defined by the corresponding chip select.

0 = Writes are allowed in this chip select address space.
1 = Writes are prohibited. An attempt to write to an address mapped by this chip select will result in a TEA to the CPU and no assertion of the chip select output.

PA — Pin Assert

This bit is used to control the chip select pin when it is operating as a programmable output pin (i.e., the CSEN bit clear). This bit is ignored if the CSEN bit is set. At reset, PA bit is set for CS[1:2] and cleared for CS3.

0 = Brings chip select output to logic-low level
1 = Brings chip select output to logic-high level

CSEN — Chip Select Enable

This bit controls the operation of the chip select pin.

0 = Chip select function is disabled. An attempted access to an address mapped by this chip select will result in TEA assertion to the CPU and no assertion of the chip select output.

When disabled, the pin is a general-purpose output controlled by the value of the PA control bit. When CSEN0 is clear, CS0 is inactive.

1 = Chip select is enabled and is asserted when an access address falls within the range specified by the memory map in **Table 7-1**. With the exception of CS0, this bit is cleared by reset, disabling the chip select output pin.

CSEN0 is set at reset to allow CS0 to select from an external boot ROM if MOD is driven to a logic-low level four LOW_REFCLK clock cycles before RSTOUT negation. When the chip select is enabled, the PA control bit is ignored.

7.7 EIM Configuration Register

The EIM configuration register contains control bits that configure the EIM and other internal blocks for certain operation modes.

Access this register with 32-bit loads and stores only.

EIMCR — EIM Configuration Register **10004018**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	SZEN	PSTEN	SP RAM	SP ROM	HDB	SHEN	
W																
RESET:										0	0	1	1	0	0	0

Figure 7-5 EIM Configuration Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

SZEN — Enable SIZ signal to UART CH1 pins

This bit is used to select the function provided by the UART channel 1 pins. On reset, this bit is cleared.

0 = UART channel 0 operation. Pins function as TxD1, RxD1.
1 = SIZ function operation. Pins function as TSIZ0 and TSIZ1 outputs independent of function or direction programmed in UART channel 1 control registers.

PSTEN — Enable PSTAT signals to UART CH0 pins

This bit is used to select the function provided by the UART channel 0 pins. On reset, this bit is cleared.

0 = UART channel 0 operation. Pins function as TxD0, RxD0, $\overline{\text{CTS0}}$, and $\overline{\text{RTS0}}$.
1 = PSTAT function operation. Pins function as PSTAT outputs independent of function or direction programmed in UART channel 0 control registers.

SPRAM — Internal RAM Supervisor Protect

This bit is used to restrict accesses to the internal RAM space if the access is attempted in the user mode of CPU operation. On reset, this bit is set.

0 = User mode accesses are allowed to the internal RAM.
1 = User mode accesses are prohibited. An attempted access to the internal RAM in user mode will result in $\overline{\text{TEA}}$ assertion to the CPU.

SPROM — Internal ROM Supervisor Protect

This bit is used to restrict accesses to the internal ROM space if the access is attempted in the user mode of CPU operation. On reset, this bit is set.

0 = User mode accesses are allowed to the internal ROM.
1 = User mode accesses are prohibited. An attempted access to the internal ROM in user mode will result in a $\overline{\text{TEA}}$ to the CPU.

HDB — High Data Bus

This bit is used to determine which byte lanes of the internal data bus are driven onto the external data bus when show cycles are enabled. This bit is ignored if SHEN is cleared.

0 = Lower internal data bus bits DATA[15:0] are driven externally.
1 = Upper internal data bus bits DATA[31:16] are driven externally.

SHEN — Show Cycle Enable

These two bits are used to determine what the EIM does with the external bus during internal transfers (i.e., an access to the internal ROM, RAM or peripherals). When show cycles are enabled, the internal address and data bus are driven externally. On reset, show cycles are disabled.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 7-6 Show Cycle Enable Field Settings

Value	Meaning
00	Show cycles disabled. The external address bus is driven with the last valid external address, and the data bus values are held by bus keepers.
01	Show cycles enabled. Internal termination to the CPU during idle cycles caused by EDC or CSA being set follows normal operation as shown in **Figure 7-11**, **Figure 7-12**, and **Figure 7-13**. This means that internal transfers that occur during EDC/CSA idle cycles will not be visible externally.
10	Show cycles enabled. Internal termination to the CPU during idle cycles caused by EDC or CSA being set is delayed by one clock. This ensures that all internal transfers can be externally monitored, at the expense of performance.
11	Reserved

7.8 External Bus Timing Diagrams

The following timing diagrams show the timing of accesses to memory or peripherals.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

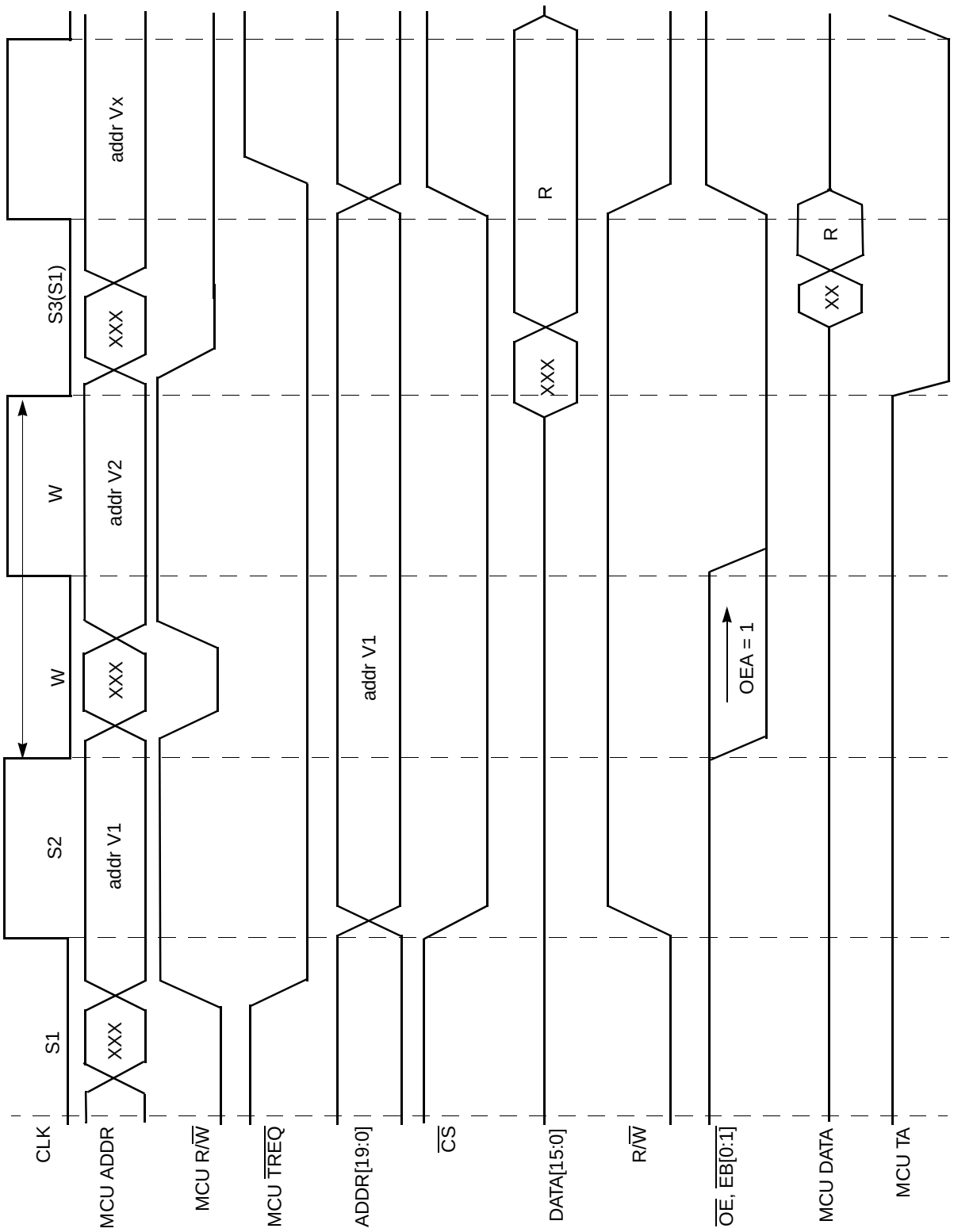


Figure 7-6 Read Memory Access (CSA = 0, WSC = 1)

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

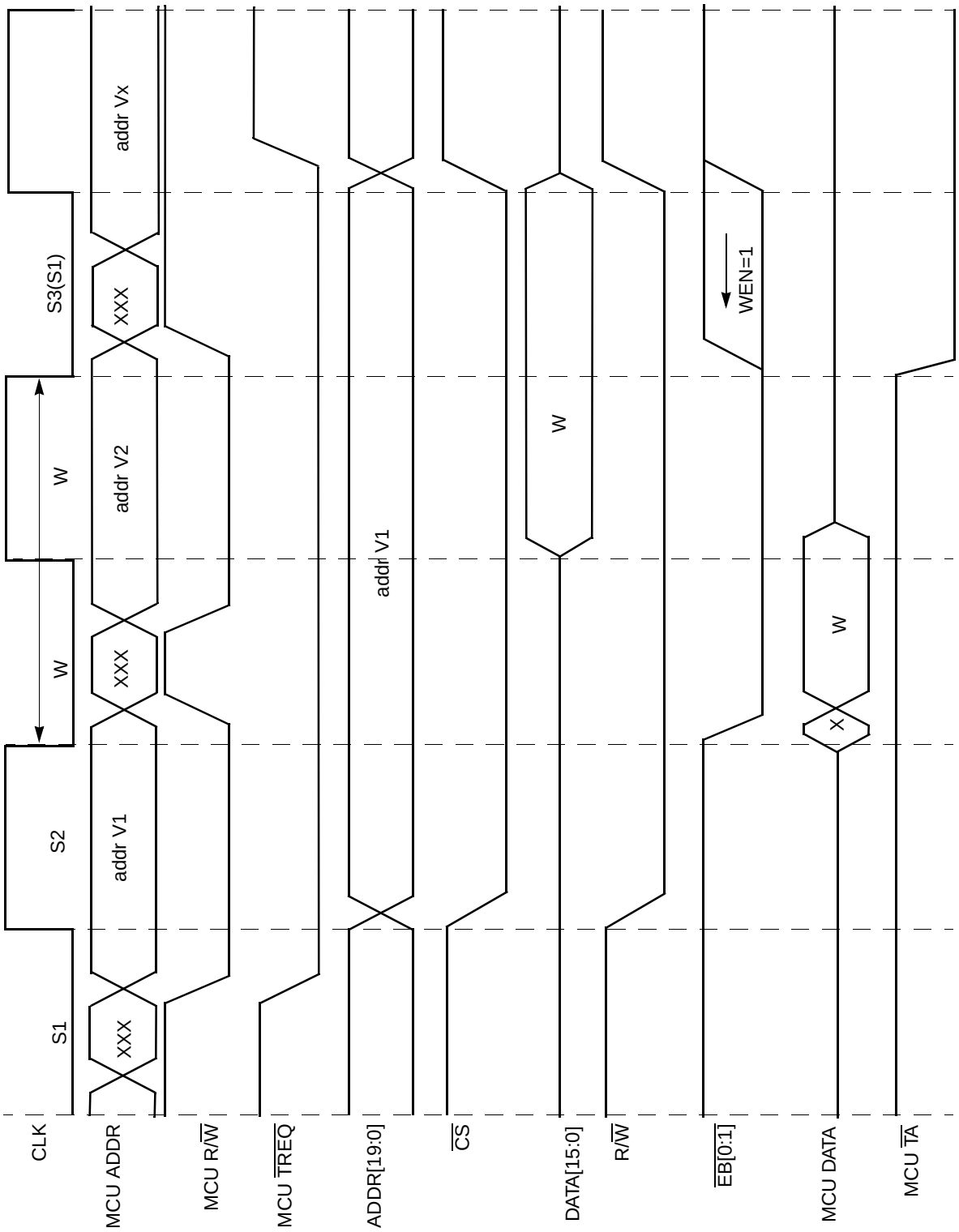


Figure 7-7 Write Memory Access (CSA = 0, WSC = 1, WWS = 0)

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

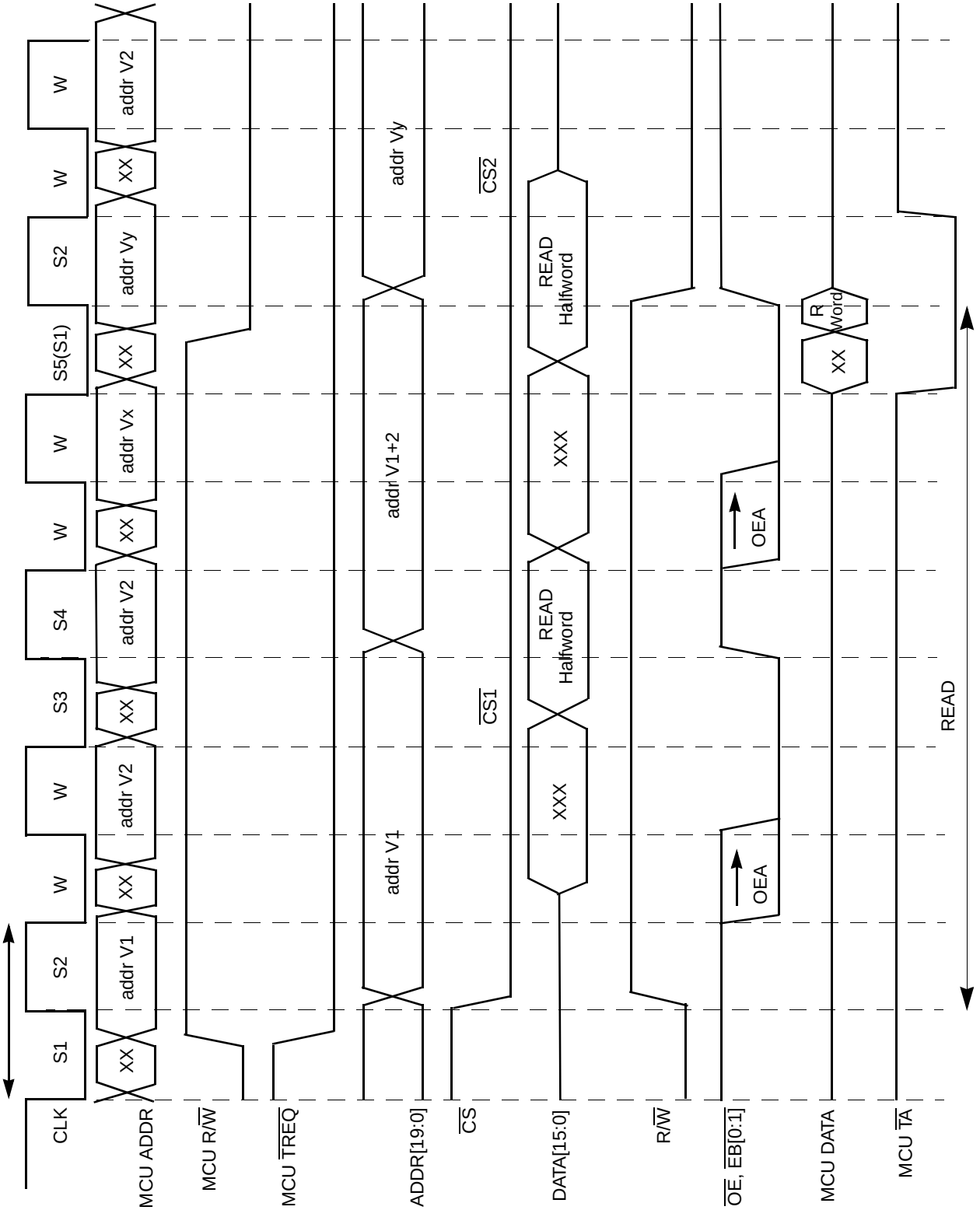


Figure 7-8 Word Read Access from Halfword Width Memory

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

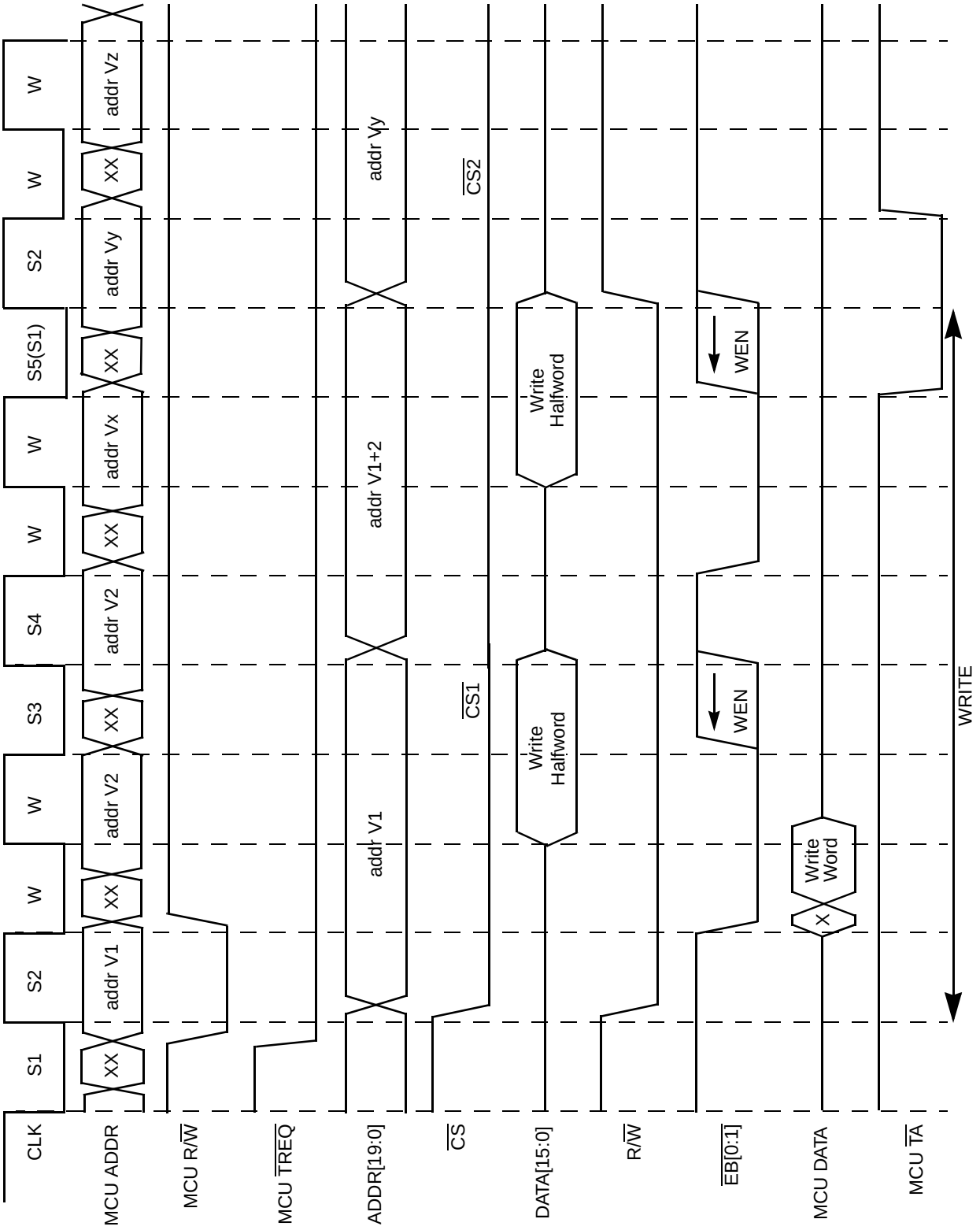


Figure 7-9 Word Write Access to Halfword Width Memory

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

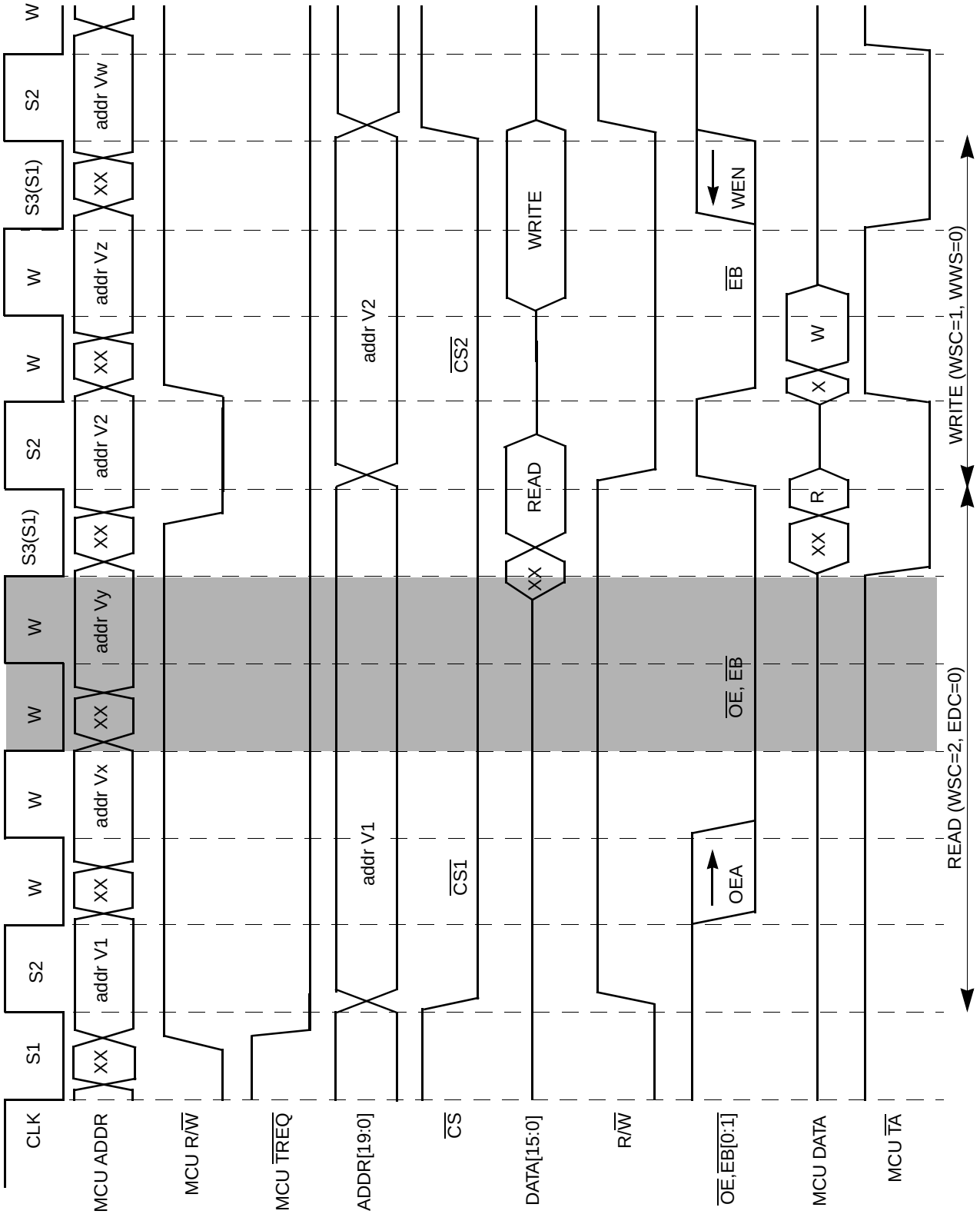


Figure 7-10 Write after Read Memory Access (CSA = 0, WSC = 2, EDC = 0)

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

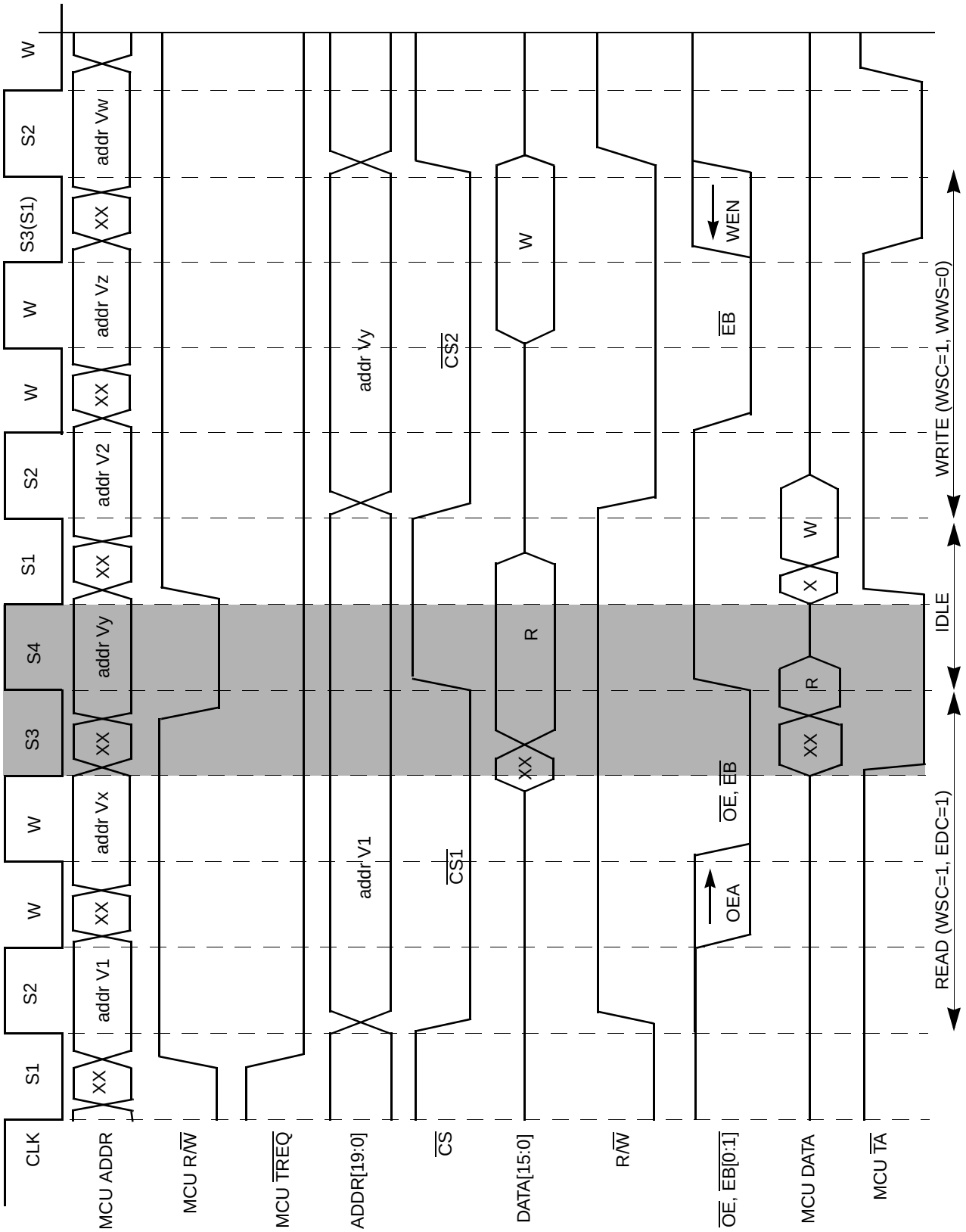


Figure 7-11 Write after Read Memory Access (CSA = 0, WSC = 1, EDC = 1)

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

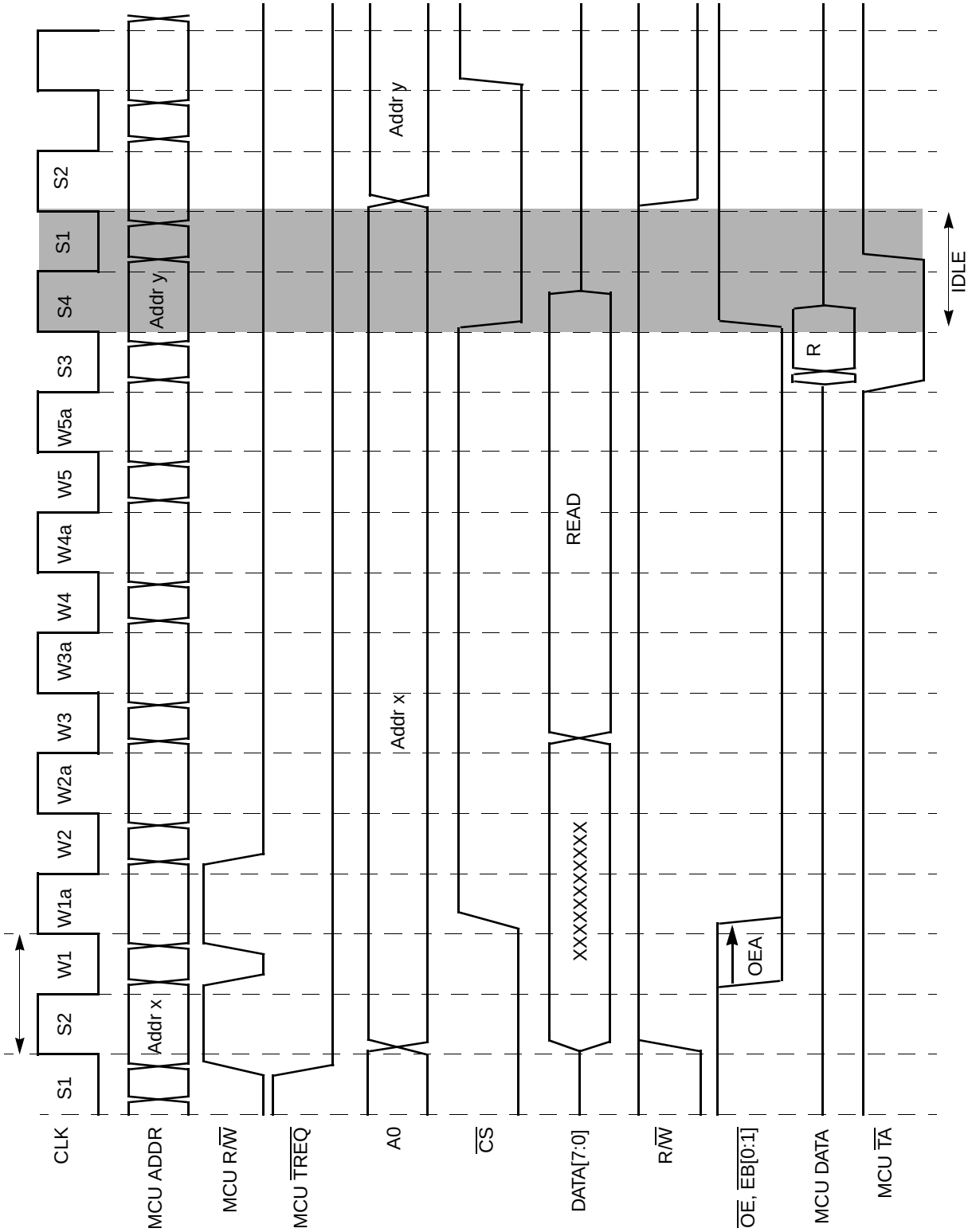


Figure 7-12 Peripheral Read Access (CSA = 1, WSC = 5)

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

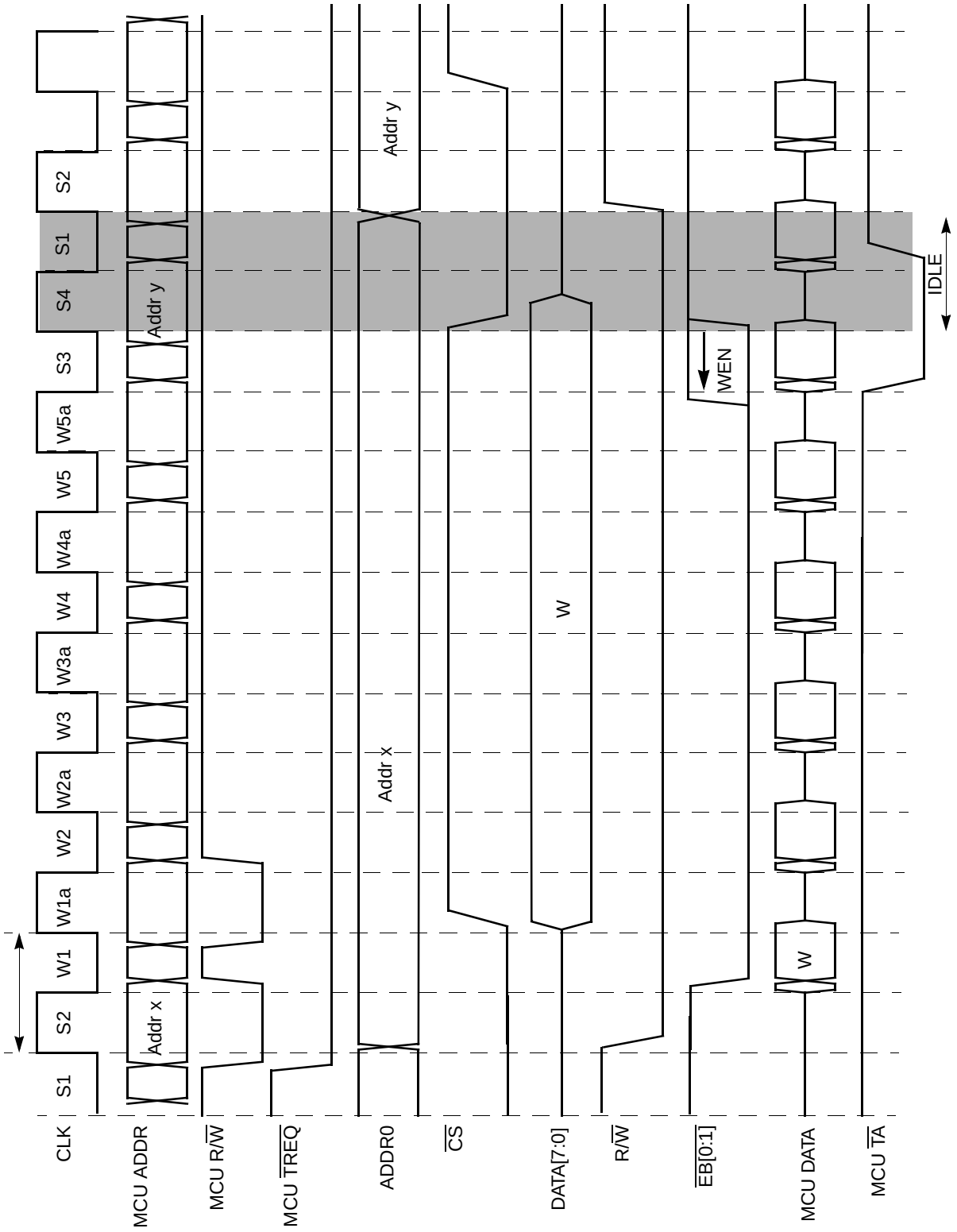


Figure 7-13 Peripheral Write Access (CSA = 1, WSC = 5)

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

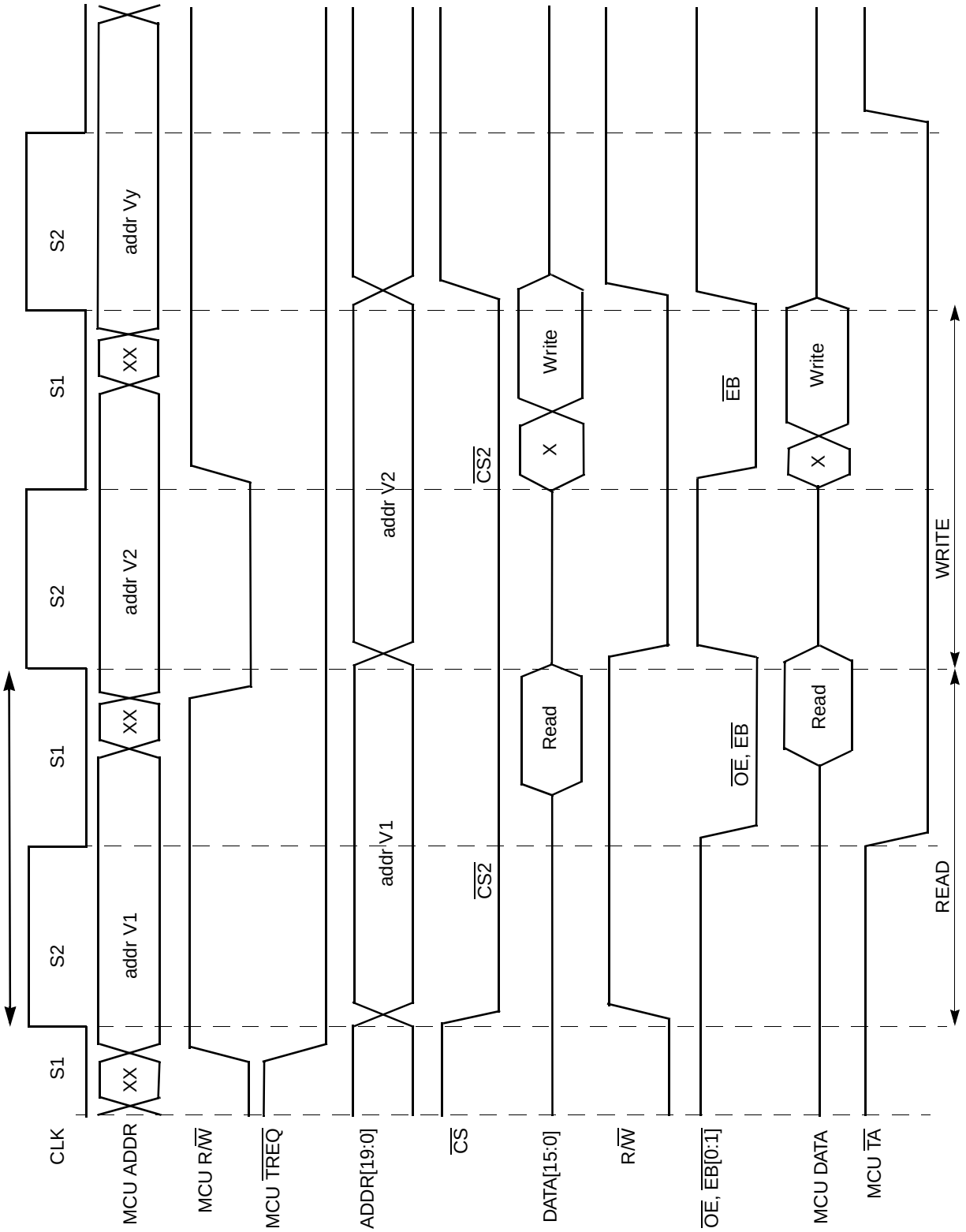


Figure 7-14 Read and Write Fast Memory Access (CSA = 0, WSC = 0, WWS = 0)

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 8
CLOCK MODULE AND LOW-POWER MODES

8.1 Overview

The two clock sources to the MMC2001 are a 16.38-MHz clock input pin (HI_REFCLK) and a 32.768-kHz crystal interface. A low-power crystal oscillator circuit (OSC) generates the constant crystal frequency clock LOW_REFCLK from the external 32.768 kHz crystal.

The CPU core is clocked by HI_REFCLK.

Peripheral modules can use the following as reference frequencies:

- The 32.768-kHz LOW_REFCLK
- The 16.38-MHz HI_REFCLK
- A prescaled version of LOW_REFCLK
- A combination of several of the above references.

The use of different clocks allows some of the peripherals to maintain a constant operational frequency even if the 16.38-MHz input signal is shut down. Changing the peripheral frequency or allowing different parts of the same peripheral to use different frequencies imply rigid constraints on clock frequencies and synchronization issues which must be addressed carefully. Refer to each peripheral definition for further details.

In addition, the CLKOUT pin can be driven by one of the internal clock sources (LOW_REFCLK or HI_REFCLK) under software control.

The following table describes the clock source for each system block.

Table 8-1 CPU Core and Peripherals Clock Source

Peripheral	Peripheral Clock Source
CPU Core	HI_REFCLK
External Interface Module	HI_REFCLK
UART	HI_REFCLK
ISPI	HI_REFCLK
Interrupt Controller	HI_REFCLK
PWM	HI_REFCLK
Watchdog Timer	LOW_REFCLK/16384 (2 Hz)
Time-of-Day Timer	LOW_REFCLK/128 (256 Hz)
Interval Timer (PIT)	LOW_REFCLK/4 (8.192 kHz)[1]
GPIO/Keypad	LOW_REFCLK/128 (256 Hz) for interrupt debouncer
JTAG/OnCE	HI_REFCLK, TCK

NOTES:
1. When enabled and not low-power disabled

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

The clock module includes the following components:

- Low-power oscillator for generation of LOW_REFCLK
- HI_REFCLK receiver and buffer
- CLKOUT circuitry
- Control circuitry for generation and gating of high frequency clocks to the CPU and peripherals
- Divider circuits for generation of low frequency clocks for peripherals.

A diagram of the clock module is shown in **Figure 8-1**.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

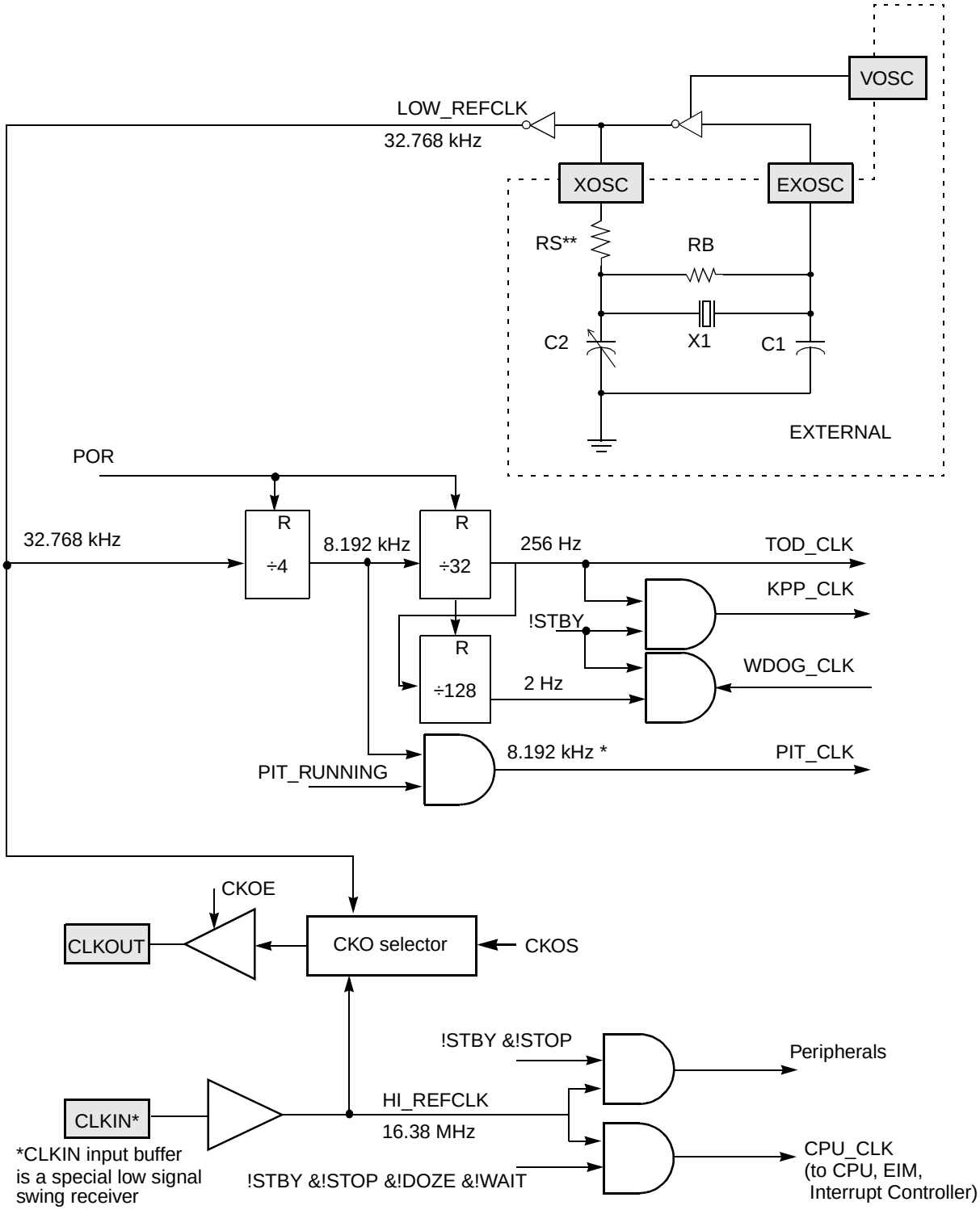


Figure 8-1 MMC2001 Clock Module

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

8.2 Low-Power Modes

The MMC2001 supports the following features:

- Four low-power modes:
  - — Run
  - — Wait
  - — Doze
  - — Stop
- Ability to shut down all peripherals independently
- Ability to shut down the CLKOUT pin.

8.2.1 CPU Core Low-Power Modes

The CPU supports four low-power modes: run, wait, doze, and stop.

8.2.1.1 Run Mode

Run mode is the normal CPU operating mode. Current consumption in this mode is related directly to the frequency chosen for the HI_REFCLK (16.38 MHz).

8.2.1.2 Wait Mode

Wait mode is intended to be used to stop only the CPU clock until an interrupt is detected. In this mode, all peripherals continue operation and can generate interrupts, which cause the CPU to exit from wait mode.

8.2.1.3 Doze Mode

Doze mode affects the CPU in the same manner as wait, but with a different code on the CPU LPMD output signals, which are monitored by peripherals. Each peripheral defines individual operational characteristics in doze mode. Peripherals which continue to run and have the capability of producing interrupts may cause the CPU to exit the doze mode and return to the run mode. Peripherals which were stopped will restart operation on exit from doze mode as defined for each peripheral.

8.2.1.4 Stop Mode

Stop mode affects the CPU in the same manner as wait, but with a different code on the CPU LPMD output signals, which are monitored by peripherals. In this mode, peripherals cease operation except for the PIT and watchdog timer, which may continue to run (if enabled) using prescaled versions of the LOW_REFCLK clock source. The TOD timer is unaffected in this mode. Several sources exist to exit this mode and return to the run mode. See **Table 8-2** for more details.

Stopped peripherals stop after entering their individual reset states. When exiting stop mode, most peripherals retain their pre-stop control register values and resume operation. Stop mode must be entered in a controlled manner, by shutting down each peripheral that is going to be stopped after it is ensured that its current operation is properly terminated.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

8.2.2 Peripheral Behavior in Low-Power Modes

8.2.2.1 UART

In doze mode, the UART stops activity, if so programmed, after finishing the current character transmission or reception. Clocks are then shut down until the CPU exits the doze mode. When the doze mode is exited, the UART activates its internal clocks, and operation continues from the point reached before entering the doze mode.

In stop mode, the UART stops immediately and freezes its operation, register values, state machines, and external pins. During the stop mode, the UART clocks are shut down. Coming out of stop mode returns the UART to operation from the mode prior to stop mode entry. To avoid erroneous operation, ensure that the UART is in an idle state before stop mode is entered.

8.2.2.2 ISPI

In doze mode, the ISPI stops activity, if so programmed, after finishing the current character transmission or reception. Clocks are then shut down until the CPU exits the doze mode. When the doze mode is exited, the ISPI activates its internal clocks, and operation continues from the point reached before entering the doze mode.

In stop mode, the ISPI stops immediately, aborts any transfer in progress, freezes its operation and register values, and forgets the state of any transfer in operation. (The state machine is reset, and the shift register is cleared.) It is assumed that when stop mode is initiated, there is some other method of shutting down external devices. During the stop mode, the ISPI clocks are shut down. Coming out of stop mode returns the ISPI to an idle mode. To avoid erroneous operation, ensure that the ISPI is in an idle state before entering the stop mode.

8.2.2.3 PWM

In doze mode, the PWM channel stops activity, if so programmed, after finishing the current period. Clocks are then shut down until the CPU exits the doze mode. When the doze mode is exited, the PWM channel is activated, and operation continues from the point reached before doze mode was entered.

In stop mode, the PWM stops immediately and freezes its operation, register values, state machines, and external pins. During the stop mode, the PWM clocks are shut down. Coming out of stop mode returns the PWM to operation from the state prior to stop mode entry. To avoid erroneous operation, ensure that the PWM is in an idle state before entering the stop mode.

8.2.2.4 Interrupt Controller

Because the interrupt controller module logic does not depend on the clock for asserting an enabled and requested interrupt to the CPU, none of the low-power modes have any influence on the module's functional behavior. The module registers will not be accessed by the CPU during the low-power mode states. Once a pending

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

interrupt causes an exit from a low power mode (and therefore activation of the CPU clocks as well as the peripheral module clocks), the CPU may access the module registers to determine the interrupt source.

8.2.2.5 Watchdog Timer

The watchdog timer uses a prescaled version of the LOW_REFCLK for its reference clock. Low-power modes affect this module only if it is considered undesirable for the watchdog to time out (causing the MMC2001 to reset) when the chip is in stop or doze mode. In stop and doze mode, if so programmed, the watchdog ceases operation and freezes at the current value. When exiting these modes, the watchdog resumes operation from the freeze value. It is the responsibility of software to avoid erroneous operation.

8.2.2.6 Interval Timer

The interval timer uses a prescaled version of the LOW_REFCLK for its reference clock. In stop and doze mode, if so programmed, the PIT ceases operation, and freezes at the current value. When exiting these modes, the PIT resumes operation from the freeze value. It is the responsibility of software to avoid erroneous operation.

8.2.2.7 Time-of-Day Timer

The time-of-day timer module uses a prescaled version of the LOW_REFCLK for its reference clock and does not stop in any of the low power modes.

8.2.2.8 Keypad Port

The keypad port module uses the CPU_CLK for its internal operation only for CPU accesses, thus the module is not affected by the low power modes and is capable of waking up the CPU from all low power modes unless it is explicitly disabled. A prescaled version of LOW_REFCLK is used for the debounce logic and continues to run in all modes.

8.2.2.9 Peripheral State During Low-Power Modes Summary

The functionality of each of the peripherals and CPU core during the various low power modes is summarized in **Table 8-2**. The status of each peripheral during a given mode refers to the condition the peripheral automatically assumes when the particular instruction (wait, doze, or stop) is executed. (It is possible to disable an individual peripheral by programming its dedicated control bits.) The wake-up capability field refers to the ability of an interrupt from that peripheral to force the CPU into run mode.

8.2.2.10 Standby Mode Summary

The functionality of each of the peripherals and CPU core during the standby mode is summarized in **Table 8-2**. Standby mode is selected when the $\overline{\text{LVRSTIN}}$ pin is asserted. All modules except for the low-power oscillator (OSC) and the time-of-day timer (TOD) are held in reset in this mode. The TOD continues to run.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Power may or may not be available to peripherals (except for the TOD and OSC) and CPU while in standby mode, depending on the level of V_{DD}. The purpose of this mode is to enable the SRAM and TOD to remain valid while the state of the other on-chip components is undefined.

Table 8-2 CPU Core and Peripherals in Low-Power Modes

Module	Peripheral Clock Status during Mode/Wake-up Capability				
	Run	Wait	Doze	Stop	Standby
CPU Core	Running	Stopped	Stopped	Stopped	Stopped
UART	Running	Running/Yes	Prog.*/Yes	Stopped/No	Stopped/No
ISPI	Running	Running/Yes	Prog.*/Yes	Stopped/No	Stopped/No
Interrupt Controller	Running	Stopped/Yes	Stopped/Yes	Stopped/Yes	Stopped/No
EIM	Running	Stopped/No	Stopped/No	Stopped/No	Stopped/No
PWMs	Running	Running/Yes	Prog.*/Yes	Stopped/No	Stopped/No
Watchdog Timer	Running	Running/Yes	Prog.*/Yes	Prog.*/Yes	Stopped/No
Interval Timer (PIT)	Running	Running/Yes	Prog.*/Yes	Prog.*/Yes	Stopped/No
Time-of-Day Timer (TOD)	Running	Running/Yes	Running/Yes	Running/Yes	Running/No
GPIO/Keypad	Running	Running/Yes	Running/Yes	Stopped/Yes	Stopped/No
OnCE	Running	Running/Yes	Running/Yes	Running/Yes	Stopped/No

*Dependent on programming

8.2.3 General Low-Power Features

8.2.3.1 Peripheral Shut Down

Each peripheral may be disabled by software in order to cease internal clock generation and remain in a static state. Each peripheral has its own specific disabling sequence (refer to each peripheral description for further details).

8.2.3.2 CLKOUT Pin Shut Down

This output pin may be disabled in the low state to lower power consumption via the CLKOUT enable (CKOE) bit in the reset/timer block.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 9
TIMER/RESET MODULE

9.1 Overview

The timer/reset module contains the following timer submodules: the time-of-day with alarm (TOD), the interval timer (PIT), and the watchdog timer (WD). In addition, this module contains a reset source/chip configuration register.

9.2 Timer/Reset Programming Model

Table 9-1 shows the timer/reset module address map. These registers are described in the subsections that describe each submodule.

Table 9-1 Timer/Reset Module Address Map

Address	Use	Access
10001000	Reset Source/Chip Configuration Register (RSCR)	Supervisor Only
10001004	Time-of-Day Control/Status Register (TODCSR)	Supervisor Only
10001008	Time-of-Day Seconds Register (TODSR)	Supervisor Only
1000100C	Time-of-Day Fraction Register (TODFR)	Supervisor Only
10001010	Time-of-Day Seconds Alarm Register (TODSAR)	Supervisor Only
10001014	Time-of-Day Fraction Alarm Register (TODFAR)	Supervisor Only
10001018	Reserved	Supervisor Only
1000101C	Watchdog Control Register (WCR)	Supervisor Only
10001020	Watchdog Service Register (WSR)	Supervisor Only
10001024	Interval Timer Control/Status Register (ITCSR)	Supervisor Only
10001028	PIT Data Register (ITDR)	Supervisor Only
1000102C	PIT Alternate Data Register (ITADR)	Supervisor Only
1000102E to 10001FFF	Reserved	Supervisor Only
10002000 to 10002FFF	Not Used (Access causes transfer error)	Not Applicable

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

9.3 Reset Operation

9.3.1 Reset Pins

Three reset related pins are provided on MMC2001: a reset input ($\overline{\text{RSTIN}}$), a low-voltage detect reset input ($\overline{\text{LVRSTIN}}$), and a reset output ($\overline{\text{RSTOUT}}$).

The $\overline{\text{RSTIN}}$ input is an active low input which is connected to external power-on and other reset control sources. This input is qualified as a valid reset if held low for at least four LOW_REFCLK clock cycles.

The $\overline{\text{LVRSTIN}}$ input is an active low input which is connected to low-voltage detection circuitry. No qualification is performed for this input; it must thus be driven in a well-controlled manner.

The $\overline{\text{RSTOUT}}$ output is an active drive pin used to reset external devices. This pin is driven low when either qualified external resets are detected or on-chip resets (watchdog timer, power-up-reset circuitry) are generated. The minimum length is eight LOW_REFCLK cycles. The $\overline{\text{RSTOUT}}$ pin remains low if either reset input pin is held low.

9.3.2 Reset Sources

Three sources are capable of generating a reset condition:

- An external qualified low condition on either reset input pin. The $\overline{\text{RSTIN}}$ signal must be valid low for four LOW_REFCLK clock cycles. $\overline{\text{LVRSTIN}}$ has no qualification requirements.
- Internal watchdog timer.
- Internal power-on reset circuitry.

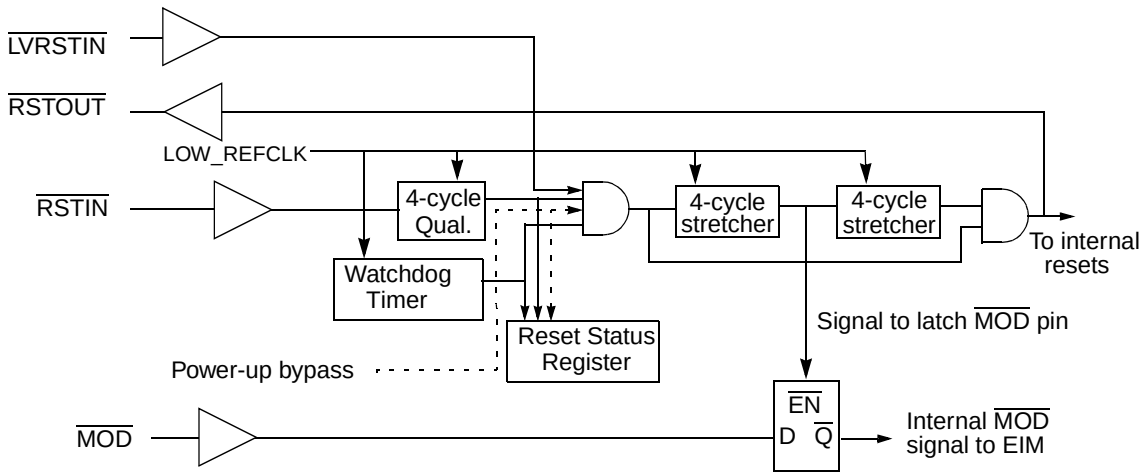


Figure 9-1 Reset Functional Block Diagram

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

9.3.3 Reset Sequence

A reset sequence depends on the operational state of the chip when the reset occurs.

- An internally provided circuit detects the power-on condition and provides an immediate reset to the “eight cycle stretcher” input, ensuring that all the on-chip circuits are properly initialized (i.e., all the peripherals and the CPU core are in their reset state). The CLKOUT pin is disabled.
- The reset circuit propagates the negation of the reset source and, following the stretcher delay of eight LOW_REFCLK cycles, negates the internal CPU core reset.
- After an initial power-up interval, the power-up-reset circuitry is not activated; thus an external reset trigger is first qualified for four LOW_REFCLK clock cycles before being propagated into the internal reset circuitry. It then functions with the same sequence as the power-up sequence.
- A watchdog reset is propagated immediately to the internal reset circuitry.

The state of the external $\overline{\text{MOD}}$ pin is latched four LOW_REFCLK cycles before the $\overline{\text{RSTOUT}}$ pin is negated, and this state is used to control the first memory access for the initial program counter value. The memory access is from either the internal ROM or external memory connected to $\overline{\text{CS0}}$.

9.3.4 Reset Source/Chip Configuration Register (RSCR)

This status and control register gives the state of the reset sources and serves to control the CLKOUT pin. Writes to this register clear any previously set status bits.

Access this register with 32-bit loads and stores only.

RSCR — Reset Source/Chip Configuration Register **10001000**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	CKOS	CKOE	0	0	0	0	LVRSTIN	RST	POR	WDR
W																
RESET:							0*	0*					*	*	*	0*

* = See bit description

Figure 9-2 Reset Source Register

CKOS — CLKOUT Source

This bit controls the clock source for the CLKOUT pin. Modify this pin only when the clock output has been disabled. This bit is cleared by POR, a qualified assertion of the $\overline{\text{RSTIN}}$ pin, or a qualified assertion of the $\overline{\text{LVRSTIN}}$ pin.

0 = CLKOUT source is HI_REFCLK
1 = CLKOUT source is LOW_REFCLK

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

CKOE — CLKOUT Enable

This bit controls the drive enable for the CLKOUT pin. It is cleared by POR, a qualified assertion of the RSTIN pin, or a qualified assertion of the LVRSTIN pin.

0 = CLKOUT is disabled and forced to the low state
1 = CLKOUT is enabled and driven from the source selected by CKOS

LVRSTIN — LVRSTIN Pin

This bit is set when the LVRSTIN pin is asserted to reset the MMC2001 chip. It is not affected by the other reset sources. When the POR bit is set, however, this bit is undefined. This bit is cleared by writing to RSCR.

RST — RSTIN Pin

This bit is set when the RSTIN pin is asserted and qualified by the four-cycle qualifier to reset the MMC2001 chip. It is not affected by the other reset sources. When the POR bit is set, however, this bit is undefined. It is cleared by writing to RSCR.

POR — Power-On Reset

This bit is set when an internal POR occurs to reset the chip. It is not affected by the other reset sources. It is cleared by writing to RSCR.

WDR — Watchdog Reset

This bit is set when the watchdog timer expires. It is cleared by POR, a qualified assertion of the RSTIN pin, a qualified assertion of the LVRSTIN pin, or by writing to RSCR.

9.4 Time-of-Day Timer

The time-of-day timer (TOD) is a free-running timer that is clocked by the crystal oscillator at a frequency of 256 Hz (LOW_REFCLK/128). It is comprised of a 32-bit register which counts seconds, and another 32-bit register which holds an 8-bit fraction of a second, where bit 31 is 1/2 of a second and bit 24 is 1/256 of a second.

The TOD has an alarm function which compares the seconds and fraction registers with the seconds alarm and fraction alarm register and issues an interrupt if there is a match.

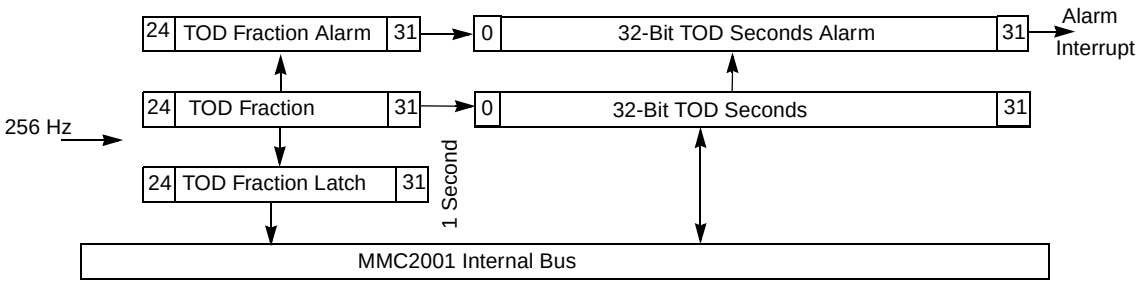


Figure 9-3 TOD Block Diagram

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

9.4.1 TOD Operation

The TOD seconds register (TODSR) is readable and writable at any time. The TOD fraction register (TODFR) is always cleared to zero when the TODSR is written. Writes to the TODFR cause the fraction register to be set to all ones, regardless of the data written. For read operations, always read the TODSR before the TODFR. This latches the TODFR, which prevents any loss of carry information from the fraction to the seconds registers.

The TOD value is matched every 1/256 second to the alarm register if the alarm function is enabled. When a match occurs, the alarm interrupt flag (AIF) is set, and when the alarm interrupt enable bit (AIE) is set, an interrupt is issued to the CPU. A write to the TOD fraction alarm register (TODFAR) clears the AIF. Writes to the TOD seconds alarm register (TODSAR) do not affect the clearing of the AIF.

The alarm match function is temporarily disabled after a write to the TODSAR until the TODFAR is written. This prevents an alarm match from occurring before the entire 40-bit alarm value is written.

The TOD fraction and seconds counters are undefined after a POR operation and are unaffected by any other reset source. The TOD alarm is disabled by any reset source.

9.4.2 TOD in Low-Power Modes

The TOD is unaffected by the low-power modes. An alarm interrupt may be used to exit from a low power state.

9.4.3 Time-of-Day Control/Status Register (TODCSR)

TODCSR — Time-of-Day Control/Status Register **10001004**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	AE	AIE	AIF
W																
RESET:														0	0	0

Figure 9-4 TOD Control/Status Register

Access this register with 32-bit loads and stores only.

AE — Alarm Enable

This bit controls the function of the TOD alarm

0 = Alarm function is off to save power
1 = Alarm function is on

AIE — Alarm Interrupt Enable

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

This bit controls the alarm interrupt function

0 = AIF is inhibited from reaching the CPU
1 = AIF is allowed to request an interrupt

AIF — Alarm Interrupt Flag

This read-only bit is the alarm interrupt flag. It is cleared by writing to the TOD alarm fraction register (TODFAR).

0 = No alarm interrupt is present
1 = Alarm interrupt is present

9.4.4 TOD Seconds Register (TODSR)

The time-of-day seconds register is a 32-bit read/write register that holds the number of elapsed seconds. It is clocked by a 1-Hz signal generated as a carry from the TOD fraction register. When TODSR is read, the content of the fraction counter is latched into a holding buffer to be read later. This prevents a fraction rollover between reads of the two registers from causing incorrect data to be read. When TODSR is written, the TODFR is cleared to all zeros. TODSR is not affected by the watchdog reset or by a reset initiated by the external reset signal, but is undefined after a POR.

Access this register with 32-bit loads and stores only.

TODSR — Time-of-Day Seconds Register **10001008**

	31	0
R W	TOD SECONDS REGISTER	
RESET:	Undefined on POR	

Figure 9-5 TOD Seconds Register

9.4.5 TOD Fraction Register (TODFR)

The 32-bit time-of-day fraction register holds eight bits of data that represent the binary fraction of a second. It is clocked by the 32.768-kHz LOW_REFCLK divided by 128 (256 Hz). Reads of this register return the value latched when the TOD seconds register was previously read. Continuous reads return the same value until the TODSR is read and new data is latched from the fraction counter. These eight bits are cleared whenever the TODSR is written but are not affected by either reset pin or the watchdog reset conditions. The fraction counter is undefined after a POR. Writes to this register cause all eight bits to be set.

Access this register with 32-bit accesses only.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

TODFR — TOD Fraction Register **1000100C**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TOD FRACTION REGISTER								0	0	0	0	0	0	0	0
W	Set to ones															
RESET:	Undefined on POR															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

Figure 9-6 TOD Fraction Register

9.4.6 TOD Seconds Alarm Register (TODSAR)

The time-of-day seconds alarm register is a 32-bit read/write register which holds data (in seconds) to be compared to the TOD seconds register. The comparison is made every 1/256 of a second if the alarm function is enabled by the AE bit in the TODCSR. Writes to this register inhibit alarm compares until the TODFAR is written. For proper alarm operation, the fraction alarm register must be (re)written after a write to this register. This register is not affected by any of the reset conditions.

Access this register with 32-bit loads and stores only.

TODSAR — Time-of-Day Seconds Alarm Register **10001010**

	31	0
R	TOD SECONDS ALARM REGISTER	
W		
RESET:	Unaffected	

Figure 9-7 TOD Seconds Alarm Register

9.4.7 TOD Fraction Alarm Register (TODFAR)

The time-of-day fraction alarm register is a 32-bit read/write register which holds eight bits of data to be compared to the TOD fraction register. The comparison is made every 1/256 of a second if the alarm function is enabled by the AE bit in the TODCSR. This register is not affected by any of the reset conditions.

Access this register with 32-bit loads and stores only.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

TODFAR — TOD Fraction Alarm Register **10001014**

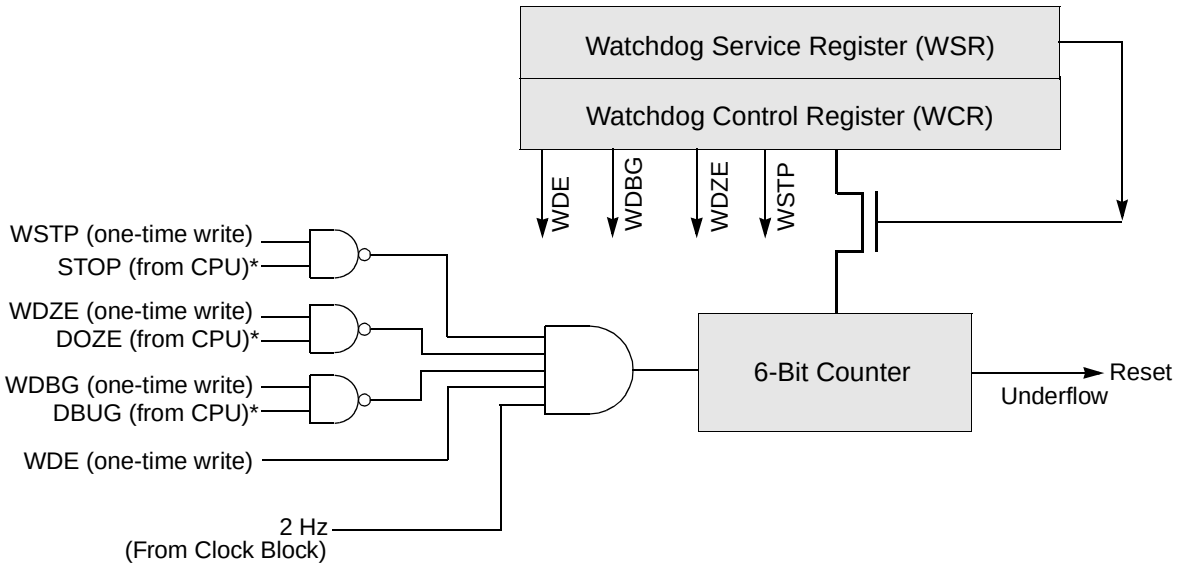
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TOD FRACTION ALARM REGISTER								0	0	0	0	0	0	0	0
W																
RESET:	Undefined on POR															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

Figure 9-8 TOD Fraction Alarm Register

9.5 Watchdog Timer

The watchdog timer is used to protect against system failures by providing a means of escape from unexpected events or programming errors. Once started, the watchdog timer must be serviced by software on a periodic basis. If servicing does not take place, the watchdog times out and asserts a reset signal.



* NOTE: DOZE and STOP are generated from the CPU signals LPMD1 and LPMD0
DBUG is an active high signal from the CPU indicating debug mode

Figure 9-9 Watchdog Timer Block Diagram

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

9.5.1 Watchdog Timing Specifications

The watchdog timer provides time-out periods from 0.5 seconds up to 32 seconds with a time resolution of 0.5 seconds. It uses a 2-Hz clock derived from a LOW_REFCLK prescaler (divide by 16384) to achieve the resolution of 0.5 seconds. The output of the prescaler circuitry is connected to the input of a 6-bit counter, resulting in a range of 0.5 to 32 seconds. The time-out period is determined by writing the watchdog time-out field (WT) in the watchdog control register. **Figure 9-9** shows a block diagram of the watchdog timer.

NOTE

The internal count registers are clocked using the 2-Hz reference. In order to read and write these registers accurately via the CPU core (which runs from the CPU_CLK), clock synchronization logic is required internal to the watchdog. This logic requires that the CPU clock frequency be greater than or equal to the LOW_REFCLK clock from which the counter clocks are derived.

9.5.2 Watchdog Timer after Reset

The watchdog is disabled by default after reset. Once enabled by software, it cannot be disabled. The watchdog enable bit (WDE) is located in the watchdog control register. At reset, the watchdog control register and watchdog service register are initialized to zero.

9.5.3 Watchdog Timer Service Operation

A service sequence must be executed periodically to keep the watchdog from timing out and causing a reset. The service routine is based on writing to the watchdog service register. See **9.5.8.2 Watchdog Service Register (WSR)**.

9.5.4 Watchdog Timer in Wait Mode

In wait mode, all peripheral clocks run normally. The watchdog is not affected.

9.5.5 Watchdog Timer in Doze Mode

In doze mode, the watchdog may either continue to run or be halted. If the WDZE (watchdog doze enable) bit is set in the watchdog control register (WCR), the watchdog is halted. When doze mode is exited, the watchdog operation reverts to what it was prior to entering doze mode.

9.5.6 Watchdog Timer in Stop Mode

In stop mode, the watchdog may either continue to run or be halted. If the WSTP (watchdog stop enable) bit is set in the watchdog control register (WCR), the watchdog is halted. When stop mode is exited, the watchdog operation reverts to what it was prior to entering stop mode.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

9.5.7 Watchdog Timer in Debug Mode

In debug mode, the watchdog may either continue to run or be halted. If the WDBG (watchdog debug enable) bit is set in the watchdog control register (WCR), the watchdog is halted. At this point, the timer is stopped, but register read and write accesses function normally. In this mode, the WCR one-time-write lock is disabled and the control bits can be updated.

NOTE

If the WCR is updated in debug mode, it will remain updated when debug mode is exited.

9.5.8 Watchdog Timer Programming Model

The watchdog programming model consists of a control register and a service register.

9.5.8.1 Watchdog Control Register (WCR)

This register contains fields that control the operation of the watchdog in different modes of operation. The write-once bits can only be written once after a reset condition. Subsequent attempts to write to them will not affect the data previously written.

Access this register with 32-bit loads and stores only.

WCR — Watchdog Control Register **1000101C**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WT						0	0	0	0	0	0	WSTP	WDE	WDBG	WDZE
W																
RESET:	0	0	0	0	0	0							0	0	0	0

Figure 9-10 Watchdog Control Register

WT — Watchdog Time-Out

The six-bit WT field contains the time-out value. These bits are reloaded into the watchdog timer when it has been serviced. After reset, write WT before enabling the watchdog. The value in WT is loaded into the watchdog counter after running the service routine as well as on enabling the watchdog timer.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

WSTP — Watchdog Stop Enable (one-time writable)

0 = Watchdog not affected in stop mode
1 = Watchdog disabled in stop mode

WDE — Watchdog Enable (one-time writable)

0 = Watchdog is disabled
1 = Watchdog is enabled (once set, watchdog cannot be disabled)

WDBG — Watchdog Debug Enable (one-time writable)

0 = Watchdog not affected in debug mode
1 = Watchdog disabled in debug mode

WDZE — Watchdog Doze Enable (one-time writable)

0 = Watchdog not affected in doze mode
1 = Watchdog disabled in doze mode

9.5.8.2 Watchdog Service Register (WSR)

When enabled, the watchdog requires that a service sequence be written to the watchdog service register (WSR). This register controls the clearing of the watchdog counter to keep it from timing out and causing a reset. If this register is not written with 0x5555 followed by 0xAAAA before the selected rate expires, the watchdog sets the WDR bit in the reset source register and asserts a system reset.

Both writes must occur in the order listed prior to the time-out, but any number of instructions can be executed between the two writes.

Access this register with 32-bit loads and stores only.

WSR — Watchdog Service Register **10001020**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	WATCHDOG SERVICE REGISTER															
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-11 Watchdog Service Register

9.6 Interval Timer (PIT)

The interval timer (PIT) is a 16-bit “set-and-forget” timer that provides precise interrupts at regular intervals with minimal processor intervention. The timer can either count down from the value written in the modulus latch, or it can be a free-running down-counter.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

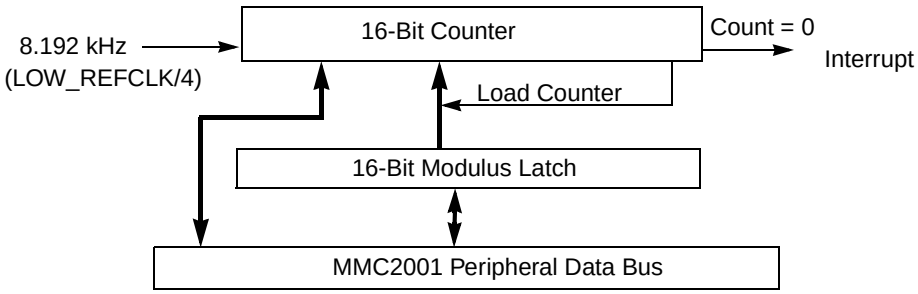


Figure 9-12 PIT Block Diagram

9.6.1 PIT Operation

The PIT can be written at any time. The value written to the PIT data register (ITDR) determines the modulus of the timer. Data read from the ITDR is the present value of the modulus latch. The present counter value is read by reading the PIT alternate data register (ITADR).

The counter is clocked at a fixed rate of 1/8192 seconds (~122 μs) which is derived from the 32.768 kHz LOW_REFCLK divided by four.

NOTE

The internal count registers are clocked using the above divide-by-four reference. In order to read and write these registers accurately via the CPU core (which runs from the CPU_CLK), clock synchronization logic is required internal to the PIT. This logic requires that the CPU clock frequency (driven from HI_REFCLK) be greater than or equal to the LOW_REFCLK clock that the counter clocks are derived from.

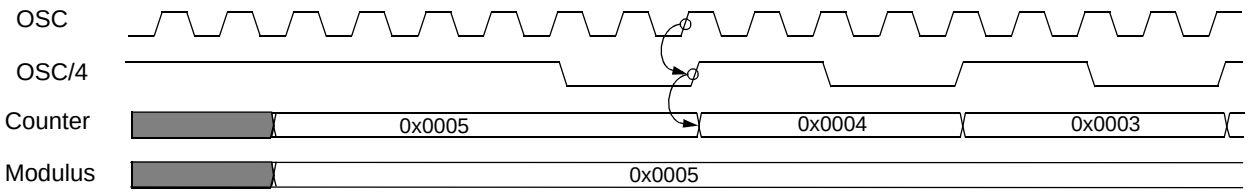


Figure 9-13 Starting a Count from an Off State

9.6.2 PIT as a “Set-and-Forget” Timer

This mode of operation is selected when the RLD bit in the PIT control/status register (ITCSR) is set to a value of one. The counter is not directly writable from the module data bus; instead, it gets its data from the modulus latch.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

When the counter reaches a count of zero, the PIT interrupt flag (ITIF) is set in the ITCSR and the value in the modulus latch is loaded into the counter to be decremented towards zero. If the PIT interrupt enable (ITIE) bit is set in the ITCSR, the interrupt flag issues an interrupt to the CPU.

The counter may by directly initialized, without having to wait for the count to reach zero, when the ITDR is written with the OVW bit in the ITCSR set.

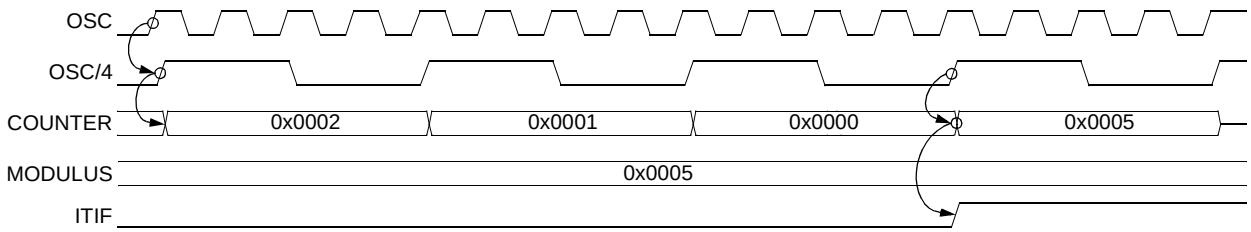


Figure 9-14 Counter Reloading from the Modulus Latch

9.6.3 PIT as a “Free-Running” Timer

This mode of operation is selected when the RLD bit in the ITCSR is cleared to a value of zero. In this mode, the counter rolls over from 0x0000 to 0xFFFF, without reloading from the modulus latch, and continues to count.

When the counter reaches a count of zero, the PIT interrupt flag (ITIF) is set in the ITCSR. If the PIT interrupt enable (ITIE) bit is set in the ITCSR, the interrupt flag can issue an interrupt to the CPU.

The counter may by directly initialized, without having to wait for the count to reach zero, when the ITDR is written while the OVW bit is set.

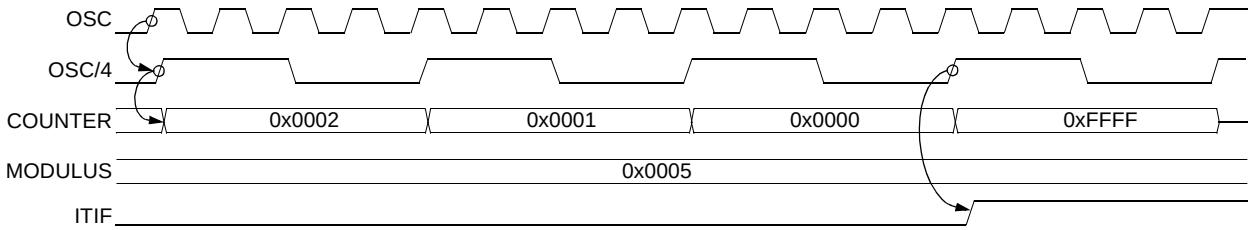


Figure 9-15 Counter in Free-Running Mode

9.6.4 Interval Timer Registers

The interval timer has three registers: the control/status register (ITCSR), the data register (ITDR), and the alternate data register (ITADR).

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

9.6.5 PIT Control/Status Register (ITCSR)

ITCSR — Interval Timer Control and Status Register **10001024**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	STOP	DOZE	DBG	OVW	ITIE	ITIF	RLD	EN
W																
RESET:									0	0	0	0	0	0	0	0

Figure 9-16 PIT Control and Status Register

Access this register with 32-bit loads and stores only.

STOP — Stop Mode Control

This bit controls the function of the PIT in stop mode

0 = PIT function is not affected in stop mode
1 = PIT function is frozen in stop mode

DOZE — Doze Mode Control

This bit controls the function of the PIT in doze mode

0 = PIT function is not affected in doze mode
1 = PIT function is frozen in doze mode

DBG — Debug Mode Control

This bit controls the function of the PIT in debug mode

0 = PIT function is not affected in debug mode
1 = PIT function is frozen in debug mode

OVW — Counter Overwrite Enable

This bit controls what happens to the counter value when the modulus latch is written.

0 = Modulus latch is a holding register for values to be loaded into the counter when the count expires to zero.
1 = Modulus latch is transparent. All writes to the latch will also overwrite the counter contents.

ITIE — PIT Interrupt Enable

This bit controls the PIT interrupt function.

0 = ITIF is inhibited from reaching the CPU.
1 = ITIF is allowed to request an interrupt.

ITIF — PIT Interrupt Flag

This bit is the PIT interrupt flag. It is cleared by writing a one to this bit or by writing to the PIT data register.

0 = No PIT interrupt is present
1 = PIT interrupt is present

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

RLD — Counter Reload Control

This bit controls whether the value contained in the modulus latch is reloaded into the counter when the counter reaches a count of zero or whether the counter rolls over from zero to 0xFFFF

0 = Counter rolls over to 0xFFFF
1 = Counter is reloaded from the modulus latch

EN — PIT Enable

This bit controls the PIT enable function

0 = PIT is disabled
1 = PIT is enabled

9.6.6 PIT Data Register (ITDR)

On a write, the data becomes the new timer modulus. This value is retained and is used at the next and all subsequent reloads until changed by another write to ITDR. This value is initialized to the maximum count of 0xFFFF on reset. On a read, the ITDR returns the value written in the modulus latch. The only way to change the value of the count directly is to preload a modulus with the OVW bit set to one. The counter value can be read from the PIT alternate data register.

Access this register with 32-bit loads and stores only.

ITDR — PIT Data Register **10001028**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIT DATA															
W																
RESET:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 9-17 PIT Data Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

9.6.7 PIT Alternate Data Register (ITADR)

The PIT alternate data register is a read-only register that provides access to the counter value. Access this register with 32-bit loads and stores only.

ITADR — PIT Alternate Data Register **1000102C**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIT COUNTER															
W																
RESET:																

Figure 9-18 PIT Alternate Data Register

9.6.8 PIT in Low-Power Modes

The PIT is unaffected by wait mode. In stop or doze mode, the timer may either continue to run or be halted. If the DOZE or STOP bit is set in the control/status register, the timer is halted in the respective mode. When doze or stop mode is exited, timer operation reverts to what it was prior to entering doze or stop mode.

9.6.9 PIT in Debug Mode

In debug mode, the module may either continue to run or be halted. If the DBG bit is set in the PIT control/status register, the timer is halted. When debug mode is exited, the timer operation reverts to what it was prior to entering debug mode.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 10
INTERRUPT CONTROLLER

10.1 Overview

The interrupt controller module collects requests from multiple interrupt sources and provides an interface to the M•CORE interrupt control lines.

The MMC2001 interrupt controller supports up to 32 interrupt sources (although not all are currently defined). The processor supports two categories of interrupts: normal and fast.

The interrupt controller performs the following functions:

- Indicates pending interrupt sources via a register
- Independently enables/disables any interrupt source
- Selects normal or fast interrupt request for any interrupt source
- Provides a mechanism for software to schedule an interrupt.

The interrupt controller consists of a set of control registers and associated logic to perform interrupt masking and priority support.

The interrupt source register (INTSRC) is a 32-bit control register with a single interrupt source associated with each bit. One or more interrupt lines are routed from each interrupt source to the INTSRC register. This allows up to 32 distinct interrupt sources in an implementation.

A corresponding 32-bit normal interrupt enable register (NIER) allows individual bit masking of the INTSRC register, and a normal interrupt pending register (NIPND) indicates pending normal interrupt requests. A logical AND is performed on the INTSRC register and the content of the NIER register to form the content of the normal interrupt pending (NIPND) register. A logical bit-wise OR is performed on all the NIPND register bits to form the $\overline{\text{INT}}$ signal routed to the CPU core. This core input signal is maskable by a bit in the PSR.

Two registers support fast interrupt requests. The fast interrupt enable register (FIER) allows individual bit masking of the INTSRC register, and the fast interrupt pending register (FIPND) indicates pending fast interrupt requests. A logical AND is performed on the INTSRC register and the content of the FIER register to form the content of the fast interrupt pending (FIPND) register. A logical bit-wise OR is performed on the FIPND register bits to form the $\overline{\text{FINT}}$ signal routed to the CPU core. This core input signal is maskable by a bit in the PSR.

These registers are readable by software. In addition, the NIER and FIER registers are writable. Attempted writes to read-only registers are ignored. Access these registers with 32-bit loads and stores only.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

In the MMC2001, interrupt requests to the CPU are always treated as autovectored. The interrupt controller accomplishes this by asserting the CPU $\overline{\text{AVEC}}$ input along with the appropriate interrupt request. An interrupt handler can read the NIPND or FIPND register and then vector based on the value received.

The INTSRC register inputs at bit positions [0:2] are reserved for software generation of interrupts and are always forced to a one. By enabling interrupts for these bit positions, software can force an interrupt request.

The interrupt requests are prioritized in the following sequence:

1. Fast interrupt requests

2. Normal interrupt requests

The two interrupt lines $\overline{\text{INT}}$ and $\overline{\text{FINT}}$ are mutually exclusive. If $\overline{\text{FINT}}$ is asserted while $\overline{\text{INT}}$ is already asserted, $\overline{\text{INT}}$ is automatically negated.

10.2 Interrupt Controller Programming Model

Control and status registers for the interrupt controller begin at address 0x40002000.

Table 10-1 Interrupt Controller Address Map

Address	Use	Access
10000000	Interrupt Source Register (INTSRC)	Supervisor Only
10000004	Normal Interrupt Enable Register (NIER)	Supervisor Only
10000008	Fast Interrupt Enable Register (FIER)	Supervisor Only
1000000C	Normal Interrupt Pending Register (NIPND)	Supervisor Only
10000010	Fast Interrupt Pending Register (FIPND)	Supervisor Only

10.2.1 Interrupt Source Register (INTSRC)

Access the 32-bit interrupt source register with 32-bit loads only.

INTSRC — Interrupt Source Register **10000000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IN31	IN30	IN29	IN28	IN27	IN26	IN25	IN24	IN23	IN22	IN21	IN20	IN19	IN18	IN17	IN16
RESET:															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN15	IN14	IN13	IN12	IN11	IN10	IN9	IN8	IN7	IN6	IN5	IN4	IN3	1	1	1
RESET:															

Figure 10-1 Interrupt Source Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

INx — Interrupt Source x

This bit indicates the state of the corresponding interrupt source.

0 = Negated
1 = Asserted

Bits [0:2] of this register are tied to logic level one to allow software to schedule interrupts by enabling one or more of these “sources” in the appropriate interrupt enable register(s) (NIER, FIER).

10.2.2 Normal Interrupt Enable Register (NIER)

Access the 32-bit normal interrupt enable register with 32-bit loads and stores only.

NIER — Normal Interrupt Enable Register **10000004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10-2 Normal Interrupt Enable Register

ENx — Enable Normal Interrupt Flag x

This bit enables the corresponding interrupt source to request a normal interrupt.

0 = Disable
1 = Enable

A reset operation clears this bit.

When the enable flag is set and the corresponding interrupt line is asserted, the interrupt controller asserts a normal interrupt request. Enabling an interrupt source which has an asserted request causes that interrupt to become pending, and a request to the CPU is asserted if not already outstanding.

10.2.3 Fast Interrupt Enable Register (FIER)

Access the 32-bit fast interrupt enable register with 32-bit loads and stores only.

FIER — Fast Interrupt Enable Register **10000008**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EF31	EF30	EF29	EF28	EF27	EF26	EF25	EF24	EF23	EF22	EF21	EF20	EF19	EF18	EF17	EF16
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EF15	EF14	EF13	EF12	EF11	EF10	EF9	EF8	EF7	EF6	EF5	EF4	EF3	EF2	EF1	EF0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10-3 Fast Interrupt Enable Register

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

EFx — Enable Fast Interrupt Flag x

This bit enables the corresponding interrupt source to request a fast interrupt.

0 = Disable
1 = Enable

A reset operation clears this bit.

When the enable flag is set and the corresponding interrupt line is asserted, the interrupt controller asserts a fast interrupt request. Enabling an interrupt source that has an asserted request causes that interrupt to become pending, and a request to the CPU is asserted if not already outstanding.

10.2.4 Normal Interrupt Pending Register (NIPND)

Access the 32-bit normal interrupt pending register with 32-bit loads only.

NIPND — Normal Interrupt Pending Register **1000000C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NP31	NP30	NP29	NP28	NP27	NP26	NP25	NP24	NP23	NP22	NP21	NP20	NP19	NP18	NP17	NP16
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NP15	NP14	NP13	NP12	NP11	NP10	NP9	NP8	NP7	NP6	NP5	NP4	NP3	NP2	NP1	NP0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10-4 Normal Interrupt Pending Register

NPx — Normal Interrupt Pending Flag x

This bit indicates a pending normal interrupt request from the corresponding interrupt source.

0 = No request
1 = Interrupt request pending

When a normal interrupt enable flag is set and the corresponding interrupt line is asserted, the interrupt controller asserts a normal interrupt request. The normal interrupt pending flags reflect the interrupt input lines which are asserted and are currently enabled to generate a normal interrupt.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

10.2.5 Fast Interrupt Pending Register (FIPND)

Access the 32-bit read-only fast interrupt pending register with 32-bit loads only.

FIPND — Fast Interrupt Pending Register **10000010**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FP31	FP30	FP29	FP28	FP27	FP26	FP25	FP24	FP23	FP22	FP21	FP20	FP19	FP18	FP17	FP16
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FP15	FP14	FP13	FP12	FP11	FP10	FP9	FP8	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10-5 Fast Interrupt Pending Register

FPx — Fast Interrupt Pending Flag x

This bit indicates a pending fast interrupt request from the corresponding interrupt source.

0 = No request
1 = Interrupt request pending

When a fast interrupt enable flag is set and the corresponding interrupt line is asserted, the interrupt controller will assert a fast interrupt request ($\overline{\text{FINT}}$ CPU input). The fast interrupt pending flags reflect the interrupt input lines which are currently enabled to generate a fast interrupt and are asserted.

10.2.6 Interrupt Request Input Assignments

The assignment of bits within the interrupt registers to interrupt sources is shown in **Table 10-2**.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 10-2 Interrupt Source Assignment

Bit Number	Use
0, 1, 2	Software
3	Unused
4	Unused
5	UART0 RTS_DELTA
6	KPP control
7	Time-of-day alarm
8	PIT
9	Unused
10	PWM0
11	PWM1
12	PWM2
13	PWM3
14	PWM4
15	PWM5
16	UART0 transmit
17	UART1 transmit
18	UART0 receive
19	UART1 receive
20	ISPI
21	INT0
22	INT1
23	INT2
24	INT3
25	INT4
26	INT5
27	INT6
28	INT7
29	Unused
30	Unused
31	Unused

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 11
UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER MODULE

11.1 Overview

The universal asynchronous receiver/transmitter (UART) module provides asynchronous serial communication with external devices such as modems and other computers. The UART module consists of two independent and almost identical channels. The description of a single channel in this section applies to both UART0 and UART1, with the exception of the $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ pin functions. These are available only for UART0.

Some of the key features of the UART channels are:

- Full-duplex operation
- Direct support of Infrared Data Association mechanism
- Robust receiver data sampling with noise filtering
- 16-entry FIFOs for transmit and receive
- 7- or 8-bit operation with optional even or odd parity and one or two stop bits
- Generation and detection of break
- 16x bit clock generator for bit rates of 300 bps to 115.2 Kbps
- Three maskable interrupts
- RTS interrupt with wake up from stop capability
- Low-power modes

The UART performs all normal operations associated with “start-stop” asynchronous communication. Serial data is transmitted and received at standard bit rates using the internal 16x bit clock generator.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

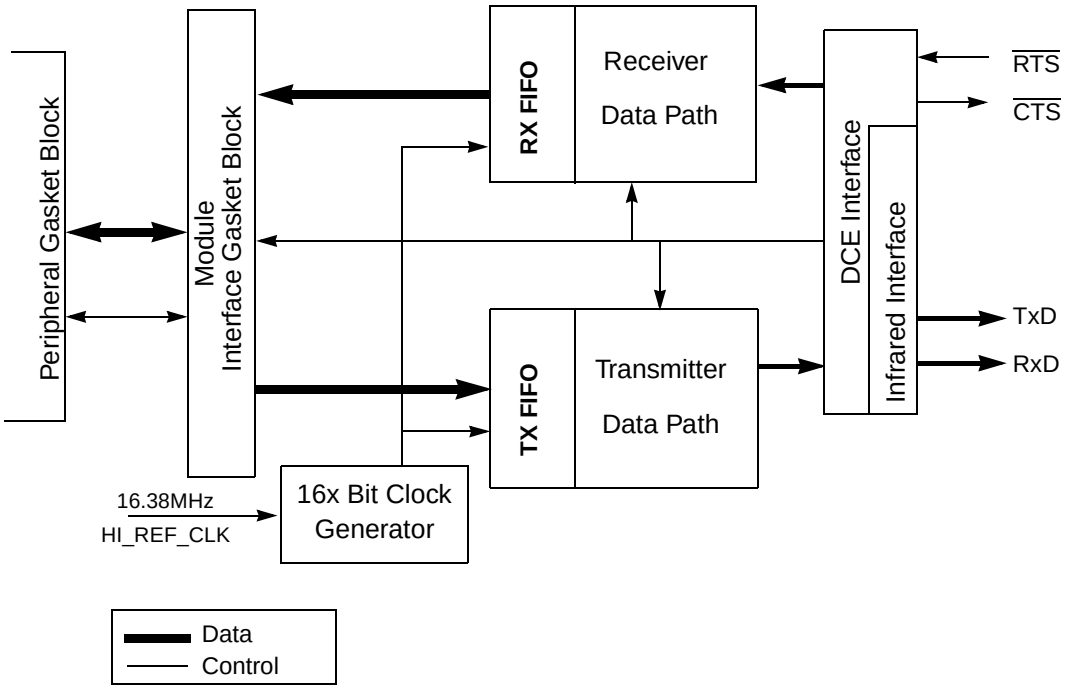


Figure 11-1 UART Channel Block Diagram

11.2 UART Signals

11.2.1 $\overline{\text{RTS}}$ — Request to Send (UART0)

An external device requests data from the UART by asserting the $\overline{\text{RTS}}$ input. When the ignore $\overline{\text{RTS}}$ (IRTS) bit is set, the UART ignores the $\overline{\text{RTS}}$ input and sends a character whenever a character is ready to transmit.

If $\overline{\text{RTS}}$ goes high during transmission, the UART finishes sending the current character and then shuts off. FIFO content (characters to be transmitted) is undisturbed.

Any transition on the pin can cause an interrupt request to wake up the MCU from low power state. Interrupts are enabled by the RTSD EN bit in UART control register 1.

This pin can also be used as a general-purpose input whose status is read in the RTSS bit in the UART status register.

11.2.2 $\overline{\text{CTS}}$ — Clear to Send (UART0)

The UART signals an external device that it is ready to receive data by asserting the $\overline{\text{CTS}}$ output. If the receiver detects a pending overrun, it negates the $\overline{\text{CTS}}$ output.

This pin can also be used as a general purpose output controlled by the CTS bit in UART control register 2.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

NOTE

The $\overline{RTS}$ and $\overline{CTS}$ signal names follow EIA232 DTE conventions. Thus, $\overline{RTS}$ is an input and $\overline{CTS}$ is an output. In many instances, these names are reversed to reflect device drive direction. Check the specification of the remote device to assure correct connection.

11.2.3 TXD — UART Transmit

This pin is the transmitter serial output. In normal mode, NRZ data is output. In infrared mode, a 3/16 bit-period pulse is output for each zero bit transmitted and no pulse for each one bit transmitted. For EIA232 standard applications this pin must be connected to an EIA232 transmitter to convert voltage levels. TXD can be programmed as a general-purpose I/O pin when the UART TXD function is not being used.

11.2.4 RXD — UART Receive

This pin is the receiver serial input. In normal mode, NRZ data is expected. In infrared mode, a narrow pulse is expected for each zero bit received and no pulse for a one bit received. External circuitry must be used to convert the infrared signal to an electrical signal. EIA232 standard applications require an external EIA232 receiver to convert voltage levels. RXD can be programmed as a general-purpose I/O pin when the UART RXD function is not being used.

11.3 Sub-Block Description

The UART contains four sub-modules. This section briefly describes the basic functionality of the four blocks.

11.3.1 Transmitter

The transmitter accepts a parallel character from the CPU and transmits it serially. The start, stop, and parity (if enabled) bits are added to the character. The transmitter posts a maskable interrupt when it is ready for parallel data. $\overline{RTS}$ can be used to control the flow of the serial data. If $\overline{RTS}$ is negated (high), the transmitter finishes sending the character in progress (if any) then stops and waits for $\overline{RTS}$ to again become asserted (low).

A break character (continuous zeros) can be generated by the transmitter as well. For debugging purposes, parity errors can be generated. The transmitter operates from the 1x clock provided by the 16x bit clock generator.

Normal NRZ is transmitted when the infrared interface is disabled.

11.3.2 Receiver

The receiver accepts a serial data stream and converts it into a parallel character. When enabled, it searches for a start bit, qualifies it, and then samples the succeeding data bits at the bit-center. Jitter tolerance and noise immunity are provided by sampling at a 16x rate and using voting techniques to clean up the samples. Once the

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

start bit has been found, the data bits, parity bit (if parity is enabled), and stop bits are shifted in. If parity is enabled, it is checked and its status is reported in the RX register. Similarly, frame errors and breaks are checked and reported. When a new character is ready to be read by the host, RX READY is asserted and an interrupt is posted (if enabled). If the receiver register is read as a 16-bit word, the interrupt is automatically cleared and the data, along with four status bits, are read by the CPU.

$\overline{\text{CTS}}$ can be configured as an output to indicate a pending overrun.

Normal NRZ is expected when the infrared interface is disabled.

11.3.3 Infrared Interface

The infrared interface converts data to be transmitted or received as specified in the IRDA Serial Infrared Physical Layer Specification.

For each zero to be transmitted, a narrow pulse which is 3/16 of a bit time is generated. For each one to be transmitted, no pulse is generated. External circuitry must be provided to drive an infrared LED.

When the UART is receiving data, a narrow pulse is expected for each zero transmitted, and no pulse is expected for each one transmitted. Circuitry external to the IC transforms the infrared signal to an electrical signal.

11.3.4 16x Bit Clock Generator

The 16x bit clock generator provides the pre-scaled bit clocks to the transmitter and receiver blocks. A divide ratio from one to 4096 may be selected in the UBRGR register. The 16x bit clock generator provides sufficient flexibility to provide almost any “standard” bit-clock from a variety of clock frequencies.

NOTE

The baud rate error is computed as follows for 115.2 Kbps:

The input clock is	16.38 Mhz
The divide ratio selected is	9 (UBRGR[11:0] = 8)
Actual baud rate:	16.38 Mhz/9/16 = 113.75 khz
Actual-required rates ratio:	115.2/113.75 = 1.0127

This results in an error-per-bit ratio of 1.27%.

11.3.5 General UART Definitions

The following definitions apply to both the transmitter and receiver operation:

Bit Time — The time required to serially transmit or receive one bit of data.

Start Bit — One bit time of logic zero that indicates the beginning of a data frame. A start bit must begin with a one-to-zero transition.

Stop Bit — One bit time of logic one that indicates the end of a data frame.

**For More Information On This Product,
Go to: www.freescale.com**

Freescale Semiconductor, Inc.

Frame — A start bit, followed by a specified number of data or information bits, terminated by one or two stop bits. The number of data or information bits must agree between the transmitting and receiving devices. The most common frame format is one start bit followed by eight data bits (LSB first) terminated by one stop bit, for a total of ten bit times in the frame. The UART optionally provides other data formats as specified through the control registers.

Break — A frame in which all the bits are logic zero. This includes the stop bit, which is normally a logic one, as well as the data bits. This kind of a frame is generally sent to signal the end of a message or the beginning of a new message.

Framing Error — An error condition in which the stop bit of the received frame is missing. A framing error results when the frame boundaries in the received bit stream are not synchronized with the receiver bit counter. Framing errors are not always detected: if a data bit in the expected stop bit time happens to be a logic one, the framing error may go undetected. A framing error is always present on the receiver side, when the transmitter is sending breaks. However, if the UART is set up to expect two stop bits, and only one is received, then this is not a framing error by definition.

Parity Error — An error condition in which the calculated parity of the received data bits in the frame is different from the parity bit received on the RXD line. Parity error is only calculated after an entire frame is received.

Overrun Error — An error condition in which the latest character received is ignored to prevent overwriting an already existing character in the UART receiver FIFO. An overrun error indicates that the software reading the FIFO is not keeping up with the actual reception of characters on the RXD line.

11.4 UART Programming Model

This section describes the registers in the UART module. Each UART channel has the following independent set of registers:

- Three working registers (UCR1, UCR2, USR) that provide all status and control functions for the UART
- A separate test register (UTS) for those applications that need it
- Bit-rate generator register (UBRG) that controls the UART bit rate
- Separate transmit and receive registers
- Port control register that defines the GPIO functionality of UART pins.

The registers are optimized for a 16-bit bus. For example, all status bits associated with the received data are available along with the data in a single read. All register bits are readable (except TX DATA field in the UART transmit register), and most are read/write.

All registers may be accessed either as a halfword or as a byte. The RX and TX data registers may also be accessed as 32-bit words. For these registers the upper 16 bits are forced to zeros.

**For More Information On This Product,
Go to: www.freescale.com**

Freescale Semiconductor, Inc.

Table 11-1 UART Module Address Map

Address	Use	Access
UART0		
10009000	UART0 Receive Register (U0RX)	Supervisor Only
10009002	Not Used	Supervisor Only
10009004 to 1000903E	U0RX Echoes On Word Boundaries	Supervisor Only
10009040	UART0 Transmit Register (U0TX)	Supervisor Only
10009042	Reserved	Supervisor Only
10009044 to 1000907E	U0TX Echoes On Word Boundaries	Supervisor Only
10009080	UART0 Control Register 1 (U0CR1)	Supervisor Only
10009082	UART0 Control Register 2 (U0CR2)	Supervisor Only
10009084	UART0 Baud Rate Generator Register (U0BRGR)	Supervisor Only
10009086	UART0 Status Register (U0SR)	Supervisor Only
10009088	UART0 Test Register (U0TSR)	Supervisor Only
1000908A	UART0 Port Control Register (U0PCR)	Supervisor Only
1000908C	UART0 Data Direction Register (U0DDR)	Supervisor Only
1000908E	UART0 Port Data Register (U0PDR)	Supervisor Only
10009090 to 10009FFF	Reserved	Supervisor Only
UART1		
1000A000	UART1 Receive Register (U1RX)	Supervisor Only
1000A002	Not Used	Supervisor Only
1000A004 to 1000A03E	U1RX Echoes On Word Boundaries	Supervisor Only
1000A040	UART1 Transmit Register (U1TX)	Supervisor Only
1000A042	Reserved	Supervisor Only
1000A044 to 1000A07E	U1TX Echoes On Word Boundaries	Supervisor Only
1000A080	UART1 Control Register 1 (U1CR1)	Supervisor Only
1000A082	UART1 Control Register 2 (U1CR2)	Supervisor Only
1000A084	UART1 Baud Rate Generator Register (U1BRGR)	Supervisor Only
1000A086	UART1 Status Register (U1SR)	Supervisor Only
1000A088	UART1 Test Register (U1TSR)	Supervisor Only
1000A08A	UART1 Port Control Register (U1PCR)	Supervisor Only
1000A08C	UART1 Data Direction Register (U1DDR)	Supervisor Only
1000A08E	UART1 Port Data Register (U1PDR)	Supervisor Only
1000A088 to 1000AFFF	Reserved	Supervisor Only
1000B000 to 1FFFFFFF	Not Used (Access causes transfer error)	Not Applicable

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

11.4.1 UART Receive Register (URX)

This read-only register contains received characters and status. After reset, if the receiver is enabled (RXEN = 1), the CHAR RDY bit is zero until the first character is received, and the remainder of the register contents are undefined. The RX register is echoed to 16 word addresses in order to support unloading the FIFO with the load register quadrant (**ldq**) instruction.

U0RX — UART0 Receive Register **10009000**
U1RX — UART1 Receive Register **1000A000**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHAR RDY	ERR	OVR RUN	FRM ERR	BRK	PR ERR	0	0	RX DATA							
W																
RESET:	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

*n = 0, 1, . . ., 15
X = Undefined

Figure 11-2 UART Receive Register

CHARRDY — Character Ready

This read-only bit indicates whether the character in the RX DATA field and associated flags are valid and ready to be read by the host.

0 = Character in RX DATA field and associated flags are invalid
1 = Character in RX DATA field and associated flags valid and ready for reading

At reset, this bit is cleared to zero.

ERR — Error Detect

When set, this read-only bit indicates that the character present in the RX DATA field has an error status. The error can be an OVRRUN, FRMERR, BRK or PRERR. This bit is updated and valid for each received character.

0 = No error status detected
1 = Error status detected

At reset, this bit is cleared to zero.

OVRRUN — Receiver Overrun

When set, this read-only bit indicates that the receiver ignored data to prevent overwriting the data in the FIFO. Under normal circumstances, this bit should never be set. It indicates that the user's software is not keeping up with the incoming data rate. This bit is updated and valid for each received character, and when set indicates that some number of characters were lost *following* the character for which the flag is set.

0 = No FIFO overrun
1 = A FIFO overrun was detected

At reset, this bit is cleared to zero.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

FRMERR — Frame Error

When set, this read-only bit indicates that the current character had a framing error (missing stop bit). The data is possibly corrupted. This bit is updated for each character read from the FIFO.

Every attempt is made to allow the receiver to correctly interpret data following a character marked as having a framing error. (This includes ignoring the start bit validation logic, if appropriate.) However, if the transmitted data includes two stop bits, and both stop bits are incorrect, then the second stop bit will be interpreted as the start bit of the next character.

0 = Character has no framing error
1 = Character has a framing error

At reset, this bit is cleared to zero.

BRK — Break Detect

When set, this read-only bit indicates that the current character was detected as a break. The data bits are all zero and the stop bit is also zero. The frame error bit is always set when this bit is set. If odd parity is selected, parity error will also be set when this bit is set. This bit is valid for each character read from the FIFO.

0 = Character is not a break character
1 = Character is a break character

At reset, this bit is cleared to zero.

PRERR — Parity Error

When set, this read-only bit indicates that the current character was detected with a parity error. The data is possibly corrupted. This bit is updated for each character read from the FIFO. While parity is disabled, this bit always reads zero.

0 = No parity error detected for data in RX DATA field
1 = Parity error detected for data in RX DATA field

At reset, this bit is cleared to zero.

RX DATA — Received Data

These read-only bits are the received character. In 7-bit mode, the MSB is forced to zero. In 8-bit mode, all bits are active.

11.4.2 UART Transmitter Register (UTX)

The UART transmitter register is used by the host to write the data to be transmitted. The low byte is write-only. When this register is read, bits TX[15:8] always return zero, and TX[7:0] is not driving the bus. The TX register is echoed to 16-word addresses in order to support filling the transmit FIFO with the store register quadrant (**stq**) instruction.

**For More Information On This Product,
Go to: www.freescale.com**

Freescale Semiconductor, Inc.

U0TX — UART0 Transmitter Register **10009040**
U1TX — UART1 Transmitter Register **1000A040**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0								
									TX DATA							
RESET:	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X

*n = 0, 1, . . ., 15
X = Undefined

Figure 11-3 UART Transmitter Register

TX DATA — Transmit Data

These write-only bits are the parallel transmit data inputs. In 7-bit mode, D7 is ignored. In 8-bit mode, all bits are used. Data is transmitted LSB first. A new character is transmitted when these bits are written. These bits must be written only while TRDY is high to ensure that corrupted data is not sent.

11.4.3 UART Control Register 1 (UCR1)

UART control register 1 is a read/write register. This register enables the UART and the transmit and receive blocks. It controls the Tx and Rx FIFO levels and enables the TRDY and RRDY interrupts.

U0CR1 — UART0 Control Register 1 **10009080**
U1CR1 — UART1 Control Register 1 **1000A080**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TxFL		TRDY-EN	TXEN	RxFL		RRDY EN	RXEN	IREN	0	RTSD EN	SND-BRK	0	0	DOZE	UART EN
W																
RESET:	0	0	0	0	0	0	0	0	0		0	0			0	0

Figure 11-4 UART Control Register 1

TxFL — Transmitter FIFO Interrupt Trigger Level

These bits control the operation of the interrupt generated by the transmitter. A maskable interrupt is generated whenever the data level in the TX FIFO drops below the selected threshold. The bits are encoded as follows:

Table 11-2 TxFL Field Settings

Value	Meaning
00	Interrupt if TX FIFO has a slot for one or more character
01	Interrupt if TX FIFO has a slot for four or more characters
10	Interrupt if TX FIFO has a slot for eight or more characters
11	Interrupt if TX FIFO has a slot for fourteen or more characters

At reset, these bits are cleared to zero.

**For More Information On This Product,
Go to: www.freescale.com**

Freescale Semiconductor, Inc.

TRDYEN — Transmitter Ready Interrupt Enable

Setting this bit enables an interrupt when the transmitter has one or more slots available in the TX FIFO. The fill level in the TX FIFO at which an interrupt is generated is controlled by the TxFL bits. While this bit is negated, the transmitter interrupt is disabled.

0 = TX interrupt disabled
1 = TX interrupt enabled

At reset, this bit is cleared to zero.

TXEN — Transmitter Enable

This bit enables or disables the transmitter. While UARTEN and TXEN bits are set, and DOZE bit is cleared, the transmitter is enabled. If this bit is cleared in the middle of a transmission, the UART disables the transmitter immediately and starts marking ones.

The transmitter FIFO cannot be written to when this bit is cleared.

0 = Transmitter disabled
1 = Transmitter enabled

At reset, this bit is cleared to zero.

RxFL — Receiver FIFO Interrupt Trigger Level

These bits control the threshold at which a maskable interrupt will be generated by the receiver. A maskable interrupt will be generated whenever the data level in the RX FIFO reaches the selected threshold.

Table 11-3 RxFL Field Settings

Value	Meaning
00	Interrupt if RX FIFO contains one or more character
01	Interrupt if RX FIFO contains four or more characters
10	Interrupt if RX FIFO contains eight or more characters
11	Interrupt if RX FIFO contains fourteen or more characters

At reset, these bits are cleared to zero.

RRDYEN — Receiver Ready Interrupt Enable

Setting this bit enables an interrupt when the receiver has data in the RX FIFO. The fill level in the RX FIFO at which an interrupt is generated is controlled by the RxFL bits. Clearing this bit disables RX interrupts.

0 = RX interrupt disabled
1 = RX interrupt enabled

At reset, this bit is cleared to zero.

RXEN — Receiver Enable

Setting this bit enables the receiver. When the receiver is enabled, if the RXD line is already low, the receiver does not recognize break characters, since it requires a valid one-to-zero transition before it can accept any character.

0 = Receiver disabled
1 = Receiver enabled

At reset, this bit is cleared to zero.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

IREN — Infrared Interface Enable

Setting this bit enables the infrared interface. Refer to **11.3.3 Infrared Interface**.

0 = Infrared interface disabled
1 = Infrared interface enabled

At reset, this bit is cleared to zero.

RTSD EN — RTS Delta Interrupt Enable

This bit enables or disables RTS delta interrupts. The current status of the $\overline{\text{RTS}}$ pin is read in the UART status register.

0 = RTS interrupt disabled
1 = RTS interrupt enabled

At reset, this bit is cleared to zero.

SNDBRK — Send Break

This bit forces the transmitter to send a break character. The transmitter will finish sending the character in progress (if any) and then send break characters until this bit is reset. The user is responsible for ensuring that this bit is high for a sufficient period of time to generate a valid break; the transmitter samples SNDBRK after every bit is transmitted.

Following completion of the break transmission, the UART transmits two mark bits. The user can continue to fill the FIFO, and any characters remaining will be transmitted when the break is terminated. This bit cannot be changed until the UART EN and TX EN bits in the UART control register 1 (CR1) are set.

0 = Do not send break
1 = Send break (continuous zeros)

At reset, this bit is cleared to zero.

DOZE — Doze Mode

When the CPU executes a **doze** instruction and the system is placed in the doze mode, the DOZE bit affects operation of the UART. If this bit is set when the system is in the doze mode, the UART is disabled. Refer to **11.7 UART Operation in Low-Power System Modes**.

0 = UART unaffected in doze mode
1 = UART disabled in doze mode

At reset, this bit is cleared to zero.

UART EN — UART Enable

This bit enables or disables the UART. If this bit is cleared in the middle of a transmission, the transmitter stops and drives the TXD line to logic one.

0 = UART disabled
1 = UART enabled

At reset, this bit is cleared to zero.

11.4.4 UART Control Register 2 (UCR2)

UART control register 2 is a read/write register. This register controls the overall operation of the UART. It controls the clock source, number of bits per character, parity generation and checking, and behavior of the $\overline{\text{RTS}}$, $\overline{\text{CTS}}$, and $\overline{\text{DTR}}$ pins.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

U0CR2 — UART0 Control Register 2 **10009082**
U1CR2 — UART1 Control Register 2 **1000A082**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	IRTS	CTSC	CTS	0	0	0	PREN	PROE	STPB	WS	0	0	0	0	0
W																
RESET:		0	0	0				0	0	0	0					

Figure 11-5 UART Control Register 2

IRTS — Ignore $\overline{\text{RTS}}$

Setting this bit forces the $\overline{\text{RTS}}$ input signal presented to the transmitter to always be asserted, effectively causing the external pin to be ignored. In this mode, the $\overline{\text{RTS}}$ pin can be used as a general-purpose input.

0 = Transmit only while $\overline{\text{RTS}}$ pin is asserted
1 = Ignore $\overline{\text{RTS}}$ pin

At reset, this bit is cleared to zero.

CTSC — $\overline{\text{CTS}}$ Pin Control

This bit controls the operation of the $\overline{\text{CTS}}$ output pin. While this bit is set, the $\overline{\text{CTS}}$ output pin is controlled by the receiver. When the RX FIFO has a pending overrun, the $\overline{\text{CTS}}$ output is negated to indicate to the far-end transmitter to stop transmitting. While the CTSC bit is negated, the $\overline{\text{CTS}}$ output pin is controlled by the CTS bit.

On reset, since this bit is cleared to zero, the $\overline{\text{CTS}}$ pin is controlled by the CTS bit, which is also cleared to zero on reset. This means that on reset the $\overline{\text{CTS}}$ signal is negated.

0 = $\overline{\text{CTS}}$ pin controlled by the CTS bit
1 = $\overline{\text{CTS}}$ pin controlled by the receiver

At reset, this bit is cleared to zero.

CTS — CTS bit

This bit controls the $\overline{\text{CTS}}$ pin while the CTSC bit is negated. While CTSC is asserted this bit has no function.

0 = $\overline{\text{CTS}}$ pin is driven high (inactive)
1 = $\overline{\text{CTS}}$ pin is driven low (active)

At reset, this bit is cleared to zero.

PREN — Parity Enable

This bit enables or disables the parity generator in the transmitter and parity checker in the receiver.

0 = Parity disabled
1 = Parity enabled

At reset, this bit is cleared to zero.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

PROE — Parity Odd/Even

This bit controls the sense of the parity generator and checker. When PROE is set, odd parity is generated and expected. When PROE is cleared, even parity is generated and expected. This bit has no function if PREN is low.

0 = Even parity
1 = Odd parity

At reset, this bit is cleared to zero.

STPB — Stop Bits

This bit controls the number of stop bits transmitted after a character. When STPB is set, two stop bits are sent. When STPB is cleared, one stop bit is sent. This bit has no effect on the receiver, which expects one or more stop bits.

0 = One stop bit transmitted
1 = Two stop bits transmitted

At reset, this bit is cleared to zero.

WS — Word Size

This bit specifies a character length of eight or seven bits (not including start, stop, or parity bits). When WS is set, the transmitter and receiver are in eight-bit mode. When WS is cleared, they are in seven-bit mode. The transmitter then ignores B7, and the receiver sets B7 to zero. This bit can be changed between transmissions or receptions. If it is changed while a transmission or reception is in progress, however, the length of the current character being transmitted or received is unpredictable.

0 = 7-bit transmit and receive character length
1 = 8-bit transmit and receive character length

At reset, this bit is cleared to zero.

11.4.5 UART BRG Register (UBRGR)

This register specifies the divide ratio of the prescaler in the UART bit clock generator.

U0BRGR — UART0 BRG Register **10009084**
U1BRGR — UART1 BRG Register **1000A084**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	CD											
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 11-6 UART BRG Register

CD — Clock Divider

These bits determine the bit clock generator output rate. The CD field is used to preset a 12-bit counter that is decremented at the system clock rate. The value 0x000 produces the maximum clock rate (equal to the system clock). The value 0xFFF produces the minimum clock rate (divide by 4096).

**For More Information On This Product,
Go to: www.freescale.com**

Freescale Semiconductor, Inc.

11.4.6 UART Status Register (USR)

The read/write UART status register indicates the status of the $\overline{RTS}$ pin, input transitions on the pin, and status of the transmit and receive FIFOs.

U0SR — UART0 Status Register **10009086**
U1SR — UART1 Status Register **1000A086**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX MPTY	RTSS	TRDY	0	0	0	RRDY	0	0	0	RTSD	0	0	0	0	0
W																
RESET:	1	0	1				0				0					

Figure 11-7 UART Status Register

TXMPTY — Transmitter Empty

When set, this bit indicates that the transmit FIFO and the transmit shift register are both empty. This bit is automatically cleared when a write to the TX FIFO is performed.

0 = TX FIFO or shifter are not both empty
1 = TX FIFO and shifter are both empty

At reset, this bit is set to one.

RTSS — $\overline{RTS}$ Pin Status

This bit indicates the current status of the $\overline{RTS}$ pin. A “snapshot” of the pin is taken immediately before this bit is presented to the data bus. While IRTS is asserted, this bit can be used as a general-purpose input.

0 = $\overline{RTS}$ pin is high (inactive)
1 = $\overline{RTS}$ pin is low (active)

This bit follows the logic value connected to the $\overline{RTS}$ pin.

TRDY — Transmitter Ready Interrupt Flag

When set, this bit indicates that the TX FIFO has emptied below its target threshold and needs data. This bit is automatically cleared when the data level in the TX FIFO goes beyond the set threshold level.

0 = Transmitter does not need data
1 = Transmitter needs data (interrupt posted)

At reset, this bit is set to one.

RRDY — Receiver Ready Interrupt Flag

When set, this bit indicates that the receive FIFO data level is above the threshold level specified by the RxFL field, and a maskable interrupt is generated. Refer to the RxFL bit description for setting the threshold level. In conjunction with the CHARRDY bit, host software can continue to read the RX FIFO in an interrupt service routine until the RX FIFO is empty. This bit is automatically cleared when the data level in the RX FIFO goes below the set threshold level.

0 = No character ready (no interrupt posted)
1 = Character(s) ready (interrupt posted)

At reset, this bit is cleared to zero.

**For More Information On This Product,
Go to: www.freescale.com**

Freescale Semiconductor, Inc.

RTSD — RTS Delta

When set, this bit indicates that the $\overline{RTS}$ pin changed state. It generates a maskable interrupt. In STOP mode, $\overline{RTS}$ assertion sets this bit to wake the CPU. The current state of the $\overline{RTS}$ pin is available in the RTSS bit. The RTSD interrupt is cleared by writing a one to this bit.

0 = $\overline{RTS}$ pin did not change state since last cleared
1 = $\overline{RTS}$ pin changed state

At reset, this bit is cleared to zero.

11.4.7 UART Test Register (UTS)

The UART test register is a read/write register. Unimplemented bits always return zero when read. This register contains miscellaneous bits to control test features of the UART block.

U0TS — UART0 Test Register **10009088**
U1TS — UART1 Test Register **1000A088**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	FRC	LOOP	0	LOOP	0	0	0	0	0	0	0	0	0	0
W			PERR			IR										
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 11-8 UART Test Register

FRC PERR — Force Parity Error

When set, this bit forces the transmitter to generate a parity error if parity is enabled. This bit is provided for system debugging.

0 = Generate normal parity
1 = Generate inverted parity (error)

At reset, this bit is cleared to zero.

LOOP — Loop TX and RX for Test

This bit controls loopback for test purposes. While this bit is high, the receiver input is internally connected to the transmitter and ignores the RxD pin. The transmitter is unaffected by this bit. This loopback operates to connect the data on the TxD pin directly to the voting logic. If infrared mode is enabled (IR_EN is active), the effect of activating this bit is to put an IR-formatted bit stream into the voting logic, which will yield odd results. Do not use this loopback if IR_EN is active.

0 = Normal receiver operation
1 = Internal connect transmitter output to receiver input

At reset, this bit is cleared to zero.

LOOP IR — Loop TX and RX for IR Test

This bit controls a loopback from transmitter to receiver in the infrared interface.

0 = No IR loop
1 = Connect IR transmit to IR receiver

At reset, this bit is cleared to zero.

**For More Information On This Product,
Go to: www.freescale.com**

Freescale Semiconductor, Inc.

11.5 GPIO Pins and Registers

The GPIO functionality of UART pins is controlled by three registers: the port control register (UPCR), data direction register (UDDR), and port data register (UPDR).

Table 11-4 UART Pins GPIO Assignment

GPIO Bit	UART Pin
P0	RXD
P1	TXD
P2	RTS
P3	CTS

11.5.1 UART Port Control Register (UPCR)

The read/write UART port control register controls the functionality of UART GPIO pins.

U0PCR — UART0 Port Control Register **1000908A**
U1PCR — UART1 Port Control Register **1000A08A**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	PC3	PC2	PC1	PC0
W																
RESET:													0	0	0	0

Figure 11-9 UART Port Control Register

PCx — Port Control Bit x
- 0 = Corresponding pin is configured as GPIO pin
- 1 = Corresponding pin is configured as UART pin

At reset, these bits are cleared to zero.

11.5.2 UART Data Direction Register (UDDR)

The read/write UART data direction register controls the direction of UART GPIO pins.

U0DDR — UART0 Data Direction Register **1000908C**
U1DDR — UART1 Data Direction Register **1000A08C**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	PDC3	PDC2	PDC1	PDC0
W																
RESET:													0	0	0	0

Figure 11-10 UART Data Direction Register

**For More Information On This Product,
Go to: www.freescale.com**

Freescale Semiconductor, Inc.

PDCx — Port Direction Control Bit x

0 = Corresponding GPIO pin is configured as input
1 = Corresponding GPIO pin is configured as output

At reset, these bits are cleared to zero.

11.5.3 UART Port Data Register (UPDR)

The UART port data register is used to read or write data to or from UART GPIO pins.

U0PDR — UART0 Port Data Register **1000908E**
U1PDR — UART1 Port Data Register **1000A08E**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	PD3	PD2	PD1	PD0
W																
RESET:													X	X	X	X

X = Undefined

Figure 11-11 UART Port Data Register

PDx — Port Data Bit x

These bits are used to read or write data from/to the corresponding port pins if they are configured as GPIO (by PC[3:0] bits in UPCR). If a port pin x is configured as a GPIO input, then the corresponding PDx bit will reflect the value present on this pin. If a port pin x is configured as a GPIO output, then the value written into the corresponding PDx bit will be reflected on the pin.

Note that since the $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ pins are not present for UART1, the corresponding port control register bits should be configured in a manner which provides deterministic data when the port data register is read. One method for doing so is to configure the missing pins as general-purpose outputs.

11.6 Data Sampling Technique on the Receiver

The UART receiver is responsible for synchronization to the serial data stream and recovery of data characters. Since the data stream has no clock, data recovery depends on the transmitting device and the receiving device operating at close to the same bit rate. The UART system can tolerate a moderate amount of system noise without losing any information.

The UART receive function is somewhat more difficult than the transmit function due to the asynchronous nature of incoming serial data. A discussion of the way the UART recognizes a start bit follows.

The receiver front-end logic uses a sampling clock that is 16 times the bit rate. This sampling clock is called the RT clock in the following discussion, and one RT is understood to be one-sixteenth of a bit-time. In the following figures, the RT clock cycles are numbered from one (start of a bit time) to sixteen (end of a bit time).

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

When the receiver is first enabled and after the reception of a stop bit at the end of a frame, an asynchronous search is initiated to find the leading edge of the next start bit. (In the case of a framing error, because the stop bit has been somehow corrupted, the start-bit validation logic may not operate (because there may not be a negative edge to detect). As long as the voting logic is able to detect a start bit, the receiver will be able to continue receiving characters following a framing error.)

The goal of this asynchronous search is to gain bit-time synchronization between the serial data stream and the internal RT clock. Once synchronization has been established, the RT clock controls where the UART perceives the bit-time boundaries to be. The first step in locating a start bit is to find a sample where RXD is zero preceded by four consecutive samples of logic ones. These five samples are called start-bit qualifiers. Until the start-bit qualifiers are detected, the RT clock is reset to state RT1 after each sample.

Once the qualifiers are found, the beginning of a start bit is tentatively assumed, and successive samples are assigned successive RT state numbers. The next six samples are taken as start-bit verification samples. If even one of these is a logic one, the low at RT1 is assumed to have been noise, and the asynchronous search is started again. When the start-bit qualifiers and the start-bit verification requirements are met, synchronization has been achieved, and the RT count state is used to determine the position of bit-time boundaries.

During each bit time, including the start and stop bit times, data samples are taken to determine the logic sense of the bit time. The samples are taken at RT9, RT10, and RT11 or at RT8, RT9, and RT10, depending upon the synchronization of the incoming data. The logic sense of the bit time is considered to be the majority of the three samples under consideration. At the end of a character reception, data is transferred from the shift register to the parallel receiver register, with the corresponding flags updated in the receiver register and the status register.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

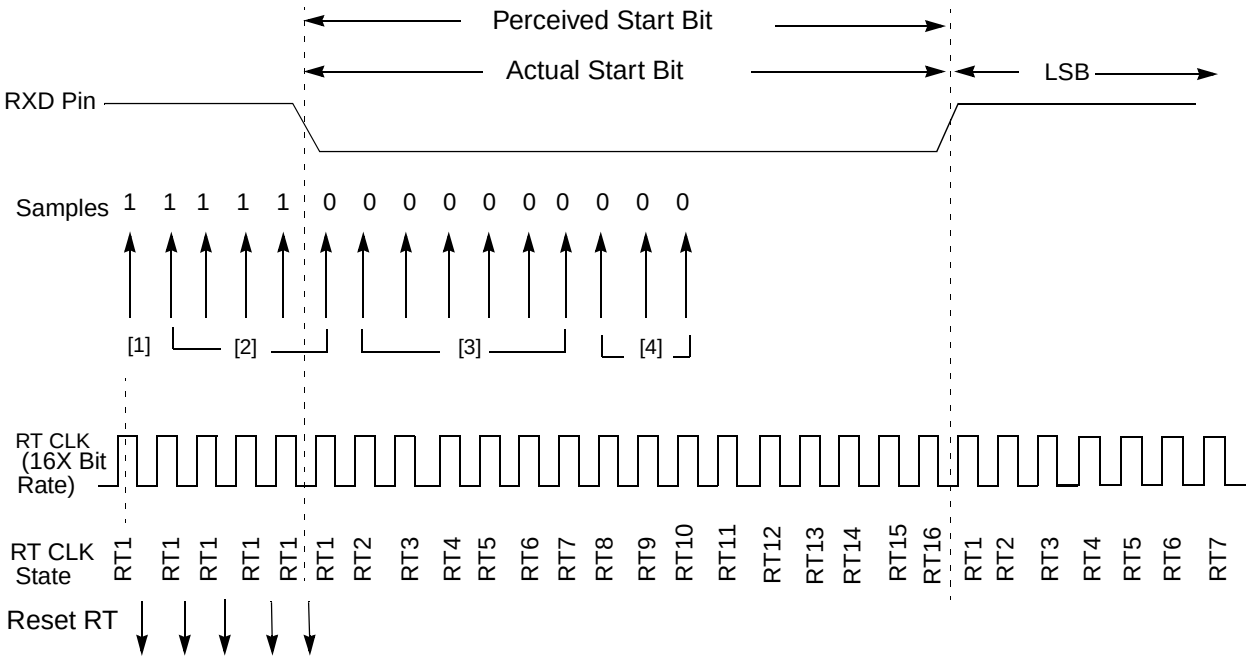


Figure 11-12 Start Bit — Ideal Case

Figure 11-12 shows the details of the ideal case of start-bit recognition. All samples taken at [1] detect logic ones on the RXD line and correspond to the idle-line time or a stop-bit time prior to this start bit. At [2] a logic zero sample is preceded by four logic one samples. These five samples are the start-bit qualifiers. The beginning of the start bit time is tentatively perceived to occur between the fourth logic one sample and the logic zero sample of the start qualifiers. Next, the samples at RT2, RT3, RT4, RT5, RT6, and RT7 [3] are taken to verify that this bit time is indeed the start bit. The samples at RT8, RT9, and RT10 (or RT9, RT10, and RT11) are called the data samples [4]. These samples drive a majority voting circuit to determine the logic sense of the bit time.

In this ideal case, the actual start bit and the perceived start bit match. The resolution of the RT clock leads to an uncertainty about the exact placement of the leading edge of the start bit. The uncertainty in the placement of the edge will be one-sixteenth of a bit-time.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

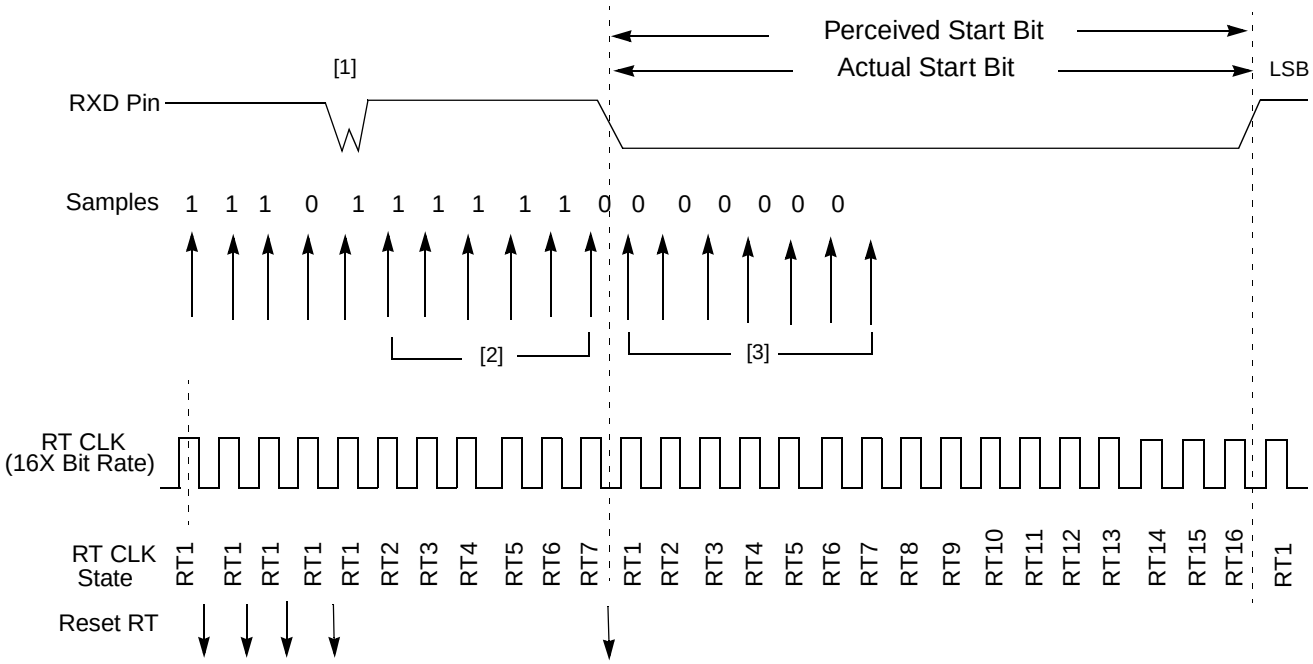


Figure 11-13 Start Bit — Noise Case One

Figure 11-13 shows what occurs if noise causes a sample to be erroneously detected as a zero before the actual beginning of the start bit. The logic zero sample [1] in conjunction with the four preceding samples of logic one meet the conditions for start qualification; thus, logic tentatively perceives the start bit as beginning here. Subsequent start-verification samples at RT2, RT3, RT4, RT5, RT6, and RT7 [2] are not all logic zeros; therefore, the tentative placement of the start edge is rejected, and the search is restarted. When the sample at the actual beginning of the start bit is detected, the preceding three samples are ones; the start bit is now perceived to begin here. In this case, the three samples taken at RT2, RT3, RT4, RT5, RT6, and RT7 [3] now verify that the start bit has been found. If the noise bit [1] is further away from the beginning of the actual start bit, the perceived start bit will still be correct.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

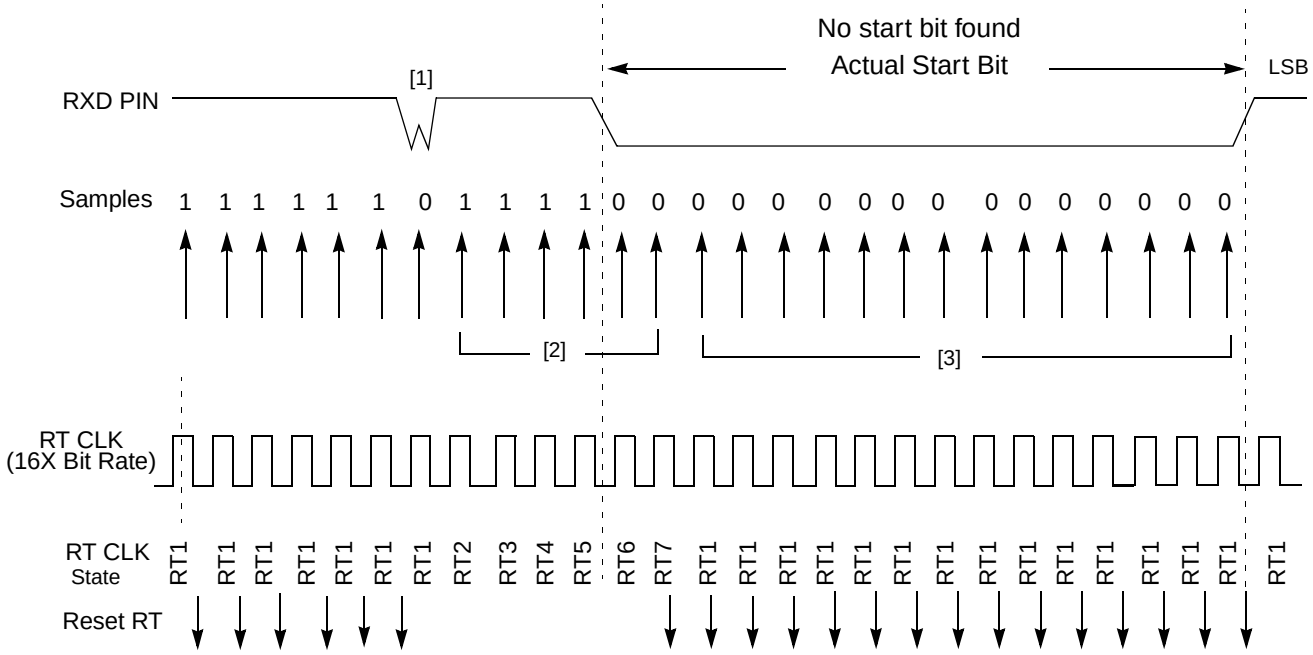


Figure 11-14 Start Bit — Noise Case Two

Figure 11-14 is similar to the previous case except noise [1] is now closer to the actual beginning of the start bit. The noise sample and the preceding four logic ones meet the start qualification requirements. The start verification samples at RT2, RT3, RT4, RT5, RT6, and RT7 [2] are not all zeros; therefore, the tentative placement of the start edge is rejected, and the search for the start qualifiers is restarted at [3]. Since there are no more cases of four logic ones in a row [3], the start bit is never detected. Because the circuit could not locate the start bit, the frame will be received as a framing error, improperly received, or missed entirely, depending on the data in the frame and when the start logic synchronized on what it thought was a start bit. This causes incorrect data reception.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

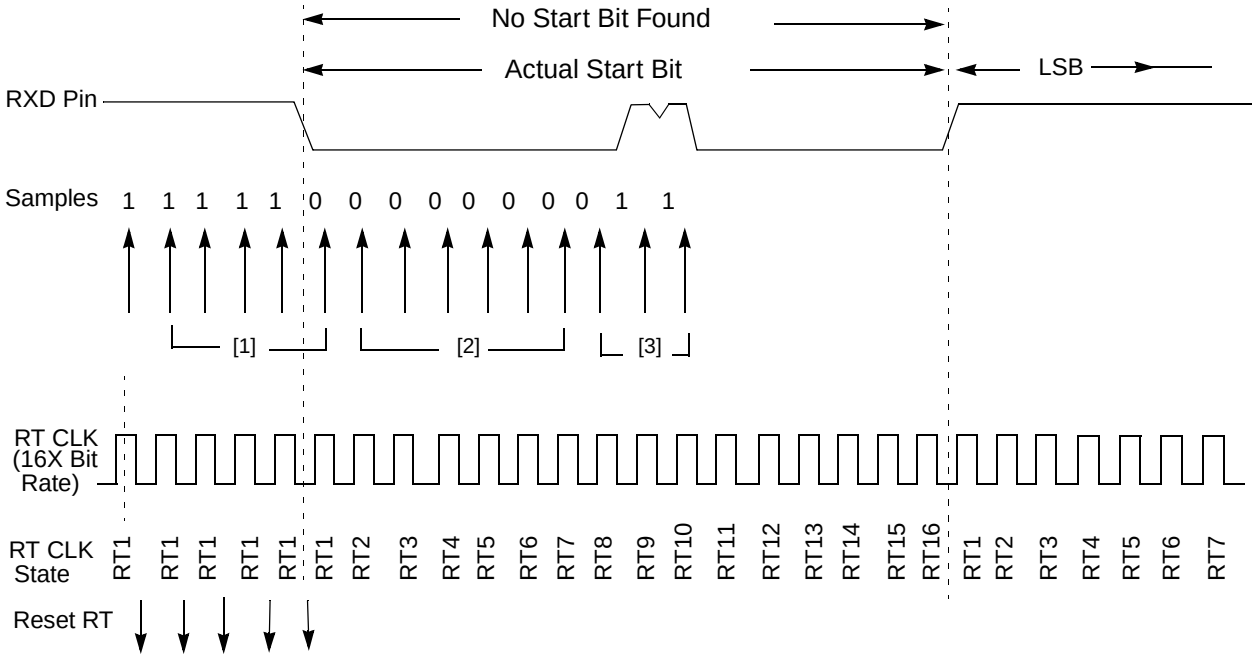


Figure 11-15 Start Bit — Noise Case Three

Figure 11-15 illustrates the case where the start edge is qualified by the four logic ones followed by a logic zero [1]. A burst of noise is present in the middle of the start bit time. The noise [3] causes two out of the three data samples to be erroneously detected as logic ones. This is rejected, therefore, as a start bit, because the majority of samples RT8, RT9, and RT10 [3] suggest it should be a logic one.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

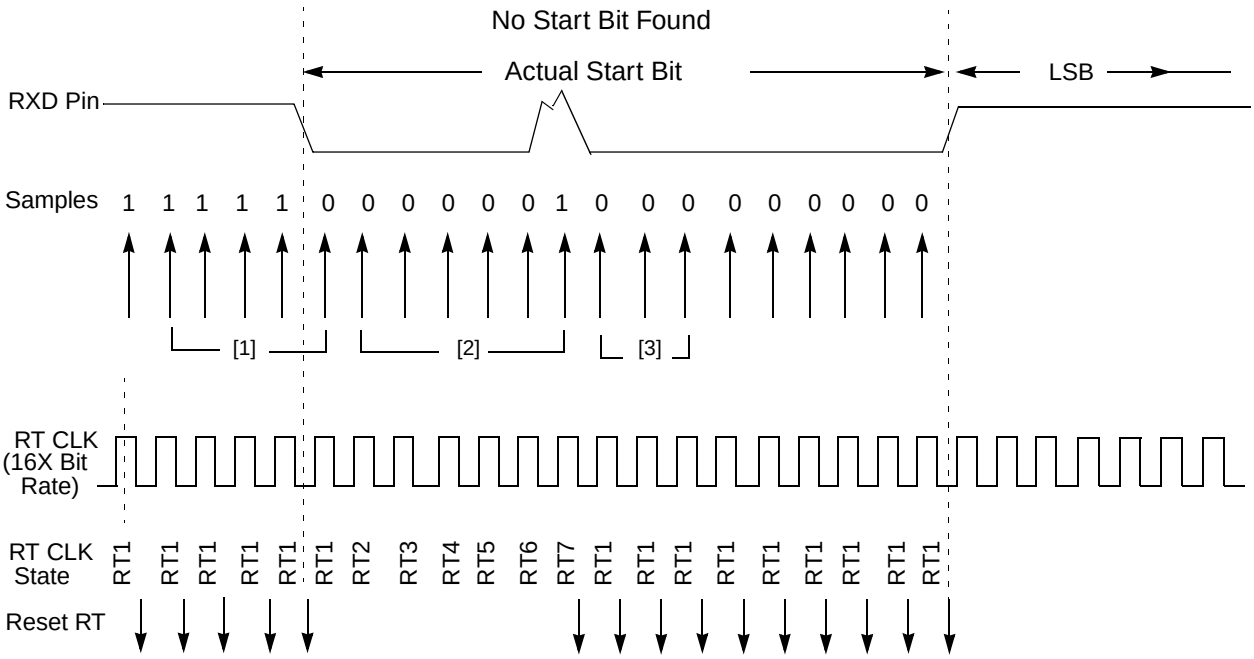


Figure 11-16 Start Bit — Noise Case Four

The case depicted in **Figure 11-16** is similar to the previous case. In this case the start bit is not detected. The frame will be lost or in error, depending on the data in the actual frame and when the start logic got synchronized. In this case, [1] qualifies the start edge. However, the samples in [2] are not all logic zero due to the presence of noise which causes the sample RT7 to be erroneously read as logic one. Thus, the tentative start edge is rejected and the search for the start edge qualifier is started once again. This happens even though all the data samples in the start bit [3] were logic zero. If this were anything but a start bit, the voting logic would have taken the right decision.

11.7 UART Operation in Low-Power System Modes

The UART serial interface operates as long as the 16x bit clock generator is provided with the system clock. The peripheral interface is operational while the CPU_CLK is running. The three bits RXEN, TXEN and UART EN, set by the user, give the capability to control low-power modes through software. **Table 11-5** shows UART functionality while in hardware controlled low-power modes.

Table 11-5 UART Low-Power Mode Operation

	Normal Mode	Wait Mode	Doze Mode		Stop Mode
			DOZE = 0	DOZE = 1	
System Clock	On	On	On	Off	Off
UART Serial Interface	On	On	On	Off	Off
Module Interface	On	On	On	Off	Off

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

In doze mode, the UART behavior depends on the DOZE control bit. While the DOZE bit is cleared, the UART serial interface is unaffected. While the system is in doze mode, and the DOZE bit is set, the UART is disabled. If the doze mode is entered with the DOZE bit set while the UART serial interface was receiving or transmitting data, it will finish receiving or transmitting the current character and signal to the far-end transmitter or receiver to stop sending or receiving. There is no guarantee that the data currently present in the receiver and transmitter FIFOs will not be corrupted.

11.8 UART Operation in System Debug Mode

In debug mode, reads of the UART receiver register do not cause the RX FIFO to “bump.” That is, the value at the read side of the RX FIFO does not change as a result of reading this register in debug mode. Repeated reads of the UART receiver register, therefore, do not cause its value to change once it contains a valid character.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 12
INTERVAL MODE SERIAL PERIPHERAL INTERFACE

The interval mode serial peripheral interface (ISPI) module provides a high-speed synchronous serial interface to communicate to external devices such as A/D converters and non-volatile RAMs. The ISPI provides the control and clock for data transfers and can be configured as either a master or a slave device. In addition, the ISPI includes a timer that delays the initiation of a serial transfer for a programmable period.

12.1 Overview

The ISPI transfers data between the MMC2001 and a peripheral device over a serial link. Enable and clock signals are used to exchange data between the two devices. If the external device is a transmit-only device (e.g., an A/D converter), the ISPI output port can be ignored or used for other purposes. **Figure 12-1** shows a block diagram of the ISPI.

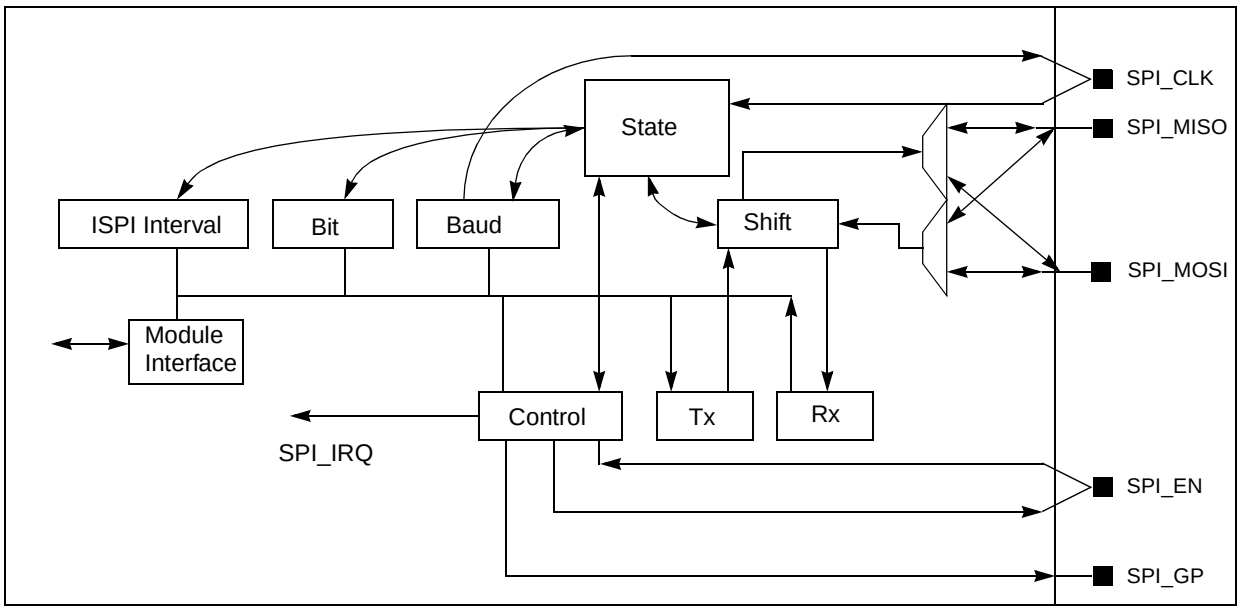


Figure 12-1 ISPI Channel Block Diagram

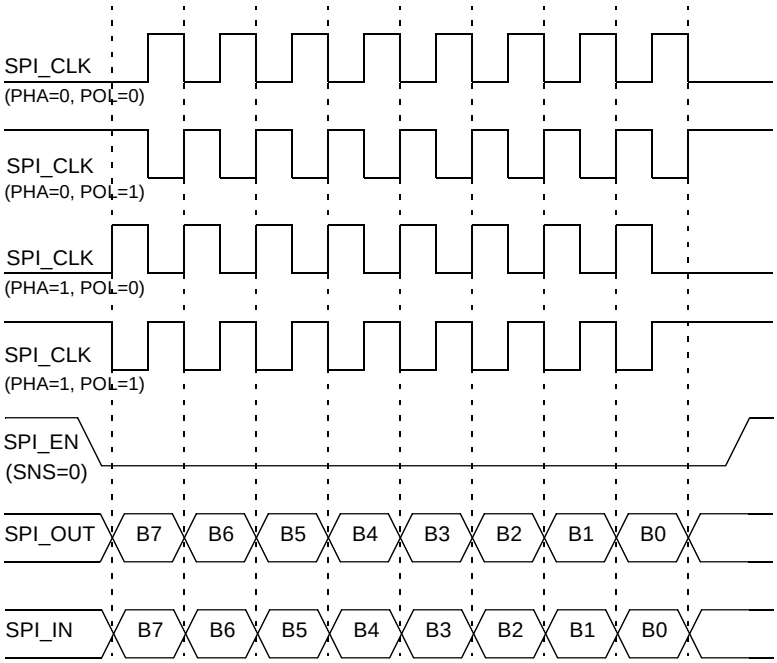
12.2 Operation

The ISPI provides three operating modes. Manual mode is a traditional SPI master operation mode. Interval mode is similar to manual mode, except that it includes a programmable timer to support timed transfers. Interval mode is suitable for controlling an external sound DAC, for example.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

In slave mode, the ISPI operates as a traditional slave SPI; the clock becomes an input, and the transfer is controlled entirely by the external master device.



Note: SPI_IN and SPI_OUT can appear on either SPI_MOSI and SPI_MISO, depending on the mode selected by the MSTR bit.

Figure 12-2 Timing Diagram of ISPI 8-Bit Operation

The ISPI supports clocked transfers of all variations of phase and polarity by controlling SPI_CLK (see **Figure 12-2**). Under normal phase (PHA=0), data is latched with the leading edge of SPI_CLK, and data changes with the trailing edge of SPI_CLK. For normal phase, the leading edge is rising if POL=0, and is falling if POL=1. Under opposite phase (PHA=1), data changes on the leading edge of SPI_CLK and is latched on the trailing edge. For opposite phase, the leading edge is rising if POL=0 and is falling if POL=1. This flexibility allows operation with most serial peripheral devices on the market.

12.2.1 Manual (Master) Mode

When a data exchange is needed, the user sets the SPI_EN bit in the ISPI control register. Control values such as the number of transfer clocks, polarity, and phase are also loaded into the ISPI control register. The transfer is initiated by writing the ISPI Tx data register. During the transfer, data in the shift register is exchanged with data in the peripheral. Setting the IRQ_EN bit enables the posting of an interrupt upon completion of the transfer. The user then negates the SPI_EN register bit to complete the operation.

For systems that need more than 16 clocks to transfer data, the SPI_EN bit can remain set between exchanges.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

12.2.2 Interval (Master) Mode

Interval mode provides the user with the ability to exchange data at programmed periodic intervals. This rate is controlled by three counters: the bit counter, the baud counter, and the ISPI interval timer. This mode begins operation as soon as the IVL_EN bit is set in the ISPI control register. (If a transfer is in progress, then operation begins upon completion of the existing transfer.) In interval mode (in contrast to manual mode), the SPI_EN pin is active only when a transfer is in progress; that is, in interval mode the state machine controls the enable pin. The SPI_EN enable bit (bit 12) must still be set in the ISPI control register.

An interval begins with the loading of the actual ISPI interval timer. Once this decrementing counter reaches zero, the state machine begins the data transfer. When the transfer is completed, an interrupt is generated (if enabled by IRQ_EN), and the interval is completed. At this point the ISPI automatically begins another interval by reloading the interval timer. The length of an interval is governed by the following equation:

Time_of_Interval = (HI_REFCLK_Period * 2 * (Interval_Count+2)) +
(HI_REFCLK_Period * Baud_Count * (Clock_Count+1))

12.2.3 Slave Mode

In slave mode, data exchanges are controlled by external devices through the pins SPI_CLK and SPI_EN. If pin SPI_EN is enabled (low), then data is latched into the shift register on every other edge of SPI_CLK; the latching edge is determined by the POL and PHA bits in the ISPI control register. Data is transferred from the shift register to the ISPI (Rx) data register when pin SPI_EN becomes inactive, or when the bit counter times out. In addition, $\overline{\text{IRQ}}$ is set at that time (if permitted by IRQ_EN). If the RX data register is not unloaded prior to a new reload, the OVR (ISPI overrun) bit is set in the ISPI status register, and the data is overwritten, causing prior received data to be lost.

SPI_CLK must not exceed HI_REFCLK/16.

12.3 Signal Descriptions

12.3.1 SPI_MISO (Master In, Slave Out)

In either master mode, this pin is the input to the shift register. A new bit is shifted in on each leading edge of SPI_CLK in normal clock mode or on each trailing edge of SPI_CLK in inverted clock mode. In slave mode, this pin is the output of the shift register. A new data bit is presented on each trailing edge of the SPI_CLK in normal clock mode (PHA=0). As a slave mode output, SPI_MISO is three-stated when the SPI_EN input is negated.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

12.3.2 SPI_MOSI (Master Out, Slave In)

In slave mode, this pin is the input to the shift register. A new bit is shifted in on each leading edge of SPI_CLK in normal clock mode or on each trailing edge of SPI_CLK in inverted clock mode. In either master mode, this pin is the output of the shift register. A new data bit is presented on each falling edge of the SPI_CLK in normal clock mode (PHA=0, POL=0).

12.3.3 SPI_EN

In manual mode, this pin is directly controlled by the SPI_EN bit in the ISPI control register (bit 12), and can be used as a general-purpose output as well as for enabling an external device. In interval mode, its control is gated by the state machine. As an input (in slave mode), SPI_EN is active low. As an output (in master modes), the active sense is determined by the value of bit SNS in the ISPI control register.

12.3.4 SPI_CLK

This pin is the clock output in manual or interval mode. When the ISPI is enabled, a selectable number of clock pulses are issued.

In slave mode, this pin is an input but controls SPI operation just as it does in the two master modes. In slave mode, SPI_CLK must not exceed HI_REFCLK/16.

12.3.5 SPI_GP

This output pin is a general-purpose output which can be used as a control signal to a selected external device. The value driven out is controlled by the SPIGP bit in the ISPI control register.

12.4 ISPI Programming Model

These registers control the operation of the ISPI and report its status. The data register exchanges data with external slave devices. After reset, all bits are cleared.

These registers should be accessed with halfword accesses. Accesses other than halfword in size result in undefined activity.

Table 12-1 ISPI Module Address Map

Address	Use	Access
10008000	ISPI Send/Receive Data Register (SPDR)	Supervisor Only
10008002	ISPI Control Register (SPCR)	Supervisor Only
10008004	ISPI Interval Control Register (SPICR)	Supervisor Only
10008006	ISPI Status Register (SPSR)	Supervisor Only
10008008 to 10008FFF	Reserved	Supervisor Only

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

12.4.1 ISPI Data Register

The ISPI data register (SPDR) contains data to be exchanged with external devices. Either writing or reading this register clears any set interrupt.

SPDR — ISPI Data Register **10008000**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Rx DATA															
W	Tx DATA															
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0+

Figure 12-3 ISPI Data Register

Rx DATA — Receive Data

This read-only register contains the data bits received from the shift register. Those bits more significant than the size determined in CLOCK COUNT (ISPI control register) return zeros when read. For example, if CLOCK COUNT = 0x8 (9-bit transfer), then bits 15 to 9 are forced to zeros. The value in this register is updated at the end of every transfer.

Tx DATA — Transmit Data

This write-only register contains the data bits to be transmitted to the external device. Data is copied from this register to the shift register at the time that the XCH bit is set.

As data is shifted MSB first, outgoing data is MSB-justified relative to the CLOCK COUNT field in the ISPI control register. For example, if the exchange length is ten bits (CLOCK COUNT = 0x9), the MSB of the outgoing data is bit nine. The first bit presented to the external device is bit 9, followed by the remaining nine less significant bits.

12.4.2 ISPI Control Register

The ISPI control register (SPCR), along with the ISPI interval control register, controls the operation of the ISPI. Follow this sequence when changing operating modes:

1. Disable the ISPI (COUNT=0) without affecting other fields.
2. Wait for any transfer to complete (XCH bit clear).
3. Update to the new mode.
4. Re-enable the ISPI (COUNT = newcount).

SPCR — ISPI Control Register **10008002**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DOZE	SPI_EN	SNS	DRV	MSTR	IRQ_EN	PHA	POL	SPIGP	BAUD RATE			CLOCK COUNT			
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12-4 ISPI Control Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

DOZE — Doze Mode

When the CPU executes a **doze** instruction and the system is placed in the doze mode, the DOZE bit affects operation of the ISPI. When this bit is set, the ISPI is disabled in doze mode. Refer to the description in **12.6 ISPI Operation in Low-Power System Modes**.

0 = ISPI unaffected in doze mode
1 = ISPI disabled in doze mode

At reset, this bit is cleared to zero.

SPI_EN — ISPI Enable

In either master mode, this bit controls the value of the SPI_EN pin. The sense of the SPI_EN pin is determined by the SNS bit. In interval mode, the SPI_EN pin is asserted only when XCH is active. The SPI_EN bit must be programmed to a one for any master mode transfer to occur. In slave mode, the ISPI state machine uses the input value on the SPI_EN pin, and this register bit is ignored. Further, the SPI_EN register bit will not reflect the value of the SPI_EN pin in slave mode.

0 = Negated
1 = Asserted

SNS — SPI_EN Sense

The SNS bit controls the sense of the SPI_EN pin relative to the SPI_EN register bit in the ISPI control register. This is required because in interval mode, the state machine must assert and then negate the SPI_EN pin. The SNS bit has an effect only when the SPI_EN pin is an output. If the SPI_EN pin is an input, then it is active low, and the SNS bit has no effect.

0 = SPI_EN pin is active low
1 = SPI_EN pin is active high

DRV — Drive Type

This bit controls the configuration of the SPI_CLK, SPI_EN and SPI_MOSI output buffers in either master mode of the ISPI (MSTR=1). In slave mode, this bit is ignored.

0 = Outputs are totem-pole in either master mode
1 = Outputs are open-drain in either master mode

MSTR — Master Mode

This bit controls the mode of the ISPI. In slave mode, the SPI_CLK and SPI_EN pins are inputs; in the master modes, they are outputs.

0 = ISPI operates in slave mode
1 = ISPI operates in either interval mode or manual mode (see IVL_EN in SICR)

IRQ_EN — Interrupt Request Enable

This bit enables/disables the ISPI interrupt request output signal. This bit is cleared to zero on reset.

0 = Interrupts disabled
1 = Interrupts enabled

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

PHA — Phase

This bit controls the phase shift of the SPI_CLK. (See **Figure 12-2**)

0 = Normal phase
1 = Shift advance to opposite phase

POL — Polarity

This bit controls the polarity of the SPI_CLK. (See **Figure 12-2**)

0 = Normal polarity
1 = Inverted polarity

SPIGP — SPI_GP Control

This bit controls the data on the SPI_GP pin.

0 = Pin driven low
1 = Pin driven high

BAUD RATE

These bits select the baud rate of the ISPI bit clock based on divisions of the system clock. The master clock for the ISPI is HI_REFCLK.

Table 12-2 BAUD RATE Field Settings

Value	Divide By
000	8
001	16
010	32
011	64
100	128
101	256
110	512
111	1024

CLOCK COUNT

These bits select the length of the transfer and control the justification of data. From two to 16 bits can be transferred. A count of all zeros causes the ISPI to be disabled.

Table 12-3 CLOCK COUNT Field Settings

Value	Meaning
0000	Disable ISPI
0001	2-bit transfer
.	.
.	.
0111	8-bit transfer
.	.
.	.
1111	16-bit transfer

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

12.4.3 ISPI Interval Control Register

The ISPI interval control register (SPICR) controls interval mode operation.

SPICR — ISPI Interval Control Register **10008004**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	LPBK	IVL_EN	INTERVAL COUNT												
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12-5 ISPI Interval Control Register

LPBK — Loopback

This bit enables a loopback test feature in the ISPI. When looping back, the ISPI operates as if the SPI_MISO and SPI_MOSI pins are wired together and there are no other external devices connected to the ISPI data input pin. Whenever loopback is enabled, the data read from the ISPI data register after a given transfer matches what was written to the ISPI data register prior to that transfer, masked if necessary to account for the number of bits transferred.

0 = Loopback disabled
1 = Loopback enabled

IVL_EN — Interval Mode Enable

Setting this bit places the ISPI in interval mode. If the MSTR bit in the ISPI control register is cleared, then the ISPI is operating in slave mode, and this bit is ignored.

0 = ISPI is not operating in interval mode
1 = ISPI is operating in interval mode if MSTR=1

INTERVAL COUNT

In interval mode, this register value is loaded into the ISPI interval timer upon completion of a transfer. Each bit-clock period, the value in this counter is decremented by one. When the value in the register reaches zero, then XCH is set, and a new transfer is begun.

12.4.4 ISPI Status Register

The ISPI status register (SPSR) contains flags indicating whether an overrun condition has occurred, whether an interrupt has been requested, and whether a transfer is being performed.

SPSR — ISPI Status Register **10008006**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OVR	IRQ	XCH	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12-6 ISPI Status Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

OVR — Overrun

This bit is set by the ISPI controller when a new value is loaded into the RX data register due to an overrun event. An overrun occurs whenever the RX data register is updated while holding previously received data that has not been read. This could occur when pin SPI_EN becomes inactive and the bit timer has already timed out, or when the bit counter times out a second time while SPI_EN remains continuously asserted. It could also be set in interval or manual master mode if the RX data register is not read between transfers. In these cases, the OVR bit may be ignored if appropriate.

0 = No overrun event has occurred
1 = An overrun event has occurred

This bit is cleared by writing it to zero or by reset.

IRQ — Interrupt Request

This register bit is cleared on either a write or a read of the ISPI data register, and when set indicates that an interrupt has been requested.

0 = No interrupt has been requested
1 = An interrupt has been requested

XCH — Exchange

This bit indicates whether the state machine is performing a transfer. In manual mode, XCH is set by writing the ISPI data register. In interval mode, XCH is set automatically by the interval timer. In slave mode, XCH is set when pin SPI_EN is asserted and is negated briefly once the counters determine the completion of a transfer. It is then reasserted if SPI_EN is still asserted. In all modes, XCH is reset upon completion of a transfer.

0 = SPI is idle or interval timer is operating
1 = Initiate exchange or exchange in progress

12.5 ISPI Programming Examples

12.5.1 Manual Mode Example

Manual mode is the simplest of the transfer methods.

Assume that the transfer to be performed is bidirectional, and the receive data is 12 bits. The data to be sent is 0x0013, and the external device will receive and keep all 12 bits. The external device is such that PHA=1, POL=0 is desired, with an active low enable. An interrupt is required following the transfer to indicate when data is available. The device accepts data at a clock rate between 100 kHz and 1 MHz, and the MMC2001 uses a 16.38-MHz clock.

To program the ISPI to perform such a transfer:

1. Write ISPI register SPCR to 0x4E4B.

2. Write ISPI register SPDR to 0x0013.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

The value to be written into the ISPI control register is determined as follows:

- (Assume DOZE = 0)
- Assign SNS = 0 to force enable to active low.
- Assign SPI_EN = 1 to enable the pin
- Assign MSTR = 1
- Assign IRQ_EN = 1 to enable interrupts
- Assign PHA = 1
- Assign POL = 0
- Assign BAUD RATE = 4 to divide 16 MHz down to approximately 128 kHz per bit
- Assign CLOCK COUNT = 0xB to transfer 12 bits.

12.5.2 Slave Mode Example

In slave mode, the timing of transfers is dependent entirely on the external device. If the transfer parameters for this example are identical to those in the manual mode example above, then the ISPI is programmed in a two-step process:

1. Write ISPI register SPCR to 0x060B.
2. Write ISPI register SPDR to 0x0013.

SPCR is programmed differently from the manual mode example because:

- SPI_EN, SNS, and BAUD RATE are ignored in slave mode.
- MSTR has to be cleared to enable slave mode.

12.5.3 Interval Model Example

With a 16.38-MHz clock, HI_REFCLK = 61 ns. To program the ISPI to transfer 10-bit words at 8-kHz intervals:

1. Program clock count to 0x9
2. Program baud rate to 0x3 (divide HI_REFCLK by 64)
3. Set SPI_EN, PHA, POL, and SNS as desired
4. Program interval count to 0x29F (671 decimal)

Per the equation:

Time_of_Interval = (HI_REFCLK_Period * 2 * (Interval_Count+2)) +
(HI_REFCLK_Period * Baud_Count * (Clock_Count +1))

Time_of_Interval is then set to the following:

61 * 2 * (671 + 2) + (61 * 64 * 11) = 125.05 µs.

The ISPI interval timer begins as soon as written, following the transfer period. This leaves approximately 1406 HI_REFCLK cycles before the Tx data register must be re-written.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

12.6 ISPI Operation in Low-Power System Modes

The following table summarizes ISPI operation in the different low-power modes.

Table 12-4 ISPI Low-Power Mode Operation

State	Operation
Normal	Runs whenever enabled
Wait	Runs whenever enabled
Doze	If DOZE is set (in SPCR), then disabled
Stop	Disabled

In most modes, the ISPI operates as long as a clock is available. In doze mode, the ISPI may be selectively disabled, depending on the value of the DOZE bit. In stop mode, the ISPI halts immediately (due to halting of system clocks) and forgets the state of any transfer in operation (the state machine is reset, and the shift register is cleared). This is done to prevent hanging a transfer in the middle; it is assumed that when stop is initiated, there is some other method of shutting down external devices. The SPDR value is retained so that the transfer can be re-initiated after the system is restarted by simply writing the SPI control register.

12.7 ISPI Operation in System Debug Mode

In debug mode, the only modification to ISPI behavior is that the clear-on-access function of the $\overline{\text{IRQ}}$ bit in the SPI status register is disabled. Normally the $\overline{\text{IRQ}}$ bit is cleared on a read or write access to the SPDR.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 13
EXTERNAL INTERRUPTS/GPIO (EDGE PORT)

13.1 Overview

The MMC2001 has eight external interrupt pins. Each pin can be configured individually as a level-sensitive interrupt, an edge-detecting interrupt (rising edge, falling edge, or both), or a general-purpose I/O pin.

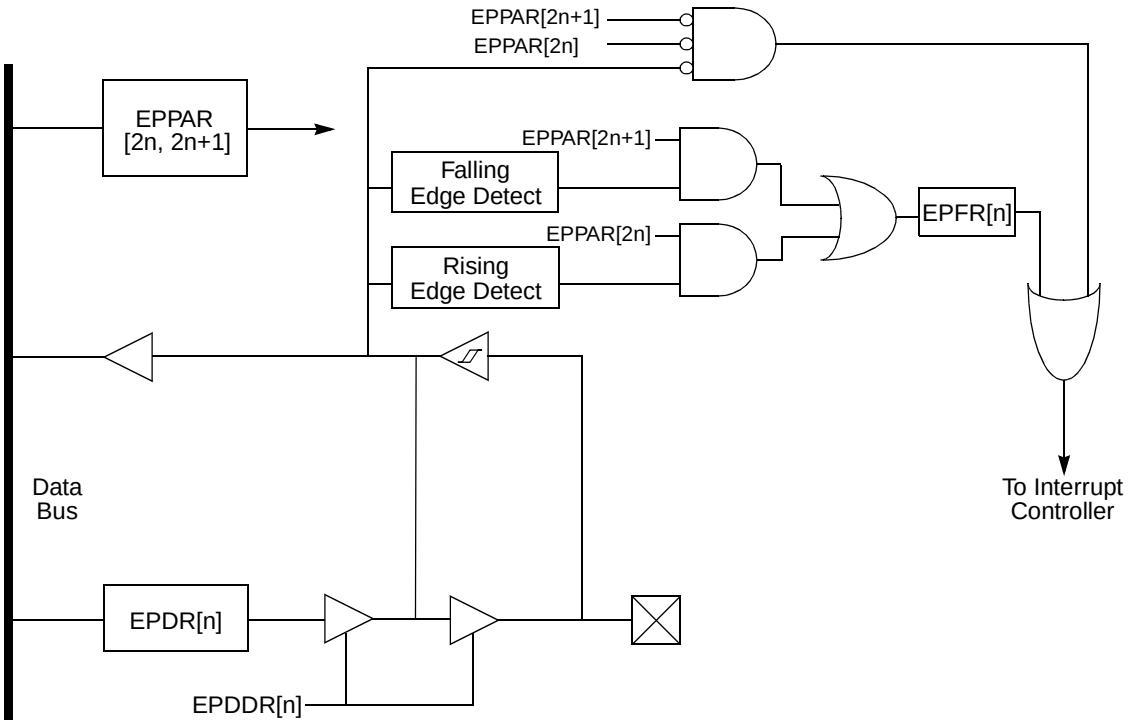


Figure 13-1 External Interrupt/GPIO Block Diagram

13.2 Interrupt/General-Purpose I/O Pin Descriptions (INT[0:7])

When programmed as inputs, these pins use Schmitt triggered input buffers. When edge triggered, triggering occurs at a voltage level and is not directly related to the fall time of the interrupt signal. However, as the fall time of the interrupt signal increases, the probability of generating multiple interrupts due to this “noise” also increases. All default to general-purpose input pins at reset. The interrupt request function on these pins is masked in the interrupt controller fast interrupt enable register (FIER) and normal interrupt enable register (NIER).

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

13.3 Edge Port Programming Model

The edge port programming model consists of the following registers:

- The edge port pin assignment register (EPPAR) controls the function of each pin individually.
- The edge port data direction register (EPDDR) controls the direction of each one of the pins individually.
- The edge port data register (EPDR) holds the data to be driven to the pins.
- The edge port flag register (EPFR) latches the edge event for each one of the pins individually.

Access the edge port registers with halfword accesses.

Table 13-1 GPIO Edge Port Address Map

Address	Use	Access
10007000	Edge Port Pin Assignment Register (EPPAR)	Supervisor Only
10007002	Edge Port Data Direction Register (EPDDR)	Supervisor Only
10007004	Edge Port Data Register (EPDR)	Supervisor Only
10007006	Edge Port Flag Register (EPFR)	Supervisor Only

13.3.1 Edge Port Pin Assignment Register (EPPAR)

The 16-bit read/write edge port pin assignment register (EPPAR) configures each of the interrupt pins as either level-sensitive or edge-triggered. Rising, falling, or both edges can be selected as the active edge. Requests are always generated out of this block but may be masked within the interrupt controller module. The functionality of this register is independent of the programmed pin direction.

EPPAR — Edge Port Pin Assignment Register **10007000**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EPPA7		EPPA6		EPPA5		EPPA4		EPPA3		EPPA2		EPPA1		EPPA0	
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 13-2 Edge Port Pin Assignment Register

EPPAx — Edge Port Pin Assignment Select Field x

Pins configured as level-sensitive are inverted so that a logic low on the external pin represents a valid interrupt request. Level-sensitive interrupt inputs are not latched. To guarantee that a level-sensitive interrupt request is acknowledged, the interrupt source must keep the signal asserted until acknowledged by software.

Pins configured as edge-sensitive interrupts are latched and need not remain asserted for interrupt generation. When the pin is programmed to use the edge detecting circuit, its state is monitored regardless of its configuration as input or output.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

These bits are cleared by hardware reset.

Table 13-2 EPPAx Field Settings

Value	Meaning
00	Pin INTx defined as level sensitive
01	Pin INTx defined as rising edge detect
10	Pin INTx defined as falling edge detect
11	Pin INTx defined as both falling and rising edge detect

13.3.2 Edge Port Data Direction Register (EPDDR)

The 16-bit read/write edge port data direction register (EPDDR) controls the direction of the port pins. Setting any bit in this register configures the corresponding pin as an output. Clearing any bit in this register configures the corresponding pin as an input. Pin direction is independent of the level/edge mode programmed.

EPDDR — Edge Port Data Direction Register **10007002**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	EPDD7	EPDD6	EPDD5	EPDD4	EPDD3	EPDD2	EPDD1	EPDD0
W																
RESET:									0	0	0	0	0	0	0	0

Figure 13-3 Edge Port Data Direction Register

EPDDx — Edge Port Data Direction x

0 = Pin INT x is an input.
1 = Pin INTx is an output.

These bits are cleared by reset.

13.3.3 Edge Port Data Register (EPDR)

The edge port data register (EPDR) is a 16-bit register. Writes to EPDR are stored in an internal latch, and if any pin of the port is configured as an output, the data stored for that bit is driven onto the pin. Reads of this register return the value sensed on the pins for those pins configured as inputs, or the data stored in the register for the pins configured as outputs.

EPDR — Edge Port Data Register **10007004**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	EPD7	EPD6	EPD5	EPD4	EPD3	EPD2	EPD1	EPD0
W																
RESET:									X	X	X	X	X	X	X	X

X = Unaffected by reset

Figure 13-4 Edge Port Data Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

EPDx — Edge Port Data x

See the description above. These bits are not affected by hardware reset.

13.3.4 Edge Port Flag Register (EPFR)

The 16-bit read/write edge port flag register (EPFR) indicates whether the selected edge has been detected on the port pins.

EPFR — Edge Port Flag Register **10007006**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	EPF7	EPF6	EPF5	EPF4	EPF3	EPF2	EPF1	EPF0
W																
RESET:									0	0	0	0	0	0	0	0

Figure 13-5 Edge Port Flag Register

EPFx — Edge Port Flag x

0 = Selected edge for INTx pin has not been detected.
1 = Selected edge for INTx pin has been detected.

Bits in this register are set when the programmed edge is detected on the corresponding pin. A bit remains set until cleared by writing it to a one. Pin transitions do not affect this register if the pin is configured as level sensitive (EPPARx=00). The corresponding flag bit(s) are cleared to zero in this case. When a pin is configured as a general-purpose output, writes to EPDR that cause the selected level or edge interrupt will set the corresponding bit in EPFR. The outputs of this register drive the corresponding input of the interrupt controller for those bits configured as edge detecting. These bits are cleared by hardware reset.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 14
KEYPAD PORT

14.1 Overview

The keypad port (KPP) is a 16-bit peripheral which can be used either for keypad matrix scanning or as general-purpose I/O. The block diagram of the KPP is shown in **Figure 14-1**.

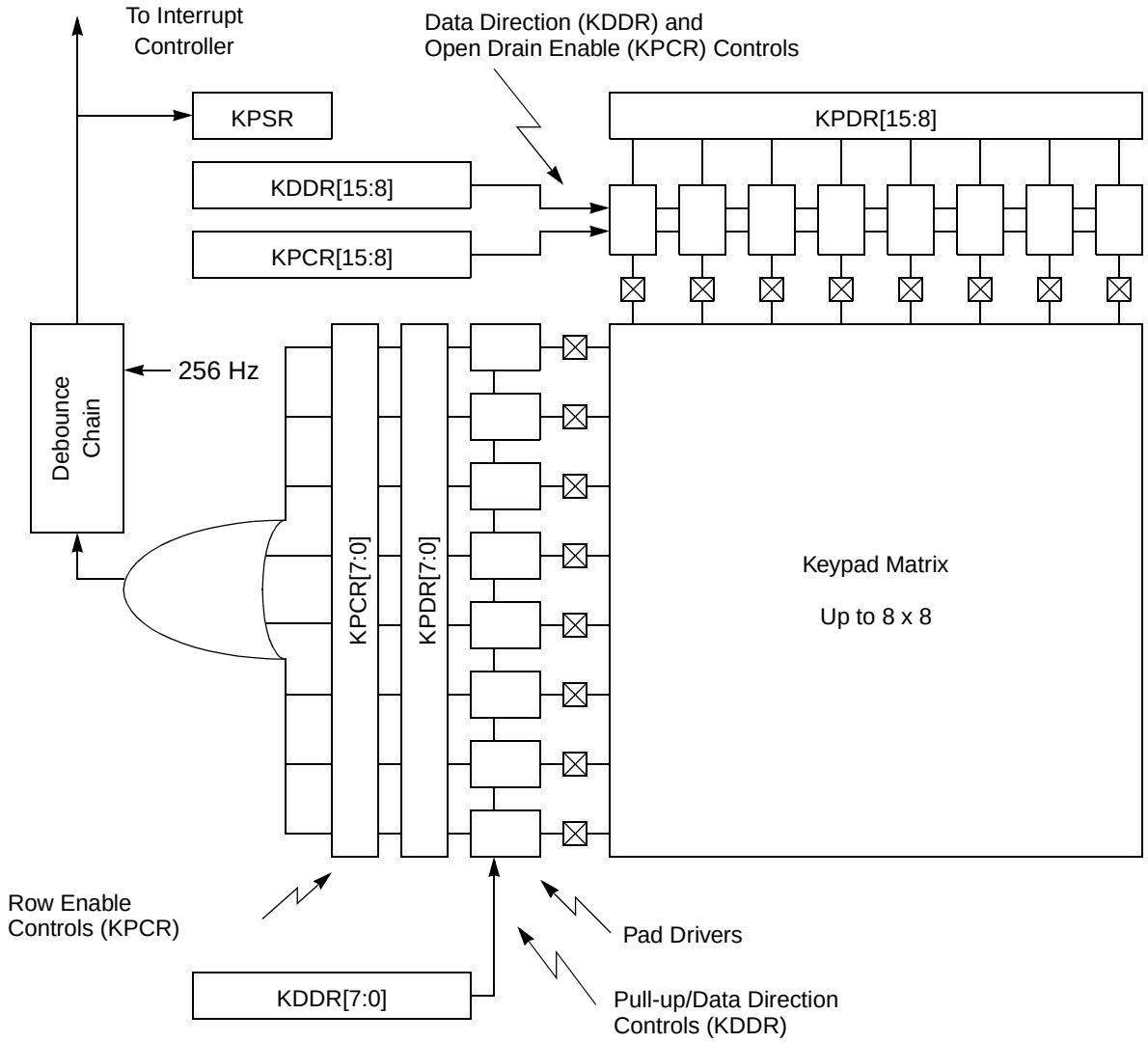


Figure 14-1 KPP Block Diagram

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

14.2 KPP Pin Description

Sixteen pins are dedicated to the KPP. Keypads of any configuration up to eight rows and eight columns are supported through software configuration of the peripheral pins. Any pins not used for the keypad are available for general-purpose input/output. The registers are configured such that the pins can be treated as an I/O port up to 16 bits wide.

14.2.1 Input Pins

Any of the 16 pins associated with the KPP can be configured as inputs by writing zeros to the appropriate bits in the KDDR. Additionally, the least significant eight bits (ROW inputs) corresponding to KDDR[7:0] have internal pull-ups that are enabled when the pin is used as an input.

14.2.2 Output Pins

Any of the 16 pins associated with the KPP can be configured as outputs by writing the appropriate bits in the KDDR to one. Additionally, the pins representing the eight most significant bits (KDDR[15:8]) can be designated as open drain outputs by writing a one into the appropriate bits in KPCR. The pins representing the lower eight bits (KDDR[7:0]) are always totem-pole style drive when configured as outputs.

Table 14-1 Keypad Port Column Modes

KDDR[15:8]	KPCR[15:8]	Pin Function
0	x	Input
1	0	Totem-Pole Output
1	1	Open-Drain Output

14.3 KPP Programming Model

Table 14-2 Keypad Port Address Map

Address	Use	Access
10003000	Keypad Control Register (KPCR)	Supervisor Only
10003002	Keypad Status Register (KPSR)	Supervisor Only
10003004	Keypad Data Direction Register (KDDR)	Supervisor Only
10003006	Keypad Data Register (KPDR)	Supervisor Only
10003008 to 10003FFF	Reserved	Supervisor Only

14.3.1 Keypad Control Register (KPCR)

The keypad control register (KPCR) determines which of the eight possible column strobes are to be open drain when configured as outputs and which of the eight row sense lines are considered in generating an interrupt to the CPU.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Setting a column open-drain enable bit (KCO[7:0]) disables the pull-up driver on that pin. Clearing the bit allows the pin to drive to the high state. This bit has no effect when the pin is configured as an input.

Setting a row enable control bit in this register enables the corresponding row line to participate in interrupt generation. Likewise, clearing a bit disables that row from being used to generate an interrupt. This register is cleared by reset, disabling all rows. The row enable logic is independent of the programmed direction of the pin. Writing a zero to the data register of pins configured as outputs causes a keypad interrupt to be generated if the row enable associated with that bit is set. It is up to the programmer to ensure that pins being used for functions other than the keypad are properly disabled.

The KPCR register is byte or halfword addressable.

KPCR — Keypad Control Register **10003000**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KCO7	KCO6	KCO5	KCO4	KCO3	KCO2	KCO1	KCO0	KRE7	KRE6	KRE5	KRE4	KRE3	KRE2	KRE1	KRE0
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-2 Keypad Control Register

KCOx — Keypad Column Strobe Open-Drain Enable x

0 = Column strobe output x is totem-pole drive (P-channel enabled).
1 = Column strobe output x is open drain (P-channel disabled).

KREx — Keypad Row Enable x

0 = Row x is not included in keypad key press detect.
1 = Row x is included in keypad key press detect.

14.3.2 Keypad Status Register (KPSR)

The keypad status register (KPSR) reflects the state of the keypress detect circuit.

The keypad key depress (KPKD) status bit is set when one or more enabled rows are detected low after synchronization. The KPKD status bit remains set until cleared by software. The KPKD bit may be used to generate a maskable key depress interrupt. If desired, software may clear the keypress synchronizer chain to allow a repeated interrupt to be generated while a key remains pressed. In this case, a new interrupt will be generated after the synchronizer delay elapses if a key remains pressed.

The keypad key release (KPKR) status bit is set when all enabled rows are detected high after synchronization. The KPKR status bit remains set until cleared by software. The bit will typically not be set again until the detect circuit senses a key depressed followed by all keys released. The KPKR bit may be used to generate a maskable key release interrupt. The key release synchronizer may be set high by software after scanning the keypad to ensure a known state.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Due to the logic function of the release and depress synchronizer chains, it is possible to see the re-assertion of a status flag (KPKD or KPKR) if it is cleared by software prior to the system exiting the state it represents. Software should ensure that the interrupt for a key release event is masked until it has entered the key pressed state (and vice versa) unless this activity is desired, as might be the case when a repeated interrupt is to be generated. The synchronizer chains are capable of being initialized to detect repeated key presses or releases. If they are not initialized when the corresponding event flag is cleared, false interrupts may be generated for depress (or release) events shorter than the length of the corresponding chain.

The KPSR register is byte or halfword addressable.

KPSR — Keypad Status Register **10003002**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	KRIE	KDIE	0	0	0	0	0	0	KPKR	KPKD
W													KRSS	KDSC		
RESET:							0	0					0	0	0	0

Figure 14-3 Keypad Status Register

KRIE — Key Release Interrupt Enable

0 = No interrupt request is generated when KPKR is set
1 = An interrupt request is generated when KPKR is set

KDIE — Key Depress Interrupt Enable

0 = No interrupt request is generated when KPKD is set
1 = An interrupt request is generated when KPKD is set

KRSS — Key Release Synchronizer Set

The key release synchronizer is set by writing a logic one into this bit. Reads return a value of zero.

KDSC — Key Depress Synchronizer Clear

The key depress synchronizer is cleared by writing a logic one into this bit. Reads return a value of zero.

KPKR — Keypad Key Release

0 = No key release detected
1 = All keys have been released

KPKR is cleared by writing a logic one into this bit.

KPKD — Keypad Key Depress

0 = No key presses detected
1 = A key has been depressed

KPKD is cleared by writing a logic one into this bit.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

14.3.3 Keypad Data Direction Register (KDDR)

The bits in the keypad data direction register (KDDR) control the direction of the keypad port pins. The upper eight bits in the register affect the pins designated as column strobes, while the lower eight bits affect the row sense pins. Setting any bit in this register configures the corresponding pin as an output. Clearing any bit in this register configures the corresponding port pin as an input. For bits 7 to 0, an internal pull-up is enabled if the corresponding bit is cleared. This register is cleared by reset, configuring all pins as inputs.

The KDDR register is byte or halfword addressable.

KDDR — Keypad Data Direction Register **10003004**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KCDD7	KCDD6	KCDD5	KCDD4	KCDD3	KCDD2	KCDD1	KCDD0	KRDD7	KRDD6	KRDD5	KRDD4	KRDD3	KRDD2	KRDD1	KRDD0
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-4 Keypad Data Direction Register

KCDDx — Keypad Column x Data Direction
- 0 = COLx pin is configured as input.
- 1 = COLx pin is configured as output.

KRDDx — Keypad Row x Data Direction
- 0 = ROWx pin is configured as input.
- 1 = ROWx pin is configured as output.

14.3.4 Keypad Data Register (KPDR)

The 16-bit keypad data register is used to access the column and row data. Data written to this register is stored in an internal latch, and for each pin configured as an output, the stored data is driven onto the pin. A read of this register returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register.

The KPDR register is byte or halfword addressable.

KPDR — Keypad Data Register **10003006**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KCD7	KCD6	KCD5	KCD4	KCD3	KCD2	KCD1	KCD0	KRD7	KRD6	KRD5	KRD4	KRD3	KRD2	KRD1	KRD0
RESET:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

* Since pins default to inputs, reset value is determined by the logic level present on the pins at reset.

Figure 14-5 Keypad Data Register

**For More Information On This Product,
Go to: www.freescale.com**

Freescale Semiconductor, Inc.

KCDx — Keypad Column x Data Bit

KRDx — Keypad Row x Data Bit

This register is not initialized by reset. Valid data should be written to this register before any bits are configured as outputs.

14.4 Keypad Operation

The keypad port is designed to simplify the software task of scanning a keypad matrix. With appropriate software support, the KPP is capable of detecting, debouncing, and decoding one or two keys pressed simultaneously in the keypad.

Logic in the KPP can detect a key press even while the processor is in one of the low-power standby modes. The KPP may generate a CPU interrupt any time a key press or key release is detected. This interrupt can force the processor out of a low-power mode.

14.4.1 Keypad Matrix Construction

The keypad port is designed to interface to a keypad matrix which shorts the intersecting row and column lines together whenever a key is depressed. The interface is not optimized for other switch configurations.

14.4.2 Keypad Port Configuration

Software must initialize the keypad port for the size of the keypad matrix. Pins connected to the keypad columns should be configured as open-drain outputs. Pins connected to the keypad rows should be configured as inputs. On-chip pull-up resistors are implemented for active keypad rows, as defined in **SECTION 4 SIGNAL DESCRIPTIONS**.

Row inputs must also be enabled in the keypad control register to be active in the interrupt generation circuit.

Discrete switches that are not part of the matrix may be connected to any unused row inputs. The second terminal of the discrete switch is connected to ground. The hardware detects closures of these switches without the need for software polling.

14.4.3 Keypad Matrix Scanning

Keypad scanning is performed by a software loop that walks a zero across each of the keypad columns, reading the value on the rows at each step. The process is repeated several times in succession, with the results of each pass optionally compared with those from the previous pass. When three or four consecutive scans yield the same key closures, a valid key press has been detected. Software can then decode exactly which switch was depressed and pass the value up to the next higher software layer.

The basic debouncing period to be defined in the software routine can be controlled with an internal timer. The basic period is the period between the scan of two consecutive columns, so the debounce time between two consecutive scans of the whole matrix equals the number of columns multiplied by the basic period.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

14.4.4 Keypad Standby

There is no need for the CPU to scan the keypad continually. Between key presses, the keypad can be left in a state that requires no software intervention until the next key press is detected. To place the keypad in a standby state, software writes all column outputs low. Row inputs are left enabled. At this point the CPU can attend to other tasks or revert to a low-power standby mode. The keypad port will interrupt the CPU if any key is pressed.

Upon receiving a keypad interrupt, the CPU should set all the column strobes high and begin a normal keypad scanning routine to determine which key was pressed. It is important that open-drain drivers be used when scanning to prevent a possible DC path between power and ground through two or more switches.

14.4.5 Glitch Suppression on Keypad Inputs

A glitch suppression circuit qualifies the keypad inputs to prevent noise from inadvertently interrupting the CPU. The circuit is a four-state synchronizer clocked from a 256-Hz clock source. This clock must continue to run in any low-power mode for which the keypad is a wake-up source, as the CPU interrupt is generated from the synchronized input. An interrupt is not generated until all four synchronizer stages have latched a valid key assertion, effectively filtering out any noise less than 16 ms in duration. The interrupt output is latched in an S-R latch and remains asserted until cleared by software. The set input of the latch is clocked on the rising edge.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

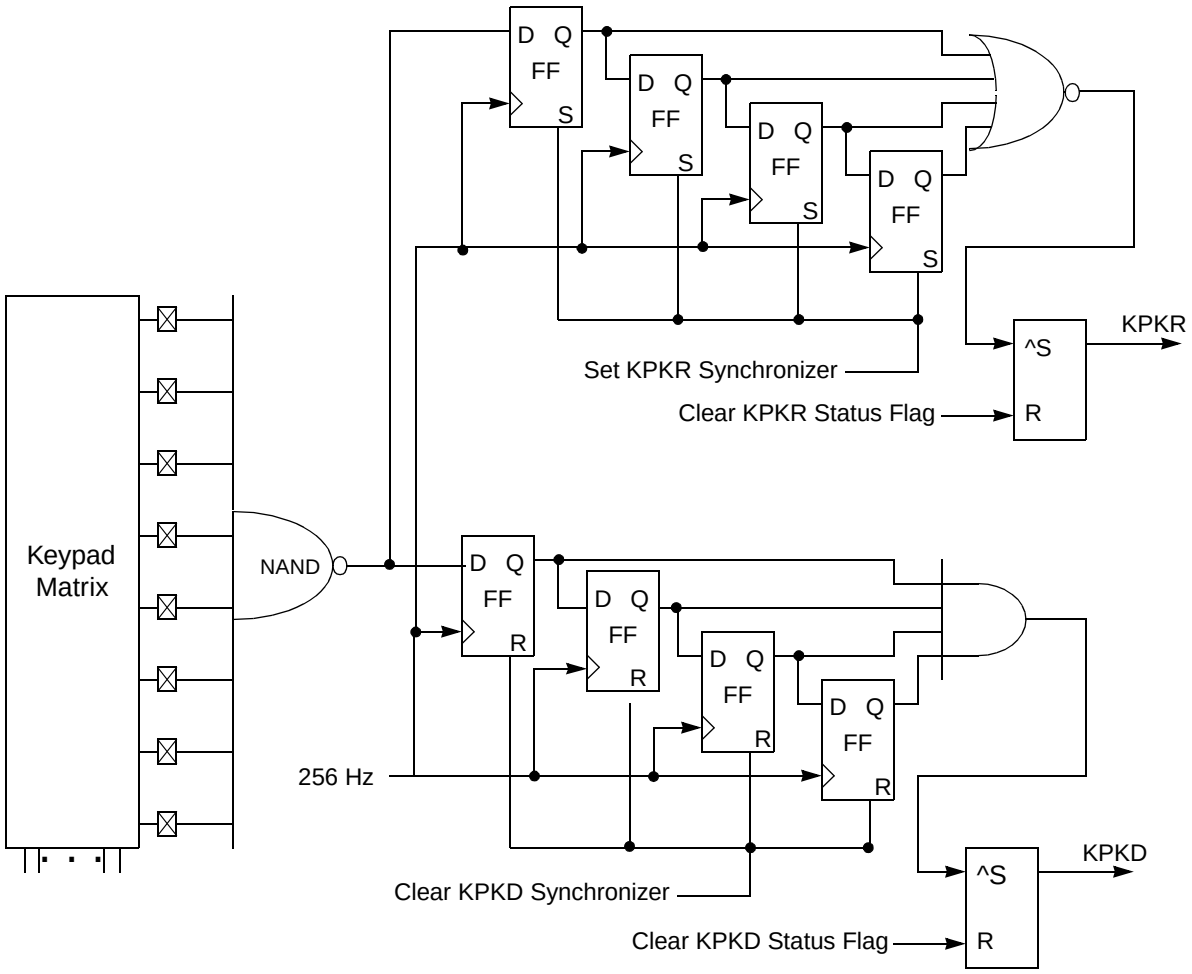


Figure 14-6 Keypad Synchronizer Functional Diagram

14.4.6 Multiple Key Closures

One or two keys pressed simultaneously are easily detected by the software. When three or more keys are pressed, however, it is possible that errant key closures may be detected. As can be seen in **Figure 14-7**, three keys pressed simultaneously can short between the column currently “scanned” by software and another column. Depending on the location of the third key pressed, a “ghost” key press may be detected.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

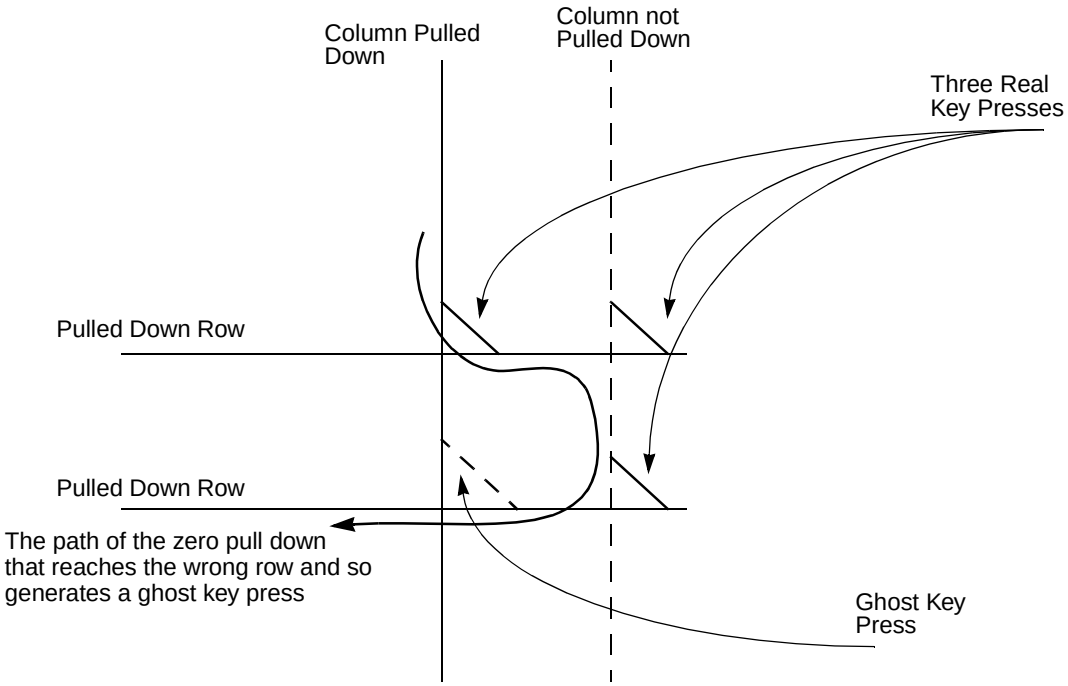


Figure 14-7 Decoding Wrong Three Key Presses

14.4.7 Typical Keypad Configuration and Scanning Sequence

1. Configure keypad
   a. Enable number of rows in keypad (KPCR[7:0]).
   b. Write zeros to KPDR[15:8].
   c. Configure keypad columns as open-drain (KPCR[15:8]).
   d. Configure columns as output, rows as input (KDDR[15:0]).
   e. Clear the KPKD status flag and synchronizer chain.
   f. Set the KDIE control bit, and clear the KRIE control bit (avoid false release events).

      (now in standby mode, awaiting a keypress...)

2. Keypress interrupt detected
3. Begin keypad scanning routine
   a. Disable keypad interrupts.
   b. Write ones to KPDR[15:8], setting column data to ones.
   c. Configure columns as totem-pole outputs.
   d. Configure columns as open-drain.
   e. Write a single column to zero, and write others to one.
   f. Sample row inputs and save data. Multiple key presses can be detected on a single column.
   g. Repeat steps b to f for remaining columns.
   h. Return all columns to zero in preparation for standby mode.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

i. Clear interrupt status bit(s) by writing to a one;
set the KPKR synchronizer chain,
clear the KPKD synchronizer chain.

j. Re-enable the appropriate keypad interrupt(s);
KDIE to detect a key hold condition, or
KRIE to detect a key release event.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 15
PULSE WIDTH MODULATOR

The pulse width modulator (PWM) module contains six identical channels, PWM5 – PWM0.

15.1 Overview

Each PWM channel consists of a simple free-running counter with two compare registers. Each compare register performs a particular task when it matches the count value. The period comparator causes the output pin to be set and the free-running counter to reset when its value matches the period value. The width comparator causes the output pin to reset when the counter value matches. With a suitable low-pass filter, the PWM channel can be used as a digital-to-analog converter.

Figure 15-1 is a block diagram of a single PWM channel.

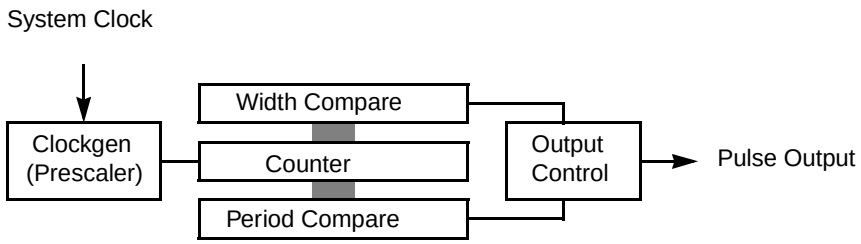


Figure 15-1 PWM Block Diagram

By feeding a stream of sample values to the PWM into the width register and providing a low-pass filter on the output, the output pin can provide a digitally-generated sound source. The reconstruction rate is determined by the selected period. Typically, for voice quality, the rate is between 6 kHz and 8 kHz. **Figure 15-2** relates the pulse stream to the filtered audio output.

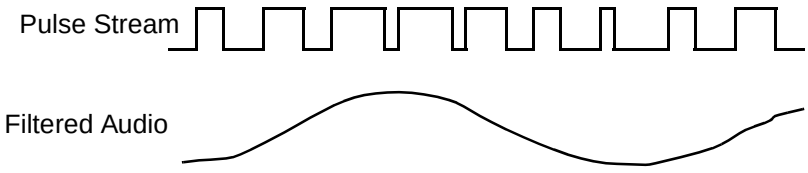


Figure 15-2 PWM Generating Audio

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

The width and period registers are double-buffered so that a new value can be loaded for the next cycle without disturbing the current cycle. At the beginning of each period, the contents of the buffer registers are loaded into the comparator for the next cycle. Sampled audio can be recreated by feeding a new sample value into the width register on each interrupt.

A single shared prescaler provides operating flexibility. **Figure 15-3** describes its functionality. The prescaler contains a variable divider that can divide the incoming clock by certain values between four and 65536.

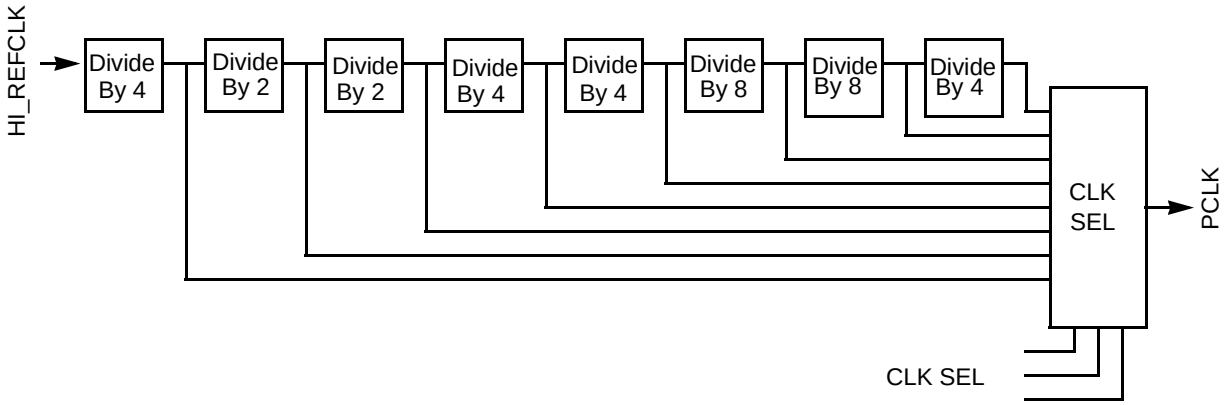


Figure 15-3 PWM Prescaler

Each PWM channel can independently select a prescaler tap point. In addition, each channel provides a maskable interrupt request that can be asserted after each period compare event.

Channels can be used as periodic interrupt sources. In this case, the output pin associated with a channel can be used as a general-purpose I/O pin operating independently of the timing function.

15.2 PWM Programming Model

This section describes the registers and control bits in the PWM module. All registers reset to 0x0000 after reset.

These registers must be accessed with halfword accesses. Accesses other than halfword in size result in undefined activity.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 15-1 PWM Address Map

Address	Use	Access
10005000	PWM0 Control Register (PWMCR0)	Supervisor Only
10005002	PWM0 Period Register (PWMPR0)	Supervisor Only
10005004	PWM0 Width Register (PWMWR0)	Supervisor Only
10005006	PWM0 Counter Register (PWMCTR0)	Supervisor Only
10005008	PWM1 Control Register (PWMCR1)	Supervisor Only
1000500A	PWM1 Period Register (PWMPR1)	Supervisor Only
1000500C	PWM1 Width Register (PWMWR1)	Supervisor Only
1000500E	PWM1 Counter Register (PWMCTR1)	Supervisor Only
10005010	PWM2 Control Register (PWMCR2)	Supervisor Only
10005012	PWM2 Period Register (PWMPR2)	Supervisor Only
10005014	PWM2 Width Register (PWMWR2)	Supervisor Only
10005016	PWM2 Counter Register (PWMCTR2)	Supervisor Only
10005018	PWM3 Control Register (PWMCR3)	Supervisor Only
1000501A	PWM3 Period Register (PWMPR3)	Supervisor Only
1000501C	PWM3 Width Register (PWMWR3)	Supervisor Only
1000501E	PWM3 Counter Register (PWMCTR3)	Supervisor Only
10005020	PWM4 Control Register (PWMCR4)	Supervisor Only
10005022	PWM4 Period Register (PWMPR4)	Supervisor Only
10005024	PWM4 Width Register (PWMWR4)	Supervisor Only
10005026	PWM4 Counter Register (PWMCTR4)	Supervisor Only
10005028	PWM5 Control Register (PWMCR5)	Supervisor Only
1000502A	PWM5 Period Register (PWMPR5)	Supervisor Only
1000502C	PWM5 Width Register (PWMWR5)	Supervisor Only
1000502E	PWM5 Counter Register (PWMCTR5)	Supervisor Only
10005030 to 10005FFF	Reserved	Supervisor Only
10006000 to 10006FFF	Not Used (Access causes transfer error)	Not Applicable

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

15.2.1 PWM Control Register

The PWM control register (PWMCR) controls the overall operation of the PWM channel. The status of the channel pin is also accessible.

PWMCR0 — PWM0 Control Register **10005000**
PWMCR1 — PWM1 Control Register **10005008**
PWMCR2 — PWM2 Control Register **10005010**
PWMCR3 — PWM3 Control Register **10005018**
PWMCR4 — PWM4 Control Register **10005020**
PWMCR5 — PWM5 Control Register **10005028**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	DOZE	PWM IRQ	IRQ EN	LOAD	DATA	DIR	POL	MODE	COUNT EN	CLKSEL		
W																
RESET:					0	0	0	0		0	0	0	0	0	0	0

Figure 15-4 PWM Control Registers

DOZE — Doze Mode

When the CPU executes a **doze** instruction and the system is placed in doze mode, the DOZE bit affects operation of the PWM channel. If this bit is set, the PWM channel is disabled in doze mode. PWM channel operation is suspended at the end of the current period. If IRQ_EN is set, an interrupt request is still generated following the period compare that causes suspension. This interrupt can selectively cause the CPU to exit doze mode.

0 = PWM channel is unaffected in doze mode
1 = PWM channel is disabled in doze mode

At reset, this bit is cleared to zero.

PWM IRQ — PWM Interrupt Request

This bit indicates that an interrupt was posted by a period compare. This bit can be set by the user to post a PWM interrupt immediately for debugging purposes. This bit is cleared automatically after it is read while set. If IRQ EN is cleared, this bit will not be set.

0 = No interrupt posted
1 = PWM period rolled over

IRQ EN — Interrupt Request Enable

This bit controls PWM interrupt generation. While this bit is low, the interrupt is disabled.

0 = PWM interrupt disabled
1 = PWM interrupt enabled

LOAD — Load PWMPR and PWMWR

Setting this bit forces a new period. The period and width registers are loaded into the comparator latches and the counter is reset. This bit is cleared automatically after the load has been performed. The actual load occurs some time after the CPU writes this

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

bit, as the load occurs on the next rising PCLK edge following internal synchronization. Forcing a load of the comparator latches and counter in this manner must be done with caution to avoid unexpected pin behavior.

DATA — PWM Data

This bit indicates or controls the current state of the PWM pin. When the pin is configured as a general-purpose output, the logical value written to this bit is used to drive the pin. When the pin is configured as a general-purpose input, the pin value is reflected by this bit. When the pin is configured in PWM mode, the bit reflects the value being driven on the pin by the PWM logic.

DIR — Direction

This bit controls the direction of the pin when used as a GPIO pin. This bit has no effect when MODE indicates PWM mode.

0 = Pin is an input pin
1 = Pin is an output pin

POL — Polarity

This bit controls the polarity of the pin when used as a PWM output pin. Normally, the output pin is set high at period boundaries and goes low when a width compare event occurs.

This bit is ignored if the pin is being used as a GPIO pin.

0 = Normal PWM polarity
1 = Inverted PWM polarity

MODE — PWM Mode

This bit selects whether the PWM pin is used for GPIO or for the PWM function.

0 = General-purpose I/O mode
1 = PWM mode

COUNT EN — Counter Enable

This bit enables or disables the PWM counter. The counter is actually enabled or disabled some time after the CPU writes this bit. It is enabled on the next rising PCLK edge following internal synchronization. If running, the counter is disabled following the next period match.

0 = PWM disabled. While disabled, the counter is in a low-power mode and does not count. The following events occur:

When the output pin is configured to operate in PWM mode (MODE = 1), the output pin is forced to the setting of the POL bit.

The counter is reset to 00 and frozen.

The contents of the width and period registers are loaded into the comparators.

The comparators are disabled.

If the counter has been running, and the actual disable occurs at the occurrence of a period match, an interrupt request may still be generated, even though the counter is being disabled. To prevent this, write the interrupt enable control bit (IRQ_EN) to zero when disabling the counter.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

1 = PWM is enabled and begins a new period. The following events occur:

The output pin changes state to start a new period (if width != 0 and period != 0 and width < period)

The counter is released and begins counting

The comparators are enabled

The PWM IRQ bit is set, indicating the start of a new period if IRQ EN is set.

CLK SEL — Clock Select

These bits select the output of the divider chain.

Table 15-2 CLK SEL Field Settings

Value	Divide By
000	4
001	8
010	16
011	64
100	256
101	2048
110	16384
111	65536

15.2.2 PWM Period Register

The PWM period register (PWMPR) controls the period of the PWM by defining the number of PCLKs in the period. When the counter value matches the value in this register, an interrupt is posted and the counter is reset to start another period.

PWMPR0 — PWM0 Period Register **10005002**
PWMPR1 — PWM1 Period Register **1000500A**
PWMPR2 — PWM2 Period Register **10005012**
PWMPR3 — PWM3 Period Register **1000501A**
PWMPR4 — PWM4 Period Register **10005022**
PWMPR5 — PWM5 Period Register **1000502A**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PERIOD									
W																
RESET:							0	0	0	0	0	0	0	0	0	0

Figure 15-5 PWM Period Registers

PERIOD — Pulse Period

This is the value that causes the counter to be reset. There is one special case. When PERIOD = 0, the output is never set high (0% duty cycle). In this case, the comparator is loaded and the counter reset on every PCLK. In addition, if enabled, an interrupt request is generated on every PCLK.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

15.2.3 PWM Width Register

The PWM width register (PWMWR) defines the width of the pulse in PCLKs. When the counter matches the value in this register, the output is reset for the duration of the period. Note that if the value in this register is not less than the period register, the output will never be reset, resulting in a 100% duty cycle.

PWMWR0 — PWM0 Width Register **10005004**
PWMWR1 — PWM1 Width Register **1000500C**
PWMWR2 — PWM2 Width Register **10005014**
PWMWR3 — PWM3 Width Register **1000501C**
PWMWR4 — PWM4 Width Register **10005024**
PWMWR5 — PWM5 Width Register **1000502C**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	WIDTH									
W																
RESET:							0	0	0	0	0	0	0	0	0	0

Figure 15-6 PWM Width Registers

WIDTH — Pulse Width

When the counter reaches the value in this register, the output is reset.

15.2.4 PWM Counter Register

The read-only PWM counter register (PWMCTR) holds the current count value. It can be read at any time without disturbing the counter.

PWMCTR0 — PWM0 Counter Register **10005006**
PWMCTR1 — PWM1 Counter Register **1000500E**
PWMCTR2 — PWM2 Counter Register **10005016**
PWMCTR3 — PWM3 Counter Register **1000501E**
PWMCTR4 — PWM4 Counter Register **10005026**
PWMCTR5 — PWM5 Counter Register **1000502E**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	COUNT									
W																
RESET:							0	0	0	0	0	0	0	0	0	0

Figure 15-7 PWM Count Registers

COUNT — Count Value

This is the current count value.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

15.3 PWM Operating Range

Table 15-3 shows the operating range and resolution of the PWM with a 16-MHz HI_REFCLK. The minimum period range assumes a value of two in the period register, while the maximum assumes a value of 256.

Table 15-3 PWM Range at 16 MHz

Divide By	Approximate Period Range at 16 MHz	Resolution at 16 MHz
4	0.5 µs – 256 µs	.25 µs
8	1 µs – 512 µs	.5 µs
16	2 µs – 1 ms	1 µs
64	8 µs – 4.1 ms	4 µs
256	32 µs – 16.5 ms	16 µs
2048	256 µs – 65.6 ms	128 µs
16384	2 ms – 1 s	1 ms
65536	8.2 ms – 4 s	4.1 ms

15.4 PWM Operation in Low-Power System Modes

Table 15-4 summarizes PWM operation in the different low-power modes.

Table 15-4 PWM Low-Power Mode Operation

Mode	Operation
Normal	Runs whenever enabled
Wait	Runs whenever enabled
Doze	If DOZE is set (in PWM control register), then disabled.
Stop	Disabled

In most modes, the PWM operates as long as a clock is available. In doze mode, the PWM channels may be selectively disabled, depending on the value of the DOZE control bit. PWM channel operation will be suspended at the end of the current period. In stop mode, the PWM halts immediately (due to halting of system clocks) and forgets the state of any period (the state machine is reset, and the shift register is cleared). It is assumed that when stop is initiated, the channels have been disabled.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SECTION 16
OnCE™ DEBUG MODULE

16.1 Overview

The on-chip emulation (OnCE™) circuitry provides a simple, inexpensive debugging interface that allows external access to the processor's internal registers and to memory/peripherals. OnCE capabilities are controlled through a serial interface, mapped onto a JTAG test access port (TAP) protocol. **Figure 16-1** shows the components of the OnCE circuitry.

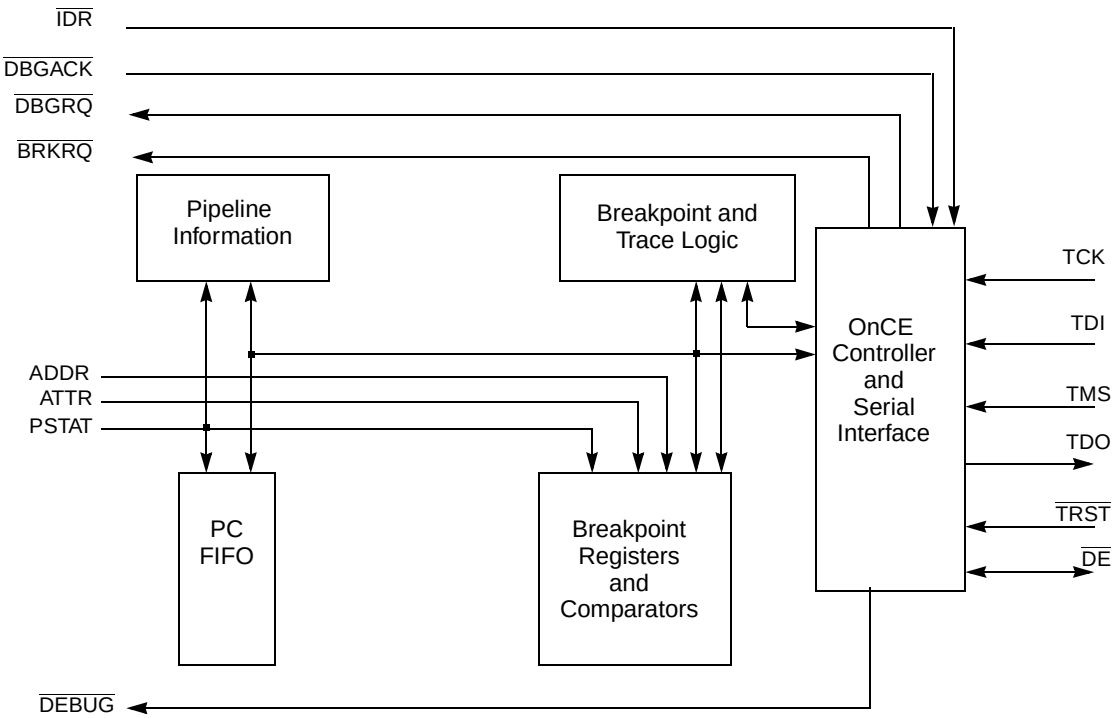


Figure 16-1 OnCE Block Diagram

The interface to the OnCE controller and its resources is based on the TAP defined for JTAG in the IEEE-1149.1a-1993 standard.

16.2 Operation

An instruction is scanned into the OnCE module through the serial interface and then decoded. Data may then be scanned in and used to update a register or resource on a write to the resource, or data associated with a resource may be scanned out for a read of the resource.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

For accesses to the CPU internal state, the OnCE controller requests the CPU to enter debug mode via the CPU $\overline{\text{DBGRQ}}$ input. Once CPU debug mode has been entered, as indicated by the OnCE status register, the processor state may be accessed through the CPU scan register.

The OnCE controller is implemented as a 16-state FSM, with a one-to-one correspondence to the states defined for the JTAG TAP controller.

CPU registers and the contents of memory locations are accessed by scanning instructions and data into and out of the CPU scan chain. Required data is accessed by executing the scanned instructions. Memory locations may be read by scanning in a load instruction to the CPU that references the desired memory location, executing the load instruction, and then scanning out the result of the load. Other resources are accessed in a similar manner.

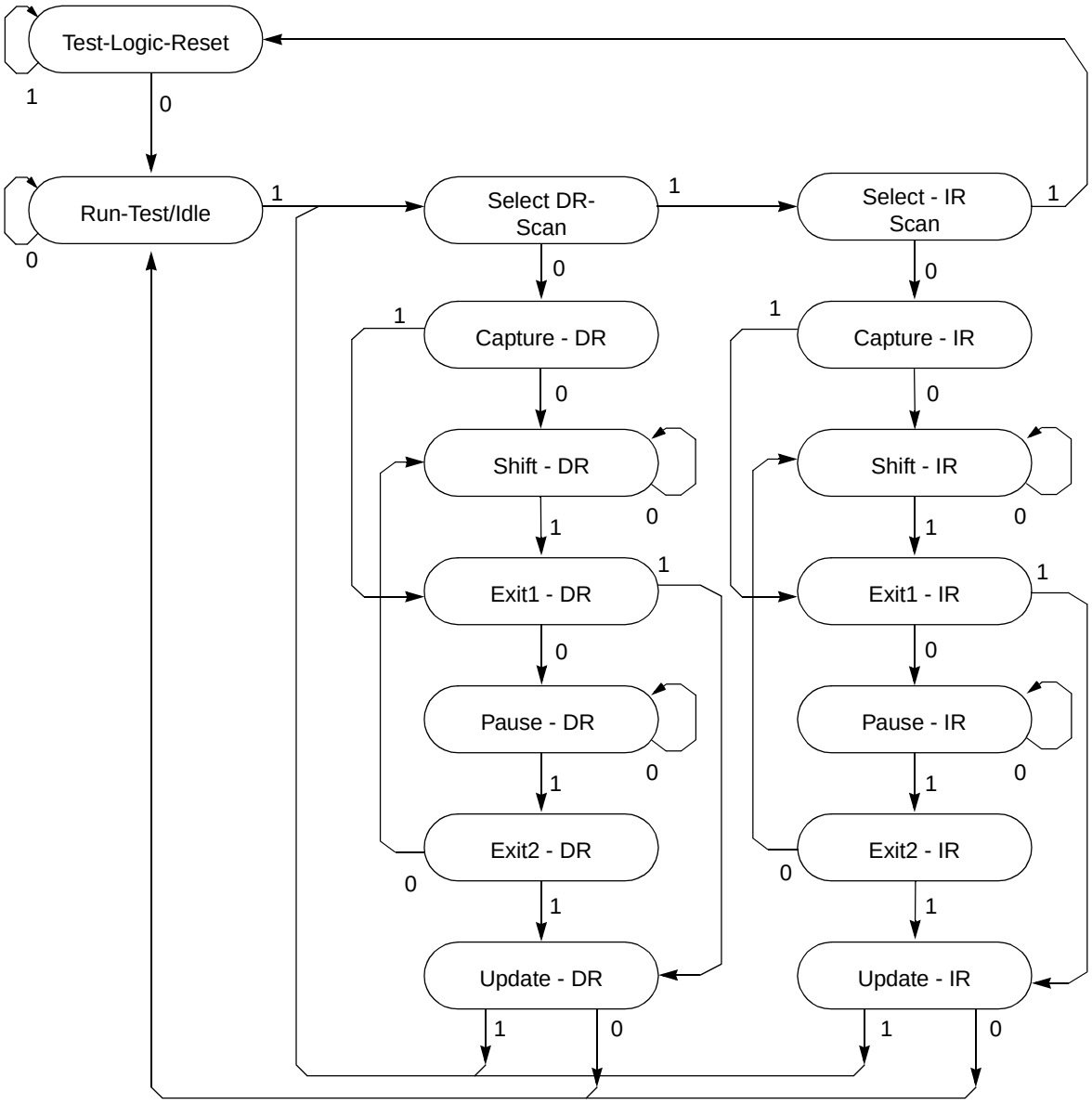


Figure 16-2 OnCE Controller

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Resources contained in the OnCE module that do not require the CPU to be halted for access may be controlled while the CPU is executing and do not interfere with normal processor execution. Accesses to certain resources, such as the PC FIFO and the count registers, while not part of the CPU, may require the CPU to be stopped to allow access to avoid synchronization hazards. If it is known that the CPU clock is enabled and running no slower than the TCK input, there is sufficient synchronization performed to allow reads but not writes of these specific resources. Debug firmware may ensure that it is safe to access these resources by reading the OSR to determine the state of the CPU prior to access. All other cases require the CPU to be in the debug state for deterministic operation.

16.3 OnCE Pins

The following paragraphs describe the pins associated with the OnCE controller and serial interface component.

The OnCE pin interface is used to transfer OnCE instructions and data to the OnCE control block. Depending on the particular resource being accessed, the CPU may need to be placed in debug mode. For resources outside of the CPU block and contained in the OnCE block, the processor is not disturbed and may continue execution. If a processor resource is required, the OnCE controller may assert a debug request ($\overline{\text{DBGRQ}}$) to the CPU. This causes the CPU to finish the instruction being executed, save the instruction pipeline information, enter debug mode, and wait for further commands. Asserting $\overline{\text{DBGRQ}}$ causes the device to exit stop, doze, or wait mode.

16.3.1 Debug Serial Input (TDI)

Data and commands are provided to the OnCE controller through the TDI pin. Data is latched on the rising edge of the TCK serial clock. Data is shifted into the OnCE serial port least significant bit (LSB) first.

16.3.2 Debug Serial Clock (TCK)

The TCK pin supplies the serial clock to the OnCE control block. The serial clock provides pulses required to shift data and commands into and out of the OnCE serial port. (Data is clocked into the OnCE on the rising edge and is clocked out of the OnCE serial port on the falling edge.) The debug serial clock frequency must be no greater than 50% of the processor clock frequency.

16.3.3 Debug Serial Output (TDO)

Serial data is read from the OnCE block through the TDO pin. Data is always shifted out the OnCE serial port LSB first. Data is clocked out of the OnCE serial port on the falling edge of TCK. TDO is three-stateable and is actively driven in the shift-IR and shift-DR controller states. TDO changes on the falling edge of TCK.

16.3.4 Debug Mode Select (TMS)

The debug mode select input is used to cycle through states in the OnCE debug controller. Toggling the TMS pin while clocking with TCK controls the transitions through the TAP state controller.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

16.3.5 Test Reset ($\overline{\text{TRST}}$)

The test reset input is used to reset the OnCE controller externally by placing the OnCE control logic in a test logic reset state. OnCE operation is disabled in the reset controller and reserved states.

16.3.6 Debug Event ($\overline{\text{DE}}$)

The debug event ($\overline{\text{DE}}$) pin is a bidirectional open drain pin. As an input, $\overline{\text{DE}}$ provides a fast means of entering debug mode from an external command controller. As an output, this pin provides a fast means of acknowledging debug mode entry to an external command controller.

The assertion of this pin by a command controller causes the CPU to finish the current instruction being executed, save the instruction pipeline information, enter debug mode, and wait for commands to be entered from the TDI line. If $\overline{\text{DE}}$ was used to enter debug mode, then $\overline{\text{DE}}$ must be negated after the OnCE responds with an acknowledgment and before sending the first OnCE command.

The assertion of this pin by the CPU acknowledges that it has entered debug mode and is waiting for commands to be entered from the TDI line.

16.4 OnCE Controller and Serial Interface

Figure 16-3 is a block diagram of the OnCE controller and serial interface.

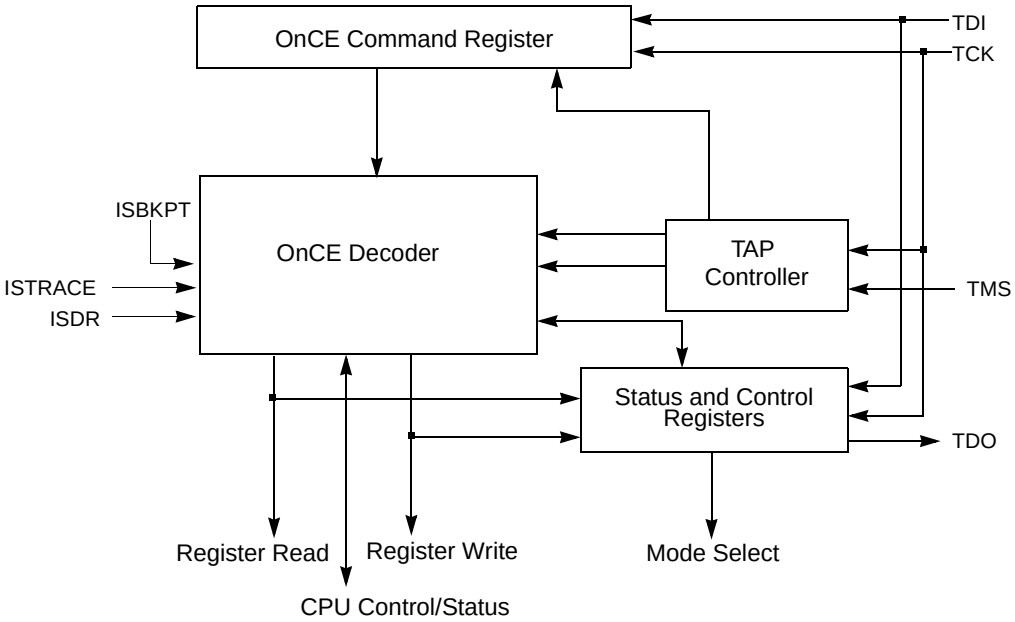


Figure 16-3 OnCE Controller and Serial Interface

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

The OnCE controller and serial interface contain the following blocks: OnCE TAP controller, the OnCE command register, OnCE decoder, and the OnCE control and status registers.

The OnCE command register acts as the IR for the TAP controller. All other OnCE resources are treated as data registers (DR) by the TAP controller. The command register is loaded by serially shifting in commands during the TAP controller shift-IR state, and is loaded during the update-IR state. The command register selects a OnCE resource to be accessed as a data register (DR) during the TAP controller capture-DR, shift-DR and update-DR states.

16.5 OnCE Interface Signals

The following paragraphs describe the OnCE interface signals to other internal blocks associated with the OnCE controller. These signals are not available externally, and descriptions are provided to improve understanding of OnCE operation.

16.5.1 Internal Debug Request Input ($\overline{\text{IDR}}$)

The internal debug request input is a hardware signal which is used in some implementations to force an immediate debug request to the CPU. If present and enabled, it functions in an identical manner to the control function provided by the DR control bit in the OnCE control register (OCR). This input is maskable by a control bit in OCR.

16.5.2 CPU Debug Request ($\overline{\text{DBGRQ}}$)

The $\overline{\text{DBGRQ}}$ signal is asserted by the OnCE control logic to request the CPU to enter the debug state. It may be asserted for a number of different conditions. Assertion of this signal causes the CPU to finish the current instruction being executed, save the instruction pipeline information, enter debug mode, and wait for further commands. Asserting $\overline{\text{DBGRQ}}$ causes the device to exit stop, doze, or wait mode.

16.5.3 CPU Debug Acknowledge ($\overline{\text{DBGACK}}$)

The CPU asserts the $\overline{\text{DBGACK}}$ signal upon entering the debug state. This signal is part of the handshake mechanism between the OnCE control logic and the CPU.

16.5.4 CPU Breakpoint Request ($\overline{\text{BRKRQ}}$)

The $\overline{\text{BRKRQ}}$ signal is asserted by the OnCE control logic to signal that a breakpoint condition has occurred for the current CPU bus access.

16.5.5 CPU Address, Attributes (ADDR, ATTR)

The CPU address and attribute information may be used in the memory breakpoint logic to qualify memory breakpoints with access address and cycle type information.

16.5.6 CPU Status (PSTAT)

The trace logic uses the CPU PSTAT signals to qualify trace count decrements with specific CPU activity.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

16.5.7 OnCE Debug Output ($\overline{\text{DEBUG}}$)

The OnCE debug output ($\overline{\text{DEBUG}}$) is used to indicate to on-chip resources that a debug session is in progress. Peripherals and other units may use this signal to modify normal operation for the duration of a debug session. This may involve the CPU executing a sequence of instructions solely for the purpose of visibility/system control. These instructions are not part of the normal instruction stream the CPU would have executed had it not been placed in debug mode.

This signal is asserted the first time the CPU enters the debug state and remains asserted until the CPU is released by a write to the OnCE command register with the GO and EX bits set, and a register specified as either “No register selected” or the CPUSCR. This signal remains asserted even though the CPU may enter and exit the debug state for each instruction executed under control of the OnCE controller. See **16.6.1 OnCE Command Register (OCMR)** for more information on the function of the GO and EX bits.

16.6 OnCE Controller Registers

This section describes the OnCE controller registers:

- OnCE Command Register (OCMR)
- OnCE Control Register (OCR)
- OnCE Status Register (OSR)

All OnCE registers are addressed by means of the RS field in the OMCR, as shown in **Table 16-1**.

Other OnCE registers are described in **16.8 Memory Breakpoint Logic** and **16.9 OnCE Trace Logic**.

16.6.1 OnCE Command Register (OCMR)

The OnCE command register (OCMR) is an 8-bit shift register that receives its serial data from the TDI pin. This register corresponds to the JTAG IR, and is loaded when the update-IR TAP controller state is entered. It holds the 8-bit commands shifted in during the shift-IR controller state to be used as input for the OnCE decoder. The OCMR contains fields for controlling access to a OnCE resource, as well as controlling single-step operation, and exit from OnCE mode.

Although the OCMR is updated during the update-IR TAP controller state, the corresponding resource is accessed in the DR scan sequence of the TAP controller, and as such, the update-DR state must be transitioned through in order for an access to occur. In addition, the update-DR state must also be transitioned through in order for the single-step and/or exit functionality to be performed, even though the command appears to have no data resource requirement associated with it.

The command register is shown in **Figure 16-4**.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

OCMR — OnCE Command Register

BIT 7	6	5	4	3	2	1	BIT 0
R/W	GO	EX	RS				

Figure 16-4 OnCE Command Register

R/W — Read/Write Command

The R/W bit specifies the direction of data transfer.

0 = Write the data associated with the command into the register specified by the RS field.

1 = Read the data contained in the register specified by the RS field.

GO — Go Command

When the GO bit is set, the device executes the instruction that resides in the IR register in the CPUSCR. To execute the instruction, the processor leaves debug mode, executes the instruction, and if the EX bit is cleared, returns to debug mode immediately after executing the instruction. The processor resumes normal operation if the EX bit is set. The GO command is executed only if the operation is a read/write to either CPUSCR or “No register selected”. Otherwise, the GO bit is ignored. The processor leaves debug mode after the TAP controller update-DR state is entered.

0 = Inactive (no action taken)

1 = Execute instruction in IR

EX — Exit Command

When the EX bit is set, the processor leaves debug mode and resumes normal operation until another debug request is generated. The exit command is executed only if the Go command is issued, and the operation is a read/write to CPUSCR or read/write to “No register selected”. Otherwise the EX bit is ignored. The processor exits debug mode after the TAP controller update-DR state is entered.

0 = Remain in debug mode

1 = Leave debug mode

RS — Register Select

The register select bits define the source or destination register for the read or write operation, respectively. **Table 16-1** shows OnCE register addresses.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

Table 16-1 OnCE Register Addressing

RS	Register Selected
00000	Reserved
00001	Reserved
00010	Reserved
00011	Trace Counter (OTC)
00100	Memory Breakpoint Counter A (MBCA)
00101	Memory Breakpoint Counter B (MBCB)
00110	Program Counter FIFO and Increment Counter
00111	Breakpoint Address Base Register A (BABA)
01000	Breakpoint Address Base Register B (BABB)
01001	Breakpoint Address Mask Register A (BAMA)
01010	Breakpoint Address Mask Register B (BAMB)
01011	CPU Scan Register (CPUSCR)
01100	No Register Selected (Bypass)
01101	OnCE Control Register (OCR)
01110	OnCE Status Register (OSR)
01111	Reserved (Factory Test Control Register — do not access)
10000	Reserved (MEM_BIST, do not access)
10001 – 10110	Reserved (Bypass, do not access)
10111	Reserved (LSRL, do not access)
11000 – 11110	Reserved (Bypass, do not access)
11111	Bypass

16.6.2 OnCE Control Register (OCR)

The OnCE control register (OCR) is a 32-bit register used to select the events that will put the device in debug mode and to enable or disable sections of the OnCE logic. The control bits are read/write.

OCR — OnCE Control Register

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SQC	
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DR	IDRE	TME	FRZC	RCB	BCB					RCA	BCA				
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 16-5 OnCE Control Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

SQC — Sequential Control

The SQC field allows memory breakpoint B and trace occurrences to be suspended until a qualifying event occurs. This field is cleared on test logic reset.

Table 16-2 Sequential Control Field Settings

SQC[1:0]	Meaning
00	Disable sequential control operation. Memory breakpoints and trace operation are unaffected by this field.
01	Suspend normal trace counter operation until a breakpoint condition occurs for memory breakpoint B. If this mode is selected, memory breakpoint B occurrences no longer cause a breakpoint request to be generated. Instead, trace counter comparisons are suspended until the first memory breakpoint B occurrence. After the first memory breakpoint B occurrence, trace counter control is released to perform normally (assuming TME is set). This allows a sequence of breakpoint conditions to be specified prior to trace counting.
10	Qualify memory breakpoint B matches with a breakpoint occurrence for memory breakpoint A. If this bit is set, memory breakpoint A occurrences no longer cause a breakpoint request to be generated. Instead, memory breakpoint B comparisons are suspended until the first memory breakpoint A occurrence. After the first memory breakpoint A occurrence, memory breakpoint B is enabled to perform normally. This allows a sequence of breakpoint conditions to be specified.
11	Combine the qualifications specified by the 01 and 10 encodings of this field. In this mode, no breakpoint requests are generated, and trace count operation is enabled (if TME is set) once a memory breakpoint B occurrence follows a memory breakpoint A occurrence.

DR — CPU Debug Request Control

This control bit is used to request the CPU to enter debug mode unconditionally. The CPU indicates that debug mode has been entered via the PM bits in the OnCE status register. Once the CPU enters debug mode, it returns there even with a write to the OCMR with GO and EX set until the DR bit is cleared. This bit is cleared on test logic reset.

IDRE — Internal Debug Request Enable

This control bit is used to enable internally generated debug requests. The internal debug request input to the OnCE control logic ($\overline{\text{IDR}}$) may not be used in all implementations. In some implementations, the $\overline{\text{IDR}}$ control input may be connected and used as an additional hardware debug request. This bit is cleared on test logic reset.

0 = Disable $\overline{\text{IDR}}$ input operation
1 = Enable $\overline{\text{IDR}}$ input operation

TME — Trace Mode Enable

The TME control bit enables the OnCE trace mode operation (see **16.9 OnCE Trace Logic**). This bit is cleared on test logic reset. Trace operation is also affected by the SQC field described above.

0 = Disable trace operation
1 = Enable trace operation

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

FRZC — Freeze Control

This control bit is used in conjunction with memory breakpoint B registers to select between asserting a breakpoint condition when a memory breakpoint B occurs, or freezing the PC FIFO from further updates when memory breakpoint B occurs while allowing the CPU to continue execution. The PC FIFO remains frozen until the FRZO bit in the OSR is cleared.

0 = Memory breakpoint B occurrence causes assertion of a breakpoint condition
1 = Memory breakpoint B occurrence causes a freeze of PC FIFO from further updates and no breakpoint assertion

RCB, RCA — Memory Breakpoint B, A Range Control

These control bits condition enabled memory breakpoints. They condition whether memory breakpoint matches will occur when a memory address falls either within the range defined by memory base address and mask, or outside the range.

0 = Condition breakpoint on access within range
1 = Condition breakpoint on access outside of range

BCB, BCA — Memory Breakpoint B, A Control

These control bits enable memory breakpoints and qualify the access attributes to select whether the breakpoint match will be recognized for read, write, or instruction fetch (program space) accesses. These bits are cleared on test logic reset. See **Table 16-3** for the definition of the BCA and BCB fields.

Table 16-3 Memory Breakpoint Control Field Settings

BC4	BC3	BC2	BC1	BC0	Description
0	0	0	0	0	Breakpoint disabled
0	0	0	0	1	Qualify match with any access
0	0	0	1	0	Qualify match with any instruction access
0	0	0	1	1	Qualify match with any data access
0	0	1	0	0	Qualify match with any change of flow instruction access
0	0	1	0	1	Qualify match with any data write
0	0	1	1	0	Qualify match with any data read
0	0	1	1	1	Reserved
0	1	x	x	x	Reserved
1	0	0	0	0	Reserved
1	0	0	0	1	Qualify match with any user access
1	0	0	1	0	Qualify match with any user instruction access
1	0	0	1	1	Qualify match with any user data access
1	0	1	0	0	Qualify match with any user change of flow access
1	0	1	0	1	Qualify match with any user data write
1	0	1	1	0	Qualify match with any user data read
1	0	1	1	1	Reserved
1	1	0	0	0	Reserved
1	1	0	0	1	Qualify match with any supervisor access
1	1	0	1	0	Qualify match with any supervisor instruction access
1	1	0	1	1	Qualify match with any supervisor data access
1	1	1	0	0	Qualify match with any supervisor change of flow access
1	1	1	0	1	Qualify match with any supervisor data write
1	1	1	1	0	Qualify match with any supervisor data read
1	1	1	1	1	Reserved

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

16.6.3 OnCE Status Register (OSR)

The OnCE status register (OSR) is a 16-bit register used to indicate the reason(s) that debug mode was entered and the current operating mode of the CPU. These status bits are read only.

OSR — OnCE Status Register

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	HDRO	DRO	MBO	SWO	TO	FRZO	SQB	SQA	PM	
W																
RESET:							0	0	0	0	0	0	0	0	0	0

Figure 16-6 OnCE Status Register

HDRO — Hardware Debug Request Occurrence

This read-only status bit is set when the processor enters debug mode as a result of a hardware debug request from the $\overline{\text{IDR}}$ signal or the $\overline{\text{DE}}$ pin. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

DRO — Debug Request Occurrence

This read-only status bit is set when the processor enters debug mode and the debug request (DR) control bit in the OnCE control register is set. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

MBO — Memory Breakpoint Occurrence

This read-only status bit is set when a memory breakpoint request has been issued to the CPU via the $\overline{\text{BRKRQ}}$ input and the CPU enters debug mode. In some situations involving breakpoint requests on instruction prefetches, the CPU may discard the request along with the prefetch. In this case, this bit may become set due to the CPU entering debug mode for another reason. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

SWO — Software Debug Occurrence

This read-only status bit is set when the processor enters debug mode of operation as a result of the execution of the **bkpt** instruction. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

TO — Trace Count Occurrence

This read-only status bit is set when the trace counter reaches zero with the trace mode enabled and the CPU enters debug mode. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

FRZO — FIFO Freeze Occurrence

This read-only status bit is set when a FIFO freeze occurs. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

SQB — Sequential Breakpoint B Arm Occurrence

This read-only status bit is set when sequential operation is enabled and a memory breakpoint B event has occurred to enable trace counter operation. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SQA — Sequential Breakpoint A Arm Occurrence

This read-only status bit is set when sequential operation is enabled and a memory breakpoint A event has occurred to enable memory breakpoint B operation. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

PM — Processor Mode

These status bits indicate the processor operating mode. They allow coordination of the OnCE controller with the CPU to synchronize the two.

Table 16-4 Processor Mode Field Settings

PM[1:0]	Meaning
00	Processor in normal mode
01	Processor in stop, doze, or wait mode
10	Processor in debug mode
11	Reserved

16.7 OnCE Decoder (ODEC)

The OnCE decoder (ODEC) receives as input the 8-bit command from the OCMR and status signals from the processor. The ODEC generates all the strobes required for reading and writing the selected OnCE registers.

16.8 Memory Breakpoint Logic

Memory breakpoints can be set for a particular memory location or on accesses within an address range. The breakpoint logic contains an input latch for addresses, registers that store the base address and address mask, comparators, attribute qualifiers, and a breakpoint counter. **Figure 16-7** illustrates the basic functionality of the OnCE memory breakpoint logic. This logic is duplicated to provide two independent breakpoint resources.

Address comparators can be used to determine where a program may be getting lost or when data is being written to areas which should not be written. They are also useful in halting a program at a specific point to examine or change registers or memory. Using address comparators to set breakpoints enables the user to set breakpoints in RAM or ROM in any operating mode. Memory accesses are monitored according to the contents of the OCR.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

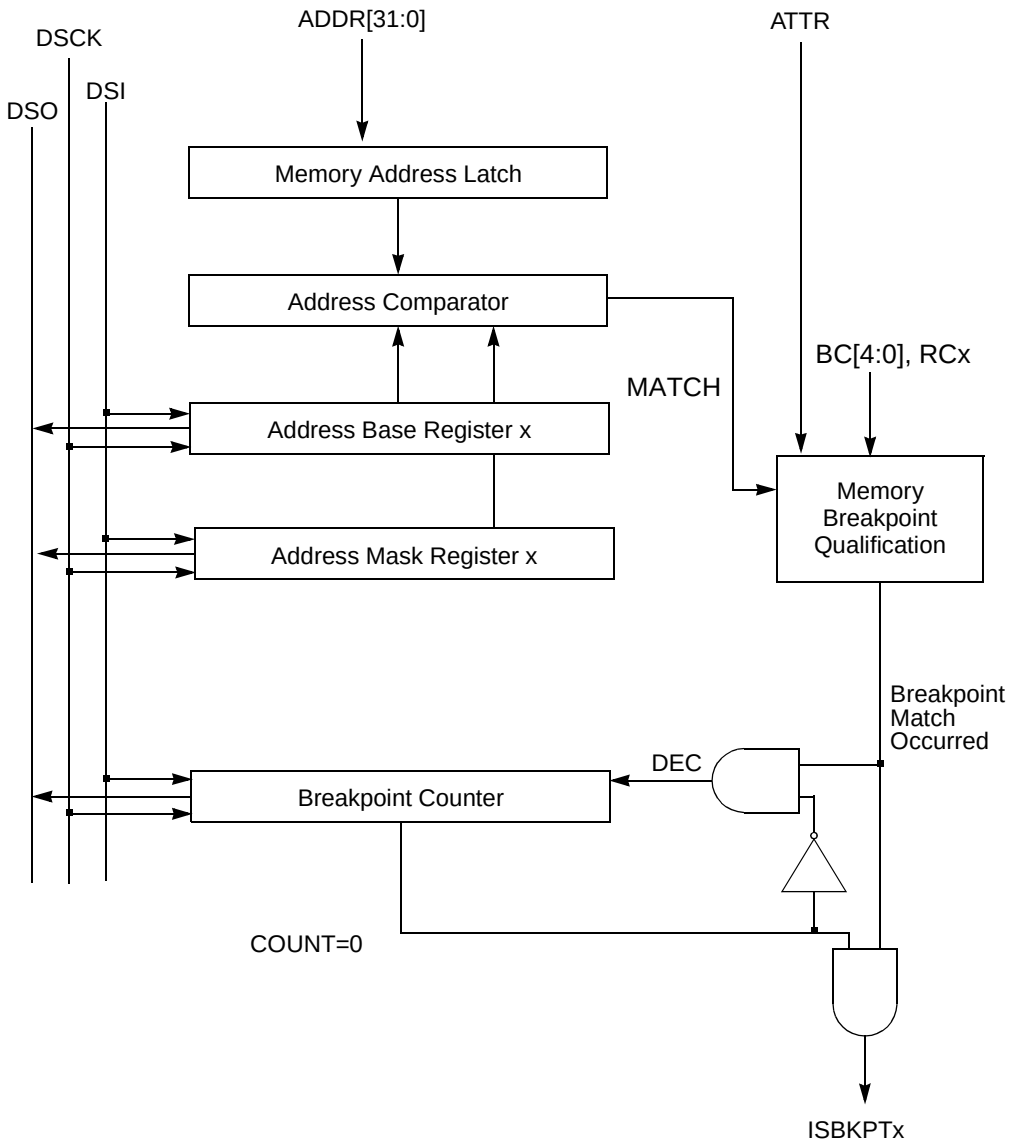


Figure 16-7 OnCE Memory Breakpoint Logic

The address comparator generates a match signal when the address on the bus matches the address stored in the breakpoint address base register, as masked with individual bit masking capability provided by the breakpoint address mask register. The address match signal and the access attributes are further qualified with the RCx and BCx[4:0] control bits. This qualification is used to decrement the breakpoint counter conditionally if its contents are non-zero. If the contents are zero, the counter is not decremented and the breakpoint event occurs (ISBKPTx asserted).

16.8.1 Memory Address Latch (MAL)

The memory address latch (MAL) is a 32-bit register that latches the address bus on every access.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

16.8.2 Breakpoint Address Base Registers (BABA, BABB)

The 32-bit breakpoint address base registers (BABA, BABB) store memory breakpoint base addresses. BABA and BABB can be read or written through the OnCE serial interface. Before enabling breakpoints, the external command controller should load these registers.

16.8.3 Breakpoint Address Mask Registers (BAMA, BAMB)

The 32-bit breakpoint address mask registers (BAMA, BAMB) store memory breakpoint base address masks. BAMA and BAMB can be read or written through the OnCE serial interface. Before enabling breakpoints, the external command controller should load these registers.

16.8.4 Breakpoint Address Comparators

The breakpoint address comparators are not externally accessable. Each compares the memory address stored in MAL with the contents of BABx, as masked by BAMx, and signals the control logic when a match occurs.

16.8.5 Memory Breakpoint Counters (MBCA, MBCB)

The 16-bit memory breakpoint counter x (MBCx) register is loaded with a value equal to the number of times, minus one, that a memory access event should occur before a memory breakpoint is declared. The memory access event is specified by the RCx and BCx[4:0] bits in the OCR register and by the memory base and mask registers. On each occurrence of the memory access event, the breakpoint counter, if currently non-zero, is decremented. When the counter has reached the value of zero and a new occurrence takes place, the ISBKPTx signal is asserted and causes the CPU's $\overline{\text{BRKRQ}}$ input to be asserted. The MBCx can be read or written through the OnCE serial interface.

Anytime the breakpoint registers are changed, or a different breakpoint event is selected in the OCR, the breakpoint counter must be written afterward. This assures that the OnCE breakpoint logic is reset and that no previous events will affect the new breakpoint event selected.

16.9 OnCE Trace Logic

The OnCE trace logic allows the user to execute instructions in single or multiple steps before the device returns to debug mode and awaits OnCE commands from the debug serial port. (The OnCE trace logic is independent of the M•CORE trace facility, which is controlled through the trace mode bits in the M•CORE processor status register). The OnCE trace logic block diagram is shown in **Figure 16-8**.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

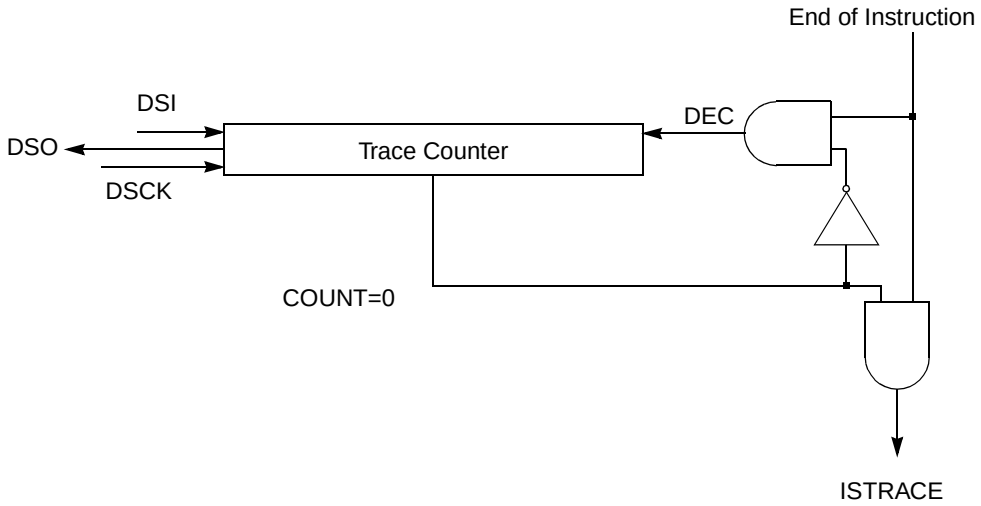


Figure 16-8 OnCE Trace Logic Block Diagram

16.9.1 Trace Counter (OTC)

The trace counter (OTC) allows more than one instruction to be executed in real time before the device returns to debug mode. This feature helps the software developer debug sections of code that are time-critical. The trace counter also enables the user to count the number of instructions executed in a code segment.

The OTC is a 16-bit counter that can be read, written, or cleared through the OnCE serial interface. If N instructions are to be executed before entering debug mode, the trace counter should be loaded with N–1. N must not equal zero unless the sequential breakpoint control capability described in **16.6.2 OnCE Control Register (OCR)** is being used. In this case a value of zero (indicating a single instruction) is allowed.

The trace counter is cleared by hardware reset.

16.9.2 Trace Operation

The following steps initiate trace mode operation:

1. Load the counter with a value. This value must be non-zero, unless the sequential breakpoint control capability described in **16.6.2 OnCE Control Register (OCR)** is being used. In this case a value of zero (indicating a single instruction) is allowed.
2. Initialize the program counter and instruction register in the CPUSCR with values corresponding to the start location of the instruction(s) to be executed real-time.
3. Set the TME bit in the OCR.
4. Release the processor from debug mode by executing the appropriate command issued by the external command controller.

When debug mode is exited, the counter is decremented after each execution of an instruction. Interrupts can be serviced, and all instructions executed (including interrupt services) will decrement the trace counter.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

When the trace counter decrements to zero, the OnCE control logic requests that the processor re-enter debug mode, and the trace occurrence bit TO in the OSR is set to indicate that debug mode has been requested as a result of the trace count function. The trace counter allows a minimum of two instructions to be specified for execution prior to entering trace (specified by a count value of one), unless the sequential breakpoint control capability described in **16.6.2 OnCE Control Register (OCR)** is being used. In this case a value of zero (indicating a single instruction) is allowed.

16.10 Methods of Entering Debug Mode

The OSR indicates that the CPU has entered debug mode via the PM status field. The following paragraphs discuss conditions that invoke debug mode.

16.10.1 Debug Request During $\overline{\text{RESET}}$

When the DR bit in the OCR is set, assertion of $\overline{\text{RESET}}$ causes the device to enter debug mode. In this case the device may fetch the reset vector and the first instruction of the reset exception handler but does not execute an instruction before entering debug mode.

16.10.2 Debug Request During Normal Activity

Setting the DR bit in the OCR during normal device activity causes the device to finish the execution of the current instruction and then enter debug mode. Note that in this case the device completes the execution of the current instruction and stops after the newly fetched instruction enters the CPU instruction latch. This process is the same for any newly fetched instruction, including instructions fetched by interrupt processing or those that will be aborted by interrupt processing.

16.10.3 Debug Request During Stop, Doze, or Wait Mode

Setting the DR bit in the OCR when the device is in stop, doze, or wait mode (i. e., has executed a **stop, doze,** or **wait** instruction) causes the device to exit the low-power state and enter the debug mode. Note that in this case, the device completes the execution of the **stop, doze,** or **wait** instruction and halts after the next instruction enters the instruction latch.

16.10.4 Software Request During Normal Activity

Executing the **bkpt** instruction when the FDB (force debug enable mode) control bit in the control state register is set, causes the CPU to enter debug mode after the instruction following the **bkpt** instruction has entered the instruction latch.

16.10.5 Enabling OnCE Trace Mode

When the OnCE trace mode mechanism is enabled and the trace count is greater than zero, the trace counter is decremented for each instruction executed. Completing execution of an instruction when the trace counter is zero causes the CPU to enter debug mode.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

NOTE

Only instructions actually executed cause the trace counter to decrement, i.e., an aborted instruction does not decrement the trace counter and does not invoke debug mode.

16.10.6 Enabling OnCE Memory Breakpoints

When the OnCE memory breakpoint mechanism is enabled with a breakpoint counter value of zero, the device enters debug mode after completing the execution of the instruction that caused the memory breakpoint to occur. In case of breakpoints on instruction fetches, the breakpoint is acknowledged immediately after the execution of the fetched instruction. In case of breakpoints on data memory addresses, the breakpoint is acknowledged after the completion of the memory access instruction.

16.11 Pipeline Information and Write-Back Bus Register

A number of on-chip registers store the CPU pipeline status and are configured in a single scan chain for access by the OnCE controller. The CPUSCR OnCE register contains these processor resources, which are used to restore the pipeline and resume normal device activity upon return from debug mode. These resources also provide a mechanism for the emulator software to access processor and memory contents. **Figure 16-9** shows the block diagram of the pipeline information registers contained in the CPUSCR.

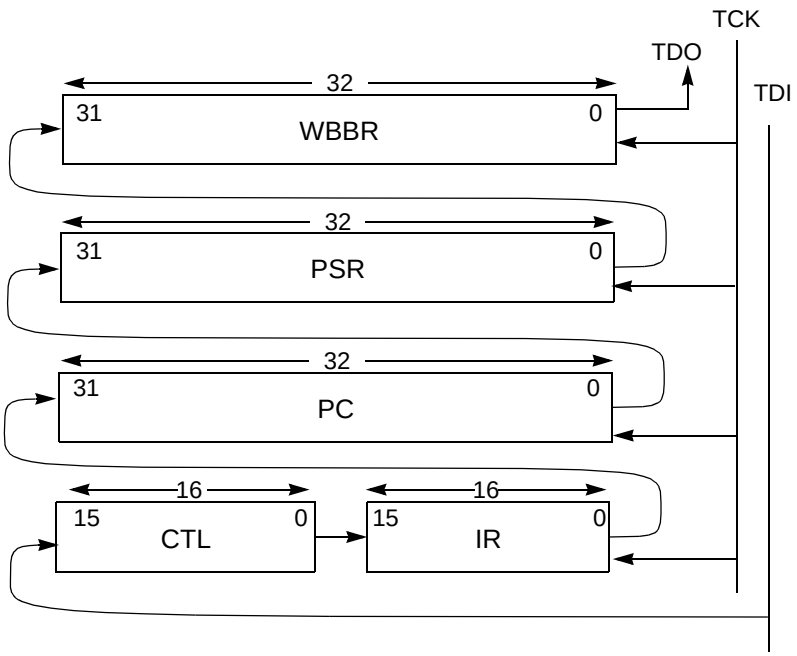


Figure 16-9 CPU Scan Chain Register (CPUSCR)

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

16.11.1 Program Counter Register (PC)

The OnCE program counter register (PC) is a 32-bit latch that stores the value in the CPU program counter when the device enters debug mode. The CPU PC is affected by operations performed during debug mode and must be restored by the external command controller when the CPU returns to normal mode.

16.11.2 Instruction Register (IR)

The instruction register (IR) provides a mechanism for controlling the debug session. The IR allows the debug control block to execute selected instructions; the debug control module provides single-step capability.

When scan-out begins, the IR contains the opcode of the next instruction to be executed at the time debug mode was entered. This opcode must be saved in order to resume normal execution at the point debug mode was entered.

On scan-in, the IR can be filled with an opcode selected by debug control software in preparation for exiting debug mode. Selecting appropriate instructions allows a user to examine or change memory locations and processor registers.

Once the debug session is complete and normal processing is to be resumed, the IR can be loaded with the value originally scanned out.

16.11.3 Control State Register (CTL)

The control state register (CTL) is used to set control values when debug mode is exited. On scan-in, this register is used to control specific aspects of the CPU. Certain bits reflect internal processor status and should be restored to their original values.

The CTL is a 16-bit latch that stores the value of certain internal CPU state variables before debug mode is entered. This register is affected by the operations performed during the debug session and should be restored by the external command controller when returning to normal mode. In addition to saved internal state variables, the bits are used by emulation firmware to control the debug process.

Reserved bits represent the internal processor state. Restore these bits to their original value after a debug session is completed, i.e., when a OnCE command is issued with the GO and EX bits set and not ignored. Set these bits to ones while instructions are executed during a debug session.

CTL — Control State Register

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							FFY	FDB	SZ		TC			Reserved	
RESET:								0	0	0	0	0	0	0		

Figure 16-10 Control State Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

FFY — Feed Forward Y Operand

This control bit is used to force the content of the WBBR to be used as the Y operand value of the first instruction to be executed following an update of the CPUSCR. This gives the debug firmware the capability of updating processor registers by initializing the WBBR with the desired value, setting the FFY bit, and executing a **mov** instruction to the desired register.

FDB — Force PSR Debug Enable Mode

Setting this control bit places the processor in debug enable mode. In debug enable mode, execution of the **bkpt** instruction as well as recognition of the $\overline{\text{BRKRQ}}$ input causes the processor to enter debug mode, as if the $\overline{\text{DBGRQ}}$ input had been asserted.

SZ — Prefetch Size

This control field is used to drive the CPU SIZ[1:0] outputs on the first instruction prefetch caused by issuing a OnCE command with the GO bit set and not ignored. It should be set to indicate a 16-bit size, i.e., 0b10. This field should be restored to its original value after a debug session is completed, i.e., when a OnCE command is issued with the GO and EX bits set and not ignored.

TC — Prefetch Transfer Code

This control field is used to drive the CPU TC[2:0] outputs on the first instruction prefetch caused by issuing a OnCE command with the GO bit set and not ignored. It should typically be set to indicate a supervisor instruction access, i.e., 0b110. This field should be restored to its original value after a debug session is completed, i.e., when a OnCE command is issued with the GO and EX bits set and not ignored.

16.11.4 Write-Back Bus Register (WBBR)

The write-back bus register (WBBR) is used as a means of passing operand information between the CPU and the external command controller. Whenever the external command controller needs to read the contents of a register or memory location, it forces the device to execute an instruction that brings that information to WBBR.

For example, to read the content of processor register **r0**, a **mov r0,r0** instruction is executed, and the result value of the instruction is latched into the WBBR. The contents of WBBR can then be delivered serially to the external command controller.

To update a processor resource, this register is initialized with a data value to be written, and a **mov** instruction is executed which uses this value as a write-back data value. The FFY bit in the control state register forces the value of the WBBR to be substituted for the normal source value of a **mov** instruction, thus allowing updates to processor registers to be performed.

16.11.5 Processor Status Register (PSR)

The OnCE processor status register (PSR) is a 32-bit latch used to read or write the M•CORE processor status register. Whenever the external command controller needs to save or modify the contents of the M•CORE processor status register, this register is used. This register is affected by the operations performed in debug mode and must be restored by the external command controller when returning to normal mode.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

16.12 Instruction Address FIFO Buffer (PC FIFO)

To ease debugging activity and keep track of program flow, a first-in-first-out (FIFO) buffer stores the addresses of the last eight instruction change-of-flow prefetches that were issued.

The FIFO is implemented as a circular buffer containing eight 32-bit registers and one 3-bit counter. All the registers have the same address, but any read access to the FIFO address causes the counter to increment and point to the next FIFO register. The registers are serially available to the external command controller through the common FIFO address. **Figure 16-11** shows the block diagram of the PC FIFO.

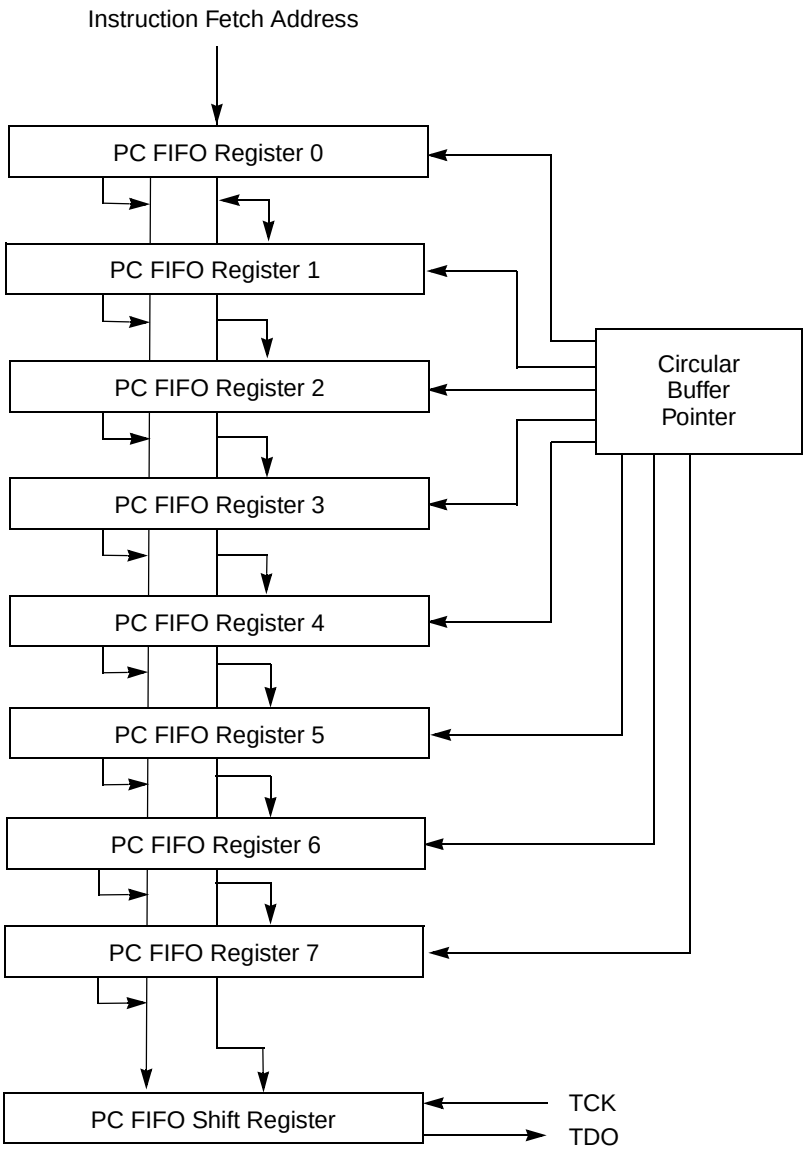


Figure 16-11 OnCE PC FIFO

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

The FIFO is not affected by operations performed in debug mode, except for incrementing the FIFO pointer when the FIFO is read. When debug mode is entered, the FIFO counter points to the FIFO register containing the address of the oldest of the eight change-of-flow prefetches. The first FIFO read obtains the oldest address, and the following FIFO reads return the other addresses from the oldest to the newest (in order of execution).

To ensure FIFO coherence, a complete set of eight reads of the FIFO must be performed. Each read increments the FIFO pointer, causing it to point to the next location. After eight reads, the pointer points to the same location as before the start of the read procedure.

16.12.1 Reserved Test Control Registers (Reserved, MEM_BIST, FTCR, LSRL)

These registers are reserved for factory testing.

WARNING

To prevent damage to the device or system, do not access these registers during normal operation.

16.13 Serial Protocol Description

The following protocol permits an efficient means of communication between the OnCE external command controller and the MMC2001. Before starting any debugging activity, the external command controller must wait for an acknowledgment that the device has entered debug mode. The external command controller communicates with the device by sending 8-bit commands to the OnCE command register and 16 to 128 bits of data to one of the other OnCE registers. Both commands and data are sent or received LSB first. After sending a command, the external command controller must wait for the processor to acknowledge execution of certain commands before it can properly access another OnCE register.

16.13.1 OnCE Commands

The OnCE commands can be classified as follows:

- Read commands (the device delivers the required data)
- Write commands (the device receives data and writes the data in one of the OnCE registers)
- Commands that do not have data transfers associated with them.

The commands are eight bits long and have the format shown in **Figure 16-4**.

16.14 Target Site Debug System Requirements

A typical debug environment consists of a target system in which the MMC2001 resides in the user-defined hardware.

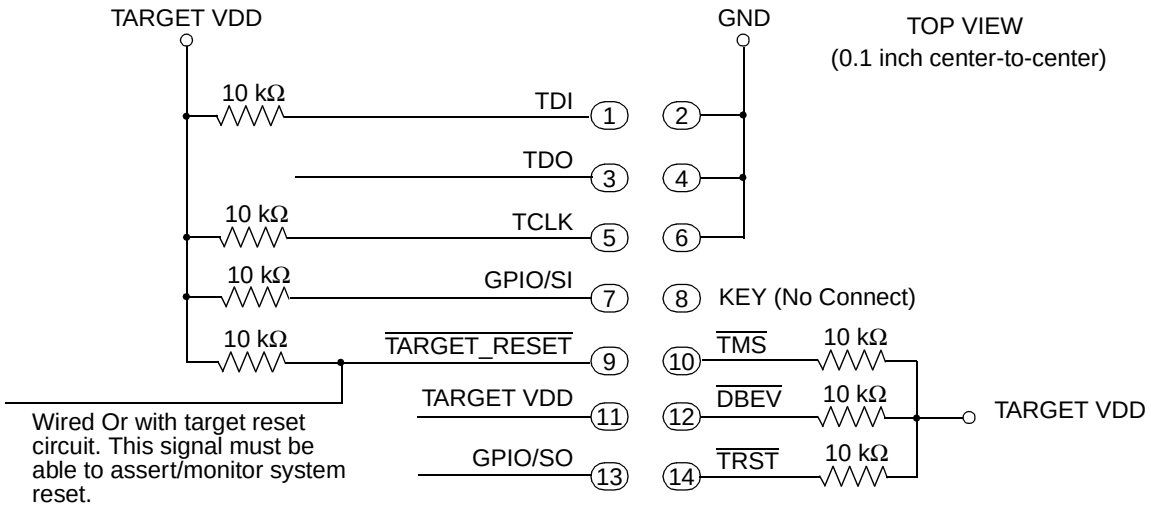
For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

The external command controller acts as the medium between the MMC2001 target system and a host computer. The external command controller circuit acts as a serial debug port driver and host computer command interpreter. The controller issues commands based on the host computer inputs from a user interface program which communicates with the user.

16.15 Interface Connector For JTAG/OnCE Serial Port

Figure 16-12 shows the recommended connector pinout and interface requirements for debug controllers that access the JTAG/OnCE port. The connector has two rows of seven pins with 0.1 inch center-to-center spacing between pins in each row and each column.



Note: GPIO/SI and GPIO/SO are not required for OnCE operation at this time.
These pins can be used for high speed downloads with a recommended interface.

Figure 16-12 Recommended Connector Interface to JTAG/OnCE Port

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

APPENDIX A
ELECTRICAL CHARACTERISTICS

This section contains preliminary information on DC/AC electrical characteristics and AC timing specifications of the MMC2001.

A.1 Maximum Ratings

Table A-1 Maximum Ratings

Rating	Symbol	Value	Unit
Internal Supply Voltage	V_{CCI}	–0.3 to + 4.5	V
External Supply Voltage	V_{CCE}	–0.3 to + 4.5	V
Operating Temperature Range	T_a	–40 to + 85	°C
Storage Temperature	T_{STG}	–55 to + 150	°C

A.2 DC Electrical Specifications

Table A-2 DC Electrical Specifications

Characteristic	Symbol	Min	Max	Unit
Internal Supply Voltage	V_{CCI}	1.8	3.6	V
External Supply Voltage	V_{CCE}	V_{CCI}	3.6	V
Battery Supply Voltage	V_{CCB}	V_{CCE} – 1.0	V_{CCE}	V
Input High Voltage	V_{IH}	0.7 * V_{CCE}	V_{CCE} + 0.2	V
Input Low Voltage	V_{IL}	–0.3	0.2 * V_{CCE}	V
Input Leakage Current (All Input Only Pins)	I_{IN}	–10	10	µA
Hi-Z (Off State) Leakage Current (All Input, Non-Crystal Outputs, and I/O Pins)	I_{OZ}	–10	10	µA
Signal Low Input Current (TMS, TDI, TCK, $\overline{TRST}$, $\overline{DE}$, ROW[7:0])	I_L	–0.015	0.2	mA
Signal High Input Current, (TMS, TDI, TCK, $\overline{TRST}$, $\overline{DE}$, ROW[7:0])	I_H	–0.015	0.2	mA
Output High Voltage, I_{OH} = 0.4 mA	V_{OH}	0.75 * V_{CCE}	V_{CCE}	V
Output Low Voltage, I_{OL} = 0.8 mA	V_{OL}	0	0.18 * V_{CCE}	V
V_{CCB} Standby Current ($\overline{LVRSTIN}$ asserted) V_{CCI} Supply Current at 2.0 V @ 34 MHz STOP DOZE WAIT RUN	IBATT_standby I_{CC_stop} I_{CC_doze} I_{CC_wait} I_{CC_run}	— — — — —	3 60 3 3 40	µA µA mA mA mA
Pin Capacitance	C_{IN}	—	10	pF
Load Capacitance	C_L	—	50	pF

PRELIMINARY

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

A.3 Clock Input Specifications

Table A-3 Clock Input Specifications

Num	Characteristic	Symbol	Min	Max	Unit
	CLKIN Frequency[1]	CLK	8	34	MHz
1	CLKIN Period	T_{HRC}	29.4	—	ns
2	CLKIN Rise Time (for square wave input)		—	3	ns
3	CLKIN Fall Time (for square wave input)		—	3	ns
	CLKIN Duty Cycle		45	55	%
	CLKIN Input Voltage		0.8	V_{CCE}	Vpp
	Crystal Frequency		—	32.768	kHz
	Crystal Period	T_{LRC}	30.5	—	µs

NOTES:
1. CLKIN is an AC-coupled input requiring a periodic waveform, either a sine wave or a square wave. The DC bias level must keep the minimum level of the signal greater than GND and the maximum level of the signal less than V_{CCE}.

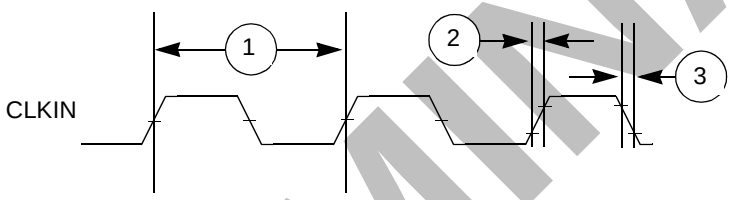


Figure A-1 CLKIN Timing (for Square Wave Input)

A.4 AC Electrical Specifications

The following AC electrical specifications are given for V_{CCI} = 2.0 +/– 10% and V_{CCE} = 3.3 V +/– 10% with a maximum capacitive load of 50 pF on the outputs.

A.4.1 Reset, $\overline{MOD}$ Timing Specifications

Table A-4 Reset, $\overline{MOD}$ Timing Specifications

Num	Characteristic	Expression	Min	Max	Unit
11	$\overline{RSTIN}$ Duration to be Qualified as Valid	$4*T_{LRC}$ + 0.05	122.12	—	µs
12	Delay from $\overline{RSTIN}$ Assertion to $\overline{RSTOUT}$ Assertion	min: 4.5 * T_{LRC} max: 5.5 * T_{LRC}	137.33	167.85	µs
13	Delay from $\overline{RSTIN}$ Negation to $\overline{RSTOUT}$ Negation	8 * T_{LRC}	244.14	—	µs
14	Delay from $\overline{RSTIN}$ Assertion to all Pins at Reset Value (periodically sampled and not 100% tested)	min: 4.5 * T_{LRC} max: 5.5 * T_{LRC}	137.33	167.85	µs
15	$\overline{MOD}$ Setup Time to $\overline{RSTOUT}$ Negation	$4*T_{LRC}$ + 0.05	122.12	—	µs
16	$\overline{MOD}$ Hold Time	—	0	—	ns

Freescale Semiconductor, Inc.
PRELIMINARY

For More Information On This Product,
Go to: www.freescale.com

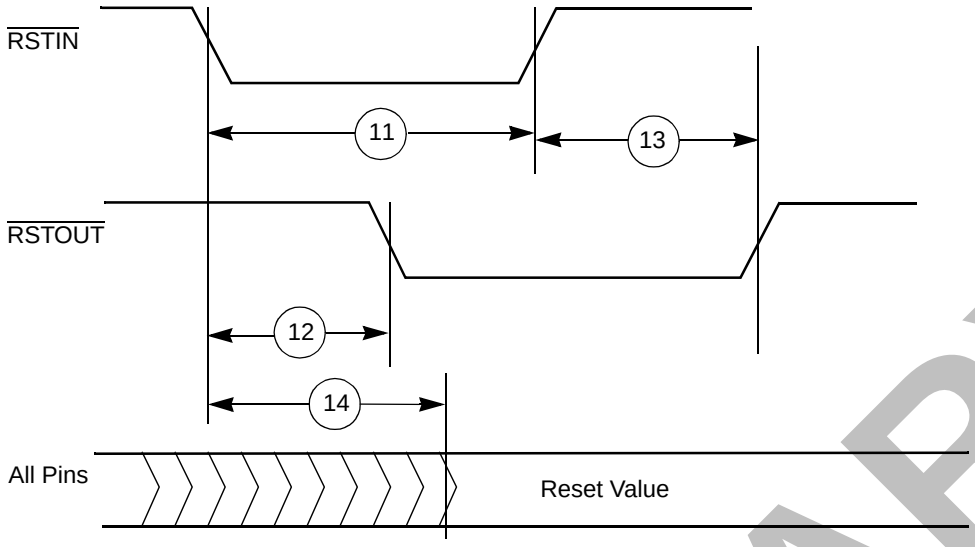


Figure A-2 Reset Timing

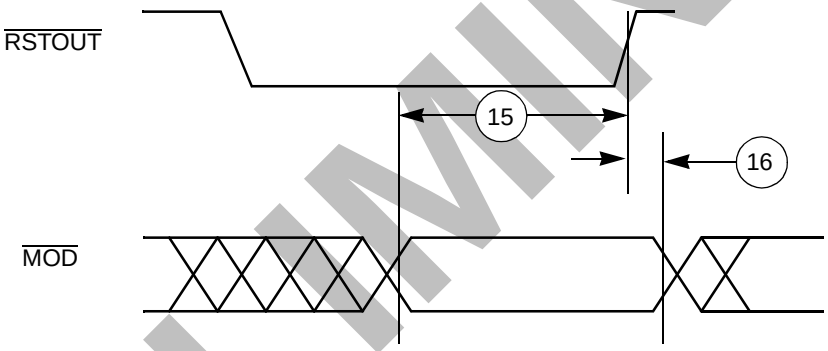


Figure A-3 $\overline{\text{MOD}}$ Timing

A.4.2 External Interrupt Timing Specifications

Table A-5 External Interrupt Timing Specifications

Num	Characteristic	Symbol	Min	Max	Unit
21	Minimum Edge-Triggered INTn Width High	$2 * T_{HRC}+2$	60.8	—	ns
22	Minimum Edge-Triggered INTn Width Low	$2 * T_{HRC}+2$	60.8	—	ns

Freescale Semiconductor, Inc.
PRELIMINARY

For More Information On This Product,
Go to: www.freescale.com

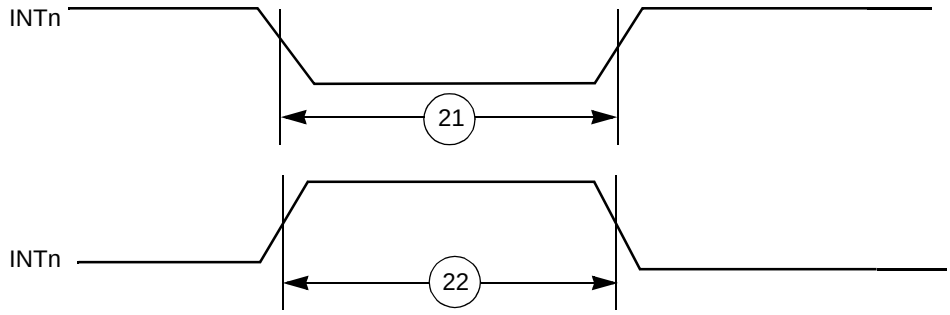


Figure A-4 External Interrupt Timing

A.4.3 EIM Timing Specifications

Table A-6 EIM Timing Specifications[1]

Num	Characteristic	Min	Max	Unit
31	CLKOUT Rise to Address, R/$\overline{W}$ Valid	0	27	ns
32	CLKOUT Rise to Address, R/$\overline{W}$ Invalid (Output Hold)	0	—	ns
33	CLKOUT Rise to CS Asserted	0	29	ns
34	CLKOUT Rise to CS Negated (Output Hold)	0	—	ns
35	CLKOUT Fall to $\overline{OE}$, $\overline{EB}$ Asserted (Read, OEA=0), $\overline{EB}$ Asserted (Write)[2]	0	16	ns
36	CLKOUT Rise to $\overline{OE}$, $\overline{EB}$ Asserted (Read, OEA=1)[2]	0	16	ns
37	CLKOUT Rise to $\overline{OE}$, $\overline{EB}$ Negated (Output Hold) (Read)[2]	0	—	ns
37	CLKOUT Rise to $\overline{EB}$ Negated (Output Hold) (Write, WEN=0)	0	—	ns
38	CLKOUT Fall to $\overline{EB}$ Negated (Output Hold) (Write, WEN=1)	0	—	ns
39	CLKOUT Fall to $\overline{OE}$, $\overline{EB}$ Asserted (WSC=0)[2]	0	13	ns
40	CLKOUT Rise to $\overline{OE}$, $\overline{EB}$ Negated (Output Hold) (WSC=0)[2]	0	—	ns
41	Data In Valid to CLKOUT rise (setup)	17	—	ns
42	CLKOUT Rise to Data In Invalid (hold)	0	—	ns
43	CLKOUT Rise to Data Out Valid (WSC>0)	—	15	ns
44	CLKOUT Rise to Data Out Invalid (Output Hold) (WSC>0)	0	—	ns
45	CLKOUT Rise to Data Out High Impedance (WSC>0)	—	15	ns
46	CLKOUT Fall to Data Out Valid (WSC=0)	—	17	ns
47	CLKOUT Rise to Data Out Invalid (Output Hold) (WSC=0)	0	—	ns
48	CLKOUT Rise to Data Out High Impedance (WSC=0)	—	17	ns

NOTES:
1. Output timing is measured at the pin. The specifications assume a capacitive load of 50 pF.
2. $\overline{EB}$ outputs are asserted for reads if the EBC bit in the corresponding CS control register is cleared.

Freescale Semiconductor, Inc.
PRELIMINARY

For More Information On This Product,
Go to: www.freescale.com

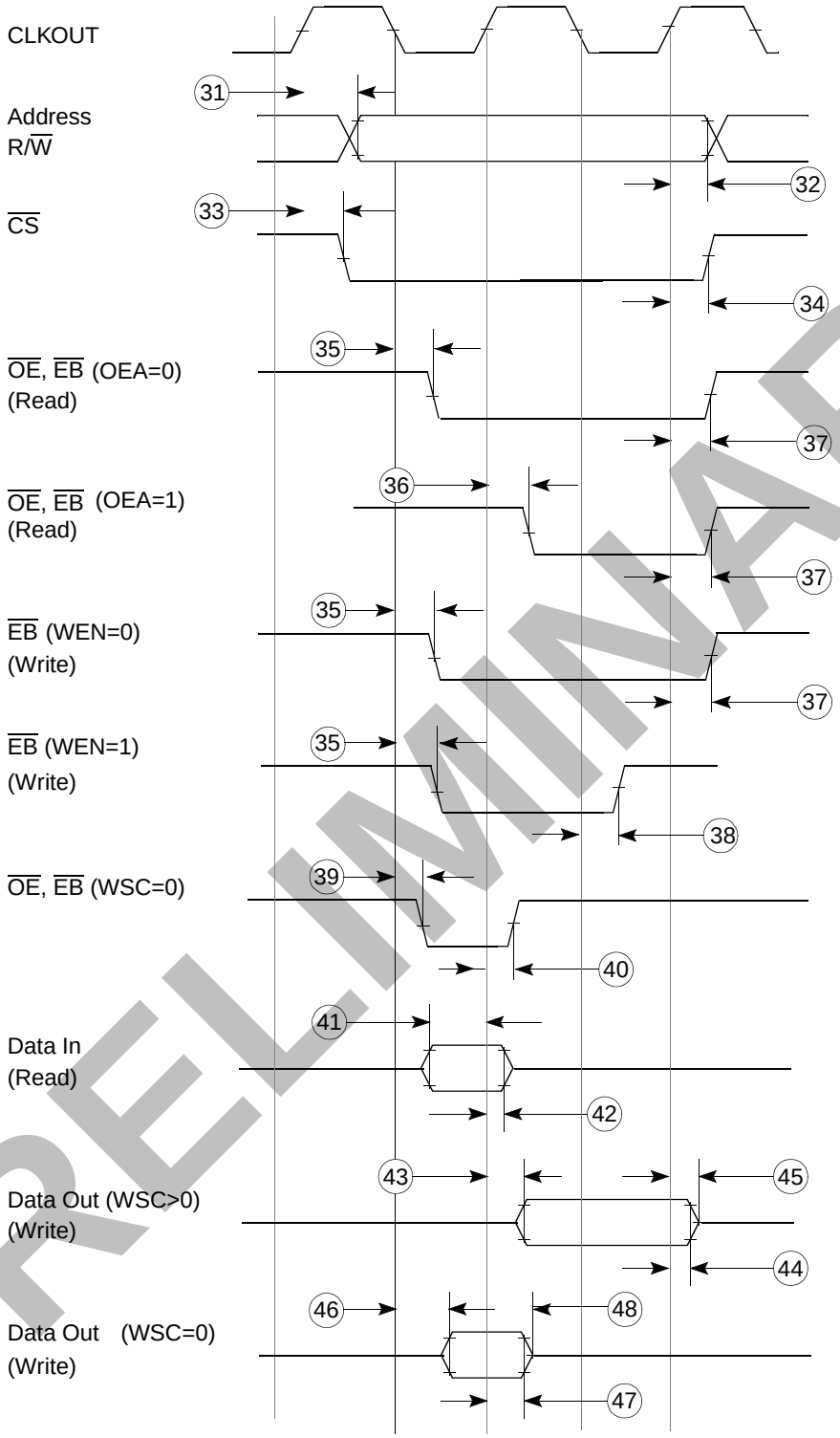


Figure A-5 EIM Read/Write Timing

Freescale Semiconductor, Inc.
PRELIMINARY

For More Information On This Product,
Go to: www.freescale.com

A.4.4 ISPI Timing Specifications

Table A-7 ISPI Timing Specifications

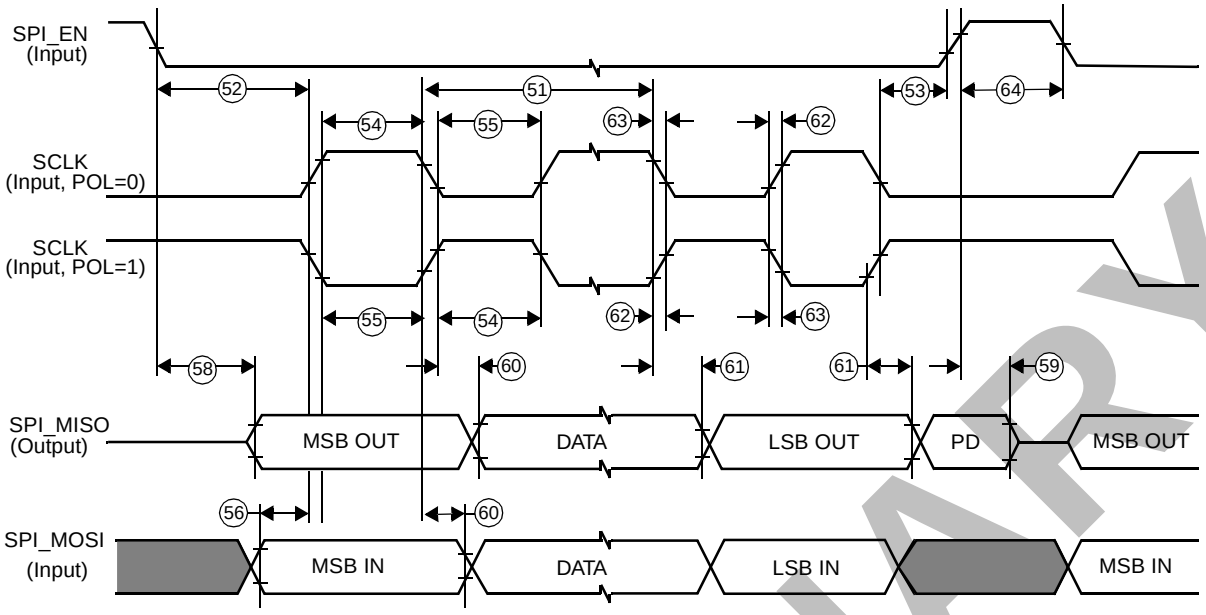
Num	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency (SCLK)	$f_{op(s)}$	DC	1/8	CLK
51	Cycle Time	t_{cyc}	8	1024	CLK
52	Enable Lead Time PHA=0 PHA=1	t_{lead}	 1 1	 — —	SCLK
53	Enable Lag Time PHA=0 PHA=1	t_{lag}	 1 1	 1 1	SCLK
54	Clock (SCLK) High Time	$t_{w(SCKH)}$	4	512	CLK
55	Clock (SCLK) Low Time	$t_{w(SCKL)}$	4	512	CLK
56	Data Setup Time (Inputs)	t_{su}	8	—	ns
57	Data Hold Time (Inputs)	t_h	8	—	ns
58	Access Time Slave	t_a	1	—	SCLK
59	Disable Time (Hold Time)	t_{dis}	—	1	SCLK
60	Data Valid (After Enable Edge)	$t_{v(s)}$	—	5	ns
61	Data Hold Time (Output) (After Enable Edge)	t_{ho}	0	—	ns
62	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 20pF) Manual/Interval Mode Slave Mode	t_{rs}	 — —	 10 10	ns
63	Fall Time (20% V_{DD} to 70% V_{DD}, C_L = 20pF) Manual/Interval Mode Slave Mode	t_{fs}	 0 0	 10 10	ns
64	Sequential Transfer Delay[1]	t_{bt}	1	—	SCLK

NOTES:
1. Signal depends on software in interval mode.

PRELIMINARY

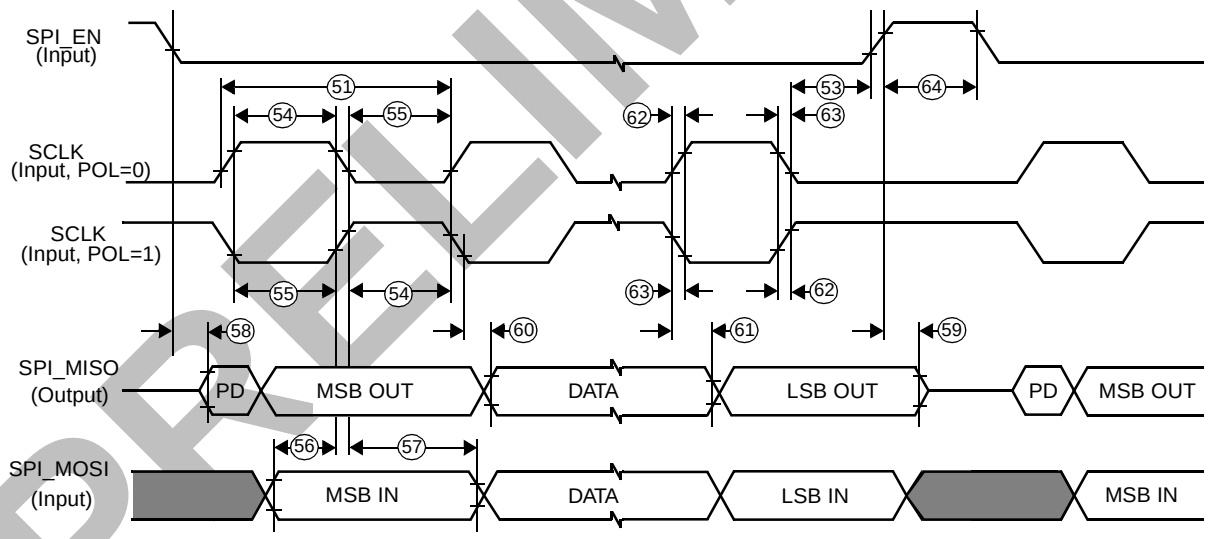
Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com



NOTE: PD — Port data is not defined, but normally LSB of character previously transmitted (likely not to be LSB in slave mode).

Figure A-6 SPI Slave Timing (PHA = 0)

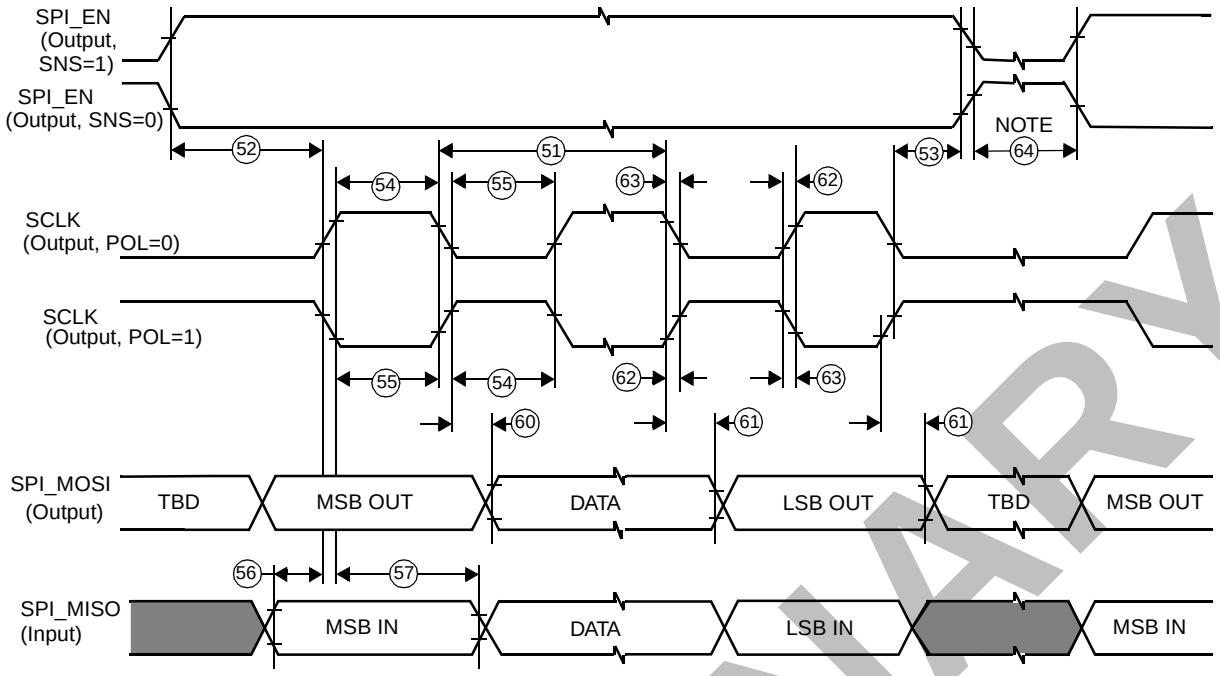


NOTE: PD — Port data is not defined, but normally LSB of character previously transmitted (likely not to be LSB in slave mode).

Figure A-7 SPI Slave Timing (PHA = 1)

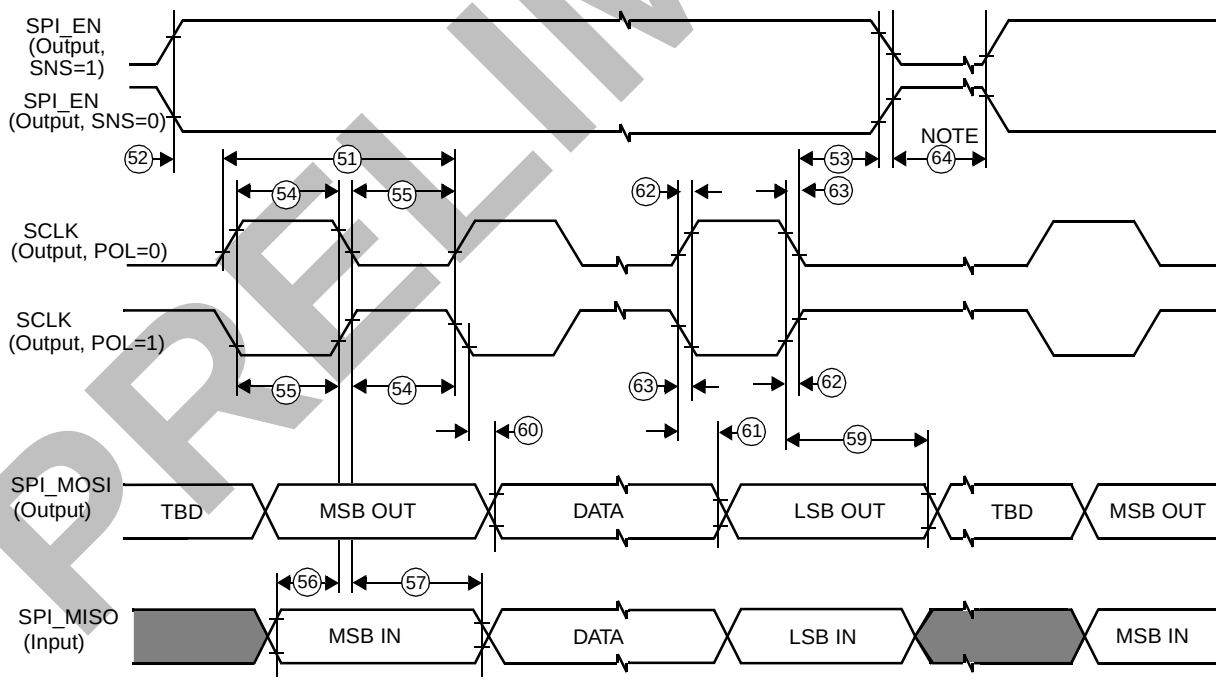
Freescale Semiconductor, Inc.
PRELIMINARY

For More Information On This Product,
Go to: www.freescale.com



NOTE: The sequential transfer delay is determined by the settings in the SPI interval control register.

Figure A-8 SPI Manual/Interval Mode Timing (PHA = 0)



NOTE: The sequential transfer delay is determined by the settings in the SPI interval control register.

Figure A-9 SPI Manual/Interval Mode Timing (PHA = 1)

PRELIMINARY
Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

A.4.5 OnCE Timing Specifications

Table A-8 OnCE Timing Specifications

Num	Characteristic	Min	Max	Unit
	TCK Frequency of Operation	0	CLK/2	MHz
92	TCK Clock Pulse Width Measured at 1.5 V	20	—	ns
93	TCK Rise and Fall Times	0	8	ns
94	TMS, TDI Data Setup Time	7	—	ns
95	TMS, TDI Data Hold Time	25	—	ns
96	TCK Low to TDO Data Valid	0	44	ns
97	TCK Low to TDO High Z	0	44	ns
98	$\overline{\text{TRST}}$ Assert Time	100	—	ns
99	$\overline{\text{TRST}}$ Setup Time to TCK Low	40	—	ns

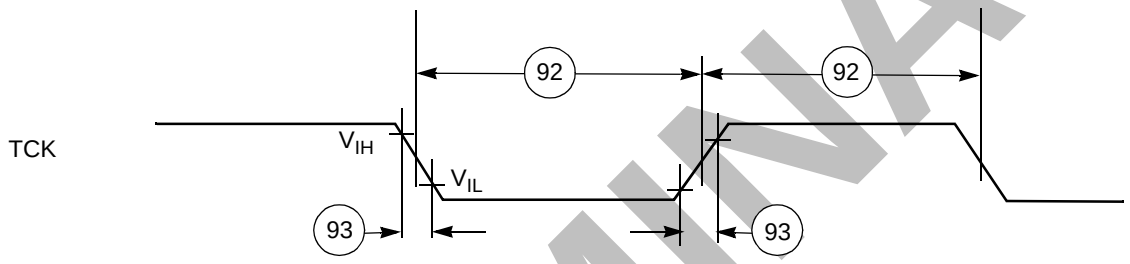


Figure A-10 Test Clock Input Timing

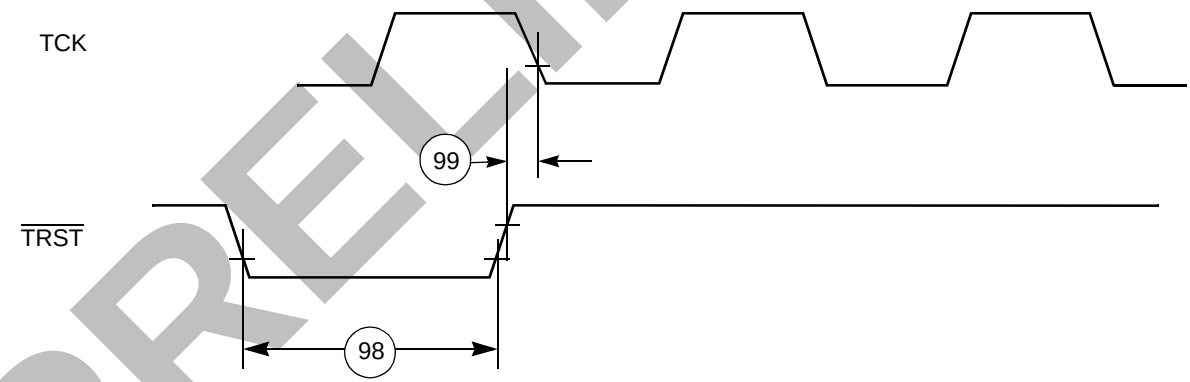


Figure A-11 $\overline{\text{TRST}}$ Timing

Freescale Semiconductor, Inc.
PRELIMINARY

For More Information On This Product,
Go to: www.freescale.com

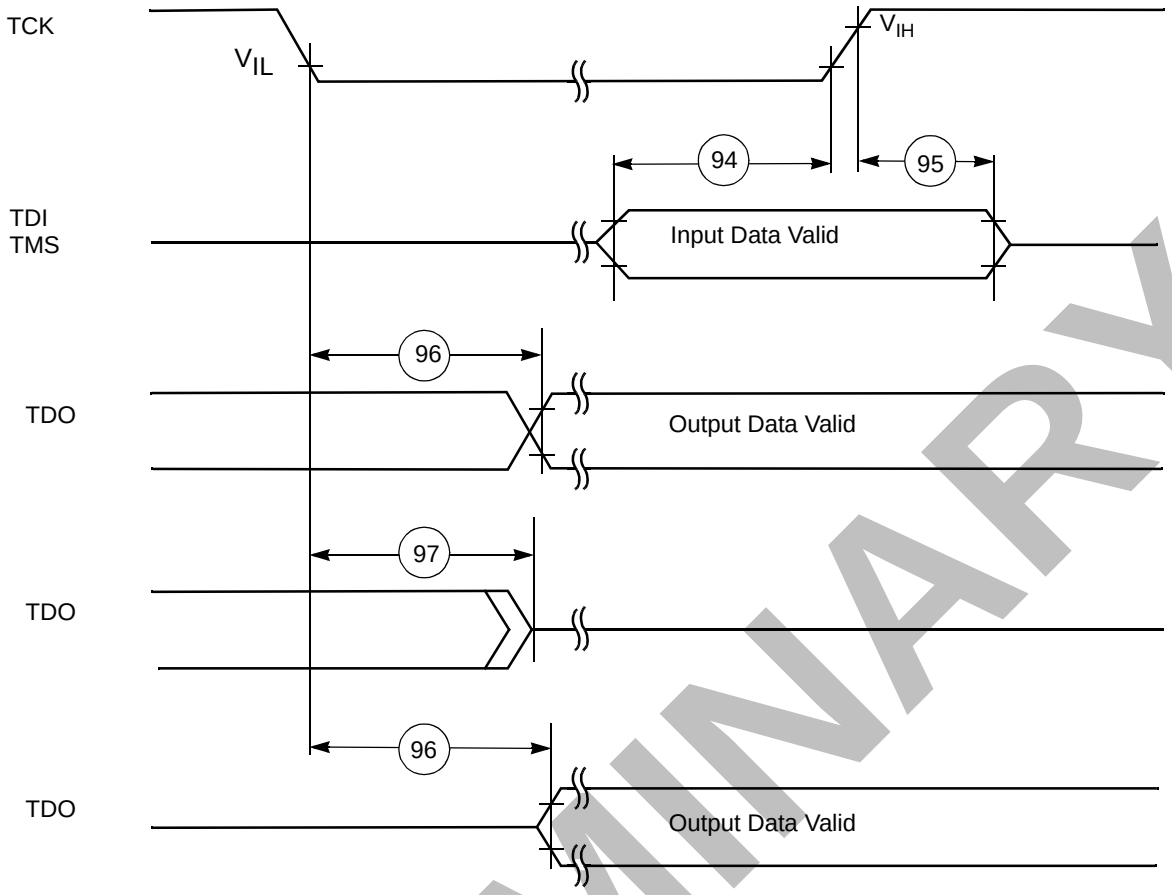


Figure A-12 Test Access Port Timing

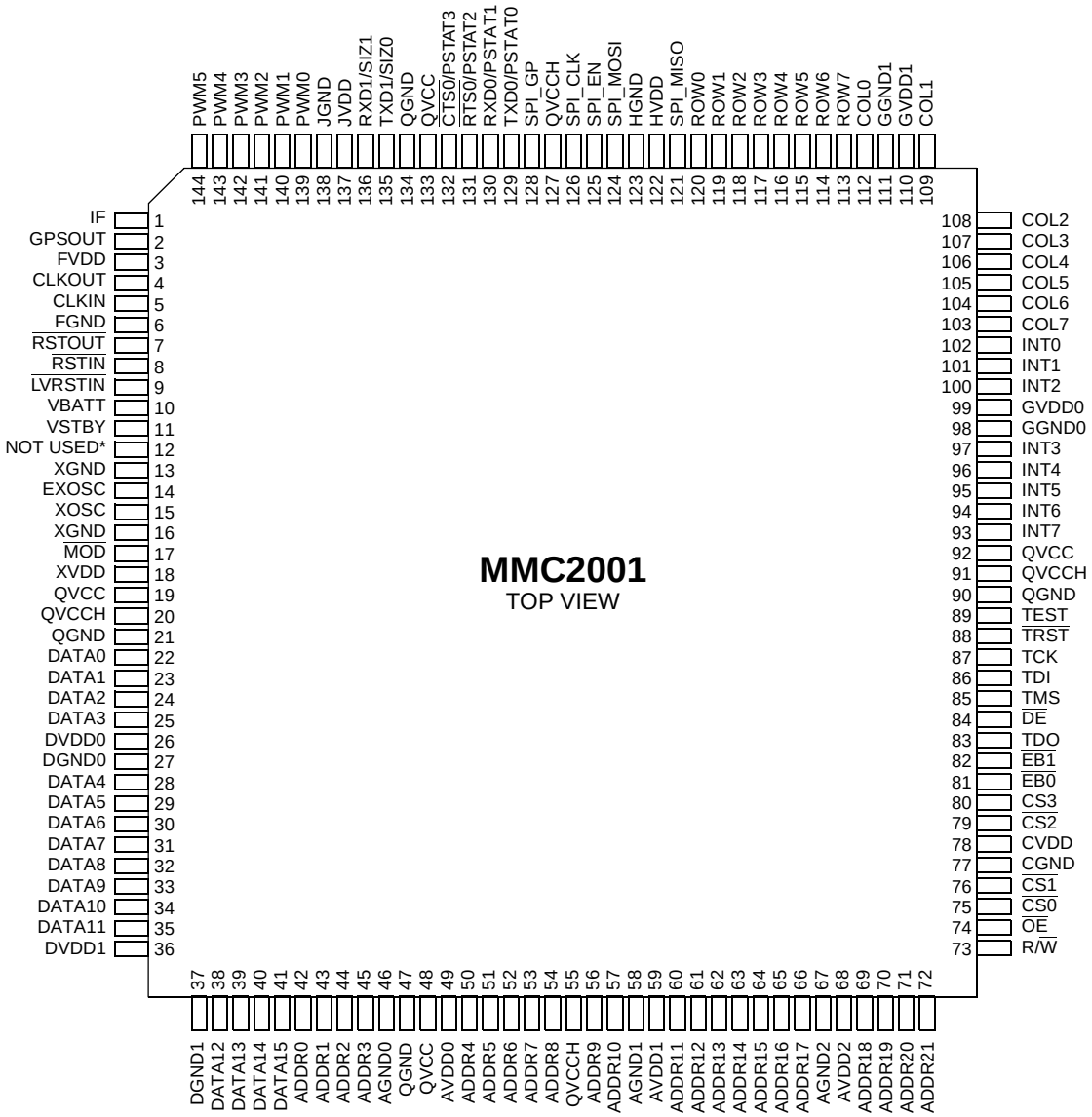
Freescale Semiconductor, Inc.
PRELIMINARY

For More Information On This Product,
Go to: www.freescale.com

APPENDIX B
PACKAGING AND PIN ASSIGNMENTS

B.1 Overview

The following diagram shows the pin assignments of the MMC2001.



*Pin 12 must be grounded for proper operation but is NOT a system ground.

Figure B-1 144-Lead Plastic Thin Quad Flat Pack Pin Assignment

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

APPENDIX C
PROGRAMMING REFERENCE

C.1 Peripheral Module Address Assignment

The register maps of all peripheral devices for MMC2001 are located on 4096-byte boundaries. **Table C-1** defines the address assignment for the on-chip components.

Table C-1 MMC2001 Address Map

Address Range (Hex)	Use	Access
00000000 – 0003FFFF	On-Chip ROM Array	Supervisor, Selective User
00040000 – 000FFFFF	ROM Echoes	Supervisor, Selective User
00100000 – 0FFFFFFF	Not Used (Access causes transfer error)	—
10000000 – 10000FFF	Interrupt Controller	Supervisor Only
10001000 – 10001FFF	Timer/Reset Unit	Supervisor Only
10002000 – 10002FFF	Not Used (Access causes transfer error)	—
10003000 – 10003FFF	Keypad Port	Supervisor Only
10004000 – 10004FFF	External Interface Module	Supervisor Only
10005000 – 10005FFF	Pulse-width Modulator	Supervisor Only
10006000 – 10006FFF	Not Used (Access causes transfer error)	—
10007000 – 10007FFF	GPIO Edge Port	Supervisor Only
10008000 – 10008FFF	Interval SPI	Supervisor Only
10009000 – 10009FFF	UART 0	Supervisor Only
1000A000 – 1000AFFF	UART 1	Supervisor Only
1000B000 – 1FFFFFFF	Not Used (Access causes transfer error)	—
20000000 – 2FFFFFFF	External Devices	Supervisor, Selective User
30000000 – 30007FFF	On-Chip RAM Array	Supervisor, Selective User
30008000 – 3000FFFF	RAM Echoes	Supervisor, Selective User
30100000 – 40000000	Not Used (Access causes transfer error)	—

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

C.2 Interrupt Controller Programming Model

Control and status registers for the interrupt controller begin at address 0x40002000.

Table C-2 Interrupt Controller Address Map

Address	Use	Access
10000000	Interrupt Source Register (INTSRC)	Supervisor Only
10000004	Normal Interrupt Enable Register (NIER)	Supervisor Only
10000008	Fast Interrupt Enable Register (FIER)	Supervisor Only
1000000C	Normal Interrupt Pending Register (NIPND)	Supervisor Only
10000010	Fast Interrupt Pending Register (FIPND)	Supervisor Only

C.2.1 Interrupt Source Register (INTSRC)

Access the 32-bit interrupt source register with 32-bit loads only.

INTSRC — Interrupt Source Register **10000000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IN31	IN30	IN29	IN28	IN27	IN26	IN25	IN24	IN23	IN22	IN21	IN20	IN19	IN18	IN17	IN16
RESET:															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN15	IN14	IN13	IN12	IN11	IN10	IN9	IN8	IN7	IN6	IN5	IN4	IN3	1	1	1
RESET:															

Figure C-1 Interrupt Source Register

INx — Interrupt Source x

This bit indicates the state of the corresponding interrupt source.

0 = Negated
1 = Asserted

Bits [0:2] of this register are tied to logic level one to allow software to schedule interrupts by enabling one or more of these “sources” in the appropriate interrupt enable register(s) (NIER, FIER).

C.2.2 Normal Interrupt Enable Register (NIER)

Access the 32-bit normal interrupt enable register with 32-bit loads and stores only.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

NIER — Normal Interrupt Enable Register **10000004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-2 Normal Interrupt Enable Register

ENx — Enable Normal Interrupt Flag x

This bit enables the corresponding interrupt source to request a normal interrupt.

0 = Disable
1 = Enable

A reset operation clears this bit.

When the enable flag is set and the corresponding interrupt line is asserted, the interrupt controller asserts a normal interrupt request. Enabling an interrupt source which has an asserted request causes that interrupt to become pending, and a request to the CPU is asserted if not already outstanding.

C.2.3 Fast Interrupt Enable Register (FIER)

Access the 32-bit fast interrupt enable register with 32-bit loads and stores only.

FIER — Fast Interrupt Enable Register **10000008**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EF31	EF30	EF29	EF28	EF27	EF26	EF25	EF24	EF23	EF22	EF21	EF20	EF19	EF18	EF17	EF16
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EF15	EF14	EF13	EF12	EF11	EF10	EF9	EF8	EF7	EF6	EF5	EF4	EF3	EF2	EF1	EF0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-3 Fast Interrupt Enable Register

EFx — Enable Fast Interrupt Flag x

This bit enables the corresponding interrupt source to request a fast interrupt.

0 = Disable
1 = Enable

A reset operation clears this bit.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

When the enable flag is set and the corresponding interrupt line is asserted, the interrupt controller asserts a fast interrupt request. Enabling an interrupt source that has an asserted request causes that interrupt to become pending, and a request to the CPU is asserted if not already outstanding.

C.2.4 Normal Interrupt Pending Register (NIPND)

Access the 32-bit normal interrupt pending register with 32-bit loads only.

NIPND — Normal Interrupt Pending Register **1000000C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NP31	NP30	NP29	NP28	NP27	NP26	NP25	NP24	NP23	NP22	NP21	NP20	NP19	NP18	NP17	NP16
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NP15	NP14	NP13	NP12	NP11	NP10	NP9	NP8	NP7	NP6	NP5	NP4	NP3	NP2	NP1	NP0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-4 Normal Interrupt Pending Register

NPx — Normal Interrupt Pending Flag x

This bit indicates a pending normal interrupt request from the corresponding interrupt source.

0 = No request
1 = Interrupt request pending

When a normal interrupt enable flag is set and the corresponding interrupt line is asserted, the interrupt controller asserts a normal interrupt request. The normal interrupt pending flags reflect the interrupt input lines which are asserted and are currently enabled to generate a normal interrupt.

C.2.5 Fast Interrupt Pending Register (FIPND)

Access the 32-bit read-only fast interrupt pending register with 32-bit loads only.

FIPND — Fast Interrupt Pending Register **10000010**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FP31	FP30	FP29	FP28	FP27	FP26	FP25	FP24	FP23	FP22	FP21	FP20	FP19	FP18	FP17	FP16
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FP15	FP14	FP13	FP12	FP11	FP10	FP9	FP8	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-5 Fast Interrupt Pending Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

FPx — Fast Interrupt Pending Flag x

This bit indicates a pending fast interrupt request from the corresponding interrupt source.

0 = No request
1 = Interrupt request pending

When a fast interrupt enable flag is set and the corresponding interrupt line is asserted, the interrupt controller will assert a fast interrupt request ($\overline{\text{FINT}}$ CPU input). The fast interrupt pending flags reflect the interrupt input lines which are currently enabled to generate a fast interrupt and are asserted.

C.3 Timer/Reset Programming Model

Table C-3 shows the timer/reset module address map.

Table C-3 Timer/Reset Module Address Map

Address	Use	Access
10001000	Reset Source/Chip Configuration Register (RSCR)	Supervisor Only
10001004	Time-of-Day Control/Status Register (TODCSR)	Supervisor Only
10001008	Time-of-Day Seconds Register (TODSR)	Supervisor Only
1000100C	Time-of-Day Fraction Register (TODFR)	Supervisor Only
10001010	Time-of-Day Seconds Alarm Register (TODSAR)	Supervisor Only
10001014	Time-of-Day Fraction Alarm Register (TODFAR)	Supervisor Only
10001018	Reserved	Supervisor Only
1000101C	Watchdog Control Register (WCR)	Supervisor Only
10001020	Watchdog Service Register (WSR)	Supervisor Only
10001024	Interval Timer Control/Status Register (ITCSR)	Supervisor Only
10001028	PIT Data Register (ITDR)	Supervisor Only
1000102C	PIT Alternate Data Register (ITADR)	Supervisor Only
1000102E to 10001FFF	Reserved	Supervisor Only
10002000 to 10002FFF	Not Used (Access causes transfer error)	Not Applicable

C.3.1 Reset Source/Chip Configuration Register (RSCR)

This status and control register gives the state of the reset sources and serves to control the CLKOUT pin. Writes to this register clear any previously set status bits.

Access this register with 32-bit loads and stores only.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

RSCR — Reset Source/Chip Configuration Register **10001000**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	CKOS	CKOE	0	0	0	0	LVRSTIN	RST	POR	WDR
W																
RESET:							0*	0*					*	*	*	0*

* = See bit description

Figure C-6 Reset Source Register

CKOS — CLKOUT Source

This bit controls the clock source for the CLKOUT pin. Modify this pin only when the clock output has been disabled. This bit is cleared by POR, a qualified assertion of the $\overline{\text{RSTIN}}$ pin, or a qualified assertion of the $\overline{\text{LVRSTIN}}$ pin.

0 = CLKOUT source is HI_REFCLK
1 = CLKOUT source is LOW_REFCLK

CKOE — CLKOUT Enable

This bit controls the drive enable for the CLKOUT pin. It is cleared by POR, a qualified assertion of the $\overline{\text{RSTIN}}$ pin, or a qualified assertion of the $\overline{\text{LVRSTIN}}$ pin.

0 = CLKOUT is disabled and forced to the low state
1 = CLKOUT is enabled and driven from the source selected by CKOS

LVRSTIN — $\overline{\text{LVRSTIN}}$ Pin

This bit is set when the $\overline{\text{LVRSTIN}}$ pin is asserted to reset the MMC2001. It is not affected by the other reset sources. When the POR bit is set, however, this bit is undefined. This bit is cleared by writing to RSCR.

RST — $\overline{\text{RSTIN}}$ Pin

This bit is set when the $\overline{\text{RSTIN}}$ pin is asserted and qualified by the four-cycle qualifier to reset the MMC2001. It is not affected by the other reset sources. When the POR bit is set, however, this bit is undefined. It is cleared by writing to RSCR.

POR — Power-On Reset

This bit is set when an internal POR occurs to reset the chip. It is not affected by the other reset sources. It is cleared by writing to RSCR.

WDR — Watchdog Reset

This bit is set when the watchdog timer expires. It is cleared by POR, a qualified assertion of the $\overline{\text{RSTIN}}$ pin, a qualified assertion of the $\overline{\text{LVRSTIN}}$ pin, or by writing to RSCR.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

C.3.2 Time-of-Day Control/Status Register (TODCSR)

TODCSR — Time-of-Day Control/Status Register **10001004**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	AE	AIE	AIF
W																
RESET:														0	0	0

Figure C-7 TOD Control/Status Register

Access this register with 32-bit loads and stores only.

AE — Alarm Enable

This bit controls the function of the TOD alarm

0 = Alarm function is off to save power
1 = Alarm function is on

AIE — Alarm Interrupt Enable

This bit controls the alarm interrupt function

0 = AIF is inhibited from reaching the CPU
1 = AIF is allowed to request an interrupt

AIF — Alarm Interrupt Flag

This read-only bit is the alarm interrupt flag. It is cleared by writing to the TOD fraction alarm register (TODFAR).

0 = No alarm interrupt is present
1 = Alarm interrupt is present

C.3.3 TOD Seconds Register (TODSR)

The time-of-day seconds register is a 32-bit read/write register that holds the number of elapsed seconds. It is clocked by a 1-Hz signal generated as a carry from the TOD fraction register. When TODSR is read, the content of the fraction counter is latched into a holding buffer to be read later. This prevents a fraction rollover between reads of the two registers from causing incorrect data to be read. When TODSR is written, the TODFR is cleared to all zeros. TODSR is not affected by the watchdog reset or by a reset initiated by the external reset signal, but is undefined after a POR.

Access this register with 32-bit loads and stores only.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

TODSR — Time-of-Day Seconds Register **10001008**

	31															0
R	TOD SECONDS REGISTER															
W																
RESET:	Undefined on POR															

Figure C-8 TOD Seconds Register

C.3.4 TOD Fraction Register (TODFR)

The 32-bit time-of-day fraction register holds eight bits of data that represent the binary fraction of a second. It is clocked by the 32.768-kHz LOW_REFCLK divided by 128 (256 Hz). Reads of this register return the value latched when the TOD seconds register was previously read. Continuous reads return the same value until the TODSR is read and new data is latched from the fraction counter. These eight bits are cleared whenever the TODSR is written but are not affected by either reset pin or the watchdog reset conditions. The fraction counter is undefined after a POR. Writes to this register cause all eight bits to be set.

Access this register with 32-bit accesses only.

TODFR — TOD Fraction Register **1000100C**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TOD FRACTION REGISTER								0	0	0	0	0	0	0	0
W	Set to ones															
RESET:	Undefined on POR															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

Figure C-9 TOD Fraction Register

C.3.5 TOD Seconds Alarm Register (TODSAR)

The time-of-day seconds alarm register is a 32-bit read/write register which holds data (in seconds) to be compared to the TOD seconds register. The comparison is made every 1/256 of a second if the alarm function is enabled by the AE bit in the TODCSR. Writes to this register inhibit alarm compares until the TODFAR is written. For proper alarm operation, the fraction alarm register must be (re)written after a write to this register. This register is not affected by any of the reset conditions.

Access this register with 32-bit loads and stores only.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

TODSAR — Time-of-Day Seconds Alarm Register **10001010**

	31															0
R	TOD SECONDS ALARM REGISTER															
W																
RESET:	Unaffected															

Figure C-10 TOD Seconds Alarm Register

C.3.6 TOD Fraction Alarm Register (TODFAR)

The time-of-day fraction alarm register is a 32-bit read/write register which holds eight bits of data to be compared to the TOD fraction register. The comparison is made every 1/256 of a second if the alarm function is enabled by the AE bit in the TODCSR. This register is not affected by any of the reset conditions.

Access this register with 32-bit loads and stores only.

TODFAR — TOD Fraction Alarm Register **10001014**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TOD FRACTION ALARM REGISTER								0	0	0	0	0	0	0	0
W																
RESET:	Undefined on POR															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

Figure C-11 TOD Fraction Alarm Register

C.3.7 Watchdog Control Register (WCR)

This register contains fields that control the operation of the watchdog in different modes of operation. The write-once bits can only be written once after a reset condition. Subsequent attempts to write to them will not affect the data previously written.

Access this register with 32-bit loads and stores only.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

WCR — Watchdog Control Register **1000101C**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WT						0	0	0	0	0	0	WSTP	WDE	WDBG	WDZE
W																
RESET:	0	0	0	0	0	0							0	0	0	0

Figure C-12 Watchdog Control Register

WT — Watchdog Time-Out

The six-bit WT field contains the time-out value. These bits are reloaded into the watchdog timer when it has been serviced. After reset, write WT before enabling the watchdog. The value in WT is loaded into the watchdog counter after running the service routine as well as on enabling the watchdog timer.

WSTP — Watchdog Stop Enable (one-time writable)

0 = Watchdog not affected while in stop mode
1 = Watchdog disabled while in stop mode

WDE — Watchdog Enable (one-time writable)

0 = Watchdog is disabled
1 = Watchdog is enabled (once set, watchdog cannot be disabled)

WDBG — Watchdog Debug Enable (one-time writable)

0 = Watchdog not affected while in debug mode
1 = Watchdog disabled while in debug mode

WDZE — Watchdog Doze Enable (one-time writable)

0 = Watchdog not affected while in doze mode
1 = Watchdog disabled while in doze mode

C.3.8 Watchdog Service Register (WSR)

When enabled, the watchdog requires that a service sequence be written to the watchdog service register (WSR). This register controls the clearing of the watchdog counter to keep it from timing out and causing a reset. If this register is not written with 0x5555 followed by 0xAAAA before the selected rate expires, the watchdog sets the WDR bit in the reset source register and asserts a system reset.

Both writes must occur in the order listed prior to the time-out, but any number of instructions can be executed between the two writes.

Access this register with 32-bit loads and stores only.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

WSR — Watchdog Service Register **10001020**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	WATCHDOG SERVICE REGISTER															
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-13 Watchdog Service Register

C.3.9 PIT Control/Status Register (ITCSR)

ITCSR — Interval Timer Control and Status Register **10001024**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	STOP	DOZE	DBG	OVW	ITIE	ITIF	RLD	EN
W																
RESET:									0	0	0	0	0	0	0	0

Figure C-14 PIT Control and Status Register

Access this register with 32-bit loads and stores only.

STOP — Stop Mode Control

This bit controls the function of the PIT in stop mode

0 = PIT function is not affected while in stop mode
1 = PIT function is frozen while in stop mode

DOZE — Doze Mode Control

This bit controls the function of the PIT in doze mode

0 = PIT function is not affected while in doze mode
1 = PIT function is frozen while in doze mode

DBG — Debug Mode Control

This bit controls the function of the PIT in debug mode

0 = PIT function is not affected while in debug mode
1 = PIT function is frozen while in debug mode

For More Information On This Product, Go to: www.freescale.com

Freescale Semiconductor, Inc.

OVW — Counter Overwrite Enable

This bit controls what happens to the counter value when the modulus latch is written.

0 = Modulus latch is a holding register for values to be loaded into the counter when the count expires to zero.
1 = Modulus latch is transparent. All writes to the latch will also overwrite the counter contents

ITIE — PIT Interrupt Enable

This bit controls the PIT interrupt function.

0 = ITIF is inhibited from reaching the CPU
1 = ITIF is allowed to request an interrupt

ITIF — PIT Interrupt Flag

This bit is the PIT interrupt flag. It is cleared by writing a one to this bit or by writing to the PIT data register.

0 = No PIT interrupt present.
1 = PIT interrupt is present.

RLD — Counter Reload Control

This bit controls whether the value contained in the modulus latch is reloaded into the counter when the counter reaches a count of zero or whether the counter rolls over from 0 to 0xFFFF.

0 = Counter rolls over to 0xFFFF.
1 = Counter is reloaded from the modulus latch.

EN — PIT Enable

This bit controls the PIT enable function.

0 = PIT is disabled
1 = PIT is enabled

C.3.10 PIT Data Register (ITDR)

On a write, the data becomes the new timer modulus. This value is retained and is used at the next and all subsequent reloads until changed by another write to ITDR. This value is initialized to the maximum count of 0xFFFF on reset. On a read, the ITDR returns the value written in the modulus latch. The only way to directly change the value of the count is to preload a modulus with the OVW bit set to one. The counter value can be read from the PIT alternate data register.

Access this register with 32-bit loads and stores only.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

ITDR — PIT Data Register **10001028**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIT DATA															
W																
RESET:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure C-15 PIT Data Register

C.3.11 PIT Alternate Data Register (ITADR)

The PIT alternate data register is a read-only register which provides access to the counter value. Access this register with 32-bit loads and stores only.

ITADR — PIT Alternate Data Register **1000102C**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIT COUNTER															
W																
RESET:																

Figure C-16 PIT Alternate Data Register

C.4 KPP Programming Model

Table C-4 Keypad Port Address Map

Address	Use	Access
10003000	Keypad Control Register (KPCR)	Supervisor Only
10003002	Keypad Status Register (KPSR)	Supervisor Only
10003004	Keypad Data Direction Register (KDDR)	Supervisor Only
10003006	Keypad Data Register (KPDR)	Supervisor Only
10003008 to 10003FFF	Reserved	Supervisor Only

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

C.4.1 Keypad Control Register (KPCR)

The keypad control register (KPCR) determines which of the eight possible column strobes are to be open drain when configured as outputs and which of the eight row sense lines are considered in generating an interrupt to the core.

The KPCR register is byte or halfword addressable.

KPCR — Keypad Control Register **10003000**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KCO7	KCO6	KCO5	KCO4	KCO3	KCO2	KCO1	KCO0	KRE7	KRE6	KRE5	KRE4	KRE3	KRE2	KRE1	KRE0
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-17 Keypad Control Register

KCOx — Keypad Column Strobe Open-Drain Enable x

0 = Column strobe output x is totem-pole drive (P-channel enabled).
1 = Column strobe output x is open drain (P-channel disabled).

KREx — Keypad Row Enable x

0 = Row x is not included in keypad key press detect.
1 = Row x is included in keypad key press detect.

C.4.2 Keypad Status Register (KPSR)

The keypad status register (KPSR) reflects the state of the keypress detect circuit.

The KPSR register is byte or halfword addressable.

KPSR — Keypad Status Register **10003002**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	KRIE	KDIE	0	0	0	0	0	0	KPKR	KPKD
W													KRSS	KDSC		
RESET:							0	0					0	0	0	0

Figure C-18 Keypad Status Register

KRIE — Key Release Interrupt Enable

0 = No interrupt request is generated when KPKR is set
1 = An interrupt request is generated when KPKR is set

KDIE — Key Depress Interrupt Enable

0 = No interrupt request is generated when KPKD is set
1 = An interrupt request is generated when KPKD is set

KRSS — Key Release Synchronizer Set

The key release synchronizer is set by writing a logic one into this bit. Reads return a value of zero.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

KDSC — Key Depress Synchronizer Clear

The key depress synchronizer is cleared by writing a logic one into this bit. Reads return a value of zero.

KPKR — Keypad Key Release

0 = No key release detected
1 = All keys have been released

KPKR is cleared by writing a logic one into this bit.

KPKD — Keypad Key Depress

0 = No key presses detected
1 = A key has been depressed

KPKD is cleared by writing a logic one into this bit.

C.4.3 Keypad Data Direction Register (KDDR)

The bits in the keypad data direction register (KDDR) control the direction of the keypad port pins.

The KDDR register is byte or halfword addressable.

KDDR — Keypad Data Direction Register **10003004**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KCDD7	KCDD6	KCDD5	KCDD4	KCDD3	KCDD2	KCDD1	KCDD0	KRDD7	KRDD6	KRDD5	KRDD4	KRDD3	KRDD2	KRDD1	KRDD0
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-19 Keypad Data Direction Register

KCDDx — Keypad Column x Data Direction

0 = COLx pin is configured as input.
1 = COLx pin is configured as output.

KRDDx — Keypad Row x Data Direction

0 = ROWx pin is configured as input.
1 = ROWx pin is configured as output.

C.4.4 Keypad Data Register (KPDR)

The 16-bit keypad data register is used to access the column and row data. Data written to this register is stored in an internal latch, and for each pin configured as an output, the stored data is driven onto the pin. A read of this register returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register.

The KPDR register is byte or halfword addressable.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

KPDR — Keypad Data Register **10003006**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KCD7	KCD6	KCD5	KCD4	KCD3	KCD2	KCD1	KCD0	KRD7	KRD6	KRD5	KRD4	KRD3	KRD2	KRD1	KRD0
RESET:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

* Since pins default to inputs, reset value is determined by the logic level present on the pins at reset.

Figure C-20 Keypad Data Register

KCDx — Keypad Column x Data Bit

KRDx — Keypad Row x Data Bit

This register is not initialized by reset. Valid data should be written to this register before any bits are configured as outputs.

C.5 EIM Programming Model

Table C-5 lists the registers in the EIM.

Table C-5 EIM Address Map

Address	Use	Access
10004000	$\overline{CS0}$ Control Register (CS0CR)	Supervisor Only
10004004	$\overline{CS1}$ Control Register (CS1CR)	Supervisor Only
10004008	$\overline{CS2}$ Control Register (CS2CR)	Supervisor Only
1000400C	CS3 Control Register (CS3CR)	Supervisor Only
10004010 to 10004014	Reserved	Supervisor Only
10004018	EIM Configuration Register (EIMCR)	Supervisor Only
10004020 to 10004FFF	Reserved	Supervisor Only

C.5.1 Chip-Select Control Registers

Each of the external chip selects has an enable bit as well as other control attributes. The layout of the control register is slightly different for the $\overline{CS0}$ output, which does not support the programmable output function. For CS1–CS3 control registers, bits two to 15 (i.e., bits other than the PA and CSEN bits) are undefined at reset.

Access these registers with 32-bit loads and stores only.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

CS0CR — $\overline{\text{CS0}}$ Control Register **10004000**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WSC				WWS	EDC	CSA	OEA	WEN	EBC	DSZ		SP	WP	0	CSEN
W																
RESET:	1	1	1	1	1	0	0	0	0	1	1	0	0	0		1

Figure C-21 $\overline{\text{CS0}}$ Control Register

CS1CR — $\overline{\text{CS1}}$ Control Register **10004004**
CS2CR — $\overline{\text{CS2}}$ Control Register **10004008**
CS3CR — CS3 Control Register **1000400C**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WSC				WWS	EDC	CSA	OEA	WEN	EBC	DSZ		SP	WP	PA	CSEN
W																
RESET:	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0/1*	0

X = Undefined
* PA reset value equals zero for CS3 and one for $\overline{\text{CS[1:2]}}$

Figure C-22 $\overline{\text{CS1}}$, $\overline{\text{CS2}}$, CS3 Control Registers

WSC — Wait-State Control

These four bits program the number of wait states for an access to the external device connected to the chip select. **Table C-6** shows the encoding of this field. When WWS is cleared, setting WSC=0000 results in 1-clock transfers, WSC=0001 results in 2-clock transfers, and WSC=1111 results in 16-clock transfers. When WSC=0000, the WEN and CSA bits are ignored.

Set WSC=0000 and WWS=0 for access to fast SRAM devices (one-clock read and write access), CSA=0, WSC=0001 and WWS=0 for access to normal SRAM (two-clock read and write access), CSA=0, WSC=0001 and WWS=1 for access to Flash memory (two-clock read access and three-clock write access), EDC, CSA and WSC to the appropriate number for access to an LCD controller.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table C-6 Wait State Control Field Settings

WSC[3:0]	Number of Wait States			
	WWS = 0		WWS = 1	
	Read Access	Write Access	Read Access	Write Access
0000	0	0	0	1
0001	1	1	1	2
0010	2	2	2	3
0011	3	3	3	4
0100	4	4	4	5
0101	5	5	5	6
0110	6	6	6	7
0111	7	7	7	8
1000	8	8	8	9
1001	9	9	9	10
1010	10	10	10	11
1011	11	11	11	12
1100	12	12	12	13
1101	13	13	13	14
1110	14	14	14	15
1111	15	15	15	15

WWS — Write Wait State

This bit is used to determine if an additional wait state is required for write cycles. This is useful for writing to Flash memories that require additional data setup time.

0 = Reads and writes are the same length.

1 = An additional wait state is inserted for write cycles unless WSC is set to 1111. Setting WSC to 1111 results in 16-clock transfers regardless of the WWS bit. Read cycles are not affected.

EDC — Extra Dead Cycle

This bit is used to determine if an idle cycle is inserted after a read cycle for back-to-back external transfers to eliminate data bus contention. This is useful for slow memory and peripherals.

0 = Back-to-back external transfers occur normally, i.e., no idle cycle is inserted after a read cycle.

1 = An idle cycle is inserted after a read cycle for back-to-back external transfers, unless the next cycle is a read cycle to the same $\overline{CS}$ bank.

CSA — Chip Select Assert

This bit is used for devices that require additional address setup time and additional address/data hold times. It determines when the chip select is asserted and whether an idle cycle is inserted between back-to-back external transfers. If WSC=0000, this bit is ignored.

0 = Chip select is asserted normally, i.e., as early as possible. No idle cycle is inserted between back-to-back external transfers.

1 = Chip select is asserted a clock later during both read and write cycles. In addition, an idle cycle is inserted between back-to-back external transfers.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

OEA — $\overline{OE}$ Assert

This bit is used to determine when $\overline{OE}$ is asserted during a read cycle. If WSC=0000, this bit is ignored and $\overline{OE}$ is asserted for one half of a clock cycle only. If EBC in the corresponding register is cleared, then the $\overline{EB}$[0:1] outputs are similarly affected.

0 = $\overline{OE}$ is asserted normally, i.e., as early as possible.
1 = $\overline{OE}$ is asserted one half of a clock cycle later during a read cycle to this chip select address space. The cycle length and write cycles are not affected.

WEN — $\overline{EB}$ Negate

This bit is used to determine when $\overline{EB}$[0:1] outputs are negated during a write cycle. This is useful to meet data hold time requirements for slow memories. If WSC=0000, this bit is ignored and $\overline{EB}$[0:1] outputs are asserted for one half of a clock cycle only.

0 = $\overline{EB}$[0:1] are negated normally, i.e., as late as possible.
1 = $\overline{EB}$[0:1] are negated one half of a clock cycle earlier during a write cycle to this chip select address space. The cycle length and read cycles are not affected.

EBC — Enable Byte Control

This bit is used to indicate which access types should assert the enable byte outputs ($\overline{EB}$[0:1]).

0 = Read and write accesses are both allowed to assert the $\overline{EB}$[0:1] outputs, thus configuring them as byte enables.
1 = Only write accesses are allowed to assert the $\overline{EB}$[0:1] outputs, thus configuring them as byte write enables. The $\overline{EB}$[0:1] outputs should be configured as byte write enables for accesses to dual x8 memories.

DSZ — Data Port Size

This field defines the width of the device data port.

Table C-7 Data Port Size Field Settings

Value	Meaning
00	8-bit port, resides on DATA[15:8] pins
01	8-bit port, resides on DATA[7:0] pins
10	16-bit port
11	Reserved

SP — Supervisor Protect

This bit is used to restrict accesses to the address range defined by the corresponding chip select if the access is attempted in the user mode of CPU operation.

0 = User mode accesses are allowed in this chip select address range
1 = User mode accesses are prohibited. An attempted access to an address mapped by this chip select in user mode will result in a $\overline{TEA}$ to the CPU and no assertion of the chip select output.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

WP — Write Protect

This bit is used to restrict writes to the address range defined by the corresponding chip select.

0 = Writes are allowed in this chip select address space.
1 = Writes are prohibited. An attempt to write to an address mapped by this chip select will result in a $\overline{\text{TEA}}$ to the CPU and no assertion of the chip select output.

PA — Pin Assert

This bit is used to control the chip select pin when it is operating as a programmable output pin (i.e. the CSEN bit clear). This bit is ignored if the CSEN bit is set. At reset, PA bit is set for $\overline{\text{CS}}$[1:2] and cleared for CS3.

0 = Brings chip select output to logic-low level
1 = Brings chip select output to logic-high level

CSEN — Chip Select Enable

This bit controls the operation of the chip select pin.

0 = Chip select function is disabled. An attempted access to an address mapped by this chip select will result in $\overline{\text{TEA}}$ assertion to the CPU and no assertion of the chip select output.

When disabled, the pin is a general-purpose output controlled by the value of the PA control bit. When CSEN0 is clear, $\overline{\text{CS0}}$ is inactive.

1 = Chip select is enabled and is asserted when an access address falls within the range specified in **Table C-8**. With the exception of $\overline{\text{CS0}}$, this bit is cleared by reset, disabling the chip select output pin.

Table C-8 Chip-Select Address Range

ADDR[31:24]	Chip Select
—	Inactive
00101101	$\overline{\text{CS0}}$
00101111	$\overline{\text{CS1}}$
00101110	$\overline{\text{CS2}}$
00101100	CS3

CSEN0 is set at reset to allow $\overline{\text{CS0}}$ to select from an external boot ROM if $\overline{\text{MOD}}$ is driven to a logic-low level four LOW_REFCLK clock cycles before $\overline{\text{RSTOUT}}$ negation.

When the chip select is enabled, the PA control bit is ignored.

C.5.2 EIM Configuration Register

The EIM configuration register contains control bits that configure the EIM and other internal blocks for certain operation modes.

Access this register with 32-bit loads and stores only.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

EIMCR — EIM Configuration Register **10004018**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	SZEN	PST EN	SP RAM	SP ROM	HDB	SHEN	
W																
RESET:												1	1	0	0	0

Figure C-23 EIM Configuration Register

SZEN — Enable SIZ signal to UART CH1 pins

This bit is used to select the function provided by the UART channel 1 pins. On reset, this bit is cleared.

0 = UART channel 0 operation. Pins function as TxD1, RxD1.
1 = SIZ function operation. Pins function as SIZ0 and SIZ1 outputs independent of function or direction programmed in UART channel 1 control registers.

PSTEN — Enable PSTAT signals to UART CH0 pins

This bit is used to select the function provided by the UART channel 0 pins. On reset, this bit is cleared.

0 = UART channel 0 operation. Pins function as TxD0, RxD0, $\overline{\text{CTS0}}$, and $\overline{\text{RTS0}}$.
1 = PSTAT function operation. Pins function as PSTAT outputs independent of function or direction programmed in UART channel 0 control registers.

SPRAM — Internal RAM Supervisor Protect

This bit is used to restrict accesses to the internal RAM space if the access is attempted in the user mode of CPU operation. On reset, this bit is set.

0 = User mode accesses are allowed to the internal RAM.
1 = User mode accesses are prohibited. An attempted access to the internal RAM in user mode will result in $\overline{\text{TEA}}$ assertion to the CPU.

SPROM — Internal ROM Supervisor Protect

This bit is used to restrict accesses to the internal ROM space if the access is attempted in the user mode of CPU operation. On reset, this bit is set.

0 = User mode accesses are allowed to the internal ROM.
1 = User mode accesses are prohibited. An attempted access to the internal ROM in user mode will result in a $\overline{\text{TEA}}$ to the CPU.

HDB — High Data Bus

This bit is used to determine which byte lanes of the internal data bus are driven onto the external data bus when show cycles are enabled. This bit is ignored if SHEN is cleared.

0 = Lower internal data bus bits DATA[15:0] are driven externally.
1 = Upper internal data bus bits DATA[31:16] are driven externally.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

SHEN — Show Cycle Enable

These two bits are used to determine what the EIM does with the external bus during internal transfers (i.e., an access to the internal ROM, RAM or peripherals). When show cycles are enabled, the internal address and data bus are driven externally. On reset, show cycles are disabled.

Table C-9 Show Cycle Enable Field Settings

Value	Meaning
00	Show cycles disabled. The external address bus is driven with the last valid external address, and the data bus values are held by bus keepers.
01	Show cycles enabled. Internal termination to the CPU during idle cycles caused by EDC or CSA being set follows normal operation. This means that internal transfers that occur during EDC/CSA idle cycles will not be visible externally.
10	Show cycles enabled. Internal termination to the CPU during idle cycles caused by EDC or CSA being set is delayed by one clock. This ensures that all internal transfers can be externally monitored, at the expense of performance.
11	Reserved

C.6 PWM Module

This section describes the registers and control bits in the PWM module. All registers reset to 0x0000.

These registers must be accessed with halfword accesses. Accesses other than half-word in size result in undefined activity.

Table C-10 PWM Address Map

Address	Use	Access
10005000	PWM0 Control Register (PWMCR0)	Supervisor Only
10005002	PWM0 Period Register (PWMPR0)	Supervisor Only
10005004	PWM0 Width Register (PWMWR0)	Supervisor Only
10005006	PWM0 Counter Register (PWMCTR0)	Supervisor Only
10005008	PWM1 Control Register (PWMCR1)	Supervisor Only
1000500A	PWM1 Period Register (PWMPR1)	Supervisor Only
1000500C	PWM1 Width Register (PWMWR1)	Supervisor Only
1000500E	PWM1 Counter Register (PWMCTR1)	Supervisor Only
10005010	PWM2 Control Register (PWMCR2)	Supervisor Only
10005012	PWM2 Period Register (PWMPR2)	Supervisor Only
10005014	PWM2 Width Register (PWMWR2)	Supervisor Only
10005016	PWM2 Counter Register (PWMCTR2)	Supervisor Only
10005018	PWM3 Control Register (PWMCR3)	Supervisor Only
1000501A	PWM3 Period Register (PWMPR3)	Supervisor Only
1000501C	PWM3 Width Register (PWMWR3)	Supervisor Only
1000501E	PWM3 Counter Register (PWMCTR3)	Supervisor Only
10005020	PWM4 Control Register (PWMCR4)	Supervisor Only
10005022	PWM4 Period Register (PWMPR4)	Supervisor Only

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Table C-10 PWM Address Map (Continued)

Address	Use	Access
10005024	PWM4 Width Register (PWMWR4)	Supervisor Only
10005026	PWM4 Counter Register (PWMCTR4)	Supervisor Only
10005028	PWM5 Control Register (PWMCR5)	Supervisor Only
1000502A	PWM5 Period Register (PWMPR5)	Supervisor Only
1000502C	PWM5 Width Register (PWMWR5)	Supervisor Only
1000502E	PWM5 Counter Register (PWMCTR5)	Supervisor Only
10005030 to 10005FFF	Reserved	Supervisor Only
10006000 to 10006FFF	Not Used (Access causes transfer error)	Not Applicable

C.6.1 PWM Control Register

The PWM control register (PWMCR) controls the overall operation of the PWM channel. The status of the channel pin is also accessible.

PWMCR0 — PWM0 Control Register **10005000**
PWMCR1 — PWM1 Control Register **10005008**
PWMCR2 — PWM2 Control Register **10005010**
PWMCR3 — PWM3 Control Register **10005018**
PWMCR4 — PWM4 Control Register **10005020**
PWMCR5 — PWM5 Control Register **10005028**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	DOZE	PWM IRQ	IRQ EN	LOAD	DATA	DIR	POL	MODE	COUNT EN	CLKSEL		
W																
RESET:					0	0	0	0		0	0	0	0	0	0	0

Figure C-24 PWM Control Registers

DOZE — Doze Mode

When the CPU executes a **doze** instruction and the system is placed in doze mode, the DOZE bit affects operation of the PWM channel. If this bit is set, the PWM channel is disabled in doze mode. PWM channel operation is suspended at the end of the current period. If IRQ_EN is set, an interrupt request is still generated following the period compare that causes suspension. This interrupt may selectively cause the CPU to exit doze mode.

0 = PWM channel is unaffected in doze mode
1 = PWM channel is disabled in doze mode

At reset, this bit is cleared to zero.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

PWM IRQ — PWM Interrupt Request

This bit indicates that an interrupt was posted by a period compare. This bit can be set by the user to post a PWM interrupt immediately for debugging purposes. This bit is cleared automatically after it is read while set. If IRQ EN is cleared, this bit remains cleared.

0 = No interrupt posted
1 = PWM period rolled over

IRQ EN — Interrupt Request Enable

This bit controls PWM interrupt generation. While this bit is low, the interrupt is disabled.

0 = PWM interrupt disabled
1 = PWM interrupt enabled

LOAD — Load PWMPR and PWMWR

Setting this bit forces a new period. The period and width registers are loaded into the comparator latches and the counter is reset. This bit is cleared automatically after the load has been performed. The actual load occurs some time after the CPU writes this bit, as the load occurs on the next rising PCLK edge following internal synchronization. Forcing a load of the comparator latches and counter in this manner must be done with caution to avoid unexpected pin behavior.

DATA — PWM Data

This bit indicates or controls the current state of the PWM pin. When the pin is configured as a general-purpose output, the logical value written to this bit is used to drive the pin. When the pin is configured as a general-purpose input, the pin value is reflected by this bit. When the pin is configured in PWM mode, the bit reflects the value being driven on the pin by the PWM logic.

DIR — Direction

This bit controls the direction of the pin when used as a GPIO pin. This bit has no affect when MODE indicates PWM mode.

0 = Pin is an input pin
1 = Pin is an output pin

POL — Polarity

This bit controls the polarity of the pin when used as a PWM output pin. Normally, the output pin is set high at period boundaries and goes low when a width compare event occurs.

This bit is ignored if the pin is being used as a GPIO pin.

0 = Normal PWM polarity
1 = Inverted PWM polarity

MODE — PWM Mode

This bit selects whether the PWM pin is used for GPIO or for the PWM function.

0 = General-purpose I/O mode
1 = PWM mode

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

COUNT EN — Counter Enable

This bit enables or disables the PWM counter. The counter is actually enabled or disabled some time after the CPU writes this bit, as the enable occurs on the next rising PCLK edge following internal synchronization. Disabling the counter once it has been running occurs following the next period match.

0 = PWM disabled. While disabled, the counter is in low-power mode and does not count. The following events occur:

When the output pin is configured to operate in PWM mode (MODE =1), the output pin is forced to the setting of the POL bit.

The counter is reset to 00 and frozen.

The contents of the width and period registers are loaded into the comparators.

The comparators are disabled.

If the counter has been running, and the actual disable occurs at the occurrence of a period match, an interrupt request may still be generated, even though the counter is being disabled. To avoid this, write the interrupt enable control bit (IRQ_EN) to zero when disabling the counter.

1 = PWM is enabled and begins a new period. The following events occur:

The output pin changes state to start a new period (if width != 0 and period != 0 and width < period).

The counter is released and begins counting

The comparators are enabled

The PWM IRQ bit is set, indicating the start of a new period if IRQ EN is set.

CLK SEL — Clock Select

These bits select the output of the divider chain.

Table C-11 Clock Select Field Values

Value	Divide By
000	4
001	8
010	16
011	64
100	256
101	2048
110	16384
111	65536

C.6.2 PWM Period Register

The PWM period register (PWMPR) controls the period of the PWM by defining the number of PCLKs in the period. When the counter value matches the value in this register, an interrupt is posted and the counter is reset to start another period.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

PWMPR0 — PWM0 Period Register **10005002**
PWMPR1 — PWM1 Period Register **1000500A**
PWMPR2 — PWM2 Period Register **10005012**
PWMPR3 — PWM3 Period Register **1000501A**
PWMPR4 — PWM4 Period Register **10005022**
PWMPR5 — PWM5 Period Register **1000502A**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PERIOD									
W																
RESET:							0	0	0	0	0	0	0	0	0	0

Figure C-25 PWM Period Registers

PERIOD — Pulse Period

This value causes the counter to be reset. There is one special case. When PERIOD = 0, the output is never set high (0% duty cycle). In this case, the comparator is loaded and the counter is reset on every PCLK. In addition, if enabled, an interrupt request is generated on every PCLK.

C.6.3 PWM Width Register

The PWM width register (PWMWR) defines the pulse width in PCLKs. When the counter matches the value in this register, the output is reset for the duration of the period. Note that if the value in this register is not less than the period register, the output will never be reset, resulting in a 100% duty cycle.

PWMWR0 — PWM0 Width Register **10005004**
PWMWR1 — PWM1 Width Register **1000500C**
PWMWR2 — PWM2 Width Register **10005014**
PWMWR3 — PWM3 Width Register **1000501C**
PWMWR4 — PWM4 Width Register **10005024**
PWMWR5 — PWM5 Width Register **1000502C**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	WIDTH									
W																
RESET:							0	0	0	0	0	0	0	0	0	0

Figure C-26 PWM Width Registers

WIDTH — Pulse Width

When the counter reaches the value in this register, the output is reset.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

C.6.4 PWM Counter Register

The read-only PWM counter register (PWMCR) holds the current count value. It can be read at any time without disturbing the counter.

PWMCR0 — PWM0 Counter Register **10005006**
PWMCR1 — PWM1 Counter Register **1000500E**
PWMCR2 — PWM2 Counter Register **10005016**
PWMCR3 — PWM3 Counter Register **1000501E**
PWMCR4 — PWM4 Counter Register **10005026**
PWMCR5 — PWM5 Counter Register **1000502E**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	COUNT									
W																
RESET:							0	0	0	0	0	0	0	0	0	0

Figure C-27 PWM Count Registers

COUNT — Count Value
This is the current count value.

C.7 Edge Port Programming Model

Access the edge port registers with halfword accesses.

Table C-12 GPIO Edge Port Address Map

Address	Use	Access
10007000	Edge Port Pin Assignment Register (EPPAR)	Supervisor Only
10007002	Edge Port Data Direction Register (EPDDR)	Supervisor Only
10007004	Edge Port Data Register (EPDR)	Supervisor Only
10007006	Edge Port Flag Register (EPFR)	Supervisor Only
10007008 to 10007FFF	Reserved	Supervisor Only

C.7.1 Edge Port Pin Assignment Register (EPPAR)

The 16-bit read/write edge port pin assignment register (EPPAR) configures each of the interrupt pins as either level-sensitive or edge-sensitive. Rising, falling, or both edges can be selected as the active edge. Requests are always generated out of this block but may be masked within the interrupt controller module. The functionality of this register is independent of the programmed pin direction.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

EPPAR — Edge Port Pin Assignment Register **10007000**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EPPA7		EPPA6		EPPA5		EPPA4		EPPA3		EPPA2		EPPA1		EPPA0	
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-28 Edge Port Pin Assignment Register

EPPAx — Edge Port Pin Assignment Select Field x

Pins configured as level-sensitive are inverted so that a logic low on the external pin represents a valid interrupt request. Level-sensitive interrupt inputs are not latched. To guarantee that a level-sensitive interrupt request is acknowledged, the interrupt source must keep the signal asserted until acknowledged by software.

Pins configured as edge-sensitive interrupts are latched for interrupt generation purposes and need not remain asserted for interrupt generation. When the pin is programmed to use the edge detecting circuit, its state is monitored regardless of its configuration as input or output.

These bits are cleared by hardware reset.

Table C-13 EPPAx Field Settings

Value	Meaning
00	Pin INTx defined as level sensitive
01	Pin INTx defined as rising edge detect
10	Pin INTx defined as falling edge detect
11	Pin INTx defined as both falling and rising edge detect

C.7.2 Edge Port Data Direction Register (EPDDR)

The 16-bit read/write edge port data direction register (EPDDR) controls the direction of the port pins. Setting any bit in this register configures the corresponding pin as an output. Clearing any bit in this register configures the corresponding pin as an input. Pin direction is independent of the level/edge mode programmed.

EPDDR — Edge Port Data Direction Register **10007002**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	EPDD7	EPDD6	EPDD5	EPDD4	EPDD3	EPDD2	EPDD1	EPDD0
W																
RESET:									0	0	0	0	0	0	0	0

Figure C-29 Edge Port Data Direction Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

EPDDx — Edge Port Data Direction x

0 = Pin INTx is an input.
1 = Pin INTx is an output.

These bits are cleared by reset.

C.7.3 Edge Port Data Register (EPDR)

The edge port data register (EPDR) is a 16-bit register. Writes to EPDR are stored in an internal latch, and if any pin of the port is configured as an output, the data stored for that bit is driven onto the pin. Reads of this register return the value sensed on the pins for those pins configured as inputs, or the data stored in the register for the pins configured as outputs.

EPDR — Edge Port Data Register **10007004**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	EPD7	EPD6	EPD5	EPD4	EPD3	EPD2	EPD1	EPD0
W																
RESET:									X	X	X	X	X	X	X	X

X = Unaffected by reset

Figure C-30 Edge Port Data Register

EPD[7:0] — Edge Port Data Bits 7:0

See the above description. These bits are not affected by hardware reset.

C.7.4 Edge Port Flag Register (EPFR)

The 16-bit read/write edge port flag register (EPFR) indicates whether the selected edge has been detected on the port pins.

EPFR — Edge Port Flag Register **10007006**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	EPF7	EPF6	EPF5	EPF4	EPF3	EPF2	EPF1	EPF0
W																
RESET:									0	0	0	0	0	0	0	0

Figure C-31 Edge Port Flag Register

EPFx — Edge Port Flag Bit x

0 = Selected edge for INTx pin has not been detected.
1 = Selected edge for INTx pin has been detected.

Bits in this register are set when the programmed edge is detected on the corresponding pin. A bit remains set until cleared by writing it to a one. Pin transitions do not affect this register if the pin is configured as level sensitive (EPPARn=00), and the corresponding flag bit(s) are cleared to zero in this case. When a pin is configured as a general-purpose output, writes to EPDR which cause the selected level or edge

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

interrupt will set the corresponding bit in EPFR. The outputs of this register drive the corresponding input of the interrupt controller for those bits configured as edge detecting. These bits are cleared by hardware reset.

C.8 ISPI Programming Model

These registers control the operation of the ISPI and report its status. The data register exchanges data with external slave devices. After reset, all bits are cleared.

Access these registers with halfword accesses. Accesses other than halfword in size result in undefined activity.

Table C-14 Interval Mode Serial Peripheral Interface Address Map

Address	Use	Access
10008000	ISPI Send/Receive Data Register (SPDR)	Supervisor Only
10008002	ISPI Control Register (SPCR)	Supervisor Only
10008004	ISPI Interval Control Register (SPICR)	Supervisor Only
10008006	ISPI Status Register (SPSR)	Supervisor Only
10008008 to 10008FFF	Reserved	Supervisor Only

C.8.1 ISPI Send/Receive Data Register

The ISPI send/receive data register (SPDR) contains data to be exchanged with external devices. Either writing or reading this register clears any set interrupt.

SPDR — ISPI Data Register **10008000**

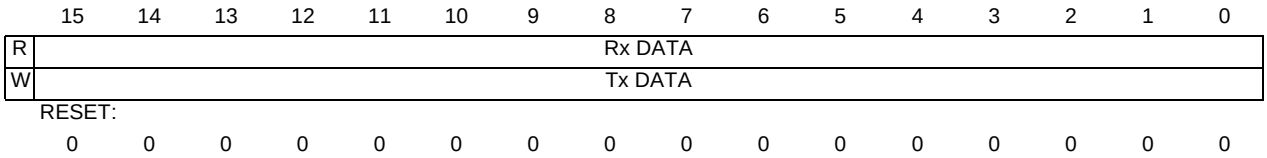


Figure C-32 ISPI Data Register

Rx DATA — Receive Data

This read-only register contains the data bits received from the shift register. Those bits more significant than the size determined in CLOCK COUNT (ISPI control register) return zeros when read. For example, if CLOCK COUNT = 0x8 (9-bit transfer), then bits 15 to 9 are forced to zeros. The value in this register is updated at the end of every transfer.

Tx DATA — Transmit Data

This write-only register contains the data bits to be transmitted to the external device. Data is copied from this register to the shift register at the time that the XCH bit is set.

As data is shifted MSB first, outgoing data is MSB-justified relative to the CLOCK COUNT field in the ISPI control register. For example, if the exchange length is ten bits (CLOCK COUNT = 0x9), the MSB of the outgoing data is bit nine. The first bit presented to the external device is bit nine, followed by the remaining nine less significant bits.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

C.8.2 ISPI Control Register

The ISPI control register (SPCR), along with the ISPI interval control register, controls the operation of the ISPI. Follow this sequence when changing operating modes:

1. Disable the ISPI (COUNT = 0).
2. Wait for any transfer to complete (XCH bit clear).
3. Update to the new mode.
4. Re-enable the ISPI (COUNT = newcount).

SPCR — ISPI Control Register **10008002**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DOZE	SPI_EN	SNS	DRV	MSTR	IRQ_EN	PHA	POL	SPIGP	BAUD RATE			CLOCK COUNT			
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-33 ISPI Control Register

DOZE — Doze Mode

When the CPU executes a **doze** instruction and the system is placed in doze mode, the DOZE bit affects operation of the ISPI. When this bit is set, the ISPI is disabled in doze mode.

0 = ISPI unaffected in doze mode
1 = ISPI disabled in doze mode

At reset, this bit is cleared to zero.

SPI_EN — ISPI Enable

In either master mode, this bit controls the value of the SPI_EN pin. The sense of the SPI_EN pin is determined by the SNS bit. In interval mode, the SPI_EN pin is asserted only when XCH is active. The SPI_EN bit must be programmed to a one for any master mode transfer to occur. In slave mode, the ISPI state machine uses the input value on the SPI_EN pin, and this register bit is ignored. Further, the SPI_EN register bit will not reflect the value of the SPI_EN pin in slave mode.

0 = Negated
1 = Asserted

SNS — SPI_EN Sense

The SNS bit controls the sense of the SPI_EN pin relative to the SPI_EN register bit in the ISPI control register. This is required because in interval mode, the state machine must assert and then negate the SPI_EN pin. The SNS bit has an affect only when the SPI_EN pin is an output. If the SPI_EN pin is an input, then it is active low, and the SNS bit has no effect.

0 = SPI_EN pin is active low
1 = SPI_EN pin is active high

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

DRV — Drive Type

This bit controls the configuration of the SPI_CLK, SPI_EN and SPI_MOSI output buffers in either master mode of the ISPI (MSTR=1). In slave mode, this bit is ignored.

0 = Outputs are totem-pole while in either master mode
1 = Outputs are open-drain while in either master mode

MSTR — Master Mode

This bit controls the mode of the ISPI. In slave mode, the SPI_CLK and SPI_EN pins are inputs; in master modes, they are outputs.

0 = ISPI operates in slave mode
1 = ISPI operates in either interval mode or manual mode (see IVL_EN in SICR)

IRQ_EN — Interrupt Request Enable

This bit enables/disables the ISPI interrupt request output signal. This bit is cleared to zero on reset.

0 = Interrupts disabled
1 = Interrupts enabled

PHA — Phase

This bit controls the phase shift of the SPI_CLK.

0 = Normal phase
1 = Shift advance to opposite phase

POL — Polarity

This bit controls the polarity of the SPI_CLK.

0 = Normal polarity
1 = Inverted polarity

SPIGP — SPI_GP Control

This bit controls the data on the SPI_GP pin.

0 = Pin driven low
1 = Pin driven high

BAUD RATE

These bits select the baud rate of the ISPI bit clock based on divisions of the system clock. The master clock for the ISPI is HI_REFCLK.

Table C-15 BAUD RATE Values

Value	Divide By
000	8
001	16
010	32
011	64
100	128
101	256
110	512
111	1024

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

CLOCK COUNT

These bits select the length of the transfer and control the justification of data. From two to 16 bits can be transferred. A count of all zeros causes the ISPI to be disabled.

Table C-16 CLOCK COUNT Values

Value	Meaning
0000	Disable ISPI
0001	2-bit transfer
.	.
.	.
0111	8-bit transfer
.	.
.	.
1111	16-bit transfer

C.8.3 ISPI Interval Control Register

The ISPI interval control register (SPICR) controls interval mode operation.

SPICR — ISPI Interval Control Register **10008004**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	LPBK	IVL_EN	INTERVAL COUNT												
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-34 ISPI Interval Control Register

LPBK — Loopback

This bit enables a loopback test feature in the ISPI. When looping back, the ISPI operates as if the SPI_MISO and SPI_MOSI pins are wired together and there are no other external devices connected to the ISPI data input pin. Whenever loopback is enabled, the data read from the ISPI data register after a given transfer matches what was written to the ISPI data register prior to that transfer, masked if necessary to account for the number of bits transferred.

0 = Loopback disabled
1 = Loopback enabled

IVL_EN — Interval Mode Enable

This bit, when set, places the ISPI in interval mode. If the MSTR bit in the ISPI control register is cleared, then the ISPI is operating in slave mode, and this bit is ignored.

0 = ISPI is not operating in interval mode
1 = ISPI is operating in interval mode if MSTR=1

INTERVAL COUNT

In interval mode, this register value is loaded into the ISPI interval timer upon completion of a transfer. Each bit-clock period, the value in this counter is decremented by one. When the value in the register reaches zero, then XCH is set, and a new transfer is begun.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

C.8.4 ISPI Status Register

The ISPI status register (SPSR) contains flags indicating whether an overrun condition has occurred, an interrupt has been requested, and whether a transfer is being performed.

SPSR — ISPI Status Register **10008006**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OVR	IRQ	XCH	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-35 ISPI Status Register

OVR — Overrun

This register bit is set by the ISPI controller when a new value is loaded into the RX while the RX is holding previously received data which has not been read. This occurs when SPI_EN becomes inactive and the bit timer has already timed out or when the bit counter times out a second time while SPI_EN remains asserted. It could also be set in interval or manual master mode if the RX data register is not read between transfers. In these cases, the OVR bit may be ignored if appropriate.

0 = No overrun event has occurred
1 = An overrun event has occurred

This bit is cleared by writing it to a one or by reset.

IRQ — Interrupt Request

This register bit is cleared on either a write or a read of the ISPI data register, and when set indicates that an interrupt has been requested.

0 = No interrupt has been requested
1 = An interrupt has been requested

XCH — Exchange

This bit reads the value of XCH, which indicates when the state machine is performing a transfer. In manual mode, XCH is set by writing the ISPI data register. In interval mode, XCH is set automatically by the interval timer. In slave mode, XCH is set when pin SPI_EN is asserted and is negated briefly once the counters determine the completion of a transfer. It is then reasserted if SPI_EN is still asserted. In all modes, XCH is reset upon completion of a transfer.

0 = SPI is idle or interval timer is operating
1 = Initiate exchange or exchange in progress

C.9 UART Programming Model

All UART registers may be accessed either as a halfword or as a byte. The RX and TX data registers may also be accessed as 32-bit words. For these registers the upper 16 bits are forced to zeros.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Table C-17 UART Module Address Map

Address	Use	Access
UART0		
10009000	UART0 Receive Register (U0RX)	Supervisor Only
10009002	Not Used	Supervisor Only
10009004 to 1000903E	U0RX Echoes On Word Boundaries	Supervisor Only
10009040	UART0 Transmit Register (U0TX)	Supervisor Only
10009042	Reserved	Supervisor Only
10009044 to 1000907E	U0TX Echoes On Word Boundaries	Supervisor Only
10009080	UART0 Control Register 1 (U0CR1)	Supervisor Only
10009082	UART0 Control Register 2 (U0CR2)	Supervisor Only
10009084	UART0 Baud Rate Generator Register (U0BRGR)	Supervisor Only
10009086	UART0 Status Register (U0SR)	Supervisor Only
10009088	UART0 Test Register (U0TSR)	Supervisor Only
1000908A	UART0 Port Control Register (U0PCR)	Supervisor Only
1000908C	UART0 Data Direction Register (U0DDR)	Supervisor Only
1000908E	UART0 Port Data Register (U0PDR)	Supervisor Only
10009090 to 10009FFF	Reserved	Supervisor Only
UART1		
1000A000	UART1 Receive Register (U1RX)	Supervisor Only
1000A002	Not Used	Supervisor Only
1000A004 to 1000A03E	U1RX Echoes On Word Boundaries	Supervisor Only
1000A040	UART1 Transmit Register (U1TX)	Supervisor Only
1000A042	Reserved	Supervisor Only
1000A044 to 1000A07E	U1TX Echoes On Word Boundaries	Supervisor Only
1000A080	UART1 Control Register 1 (U1CR1)	Supervisor Only
1000A082	UART1 Control Register 2 (U1CR2)	Supervisor Only
1000A084	UART1 Baud Rate Generator Register (U1BRGR)	Supervisor Only
1000A086	UART1 Status Register (U1SR)	Supervisor Only
1000A088	UART1 Test Register (U1TSR)	Supervisor Only
1000A08A	UART1 Port Control Register (U1PCR)	Supervisor Only
1000A08C	UART1 Data Direction Register (U1DDR)	Supervisor Only
1000A08E	UART1 Port Data Register (U1PDR)	Supervisor Only
1000A088 to 1000AFFF	Reserved	Supervisor Only
1000B000 to 1FFFFFFF	Not Used (Access causes transfer error)	Not Applicable

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

C.9.1 UART Receive Register (URX)

This read-only register contains received characters and status. After reset, if the receiver is enabled (RXEN = 1), the CHARRDY bit is zero until the first character is received, and the remainder of the register contents are undefined. The RX register is echoed to 16-word addresses in order to support unloading the FIFO with the load register quadrant (**ldq**) instruction.

U0RX — UART0 Receive Register **10009000***
U1RX — UART1 Receive Register **1000A000***

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHAR RDY	ERR	OVR RUN	FRM ERR	BRK	PR ERR	0	0	RX DATA							
W																
RESET:	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

*Echoes begin at 10009004 and 1000A004 respectively.
X = Undefined

Figure C-36 UART Receive Register

CHARRDY — Character Ready

This read-only bit indicates whether the character in the RX DATA field and associated flags are valid and ready to be read by the host.

0 = Character in RX DATA field and associated flags are invalid
1 = Character in RX DATA field and associated flags valid and ready for reading

At reset, this bit is cleared to zero.

ERR — Error Detect

When set, this read-only bit indicates that the character present in the RX DATA field has an error status. The error can be an OVRRUN, FRMERR, BRK or PRERR. This bit is updated and valid for each received character.

0 = No error status detected
1 = Error status detected

At reset, this bit is cleared to zero.

OVRRUN — Receiver Overrun

When set, this read-only bit indicates that the receiver ignored data to prevent overwriting the data in the FIFO. Under normal circumstances, this bit should never be set. It indicates that the user's software is not keeping up with the incoming data rate. This bit is updated and valid for each received character, and when set indicates that some number of characters were lost *following* the character for which the flag is set.

0 = No FIFO overrun
1 = A FIFO overrun was detected

At reset, this bit is cleared to zero.

FRMERR — Frame Error

When set, this read-only bit indicates that the current character had a framing error (missing stop bit). The data is possibly corrupted. This bit is updated for each character read from the FIFO.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Every attempt has been made to allow the receiver to correctly interpret data following a character marked as having a framing error. (This includes ignoring the start bit validation logic, if appropriate.) However, when the transmitted data includes two stop bits, and both stop bits are incorrect, then the second stop bit will be interpreted as the start bit of the next character.

0 = Character has no framing error
1 = Character has a framing error

At reset, this bit is cleared to zero.

BRK — Break Detect

When set, this read-only bit indicates that the current character was detected as a break. The data bits are all zero and the stop bit is also zero. The frame error bit is always set when this bit is set. When odd parity is selected, parity error is set when this bit is set. This bit is valid for each character read from the FIFO.

0 = Character is not a break character
1 = Character is a break character

At reset, this bit is cleared to zero.

PRERR — Parity Error

When set, this read-only bit indicates that the current character was detected with a parity error. The data is possibly corrupted. This bit is updated for each character read from the FIFO. While parity is disabled, this bit always reads zero.

0 = No parity error detected for data in RX DATA field
1 = Parity error detected for data in RX DATA field

At reset, this bit is cleared to zero.

RX DATA — Received Data

These read-only bits are the received character. In 7-bit mode, the MSB is forced to zero. In 8-bit mode, all bits are active.

C.9.2 UART Transmit Register (UTX)

The UART transmit register is used by the host to write the data to be transmitted. The low byte is write-only. When this register is read, bits TX[15:8] always return zero, and TX[7:0] is not driving the bus. The TX register is echoed to 16-word addresses in order to support filling the transmit FIFO with the store register quadrant (**stq**) instruction.

U0TX — UART0 Transmit Register **10009040***
U1TX — UART1 Transmit Register **1000A040***

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0								
W									TX DATA							
RESET:	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X

*Echoes begin at 10009044 and 1000A044 respectively.
X = Undefined

Figure C-37 UART Transmit Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

TX DATA — Transmit Data

These write-only bits are the parallel transmit data inputs. In 7-bit mode, D7 is ignored. In 8-bit mode, all bits are used. Data is transmitted LSB first. A new character is transmitted when these bits are written. These bits must be written only while TRDY is high to ensure that corrupted data is not sent.

C.9.3 UART Control Register 1 (UCR1)

UART control register 1 is a read/write register. This register enables the UART and the transmit and receive blocks. It controls the Tx and Rx FIFO levels and enables the TRDY and RRDY interrupts.

U0CR1 — UART0 Control Register 1 **10009080**
U1CR1 — UART1 Control Register 1 **1000A080**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TxFL		TRDY-EN	TXEN	RxFL		RRDY EN	RXEN	IREN	0	RTSD EN	SND-BRK	0	0	DOZE	UART EN
W																
RESET:	0	0	0	0	0	0	0	0	0		0	0			0	0

Figure C-38 UART Control Register 1

TxFL — Transmitter FIFO Interrupt Trigger Level

These bits control the operation of the interrupt generated by the transmitter. A maskable interrupt is generated whenever the data level in the TX FIFO drops below the selected threshold. The bits are encoded as follows:

Table C-18 TxFL Field Settings

Value	Meaning
00	Interrupt if TX FIFO has a slot for one or more character
01	Interrupt if TX FIFO has a slot for four or more characters
10	Interrupt if TX FIFO has a slot for eight or more characters
11	Interrupt if TX FIFO has a slot for fourteen or more characters

At reset, these bits are cleared to zero.

TRDYEN — Transmitter Ready Interrupt Enable

Setting this bit enables an interrupt when the transmitter has one or more slots available in the TX FIFO. The fill level in the TX FIFO at which an interrupt is generated is controlled by the TxFL bits. While this bit is negated, the transmitter interrupt is disabled.

0 = TX interrupt disabled
1 = TX interrupt enabled

At reset, this bit is cleared to zero.

TXEN — Transmitter Enable

This bit enables or disables the transmitter. While UARTEN and TXEN bits are set, and DOZE bit is cleared, the transmitter is enabled. If this bit is cleared in the middle of a transmission, the UART disables the transmitter immediately and starts marking ones.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

The transmitter FIFO cannot be written when this bit is cleared.

0 = Transmitter disabled
1 = Transmitter enabled

At reset, this bit is cleared to zero.

RxFL — Receiver FIFO Interrupt Trigger Level

These bits control the threshold at which a maskable interrupt is generated by the receiver. A maskable interrupt is generated whenever the data level in the RX FIFO reaches the selected threshold.

Table C-19 RxFL Field Settings

Value	Meaning
00	Interrupt if RX FIFO contains one or more character
01	Interrupt if RX FIFO contains four or more characters
10	Interrupt if RX FIFO contains eight or more characters
11	Interrupt if RX FIFO contains fourteen or more characters

At reset, these bits are cleared to zero.

RRDYEN — Receiver Ready Interrupt Enable

Setting this bit enables an interrupt when the receiver has data in the RX FIFO. The fill level in the RX FIFO at which an interrupt is generated is controlled by the RxFL bits. Clearing this bit disables RX interrupts.

0 = RX interrupt disabled
1 = RX interrupt enabled

At reset, this bit is cleared to zero.

RXEN — Receiver Enable

Setting this bit enables the receiver. If the RXD line is already low when the receiver is enabled, the receiver does not recognize break characters, since it requires a valid one-to-zero transition before it can accept any character.

0 = Receiver disabled
1 = Receiver enabled

At reset, this bit is cleared to zero.

IREN — Infrared Interface Enable

This active high bit enables the infrared interface.

0 = Infrared interface disabled
1 = Infrared interface enabled

At reset, this bit is cleared to zero.

RTSD EN — RTS Delta Interrupt Enable

This bit enables or disables RTS delta interrupts. The current status of the $\overline{\text{RTS}}$ pin is read in the UART status register.

0 = RTS interrupt disabled
1 = RTS interrupt enabled

At reset, this bit is cleared to zero.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SNDBRK — Send Break

This bit forces the transmitter to send a break character. The transmitter will finish sending the character in progress (if any) and then send break characters until this bit is reset. The user is responsible for ensuring that this bit is high for a sufficient period of time to generate a valid break; the transmitter samples SNDBRK after every bit is transmitted.

Following completion of the break transmission, the UART transmits two mark bits. The user can continue to fill the FIFO, and any characters remaining will be transmitted when the break is terminated. This bit cannot be changed until the UART EN and TX EN bits in the UART control register 1 (UxCR1) are set.

0 = Do not send break
1 = Send break (continuous zeros)

At reset, this bit is cleared to zero.

DOZE

When the CPU executes a **doze** instruction and the system is placed in the doze mode, the DOZE bit affects operation of the UART. If this bit is set when the system is in the doze mode, the UART is disabled.

0 = UART unaffected while the MCU is in doze mode
1 = UART disabled while the MCU is in doze mode

At reset, this bit is cleared to zero.

UART EN — UART Enable

This bit enables or disables the UART. If this bit is cleared in the middle of a transmission, the transmitter stops and drives the TXD line to logic one.

0 = UART disabled
1 = UART enabled

At reset, this bit is cleared to zero.

C.9.4 UART Control Register 2 (UCR2)

UART control register 2 is a read/write register. This register controls the overall operation of the UART. It controls the clock source, number of bits per character, parity generation and checking, and behavior of the $\overline{\text{RTS}}$, $\overline{\text{CTS}}$, and $\overline{\text{DTR}}$ pins.

U0CR2 — UART0 Control Register 2 **10009082**
U1CR2 — UART1 Control Register 2 **1000A082**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	IRTS	CTSC	CTS	0	0	0	PREN	PROE	STPB	WS	0	0	0	0	0
W																
RESET:		0	0	0				0	0	0	0	0				

Figure C-39 UART Control Register 2

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

IRTS — Ignore $\overline{\text{RTS}}$

Setting this bit forces the $\overline{\text{RTS}}$ input signal presented to the transmitter to always be asserted, effectively causing the external pin to be ignored. In this mode, the $\overline{\text{RTS}}$ pin can be used as a general-purpose input.

0 = Transmit only while $\overline{\text{RTS}}$ pin is asserted
1 = Ignore $\overline{\text{RTS}}$ pin

At reset, this bit is cleared to zero.

CTSC — $\overline{\text{CTS}}$ Pin Control

This bit controls the operation of the $\overline{\text{CTS}}$ output pin. While this bit is set, the $\overline{\text{CTS}}$ output pin is controlled by the receiver. When the RX FIFO has a pending overrun, the $\overline{\text{CTS}}$ output pin is negated to indicate to the far-end transmitter to stop transmitting. While the CTSC bit is negated, the $\overline{\text{CTS}}$ output pin is controlled by the CTS bit.

On reset, since this bit is cleared to zero, the $\overline{\text{CTS}}$ pin is controlled by the CTS bit, which is also cleared to zero on reset. This means that on reset the $\overline{\text{CTS}}$ signal is negated.

0 = $\overline{\text{CTS}}$ pin controlled by the CTS bit
1 = $\overline{\text{CTS}}$ pin controlled by the receiver

At reset, this bit is cleared to zero.

CTS — CTS bit

This bit controls the $\overline{\text{CTS}}$ pin while the CTSC bit is negated. While CTSC is asserted this bit has no function.

0 = $\overline{\text{CTS}}$ pin is driven high (inactive)
1 = $\overline{\text{CTS}}$ pin is driven low (active)

At reset, this bit is cleared to zero.

PREN — Parity Enable

This bit enables or disables the parity generator in the transmitter and parity checker in the receiver.

0 = Parity disabled
1 = Parity enabled

At reset, this bit is cleared to zero.

PROE — Parity Odd/Even

This bit controls the sense of the parity generator and checker. When PROE is set, odd parity is generated and expected. When PROE is cleared, even parity is generated and expected. This bit has no function if PREN is low.

0 = Even parity
1 = Odd parity

At reset, this bit is cleared to zero.

STPB — Stop Bits

This bit controls the number of stop bits transmitted after a character. When STPB is set, two stop bits are sent. When STPB is cleared, one stop bit is sent. This bit has no effect on the receiver, which expects one or more stop bits.

0 = One stop bit transmitted
1 = Two stop bits transmitted

At reset, this bit is cleared to zero.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

WS — Word Size

This bit specifies a character length of eight or seven bits (not including start, stop, or parity bits). When WS is set, the transmitter and receiver are in eight-bit mode. When WS is cleared, they are in seven-bit mode. The transmitter then ignores B7, and the receiver sets B7 to zero. This bit can be changed between transmissions or receptions. If it is changed while a transmission or reception is in progress, however, the length of the current character being transmitted or received is unpredictable.

0 = 7-bit transmit and receive character length
1 = 8-bit transmit and receive character length

At reset, this bit is cleared to zero.

C.9.5 UART BRG Register (UBRGR)

This register specifies the divide ratio of the prescaler in the UART bit clock generator.

U0BRGR — UART0 BRG Register **10009084**
U1BRGR — UART1 BRG Register **1000A084**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	CD											
W																
RESET:					0	0	0	0	0	0	0	0	0	0	0	0

Figure C-40 UART BRG Register

CD — Clock Divider

These bits determine the bit clock generator output rate. The CD field is used to preset a 12-bit counter that is decremented at the system clock rate. The value 0x000 produces the maximum clock rate (equal to the system clock). The value 0xFFF produces the minimum clock rate (divide by 4096).

C.9.6 UART Status Register (USR)

The read/write UART status register indicates the status of the $\overline{\text{RTS}}$ pin, input transitions on the pin, and status of the transmit and receive FIFOs.

U0SR — UART0 Status Register **10009086**
U1SR — UART1 Status Register **1000A086**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX MPTY	RTSS	TRDY	0	0	0	RRDY	0	0	0	RTSD	0	0	0	0	0
W																
RESET:	1	0	1				0				0					

Figure C-41 UART Status Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

TXMPTY — Transmitter Empty

When set, this bit indicates that the transmit FIFO and the transmit shift register are both empty. This bit is automatically cleared when a write to the TX FIFO is performed.

0 = TX FIFO or shifter are not both empty
1 = TX FIFO and shifter are both empty

At reset, this bit is set to one.

RTSS — $\overline{\text{RTS}}$ Pin Status

This bit indicates the current status of the $\overline{\text{RTS}}$ pin. A “snapshot” of the pin is taken immediately before this bit is presented to the data bus. While IRTS is asserted, this bit can be used as a general-purpose input.

0 = $\overline{\text{RTS}}$ pin is high (inactive)
1 = $\overline{\text{RTS}}$ pin is low (active)

This bit follows the logic value connected to the $\overline{\text{RTS}}$ pin.

TRDY — Transmitter Ready Interrupt Flag

When set, this bit indicates that the TX FIFO has emptied below its target threshold and needs data. This bit is automatically cleared when the data level in the TX FIFO goes beyond the set threshold level.

0 = Transmitter does not need data
1 = Transmitter needs data (interrupt posted)

At reset, this bit is set to one.

RRDY — Receiver Ready Interrupt Flag

When set, this bit indicates that the receive FIFO data level is above the threshold level specified by the RxFL field, and a maskable interrupt is generated. Refer to the RxFL bit description for setting the threshold level. In conjunction with the CHARRDY bit, host software can continue to read the RX FIFO in an interrupt service routine until the RX FIFO is empty. This bit is automatically cleared when the data level in the RX FIFO goes below the set threshold level.

0 = No character ready (no interrupt posted)
1 = Character(s) ready (interrupt posted)

At reset, this bit is cleared to zero.

RTSD — RTS Delta

When set, this bit indicates that the $\overline{\text{RTS}}$ pin changed state. It generates a maskable interrupt. In STOP mode, $\overline{\text{RTS}}$ assertion sets this bit to wake the CPU. The current state of the $\overline{\text{RTS}}$ pin is available in the RTSS bit. The RTSD interrupt is cleared by writing a one to this bit.

0 = $\overline{\text{RTS}}$ pin did not change state since last cleared
1 = $\overline{\text{RTS}}$ pin changed state

At reset, this bit is cleared to zero.

C.9.7 UART Test Register (UTSR)

The UART test register is a read/write register. Unimplemented bits always return zero when read. This register contains miscellaneous bits to control test features of the UART block.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

U0TSR — UART0 Test Register **10009088**
U1TSR — UART1 Test Register **1000A088**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	FRC PERR	LOOP	0	LOOP IR	0	0	0	0	0	0	0	0	0	0
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-42 UART Test Register

FRC PERR — Force Parity Error

When set, this bit forces the transmitter to generate a parity error if parity is enabled. This bit is provided for system debugging.

0 = Generate normal parity
1 = Generate inverted parity (error)

At reset, this bit is cleared to zero.

LOOP — Loop TX and RX for Test

This bit controls loopback for test purposes. When this bit is high, the receiver input is internally connected to the transmitter and ignores the RxD pin. The transmitter is unaffected by this bit. This loopback operates to connect the data on the TxD pin directly to the voting logic. If infrared mode is enabled (IR_EN is active), the effect of activating this bit is to put an IR-formatted bit stream into the voting logic, which will yield odd results. Do not use this loopback if IR_EN is active.

0 = Normal receiver operation
1 = Internal connect transmitter output to receiver input

At reset, this bit is cleared to zero.

LOOP IR — Loop TX and RX for IR Test

This bit controls a loopback from transmitter to receiver in the infrared interface.

0 = No IR loop
1 = Connect IR transmit to IR receiver

At reset, this bit is cleared to zero.

C.9.8 UART Port Control Register (UPCR)

The read/write UART port control register controls the functionality of UART GPIO pins.

U0PCR — UART0 Port Control Register **1000908A**
U1PCR — UART1 Port Control Register **1000A08A**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	PC3	PC2	PC1	PC0
W																
RESET:													0	0	0	0

Figure C-43 UART Port Control Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

PCx — Port Control Bit x

0 = Corresponding pin is configured as GPIO pin
1 = Corresponding pin is configured as UART pin

At reset, these bits are cleared to zero.

C.9.9 UART Data Direction Register (UDDR)

This register controls the direction of UART GPIO pins.

U0DDR — UART0 Data Direction Register **1000908C**
U1DDR — UART1 Data Direction Register **1000A08C**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	PDC3	PDC2	PDC1	PDC0
W																
RESET:													0	0	0	0

Figure C-44 UART Data Direction Register

PDCx — Port Direction Control Bit x

0 = Corresponding GPIO pin is configured as input
1 = Corresponding GPIO pin is configured as output

At reset, these bits are cleared to zero.

C.9.10 UART Port Data Register (UPDR)

The UART port data register is used to read or write data to or from UART GPIO pins.

U0PDR — UART0 Port Data Register **1000908E**
U1PDR — UART1 Port Data Register **1000A08E**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	PD3	PD2	PD1	PD0
W																
RESET:													X	X	X	X

X = Undefined

Figure C-45 UART Port Data Register

PDx — Port Data Bit x

These bits are used to read or write data from/to the corresponding port pins if they are configured as GPIO (by PC[3:0] bits in UPCR). If a port pin x is configured as a GPIO input, then the corresponding PDx bit will reflect the value present on this pin. If a port pin x is configured as a GPIO output, then the value written into the corresponding PDx bit will be reflected on the pin.

Note that since the $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ pins are not present for UART1, the corresponding port control register bits should be configured in a manner which provides deterministic data when the port data register is read. One method for doing this is to configure the missing pins as general-purpose outputs.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

C.10 OnCE Registers

C.10.1 OnCE Command Register (OCMR)

The OnCE command register (OCMR) is an 8-bit shift register that receives its serial data from the TDI pin. This register corresponds to the JTAG IR, and is loaded when the update-IR TAP controller state is entered.

OCMR — OnCE Command Register

BIT 7	6	5	4	3	2	1	BIT 0
R/W	GO	EX	RS				

Figure C-46 OnCE Command Register

R/W — Read/Write Command

The R/W bit specifies the direction of data transfer.

0 = Write the data associated with the command into the register specified by the RS field.
1 = Read the data contained in the register specified by the RS field.

GO — Go Command

If the GO bit is set, the chip executes the instruction that resides in the IR register in the CPUSCR. To execute the instruction, the processor leaves debug mode, executes the instruction, and if the EX bit is cleared, returns to debug mode immediately after executing the instruction. The processor resumes normal operation if the EX bit is set. The GO command is executed only if the operation is a read/write to either CPUSCR or “No register selected”. Otherwise, the GO bit is ignored. The processor leaves debug mode after the TAP controller update-DR state is entered.

0 = Inactive (no action taken)
1 = Execute instruction in IR

EX — Exit Command

If the EX bit is set, the processor leaves debug mode and resumes normal operation until another debug request is generated. The exit command is executed only if the go command is issued, and the operation is a read/write to CPUSCR or read/write to “No register selected”. Otherwise the EX bit is ignored. The processor exits debug mode after the TAP controller update-DR state is entered.

0 = Remain in debug mode
1 = Leave debug mode

RS — Register Select

The register select bits define the source or destination register for the read or write operation, respectively. **Table C-20** indicates the OnCE register addresses.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table C-20 OnCE Register Addressing

RS	Register Selected
00000	Reserved
00001	Reserved
00010	Reserved
00011	Trace Counter (OTC)
00100	Memory Breakpoint Counter A (MBCA)
00101	Memory Breakpoint Counter B (MBCB)
00110	Program Counter FIFO and Increment Counter
00111	Breakpoint Address Base Register A (BABA)
01000	Breakpoint Address Base Register B (BABB)
01001	Breakpoint Address Mask Register A (BAMA)
01010	Breakpoint Address Mask Register B (BAMB)
01011	CPU Scan Register (CPUSCR)
01100	No Register Selected (Bypass)
01101	OnCE Control Register (OCR)
01110	OnCE Status Register (OSR)
01111	Reserved (Factory Test Control Register — do not access)
10000	Reserved (MEM_BIST, do not access)
10001 – 10110	Reserved (Bypass, do not access)
10111	Reserved (LSRL, do not access)
11000 – 11110	Reserved (Bypass, do not access)
11111	Bypass

C.10.2 OnCE Control Register (OCR)

The OnCE control register (OCR) is a 32-bit register used to select the events that put the chip in debug mode and to enable or disable sections of the OnCE logic. The control bits are read/write.

OCR — OnCE Control Register

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SQC	
W																
RESET:															0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DR	IDRE	TME	FRZC	RCB	BCB					RCA	BCA				
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-47 OnCE Control Register

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

SQC — Sequential Control

The SQC field allows memory breakpoint B and trace occurrences to be suspended until a qualifying event occurs. This field is cleared on test logic reset.

Table C-21 Sequential Control Field Definition

SQC[1:0]	Meaning
00	Disable sequential control operation. Memory breakpoints and trace operation are unaffected by this field.
01	Suspend normal trace counter operation until a breakpoint condition occurs for memory breakpoint B. When this mode is selected, memory breakpoint B occurrences no longer cause a breakpoint request to be generated. Instead, trace counter comparisons are suspended until the first memory breakpoint B occurrence. After the first memory breakpoint B occurrence, trace counter control is released to perform normally (assuming TME is set). This allows a sequence of breakpoint conditions to be specified prior to trace counting.
10	Qualify memory breakpoint B matches with a breakpoint occurrence for memory breakpoint A. When this bit is set, memory breakpoint A occurrences no longer cause a breakpoint request to be generated. Instead, memory breakpoint B comparisons are suspended until the first memory breakpoint A occurrence. After the first memory breakpoint A occurrence, memory breakpoint B is enabled to perform normally. This allows a sequence of breakpoint conditions to be specified.
11	Combine the qualifications specified by the 01 and 10 encodings of this field. In this mode, no breakpoint requests are generated, and trace count operation is enabled (when TME is set) once a memory breakpoint B occurrence follows a memory breakpoint A occurrence.

DR — CPU Debug Request Control

This control bit is used to request the CPU to enter debug mode unconditionally. The CPU indicates that debug mode has been entered via the PM bits in the OnCE status register. Once the CPU enters debug mode, it returns there even with a write to the OCMR with GO and EX set until the DR bit is cleared. This bit is cleared on test logic reset.

IDRE — Internal Debug Request Enable

This control bit is used to enable internally generated debug requests. The internal debug request input to the OnCE control logic ($\overline{\text{IDR}}$) may not be used in all implementations. In some implementations, the $\overline{\text{IDR}}$ control input may be connected and used as an additional hardware debug request. This bit is cleared on test logic reset.

0 = Disable $\overline{\text{IDR}}$ input operation
1 = Enable $\overline{\text{IDR}}$ input operation

TME — Trace Mode Enable

The TME control bit enables the OnCE trace mode operation. This bit is cleared on test logic reset. Trace operation is also affected by the SQC field described above.

0 = Disable trace operation
1 = Enable trace operation

FRZC — Freeze Control

This control bit is used in conjunction with memory breakpoint B registers to select between asserting a breakpoint condition when a memory breakpoint B occurs, or freezing the PC FIFO from further updates when memory breakpoint B occurs while allowing the CPU to continue execution. The PC FIFO remains frozen until the FRZO bit in the OSR is cleared.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

0 = Memory breakpoint B occurrence causes assertion of a breakpoint condition
1 = Memory breakpoint B occurrence causes a freeze of PC FIFO from further updates and no breakpoint assertion

RCB, RCA — Memory Breakpoint B, A Range Control

These control bits condition enabled memory breakpoints. They condition whether memory breakpoint matches will occur when a memory address falls either within the range defined by memory base address and mask, or outside the range.

0 = Condition breakpoint on access within range
1 = Condition breakpoint on access outside of range

BCB, BCA — Memory Breakpoint B, A Control

These control bits enable memory breakpoints and qualify the access attributes to select whether the breakpoint match will be recognized for read, write, or instruction fetch (program space) accesses. These bits are cleared on test logic reset. See **Table C-22** for the definition of the BCA and BCB fields.

Table C-22 Memory Breakpoint Control Field Definition

BC4	BC3	BC2	BC1	BC0	Description
0	0	0	0	0	Breakpoint disabled
0	0	0	0	1	Qualify match with any access
0	0	0	1	0	Qualify match with any instruction access
0	0	0	1	1	Qualify match with any data access
0	0	1	0	0	Qualify match with any change of flow instruction access
0	0	1	0	1	Qualify match with any data write
0	0	1	1	0	Qualify match with any data read
0	0	1	1	1	Reserved
0	1	x	x	x	Reserved
1	0	0	0	0	Reserved
1	0	0	0	1	Qualify match with any user access
1	0	0	1	0	Qualify match with any user instruction access
1	0	0	1	1	Qualify match with any user data access
1	0	1	0	0	Qualify match with any user change of flow access
1	0	1	0	1	Qualify match with any user data write
1	0	1	1	0	Qualify match with any user data read
1	0	1	1	1	Reserved
1	1	0	0	0	Reserved
1	1	0	0	1	Qualify match with any supervisor access
1	1	0	1	0	Qualify match with any supervisor instruction access
1	1	0	1	1	Qualify match with any supervisor data access
1	1	1	0	0	Qualify match with any supervisor change of flow access
1	1	1	0	1	Qualify match with any supervisor data write
1	1	1	1	0	Qualify match with any supervisor data read
1	1	1	1	1	Reserved

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

C.10.3 OnCE Status Register (OSR)

The OnCE status register (OSR) is a 16-bit register used to indicate the reason(s) that debug mode was entered and the current operating mode of the CPU. These status bits are read only.

OSR — OnCE Status Register

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	HDRO	DRO	MBO	SWO	TO	FRZO	SQB	SQA	PM	
W																
RESET:							0	0	0	0	0	0	0	0	0	0

Figure C-48 OnCE Status Register

HDRO — Hardware Debug Request Occurrence

This read-only status bit is set when the processor enters debug mode as a result of a hardware debug request from the $\overline{\text{IDR}}$ signal or the $\overline{\text{DE}}$ pin. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

DRO — Debug Request Occurrence

This read-only status bit is set when the processor enters debug mode and the debug request (DR) control bit in the OnCE control register is set. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

MBO — Memory Breakpoint Occurrence

This read-only status bit is set when a memory breakpoint request has been issued to the CPU via the $\overline{\text{BRKRQ}}$ input and the CPU enters debug mode. In some situations involving breakpoint requests on instruction prefetches, the CPU may discard the request along with the prefetch. In this case, this bit may become set due to the CPU entering debug mode for another reason. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

SWO — Software Debug Occurrence

This read-only status bit is set when the processor enters debug mode of operation as a result of the execution of the **bkpt** instruction. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

TO — Trace Count Occurrence

This read-only status bit is set when the trace counter reaches zero with the trace mode enabled and the CPU enters debug mode. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

FRZO — FIFO Freeze Occurrence

This read-only status bit is set when a FIFO freeze occurs. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

SQB — Sequential Breakpoint B Arm Occurrence

This read-only status bit is set when sequential operation is enabled and a memory breakpoint B event has occurred to enable trace counter operation. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SQA — Sequential Breakpoint A Arm Occurrence

This read-only status bit is set when sequential operation is enabled and a memory breakpoint A event has occurred to enable memory breakpoint B operation. This bit is cleared on test logic reset or when debug mode is exited with the GO and EX bits set.

PM — Processor Mode

These status bits indicate the processor operating mode. They allow coordination of the OnCE controller with the CPU to synchronize the two.

Table C-23 Processor Mode Field Definition

PM[1:0]	Meaning
00	Processor in normal mode
01	Processor in stop, doze, or wait mode
10	Processor in debug mode
11	Reserved

C.10.4 Memory Address Latch (MAL)

The memory address latch (MAL) is a 32-bit register that latches the address bus on every access.

C.10.5 Breakpoint Address Base Registers (BABA, BABB)

The 32-bit breakpoint address base registers (BABA, BABB) store memory breakpoint base addresses. BABA and BABB can be read or written through the OnCE serial interface. Before enabling breakpoints, the external command controller should load these registers.

C.10.6 Breakpoint Address Mask Registers (BAMA, BAMB)

The 32-bit breakpoint address mask registers (BAMA, BAMB) store memory breakpoint base address masks. BAMA and BAMB can be read or written through the OnCE serial interface. Before enabling breakpoints, the external command controller should load these registers.

C.10.7 Breakpoint Address Comparators

Each breakpoint address comparator compares the current memory address (stored in MAL) with the contents of the base, as appropriately masked by the BAMx. When a match occurs, the comparator signals the breakpoint logic.

C.10.8 Memory Breakpoint Counters (MBCA, MBCB)

The 16-bit memory breakpoint counter x (MBCx) register is loaded with a value equal to the number of times, minus one, that a memory access event can occur before a memory breakpoint is declared.

C.10.9 Program Counter Register (PC)

The program counter register (PC) is a 32-bit latch that stores the value of the program counter that was present when the chip entered debug mode.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

C.10.10 Instruction Register (IR)

The instruction register (IR) provides a mechanism for controlling the debug session by forcing in selected instructions and then causing them to be executed in a controlled manner by the debug control block.

C.10.11 Control State Register (CTL)

The control state register (CTL) is used to set control values when debug mode is exited. On scan-in, this register is used to control specific aspects of the CPU. Certain bits reflect internal processor status and should be restored to their original values.

The CTL is a 16-bit latch that stores the value of certain internal CPU state variables before debug mode is entered. This register is affected by the operations performed during the debug session and should be restored by the external command controller when returning to normal mode. In addition to saved internal state variables, the bits are used by emulation firmware to control the debug process.

Set reserved bits to ones.

CTL — Control State Register

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							FFY	FDB	SZ		TC			Reserved	
RESET:								0	0	0	0	0	0	0		

Figure C-49 Control State Register

FFY — Feed Forward Y Operand

This control bit is used to force the content of the WBBR to be used as the Y operand value of the first instruction to be executed following an update of the CPUSCR. This gives the debug firmware the capability of updating processor registers by initializing the WBBR with the desired value, setting the FFY bit, and executing a **mov** instruction to the desired register.

FDB — Force PSR Debug Mode

A logical OR of this control bit with the PSR(DB) bit determines whether the processor is operating in debug enable mode or not. The processor can be placed in debug enable mode by setting this bit regardless of the state of the PSR(DB) bit. In debug enable mode, execution of the **bkpt** instruction as well as recognition of the $\overline{\text{BRKRQ}}$ input cause the processor to enter debug mode, as if the $\overline{\text{DBGRQ}}$ input had been asserted.

SZ — Prefetch Size

This control field is used to drive the CPU SIZ[1:0] outputs on the first instruction prefetch caused by issuing a OnCE command with the GO bit set and not ignored. It should be set to indicate a 16-bit size, i.e., 0b10. This field should be restored to its original value after a debug session is completed, i.e., when a OnCE command is issued with the GO and EX bits set and not ignored.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

TC — Prefetch Transfer Code

This control field is used to drive the CPU TC[2:0] outputs on the first instruction prefetch caused by issuing a OnCE command with the GO bit set and not ignored. It should typically be set to indicate a supervisor instruction access, i.e., 0b110. This field should be restored to its original value after a debug session is completed, i.e., when a OnCE command is issued with the GO and EX bits set and not ignored.

C.10.12 Write-Back Bus Register (WBBR)

The write-back bus register (WBBR) is used as a means of passing operand information between the CPU and the external command controller.

To update a processor resource, this register is initialized with a data value to be written, and a **mov** instruction is executed which uses this value as a write-back data value. The FFY bit in the control state register forces the value of the WBBR to be substituted for the normal source value of a **mov** instruction, thus allowing updates to processor registers to be performed.

C.10.13 Processor Status Register (PSR)

The OnCE processor status register (PSR) is a 32-bit latch used to read or write the M•CORE processor status register. Whenever the external command controller needs to save or modify the contents of the M•CORE processor status register, this register is used. This register is affected by the operations performed in debug mode and must be restored by the external command controller when returning to normal mode.

C.10.14 Reserved Test Control Registers (Reserved, MEM_BIST, FTCR, LSRL)

These registers are reserved for factory testing.

WARNING

To prevent damage to the device or system, do not access these registers during normal operation.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

INDEX

–A–

AC electrical specifications A-2
ADDR (address bus) signals 4-4, 7-1
Address bus 4-4, 7-1
AE bit 9-5, C-7
AF bit 2-4
AIE bit 9-5, 9-6, C-7
AIF bit 9-5, 9-6, C-7
Alarm Enable bit 9-5, C-7
Alarm Interrupt Enable bit 9-5, 9-6, C-7
Alarm Interrupt Flag 9-5, 9-6, C-7
Alternate file 2-3
Alternate File bit 2-4

–B–

BABA 16-14, C-51
BABB 16-14, C-51
BACA 16-14, C-51
BACB 16-14, C-51
BAMA 16-14, C-51
BAMB 16-14, C-51
BAUD RATE field 12-7, C-32
BCA bit 16-10, C-49
BCB bit 16-10, C-49
Big-endian byte ordering 2-5
Bit clock generator 11-4
Bit time 11-4
Boot mode 4-4, 7-2, 7-6
Break Detect bit 11-8, C-37
Break frame 11-3, 11-4
Breakpoint
    Address Base Registers 16-14, C-51
    Address Comparators 16-14, C-51
    Address Mask Registers 16-14, C-51
    logic 16-12
BRK bit 11-8, C-37
$\overline{\text{BRKRQ}}$ signal 16-5
Bus sizing 7-4
Bus watchdog 7-6

–C–

C bit 2-3
CD bit 11-13, C-42
CHARRDY bit 11-7, C-36
Chip Select 4-4, 7-2
    Assert bit 7-9, C-18
    Control Registers 7-7, C-16
    Enable bit 7-11, C-20
CKOE bit 8-7, 9-4, C-6
CKOS bit 9-3, C-6
Clear to send 4-7, 11-2
CLK SEL field 15-6, C-25
CLKIN signal 4-5
CLKOUT Enable bit 8-7, 9-4, C-6
CLKOUT signal 4-5, 8-7
CLKOUT Source bit 9-3, C-6
CLKSRC bit 11-13
Clock
    Divider bit 11-13, C-42
    input 4-5
    input specifications A-2
    module 8-1
        block diagram 8-3
    output 4-5, 8-7
    Select field 15-6, C-25
    source 8-1
    Source bit 11-13
CLOCK COUNT field 12-7, C-33
Column strobes 4-7
Condition Code/Carry bit 2-3
Control registers 2-3
Control State Register 16-18, C-52
COUNT EN bit 15-5, C-25
COUNT field 15-7, C-27
Counter Enable bit 15-5, C-25
Counter Overwrite Enable bit 9-14, C-12
Counter Reload Control bit 9-12, 9-13, 9-15, C-12
CPU
    breakpoint request 16-5
    debug acknowledge 16-5
    debug request 16-5
    Debug Request Control bit 16-9, C-48
    status signals 16-5
Crystal oscillator 4-5
CS signals 4-4, 7-2
CSA bit 7-9, C-18
CSEN bit 7-11, C-20
CSxCR 7-7, C-16
CTL 16-18, C-52
$\overline{\text{CTS}}$ bit 11-12, C-41
$\overline{\text{CTS}}$ Pin Control bit 11-12, C-41
$\overline{\text{CTS}}$ signal 11-2
$\overline{\text{CTS0}}$ 4-7
CTSC bit 11-12, C-41

–D–

DATA (data bus) signals 4-4, 7-2

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

DATA bit 15-5, C-24
Data bus 4-4, 7-2
Data organization
in memory 2-5
in registers 2-5
Data Port Size bit 7-10, C-19
DBG bit 9-14, C-11
DBGACK signal 16-5
DBGRQ signal 16-5
DC electrical specifications A-1
DE signal 4-6, 16-4
Debug
event 4-6, 16-4
mode
entering 16-16
ISP 12-11
select 16-3
UART 11-24
Mode Control bit 9-14, C-11
Request Occurrence bit 16-11, C-50
serial clock 16-3
serial input 16-3
serial output 16-3
DEBUG signal 16-6
DIR bit 15-5, C-24
Direction bit 15-5, C-24
DOZE bit
ISPI 12-6, C-31
PIT 9-14, C-11
PWM 15-4, C-23
UART 11-11, C-40
Doze mode 8-4
DR bit 16-9, C-48
Drive Type bit 12-6, C-32
DRO bit 16-11, C-50
DRV bit 12-6, C-32
DSZ bit 7-10, C-19

–E–

EB Negate bit 7-10, C-19
EB signals 4-4, 7-2
EBC bit 7-10, C-19
EBRO bit 16-11, C-50
EDC bit 7-9, C-18
Edge Port 13-1
block diagram 13-1
Data Direction Register 13-3, C-28
Data Register 13-3, C-29
Flag Register 13-4, C-29
Pin Assignment Register 13-2, C-27
programming model 13-2
signals 13-1
Edge-triggered interrupts 13-2, C-27
EF flags 10-4, C-3
EIM 7-1
bus sizing 7-4
Configuration Register 7-11, C-20
programming model 7-7
signals 7-1
timing specifications A-4
EIMCR 7-11, C-20
Electrical characteristics A-1
EN bit 9-15, C-12
EN flags 10-3, C-3
Enable
Byte Control bit 7-10, C-19
bytes 4-4, 7-2
Fast Interrupt Flags 10-4, C-3
Normal Interrupt Flags 10-3, C-3
EPDDR 13-3, C-28
EPDR 13-3, C-29
EPFR 13-4, C-29
EPPAR 13-2, C-27
EPSR 2-3
ERR bit 11-7, C-36
Error Detect bit 11-7, C-36
EX bit 16-7, C-46
Exceptions
cycles 2-14
processing 2-3
shadow registers 2-4
Exchange flag 12-9, C-34
Exit Command bit 16-7, C-46
EXOSC pin 4-5
External Breakpoint Request Occurrence bit 16-11, C-50
External bus
timing diagrams 7-13
External interface module. *See* EIM
External interrupts 4-6, 13-1
timing specifications A-3
Extra Dead Cycle bit 7-9, C-18

–F–

Factory test mode 4-6
Fast Interrupt
Enable Register 10-3, C-3
Pending Flags 10-5, C-5
Pending Register 10-5, C-4
FDB bit 16-19, C-52
Feed Forward Y Operand bit 16-19, C-52
FFY bit 16-19, C-52
FIER 10-3, C-3
FIFO buffer 16-20
FIFO Freeze Occurrence bit 16-11, C-50
FIPND 10-5, C-4
Force Parity Error bit 11-15, C-44
Force PSR Debug Enable Mode bit 16-19
Force PSR Debug Mode bit C-52
FP flags 10-5, C-5
FPSR 2-3
Frame 11-4
Frame Error bit 11-8, C-36
Framing error 11-5
FRC PERR bit 11-15, C-44
Freeze Control bit 16-10, C-48
FRMERR bit 11-8, C-36
FRZC bit 16-10, C-48
FRZO bit 16-11, C-50

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

–G–

General-purpose I/O 13-1
General-purpose registers 2-3
Glitch suppression, KPP 14-7
GND pin 4-9
GO bit 16-7, C-46
GPIO 13-1
Ground 4-9

–H–

Hardware Debug Request Occurrence bit 16-11, C-50
HDB bit 7-7, 7-12, C-21
HDRO bit 16-11, C-50
HI_REFCLK 8-1
High Data Bus bit 7-7, 7-12, C-21

–I–

$\overline{\text{IDR}}$ signal 16-5
IDRE bit 16-9, C-48
Ignore $\overline{\text{RTS}}$ bit 11-2, 11-12, C-41
IN bits 10-3, C-2
Infrared interface 11-4
Infrared Interface Enable bit 11-11, C-39
Instruction
    address FIFO buffer 16-20
    Register 16-18, C-52
    timing 2-2
INT signals 4-6
Internal
    Debug Request Enable bit 16-9, C-48
    debug request input 16-5
    RAM Supervisor Protect bit 7-12, C-21
    ROM
        disable 4-4
        Supervisor Protect bit 7-12, C-21
    ROM disable 7-2
Interrupt
    controller 10-1
        in low-power modes 8-5
        programming model 10-2
    Request Enable bit 15-4, C-24
    Request flag 12-9, C-34
    source assignments 10-5
    Source bits 10-3, C-2
    Source Register 10-2, C-2
Interrupt Request Enable bit 12-6, C-32
INTERVAL COUNT field 12-8, C-33
Interval mode 12-3, 12-10
    serial peripheral interface. *See* ISPI
Interval Mode Enable bit 12-8, C-33
Interval timer. *See* PIT
INTSRC 10-2, C-2
IR 16-18, C-52
IREN bit 11-11, C-39
IRQ bit 12-9, C-34
IRQ EN bit 15-4, C-24
IRQ_EN bit 12-6, C-32
IRTS bit 11-2, 11-12, C-41
ISPI 12-1
    block diagram 12-1
    clock 12-4
    Control Register 12-5, C-31
    Data Register 12-5, C-30
    Debug mode and 12-11
    Enable bit 12-6, C-31
    enable signal 12-4
    general-purpose output 12-4
    Interval Control Register 12-8, C-33
    interval mode 12-3, 12-10
    low-power modes and 8-5
    low-power operation 12-11
    manual mode 12-2, 12-9
    operation 12-1
    programming model 12-4, C-30
    signals 12-3
    slave mode 12-3, 12-10
    Status Register 12-8, C-34
    timing specifications A-6
ITADR 9-12, 9-16, C-13
ITCSR 9-14, C-11
ITDR 9-12, 9-15, C-12
ITIE bit 9-13, 9-14, C-12
ITIF bit 9-13, 9-14, C-12
IVL_EN bit 12-8, C-33

–J–

JTAG Test Access Port 16-1

–K–

KCDD bits 14-5, C-15
KCO bits 14-3, C-14
KDDR 14-5, C-15
KDIE bit 14-4, C-14
KDSC bit 14-4, C-15
Key Depress Interrupt Enable bit 14-4, C-14
Key Depress Synchronizer Clear bit 14-4, C-15
Key Release Interrupt Enable bit 14-4, C-14
Keypad
    Column Data Direction bits 14-5, C-15
    Column Strobe Open-Drain Enable bits 14-3, C-14
    Control Register 14-2, C-14
    Data Direction Register 14-5, C-15
    Data Register 14-5, C-15
    Key Depress bit 14-3, 14-4, C-15
    Key Release bit 14-3, 14-4, C-15
    Row Data Direction bits 14-5, C-15
    Row Enable bits 14-3, C-14
    Status Register 14-3, C-14
Keypad port. *See* KPP
KPCR 14-2, C-14
KPDR 14-5, C-15
KPKD bit 14-3, 14-4, C-15
KPKR bit 14-3, 14-4, C-15

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

KPP 14-1
block diagram 14-1
configuration 14-6
low-power modes and 8-6
matrix construction 14-6
matrix scanning 14-6
multiple key closures 14-8
operation 14-6
programming model 14-2, C-13
signals 14-2
standby 14-7
KPSR 14-3, C-14
KRDD bits 14-5, C-15
KRE bits 14-3, C-14
KRIE bit 14-4, C-14

–L–

Level-sensitive interrupts 13-2, C-27
Link register 2-3
Load and store 2-2
LOAD bit 15-4, C-24
Load PWMPR and PWMWR bit 15-4, C-24
LOOP bit 11-15, C-44
LOOP IR bit 11-15, C-44
Loop TX and RX for IR Test bit 11-15, C-44
Loop TX and RX for Test bit 11-15, C-44
Loopback bit 12-8, C-33
LOW_REFCLK 8-1
Low-power modes 8-1, 8-4
ISPI operation in 12-11
peripheral behavior 8-5
PWM operation in 15-8
UART operation in 11-23
Low-voltage reset 4-5, 8-6
LPBK bit 12-8, C-33
LPMD (CPU) signals 8-4
LVRSTIN bit 9-4, C-6
LVRSTIN signal 4-5, 8-6

–M–

M•CORE 2-1
architecture 2-2
bus interface 2-8
data format 2-5
instruction set 2-6
operand addressing 2-6
programming model 2-3
MAL 16-13, C-51
Manual mode 12-2, 12-9
Master in, slave out 12-3
Master Mode bit 12-6, C-32
Master out, slave in 12-4
Maximum ratings A-1
MBCA 16-14, C-51
MBCB 16-14, C-51
MBO bit 16-11, C-50
Memory
Address Latch 16-13, C-51
Breakpoint 16-12
B, A Control bits 16-10, C-49
B, A Range Control bits 16-10, C-49
Counters 16-14, C-51
Occurrence bit 16-11, C-50
load and store 2-2
map
MMC2001 3-1
peripheral modules 3-2, C-1
organization 2-5
MFCR instruction 2-4
MMC2001
memory
map 3-1
signals 4-1
MOD
timing specifications A-2
MOD signal 4-4, 7-2, 7-6
MODE bit 15-5, C-24
Modules
address map 3-2, C-1
interface operation 3-2
Move from Control Register 2-4
Move to Control Register 2-4
MSTR bit 12-6, C-32
MTCR instruction 2-4
Multiply 2-2

–N–

NIER 10-3, C-2
NIPND 10-4, C-4
Normal Interrupt
Enable Register 10-3, C-2
Pending Flags 10-4, C-4
Pending Register 10-4, C-4
NP flags 10-4, C-4

–O–

OCMR 16-6, C-46
OCR 16-8, C-47
ODEC 16-12
OE Assert bit 7-10, C-19
OE signal 4-4, 7-2
OEA bit 7-10, C-19
OnCE 16-1
block diagram 16-1
Command Register 16-6, C-46
commands 16-21
Control Register 16-8, C-47
controller 16-2, 16-5
debug output 16-6
decoder 16-12
interface signals 16-5
operation 16-1
serial interface 16-5
signals 16-3

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Status Register 16-11, C-50
timing specifications A-9
trace logic 16-14
On-Chip Emulation module. *See* OnCE
OSR 16-11, C-50
OTC 16-15
Output enable 4-4, 7-2
Overrun error 11-5
Overrun flag 12-9, C-34
OVR bit 12-9, C-34
OVRRUN bit 11-7, C-36
OVW bit 9-14, C-12

–P–

PA bit 7-11, C-20
Parity
Enable bit 11-12, C-41
error 11-5
Error bit 11-8, C-37
Odd/Even bit 11-13, C-41
PC 2-2, 2-3, 16-18, C-51
PC FIFO 16-20
PD bits 11-17, C-45
PERIOD field 15-6, C-26
Peripheral modules
address map 3-2, C-1
interface operation 3-2
PHA bit 12-7, C-32
Phase bit 12-7, C-32
Pin Assert bit 7-11, C-20
Pipeline information 16-17
Pipeline, execution 2-2
PIT 9-11
Alternate Data Register 9-12, 9-16, C-13
as a "free-running" timer 9-13
as a "set-and-forget" timer 9-12
block diagram 9-12
Control/Status Register 9-14, C-11
Data Register 9-12, 9-15, C-12
Enable bit 9-15, C-12
in Debug mode 9-16
Interrupt Enable bit 9-13, 9-14, C-12
Interrupt Flag 9-13, 9-14, C-12
low-power modes and 9-16
operation 9-12
registers 9-13
PM field 16-12, C-51
POL bit 12-7, 15-5, C-24, C-32
Polarity bit 12-7, 15-5, C-24, C-32
POR bit 9-4, C-6
Port Data bits 11-17, C-45
Positive supply 4-9
Power and ground pins 4-9
Power-On Reset bit 9-4, C-6
Prefetch Size field 16-19, C-52
Prefetch Transfer Code field 16-19, C-53
PREN bit 11-12, C-41
PRERR bit 11-8, C-37
Prescaler, PWM 15-2
Privilege modes 2-3
supervisor 2-3
user 2-3
Processor Mode field 16-12, C-51
Processor Status Register 16-19, C-53
PROE bit 11-13, C-41
Program counter 2-2, 2-3
Program Counter Register 16-18, C-51
Programming model
edge port 13-2
EIM 7-7
interrupt controller 10-2
ISPI 12-4, C-30
KPP 14-2, C-13
M•CORE 2-3
PWM 15-2, C-22
timer/reset module 9-1, C-5
UART 11-5
watchdog timer 9-10
PSR 16-19, C-53
PSTAT signals 16-5
Pulse Period field 15-6, C-26
Pulse Width field 15-7, C-26
Pulse width modulator. *See* PWM
PWM 15-1
block diagram 15-1
Control Register 15-4, C-23
Counter Register 15-7, C-27
Data bit 15-5, C-24
generating audio 15-1
Interrupt Request bit 15-4, C-24
IRQ bit 15-4, C-24
low-power modes and 8-5, 15-8
Mode bit 15-5, C-24
operating range 15-8
Period Register 15-6, C-25
prescaler 15-2
programming model 15-2, C-22
Width Register 15-7, C-26
PWM signals 4-8
PWMCR 15-4, 15-7, C-23, C-27
PWMPR 15-6, C-25
PWMWR 15-7, C-26

–R–

R/$\overline{W}$ bit 16-7, C-46
R/$\overline{W}$ signal 4-4, 7-2
R0 (stack pointer) 2-3, 2-6
R15 (link register) 2-3
RCA bit 16-10, C-49
RCB bit 16-10, C-49
Read/write 7-2
enable 4-4
Read/Write Command bit 16-7, C-46
Receive data 4-7, 11-3
Receive Data Register 12-5, C-30
Received Data bits 11-8, C-37
Receiver 11-3
Enable bit 11-10, C-39

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

FIFO Interrupt Trigger Level bits 11-10, C-39
Overrun bit 11-7, C-36
Ready Interrupt Enable bit 11-10, C-39
Ready Interrupt flag 11-14, C-43
Register Select field 16-7, C-46
Request to send 4-8, 11-2
Reset 4-4
block diagram 9-2
out 4-5
pins 9-2
sequence 9-3
Source/Chip Configuration Register 9-3, C-6
sources 9-2
timing specifications A-2
Return
from Exception 2-3
from Fast Interrupt 2-3
RFI instruction 2-3
RISC 2-1
RLD bit 9-12, 9-13, 9-15, C-12
ROM module 5-1
Row senses 4-7
RRDY bit 11-14, C-43
RRDYEN bit 11-10, C-39
RS field 16-7, C-46
RSCR 9-3, C-6
RST bit 9-4, C-6
RSTIN signal 4-4, 9-4, C-6
RSTOUT signal 4-5
RTE instruction 2-3
RTS Delta bit 11-15, C-43
RTS Delta Interrupt Enable bit 11-11, C-39
RTS Pin Status bit 11-14, C-43
RTS signal 11-2
RTS0 4-8
RTSD bit 11-15, C-43
RTSD EN bit 11-11, C-39
RTSS bit 11-14, C-43
Run mode 8-4
RX DATA bits 11-8, C-37
Rx DATA register 12-5, C-30
RxD signals 4-7, 11-3
RXEN bit 11-10, C-39
RxFL bit 11-10, C-39

–S–

S bit 2-3
Scratch registers 2-3, 2-4
Send Break bit 11-11, C-40
Sequential Breakpoint A, B Arm Occurrence bits 16-12, C-51
Sequential Control field 16-9, C-48
Serial protocol, OnCE 16-21
Shadow registers 2-3, 2-4
SHEN bits 7-7, 7-12, C-22
Show Cycle Enable bits 7-7, 7-12, C-22
Signals 4-1
edge port 13-1
EIM 7-1
ISPI 12-3
KPP 14-2
M•CORE bus 2-9
OnCE 16-3
UART 11-2
Slave mode 12-3, 12-10
SNDBRK bit 11-11, C-40
SNS bit 12-6, C-31
Software Debug Occurrence bit 16-11, C-50
SP bit 7-6, 7-10, C-19
SPCR 12-5, C-31
SPDR 12-5, C-30
SPI
data master in/slave out 4-8
data master out/slave in 4-8
enable 4-8
general-purpose output 4-8
serial clock 4-8
SPI_CLK signal 4-8, 12-4
SPI_EN bit 12-6, C-31
SPI_EN Sense bit 12-6, C-31
SPI_EN signal 4-8, 12-4
SPI_GP Control bit 12-7, C-32
SPI_GP signal 4-8, 12-4
SPI_MISO signal 4-8, 12-3
SPI_MOSI signal 4-8, 12-4
SPICR 12-8, C-33
SPIGP bit 12-7, C-32
SPRAM bit 7-12, C-21
SPROM bit 7-12, C-21
SPSR 12-8, C-34
SQA bit 16-12, C-51
SQB bit 16-11, C-50
SQC field 16-9, C-48
SRAM 6-1
Stack pointer 2-3, 2-6
Standby battery power 4-9
Standby mode 8-6
Standby power filter 4-9
Start bit 11-4
Static RAM 6-1
Status registers 2-3
Stop bit 11-4
STOP bit (PIT) 9-14, C-11
Stop bits 11-13, C-41
Stop mode 8-4
STPB bit 11-13, C-41
Subroutine calls 2-3
Supervisor mode 2-3
Supervisor Protect bit 7-6, 7-10, C-19
SWO bit 16-11, C-50
SZ field 16-19, C-52

–T–

TAP 16-1
TC field 16-19, C-53
TCK 4-5
TCK signal 16-3
TDI signal 4-5, 16-3

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

TDO signal 4-6, 16-3
TEA signal 7-6
TEST 4-6
Test
clock 4-5
data input 4-5
data output 4-6
mode select 4-6
reset 4-6, 16-4
Time-of-day timer. *See* TOD
Timer/reset module 9-1
programming model 9-1, C-5
TME bit 16-9, C-48
TMS signal 4-6, 16-3
TO bit 16-11, C-50
TOD 9-4
Control/Status Register 9-5, C-7
Fraction Alarm Register 9-5, 9-7, C-9
Fraction Register 9-5, 9-6, C-8
low-power modes and 8-6, 9-5
Seconds Alarm Register 9-5, 9-7, C-8
Seconds Register 9-5, 9-6, C-8
TODFAR 9-5, 9-7, C-9
TODFR 9-5, 9-6, C-8
TODSAR 9-5, 9-7, C-8
TODSCR 9-5, C-7
TODSR 9-5, 9-6, C-8
Trace
Count Occurrence bit 16-11, C-50
Counter 16-15
logic, OnCE 16-14
Mode Enable bit 16-9, C-48
Transfer error acknowledge 7-6
Transmit data 4-7, 11-3
Transmit Data bits 11-9, C-38
Transmit Data Register 12-5, C-30
Transmitter 11-3
Empty bit 11-14, C-43
Enable bit 11-10, C-38
FIFO Interrupt Trigger Level bits 11-9, C-38
Ready Interrupt Enable bit 11-10, C-38
Ready Interrupt flag 11-14, C-43
TRDY bit 11-14, C-43
TRDYEN bit 11-10, C-38
TRST signal 4-6, 16-4
TX DATA bits 11-9, C-38
Tx DATA Register 12-5, C-30
TxD signals 4-7, 11-3
TXEN bit 11-10, C-38
TxFL bit 11-9, C-38
TXMPTY bit 11-14, C-43

–U–

UART 11-1
block diagram 11-2
BRG Register 11-13, C-42
Control Register 1 11-9, C-38
Control Register 2 11-11, C-40
Data Direction Register 11-16
Debug mode and 11-24
Enable bit 11-11, C-40
general-purpose I/O 11-16
low-power modes and 8-5, 11-23
Port Control Register 11-16, C-44
Port Data Register 11-17, C-45
programming model 11-5
Receiver Register 11-7, C-36
signals 11-2
Status Register 11-14, C-42
Test Register 11-15, C-43
Transmitter Register 11-8, C-37
UBRGR 11-13, C-42
UCR1 11-9, C-38
UCR2 11-11, C-40
UDDR 11-16
UPCR 11-16, C-44
UPDR 11-17, C-45
URX 11-7, C-36
User mode 2-3
USR 11-14, C-42
UTS 11-15, C-43
UTX 11-8, C-37

–V–

V_{BATT} 4-9
V_{DD} 4-9
V_{STBY} 4-9

–W–

Wait mode 8-4
Wait-State Control bit 7-8, C-17
Watchdog 7-6, 9-8
Control Register 9-10, C-9
Debug Enable bit 9-11, C-10
Debug mode and 9-10
Doze Enable bit 9-11, C-10
Doze mode and 9-9
Enable bit 9-11, C-10
low-power modes and 8-6
programming model 9-10
Reset bit 9-4, 9-11, C-6, C-10
reset, following 9-9
service operation 9-9
Service Register 9-11, C-10
Stop Enable bit 9-11, C-10
Stop mode and 9-9
Time-out field 9-10, C-10
Wait mode and 9-9
WBBR 16-19, C-53
WCR 9-10, C-9
WDBG bit 9-11, C-10
WDE bit 9-11, C-10
WDR bit 9-4, 9-11, C-6, C-10
WDZE bit 9-11, C-10
WEN bit 7-10, C-19
WIDTH field 15-7, C-26

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Word Size bit 11-13, C-42
WP bit 7-6, 7-11, C-20
Write Protect bit 7-6, 7-11, C-20
Write Wait State bit 7-9, C-18
Write-Back Bus Register 16-17, 16-19, C-53
WS bit 11-13, C-42
WSC bit 7-8, C-17
WSR 9-11, C-10
WSTP bit 9-11, C-10
WT field 9-10, C-10
WWS bit 7-9, C-18

–X–

XCH bit 12-9, C-34
XOSCpin 4-5

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

RECORD OF CHANGES

Revision	Date	Description
1.0	15 JAN 98	Original
1.1	15 OCT 98	Section 1.1 Changed number of address lines in features list from 20 to 22 Section 11.2.1 Rewrote to clarify function of $\overline{\text{RTS}}$ pin Section 11.2.2 Rewrote to clarify function of $\overline{\text{CTS}}$ pin

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

This manual is a product of the Motorola M•Core Technology Center Design Documentation team. Technical writing, illustration, and production editing performed with Adobe® FrameMaker® running on multiple platforms. Cover graphic design by Bazzirk, Inc. of Austin, Texas. Printed by Imperial Lithographics, Phoenix, Arizona.

**For More Information On This Product,
Go to: www.freescale.com**