THE DATASHEET OF

MC68HC16Z1MEH16

www.win-source.net 0086-755-83957316

Order this document by
MC68HC16ZUM/AD

HC16

M68HC16Z Series

MC68HC16Z1
MC68CK16Z1
MC68CM16Z1
MC68HC16Z2
MC68HC16Z3
MC68HC16Z4
MC68CK16Z4

User's Manual

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

© Motorola, Inc., 1997

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

TABLE OF CONTENTS

Paragraph **Title** **Page**

SECTION 1
INTRODUCTION

SECTION 2
NOMENCLATURE

2.1 Symbols and Operators 2-1
2.2 CPU16 Register Mnemonics 2-2
2.3 Register Mnemonics 2-3
2.4 Conventions 2-6

SECTION 3
OVERVIEW

3.1 M68HC16 Z-Series MCU Features 3-1
3.1.1 Central Processor Unit (CPU16/CPU16L) 3-1
3.1.2 System Integration Module (SIM/SIML) 3-1
3.1.3 Standby RAM (SRAM) 3-1
3.1.4 Masked ROM Module (MRM) — (MC68HC16Z2/Z3 Only) 3-2
3.1.5 Analog-to-Digital Converter (ADC) 3-2
3.1.6 Queued Serial Module (QSM) 3-2
3.1.7 Multichannel Communication Interface (MCCI) — (MC68HC16Z4/CKZ4 Only) 3-2
3.1.8 General-Purpose Timer (GPT) 3-2
3.2 Intermodule Bus 3-2
3.3 System Block Diagram and Pin Assignment Diagrams 3-2
3.4 Pin Descriptions 3-11
3.5 Signal Descriptions 3-13
3.6 Internal Register Map 3-16
3.7 Address Space Maps 3-19

SECTION 4
CENTRAL PROCESSOR UNIT

4.1 General 4-1
4.2 Register Model 4-1
4.2.1 Accumulators 4-3
4.2.2 Index Registers 4-3
4.2.3 Stack Pointer 4-3
4.2.4 Program Counter 4-3
4.2.5 Condition Code Register 4-4
4.2.6 Address Extension Register and Address Extension Fields 4-5

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

Freescale Semiconductor, Inc.

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

4.2.7 Multiply and Accumulate Registers 4-5
4.3 Memory Management 4-5
4.3.1 Address Extension 4-6
4.3.2 Extension Fields 4-6
4.4 Data Types 4-6
4.5 Memory Organization 4-7
4.6 Addressing Modes 4-8
4.6.1 Immediate Addressing Modes 4-9
4.6.2 Extended Addressing Modes 4-10
4.6.3 Indexed Addressing Modes 4-10
4.6.4 Inherent Addressing Mode 4-10
4.6.5 Accumulator Offset Addressing Mode 4-10
4.6.6 Relative Addressing Modes 4-10
4.6.7 Post-Modified Index Addressing Mode 4-10
4.6.8 Use of CPU16 Indexed Mode to Replace M68HC11 Direct Mode ..4-11
4.7 Instruction Set 4-11
4.7.1 Instruction Set Summary 4-11
4.8 Comparison of CPU16 and M68HC11 CPU Instruction Sets 4-31
4.9 Instruction Format 4-33
4.10 Execution Model 4-34
4.10.1 Microsequencer 4-35
4.10.2 Instruction Pipeline 4-35
4.10.3 Execution Unit 4-35
4.11 Execution Process 4-36
4.11.1 Changes in Program Flow 4-36
4.12 Instruction Timing 4-36
4.13 Exceptions 4-37
4.13.1 Exception Vectors 4-37
4.13.2 Exception Stack Frame 4-38
4.13.3 Exception Processing Sequence 4-39
4.13.4 Types of Exceptions 4-39
4.13.4.1 Asynchronous Exceptions 4-39
4.13.4.2 Synchronous Exceptions 4-39
4.13.5 Multiple Exceptions 4-40
4.13.6 RTI Instruction 4-40
4.14 Development Support 4-40
4.14.1 Deterministic Opcode Tracking 4-40
4.14.1.1 IPIPE0/IPIPE1 Multiplexing 4-41
4.14.1.2 Combining Opcode Tracking with Other Capabilities 4-41
4.14.2 Breakpoints 4-41
4.14.3 Opcode Tracking and Breakpoints 4-42

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

Freescale Semiconductor, Inc.

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

4.14.4 Background Debug Mode 4-42
4.14.4.1 Enabling BDM 4-42
4.14.4.2 BDM Sources 4-42
4.14.4.3 Entering BDM 4-42
4.14.4.4 BDM Commands 4-43
4.14.4.5 Returning from BDM 4-43
4.14.4.6 BDM Serial Interface 4-44
4.15 Recommended BDM Connection 4-45
4.16 Digital Signal Processing 4-45

SECTION 5
SYSTEM INTEGRATION MODULE

5.1 General 5-1
5.2 System Configuration 5-2
5.2.1 Module Mapping 5-2
5.2.2 Interrupt Arbitration 5-3
5.2.3 Show Internal Cycles 5-3
5.2.4 Register Access 5-3
5.2.5 Freeze Operation 5-3
5.3 System Clock 5-4
5.3.1 Clock Sources 5-5
5.3.2 Clock Synthesizer Operation 5-6
5.3.3 External Bus Clock 5-21
5.3.4 Low-Power Operation 5-21
5.4 System Protection 5-24
5.4.1 Reset Status 5-24
5.4.2 Bus Monitor 5-24
5.4.3 Halt Monitor 5-25
5.4.4 Spurious Interrupt Monitor 5-25
5.4.5 Software Watchdog 5-25
5.4.6 Periodic Interrupt Timer 5-27
5.4.7 Interrupt Priority and Vectoring 5-28
5.4.8 Low-Power STOP Operation 5-29
5.5 External Bus Interface 5-29
5.5.1 Bus Control Signals 5-31
5.5.1.1 Address Bus 5-31
5.5.1.2 Address Strobe 5-31
5.5.1.3 Data Bus 5-31
5.5.1.4 Data Strobe 5-31
5.5.1.5 Read/Write Signal 5-32
5.5.1.6 Size Signals 5-32

Freescale Semiconductor, Inc.

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

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

5.5.1.7 Function Codes 5-32
5.5.1.8 Data Size Acknowledge Signals 5-32
5.5.1.9 Bus Error Signal 5-33
5.5.1.10 Halt Signal 5-33
5.5.1.11 Autovector Signal 5-33
5.5.2 Dynamic Bus Sizing 5-33
5.5.3 Operand Alignment 5-35
5.5.4 Misaligned Operands 5-35
5.5.5 Operand Transfer Cases 5-35
5.6 Bus Operation 5-36
5.6.1 Synchronization to CLKOUT 5-36
5.6.2 Regular Bus Cycle 5-37
5.6.2.1 Read Cycle 5-37
5.6.2.2 Write Cycle 5-38
5.6.3 Fast Termination Cycles 5-39
5.6.4 CPU Space Cycles 5-40
5.6.4.1 Breakpoint Acknowledge Cycle 5-41
5.6.4.2 LPSTOP Broadcast Cycle 5-42
5.6.5 Bus Exception Control Cycles 5-43
5.6.5.1 Bus Errors 5-44
5.6.5.2 Double Bus Faults 5-45
5.6.5.3 Halt Operation 5-45
5.6.6 External Bus Arbitration 5-46
5.6.6.1 Show Cycles 5-47
5.7 Reset 5-48
5.7.1 Reset Exception Processing 5-48
5.7.2 Reset Control Logic 5-48
5.7.3 Reset Mode Selection 5-49
5.7.3.1 Data Bus Mode Selection 5-50
5.7.3.2 Clock Mode Selection 5-52
5.7.3.3 Breakpoint Mode Selection 5-52
5.7.4 MCU Module Pin Function During Reset 5-52
5.7.5 Pin State During Reset 5-53
5.7.5.1 Reset States of SIM Pins 5-54
5.7.5.2 Reset States of Pins Assigned to Other MCU Modules 5-54
5.7.6 Reset Timing 5-55
5.7.7 Power-On Reset 5-55
5.7.8 Use of the Three-State Control Pin 5-56
5.7.9 Reset Processing Summary 5-57
5.7.10 Reset Status Register 5-57
5.8 Interrupts 5-58

Freescale Semiconductor, Inc.

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

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

5.8.1 Interrupt Exception Processing 5-58
5.8.2 Interrupt Priority and Recognition 5-58
5.8.3 Interrupt Acknowledge and Arbitration 5-59
5.8.4 Interrupt Processing Summary 5-60
5.8.5 Interrupt Acknowledge Bus Cycles 5-61
5.9 Chip-Selects 5-61
5.9.1 Chip-Select Registers 5-63
5.9.1.1 Chip-Select Pin Assignment Registers 5-64
5.9.1.2 Chip-Select Base Address Registers 5-65
5.9.1.3 Chip-Select Option Registers 5-66
5.9.1.4 PORTC Data Register 5-67
5.9.2 Chip-Select Operation 5-67
5.9.3 Using Chip-Select Signals for Interrupt Acknowledge 5-68
5.9.4 Chip-Select Reset Operation 5-69
5.10 Parallel Input/Output Ports 5-70
5.10.1 Pin Assignment Registers 5-70
5.10.2 Data Direction Registers 5-70
5.10.3 Data Registers 5-71
5.11 Factory Test 5-71

SECTION 6
STANDBY RAM MODULE

6.1 SRAM Register Block 6-1
6.2 SRAM Array Address Mapping 6-2
6.3 SRAM Array Address Space Type 6-2
6.4 Normal Access 6-2
6.5 Standby and Low-Power Stop Operation 6-2
6.6 Reset 6-3

SECTION 7
MASKED ROM MODULE

7.1 MRM Register Block 7-1
7.2 MRM Array Address Mapping 7-1
7.3 MRM Array Address Space Type 7-2
7.4 Normal Access 7-2
7.5 Low-Power Stop Mode Operation 7-3
7.6 ROM Signature 7-3
7.7 Reset 7-3

Freescale Semiconductor, Inc.

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

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

SECTION 8
ANALOG-TO-DIGITAL CONVERTER

8.1 General 8-1
8.2 External Connections 8-1
8.2.1 Analog Input Pins 8-2
8.2.2 Analog Reference Pins 8-3
8.2.3 Analog Supply Pins 8-3
8.3 Programmer's Model 8-3
8.4 ADC Bus Interface Unit 8-3
8.5 Special Operating Modes 8-3
8.5.1 Low-Power Stop Mode 8-3
8.5.2 Freeze Mode 8-4
8.6 Analog Subsystem 8-4
8.6.1 Multiplexer 8-4
8.6.2 Sample Capacitor and Buffer Amplifier 8-5
8.6.3 RC DAC Array 8-5
8.6.4 Comparator 8-6
8.7 Digital Control Subsystem 8-6
8.7.1 Control/Status Registers 8-6
8.7.2 Clock and Prescaler Control 8-6
8.7.3 Sample Time 8-7
8.7.4 Resolution 8-7
8.7.5 Conversion Control Logic 8-7
8.7.5.1 Conversion Parameters 8-8
8.7.5.2 Conversion Modes 8-8
8.7.6 Conversion Timing 8-12
8.7.7 Successive Approximation Register 8-13
8.7.8 Result Registers 8-13
8.8 Pin Considerations 8-14
8.8.1 Analog Reference Pins 8-14
8.8.2 Analog Power Pins 8-14
8.8.3 Analog Supply Filtering and Grounding 8-16
8.8.4 Accommodating Positive/Negative Stress Conditions 8-18
8.8.5 Analog Input Considerations 8-19
8.8.6 Analog Input Pins 8-21
8.8.6.1 Settling Time for the External Circuit 8-22
8.8.6.2 Error Resulting from Leakage 8-23

Freescale Semiconductor, Inc.

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

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

SECTION 9
QUEUED SERIAL MODULE

9.1 General 9-1
9.2 QSM Registers and Address Map 9-2
9.2.1 QSM Global Registers 9-2
9.2.1.1 Low-Power Stop Mode Operation 9-2
9.2.1.2 Freeze Operation 9-3
9.2.1.3 QSM Interrupts 9-3
9.2.2 QSM Pin Control Registers 9-4
9.3 Queued Serial Peripheral Interface 9-5
9.3.1 QSPI Registers 9-6
9.3.1.1 Control Registers 9-6
9.3.1.2 Status Register 9-7
9.3.2 QSPI RAM 9-7
9.3.2.1 Receive RAM 9-7
9.3.2.2 Transmit RAM 9-7
9.3.2.3 Command RAM 9-8
9.3.3 QSPI Pins 9-8
9.3.4 QSPI Operation 9-8
9.3.5 QSPI Operating Modes 9-9
9.3.5.1 Master Mode 9-16
9.3.5.2 Master Wrap-Around Mode 9-19
9.3.5.3 Slave Mode 9-20
9.3.5.4 Slave Wrap-Around Mode 9-21
9.3.6 Peripheral Chip Selects 9-21
9.4 Serial Communication Interface 9-21
9.4.1 SCI Registers 9-24
9.4.1.1 Control Registers 9-24
9.4.1.2 Status Register 9-24
9.4.1.3 Data Register 9-24
9.4.2 SCI Pins 9-25
9.4.3 SCI Operation 9-25
9.4.3.1 Definition of Terms 9-25
9.4.3.2 Serial Formats 9-25
9.4.3.3 Baud Clock 9-26
9.4.3.4 Parity Checking 9-26
9.4.3.5 Transmitter Operation 9-27
9.4.3.6 Receiver Operation 9-28
9.4.3.7 Idle-Line Detection 9-29
9.4.3.8 Receiver Wake-Up 9-29
9.4.3.9 Internal Loop Mode 9-30

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

Freescale Semiconductor, Inc.

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

SECTION 10
MULTICHANNEL COMMUNICATION INTERFACE

10.1 General 10-1
10.2 MCCI Registers and Address Map 10-2
10.2.1 MCCI Global Registers 10-2
10.2.1.1 Low-Power Stop Mode 10-2
10.2.1.2 Privilege Levels 10-3
10.2.1.3 MCCI Interrupts 10-3
10.2.2 Pin Control and General-Purpose I/O 10-4
10.3 Serial Peripheral Interface (SPI) 10-4
10.3.1 SPI Registers 10-6
10.3.1.1 SPI Control Register (SPCR) 10-6
10.3.1.2 SPI Status Register (SPSR) 10-6
10.3.1.3 SPI Data Register (SPDR) 10-6
10.3.2 SPI Pins 10-6
10.3.3 SPI Operating Modes 10-7
10.3.3.1 Master Mode 10-7
10.3.3.2 Slave Mode 10-8
10.3.4 SPI Clock Phase and Polarity Controls 10-8
10.3.4.1 CPHA = 0 Transfer Format 10-9
10.3.4.2 CPHA = 1 Transfer Format 10-10
10.3.5 SPI Serial Clock Baud Rate 10-11
10.3.6 Wired-OR Open-Drain Outputs 10-11
10.3.7 Transfer Size and Direction 10-11
10.3.8 Write Collision 10-12
10.3.9 Mode Fault 10-12
10.4 Serial Communication Interface (SCI) 10-13
10.4.1 SCI Registers 10-13
10.4.1.1 SCI Control Registers 10-13
10.4.1.2 SCI Status Register 10-16
10.4.1.3 SCI Data Register 10-16
10.4.2 SCI Pins 10-16
10.4.3 Receive Data Pins (RXDA, RXDB) 10-17
10.4.4 Transmit Data Pins (TXDA, TXDB) 10-17
10.4.5 SCI Operation 10-17
10.4.5.1 Definition of Terms 10-17
10.4.5.2 Serial Formats 10-18
10.4.5.3 Baud Clock 10-18
10.4.5.4 Parity Checking 10-19
10.4.5.5 Transmitter Operation 10-19
10.4.5.6 Receiver Operation 10-20

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

Freescale Semiconductor, Inc.

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

10.4.5.7 Idle-Line Detection 10-21
10.4.5.8 Receiver Wake-Up 10-22
10.4.5.9 Internal Loop 10-22
10.5 MCCI Initialization 10-23

SECTION 11
GENERAL-PURPOSE TIMER

11.1 General 11-1
11.2 GPT Registers and Address Map 11-2
11.3 Special Modes of Operation 11-3
11.3.1 Low-Power Stop Mode 11-3
11.3.2 Freeze Mode 11-3
11.3.3 Single-Step Mode 11-4
11.3.4 Test Mode 11-4
11.4 Polled and Interrupt-Driven Operation 11-4
11.4.1 Polled Operation 11-4
11.4.2 GPT Interrupts 11-5
11.5 Pin Descriptions 11-7
11.5.1 Input Capture Pins 11-7
11.5.2 Input Capture/Output Compare Pin 11-7
11.5.3 Output Compare Pins 11-7
11.5.4 Pulse Accumulator Input Pin 11-7
11.5.5 Pulse-Width Modulation 11-8
11.5.6 Auxiliary Timer Clock Input 11-8
11.6 General-Purpose I/O 11-8
11.7 Prescaler 11-8
11.8 Capture/Compare Unit 11-10
11.8.1 Timer Counter 11-10
11.8.2 Input Capture Functions 11-10
11.8.3 Output Compare Functions 11-13
11.8.3.1 Output Compare 1 11-14
11.8.3.2 Forced Output Compare 11-14
11.9 Input Capture 4/Output Compare 5 11-14
11.10 Pulse Accumulator 11-14
11.11 Pulse-Width Modulation Unit 11-16
11.11.1 PWM Counter 11-18
11.11.2 PWM Function 11-18

Freescale Semiconductor, Inc.

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

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

APPENDIX A
ELECTRICAL CHARACTERISTICS

APPENDIX B
MECHANICAL DATA AND ORDERING INFORMATION

B.1 Obtaining Updated M68HC16 Z-Series MCU Mechanical Information B-8
B.2 Ordering Information B-8

APPENDIX C
DEVELOPMENT SUPPORT

C.1 M68MMDS1632 Modular Development System C-1
C.2 M68MEVB1632 Modular Evaluation Board C-2

APPENDIX D
REGISTER SUMMARY

D.1 Central Processing Unit D-1
D.1.1 Condition Code Register D-3
D.2 System Integration Module D-4
D.2.1 SIM Module Configuration Register D-6
D.2.2 System Integration Test Register D-7
D.2.3 Clock Synthesizer Control Register D-7
D.2.4 Reset Status Register D-8
D.2.5 System Integration Test Register E D-9
D.2.6 Port E Data Register D-9
D.2.7 Port E Data Direction Register D-9
D.2.8 Port E Pin Assignment Register D-10
D.2.9 Port F Data Register D-10
D.2.10 Port F Data Direction Register D-11
D.2.11 Port F Pin Assignment Register D-11
D.2.12 System Protection Control Register D-12
D.2.13 Periodic Interrupt Control Register D-13
D.2.14 Periodic Interrupt Timer Register D-14
D.2.15 Software Watchdog Service Register D-15
D.2.16 Port C Data Register D-15
D.2.17 Chip-Select Pin Assignment Registers D-15
D.2.18 Chip-Select Base Address Register Boot D-17
D.2.19 Chip-Select Base Address Registers D-17
D.2.20 Chip-Select Option Register Boot D-18
D.2.21 Chip-Select Option Registers D-18

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

Freescale Semiconductor, Inc.

TABLE OF CONTENTS
(Continued)

Paragraph | Title | Page

D.2.22 Master Shift Registers D-22
D.2.23 Test Module Shift Count Register D-22
D.2.24 Test Module Repetition Count Register D-22
D.2.25 Test Module Control Register D-22
D.2.26 Test Module Distributed Register D-22
D.3 Standby RAM Module D-23
D.3.1 RAM Module Configuration Register D-23
D.3.2 RAM Test Register D-24
D.3.3 Array Base Address Register High D-24
D.3.4 Array Base Address Register Low D-24
D.4 Masked ROM Module D-25
D.4.1 Masked ROM Module Configuration Register D-25
D.4.2 ROM Array Base Address Registers D-27
D.4.3 ROM Signature Registers High D-27
D.4.4 ROM Bootstrap Words D-28
D.5 Analog-to-Digital Converter Module D-29
D.5.1 ADC Module Configuration Register D-30
D.5.2 ADC Test Register D-30
D.5.3 Port ADA Data Register D-30
D.5.4 ADC Control Register 0 ..... D-31
D.5.5 ADC Control Register 1 ..... D-32
D.5.6 ADC Status Register D-36
D.5.7 Right Justified, Unsigned Result Register D-36
D.6 Queued Serial Module D-38
D.6.1 QSM Configuration Register D-38
D.6.2 QSM Test Register D-39
D.6.3 QSM Interrupt Level Register/Interrupt Vector Register D-39
D.6.4 SCI Control Register D-40
D.6.5 SCI Control Register 1 ..... D-41
D.6.6 SCI Status Register D-43
D.6.7 SCI Data Register D-44
D.6.8 Port QS Data Register D-44
D.6.9 Port QS Pin Assignment Register/Data Direction Register D-45
D.6.10 QSPI Control Register 0 ..... D-46
D.6.11 QSPI Control Register 1 ..... D-48
D.6.12 QSPI Control Register 2 ..... D-49
D.6.13 QSPI Control Register 3 ..... D-50
D.6.14 Receive Data RAM D-51
D.6.15 Transmit Data RAM D-52
D.6.16 Command RAM D-52
D.7 Multichannel Communication Interface Module D-54

Freescale Semiconductor, Inc.

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

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

D.7.1 MCCI Module Configuration Register D-54
D.7.2 MCCI Test Register D-55
D.7.3 SCI Interrupt Level Register/MCCI Interrupt Vector Register D-55
D.7.4 MCCI Interrupt Vector Register D-56
D.7.5 SPI Interrupt Level Register D-56
D.7.6 MCCI Pin Assignment Register D-57
D.7.7 MCCI Data Direction Register D-58
D.7.8 MCCI Port Data Registers D-59
D.7.9 SCI Control Register 0 ..... D-59
D.7.11 SCI Status Register D-62
D.7.12 SCI Data Register D-63
D.7.13 SPI Control Register D-64
D.7.14 SPI Status Register D-65
D.7.15 SPI Data Register D-66
D.8 General-Purpose Timer D-67
D.8.1 GPT Module Configuration Register D-67
D.8.2 GPT Test Register D-68
D.8.3 GPT Interrupt Configuration Register D-68
D.8.4 Port GP Data Direction Register/Data Register D-69
D.8.5 OC1 Action Mask Register/Data Register D-69
D.8.6 Timer Counter Register D-70
D.8.7 Pulse Accumulator Control Register/Counter D-70
D.8.8 Input Capture Registers 1–3 ..... D-71
D.8.9 Output Compare Registers 1–4 ..... D-71
D.8.10 Input Capture 4/Output Compare 5 Register D-72
D.8.11 Timer Control Registers 1 and 2 ..... D-72
D.8.12 Timer Interrupt Mask Registers 1 and 2 ..... D-72
D.8.13 Timer Interrupt Flag Registers 1 and 2 ..... D-74
D.8.14 Compare Force Register/PWM Control Register C D-74
D.8.15 PWM Registers A/B D-76
D.8.16 PWM Count Register D-76
D.8.17 PWM Buffer Registers A/B D-76
D.8.18 GPT Prescaler D-77

APPENDIX E
INITIALIZATION AND PROGRAMMING EXAMPLES

E.1 Initialization Programs E-1
E.1.1 EQUATES.ASM E-2
E.1.2 ORG00000.ASM E-6
E.1.3 ORG00008.ASM E-6
E.1.4 INITSYS.ASM E-11

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

Freescale Semiconductor, Inc.

TABLE OF CONTENTS
(Continued)

Paragraph	Title	Page
E.1.5	INITRAM.ASM	E-11
E.1.6	INITSCI.ASM	E-12
E.2	Programming Examples	E-12
E.2.1	SIM Programming Examples	E-13
E.2.1.1	Example 1 - Using Ports E and F	E-13
E.2.1.2	Example 2 - Using Chip-Selects	E-14
E.2.1.3	Example 3 - Changing Clock Frequencies	E-16
E.2.1.4	Example 4 - Software Watchdog, Periodic Interrupt, and Autovector Demo	E-18
E.2.2	CPU16 Programming Example	E-23
E.2.2.1	Example 5 - Indexed and Extended Addressing	E-23
E.2.3	QSM/SCI Programming Example	E-24
E.2.3.1	Example 6 - Using an SCI Port	E-24
E.2.4	GPT Programming Example	E-25
E.2.4.1	Example 7 - Basic GPT Functions	E-25

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

LIST OF ILLUSTRATIONS

Figure	Title	Page
3-1	MC68HC16Z1/CK16Z1/CM16Z1 Block Diagram	3-4
3-2	MC68HC16Z2/Z3 Block Diagram	3-5
3-3	MC68HC16Z4/CK16Z4 Block Diagram	3-6
3-4	MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 Pin Assignments for 132-Pin Package	3-7
3-5	MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 Pin Assignments for 144-Pin Package	3-8
3-6	MC68HC16Z4/CKZ4 Pin Assignments for 132-Pin Package	3-9
3-7	MC68HC16Z4/CKZ4 Pin Assignments for 144-Pin Package	3-10
3-8	MC68HC16Z1/CKZ1/CMZ1 Address Map	3-17
3-9	MC68HC16Z2/Z3 Address Map	3-18
3-10	MC68HC16Z4/CKZ4 Address Map	3-18
3-11	MC68HC16Z1/CKZ1/CMZ1 Combined Program and Data Space Map	3-20
3-12	MC68HC16Z2/Z3 Combined Program and Data Space Map	3-21
3-13	MC68HC16Z4/CKZ4 Combined Program and Data Space Map	3-22
3-14	MC68HC16Z1/CKZ1/CMZ1 Separate Program and Data Space Map	3-23
3-15	MC68HC16Z2/Z3 Separate Program and Data Space Map	3-24
3-16	MC68HC16Z4/CKZ4 Separate Program and Data Space Map	3-25
4-1	CPU16 Register Model	4-2
4-2	Condition Code Register	4-4
4-3	Data Types and Memory Organization	4-8
4-4	Basic Instruction Formats	4-34
4-5	Instruction Execution Model	4-35
4-6	Exception Stack Frame Format	4-38
4-7	BDM Serial I/O Block Diagram	4-44
4-8	BDM Connector Pinout	4-45
5-1	System Integration Module Block Diagram	5-2
5-2	System Clock Block Diagram	5-4
5-3	Slow Reference Crystal Circuit	5-5
5-4	Fast Reference Crystal Circuit	5-5
5-5	System Clock Filter Networks	5-7
5-6	SIM LPSTOP Flowchart	5-22
5-7	SIML LPSTOP Flowchart	5-23
5-8	System Protection	5-24
5-9	Periodic Interrupt Timer and Software Watchdog Timer	5-27
5-10	MCU Basic System	5-30
5-11	Operand Byte Order	5-34

Freescale Semiconductor, Inc.

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

LIST OF ILLUSTRATIONS
(Continued)

Figure	Title	Page
5-12	Word Read Cycle Flowchart	5-38
5-13	Write Cycle Flowchart	5-39
5-14	CPU Space Address Encoding	5-41
5-15	Breakpoint Operation Flowchart	5-42
5-16	LPSTOP Interrupt Mask Level	5-43
5-17	Bus Arbitration Flowchart for Single Request	5-47
5-18	Preferred Circuit for Data Bus Mode Select Conditioning	5-50
5-19	Alternate Circuit for Data Bus Mode Select Conditioning	5-51
5-20	Power-On Reset	5-56
5-21	Basic MCU System	5-62
5-22	Chip-Select Circuit Block Diagram	5-63
5-23	CPU Space Encoding for Interrupt Acknowledge	5-68
8-1	ADC Block Diagram	8-2
8-2	8-Bit Conversion Timing	8-12
8-3	10-Bit Conversion Timing	8-13
8-4	Analog Input Circuitry	8-15
8-5	Errors Resulting from Clipping	8-16
8-6	Star-Ground at the Point of Power Supply Origin	8-17
8-7	Input Pin Subjected to Negative Stress	8-18
8-8	Voltage Limiting Diodes in a Negative Stress Circuit	8-19
8-9	External Multiplexing of Analog Signal Sources	8-20
8-10	Electrical Model of an A/D Input Pin	8-21
9-1	QSM Block Diagram	9-1
9-2	QSPI Block Diagram	9-5
9-3	QSPI RAM	9-7
9-4	Flowchart of QSPI Initialization Operation	9-10
9-5	Flowchart of QSPI Master Operation (Part 1)	9-11
9-6	Flowchart of QSPI Master Operation (Part 2)	9-12
9-7	Flowchart of QSPI Master Operation (Part 3)	9-13
9-8	Flowchart of QSPI Slave Operation (Part 1)	9-14
9-9	Flowchart of QSPI Slave Operation (Part 2)	9-15
9-10	SCI Transmitter Block Diagram	9-22
9-11	SCI Receiver Block Diagram	9-23
10-1	MCCI Block Diagram	10-1
10-2	SPI Block Diagram	10-5
10-3	CPHA = 0 SPI Transfer Format	10-9
10-4	CPHA = 1 SPI Transfer Format	10-10
10-5	SCI Transmitter Block Diagram	10-14

Freescale Semiconductor, Inc.

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

LIST OF ILLUSTRATIONS
(Continued)

Figure	Title	Page
10-6	SCI Receiver Block Diagram	10-15
11-1	GPT Block Diagram	11-2
11-2	Prescaler Block Diagram	11-9
11-3	Capture/Compare Unit Block Diagram	11-11
11-4	Input Capture Timing Example	11-13
11-5	Pulse Accumulator Block Diagram	11-15
11-6	PWM Block Diagram	11-17
A-1	CLKOUT Output Timing Diagram	A-28
A-2	External Clock Input Timing Diagram	A-28
A-3	ECLK Output Timing Diagram	A-28
A-4	Read Cycle Timing Diagram	A-29
A-5	Write Cycle Timing Diagram	A-30
A-6	Fast Termination Read Cycle Timing Diagram	A-31
A-7	Fast Termination Write Cycle Timing Diagram	A-32
A-8	Bus Arbitration Timing Diagram — Active Bus Case	A-33
A-9	Bus Arbitration Timing Diagram — Idle Bus Case	A-34
A-10	Show Cycle Timing Diagram	A-35
A-11	Chip-Select Timing Diagram	A-36
A-12	Reset and Mode Select Timing Diagram	A-36
A-13	Background Debug Mode Timing Diagram (Serial Communication)	A-39
A-14	Background Debug Mode Timing Diagram (Freeze Assertion)	A-39
A-15	ECLK Timing Diagram	A-44
A-16	QSPI Timing — Master, CPHA = 0	A-47
A-17	QSPI Timing — Master, CPHA = 1	A-47
A-18	QSPI Timing — Slave, CPHA = 0	A-48
A-19	QSPI Timing — Slave, CPHA = 1	A-48
A-20	SPI Timing — Master, CPHA = 0	A-51
A-21	SPI Timing — Master, CPHA = 1	A-51
A-22	SPI Timing — Slave, CPHA = 0	A-52
A-23	SPI Timing — Slave, CPHA = 1	A-52
A-24	Input Signal Conditioner Timing	A-53
A-25	Pulse Accumulator — Event Counting Mode (Leading Edge)	A-54
A-26	Pulse Accumulator — Gated Mode (Count While Pin High)	A-55
A-27	Pulse Accumulator — Using TOF as Gated Mode Clock	A-56
A-28	PWMx (PWMx Register = 01, Fast Mode)	A-56
A-29	Output Compare (Toggle Pin State)	A-57
A-30	Input Capture (Capture on Rising Edge)	A-58
A-31	General-Purpose Input	A-59
A-32	General-Purpose Output (Causes Input Capture)	A-60

Freescale Semiconductor, Inc.

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

LIST OF ILLUSTRATIONS
(Continued)

Figure	Title	Page
A-33	Force Compare (CLEAR)	A-61
A-34	Low Voltage 8-Bit ADC Conversion Accuracy	A-68
A-35	8-Bit ADC Conversion Accuracy	A-69
A-36	Low Voltage 10-Bit ADC Conversion Accuracy	A-70
A-37	10-Bit ADC Conversion Accuracy	A-71
B-1	MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 Pin Assignments for 132-Pin Package	B-2
B-2	MC68HC16Z4/CKZ4 Pin Assignments for 132-Pin Package	B-3
B-3	Case 831A-01 — 132-Pin Package Dimensions	B-4
B-4	MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 Pin Assignments for 144-Pin Package	B-5
B-5	MC68HC16Z4/CKZ4 Pin Assignments for 144-Pin Package	B-6
B-6	Case 918 — 144-Pin Package Dimensions	B-7
D-1	CPU16 Register Model	D-2

Freescale Semiconductor, Inc.

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

LIST OF TABLES

Table	Title	Page
1-1	M68HC16 Z-Series MCUs	1-1
1-2	Z-Series MCU Reference Frequencies	1-2
3-1	M68HC16 Z-Series Pin Characteristics	3-11
3-2	M68HC16 Z-Series Driver Types	3-12
3-3	M68HC16 Z-Series Power Connections	3-13
3-4	M68HC16 Z-Series Signal Characteristics	3-13
3-5	M68HC16 Z-Series Signal Function	3-15
4-1	Addressing Modes	4-9
4-2	Instruction Set Summary	4-12
4-3	Instruction Set Abbreviations and Symbols	4-30
4-4	CPU16 Implementation of M68HC11 CPU Instructions	4-32
4-5	Exception Vector Table	4-38
4-6	IPIPE0/IPIPE1 Encoding	4-41
4-7	Command Summary	4-43
5-1	Show Cycle Enable Bits	5-3
5-2	16.78-MHz Clock Control Multipliers	5-9
5-3	20.97-MHz Clock Control Multipliers	5-11
5-4	25.17-MHz Clock Control Multipliers	5-13
5-5	16.78-MHz System Clock Frequencies	5-15
5-6	System Clock Frequencies for a 20.97-MHz System	5-17
5-7	System Clock Frequencies for a 25.17-MHz System	5-19
5-8	Bus Monitor Period	5-25
5-9	MODCLK Pin and SWP Bit During Reset	5-26
5-10	Software Watchdog Divide Ratio	5-27
5-11	MODCLK Pin and PTP Bit at Reset	5-28
5-12	Periodic Interrupt Priority	5-29
5-13	Size Signal Encoding	5-32
5-14	Address Space Encoding	5-32
5-15	Effect of DSACK Signals	5-34
5-16	Operand Alignment	5-36
5-17	DSACK, BERR, and HALT Assertion Results	5-44
5-18	Reset Source Summary	5-49
5-19	Reset Mode Selection	5-49
5-20	Module Pin Functions	5-53
5-21	SIM Pin Reset States	5-54
5-22	Chip-Select Pin Functions	5-64
5-23	Pin Assignment Field Encoding	5-64
5-24	Block Size Encoding	5-65

Freescale Semiconductor, Inc.

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

LIST OF TABLES
(Continued)

Table	Title	Page
5-25	Chip-Select Base and Option Register Reset Values	5-69
5-26	CSBOOT Base and Option Register Reset Values	5-70
6-1	SRAM Configuration	6-1
6-2	SRAM Array Address Space Type	6-2
7-1	ROM Array Space Field	7-2
7-2	Wait States Field	7-3
8-1	FRZ Field Selection	8-4
8-2	Multiplexer Channel Sources	8-5
8-3	Prescaler Output	8-7
8-4	TS Field Selection	8-7
8-5	Conversion Parameters Controlled by ADCTL1	8-8
8-6	ADC Conversion Modes	8-8
8-7	Single-Channel Conversions (MULT = 0)	8-10
8-8	Multiple-Channel Conversions (MULT = 1)	8-11
8-9	Result Register Formats	8-14
8-10	External Circuit Settling Time (10-Bit Conversions)	8-23
8-11	Error Resulting From Input Leakage (IOFF)	8-23
9-1	Effect of DDRQS on QSM Pin Function	9-4
9-2	QSPI Pins	9-8
9-3	Bits Per Transfer	9-18
9-4	Serial Frame Formats	9-26
9-5	Effect of Parity Checking on Data Size	9-27
10-1	MCCI Interrupt Vectors	10-3
10-2	Pin Assignments	10-4
10-3	SPI Pin Functions	10-7
10-4	SCK Frequencies	10-11
10-5	SCI Pins	10-17
10-6	Serial Frame Formats	10-18
10-7	Effect of Parity Checking on Data Size	10-19
11-1	GPT Status Flags	11-5
11-2	GPT Interrupt Sources	11-6
11-3	PWM Frequency Ranges	11-18
A-1	Maximum Ratings	A-1
A-2	Typical Ratings, 2.7 to 3.6V, 16.78-MHz Operation	A-2

Freescale Semiconductor, Inc.

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

LIST OF TABLES
(Continued)

Table	Title	Page
A-3	Typical Ratings, 5V, 16.78-MHz Operation	A-3
A-4	Typical Ratings, 20.97-MHz Operation	A-3
A-5	Typical Ratings, 25.17-MHz	A-4
A-6	Thermal Characteristics	A-5
A-7	Low Voltage Clock Control Timing	A-6
A-8	16.78-MHz Clock Control Timing	A-7
A-9	20.97-MHz Clock Control Timing	A-8
A-10	25.17-MHz Clock Control Timing	A-9
A-11	Low Voltage 16.78-MHz DC Characteristics	A-10
A-12	16.78-MHz DC Characteristics	A-12
A-13	20.97-MHz DC Characteristics	A-14
A-14	25.17-MHz DC Characteristics	A-16
A-15	Low Voltage 16.78-MHz AC Timing	A-19
A-16	16.78-MHz AC Timing	A-21
A-17	20.97-MHz AC Timing	A-23
A-18	25.17-MHz AC Timing	A-25
A-19	Low Voltage 16.78-MHz Background Debug Mode Timing	A-37
A-20	16.78-MHz Background Debug Mode Timing	A-37
A-21	20.97-MHz Background Debug Mode Timing	A-38
A-22	25.17-MHz Background Debug Mode Timing	A-38
A-23	Low Voltage ECLK Bus Timing	A-40
A-24	16.78-MHz ECLK Bus Timing	A-41
A-25	20.97-MHz ECLK Bus Timing	A-42
A-26	25.17-MHz ECLK Bus Timing	A-43
A-27	Low Voltage QSPI Timing	A-45
A-28	QSPI Timing	A-46
A-29	Low Voltage SPI Timing	A-49
A-30	SPI Timing	A-50
A-31	General-Purpose Timer AC Characteristics	A-53
A-32	ADC Maximum Ratings	A-62
A-33	Low Voltage ADC DC Electrical Characteristics (Operating)	A-63
A-34	Low Voltage ADC AC Characteristics (Operating)	A-63
A-35	5V ADC DC Electrical Characteristics (Operating)	A-64
A-36	ADC AC Characteristics (Operating)	A-65
A-37	Low Voltage ADC Conversion Characteristics (Operating)	A-66
A-38	ADC Conversion Characteristics (Operating)	A-67
B-1	M68HC16 Z-Series Ordering Information	B-8
D-1	Module Address Map	D-1
D-2	SIM Address Map	D-4

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

Freescale Semiconductor, Inc.

LIST OF TABLES
(Continued)

Table	Title	Page
D-3	Show Cycle Enable Bits	D-6
D-4	Port E Pin Assignments	D-10
D-5	Port F Pin Assignments	D-11
D-6	Software Watchdog Divide Ratio	D-12
D-7	Bus Monitor Time-Out Period	D-13
D-8	Pin Assignment Field Encoding	D-16
D-9	CSPAR0 Pin Assignments	D-16
D-10	CSPAR1 Pin Assignments	D-17
D-11	Reset Pin Function of CS[10:6]	D-17
D-12	Block Size Field Bit Encoding	D-18
D-13	BYTE Field Bit Encoding	D-19
D-14	Read/Write Field Bit Encoding	D-19
D-15	DSACK Field Encoding	D-20
D-16	Memory Access Times at 16.78, 20.97, and 25.17 MHz	D-20
D-17	Address Space Bit Encodings	D-21
D-18	Interrupt Priority Level Field Encoding	D-21
D-19	SRAM Address Map	D-23
D-20	SRAM Array Address Space Type	D-23
D-21	MRM Address Map	D-25
D-22	ROM Array Space Field	D-26
D-23	Wait States Field	D-26
D-24	ADC Module Address Map	D-29
D-25	Freeze Encoding	D-30
D-26	Sample Time Selection	D-31
D-27	Prescaler Output	D-32
D-28	ADC Conversion Mode	D-33
D-29	Single-Channel Conversions (MULT = 0)	D-34
D-30	Multiple-Channel Conversions (MULT = 1)	D-35
D-31	QSM Address Map	D-38
D-32	Examples of SCI Baud Rates	D-41
D-33	PQSPAR Pin Assignments	D-45
D-34	Effect of DDRQS on QSM Pin Function	D-46
D-35	Bits Per Transfer	D-47
D-36	Examples of SCK Frequencies	D-48
D-37	MCCI Address Map	D-54
D-38	Interrupt Vector Sources	D-56
D-39	MPAR Pin Assignments	D-57
D-40	Effect of MDDR on MCCI Pin Function	D-58
D-41	Examples of SCI Baud Rates	D-60
D-42	GPT Address Map	D-67
D-43	GPT Interrupt Sources	D-69

Freescale Semiconductor, Inc.

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

LIST OF TABLES
(Continued)

Table	Title	Page
D-44	PAMOD and PEDGE Effects	D-71
D-45	PACLK[1:0] Effects	D-71
D-46	OM/OL[5:2] Effects	D-72
D-47	EDGE[4:1] Effects	D-72
D-48	CPR[2:0]/Prescaler Select Field	D-73
D-49	PPR[2:0] Field	D-75
D-50	PWM Frequency Ranges	D-76

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 1
INTRODUCTION

M68HC16 Z-series microcontrollers (including the MC68HC16Z1, MC68CM16Z1, MC68CK16Z1, MC68HC16Z2, MC68HC16Z3, MC68HC16Z4, and MC68CK16Z4) are high-speed 16-bit control units that are upwardly code compatible with M68HC11 controllers. All are members of the M68HC16 Family of modular microcontrollers.

M68HC16 microcontroller units (MCUs) are built from standard modules that interface via a common internal bus. Standardization facilitates rapid development of devices tailored for specific applications.

M68HC16 Z-series MCUs incorporate a number of different modules. Refer to **Table 1-1** for information on the contents of a specific Z-series MCU. (x) indicates that the module is used in the MCU. All of these modules are interconnected by the intermodule bus (IMB).

Table 1-1 M68HC16 Z-Series MCUs

Modules	MC68HC16Z1 MC68CK16Z1[1] MC68CM16Z1[1]	MC68HC16Z2	MC68HC16Z3	MC68HC16Z4 MC68CK16Z4[1]
Central Processor Unit (CPU16)	X	X	X	—
Low-Power Central Processor Unit (CPU16L)	—	—	—	X
System Integration Module (SIM)	X	X	X	—
Low-Power System Integration Module (SIML)	—	—	—	X
Standby RAM (SRAM)	1 Kbyte	2 Kbytes	4 Kbytes	1 Kbyte
Masked ROM Module (MRM)	—	8 Kbytes	8 Kbytes	—
Analog-to-Digital Converter (ADC)	X	X	X	X
Queued Serial Module (QSM)	X	X	X	—
Multichannel Communication Interface (MCCI)	—	—	—	X
General-Purpose Timer (GPT)	X	X	X	X

NOTES:
1. “C” designator indicates a 2.7V to 3.6V part; “M” indicates a fast reference frequency and “K” indicates a slow reference frequency. “HC” stands for HCMOS.

The maximum system clock for M68HC16 Z-series MCUs can be either 16.78 MHz, 20.97 MHz, or 25.17 MHz. An internal phase-locked loop circuit synthesizes the system clock from a slow (typically 32.768 kHz) or fast (typically 4.194 MHz) reference, or uses an external frequency source. Refer to **Table 1-2** for information on which reference frequency is applied to a particular MCU. (x) indicates the reference frequency applicable to the MCU.

Freescale Semiconductor, Inc.

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

Table 1-2 Z-Series MCU Reference Frequencies

MCU	Nominal Reference Frequency[1]	
	Slow (32.768 kHz)	Fast (4.194 MHz)
MC68HC16Z1	X	—
MC68CM16Z1	—	X
MC68CK16Z1	X	—
MC68HC16Z2	—	X
MC68HC16Z3	—	X
MC68HC16Z4	X	—
MC68CK16Z4	X	—

NOTES:
1. The nominal slow reference frequency is 32.768 kHz, but can range from 20 to 50 kHz. The nominal fast reference frequency is 4.194 MHz, but can range from 1MHz to 6.25 MHz.

System hardware and software allow changes in clock rate during operation. Because the MCUs are a fully static design, register and memory contents are not affected by clock rate changes.

High-density complementary metal-oxide semiconductor (HCMOS) architecture makes the basic power consumption low. Power consumption can be minimized by stopping the system clocks. The M68HC16 instruction set includes a low-power stop (LPSTOP) command that efficiently implements this capability. Individual stop bits in each module allow for selective power reduction.

Documentation for the Modular Microcontroller Family follows the modular construction of the devices in the product line. Each device has a comprehensive user's manual that provides sufficient information for normal operation of the device. The user's manual is supplemented by module reference manuals that provide detailed information about module operation and applications. Refer to Freescale publication *Advanced Microcontroller Unit (AMCU) Literature* (BR1116/D) for a complete list of documentation to supplement this manual.

Freescale Semiconductor, Inc.

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

SECTION 2
NOMENCLATURE

The following tables show the nomenclature used throughout the M68HC16 Z-series manual.

2.1 Symbols and Operators

Symbol	Function
+	Addition
-	Subtraction (two's complement) or negation
*	Multiplication
/	Division
>	Greater
<	Less
=	Equal
≥	Equal or greater
≤	Equal or less
≠	Not equal
•	AND
✣	Inclusive OR (OR)
⊕	Exclusive OR (EOR)
$\overline{\text{NOT}}$	Complementation
:	Concatenation
⇒	Transferred
⇔	Exchanged
±	Sign bit; also used to show tolerance
«	Sign extension
%	Binary value
$	Hexadecimal value

Freescale Semiconductor, Inc.

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

2.2 CPU16 Register Mnemonics

Mnemonic	Register
A	Accumulator A
AM	Accumulator M
B	Accumulator B
CCR	Condition code register
D	Accumulator D
E	Accumulator E
EK	Extended addressing extension field
HR	MAC multiplier register
IR	MAC multiplicand register
IX	Index register X
IY	Index register Y
IZ	Index register Z
K	Address extension register
PC	Program counter
PK	Program counter extension field
SK	Stack pointer extension field
SP	Stack pointer
XK	Index register X extension field
YK	Index register Y extension field
ZK	Index register Z extension field
XMSK	Modulo addressing index register X mask
YMSK	Modulo addressing index register Y mask
S	LPSTOP mode control bit
MV	AM overflow flag
H	Half carry flag
EV	AM extended overflow flag
N	Negative flag
Z	Zero flag
V	Two’s complement overflow flag
C	Carry/borrow flag
IP	Interrupt priority field
SM	Saturation mode control bit

Freescale Semiconductor, Inc.

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

2.3 Register Mnemonics

Mnemonic	Register
ADCMCR	ADC Module Configuration Register
ADCTEST	ADC Test Register
ADCTL[0:1]	ADC Control Registers [0:1]
ADCSTAT	ADC Status Register
CFORC	GPT Compare Force Register
CREG	SIM Test Module Control Register
CR[0:F]	QSM Command RAM [0:F]
CSBARBT	SIM Chip-Select Base Address Register Boot
CSBAR[0:10]	SIM Chip-Select Base Address Registers [0:10]
CSORBT	SIM Chip-Select Option Register Boot
CSOR[0:10]	SIM Chip-Select Option Registers [0:10]
CSPAR[0:1]	SIM Chip-Select Pin Assignment Registers [0:1]
DDRE	SIM Port E Data Direction Register
DDRF	SIM Port F Data Direction Register
DDRGP	GPT Port GP Data Direction Register
DDRM	MCCI Data Direction Register
DDRQS	QSM Port QS Data Direction Register
DREG	SIM Test Module Distributed Register
GPTMCR	GPT Module Configuration Register
GPTMTR	GPT Module Test Register
ICR	GPT Interrupt Configuration Register
ILSCI	MCCI SCI Interrupt Register
ILSPI	MCCI SPI Interrupt Register
LJSRR[0:7]	ADC Left-Justified Signed Result Registers [0:7]
LJURR[0:7]	ADC Left-Justified Unsigned Result Registers [0:7]
MIVR	MCCI Interrupt Vector Register
MMCR	MCCI Module Configuration Register
MPAR	MCCI Pin Assignment Register
MRMCR	Masked ROM Module Configuration Register
MTEST	MCCI Test Register
OC1D	GPT Output Compare 1 Action Data Register
OC1M	GPT Output Compare 1 Action Mask Register
PACNT	GPT Pulse Accumulator Counter Register
PACTL	GPT Pulse Accumulator Control Register
PEPAR	SIM Port E Pin Assignment Register
PFPAR	SIM Port F Pin Assignment Register

Freescale Semiconductor, Inc.

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

Mnemonic	Register
PICR	SIM Periodic Interrupt Control Register
PITR	SIM Periodic Interrupt Timer Register
PORTADA	ADC Port ADA Data Register
PORTC	SIM Port C Data Register
PORTE	SIM Port E Data Register [0:1]
PORTF	SIM Port F Data Register [0:1]
PORTGP	GPT Port GP Data Register
PORTMC	MCCI Port Data Register
PORTMCP	MCCI Port Pin State Register
PORTQS	QSM Port QS Data Register
PQSPAR	QSM Port QS Pin Assignment Register
PRESCL	GPT Prescaler Register
PWMA	GPT PWM Control Register A
PWMB	GPT PWM Control Register B
PWMBUFA	GPT PWM Buffer Register A
PWMBUFB	GPT PWM Buffer Register B
PWMC	GPT PWM Control Register C
PWMCNT	GPT PWM Counter Register
QILR	QSM Interrupt Level Register
QIVR	QSM Interrupt Vector Register
QSMCR	QSM Module Configuration Register
QTEST	QSM Test Register
RAMBAH	RAM Array Base Address Register High
RAMBAL	RAM Array Base Address Register Low
RAMMCR	RAM Module Configuration Register
RAMTST	RAM Test Register
RJURR[0:7]	ADC Right-Justified Unsigned Result Registers [0:7]
ROMBAH	ROM Array Base Address Register High
ROMBAL	ROM Array Base Address Register Low
ROMBS[0:3]	ROM Bootstrap Word Registers [0:3]
RR[0:F]	QSM Receive Data RAM [0:F]
RSR	SIM Reset Status Register
SCCR[0:1]	QSM SCI Control Registers [0:1]
SCCR0[A:B]	MCCI SCIA/B Control Registers 0 [A:B]
SCCR1[A:B]	MCCI SCIA/B Control Registers 1 [A:B]
SCDR	QSM SCI Data Register
SCDR[A:B]	MCCI SCIA/B Data Registers [A:B]

Freescale Semiconductor, Inc.

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

Mnemonic	Register
SCSR	QSM SCI Status Register
SCSR[A:B]	MCCI SCIA/B Status Registers [A:B]
SIGHI	ROM Signature Register High
SIGLO	ROM Signature Register Low
SIMCR	SIM Module Configuration Register
SIMTR	SIM Test Register
SIMTRE	SIM Test Register E
SPCR	MCCI SPI Control Register
SPCR[0:3]	QSM SPI Control Registers [0:3]
SPDR	MCCI SPI Data Register
SPSR	QSM SPI Status Register
SPSR	MCCI SPI Status Register
SWSR	SIM Software Watchdog Service Register
SYNCR	SIM Clock Synthesizer Control Register
SYPCR	SIM System Protection Control Register
TCNT	GPT Timer Counter Register
TCTL[1:2]	GPT Timer Control Registers [1:2]
TFLG[1:2]	GPT Timer Flag Registers [1:2]
TI4/O5	GPT Timer Input Capture 4/Output Compare 5 Register
TIC[1:3]	GPT Timer Input Capture Registers [1:3]
TMSK[1:2]	GPT Timer Mask Register [1:2]
TOC[1:4]	GPT Timer Output Compare Registers [1:4]
TR[0:F]	QSM Transmit RAM [0:F]
TSTMSRA	SIM Test Module Master Shift Register A
TSTMSRB	SIM Test Module Master Shift Register B
TSTRC	SIM Test Module Repetition Count Register
TSTSC	SIM Test Module Shift Count Register

Freescale Semiconductor, Inc.

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

2.4 Conventions

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.

Set refers specifically to establishing logic level one on a bit or bits.

Clear refers specifically to establishing logic level zero on a bit or bits.

Asserted means that a signal is in active logic state. An active low signal changes from logic level one to logic level zero when asserted, and an active high signal changes from logic level zero to logic level one.

Negated means that an asserted signal changes logic state. An active low signal changes from logic level zero to logic level one when negated, and an active high signal changes from logic level one to logic level zero.

A specific mnemonic within a range is referred to by mnemonic and number. A15 is bit 15 of accumulator A; ADDR7 is line 7 of the address bus; CSOR0 is chip-select option register 0. **A range of mnemonics** is referred to by mnemonic and the numbers that define the range. VBR[4:0] are bits four to zero of the vector base register; CSOR[0:5] are the first six chip-select option registers.

Parentheses are used to indicate the content of a register or memory location, rather than the register or memory location itself. For example, (A) is the content of accumulator A. (M : M + 1) is the content of the word at address M.

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

LSW means least significant word or words. **MSW** means most significant word or words.

ADDR is the address bus. ADDR[7:0] are the eight LSB of the address bus.

DATA is the data bus. DATA[15:8] are the eight MSB of the data bus.

Freescale Semiconductor, Inc.

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

SECTION 3
OVERVIEW

This section provides general information on M68HC16 Z-series MCUs. It lists features of each of the modules, shows device functional divisions and pin assignments, summarizes signal and pin functions, discusses the intermodule bus, and provides system memory maps. Timing and electrical specifications for the entire microcontroller and for individual modules are provided in **APPENDIX A ELECTRICAL CHARACTERISTICS**. Comprehensive module register descriptions and memory maps are provided in **APPENDIX D REGISTER SUMMARY**.

3.1 M68HC16 Z-Series MCU Features

The following paragraphs highlight capabilities of each of the MCU modules. Each module is discussed separately in a subsequent section of this manual.

3.1.1 Central Processor Unit (CPU16/CPU16L)

- 16-bit architecture
- Full set of 16-bit instructions
- Three 16-bit index registers
- Two 16-bit accumulators
- Control-oriented digital signal processing capability
- Addresses up to 1 Mbyte of program memory; 1 Mbyte of data memory
- Background debug mode
- Fully static operation
- Expanded LPSTOP operation on CPU16L (MC68HC16Z4, MC68CK16Z4 only)

3.1.2 System Integration Module (SIM/SIML)

- External bus support
- Programmable chip-select outputs
- System protection logic
- Watchdog timer, clock monitor, and bus monitor
- Two 8-bit dual function input/output ports
- One 7-bit dual function output port
- Phase-locked loop (PLL) clock system
- Expanded LPSTOP operation on SIML (MC68HC16Z4, MC68CK16Z4 only)

3.1.3 Standby RAM (SRAM)

- 1-Kbyte static RAM (MC68HC16Z1/Z4 only)
- 2-Kbyte static RAM (MC68HC16Z2 only)
- 4-Kbyte static RAM (MC68HC16Z3 only)
- External standby voltage supply input

Freescale Semiconductor, Inc.

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

3.1.4 Masked ROM Module (MRM) — (MC68HC16Z2/Z3 Only)

- 8-Kbyte array, accessible as bytes or words
- User-selectable default base address
- User-selectable bootstrap ROM function
- User-selectable ROM verification code

3.1.5 Analog-to-Digital Converter (ADC)

- Eight channels, eight result registers
- Eight automated modes
- Three result alignment modes

3.1.6 Queued Serial Module (QSM)

- Enhanced serial communication interface
- Queued serial peripheral interface
- One 8-bit dual function port

3.1.7 Multichannel Communication Interface (MCCI) — (MC68HC16Z4/CKZ4 Only)

- Two channels of enhanced SCI (UART)
- One channel of SPI

3.1.8 General-Purpose Timer (GPT)

- Two 16-bit free-running counters with prescaler
- Three input capture channels
- Four output compare channels
- One input capture/output compare channel
- One pulse accumulator/event counter input
- Two pulse width modulation outputs
- Optional external clock input

3.2 Intermodule Bus

The intermodule bus (IMB) is a standardized bus developed to facilitate the design of modular microcontrollers. It contains circuitry that supports exception processing, address space partitioning, multiple interrupt levels, and vectored interrupts. The standardized modules in M68HC16 Z-series MCUs communicate with one another via the IMB. Although the full IMB supports 24 address and 16 data lines, M68HC16 Z-series MCUs use only 20 address lines. ADDR[23:20] follow the state of ADDR19.

3.3 System Block Diagram and Pin Assignment Diagrams

Figure 3-1 is a functional diagram of the MC68HC16Z1/CKZ1/CMZ1 MCU. Refer to **Figure 3-2** for a functional diagram of the MC68HC16Z2/Z3 MCU. **Figure 3-3** is a functional diagram of the MC68HC16Z4/CKZ4 MCU. Although diagram blocks represent the relative size of the physical modules, there is not a one-to-one correspondence between location and size of blocks in the diagram and location and size of integrated-circuit modules.

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

M68HC16 Z-series microcontrollers are available in both 132- and 144-pin packages. **Figure 3-4** shows an MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 pin assignment drawing based on a 132-pin plastic surface-mount package. **Figure 3-5** shows an MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 pin assignment drawing based on a 144-pin plastic surface-mount package. **Figure 3-6** shows an MC68HC16Z4/CKZ4 pin assignment drawing based on a 132-pin plastic surface-mount package. **Figure 3-7** shows an MC68HC16Z4/CKZ4 pin assignment drawing based on a 144-pin plastic surface-mount package. Refer to **APPENDIX B MECHANICAL DATA AND ORDERING INFORMATION** for information on how to obtain package dimensions. Refer to subsequent paragraphs in this section for pin and signal descriptions.

Freescale Semiconductor, Inc.

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

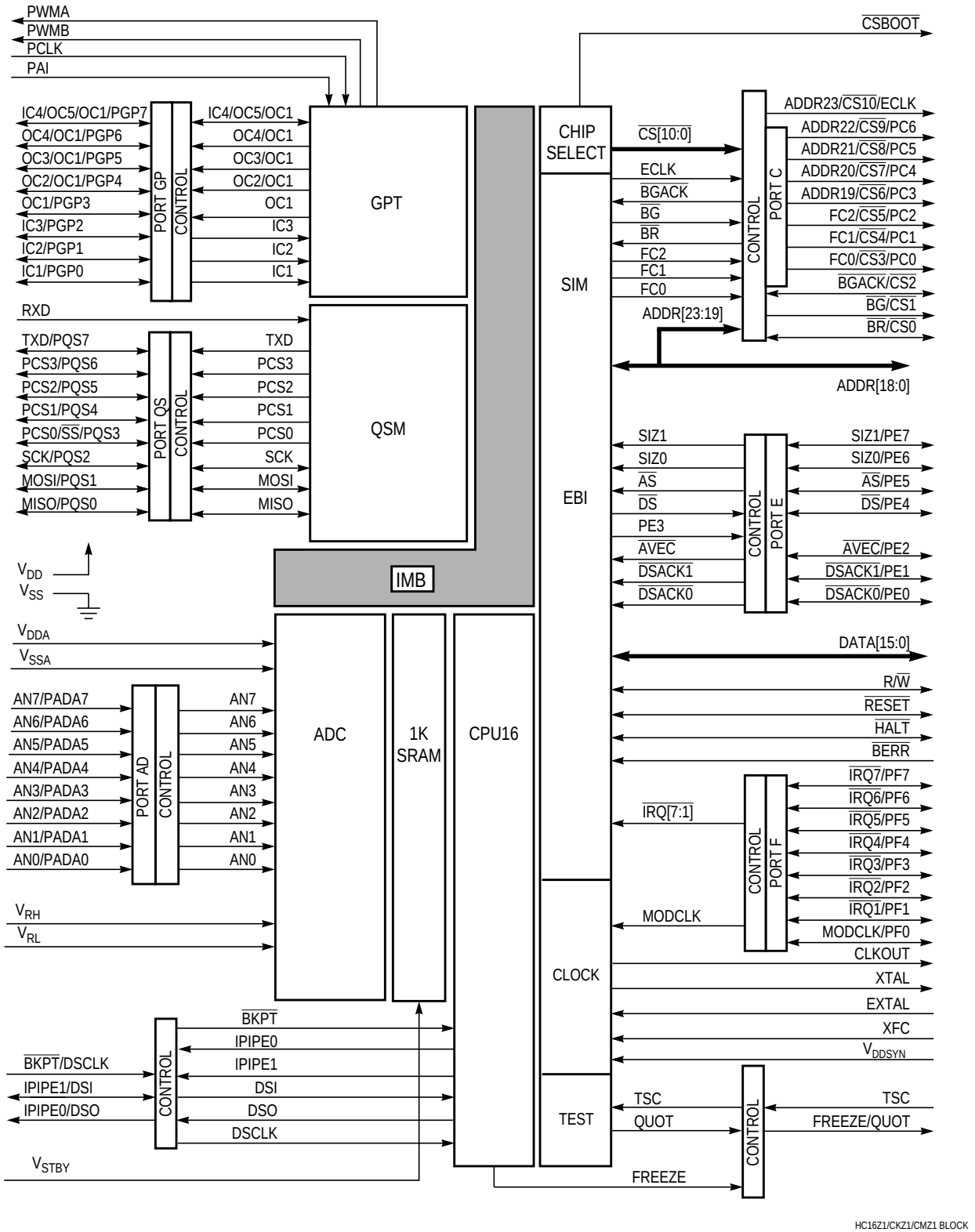


Figure 3-1 MC68HC16Z1/CK16Z1/CM16Z1 Block Diagram

Freescale Semiconductor, Inc.

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

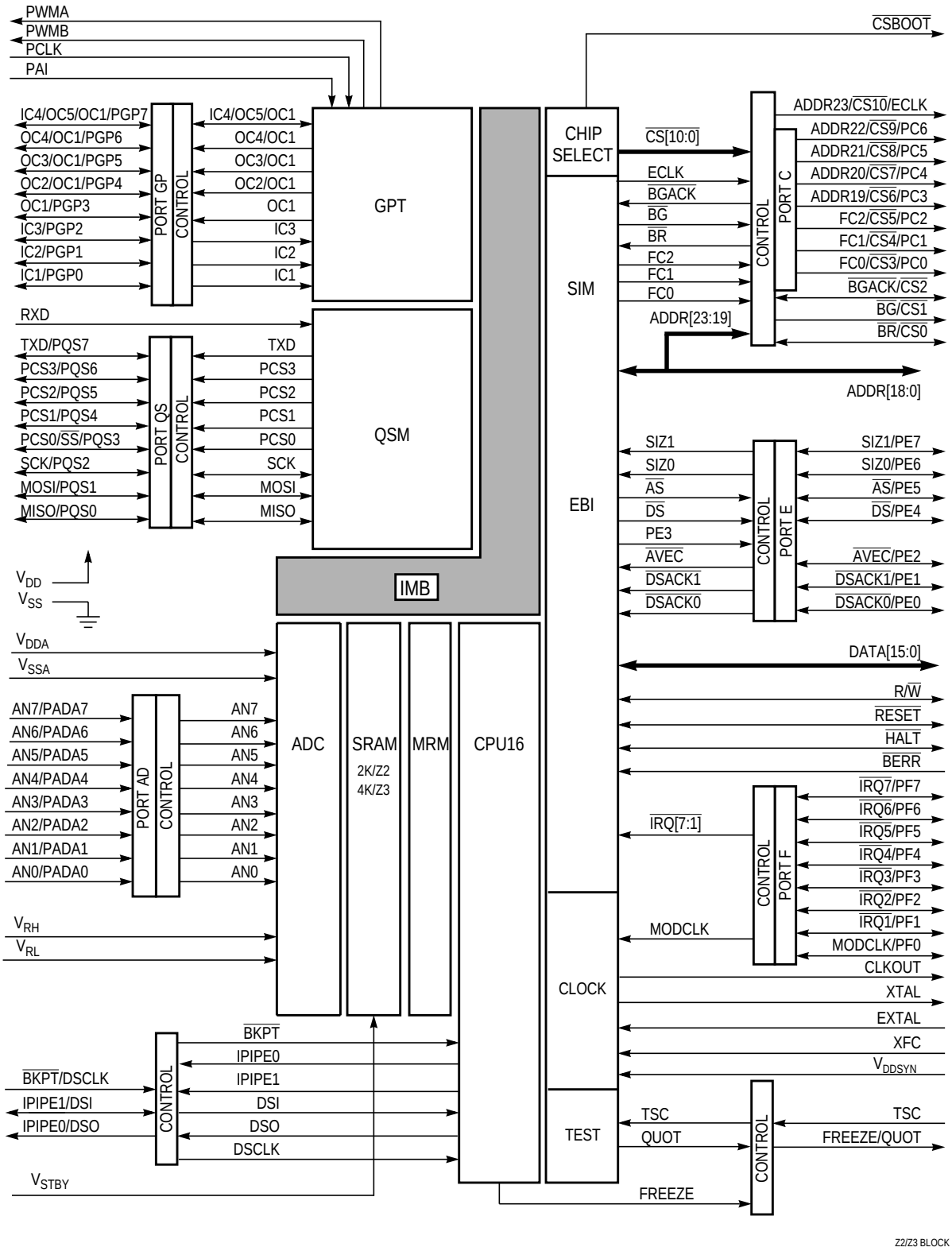


Figure 3-2 MC68HC16Z2/Z3 Block Diagram

Freescale Semiconductor, Inc.

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

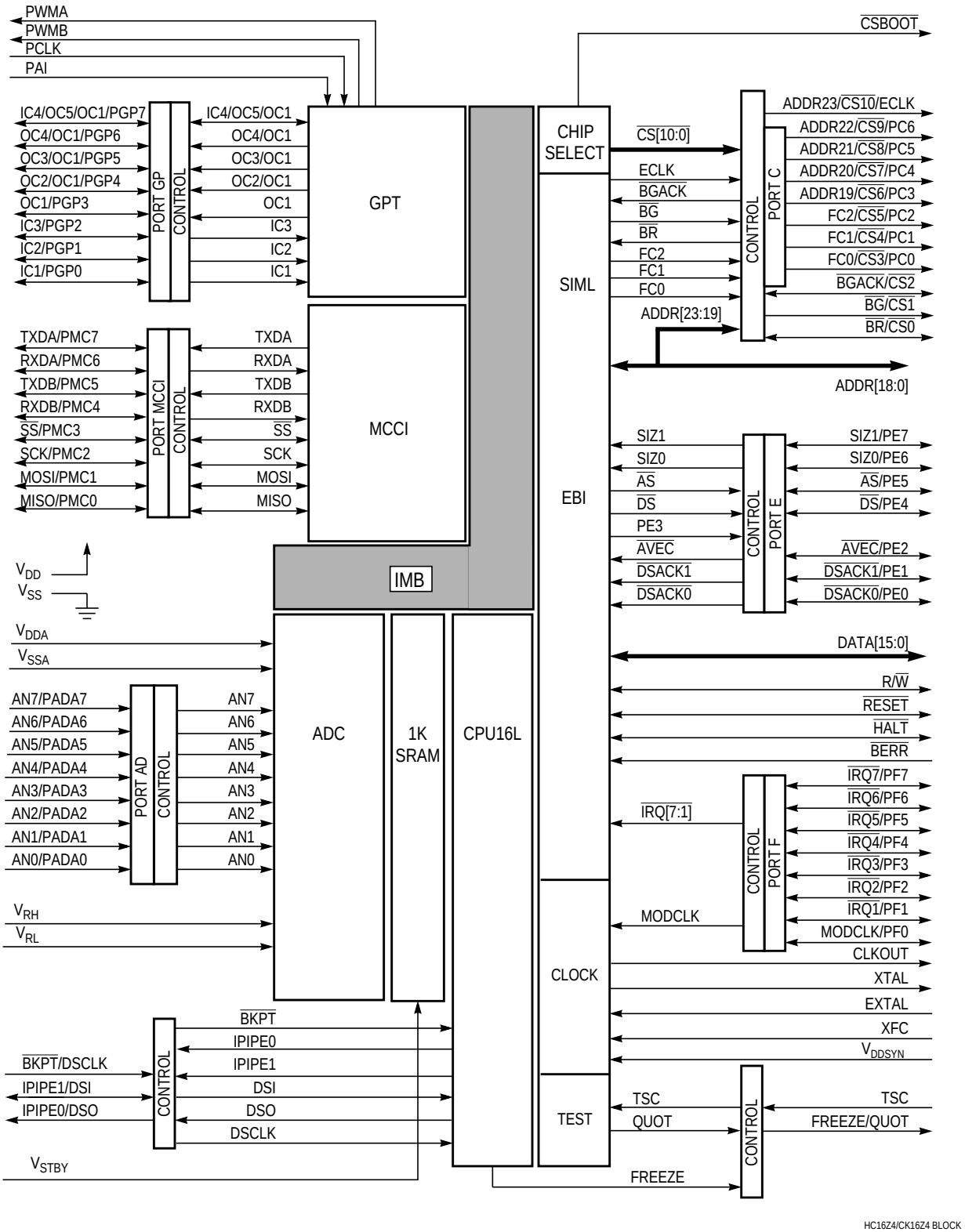
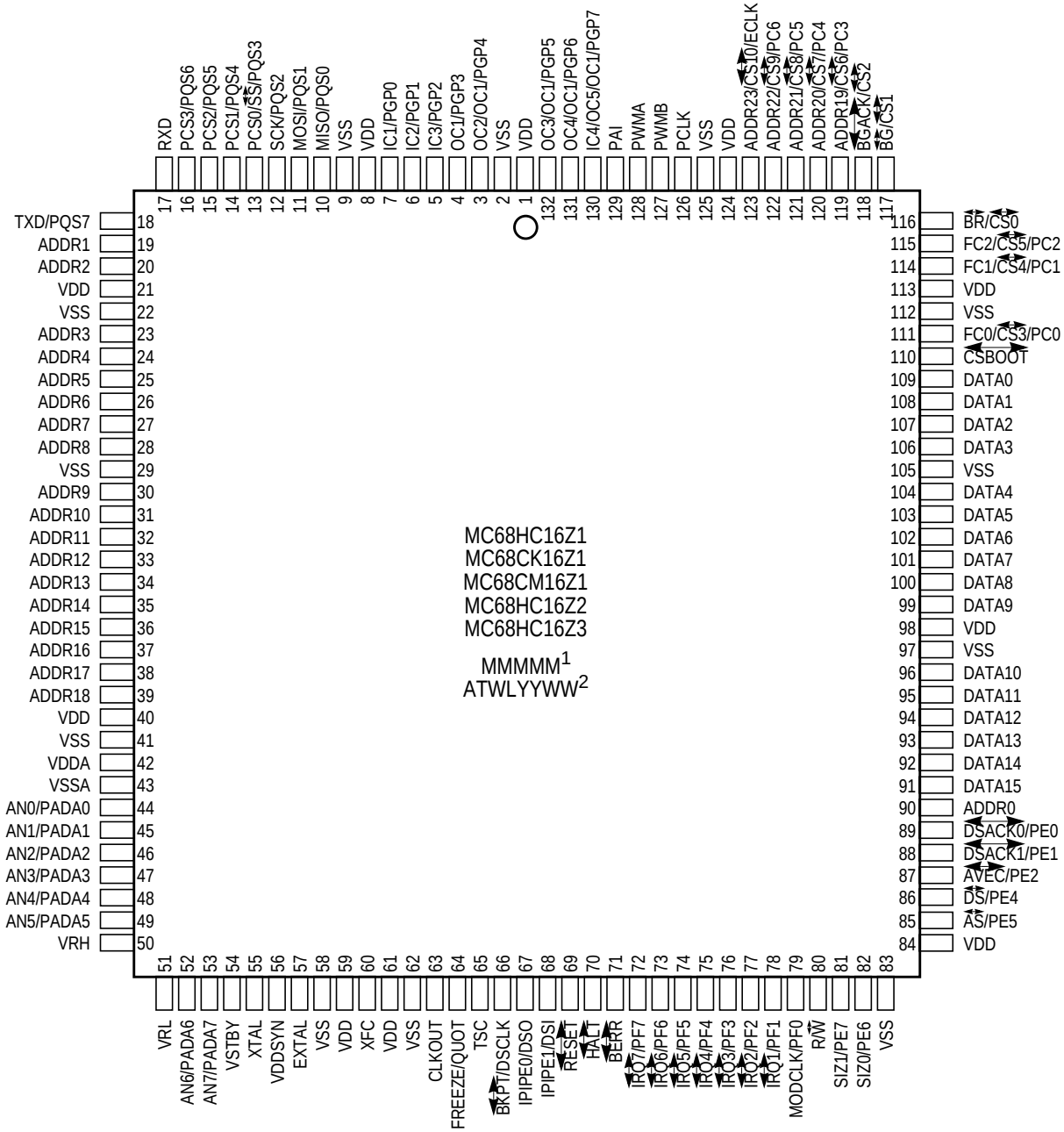


Figure 3-3 MC68HC16Z4/CK16Z4 Block Diagram

Freescale Semiconductor, Inc.

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



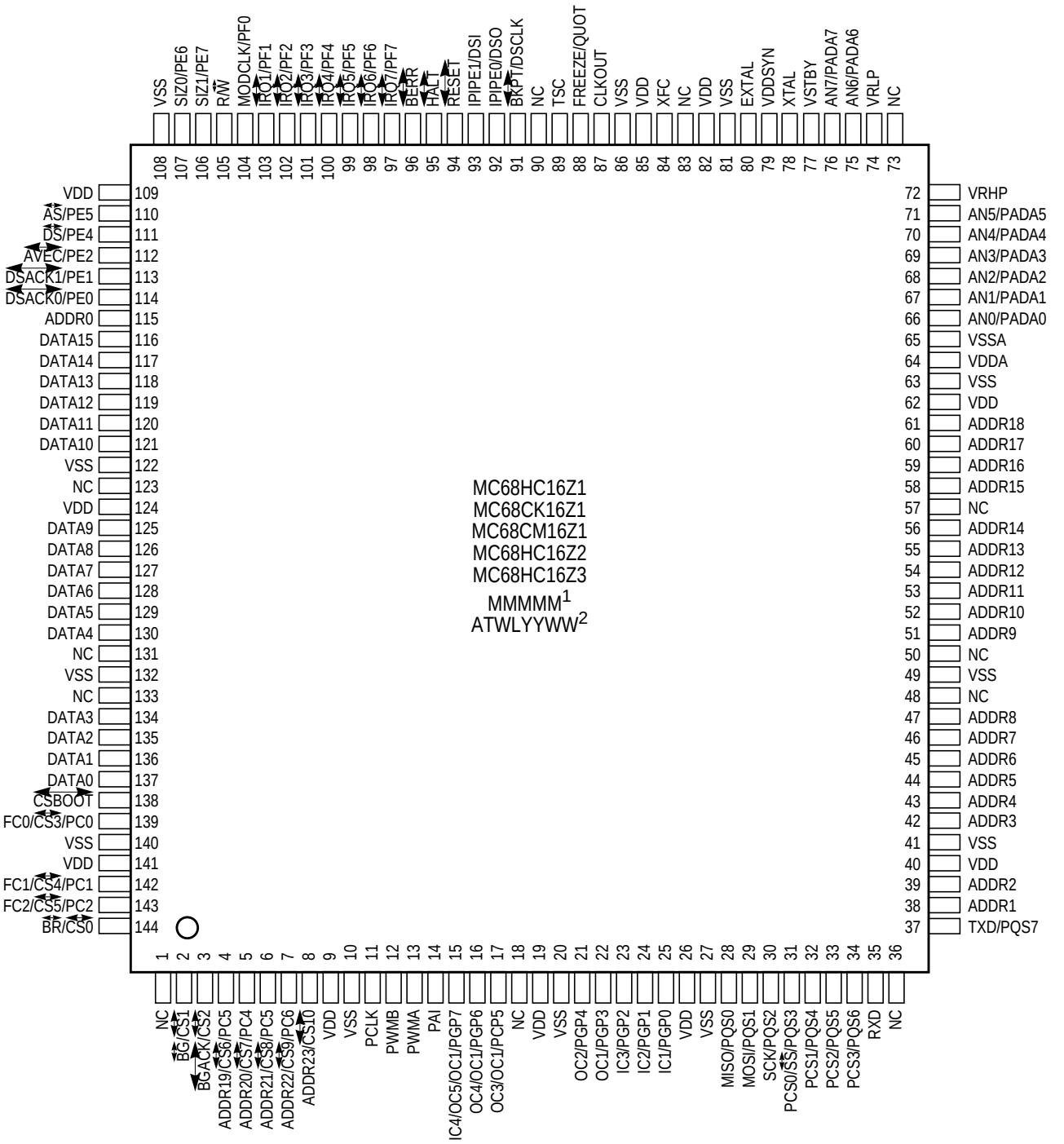
NOTES:
1. MMMMM = MASK OPTION NUMBER
2. ATWLYYWW = ASSEMBLY TEST LOCATION/YEAR, WEEK

HC16Z1/CKZ1/CMZ1/Z2/Z3 132-PIN QFP

Figure 3-4 MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 Pin Assignments for 132-Pin Package

Freescale Semiconductor, Inc.

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



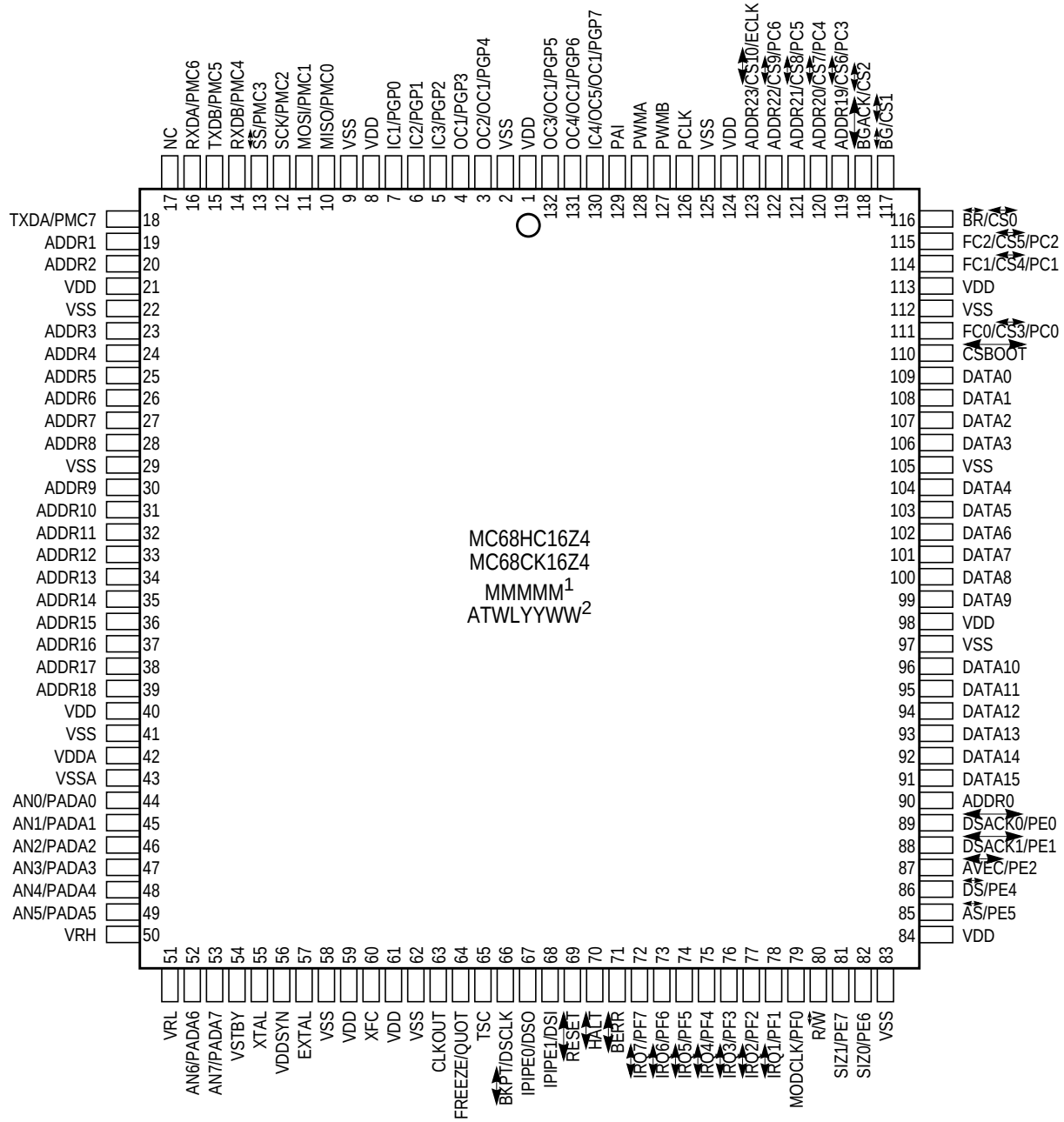
NOTES:
1. MMMMM = MASK OPTION NUMBER
2. ATWLYYWW = ASSEMBLY TEST LOCATION/YEAR, WEEK

HC16Z1/CKZ1/CMZ1/Z2/Z3 144-PIN QFP

Figure 3-5 MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 Pin Assignments for 144-Pin Package

Freescale Semiconductor, Inc.

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



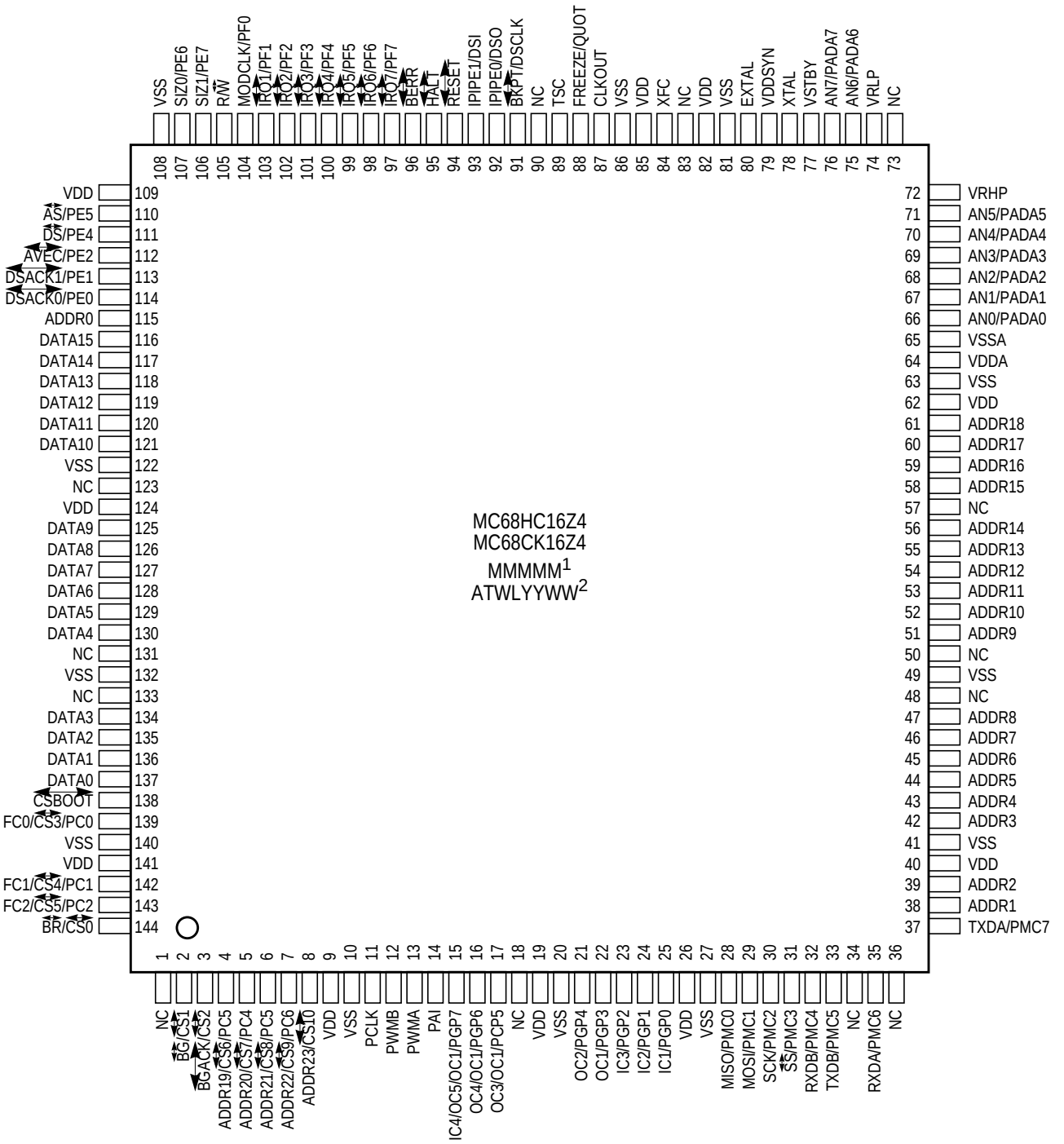
NOTES:
1. MMMMM = MASK OPTION NUMBER
2. ATWLYYWW = ASSEMBLY TEST LOCATION/YEAR, WEEK

HC16Z4/CK16Z4 132-PIN QFP

Figure 3-6 MC68HC16Z4/CKZ4 Pin Assignments for 132-Pin Package

Freescale Semiconductor, Inc.

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



NOTES:
1. MMMMM = MASK OPTION NUMBER
2. ATWLYYWW = ASSEMBLY TEST LOCATION/YEAR, WEEK

HC16Z4/CK16Z4 144-PIN QFP

Figure 3-7 MC68HC16Z4/CKZ4 Pin Assignments for 144-Pin Package

Freescale Semiconductor, Inc.

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

3.4 Pin Descriptions

The following tables are a summary of the functional characteristics of M68HC16 Z-series MCU pins. **Table 3-1** shows all inputs and outputs. Digital inputs and outputs use CMOS logic levels. An entry in the “Discrete I/O” column indicates that a pin can also be used for general-purpose input, output, or both. The I/O port designation is given when it applies. Refer to **Figure 3-1** for port organization. **Table 3-2** shows types of output drivers. **Table 3-3** shows characteristics of power pins.

Table 3-1 M68HC16 Z-Series Pin Characteristics

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
ADDR23/$\overline{CS10}$/ECLK	A	Y	N	O	—
ADDR[22:19]/$\overline{CS[9:6]}$	A	Y	N	O	PC[6:3]
ADDR[18:0]	A	Y	N	—	—
AN[7:0][1]	—	Y	N	I	PADA[7:0]
$\overline{AS}$	B	Y	Y	I/O	PE5
$\overline{AVEC}$	B	Y	N	I/O	PE2
$\overline{BERR}$	B	Y	N	—	—
$\overline{BG}$/$\overline{CS1}$	B	—	—	—	—
$\overline{BGACK}$/$\overline{CS2}$	B	Y	N	—	—
$\overline{BKPT}$/DSCLK	—	Y	Y	—	—
$\overline{BR}$/$\overline{CS0}$	B	Y	N	O	—
CLKOUT	A	—	—	—	—
$\overline{CSBOOT}$	B	—	—	—	—
DATA[15:0][1]	AW	Y	N	—	—
$\overline{DS}$	B	Y	Y	I/O	PE4
$\overline{DSACK[1:0]}$	B	Y	N	I/O	PE[1:0]
DSI/IPIPE1	A	Y	Y	—	—
DSO/IPIPE0	A	—	—	—	—
EXTAL[2]	—	—	Special	—	—
FC[2:0]/$\overline{CS[5:3]}$	A	Y	N	O	PC[2:0]
FREEZE/QUOT	A	—	—	—	—
$\overline{HALT}$	Bo	Y	N	—	—
IC4/OC5	A	Y	Y	I/O	PGP7
IC[3:1]	A	Y	Y	I/O	PGP[2:0]
$\overline{IRQ[7:1]}$	B	Y	Y	I/O	PF[7:1]
MISO	Bo	Y	Y	I/O	PQS0
MISO[3]	Bo	Y	Y	I/O	PMC0
MODCLK[1]	B	Y	N	I/O	PF0
MOSI	Bo	Y	Y	I/O	PQS1
MOSI[3]	Bo	Y	Y	I/O	PMC1
OC[4:1]	A	Y	Y	I/O	PGP[6:3]
PAI[4]	—	Y	Y	I	—

Freescale Semiconductor, Inc.

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

Table 3-1 M68HC16 Z-Series Pin Characteristics (Continued)

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
PCLK[4]	—	Y	Y	I	—
PCS0/$\overline{SS}$	Bo	Y	Y	I/O	PQS3
PCS[3:1]	Bo	Y	Y	I/O	PQS[6:4]
PWMA, PWMB[5]	A	—	—	O	—
R/$\overline{W}$	A	Y	N	—	—
$\overline{RESET}$	Bo	Y	Y	—	—
RXD	—	N	N	—	—
RXDA[3]	Bo	Y	Y	—	PMC6
RXDB[3]	Bo	Y	Y	—	PMC4
SCK[3]	Bo	Y	Y	—	PMC2
SCK	Bo	Y	Y	I/O	PQS2
SIZ[1:0]	B	Y	Y	I/O	PE[7:6]
$\overline{SS}$[3]	Bo	Y	Y	—	PMC3
TSC	—	Y	Y	—	—
TXD	Bo	Y	Y	I/O	PQS7
TXDA[3]	Bo	Y	Y	—	PMC7
TXDB[3]	Bo	Y	Y	—	PMC5
XFC[2]	—	—	—	Special	—
XTAL[2]	—	—	—	Special	—

NOTES:
1. DATA[15:0] are synchronized during reset only. MODCLK, QSM, MCCI and ADC pins are synchronized only when used as input port pins.
2. EXTAL, XFC, and XTAL are clock reference connections.
3. MCCI pins used only on the MC68HC16Z4/CK16Z4.
4. PAI and PCLK can be used for discrete input, but are not part of an I/O port.
5. PWMA and PWMB can be used for discrete output, but are not part of an I/O port.

Table 3-2 M68HC16 Z-Series Driver Types

Type	I/O	Description
A	O	Three-state capable output signals
Aw	O	Type A output with weak p-channel pull-up during reset
B	O	Three-state output that includes circuitry to pull up output before high impedance is established, to ensure rapid rise time
Bo	O	Type B output that can be operated in an open-drain mode

Freescale Semiconductor, Inc.

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

Table 3-3 M68HC16 Z-Series Power Connections

Pin Mnemonic	Description
V_{STBY}	Standby RAM power
V_{DDSYN}	Clock synthesizer power
V_{DDA}/V_{SSA}	A/D converter power
V_{RH}/V_{RL}	A/D reference voltage
V_{SS}/V_{DD}	Microcontroller power

3.5 Signal Descriptions

The following tables define the M68HC16 Z-series MCU signals. **Table 3-4** shows signal origin, type, and active state. **Table 3-5** describes signal functions. Both tables are sorted alphabetically by mnemonic. MCU pins often have multiple functions. More than one description can apply to a pin.

Table 3-4 M68HC16 Z-Series Signal Characteristics

Signal Name	MCU Module	Signal Type	Active State
ADDR[23:0]	SIM	Bus	—
AN[7:0]	ADC	Input	—
$\overline{AS}$	SIM	Output	0
$\overline{AVEC}$	SIM	Input	0
$\overline{BERR}$	SIM	Input	0
$\overline{BG}$	SIM	Output	0
$\overline{BGACK}$	SIM	Input	0
$\overline{BKPT}$	CPU16	Input	0
$\overline{BR}$	SIM	Input	0
CLKOUT	SIM	Output	—
$\overline{CS[10:0]}$	SIM	Output	0
$\overline{CSBOOT}$	SIM	Output	0
DATA[15:0]	SIM	Bus	—
$\overline{DS}$	SIM	Output	0
$\overline{DSACK[1:0]}$	SIM	Input	0
DSCLK	CPU16	Input	Serial Clock
DSI	CPU16	Input	Serial Data
DSO	CPU16	Output	Serial Data
EXTAL	SIM	Input	—
FC[2:0]	SIM	Output	—
FREEZE	SIM	Output	1
$\overline{HALT}$	SIM	Input/Output	0
IC[4:1]	GPT	Input	—
IPIPE0	CPU16	Output	—
IPIPE1	CPU16	Output	—
$\overline{IRQ[7:1]}$	SIM	Input	0

Freescale Semiconductor, Inc.

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

Table 3-4 M68HC16 Z-Series Signal Characteristics (Continued)

Signal Name	MCU Module	Signal Type	Active State
MISO	QSM	Input/Output	—
MISO[1]	MCCI	Input/Output	—
MODCLK	SIM	Input	—
MOSI	QSM	Input/Output	—
MOSI[1]	MCCI	Input/Output	—
OC[5:1]	GPT	Output	—
PADA[7:0]	ADC	Input	—
PAI	GPT	Input	—
PC[6:0]	SIM	Output	—
PE[7:0]	SIM	Input/Output	—
PF[7:0]	SIM	Input/Output	—
PGP[7:0]	GPT	Input/Output	—
PQS[7:0]	QSM	Input/Output	—
PCLK	GPT	Input	—
PCS[3:0]	QSM	Input/Output	—
PWMA, PWMB	GPT	Output	—
PMC[7:0][1]	MCCI	Input/Output	—
QUOT	SIM	Output	—
R/$\overline{W}$	SIM	Output	1/0
$\overline{RESET}$	SIM	Input/Output	0
RXD	QSM	Input	—
RXDA[1]	MCCI	Input	—
RXDB[1]	MCCI	Input	—
SCK	QSM	Input/Output	—
SCK[1]	MCCI	Input/Output	—
SIZ[1:0]	SIM	Output	1/0
$\overline{SS}$	QSM	Input	0
$\overline{SS}$[1]	MCCI	Input	0
TSC	SIM	Input	1
TXD	QSM	Output	—
TXDA[1]	MCCI	Output	—
TXDB[1]	MCCI	Output	—
XFC	SIM	Input	—
XTAL	SIM	Output	—

NOTES:
1. Used only in the MC68HC16Z4/CK16Z4.

Freescale Semiconductor, Inc.

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

Table 3-5 M68HC16 Z-Series Signal Function

Mnemonic	Signal Name	Function
ADDR[19:0]	Address Bus	20-bit address bus used by CPU16
AN[7:0]	ADC Analog Input	Inputs to ADC multiplexer
$\overline{AS}$	Address Strobe	Indicates that a valid address is on the address bus
$\overline{AVEC}$	Autovector	Requests an automatic vector during interrupt acknowledge
$\overline{BERR}$	Bus Error	Indicates that a bus error has occurred
$\overline{BG}$	Bus Grant	Indicates that the MCU has relinquished the bus
$\overline{BGACK}$	Bus Grant Acknowledge	Indicates that an external device has assumed bus mastership
$\overline{BKPT}$	Breakpoint	Signals a hardware breakpoint to the CPU
$\overline{BR}$	Bus Request	Indicates that an external device requires bus mastership
CLKOUT	System Clockout	System clock output
$\overline{CS[10:0]}$	Chip-Selects	Select external devices at programmed addresses
$\overline{CSBOOT}$	Boot Chip Select	Chip select for external boot start-up ROM
DATA[15:0]	Data Bus	16-bit data bus
$\overline{DS}$	Data Strobe	During a read cycle, indicates that an external device should place valid data on the data bus. During a write cycle, indicates that valid data is on the data bus.
$\overline{DSACK[1:0]}$	Data and Size Acknowledge	Provide asynchronous data transfers and dynamic bus sizing
DSI, DSO, DSCLK	Development Serial In, Out, Clock	Serial I/O and clock for background debug mode
EXTAL, XTAL	Crystal Oscillator	Connections for clock synthesizer circuit reference a crystal or an external oscillator can be used
FC[2:0]	Function Codes	Identify processor state and current address space
FREEZE	Freeze	Indicates that the CPU has entered background mode
$\overline{HALT}$	Halt	Suspend external bus activity
$\overline{IRQ[7:1]}$	Interrupt Request Level	Provides an interrupt priority level to the CPU
IPIPE[1:0]	Instruction Pipeline	Indicate instruction pipeline activity
MISO	Master In Slave Out	Serial input to QSPI in master mode; serial output from QSPI in slave mode
MISO[1]	Master In Slave Out	Serial input to SPI in master mode; serial output from SPI in slave mode
MODCLK	Clock Mode Select	Selects the source and type of system clock
MOSI	Master Out Slave In	Serial output from QSPI in master mode; serial input to QSPI in slave mode
MOSI[1]	Master Out Slave In	Serial output from SPI in master mode; serial input to SPI in slave mode
PADA[7:0]	Port ADA	ADC digital input port signals
PAI	Pulse Accumulator Input	Input to the GPT pulse accumulator
PCLK	Auxiliary Timer Clock	GPT external clock input
PC[6:0]	Port C	Port C digital output port signals
PCS[3:0]	Peripheral Chip Select	QSPI peripheral chip-selects
PE[7:0]	Port E	Port E digital I/O port signals

Freescale Semiconductor, Inc.

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

Table 3-5 M68HC16 Z-Series Signal Function (Continued)

Mnemonic	Signal Name	Function
PF[7:0]	Port F	Port F digital I/O port signals
PGP[7:0]	Port GP	GPT digital I/O port signals
PQS[7:0]	Port QS	QSM digital I/O port signals
PWMA, PWMB	Pulse Width Modulation	Output for PWM
QUOT	Quotient Out	Provides the quotient bit of the polynomial divider
R/$\overline{W}$	Read/Write	Indicates the direction of data transfer on the bus
$\overline{RESET}$	Reset	System reset
RXD	Receive Data (SCI)	Serial input to the SCI
RXDA[1]	SCI A Receive Data	Serial input from SCI A
RXDB[1]	SCI B Receive Data	Serial input from SCI B
SCK	Serial Clock (QSPI)	Clock output from QSPI in master mode; clock input to QSPI in slave mode
SCK[1]	Serial Clock (SPI)	Clock output from SPI in master mode; clock input to SPI in slave mode
SIZ[1:0]	Size	Indicates the number of bytes to be transferred during a bus cycle
$\overline{SS}$	Slave Select (QSPI)	Causes serial transmission when QSPI is in slave mode; causes mode fault in master mode
$\overline{SS}$[1]	Slave Select (SPI)	Causes serial transmission when the SPI is in slave mode; causes mode fault in master mode
TSC	Three-State Control	Places all output drivers in a high-impedance state
TXD	SCI Transmit Data	Serial output from the SCI
TXDA[1]	SCI A Transmit Data	Serial output from SCI A
TXDB[1]	SCI B Transmit Data	Serial output from SCI B
XFC	External Filter Capacitor	Connection for external phase-locked loop filter capacitor

NOTES:
1. MCCI signals present only in MC68HC16Z4/CK16Z4.

3.6 Internal Register Map

In **Figures 3-8**, **3-9**, and **3-10**, IMB ADDR[23:20] are represented by the letter Y. The value represented by Y determines the base address of MCU module control registers. Y is equal to M111, where M is the logic state of the module mapping (MM) bit in the system integration module configuration register (SIMCR). Since the CPU16 uses only ADDR[19:0], and ADDR[23:20] follow the logic state of ADDR19 when CPU driven, the CPU cannot access IMB addresses from $080000 to $F7FFFF. In order for the MCU to function correctly, MM must be set (Y must equal $F). If M is cleared, internal registers are mapped to base address $700000, and are inaccessible until a reset occurs. The SRAM array is positioned by a base address register in the SRAM CTRL block. Unimplemented blocks are mapped externally.

Freescale Semiconductor, Inc.

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

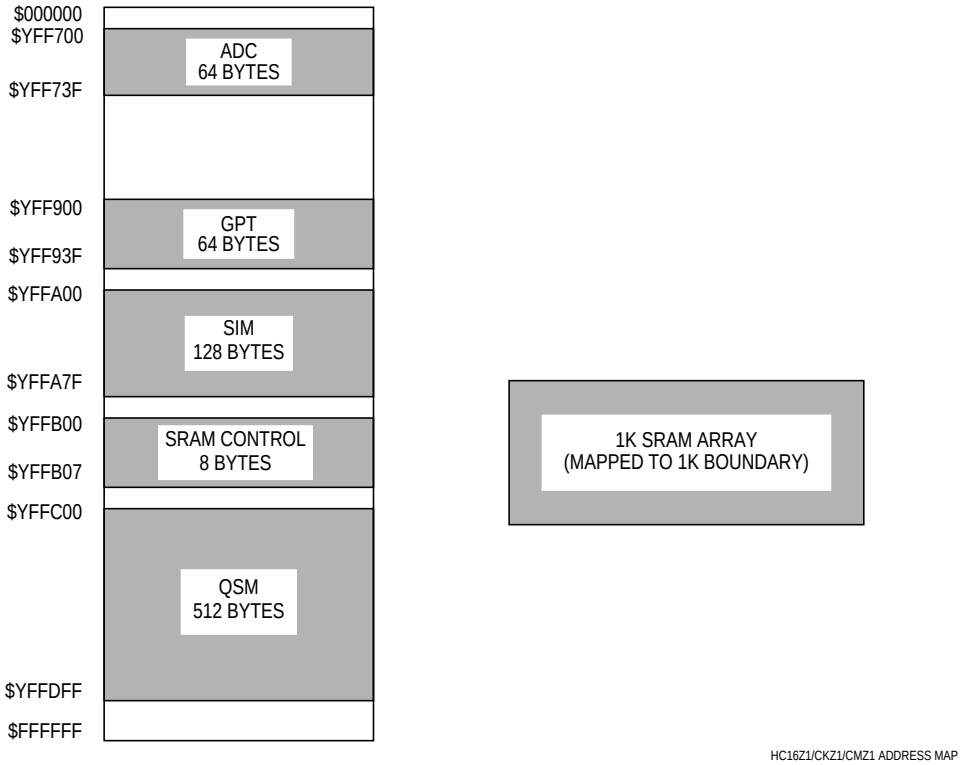


Figure 3-8 MC68HC16Z1/CKZ1/CMZ1 Address Map

Freescale Semiconductor, Inc.

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

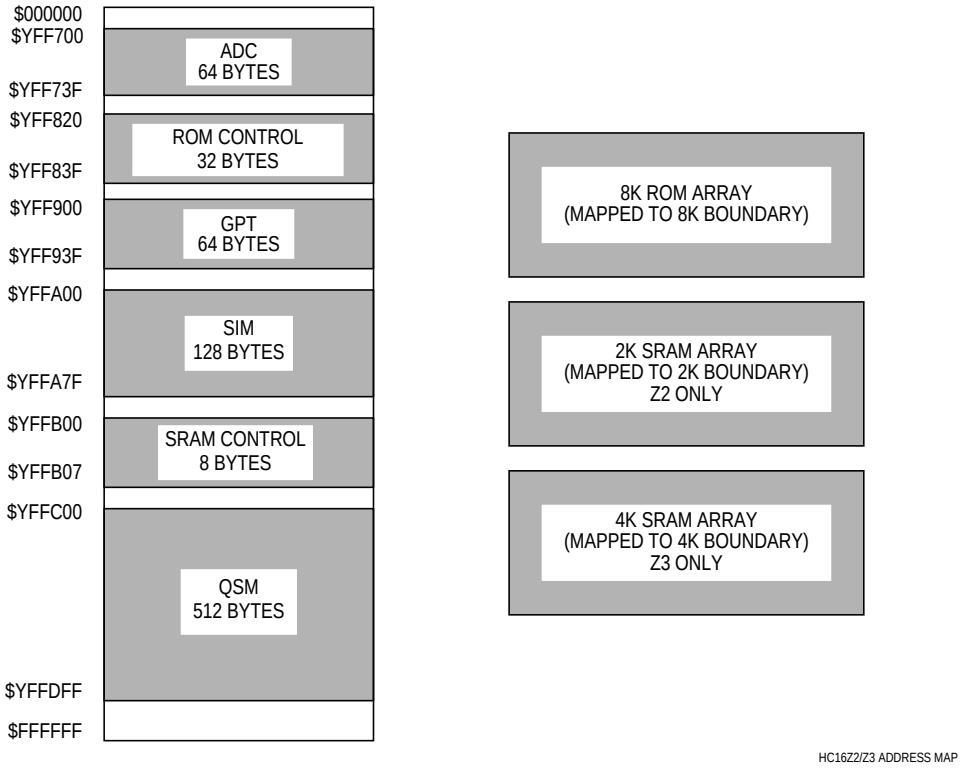


Figure 3-9 MC68HC16Z2/Z3 Address Map

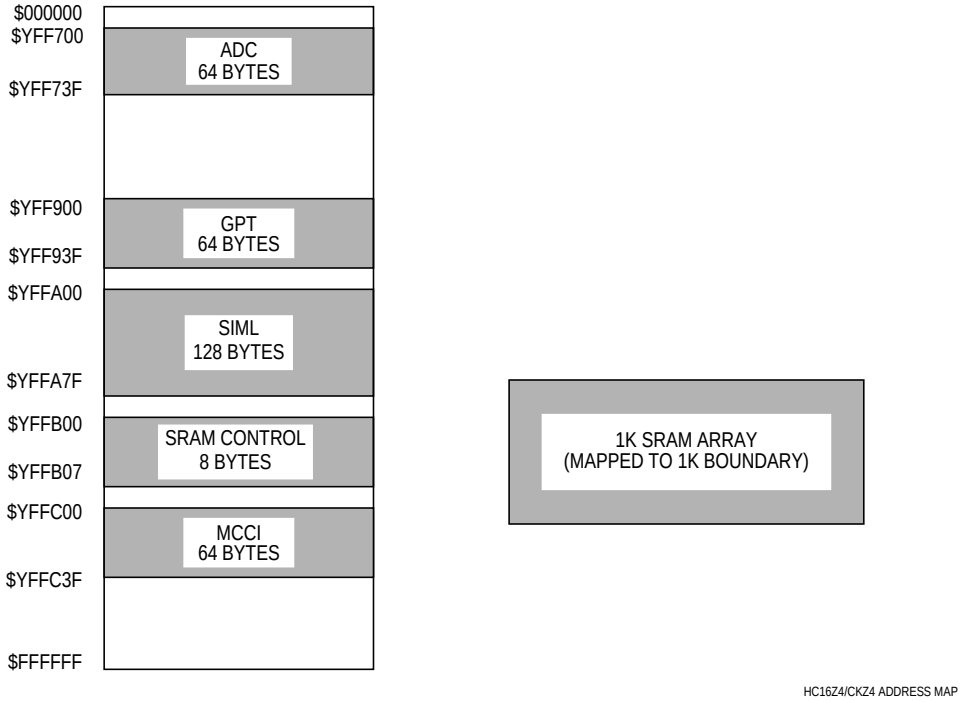


Figure 3-10 MC68HC16Z4/CKZ4 Address Map

Freescale Semiconductor, Inc.

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

3.7 Address Space Maps

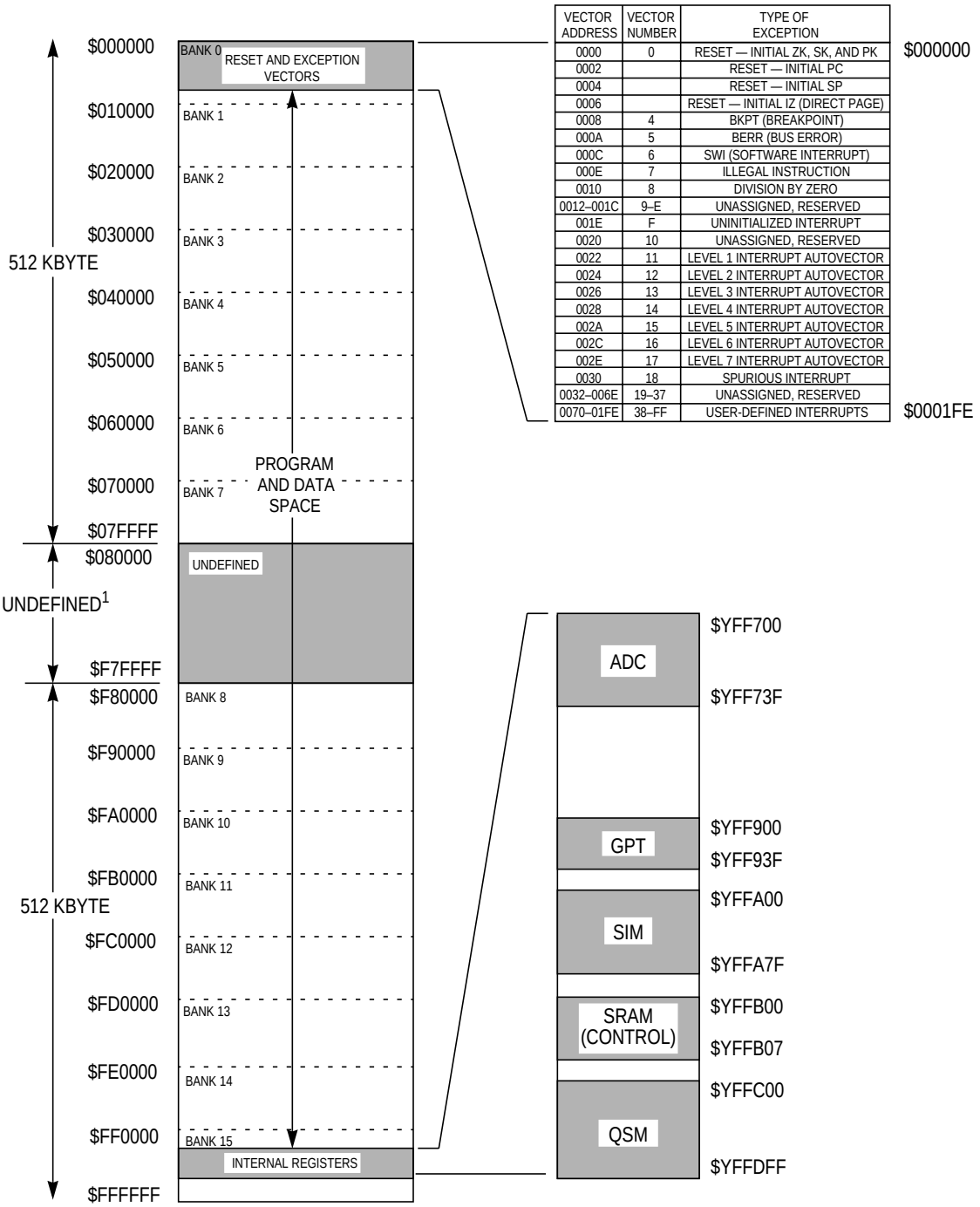
Figures 3-11 through 3-16 show CPU16 address space for M68HC16 Z-series MCUs. Address space can be split into physically distinct program and data spaces by decoding the MCU function code outputs.

Figures 3-11, 3-12, and 3-13 show the memory map of a system that has combined program and data spaces. Figures 3-14, 3-15, and 3-16 show the memory map when MCU function code outputs are decoded.

Reset and exception vectors are mapped into bank 0 and cannot be relocated. The CPU16 program counter, stack pointer, and Z index register can be initialized to any address in pseudolinear memory, but exception vectors are limited to 16-bit addresses. To access locations outside of bank 0 during exception handler routines (including interrupt exceptions), a jump table must be used. Refer to SECTION 4 CENTRAL PROCESSOR UNIT for more information concerning memory management, extended addressing, and exception processing. Refer to SECTION 5 SYSTEM INTEGRATION MODULE for more information concerning function codes, address space types, resets, and interrupts.

Freescale Semiconductor, Inc.

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



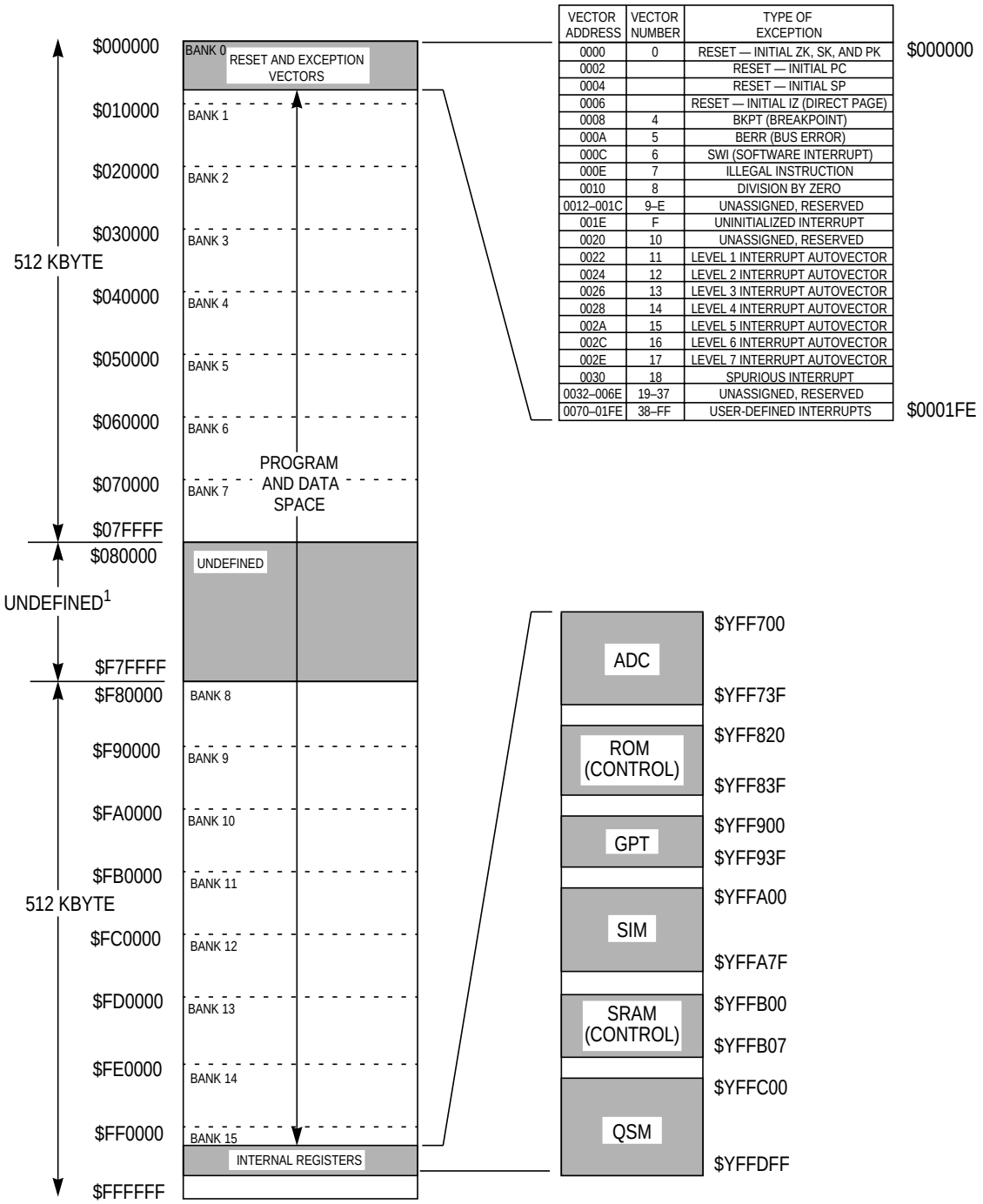
NOTE:
1. THE ADDRESSES DISPLAYED IN THIS MEMORY MAP ARE THE FULL 24-BIT IMB ADDRESSES. THE CPU16 ADDRESS BUS IS 20 BITS WIDE, AND CPU16 ADDRESS LINE 19 DRIVES IMB ADDRESS LINES [23:20]. THE BLOCK OF ADDRESSES FROM $080000 TO $F7FFFF MARKED AS UNDEFINED WILL NEVER APPEAR ON THE IMB. MEMORY BANKS 0 TO 15 APPEAR FULLY CONTIGUOUS IN THE CPU16'S FLAT 20-BIT ADDRESS SPACE. THE CPU16 NEED ONLY GENERATE A 20-BIT EFFECTIVE ADDRESS TO ACCESS ANY LOCATION IN THIS RANGE.

HC16Z1/CK/CM MEM MAP (C)

Figure 3-11 MC68HC16Z1/CKZ1/CMZ1 Combined Program and Data Space Map

Freescale Semiconductor, Inc.

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



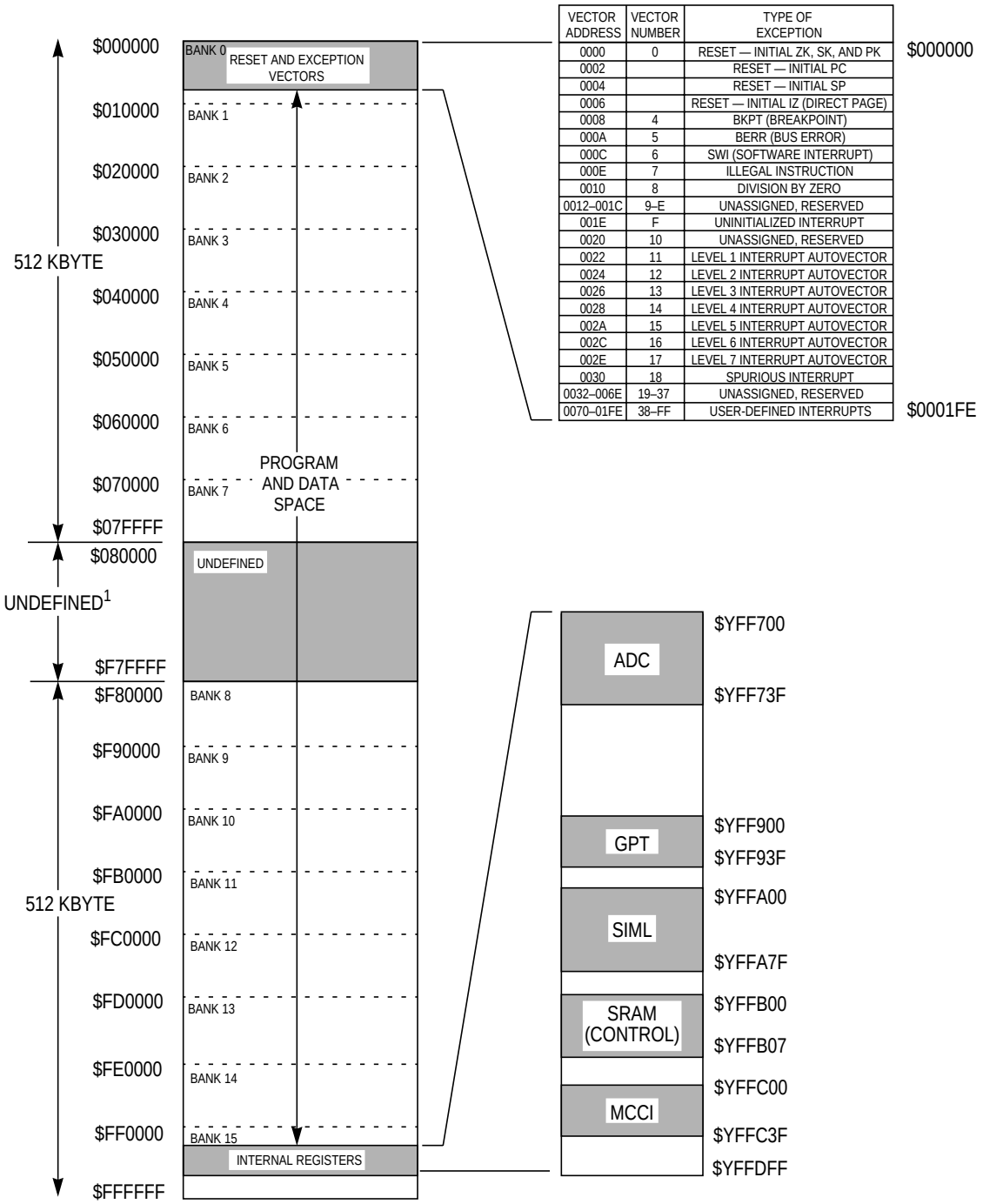
NOTE:
1. THE ADDRESSES DISPLAYED IN THIS MEMORY MAP ARE THE FULL 24-BIT IMB ADDRESSES. THE CPU16 ADDRESS BUS IS 20 BITS WIDE, AND CPU16 ADDRESS LINE 19 DRIVES IMB ADDRESS LINES [23:20]. THE BLOCK OF ADDRESSES FROM $080000 TO $F7FFFF MARKED AS UNDEFINED WILL NEVER APPEAR ON THE IMB. MEMORY BANKS 0 TO 15 APPEAR FULLY CONTIGUOUS IN THE CPU16'S FLAT 20-BIT ADDRESS SPACE. THE CPU16 NEED ONLY GENERATE A 20-BIT EFFECTIVE ADDRESS TO ACCESS ANY LOCATION IN THIS RANGE.

HC16 Z2/Z3 MEM MAP (C)

Figure 3-12 MC68HC16Z2/Z3 Combined Program and Data Space Map

Freescale Semiconductor, Inc.

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



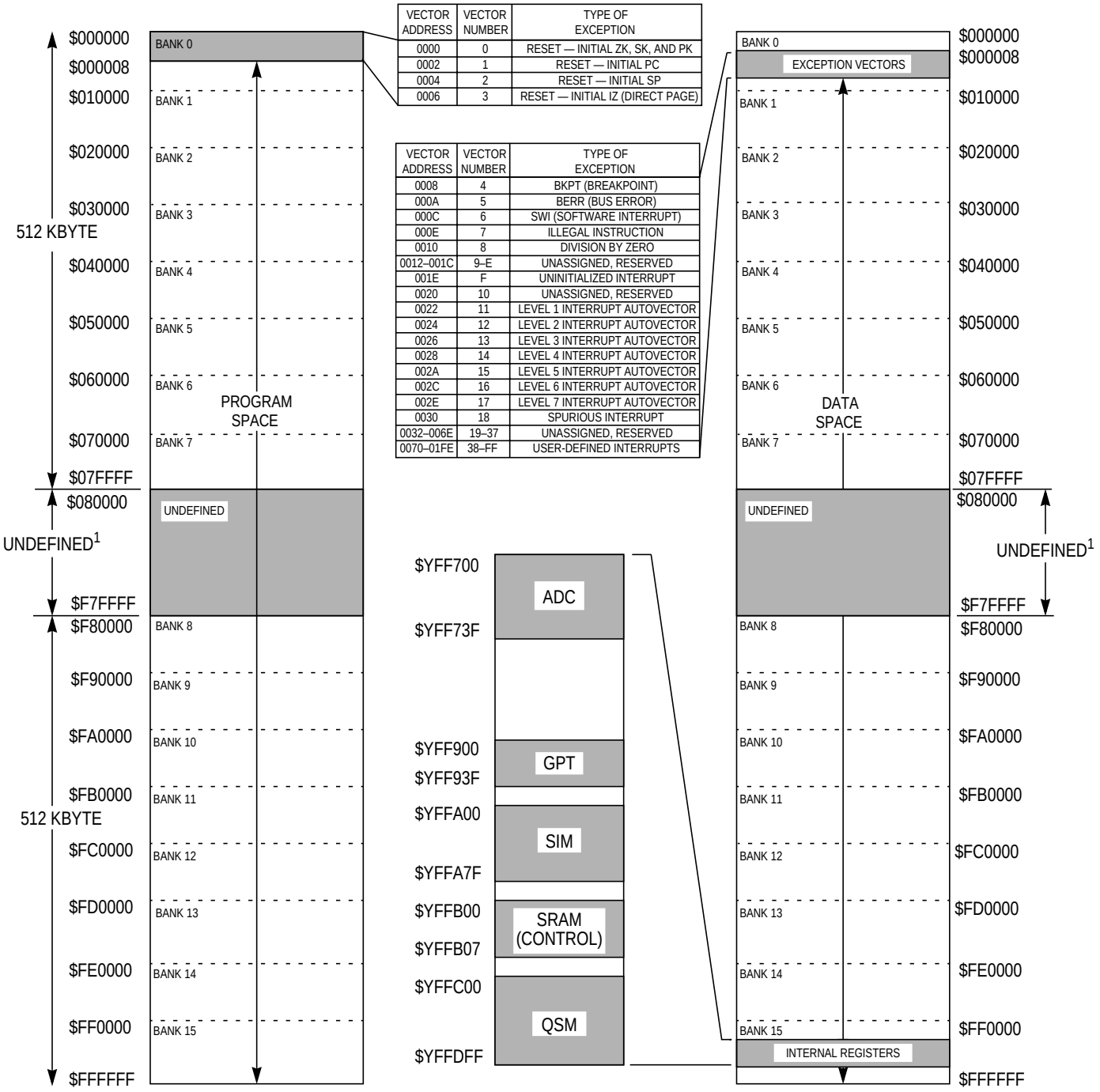
NOTE:
1. THE ADDRESSES DISPLAYED IN THIS MEMORY MAP ARE THE FULL 24-BIT IMB ADDRESSES. THE CPU16 ADDRESS BUS IS 20 BITS WIDE, AND CPU16 ADDRESS LINE 19 DRIVES IMB ADDRESS LINES [23:20]. THE BLOCK OF ADDRESSES FROM $080000 TO $F7FFFF MARKED AS UNDEFINED WILL NEVER APPEAR ON THE IMB. MEMORY BANKS 0 TO 15 APPEAR FULLY CONTIGUOUS IN THE CPU16'S FLAT 20-BIT ADDRESS SPACE. THE CPU16 NEED ONLY GENERATE A 20-BIT EFFECTIVE ADDRESS TO ACCESS ANY LOCATION IN THIS RANGE.

HC16Z4/CKZ4 MEM MAP (C)

Figure 3-13 MC68HC16Z4/CKZ4 Combined Program and Data Space Map

Freescale Semiconductor, Inc.

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



VECTOR ADDRESS	VECTOR NUMBER	TYPE OF EXCEPTION
0000	0	RESET — INITIAL ZK, SK, AND PK
0002	1	RESET — INITIAL PC
0004	2	RESET — INITIAL SP
0006	3	RESET — INITIAL IZ (DIRECT PAGE)

VECTOR ADDRESS	VECTOR NUMBER	TYPE OF EXCEPTION
0008	4	BKPT (BREAKPOINT)
000A	5	BERR (BUS ERROR)
000C	6	SWI (SOFTWARE INTERRUPT)
000E	7	ILLEGAL INSTRUCTION
0010	8	DIVISION BY ZERO
0012–001C	9–E	UNASSIGNED, RESERVED
001E	F	UNINITIALIZED INTERRUPT
0020	10	UNASSIGNED, RESERVED
0022	11	LEVEL 1 INTERRUPT AUTOVECTOR
0024	12	LEVEL 2 INTERRUPT AUTOVECTOR
0026	13	LEVEL 3 INTERRUPT AUTOVECTOR
0028	14	LEVEL 4 INTERRUPT AUTOVECTOR
002A	15	LEVEL 5 INTERRUPT AUTOVECTOR
002C	16	LEVEL 6 INTERRUPT AUTOVECTOR
002E	17	LEVEL 7 INTERRUPT AUTOVECTOR
0030	18	SPURIOUS INTERRUPT
0032–006E	19–37	UNASSIGNED, RESERVED
0070–01FE	38–FF	USER-DEFINED INTERRUPTS

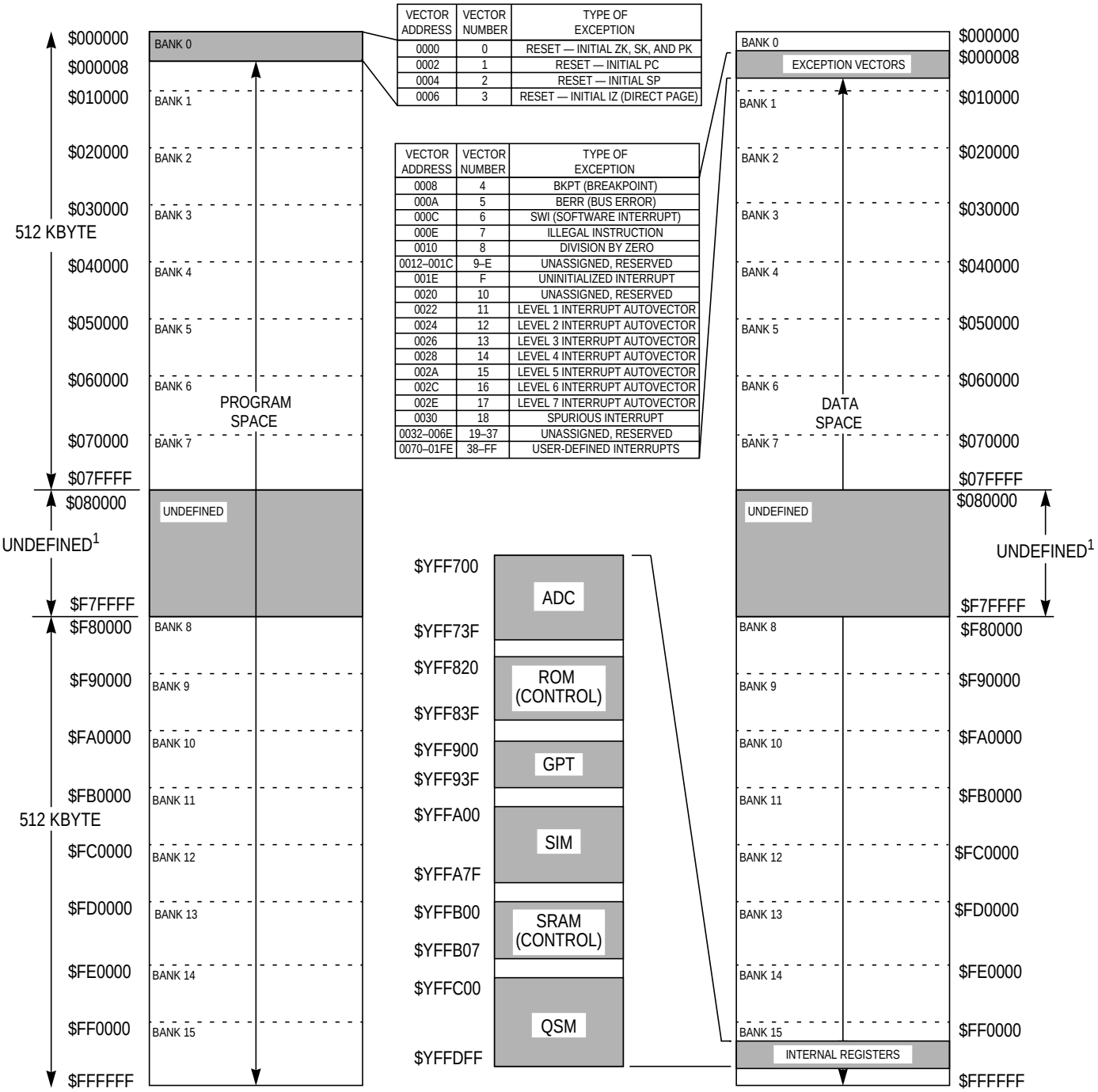
NOTE:
1. THE ADDRESSES DISPLAYED IN THIS MEMORY MAP ARE THE FULL 24-BIT IMB ADDRESSES. THE CPU16 ADDRESS BUS IS 20 BITS WIDE, AND CPU16 ADDRESS LINE 19 DRIVES IMB ADDRESS LINES [23:20]. THE BLOCK OF ADDRESSES FROM $080000 TO $F7FFFF MARKED AS UNDEFINED WILL NEVER APPEAR ON THE IMB. MEMORY BANKS 0 TO 15 APPEAR FULLY CONTIGUOUS IN THE CPU16'S FLAT 20-BIT ADDRESS SPACE. THE CPU16 NEED ONLY GENERATE A 20-BIT EFFECTIVE ADDRESS TO ACCESS ANY LOCATION IN THIS RANGE.

HC16Z1/CK/CM MEM MAP (S)

Figure 3-14 MC68HC16Z1/CKZ1/CMZ1 Separate Program and Data Space Map

Freescale Semiconductor, Inc.

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



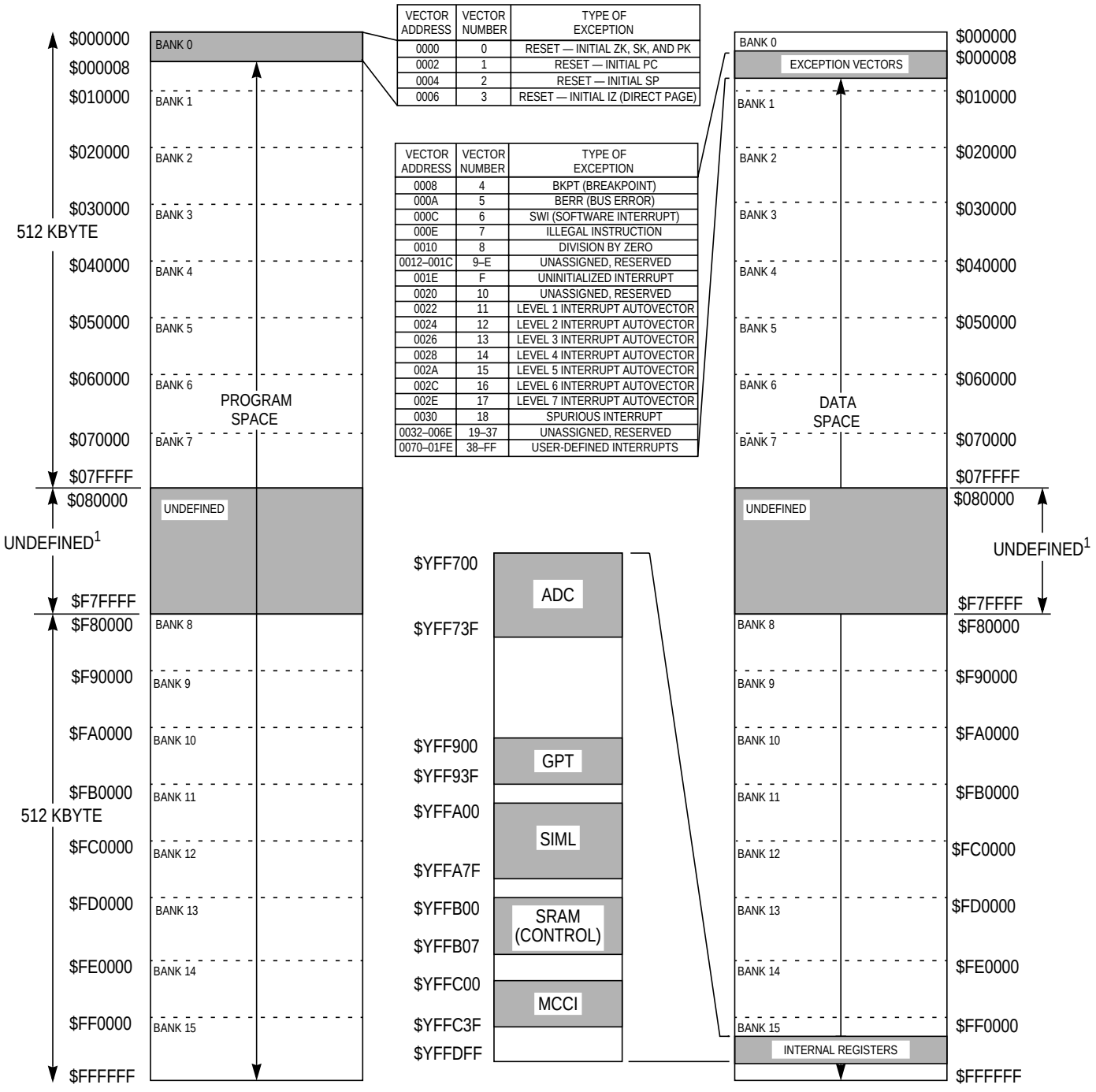
NOTE:

1. THE ADDRESSES DISPLAYED IN THIS MEMORY MAP ARE THE FULL 24-BIT IMB ADDRESSES. THE CPU16 ADDRESS BUS IS 20 BITS WIDE, AND CPU16 ADDRESS LINE 19 DRIVES IMB ADDRESS LINES [23:20]. THE BLOCK OF ADDRESSES FROM $080000 TO $F7FFFF MARKED AS UNDEFINED WILL NEVER APPEAR ON THE IMB. MEMORY BANKS 0 TO 15 APPEAR FULLY CONTIGUOUS IN THE CPU16'S FLAT 20-BIT ADDRESS SPACE. THE CPU16 NEED ONLY GENERATE A 20-BIT EFFECTIVE ADDRESS TO ACCESS ANY LOCATION IN THIS RANGE.

Figure 3-15 MC68HC16Z2/Z3 Separate Program and Data Space Map

Freescale Semiconductor, Inc.

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



NOTE:
1. THE ADDRESSES DISPLAYED IN THIS MEMORY MAP ARE THE FULL 24-BIT IMB ADDRESSES. THE CPU16 ADDRESS BUS IS 20 BITS WIDE, AND CPU16 ADDRESS LINE 19 DRIVES IMB ADDRESS LINES [23:20]. THE BLOCK OF ADDRESSES FROM $080000 TO $F7FFFF MARKED AS UNDEFINED WILL NEVER APPEAR ON THE IMB. MEMORY BANKS 0 TO 15 APPEAR FULLY CONTIGUOUS IN THE CPU16'S FLAT 20-BIT ADDRESS SPACE. THE CPU16 NEED ONLY GENERATE A 20-BIT EFFECTIVE ADDRESS TO ACCESS ANY LOCATION IN THIS RANGE.

Figure 3-16 MC68HC16Z4/CKZ4 Separate Program and Data Space Map

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 4
CENTRAL PROCESSOR UNIT

This section is an overview of the central processor unit (CPU16). For detailed information, refer to the *CPU16 Reference Manual* (CPU16RM/AD).

4.1 General

The CPU16 provides compatibility with the M68HC11 CPU and also provides additional capabilities associated with 16- and 32-bit data sizes, 20-bit addressing, and digital signal processing. CPU16 registers are an integral part of the CPU and are not addressed as memory locations.

The CPU16 treats all peripheral, I/O, and memory locations as parts of a linear one Megabyte address space. There are no special instructions for I/O that are separate from instructions for addressing memory. Address space is made up of sixteen 64-Kbyte banks. Specialized bank addressing techniques and support registers provide transparent access across bank boundaries.

The CPU16 interacts with external devices and with other modules within the microcontroller via a standardized bus and bus interface. There are bus protocols used for memory and peripheral accesses, as well as for managing a hierarchy of interrupt priorities.

4.2 Register Model

Figure 4-1 shows the CPU16 register model. Refer to the paragraphs that follow for a detailed description of each register.

Freescale Semiconductor, Inc.

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

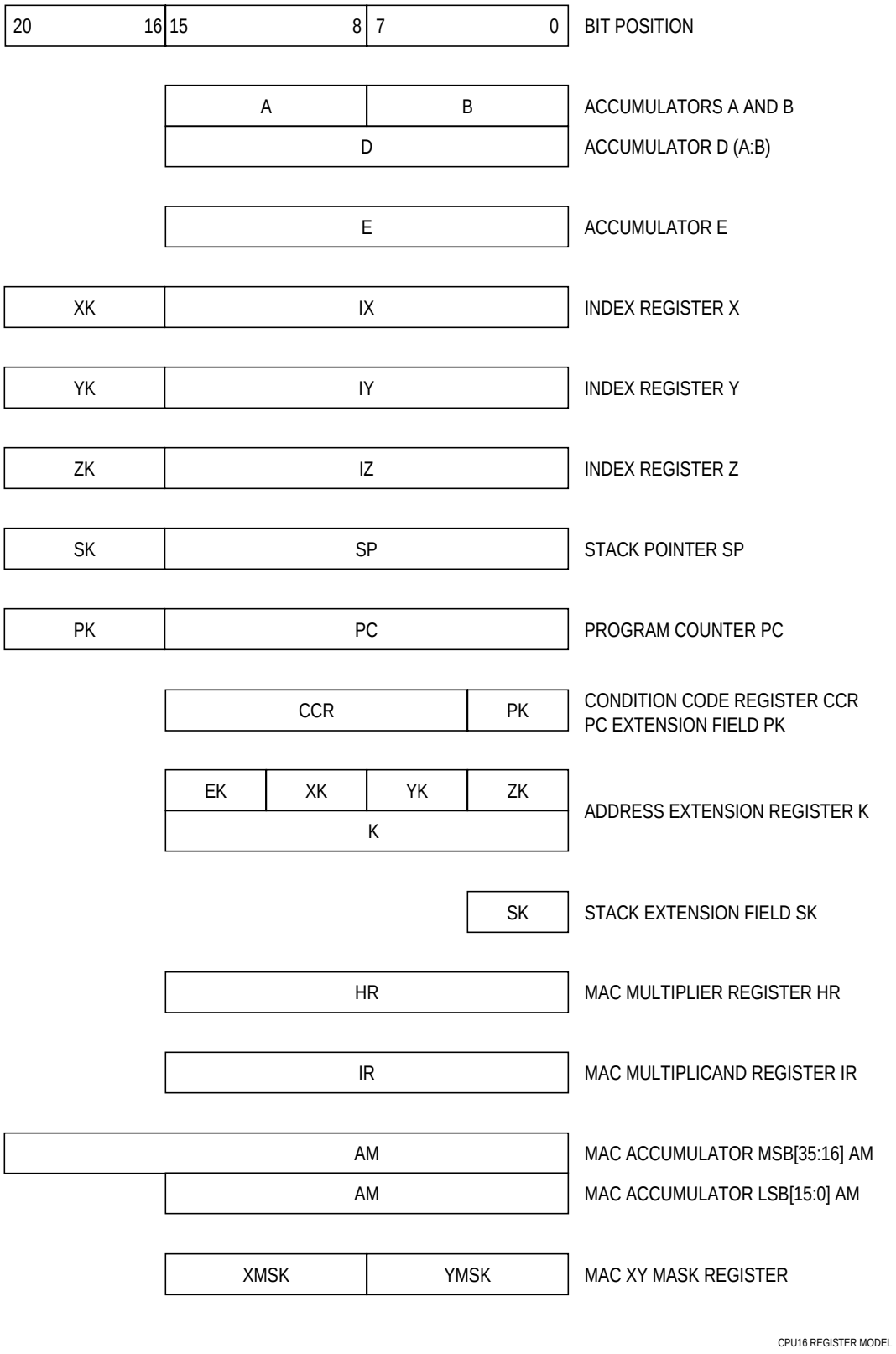


Figure 4-1 CPU16 Register Model

Freescale Semiconductor, Inc.

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

4.2.1 Accumulators

The CPU16 has two 8-bit accumulators (A and B) and one 16-bit accumulator (E). In addition, accumulators A and B can be concatenated into a second 16-bit double accumulator (D).

Accumulators A, B, and D are general-purpose registers that hold operands and results during mathematical and data manipulation operations.

Accumulator E, which can be used in the same way as accumulator D, also extends CPU16 capabilities. It allows more data to be held within the CPU16 during operations, simplifies 32-bit arithmetic and digital signal processing, and provides a practical 16-bit accumulator offset indexed addressing mode.

4.2.2 Index Registers

The CPU16 has three 16-bit index registers (IX, IY, and IZ). Each index register has an associated 4-bit extension field (XK, YK, and ZK).

Concatenated registers and extension fields provide 20-bit indexed addressing and support data structure functions anywhere in the CPU16 address space.

IX and IY can perform the same operations as M68HC11 registers of the same names, but the CPU16 instruction set provides additional indexed operations.

IZ can perform the same operations as IX and IY. IZ also provides an additional indexed addressing capability that replaces M68HC11 direct addressing mode. Initial IZ and ZK extension field values are included in the RESET exception vector, so that ZK:IZ can be used as a direct page pointer out of reset.

4.2.3 Stack Pointer

The CPU16 stack pointer (SP) is 16 bits wide. An associated 4-bit extension field (SK) provides 20-bit stack addressing.

Stack implementation in the CPU16 is from high to low memory. The stack grows downward as it is filled. SK:SP are decremented each time data is pushed on the stack, and incremented each time data is pulled from the stack.

SK:SP point to the next available stack address rather than to the address of the latest stack entry. Although the stack pointer is normally incremented or decremented by word address, it is possible to push and pull byte-sized data. Setting the stack pointer to an odd value causes data misalignment, which reduces performance.

4.2.4 Program Counter

The CPU16 program counter (PC) is 16 bits wide. An associated 4-bit extension field (PK) provides 20-bit program addressing.

CPU16 instructions are fetched from even word boundaries. Address line 0 always has a value of zero during instruction fetches to ensure that instructions are fetched from word-aligned addresses.

Freescale Semiconductor, Inc.

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

4.2.5 Condition Code Register

The 16-bit condition code register is composed of two functional blocks. The eight MSB, which correspond to the CCR on the M68HC11, contain the low-power stop control bit and processor status flags. The eight LSB contain the interrupt priority field, the DSP saturation mode control bit, and the program counter address extension field.

Figure 4-2 shows the condition code register. Detailed descriptions of each status indicator and field in the register follow the figure.

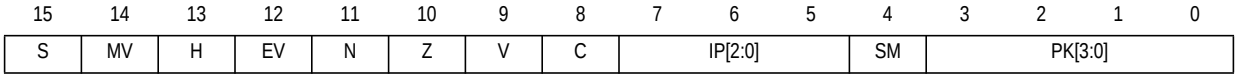


Figure 4-2 Condition Code Register

S — STOP Enable
0 = Stop clock when LPSTOP instruction is executed.
1 = Perform NOP when LPSTOP instruction is executed.

MV — Accumulator M overflow flag
MV is set when an overflow into AM35 has occurred.

H — Half Carry Flag
H is set when a carry from A3 or B3 occurs during BCD addition.

EV — Accumulator M Extension Overflow Flag
EV is set when an overflow into AM31 has occurred.

N — Negative Flag
N is set under the following conditions:
- When the MSB is set in the operand of a read operation.
- When the MSB is set in the result of a logic or arithmetic operation.

Z — Zero Flag
Z is set under the following conditions:
- When all bits are zero in the operand of a read operation.
- When all bits are zero in the result of a logic or arithmetic operation.

V — Overflow Flag
V is set when a two’s complement overflow occurs as the result of an operation.

C — Carry Flag
C is set when a carry or borrow occurs during an arithmetic operation. This flag is also used during shift and rotate to facilitate multiple word operations.

IP[2:0] — Interrupt Priority Field
The priority value in this field (0 to 7) is used to mask interrupts.

Freescale Semiconductor, Inc.

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

SM — Saturate Mode Bit

When SM is set and either EV or MV is set, data read from AM using TMER or TMET is given maximum positive or negative value, depending on the state of the AM sign bit before overflow.

PK[3:0] — Program Counter Address Extension Field

This field is concatenated with the program counter to form a 20-bit address.

4.2.6 Address Extension Register and Address Extension Fields

There are six 4-bit address extension fields. EK, XK, YK, and ZK are contained by the address extension register (K), PK is part of the CCR, and SK stands alone.

Extension fields are the bank portions of 20-bit concatenated bank:byte addresses used in the CPU16 linear memory management scheme.

All extension fields except EK correspond directly to a register. XK, YK, and ZK extend registers IX, IY, and IZ. PK extends the PC; and SK extends the SP. EK holds the four MSB of the 20-bit address used by the extended addressing mode.

4.2.7 Multiply and Accumulate Registers

The multiply and accumulate (MAC) registers are part of a CPU submodule that performs repetitive signed fractional multiplication and stores the cumulative result. These operations are part of control-oriented digital signal processing.

There are four MAC registers. Register H contains the 16-bit signed fractional multiplier. Register I contains the 16-bit signed fractional multiplicand. Accumulator M is a specialized 36-bit product accumulation register. XMSK and YMSK contain 8-bit mask values used in modulo addressing.

The CPU16 has a special subset of signal processing instructions that manipulate the MAC registers and perform signal processing calculations.

4.3 Memory Management

The CPU16 provides a 1-Mbyte address space. There are 16 banks within the address space. Each bank is made up of 64 Kbytes addressed from $0000 to $FFFF. Banks are selected by means of the address extension fields associated with individual CPU16 registers.

In addition, address space can be split into discrete 1-Mbyte program and data spaces by externally decoding the MCU's function code outputs. When this technique is used, instruction fetches and reset vector fetches access program space, while exception vector fetches (other than for reset), data accesses, and stack accesses are made in data space.

Freescale Semiconductor, Inc.

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

4.3.1 Address Extension

All CPU16 resources used to generate addresses are effectively 20 bits wide. These resources include the index registers, program counter, and stack pointer. All addressing modes use 20-bit addresses.

Twenty-bit addresses are formed from a 16-bit byte address generated by an individual CPU16 register and a 4-bit address extension contained in an associated extension field. The byte address corresponds to ADDR[15:0] and the address extension corresponds to ADDR[19:16].

4.3.2 Extension Fields

Each of the six address extension fields is used for a different type of access. All but EK are associated with particular CPU16 registers. There are several ways to manipulate extension fields and the address map. Refer to the *CPU16 Reference Manual* (CPU16RM/AD) for detailed information.

4.4 Data Types

The CPU16 uses the following types of data:

- Bits
- 4-bit signed integers
- 8-bit (byte) signed and unsigned integers
- 8-bit, 2-digit binary coded decimal (BCD) numbers
- 16-bit (word) signed and unsigned integers
- 32-bit (long word) signed and unsigned integers
- 16-bit signed fractions
- 32-bit signed fractions
- 36-bit signed fixed-point numbers
- 20-bit effective addresses

There are eight bits in a byte and 16 bits in a word. Bit set and clear instructions use both byte and word operands. Bit test instructions use byte operands.

Negative integers are represented in two's complement form. Four-bit signed integers, packed two to a byte, are used only as X and Y offsets in MAC and RMAC operations. 32-bit integers are used only by extended multiply and divide instructions, and by the associated LDED and STED instructions.

BCD numbers are packed, two digits per byte. BCD operations use byte operands.

Signed 16-bit fractions are used by the fractional multiplication instructions, and as multiplicand and multiplier operands in the MAC unit. Bit 15 is the sign bit, and there is an implied radix point between bits 15 and 14. There are 15 bits of magnitude. The range of values is –1 ($8000) to $1 - 2^{-15}$ ($7FFF).

Signed 32-bit fractions are used only by the fractional multiplication and division instructions. Bit 31 is the sign bit. An implied radix point lies between bits 31 and 30. There are 31 bits of magnitude. The range of values is –1 ($80000000) to $1 - 2^{-31}$ ($7FFFFFFF).

Freescale Semiconductor, Inc.

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

Signed 36-bit fixed-point numbers are used only by the MAC unit. Bit 35 is the sign bit. Bits [34:31] are sign extension bits. There is an implied radix point between bits 31 and 30. There are 31 bits of magnitude, but use of the extension bits allows representation of numbers in the range –16 ($800000000) to 15.999969482 ($7FFFFFFF).

4.5 Memory Organization

Both program and data memory are divided into sixteen 64-Kbyte banks. Addressing is linear. A 20-bit extended address can access any byte location in the appropriate address space.

A word is composed of two consecutive bytes. A word address is normally an even byte address. Byte 0 of a word has a lower 16-bit address than byte 1. Long words and 32-bit signed fractions consist of two consecutive words, and are normally accessed at the address of byte 0 in word 0.

Instruction fetches always access word addresses. Word operands are normally accessed at even byte addresses, but can be accessed at odd byte addresses, with a substantial performance penalty.

To permit compatibility with the M68HC11, misaligned word transfers and misaligned stack accesses are allowed. Transferring a misaligned word requires two successive byte transfer operations.

Figure 4-3 shows how each CPU16 data type is organized in memory. Consecutive even addresses show size and alignment.

Freescale Semiconductor, Inc.

Address	Type															
$0000	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
$0002	BYTE0								BYTE1							
$0004	±	X OFFSET			±	Y OFFSET			±	X OFFSET			±	Y OFFSET		
$0006	BCD1				BCD0				BCD1				BCD0			
$0008	WORD 0															
$000A	WORD 1															
$000C	MSW LONG WORD 0															
$000E	LSW LONG WORD 0															
$0010	MSW LONG WORD 1															
$0012	LSW LONG WORD 1															
$0014	±	⇐ (Radix Point)							16-BIT SIGNED FRACTION 0							
$0016	±	⇐ (Radix Point)							16-BIT SIGNED FRACTION 1							
$0018	±	⇐ (Radix Point)							MSW 32-BIT SIGNED FRACTION 0							
$001A	LSW 32-BIT SIGNED FRACTION 0															0
$001C	±	⇐ (Radix Point)							MSW 32-BIT SIGNED FRACTION 1							
$001E	LSW 32-BIT SIGNED FRACTION 1															0

MAC Data Types																
35			32	31												16
±	«	«	«	«	⇐ (Radix Point)											MSW 32-BIT SIGNED FRACTION
				15												0
				LSW 32-BIT SIGNED FRACTION												
				±	⇐ (Radix Point)											16-BIT SIGNED FRACTION

Address Data Type																
19			16	15												0
4-Bit Address Extension				16-Bit Byte Address												

Figure 4-3 Data Types and Memory Organization

4.6 Addressing Modes

The CPU16 uses nine types of addressing. There are one or more addressing modes within each type. **Table 4-1** shows the addressing modes.

Freescale Semiconductor, Inc.

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

Table 4-1 Addressing Modes

Mode	Mnemonic	Description
Accumulator Offset	E,X	Index register X with accumulator E offset
	E,Y	Index register Y with accumulator E offset
	E,Z	Index register Z with accumulator E offset
Extended	EXT	Extended
	EXT20	20-bit extended
Immediate	IMM8	8-bit immediate
	IMM16	16-bit immediate
Indexed 8-Bit	IND8, X	Index register X with unsigned 8-bit offset
	IND8, Y	Index register Y with unsigned 8-bit offset
	IND8, Z	Index register Z with unsigned 8-bit offset
Indexed 16-Bit	IND16, X	Index register X with signed 16-bit offset
	IND16, Y	Index register Y with signed 16-bit offset
	IND16, Z	Index register Z with signed 16-bit offset
Indexed 20-Bit	IND20, X	Index register X with signed 20-bit offset
	IND20, Y	Index register Y with signed 20-bit offset
	IND20, Z	Index register Z with signed 20-bit offset
Inherent	INH	Inherent
Post-Modified Index	IXP	Signed 8-bit offset added to index register X after effective address is used
Relative	REL8	8-bit relative
	REL16	16-bit relative

All modes generate ADDR[15:0]. This address is combined with ADDR[19:16] from an operand or an extension field to form a 20-bit effective address.

NOTE

Access across 64-Kbyte address boundaries is transparent. ADDR[19:16] of the effective address are changed to make an access across a bank boundary. Extension field values will not change as a result of effective address computation.

4.6.1 Immediate Addressing Modes

In the immediate modes, an argument is contained in a byte or word immediately following the instruction. For IMM8 and IMM16 modes, the effective address is the address of the argument.

There are three specialized forms of IMM8 addressing.

- The AIS, AIX, AIY, AIZ, ADDD, and ADDE instructions decrease execution time by sign-extending the 8-bit immediate operand to 16 bits, then adding it to an appropriate register.
- The MAC and RMAC instructions use an 8-bit immediate operand to specify two signed 4-bit index register offsets.
- The PSHM and PULM instructions use an 8-bit immediate mask operand to indicate which registers must be pushed to or pulled from the stack.

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

4.6.2 Extended Addressing Modes

Regular extended mode instructions contain ADDR[15:0] in the word following the op-code. The effective address is formed by concatenating the EK field and the 16-bit byte address. EXT20 mode is used only by the JMP and JSR instructions. These instructions contain a 20-bit effective address that is zero-extended to 24 bits to give the instruction an even number of bytes.

4.6.3 Indexed Addressing Modes

In the indexed modes, registers IX, IY, and IZ, together with their associated extension fields, are used to calculate the effective address.

For 8-bit indexed modes an 8-bit unsigned offset contained in the instruction is added to the value contained in an index register and its extension field.

For 16-bit modes, a 16-bit signed offset contained in the instruction is added to the value contained in an index register and its extension field.

For 20-bit modes, a 20-bit signed offset (zero-extended to 24 bits) is added to the value contained in an index register. These modes are used for JMP and JSR instructions only.

4.6.4 Inherent Addressing Mode

Inherent mode instructions use information directly available to the processor to determine the effective address. Operands, if any, are system resources and are thus not fetched from memory.

4.6.5 Accumulator Offset Addressing Mode

Accumulator offset modes form an effective address by sign-extending the content of accumulator E to 20 bits, then adding the result to an index register and its associated extension field. This mode allows use of an index register and an accumulator within a loop without corrupting accumulator D.

4.6.6 Relative Addressing Modes

Relative modes are used for branch and long branch instructions. If a branch condition is satisfied, a byte or word signed two’s complement offset is added to the concatenated PK field and program counter. The new PK : PC value is the effective address.

4.6.7 Post-Modified Index Addressing Mode

Post-modified index mode is used by the MOVB and MOVW instructions. A signed 8-bit offset is added to index register X after the effective address formed by XK : IX is used.

Freescale Semiconductor, Inc.

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

4.6.8 Use of CPU16 Indexed Mode to Replace M68HC11 Direct Mode

In M68HC11 systems, the direct addressing mode can be used to perform rapid accesses to RAM or I/O mapped from $0000 to $00FF. The CPU16 uses the first 512 bytes of bank 0 for exception vectors. To provide an enhanced replacement for the M68HC11's direct addressing mode, the ZK field and index register Z have been assigned reset initialization vectors. By resetting the ZK field to a chosen page and using indexed mode addressing, a programmer can access useful data structures anywhere in the address map.

4.7 Instruction Set

The CPU16 instruction set is based on the M68HC11 instruction set, but the opcode map has been rearranged to maximize performance with a 16-bit data bus. Most M68HC11 code can run on the CPU16 following reassembly. The user must take into account changed instruction times, the interrupt mask, and the changed interrupt stack frame (refer to *Transporting M68HC11 Code to M68HC16 Devices*,Freescale Programming Note M68HC16PN01/D, for more information).

4.7.1 Instruction Set Summary

Table 4-2 is a quick reference to the entire CPU16 instruction set. Refer to the *CPU16 Reference Manual* (CPU16RM/AD) for detailed information about each instruction, assembler syntax, and condition code evaluation. **Table 4-3** provides a key to the table nomenclature.

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
ABA	Add B to A	(A) + (B) ⇒ A	INH	370B	—	2	—	—	Δ	—	Δ	Δ	Δ	Δ
ABX	Add B to IX	(XK : IX) + (000 : B) ⇒ XK : IX	INH	374F	—	2	—	—	—	—	—	—	—	—
ABY	Add B to IY	(YK : IY) + (000 : B) ⇒ YK : IY	INH	375F	—	2	—	—	—	—	—	—	—	—
ABZ	Add B to IZ	(ZK : IZ) + (000 : B) ⇒ ZK : IZ	INH	376F	—	2	—	—	—	—	—	—	—	—
ACE	Add E to AM	(AM[31:16]) + (E) ⇒ AM	INH	3722	—	2	—	Δ	—	Δ	—	—	—	—
ACED	Add E : D to AM	(AM) + (E : D) ⇒ AM	INH	3723	—	4	—	Δ	—	Δ	—	—	—	—
ADCA	Add with Carry to A	(A) + (M) + C ⇒ A	IND8, X	43	ff	6	—	—	Δ	—	Δ	Δ	Δ	Δ
			IND8, Y	53	ff	6								
			IND8, Z	63	ff	6								
			IMM8	73	ii	2								
			IND16, X	1743	gggg	6								
			IND16, Y	1753	gggg	6								
			IND16, Z	1763	gggg	6								
			EXT	1773	hh ll	6								
			E, X	2743	—	6								
			E, Y	2753	—	6								
			E, Z	2763	—	6								
ADCB	Add with Carry to B	(B) + (M) + C ⇒ B	IND8, X	C3	ff	6	—	—	Δ	—	Δ	Δ	Δ	Δ
			IND8, Y	D3	ff	6								
			IND8, Z	E3	ff	6								
			IMM8	F3	ii	2								
			IND16, X	17C3	gggg	6								
			IND16, Y	17D3	gggg	6								
			IND16, Z	17E3	gggg	6								
			EXT	17F3	hh ll	6								
			E, X	27C3	—	6								
			E, Y	27D3	—	6								
			E, Z	27E3	—	6								
ADCD	Add with Carry to D	(D) + (M : M + 1) + C ⇒ D	IND8, X	83	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	93	ff	6								
			IND8, Z	A3	ff	6								
			IMM16	37B3	jj kk	4								
			IND16, X	37C3	gggg	6								
			IND16, Y	37D3	gggg	6								
			IND16, Z	37E3	gggg	6								
			EXT	37F3	hh ll	6								
			E, X	2783	—	6								
			E, Y	2793	—	6								
			E, Z	27A3	—	6								
ADCE	Add with Carry to E	(E) + (M : M + 1) + C ⇒ E	IMM16	3733	jj kk	4	—	—	—	—	Δ	Δ	Δ	Δ
			IND16, X	3743	gggg	6								
			IND16, Y	3753	gggg	6								
			IND16, Z	3763	gggg	6								
			EXT	3773	hh ll	6								
ADDA	Add to A	(A) + (M) ⇒ A	IND8, X	41	ff	6	—	—	Δ	—	Δ	Δ	Δ	Δ
			IND8, Y	51	ff	6								
			IND8, Z	61	ff	6								
			IMM8	71	ii	2								
			IND16, X	1741	gggg	6								
			IND16, Y	1751	gggg	6								
			IND16, Z	1761	gggg	6								
			EXT	1771	hh ll	6								
			E, X	2741	—	6								
			E, Y	2751	—	6								
			E, Z	2761	—	6								

Freescale Semiconductor, Inc.

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

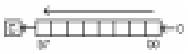
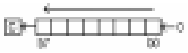
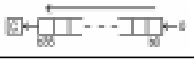
Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
ADDB	Add to B	(B) + (M) ⇒ B	IND8, X	C1	ff	6	—	—	Δ	—	Δ	Δ	Δ	Δ
			IND8, Y	D1	ff	6								
			IND8, Z	E1	ff	6								
			IMM8	F1	ii	2								
			IND16, X	17C1	gggg	6								
			IND16, Y	17D1	gggg	6								
			IND16, Z	17E1	gggg	6								
			EXT	17F1	hh ll	6								
			E, X	27C1	—	6								
			E, Y	27D1	—	6								
			E, Z	27E1	—	6								
ADDD	Add to D	(D) + (M : M + 1) ⇒ D	IND8, X	81	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	91	ff	6								
			IND8, Z	A1	ff	6								
			IMM8	FC	ii	2								
			IMM16	37B1	jj kk	4								
			IND16, X	37C1	gggg	6								
			IND16, Y	37D1	gggg	6								
			IND16, Z	37E1	gggg	6								
			EXT	37F1	hh ll	6								
			E, X	2781	—	6								
			E, Y	2791	—	6								
			E, Z	27A1	—	6								
ADDE	Add to E	(E) + (M : M + 1) ⇒ E	IMM8	7C	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			IMM16	3731	jj kk	4								
			IND16, X	3741	gggg	6								
			IND16, Y	3751	gggg	6								
			IND16, Z	3761	gggg	6								
			EXT	3771	hh ll	6								
ADE	Add D to E	(E) + (D) ⇒ E	INH	2778	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ADX	Add D to IX	(XK : IX) + (20 « D) ⇒ XK : IX	INH	37CD	—	2	—	—	—	—	—	—	—	—
ADY	Add D to IY	(YK : IY) + (20 « D) ⇒ YK : IY	INH	37DD	—	2	—	—	—	—	—	—	—	—
ADZ	Add D to IZ	(ZK : IZ) + (20 « D) ⇒ ZK : IZ	INH	37ED	—	2	—	—	—	—	—	—	—	—
AEX	Add E to IX	(XK : IX) + (20 « E) ⇒ XK : IX	INH	374D	—	2	—	—	—	—	—	—	—	—
AEY	Add E to IY	(YK : IY) + (20 « E) ⇒ YK : IY	INH	375D	—	2	—	—	—	—	—	—	—	—
AEZ	Add E to IZ	(ZK : IZ) + (20 « E) ⇒ ZK : IZ	INH	376D	—	2	—	—	—	—	—	—	—	—
AIS	Add Immediate Data to Stack Pointer	(SK : SP) + (20 « IMM) ⇒ SK : SP	IMM8	3F	ii	2	—	—	—	—	—	—	—	—
			IMM16	373F	jj kk	4								
AIX	Add Immediate Value to IX	(XK : IX) + (20 « IMM) ⇒ XK : IX	IMM8	3C	ii	2	—	—	—	—	—	Δ	—	—
			IMM16	373C	jj kk	4								
AIY	Add Immediate Value to IY	(YK : IY) + (20 « IMM) ⇒ YK : IY	IMM8	3D	ii	2	—	—	—	—	—	Δ	—	—
			IMM16	373D	jj kk	4								
AIZ	Add Immediate Value to IZ	(ZK : IZ) + (20 « IMM) ⇒ ZK : IZ	IMM8	3E	ii	2	—	—	—	—	—	Δ	—	—
			IMM16	373E	jj kk	4								
ANDA	AND A	(A) • (M) ⇒ A	IND8, X	46	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	56	ff	6								
			IND8, Z	66	ff	6								
			IMM8	76	ii	2								
			IND16, X	1746	gggg	6								
			IND16, Y	1756	gggg	6								
			IND16, Z	1766	gggg	6								
			EXT	1776	hh ll	6								
			E, X	2746	—	6								
			E, Y	2756	—	6								
			E, Z	2766	—	6								

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
ANDB	AND B	(B) • (M) ⇒ B	IND8, X	C6	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	D6	ff	6								
			IND8, Z	E6	ff	6								
			IMM8	F6	ii	2								
			IND16, X	17C6	gggg	6								
			IND16, Y	17D6	gggg	6								
			IND16, Z	17E6	gggg	6								
			EXT	17F6	hh ll	6								
			E, X	27C6	—	6								
			E, Y	27D6	—	6								
			E, Z	27E6	—	6								
ANDD	AND D	(D) • (M : M + 1) ⇒ D	IND8, X	86	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	96	ff	6								
			IND8, Z	A6	ff	6								
			IMM16	37B6	jj kk	4								
			IND16, X	37C6	gggg	6								
			IND16, Y	37D6	gggg	6								
			IND16, Z	37E6	gggg	6								
			EXT	37F6	hh ll	6								
			E, X	2786	—	6								
			E, Y	2796	—	6								
			E, Z	27A6	—	6								
ANDE	AND E	(E) • (M : M + 1) ⇒ E	IMM16	3736	jj kk	4	—	—	—	—	Δ	Δ	0	—
			IND16, X	3746	gggg	6								
			IND16, Y	3756	gggg	6								
			IND16, Z	3766	gggg	6								
			EXT	3776	hh ll	6								
ANDP[1]	AND CCR	(CCR) • IMM16⇒ CCR	IMM16	373A	jj kk	4	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
ASL	Arithmetic Shift Left		IND8, X	04	ff	8	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	14	ff	8								
			IND8, Z	24	ff	8								
			IND16, X	1704	gggg	8								
			IND16, Y	1714	gggg	8								
			IND16, Z	1724	gggg	8								
			EXT	1734	hh ll	8								
ASLA	Arithmetic Shift Left A		INH	3704	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLB	Arithmetic Shift Left B		INH	3714	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLD	Arithmetic Shift Left D		INH	27F4	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLE	Arithmetic Shift Left E		INH	2774	—	2	—	—	—	—				
ASLM	Arithmetic Shift Left AM		INH	27B6	—	4	—		—			—	—	
ASLW	Arithmetic Shift Left Word		IND16, X	2704	gggg	8	—	—	—	—				
			IND16, Y	2714	gggg	8								
			IND16, Z	2724	gggg	8								
			EXT	2734	hh ll	8								
ASR	Arithmetic Shift Right		IND8, X	0D	ff	8	—	—	—	—				
			IND8, Y	1D	ff	8								
			IND8, Z	2D	ff	8								
			IND16, X	170D	gggg	8								
			IND16, Y	171D	gggg	8								
			IND16, Z	172D	gggg	8								
			EXT	173D	hh ll	8								

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
ASRA	Arithmetic Shift Right A		INH	370D	—	2	—	—	—	—				
ASRB	Arithmetic Shift Right B		INH	371D	—	2	—	—	—	—				
ASRD	Arithmetic Shift Right D		INH	27FD	—	2	—	—	—	—				
ASRE	Arithmetic Shift Right E		INH	277D	—	2	—	—	—	—				
ASRM	Arithmetic Shift Right AM		INH	27BA	—	4	—	—	—			—	—	
ASRW	Arithmetic Shift Right Word		IND16, X IND16, Y IND16, Z EXT	270D 271D 272D 273D	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—				
BCC[2]	Branch if Carry Clear	If C = 0, branch	REL8	B4	rr	6, 2	—	—	—	—	—	—	—	—
BCLR	Clear Bit(s)	(M) • ($\overline{\text{Mask}}$) ⇒ M	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	1708 1718 1728 08 18 28 38	mm ff mm ff mm ff mm gggg mm gggg mm gggg mm hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	0	—
BCLRW	Clear Bit(s) in a Word	(M : M + 1) • ($\overline{\text{Mask}}$) ⇒ M : M + 1	IND16, X IND16, Y IND16, Z EXT	2708 2718 2728 2738	gggg mmmm gggg mmmm gggg mmmm hh ll mmmm	10 10 10 10	—	—	—	—	Δ	Δ	0	—
BCS[2]	Branch if Carry Set	If C = 1, branch	REL8	B5	rr	6, 2	—	—	—	—	—	—	—	—
BEQ[2]	Branch if Equal	If Z = 1, branch	REL8	B7	rr	6, 2	—	—	—	—	—	—	—	—
BGE[2]	Branch if Greater Than or Equal to Zero	If N ⊕ V = 0, branch	REL8	BC	rr	6, 2	—	—	—	—	—	—	—	—
BGND	Enter Background Debug Mode	If BDM enabled, begin debug; else, illegal instruction trap	INH	37A6	—	—	—	—	—	—	—	—	—	—
BGT[2]	Branch if Greater Than Zero	If Z + (N ⊕ V) = 0, branch	REL8	BE	rr	6, 2	—	—	—	—	—	—	—	—
BHI[2]	Branch if Higher	If C + Z = 0, branch	REL8	B2	rr	6, 2	—	—	—	—	—	—	—	—
BITA	Bit Test A	(A) • (M)	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	49 59 69 79 1749 1759 1769 1779 2749 2759 2769	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
BITB	Bit Test B	(B) • (M)	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	C9 D9 E9 F9 17C9 17D9 17E9 17F9 27C9 27D9 27E9	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—
BLE[2]	Branch if Less Than or Equal to Zero	If Z ✣ (N ⊕ V) = 1, branch	REL8	BF	rr	6, 2	—	—	—	—	—	—	—	—
BLS[2]	Branch if Lower or Same	If C ✣ Z = 1, branch	REL8	B3	rr	6, 2	—	—	—	—	—	—	—	—
BLT[2]	Branch if Less Than Zero	If N ⊕ V = 1, branch	REL8	BD	rr	6, 2	—	—	—	—	—	—	—	—
BMI[2]	Branch if Minus	If N = 1, branch	REL8	BB	rr	6, 2	—	—	—	—	—	—	—	—
BNE[2]	Branch if Not Equal	If Z = 0, branch	REL8	B6	rr	6, 2	—	—	—	—	—	—	—	—
BPL[2]	Branch if Plus	If N = 0, branch	REL8	BA	rr	6, 2	—	—	—	—	—	—	—	—
BRA	Branch Always	If 1 = 1, branch	REL8	B0	rr	6	—	—	—	—	—	—	—	—
BRCLR[2]	Branch if Bit(s) Clear	If (M) • (Mask) = 0, branch	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	CB DB EB 0A 1A 2A 3A	mm ff rr mm ff rr mm ff rr mm gggg rrrr mm gggg rrrr mm gggg rrrr mm hh ll rrrr	10, 12 10, 12 10, 12 10, 14 10, 14 10, 14 10, 14	—	—	—	—	—	—	—	—
BRN	Branch Never	If 1 = 0, branch	REL8	B1	rr	2	—	—	—	—	—	—	—	—
BRSET[2]	Branch if Bit(s) Set	If ($\overline{M}$) • (Mask) = 0, branch	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	8B 9B AB 0B 1B 2B 3B	mm ff rr mm ff rr mm ff rr mm gggg rrrr mm gggg rrrr mm gggg rrrr mm hh ll rrrr	10, 12 10, 12 10, 12 10, 14 10, 14 10, 14 10, 14	—	—	—	—	—	—	—	—
BSET	Set Bit(s)	(M) ✣ (Mask) ⇒ M	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	1709 1719 1729 09 19 29 39	mm ff mm ff mm ff mm gggg mm gggg mm gggg mm hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	0	Δ
BSETW	Set Bit(s) in Word	(M : M + 1) ✣ (Mask) ⇒ M : M + 1	IND16, X IND16, Y IND16, Z EXT	2709 2719 2729 2739	gggg mmmm gggg mmmm gggg mmmm hh ll mmmm	10 10 10 10	—	—	—	—	Δ	Δ	0	Δ

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
BSR	Branch to Subroutine	(PK : PC) - 2 ⇒ PK : PC Push (PC) (SK : SP) - 2 ⇒ SK : SP Push (CCR) (SK : SP) - 2 ⇒ SK : SP (PK : PC) + Offset ⇒ PK : PC	REL8	36	rr	10	—	—	—	—	—	—	—	—
BVC[2]	Branch if Overflow Clear	If V = 0, branch	REL8	B8	rr	6, 2	—	—	—	—	—	—	—	—
BVS[2]	Branch if Overflow Set	If V = 1, branch	REL8	B9	rr	6, 2	—	—	—	—	—	—	—	—
CBA	Compare A to B	(A) – (B)	INH	371B	—	2	—	—	—	—	Δ	Δ	Δ	Δ
CLR	Clear a Byte in Memory	$00 ⇒ M	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	05 15 25 1705 1715 1725 1735	ff ff ff gggg gggg gggg hh ll	4 4 4 6 6 6 6	—	—	—	—	0	1	0	0
CLRA	Clear A	$00 ⇒ A	INH	3705	—	2	—	—	—	—	0	1	0	0
CLRB	Clear B	$00 ⇒ B	INH	3715	—	2	—	—	—	—	0	1	0	0
CLRD	Clear D	$0000 ⇒ D	INH	27F5	—	2	—	—	—	—	0	1	0	0
CLRE	Clear E	$0000 ⇒ E	INH	2775	—	2	—	—	—	—	0	1	0	0
CLRM	Clear AM	$000000000 ⇒ AM[35:0]	INH	27B7	—	2	—	0	—	0	—	—	—	—
CLRW	Clear a Word in Memory	$0000 ⇒ M : M + 1	IND16, X IND16, Y IND16, Z EXT	2705 2715 2725 2735	gggg gggg gggg hh ll	6 6 6 6	—	—	—	—	0	1	0	0
CMPA	Compare A to Memory	(A) – (M)	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	48 58 68 78 1748 1758 1768 1778 2748 2758 2768	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ
CMPB	Compare B to Memory	(B) – (M)	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	C8 D8 E8 F8 17C8 17D8 17E8 17F8 27C8 27D8 27E8	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ
COM	One's Complement	$FF – (M) ⇒ M, or $\overline{M}$ ⇒ M	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	00 10 20 1700 1710 1720 1730	ff ff ff gggg gggg gggg hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	0	1
COMA	One's Complement A	$FF – (A) ⇒ A, or $\overline{M}$ ⇒ A	INH	3700	—	2	—	—	—	—	Δ	Δ	0	1
COMB	One's Complement B	$FF – (B) ⇒ B, or $\overline{B}$ ⇒ B	INH	3710	—	2	—	—	—	—	Δ	Δ	0	1
COMD	One's Complement D	$FFFF – (D) ⇒ D, or $\overline{D}$ ⇒ D	INH	27F0	—	2	—	—	—	—	Δ	Δ	0	1
COME	One's Complement E	$FFFF – (E) ⇒ E, or $\overline{E}$ ⇒ E	INH	2770	—	2	—	—	—	—	Δ	Δ	0	1
COMW	One's Complement Word	$FFFF – M : M + 1 ⇒ M : M + 1, or ($\overline{M : M + 1}$) ⇒ M : M + 1	IND16, X IND16, Y IND16, Z EXT	2700 2710 2720 2730	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	Δ	Δ	0	1

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

Freescale Semiconductor, Inc.

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
CPD	Compare D to Memory	(D) – (M : M + 1)	IND8, X	88	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	98	ff	6								
			IND8, Z	A8	ff	6								
			IMM16	37B8	jj kk	4								
			IND16, X	37C8	gggg	6								
			IND16, Y	37D8	gggg	6								
			IND16, Z	37E8	gggg	6								
			EXT	37F8	hh ll	6								
			E, X	2788	—	6								
			E, Y	2798	—	6								
			E, Z	27A8	—	6								
CPE	Compare E to Memory	(E) – (M : M + 1)	IMM16	3738	jjkk	4	—	—	—	—	Δ	Δ	Δ	Δ
			IND16, X	3748	gggg	6								
			IND16, Y	3758	gggg	6								
			IND16, Z	3768	gggg	6								
			EXT	3778	hhll	6								
CPS	Compare Stack Pointer to Memory	(SP) – (M : M + 1)	IND8, X	4F	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	5F	ff	6								
			IND8, Z	6F	ff	6								
			IMM16	377F	jj kk	4								
			IND16, X	174F	gggg	6								
			IND16, Y	175F	gggg	6								
			IND16, Z	176F	gggg	6								
			EXT	177F	hh ll	6								
CPX	Compare IX to Memory	(IX) – (M : M + 1)	IND8, X	4C	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	5C	ff	6								
			IND8, Z	6C	ff	6								
			IMM16	377C	jj kk	4								
			IND16, X	174C	gggg	6								
			IND16, Y	175C	gggg	6								
			IND16, Z	176C	gggg	6								
			EXT	177C	hh ll	6								
CPY	Compare IY to Memory	(IY) – (M : M + 1)	IND8, X	4D	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	5D	ff	6								
			IND8, Z	6D	ff	6								
			IMM16	377D	jj kk	4								
			IND16, X	174D	gggg	6								
			IND16, Y	175D	gggg	6								
			IND16, Z	176D	gggg	6								
			EXT	177D	hh ll	6								
CPZ	Compare IZ to Memory	(IZ) – (M : M + 1)	IND8, X	4E	ff	6	—	—	—	—				
			IND8, Y	5E	ff	6								
			IND8, Z	6E	ff	6								
			IMM16	377E	jj kk	4								
			IND16, X	174E	gggg	6								
			IND16, Y	175E	gggg	6								
			IND16, Z	176E	gggg	6								
			EXT	177E	hh ll	6								
DAA	Decimal Adjust A	$(A)_{10}$	INH	3721	—	2	—	—	—	—	Δ	Δ	U	Δ
DEC	Decrement Memory	(M) – $01 ⇒ M	IND8, X	01	ff	8	—	—	—	—	Δ	Δ	Δ	—
			IND8, Y	11	ff	8								
			IND8, Z	21	ff	8								
			IND16, X	1701	gggg	8								
			IND16, Y	1711	gggg	8								
			IND16, Z	1721	gggg	8								
			EXT	1731	hh ll	8								
DECA	Decrement A	(A) – $01 ⇒ A	INH	3701	—	2	—	—	—	—	Δ	Δ	Δ	—
DECB	Decrement B	(B) – $01 ⇒ B	INH	3711	—	2	—	—	—	—	Δ	Δ	Δ	—
DECW	Decrement Memory Word	(M : M + 1) – $0001 ⇒ M : M + 1	IND16, X	2701	gggg	8	—	—	—	—	Δ	Δ	Δ	—
			IND16, Y	2711	gggg	8								
			IND16, Z	2721	gggg	8								
			EXT	2731	hh ll	8								
EDIV	Extended Unsigned Integer Divide	(E : D) / (IX) Quotient ⇒ IX Remainder ⇒ D	INH	3728	—	24	—	—	—	—	Δ	Δ	Δ	Δ

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
EDIVS	Extended Signed Integer Divide	(E : D) / (IX) Quotient ⇒ IX Remainder ⇒ D	INH	3729	—	38	—	—	—	—	Δ	Δ	Δ	Δ
EMUL	Extended Unsigned Multiply	(E) ∗ (D) ⇒ E : D	INH	3725	—	10	—	—	—	—	Δ	Δ	—	Δ
EMULS	Extended Signed Multiply	(E) ∗ (D) ⇒ E : D	INH	3726	—	8	—	—	—	—	Δ	Δ	—	Δ
EORA	Exclusive OR A	(A) ⊕ (M) ⇒ A	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	44 54 64 74 1744 1754 1764 1774 2744 2754 2764	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—
EORB	Exclusive OR B	(B) ⊕ (M) ⇒ B	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	C4 D4 E4 F4 17C4 17D4 17E4 17F4 27C4 27D4 27E4	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—
EORD	Exclusive OR D	(D) ⊕ (M : M + 1) ⇒ D	IND8, X IND8, Y IND8, Z IMM16 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	84 94 A4 37B4 37C4 37D4 37E4 37F4 2784 2794 27A4	ff ff ff jj kk gggg gggg gggg hh ll — — —	6 6 6 4 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—
ERE	Exclusive OR E	(E) ⊕ (M : M + 1) ⇒ E	IMM16 IND16, X IND16, Y IND16, Z EXT	3734 3744 3754 3764 3774	jj kk gggg gggg gggg hh ll	4 6 6 6 6	—	—	—	—	Δ	Δ	0	—
FDIV	Fractional Unsigned Divide	(D) / (IX) ⇒ IX Remainder ⇒ D	INH	372B	—	22	—	—	—	—	—	Δ	Δ	Δ
FMULS	Fractional Signed Multiply	(E) ∗ (D) ⇒ E : D[31:1] 0 ⇒ D[0]	INH	3727	—	8	—	—	—	—	Δ	Δ	Δ	Δ
IDIV	Integer Divide	(D) / (IX) ⇒ IX Remainder ⇒ D	INH	372A	—	22	—	—	—	—	—	Δ	0	Δ
INC	Increment Memory	(M) + $01 ⇒ M	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	03 13 23 1703 1713 1723 1733	ff ff ff gggg gggg gggg hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	Δ	—
INCA	Increment A	(A) + $01 ⇒ A	INH	3703	—	2	—	—	—	—	Δ	Δ	Δ	—
INCB	Increment B	(B) + $01 ⇒ B	INH	3713	—	2	—	—	—	—	Δ	Δ	Δ	—
INCW	Increment Memory Word	(M : M + 1) + $0001 ⇒ M : M + 1	IND16, X IND16, Y IND16, Z EXT	2703 2713 2723 2733	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	Δ	Δ	Δ	—

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
JMP	Jump	⟨ea⟩ ⇒ PK : PC	EXT20 IND20, X IND20, Y IND20, Z	7A 4B 5B 6B	zb hh ll zg gggg zg gggg zg gggg	6 8 8 8	—	—	—	—	—	—	—	—
JSR	Jump to Subroutine	Push (PC) (SK : SP) – $0002 ⇒ SK : SP Push (CCR) (SK : SP) – $0002 ⇒ SK : SP ⟨ea⟩ ⇒ PK : PC	EXT20 IND20, X IND20, Y IND20, Z	FA 89 99 A9	zb hh ll zg gggg zg gggg zg gggg	10 12 12 12	—	—	—	—	—	—	—	—
LBCC[2]	Long Branch if Carry Clear	If C = 0, branch	REL16	3784	rrrr	6, 4	—	—	—	—	—	—	—	—
LBCS[2]	Long Branch if Carry Set	If C = 1, branch	REL16	3785	rrrr	6, 4	—	—	—	—	—	—	—	—
LBEQ[2]	Long Branch if Equal to Zero	If Z = 1, branch	REL16	3787	rrrr	6, 4	—	—	—	—	—	—	—	—
LBEV[2]	Long Branch if EV Set	If EV = 1, branch	REL16	3791	rrrr	6, 4	—	—	—	—	—	—	—	—
LBGE[2]	Long Branch if Greater Than or Equal to Zero	If N ⊕ V = 0, branch	REL16	378C	rrrr	6, 4	—	—	—	—	—	—	—	—
LBGT[2]	Long Branch if Greater Than Zero	If Z + (N ⊕ V) = 0, branch	REL16	378E	rrrr	6, 4	—	—	—	—	—	—	—	—
LBHI [2]	Long Branch if Higher	If C + Z = 0, branch	REL16	3782	rrrr	6, 4	—	—	—	—	—	—	—	—
LBLE[2]	Long Branch if Less Than or Equal to Zero	If Z + (N ⊕ V) = 1, branch	REL16	378F	rrrr	6, 4	—	—	—	—	—	—	—	—
LBLS[2]	Long Branch if Lower or Same	If C + Z = 1, branch	REL16	3783	rrrr	6, 4	—	—	—	—	—	—	—	—
LBLT[2]	Long Branch if Less Than Zero	If N ⊕ V = 1, branch	REL16	378D	rrrr	6, 4	—	—	—	—	—	—	—	—
LBMI[2]	Long Branch if Minus	If N = 1, branch	REL16	378B	rrrr	6, 4	—	—	—	—	—	—	—	—
LBMV[2]	Long Branch if MV Set	If MV = 1, branch	REL16	3790	rrrr	6, 4	—	—	—	—	—	—	—	—
LBNE[2]	Long Branch if Not Equal to Zero	If Z = 0, branch	REL16	3786	rrrr	6, 4	—	—	—	—	—	—	—	—
LBPL[2]	Long Branch if Plus	If N = 0, branch	REL16	378A	rrrr	6, 4	—	—	—	—	—	—	—	—
LBRA	Long Branch Always	If 1 = 1, branch	REL16	3780	rrrr	6	—	—	—	—	—	—	—	—
LBRN	Long Branch Never	If 1 = 0, branch	REL16	3781	rrrr	6	—	—	—	—	—	—	—	—
LBSR	Long Branch to Subroutine	Push (PC) (SK : SP) – 2 ⇒ SK : SP Push (CCR) (SK : SP) – 2 ⇒ SK : SP (PK : PC) + Offset ⇒ PK : PC	REL16	27F9	rrrr	10	—	—	—	—	—	—	—	—
LBVC[2]	Long Branch if Overflow Clear	If V = 0, branch	REL16	3788	rrrr	6, 4	—	—	—	—	—	—	—	—
LBVS[2]	Long Branch if Overflow Set	If V = 1, branch	REL16	3789	rrrr	6, 4	—	—	—	—	—	—	—	—
LDAA	Load A	(M) ⇒ A	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	45 55 65 75 1745 1755 1765 1775 2745 2755 2765	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—

Freescale Semiconductor, Inc.

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

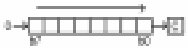
Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
LDAB	Load B	(M) ⇒ B	IND8, X	C5	ff	6	—	—	—	—	Δ	Δ	0	Δ
			IND8, Y	D5	ff	6								
			IND8, Z	E5	ff	6								
			IMM8	F5	ii	2								
			IND16, X	17C5	gggg	6								
			IND16, Y	17D5	gggg	6								
			IND16, Z	17E5	gggg	6								
			EXT	17F5	hh ll	6								
			E, X	27C5	—	6								
			E, Y	27D5	—	6								
			E, Z	27E5	—	6								
LDD	Load D	(M : M + 1) ⇒ D	IND8, X	85	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	95	ff	6								
			IND8, Z	A5	ff	6								
			IMM16	37B5	jj kk	4								
			IND16, X	37C5	gggg	6								
			IND16, Y	37D5	gggg	6								
			IND16, Z	37E5	gggg	6								
			EXT	37F5	hh ll	6								
			E, X	2785	—	6								
			E, Y	2795	—	6								
			E, Z	27A5	—	6								
LDE	Load E	(M : M + 1) ⇒ E	IMM16	3735	jj kk	4	—	—	—	—	Δ	Δ	0	—
			IND16, X	3745	gggg	6								
			IND16, Y	3755	gggg	6								
			IND16, Z	3765	gggg	6								
			EXT	3775	hh ll	6								
LDED	Load Concatenated E and D	(M : M + 1) ⇒ E (M + 2 : M + 3) ⇒ D	EXT	2771	hh ll	8	—	—	—	—	—	—	—	—
LDHI	Initialize H and I	$(M : M + 1)_X$ ⇒ H R $(M : M + 1)_Y$ ⇒ I R	INH	27B0	—	8	—	—	—	—	—	—	—	—
LDS	Load SP	(M : M + 1) ⇒ SP	IND8, X	CF	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	DF	ff	6								
			IND8, Z	EF	ff	6								
			IND16, X	17CF	gggg	6								
			IND16, Y	17DF	gggg	6								
			IND16, Z	17EF	gggg	6								
			EXT	17FF	hh ll	6								
			IMM16	37BF	jj kk	4								
LDX	Load IX	(M : M + 1) ⇒ IX	IND8, X	CC	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	DC	ff	6								
			IND8, Z	EC	ff	6								
			IMM16	37BC	jj kk	4								
			IND16, X	17CC	gggg	6								
			IND16, Y	17DC	gggg	6								
			IND16, Z	17EC	gggg	6								
			EXT	17FC	hh ll	6								
LDY	Load IY	(M : M + 1) ⇒ IY	IND8, X	CD	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	DD	ff	6								
			IND8, Z	ED	ff	6								
			IMM16	37BD	jj kk	4								
			IND16, X	17CD	gggg	6								
			IND16, Y	17DD	gggg	6								
			IND16, Z	17ED	gggg	6								
			EXT	17FD	hh ll	6								
LDZ	Load IZ	(M : M + 1) ⇒ IZ	IND8, X	CE	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	DE	ff	6								
			IND8, Z	EE	ff	6								
			IMM16	37BE	jj kk	4								
			IND16, X	17CE	gggg	6								
			IND16, Y	17DE	gggg	6								
			IND16, Z	17EE	gggg	6								
			EXT	17FE	hh ll	6								

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
LPSTOP	Low Power Stop	If $\overline{S}$ then STOP else NOP	INH	27F1	—	4, 20	—	—	—	—	—	—	—	—
LSR	Logical Shift Right		IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	0F 1F 2F 170F 171F 172F 173F	ff ff ff gggg gggg gggg hh ll	8 8 8 8 8 8 8	—	—	—	—	0	Δ	Δ	Δ
LSRA	Logical Shift Right A		INH	370F	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRB	Logical Shift Right B		INH	371F	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRD	Logical Shift Right D		INH	27FF	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRE	Logical Shift Right E		INH	277F	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRW	Logical Shift Right Word		IND16, X IND16, Y IND16, Z EXT	270F 271F 272F 273F	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	0	Δ	Δ	Δ
MAC	Multiply and Accumulate Signed 16-Bit Fractions	(HR) * (IR) ⇒ E : D (AM) + (E : D) ⇒ AM Qualified (IX) ⇒ IX Qualified (IY) ⇒ IY (HR) ⇒ IZ $(M : M + 1)_X$ ⇒ HR $(M : M + 1)_Y$ ⇒ IR	IMM8	7B	xoyo	12	—	Δ	—	Δ	—	—	Δ	—
MOVB	Move Byte	$(M_1) \Rightarrow M_2$	IXP to EXT EXT to IXP EXT to EXT	30 32 37FE	ff hh ll ff hh ll hh ll hh ll	8 8 10	—	—	—	—	Δ	Δ	0	—
MOVW	Move Word	$(M : M + 1_1) \Rightarrow M : M + 1_2$	IXP to EXT EXT to IXP EXT to EXT	31 33 37FF	ff hh ll ff hh ll hh ll hh ll	8 8 10	—	—	—	—	Δ	Δ	0	—
MUL	Multiply	(A) * (B) ⇒ D	INH	3724	—	10	—	—	—	—	—	—	—	Δ
NEG	Negate Memory	$00 – (M) ⇒ M	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	02 12 22 1702 1712 1722 1732	ff ff ff gggg gggg gggg hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
NEGA	Negate A	$00 – (A) ⇒ A	INH	3702	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGB	Negate B	$00 – (B) ⇒ B	INH	3712	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGD	Negate D	$0000 – (D) ⇒ D	INH	27F2	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGE	Negate E	$0000 – (E) ⇒ E	INH	2772	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGW	Negate Memory Word	$0000 – (M : M + 1) ⇒ M : M + 1	IND16, X IND16, Y IND16, Z EXT	2702 2712 2722 2732	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
NOP	Null Operation	—	INH	274C	—	2	—	—	—	—	—	—	—	—

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
ORAA	OR A	(A) ✣ (M) ⇒ A	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	47 57 67 77 1747 1757 1767 1777 2747 2757 2767	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—
ORAB	OR B	(B) ✣ (M) ⇒ B	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	C7 D7 E7 F7 17C7 17D7 17E7 17F7 27C7 27D7 27E7	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—
ORD	OR D	(D) ✣ (M : M + 1) ⇒ D	IND8, X IND8, Y IND8, Z IMM16 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	87 97 A7 37B7 37C7 37D7 37E7 37F7 2787 2797 27A7	ff ff ff jj kk gggg gggg gggg hh ll — — —	6 6 6 4 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—
ORE	OR E	(E) ✣ (M : M + 1) ⇒ E	IMM16 IND16, X IND16, Y IND16, Z EXT	3737 3747 3757 3767 3777	jj kk gggg gggg gggg hh ll	4 6 6 6 6	—	—	—	—	Δ	Δ	0	—
ORP [1]	OR Condition Code Register	(CCR) ✣ IMM16 ⇒ CCR	IMM16	373B	jj kk	4	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
PSHA	Push A	(SK : SP) + $0001 ⇒ SK : SP Push (A) (SK : SP) – $0002 ⇒ SK : SP	INH	3708	—	4	—	—	—	—	—	—	—	—
PSHB	Push B	(SK : SP) + $0001 ⇒ SK : SP Push (B) (SK : SP) – $0002 ⇒ SK : SP	INH	3718	—	4	—	—	—	—	—	—	—	—
PSHM	Push Multiple Registers Mask bits: 0 = D 1 = E 2 = IX 3 = IY 4 = IZ 5 = K 6 = CCR 7 = (Reserved)	For mask bits 0 to 7: If mask bit set Push register (SK : SP) – 2 ⇒ SK : SP	IMM8	34	ii	4 + 2N N = number of registers pushed	—	—	—	—	—	—	—	—
PSHMAC	Push MAC Registers	MAC Registers ⇒ Stack	INH	27B8	—	14	—	—	—	—	—	—	—	—
PULA	Pull A	(SK : SP) + $0002 ⇒ SK : SP Pull (A) (SK : SP) – $0001 ⇒ SK : SP	INH	3709	—	6	—	—	—	—	—	—	—	—
PULB	Pull B	(SK : SP) + $0002 ⇒ SK : SP Pull (B) (SK : SP) – $0001 ⇒ SK : SP	INH	3719	—	6	—	—	—	—	—	—	—	—

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
PULM[1]	Pull Multiple Registers Mask bits: 0 = CCR[15:4] 1 = K 2 = IZ 3 = IY 4 = IX 5 = E 6 = D 7 = (Reserved)	For mask bits 0 to 7: If mask bit set (SK : SP) + 2 ⇒ SK : SP Pull register	IMM8	35	ii	4+2(N+1) N = number of registers pulled	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
PULMAC	Pull MAC State	Stack ⇒ MAC Registers	INH	27B9	—	16	—	—	—	—	—	—	—	—
RMAC	Repeating Multiply and Accumulate Signed 16-Bit Fractions	Repeat until (E) < 0 (AM) + (H) ∗ (I) ⇒ AM Qualified (IX) ⇒ IX; Qualified (IY) ⇒ IY; $(M : M + 1)_X$ ⇒ H; $(M : M + 1)_Y$ ⇒ I (E) – 1 ⇒ E Until (E) < $0000	IMM8	FB	xoyo	6 + 12 per iteration	—	Δ	—	Δ	—	—	—	—
ROL	Rotate Left		IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	0C 1C 2C 170C 171C 172C 173C	ff ff ff gggg gggg gggg hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
ROLA	Rotate Left A		INH	370C	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLB	Rotate Left B		INH	371C	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLD	Rotate Left D		INH	27FC	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLE	Rotate Left E		INH	277C	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLW	Rotate Left Word		IND16, X IND16, Y IND16, Z EXT	270C 271C 272C 273C	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
ROR	Rotate Right Byte		IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	0E 1E 2E 170E 171E 172E 173E	ff ff ff gggg gggg gggg hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
RORA	Rotate Right A		INH	370E	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORB	Rotate Right B		INH	371E	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORD	Rotate Right D		INH	27FE	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORE	Rotate Right E		INH	277E	—	2	—	—	—	—	Δ	Δ	Δ	Δ

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
RORW	Rotate Right Word		IND16, X IND16, Y IND16, Z EXT	270E 271E 272E 273E	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
RTI[3]	Return from Interrupt	(SK : SP) + 2 ⇒ SK : SP Pull CCR (SK : SP) + 2 ⇒ SK : SP Pull PC (PK : PC) – 6 ⇒ PK : PC	INH	2777	—	12	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
RTS[4]	Return from Subroutine	(SK : SP) + 2 ⇒ SK : SP Pull PK (SK : SP) + 2 ⇒ SK : SP Pull PC (PK : PC) – 2 ⇒ PK : PC	INH	27F7	—	12	—	—	—	—	—	—	—	—
SBA	Subtract B from A	(A) – (B) ⇒ A	INH	370A	—	2	—	—	—	—	Δ	Δ	Δ	Δ
SBCA	Subtract with Carry from A	(A) – (M) – C ⇒ A	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	42 52 62 72 1742 1752 1762 1772 2742 2752 2762	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ
SBCB	Subtract with Carry from B	(B) – (M) – C ⇒ B	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	C2 D2 E2 F2 17C2 17D2 17E2 17F2 27C2 27D2 27E2	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ
SBCD	Subtract with Carry from D	(D) – (M : M + 1) – C ⇒ D	IND8, X IND8, Y IND8, Z IMM16 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	82 92 A2 37B2 37C2 37D2 37E2 37F2 2782 2792 27A2	ff ff ff jj kk gggg gggg gggg hh ll — — —	6 6 6 4 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ
SBCE	Subtract with Carry from E	(E) – (M : M + 1) – C ⇒ E	IMM16 IND16, X IND16, Y IND16, Z EXT	3732 3742 3752 3762 3772	jj kk gggg gggg gggg hh ll	4 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ
SDE	Subtract D from E	(E) – (D)⇒ E	INH	2779	—	2	—	—	—	—	Δ	Δ	Δ	Δ
STAA	Store A	(A) ⇒ M	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	4A 5A 6A 174A 175A 176A 177A 274A 275A 276A	ff ff ff gggg gggg gggg hh ll — — —	4 4 4 6 6 6 6 4 4 4	—	—	—	—	Δ	Δ	0	—

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
STAB	Store B	(B) ⇒ M	IND8, X	CA	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	DA	ff	4								
			IND8, Z	EA	ff	4								
			IND16, X	17CA	gggg	6								
			IND16, Y	17DA	gggg	6								
			IND16, Z	17EA	gggg	6								
			EXT	17FA	hh ll	6								
			E, X	27CA	—	4								
			E, Y	27DA	—	4								
			E, Z	27EA	—	4								
STD	Store D	(D) ⇒ M : M + 1	IND8, X	8A	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	9A	ff	4								
			IND8, Z	AA	ff	4								
			IND16, X	37CA	gggg	6								
			IND16, Y	37DA	gggg	6								
			IND16, Z	37EA	gggg	6								
			EXT	37FA	hh ll	6								
			E, X	278A	—	6								
			E, Y	279A	—	6								
			E, Z	27AA	—	6								
STE	Store E	(E) ⇒ M : M + 1	IND16, X	374A	gggg	6	—	—	—	—	Δ	Δ	0	—
			IND16, Y	375A	gggg	6								
			IND16, Z	376A	gggg	6								
			EXT	377A	hh ll	6								
STED	Store Concatenated D and E	(E) ⇒ M : M + 1 (D) ⇒ M + 2 : M + 3	EXT	2773	hh ll	8	—	—	—	—	—	—	—	—
STS	Store Stack Pointer	(SP) ⇒ M : M + 1	IND8, X	8F	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	9F	ff	4								
			IND8, Z	AF	ff	4								
			IND16, X	178F	gggg	6								
			IND16, Y	179F	gggg	6								
			IND16, Z	17AF	gggg	6								
			EXT	17BF	hh ll	6								
STX	Store IX	(IX) ⇒ M : M + 1	IND8, X	8C	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	9C	ff	4								
			IND8, Z	AC	ff	4								
			IND16, X	178C	gggg	6								
			IND16, Y	179C	gggg	6								
			IND16, Z	17AC	gggg	6								
			EXT	17BC	hh ll	6								
STY	Store IY	(IY) ⇒ M : M + 1	IND8, X	8D	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	9D	ff	4								
			IND8, Z	AD	ff	4								
			IND16, X	178D	gggg	6								
			IND16, Y	179D	gggg	6								
			IND16, Z	17AD	gggg	6								
			EXT	17BD	hh ll	6								
STZ	Store Z	(IZ) ⇒ M : M + 1	IND8, X	8E	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	9E	ff	4								
			IND8, Z	AE	ff	4								
			IND16, X	178E	gggg	6								
			IND16, Y	179E	gggg	6								
			IND16, Z	17AE	gggg	6								
			EXT	17BE	hh ll	6								
SUBA	Subtract from A	(A) – (M) ⇒ A	IND8, X	40	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	50	ff	6								
			IND8, Z	60	ff	6								
			IMM8	70	ii	2								
			IND16, X	1740	gggg	6								
			IND16, Y	1750	gggg	6								
			IND16, Z	1760	gggg	6								
			EXT	1770	hh ll	6								
			E, X	2740	—	6								
			E, Y	2750	—	6								
			E, Z	2760	—	6								

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
SUBB	Subtract from B	(B) – (M) ⇒ B	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	C0 D0 E0 F0 17C0 17D0 17E0 17F0 27C0 27D0 27E0	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ
SUBD	Subtract from D	(D) – (M : M + 1) ⇒ D	IND8, X IND8, Y IND8, Z IMM16 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	80 90 A0 37B0 37C0 37D0 37E0 37F0 2780 2790 27A0	ff ff ff jj kk gggg gggg gggg hh ll — — —	6 6 6 4 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ
SUBE	Subtract from E	(E) – (M : M + 1) ⇒ E	IMM16 IND16, X IND16, Y IND16, Z EXT	3730 3740 3750 3760 3770	jj kk gggg gggg gggg hh ll	4 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ
SWI	Software Interrupt	(PK : PC) + $0002 ⇒ PK : PC Push (PC) (SK : SP) – $0002 ⇒ SK : SP Push (CCR) (SK : SP) – $0002 ⇒ SK : SP $0 ⇒ PK SWI Vector ⇒ PC	INH	3720	—	16	—	—	—	—	—	—	—	—
SXT	Sign Extend B into A	If B7 = 1 then $FF ⇒ A else $00 ⇒ A	INH	27F8	—	2	—	—	—	—	Δ	Δ	—	—
TAB	Transfer A to B	(A) ⇒ B	INH	3717	—	2	—	—	—	—	Δ	Δ	0	—
TAP	Transfer A to CCR	(A[7:0]) ⇒ CCR[15:8]	INH	37FD	—	4	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
TBA	Transfer B to A	(B) ⇒ A	INH	3707	—	2	—	—	—	—	Δ	Δ	0	—
TBEK	Transfer B to EK	(B[3:0]) ⇒ EK	INH	27FA	—	2	—	—	—	—	—	—	—	—
TBSK	Transfer B to SK	(B[3:0]) ⇒ SK	INH	379F	—	2	—	—	—	—	—	—	—	—
TBXK	Transfer B to XK	(B[3:0]) ⇒ XK	INH	379C	—	2	—	—	—	—	—	—	—	—
TBYK	Transfer B to YK	(B[3:0]) ⇒ YK	INH	379D	—	2	—	—	—	—	—	—	—	—
TBZK	Transfer B to ZK	(B[3:0]) ⇒ ZK	INH	379E	—	2	—	—	—	—	—	—	—	—
TDE	Transfer D to E	(D) ⇒ E	INH	277B	—	2	—	—	—	—	Δ	Δ	0	—
TDMSK	Transfer D to XMSK : YMSK	(D[15:8]) ⇒ X MASK (D[7:0]) ⇒ Y MASK	INH	372F	—	2	—	—	—	—	—	—	—	—
TDP[1]	Transfer D to CCR	(D) ⇒ CCR[15:4]	INH	372D	—	4	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
TED	Transfer E to D	(E) ⇒ D	INH	27FB	—	2	—	—	—	—	Δ	Δ	0	—
TEDM	Transfer E and D to AM[31:0] Sign Extend AM	(E) ⇒ AM[31:16] (D) ⇒ AM[15:0] AM[35:32] = AM31	INH	27B1	—	4	—	0	—	0	—	—	—	—
TEKB	Transfer EK to B	(EK) ⇒ B[3:0] $0 ⇒ B[7:4]	INH	27BB	—	2	—	—	—	—	—	—	—	—
TEM	Transfer E to AM[31:16] Sign Extend AM Clear AM LSB	(E) ⇒ AM[31:16] $00 ⇒ AM[15:0] AM[35:32] = AM31	INH	27B2	—	4	—	0	—	0	—	—	—	—
TMER	Transfer Rounded AM to E	Rounded (AM) ⇒ Temp If (SM • (EV ✣ MV)) then Saturation Value ⇒ E else Temp[31:16] ⇒ E	INH	27B4	—	6	—	Δ	—	Δ	Δ	Δ	—	—

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
TMET	Transfer Truncated AM to E	If (SM • (EV ✣ MV)) then Saturation Value ⇒ E else AM[31:16] ⇒ E	INH	27B5	—	2	—	—	—	—	Δ	Δ	—	—
TMXED	Transfer AM to IX : E : D	AM[35:32] ⇒ IX[3:0] AM35 ⇒ IX[15:4] AM[31:16] ⇒ E AM[15:0] ⇒ D	INH	27B3	—	6	—	—	—	—	—	—	—	—
TPA	Transfer CCR to A	(CCR[15:8]) ⇒ A	INH	37FC	—	2	—	—	—	—	—	—	—	—
TPD	Transfer CCR to D	(CCR) ⇒ D	INH	372C	—	2	—	—	—	—	—	—	—	—
TSKB	Transfer SK to B	(SK) ⇒ B[3:0] $0 ⇒ B[7:4]	INH	37AF	—	2	—	—	—	—	—	—	—	—
TST	Test Byte Zero or Minus	(M) – $00	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	06 16 26 1706 1716 1726 1736	ff ff ff gggg gggg gggg hh ll	6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	0
TSTA	Test A for Zero or Minus	(A) – $00	INH	3706	—	2	—	—	—	—	Δ	Δ	0	0
TSTB	Test B for Zero or Minus	(B) – $00	INH	3716	—	2	—	—	—	—	Δ	Δ	0	0
TSTD	Test D for Zero or Minus	(D) – $0000	INH	27F6	—	2	—	—	—	—	Δ	Δ	0	0
TSTE	Test E for Zero or Minus	(E) – $0000	INH	2776	—	2	—	—	—	—	Δ	Δ	0	0
TSTW	Test for Zero or Minus Word	(M : M + 1) – $0000	IND16, X IND16, Y IND16, Z EXT	2706 2716 2726 2736	gggg gggg gggg hh ll	6 6 6 6	—	—	—	—	Δ	Δ	0	0
TSX	Transfer SP to X	(SK : SP) + $0002 ⇒ XK : IX	INH	274F	—	2	—	—	—	—	—	—	—	—
TSY	Transfer SP to Y	(SK : SP) + $0002 ⇒ YK : IY	INH	275F	—	2	—	—	—	—	—	—	—	—
TSZ	Transfer SP to Z	(SK : SP) + $0002 ⇒ ZK : IZ	INH	276F	—	2	—	—	—	—	—	—	—	—
TXKB	Transfer XK to B	(XK) ⇒ B[3:0] $0 ⇒ B[7:4]	INH	37AC	—	2	—	—	—	—	—	—	—	—
TXS	Transfer X to SP	(XK : IX) – $0002 ⇒ SK : SP	INH	374E	—	2	—	—	—	—	—	—	—	—
TXY	Transfer X to Y	(XK : IX) ⇒ YK : IY	INH	275C	—	2	—	—	—	—	—	—	—	—
TXZ	Transfer X to Z	(XK : IX) ⇒ ZK : IZ	INH	276C	—	2	—	—	—	—	—	—	—	—
TYKB	Transfer YK to B	(YK) ⇒ B[3:0] $0 ⇒ B[7:4]	INH	37AD	—	2	—	—	—	—	—	—	—	—
TYS	Transfer Y to SP	(YK : IY) – $0002 ⇒ SK : SP	INH	375E	—	2	—	—	—	—	—	—	—	—
TYX	Transfer Y to X	(YK : IY) ⇒ XK : IX	INH	274D	—	2	—	—	—	—	—	—	—	—
TYZ	Transfer Y to Z	(YK : IY) ⇒ ZK : IZ	INH	276D	—	2	—	—	—	—	—	—	—	—
TZKB	Transfer ZK to B	(ZK) ⇒ B[3:0] $0 ⇒ B[7:4]	INH	37AE	—	2	—	—	—	—	—	—	—	—
TZS	Transfer Z to SP	(ZK : IZ) – $0002 ⇒ SK : SP	INH	376E	—	2	—	—	—	—	—	—	—	—
TZX	Transfer Z to X	(ZK : IZ) ⇒ XK : IX	INH	274E	—	2	—	—	—	—	—	—	—	—
TZY	Transfer Z to Y	(ZK : IZ) ⇒ YK : IY	INH	275E	—	2	—	—	—	—	—	—	—	—
WAI	Wait for Interrupt	WAIT	INH	27F3	—	8	—	—	—	—	—	—	—	—
XGAB	Exchange A with B	(A) ⇔ (B)	INH	371A	—	2	—	—	—	—	—	—	—	—
XGDE	Exchange D with E	(D) ⇔ (E)	INH	277A	—	2	—	—	—	—	—	—	—	—
XGDX	Exchange D with IX	(D) ⇔ (IX)	INH	37CC	—	2	—	—	—	—	—	—	—	—

Freescale Semiconductor, Inc.

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

Table 4-2 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
XGDY	Exchange D with IY	(D) ⇔ (IY)	INH	37DC	—	2	—	—	—	—	—	—	—	—
XGDZ	Exchange D with IZ	(D) ⇔ (IZ)	INH	37EC	—	2	—	—	—	—	—	—	—	—
XGEX	Exchange E with IX	(E) ⇔ (IX)	INH	374C	—	2	—	—	—	—	—	—	—	—
XGEY	Exchange E with IY	(E) ⇔ (IY)	INH	375C	—	2	—	—	—	—	—	—	—	—
XGEZ	Exchange E with IZ	(E) ⇔ (IZ)	INH	376C	—	2	—	—	—	—	—	—	—	—

NOTES:

1. CCR[15:4] change according to the results of the operation. The PK field is not affected.
2. Cycle times for conditional branches are shown in “taken, not taken” order.
3. CCR[15:0] change according to the copy of the CCR pulled from the stack.
4. PK field changes according to the state pulled from the stack. The rest of the CCR is not affected.

Freescale Semiconductor, Inc.

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

Table 4-3 Instruction Set Abbreviations and Symbols

A	Accumulator A	X	Register used in operation
AM	Accumulator M	M	Address of one memory byte
B	Accumulator B	M +1	Address of byte at M + $0001
CCR	Condition code register	M : M + 1	Address of one memory word
D	Accumulator D	(…)X	Contents of address pointed to by IX
E	Accumulator E	(...)Y	Contents of address pointed to by IY
EK	Extended addressing extension field	(...)Z	Contents of address pointed to by IZ
IR	MAC multiplicand register	E, X	IX with E offset
HR	MAC multiplier register	E, Y	IY with E offset
IX	Index register X	E, Z	IZ with E offset
IY	Index register Y	EXT	Extended
IZ	Index register Z	EXT20	20-bit extended
K	Address extension register	IMM8	8-bit immediate
PC	Program counter	IMM16	16-bit immediate
PK	Program counter extension field	IND8, X	IX with unsigned 8-bit offset
SK	Stack pointer extension field	IND8, Y	IY with unsigned 8-bit offset
SL	Multiply and accumulate sign latch	IND8, Z	IZ with unsigned 8-bit offset
SP	Stack pointer	IND16, X	IX with signed 16-bit offset
XK	Index register X extension field	IND16, Y	IY with signed 16-bit offset
YK	Index register Y extension field	IND16, Z	IZ with signed 16-bit offset
ZK	Index register Z extension field	IND20, X	IX with signed 20-bit offset
XMSK	Modulo addressing index register X mask	IND20, Y	IY with signed 20-bit offset
YMSK	Modulo addressing index register Y mask	IND20, Z	IZ with signed 20-bit offset
S	Stop disable control bit	INH	Inherent
MV	AM overflow indicator	IXP	Post-modified indexed
H	Half carry indicator	REL8	8-bit relative
EV	AM extended overflow indicator	REL16	16-bit relative
N	Negative indicator	b	4-bit address extension
Z	Zero indicator	ff	8-bit unsigned offset
V	Two’s complement overflow indicator	gggg	16-bit signed offset
C	Carry/borrow indicator	hh	High byte of 16-bit extended address
IP	Interrupt priority field	ii	8-bit immediate data
SM	Saturation mode control bit	jj	High byte of 16-bit immediate data
PK	Program counter extension field	kk	Low byte of 16-bit immediate data
—	Bit not affected	ll	Low byte of 16-bit extended address
Δ	Bit changes as specified	mm	8-bit mask
0	Bit cleared	mmmm	16-bit mask
1	Bit set	rr	8-bit unsigned relative offset
M	Memory location used in operation	rrrr	16-bit signed relative offset
R	Result of operation	xo	MAC index register X offset
S	Source data	yo	MAC index register Y offset
		z	4-bit zero extension

+	Addition	•	AND
−	Subtraction or negation (two’s complement)	✣	Inclusive OR (OR)
*	Multiplication	⊕	Exclusive OR (EOR)
/	Division	$\overline{\text{NOT}}$	Complementation
>	Greater	:	Concatenation
<	Less	⇒	Transferred
=	Equal	⇔	Exchanged
≥	Equal or greater	±	Sign bit; also used to show tolerance
≤	Equal or less	«	Sign extension
≠	Not equal	%	Binary value
		$	Hexadecimal value

Freescale Semiconductor, Inc.

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

4.8 Comparison of CPU16 and M68HC11 CPU Instruction Sets

Most M68HC11 CPU instructions are a source-code compatible subset of the CPU16 instruction set. However, certain M68HC11 CPU instructions have been replaced by functionally equivalent CPU16 instructions, and some CPU16 instructions with the same mnemonics as M68HC11 CPU instructions operate differently.

Table 4-4 shows the M68HC11 CPU instructions that either have been replaced by CPU16 instructions or that operate differently on the CPU16. Replacement instructions are not identical to M68HC11 CPU instructions. M68HC11 code must be altered to establish proper preconditions.

All CPU16 instruction execution times differ from those of the M68HC11. *Transporting M68HC11 Code to M68HC16 Devices*, (M68HC16PN01/D), contains detailed information about differences between the two instruction sets. Refer to the *CPU16 Reference Manual* (CPU16RM/AD) for further details about CPU operations.

Freescale Semiconductor, Inc.

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

Table 4-4 CPU16 Implementation of M68HC11 CPU Instructions

M68HC11 Instruction	CPU16 Implementation
BHS	BCC only
BLO	BCS only
BSR	Generates a different stack frame
CLC	Replaced by ANDP
CLI	Replaced by ANDP
CLV	Replaced by ANDP
DES	Replaced by AIS
DEX	Replaced by AIX
DEY	Replaced by AIY
INS	Replaced by AIS
INX	Replaced by AIX
INY	Replaced by AIY
JMP	IND8 and EXT addressing modes replaced by IND20 and EXT20 modes
JSR	IND8 and EXT addressing modes replaced by IND20 and EXT20 modes. Generates a different stack frame
LSL, LSLD	Use ASL instructions[1]
PSHX	Replaced by PSHM
PSHY	Replaced by PSHM
PULX	Replaced by PULM
PULY	Replaced by PULM
RTI	Reloads PC and CCR only
RTS	Uses two-word stack frame
SEC	Replaced by ORP
SEI	Replaced by ORP
SEV	Replaced by ORP
STOP	Replaced by LPSTOP
TAP	CPU16 CCR bits differ from M68HC11 CPU16 interrupt priority scheme differs from M68HC11
TPA	CPU16 CCR bits differ from M68HC11 CPU16 interrupt priority scheme differs from M68HC11
TSX	Adds two to SK : SP before transfer to XK : IX
TSY	Adds two to SK : SP before transfer to YK : IY
TXS	Subtracts two from XK : IX before transfer to SK : SP
TXY	Transfers XK field to YK field
TYS	Subtracts two from YK : IY before transfer to SK : SP
TYX	Transfers YK field to XK field
WAI	Waits indefinitely for interrupt or reset Generates a different stack frame

NOTES:
1. Freescale assemblers automatically translate ASL mnemonics.

Freescale Semiconductor, Inc.

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

4.9 Instruction Format

CPU16 instructions consist of an 8-bit opcode that can be preceded by an 8-bit prebyte and followed by one or more operands.

Opcodes are mapped in four 256-instruction pages. Page 0 opcodes stand alone. Page 1, 2, and 3 opcodes are pointed to by a prebyte code on page 0. The prebytes are $17 (page 1), $27 (page 2), and $37 (page 3).

Operands can be four bits, eight bits or sixteen bits in length. Since the CPU16 fetches 16-bit instruction words from even-byte boundaries, each instruction must contain an even number of bytes.

Operands are organized as bytes, words, or a combination of bytes and words. Operands of four bits are either zero-extended to eight bits, or packed two to a byte. The largest instructions are six bytes in length. Size, order, and function of operands are evaluated when an instruction is decoded.

A page 0 opcode and an 8-bit operand can be fetched simultaneously. Instructions that use 8-bit indexed, immediate, and relative addressing modes have this form. Code written with these instructions is very compact.

Figure 4-4 shows basic CPU16 instruction formats.

Freescale Semiconductor, Inc.

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

8-Bit Opcode with 8-Bit Operand

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

8-Bit Opcode with 4-Bit Index Extensions

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Opcode								X Extension				Y Extension			

8-Bit Opcode, Argument(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Opcode								Operand							
Operand(s)															
Operand(s)															

8-Bit Opcode with 8-Bit Prebyte, No Argument

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

8-Bit Opcode with 8-Bit Prebyte, Argument(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Prebyte								Opcode							
Operand(s)															
Operand(s)															

8-Bit Opcode with 20-Bit Argument

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Opcode								$0				Extension			
Operand															

Figure 4-4 Basic Instruction Formats

4.10 Execution Model

This description builds up a conceptual model of the mechanism the CPU16 uses to fetch and execute instructions. The functional divisions in the model do not necessarily correspond to physical subunits of the microprocessor.

As shown in **Figure 4-5**, there are three functional blocks involved in fetching, decoding, and executing instructions. These are the microsequencer, the instruction pipeline, and the execution unit. These elements function concurrently. All three may be active at any given time.

Freescale Semiconductor, Inc.

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

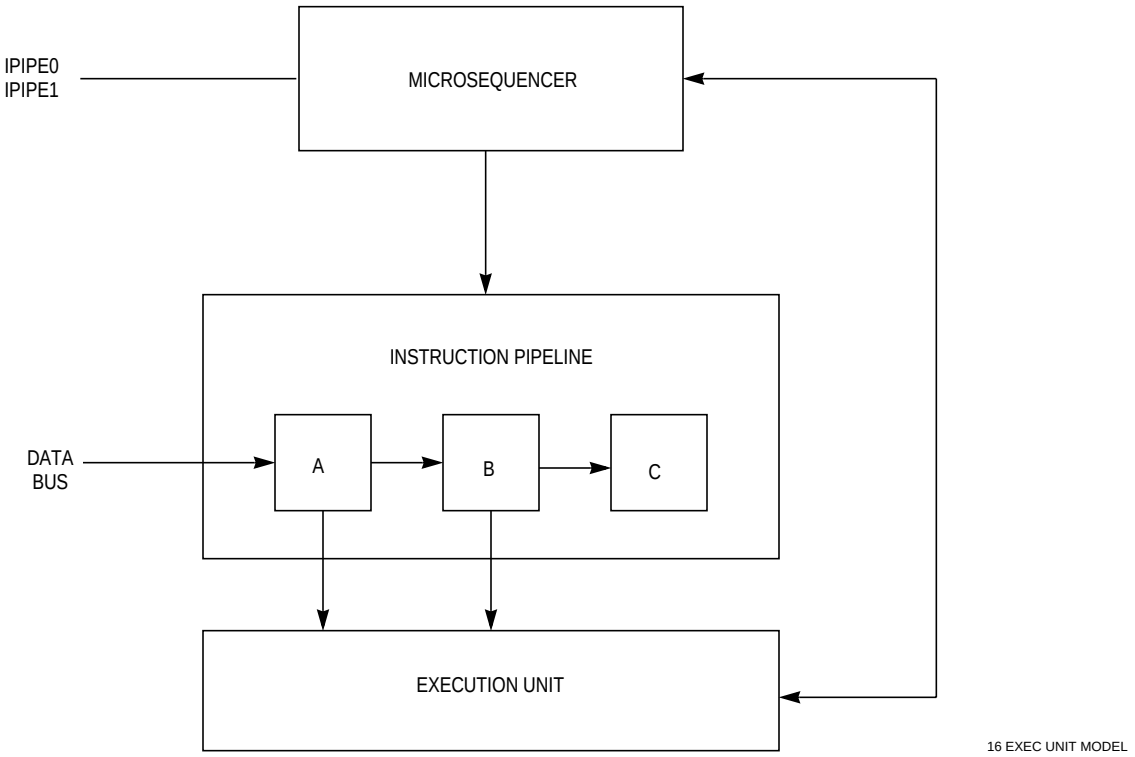


Figure 4-5 Instruction Execution Model

4.10.1 Microsequencer

The microsequencer controls the order in which instructions are fetched, advanced through the pipeline, and executed. It increments the program counter and generates multiplexed external tracking signals IPIPE0 and IPIPE1 from internal signals that control execution sequence.

4.10.2 Instruction Pipeline

The pipeline is a three stage FIFO that holds instructions while they are decoded and executed. Depending upon instruction size, as many as three instructions can be in the pipeline at one time (single-word instructions, one held in stage C, one being executed in stage B, and one latched in stage A).

4.10.3 Execution Unit

The execution unit evaluates opcodes, interfaces with the microsequencer to advance instructions through the pipeline, and performs instruction operations.

Freescale Semiconductor, Inc.

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

4.11 Execution Process

Fetched opcodes are latched into stage A, then advanced to stage B. Opcodes are evaluated in stage B. The execution unit can access operands in either stage A or stage B (stage B accesses are limited to 8-bit operands). When execution is complete, opcodes are moved from stage B to stage C, where they remain until the next instruction is complete.

A prefetch mechanism in the microsequencer reads instruction words from memory and increments the program counter. When instruction execution begins, the program counter points to an address six bytes after the address of the first word of the instruction being executed.

The number of machine cycles necessary to complete an execution sequence varies according to the complexity of the instruction. Refer to the *CPU16 Reference Manual* (CPU16RM/AD) for details.

4.11.1 Changes in Program Flow

When program flow changes, instructions are fetched from a new address. Before execution can begin at the new address, instructions and operands from the previous instruction stream must be removed from the pipeline. If a change in flow is temporary, a return address must be stored, so that execution of the original instruction stream can resume after the change in flow.

When an instruction that causes a change in program flow executes, PK : PC point to the address of the first word of the instruction + $0006. During execution of the instruction, PK : PC is loaded with the address of the first instruction word in the new instruction stream. However, stages A and B still contain words from the old instruction stream. Extra processing steps must be performed before execution from the new instruction stream.

4.12 Instruction Timing

The execution time of CPU16 instructions has three components:

- Bus cycles required to prefetch the next instruction
- Bus cycles required for operand accesses
- Time required for internal operations

A bus cycle requires a minimum of two system clock periods. If the access time of a memory device is greater than two clock periods, bus cycles are longer. However, all bus cycles must be an integer number of clock periods. CPU16 internal operations are always an integer multiple of two clock periods.

Dynamic bus sizing affects bus cycle time. The integration module manages all accesses. Refer to **SECTION 5 SYSTEM INTEGRATION MODULE** for more information.

The CPU16 does not execute more than one instruction at a time. The total time required to execute a particular instruction stream can be calculated by summing the individual execution times of each instruction in the stream.

Freescale Semiconductor, Inc.

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

Total execution time is calculated using the expression:

$$(CL_T) = (CL_P) + (CL_O) + (CL_I)$$

Where:

(CL_T) = Total clock periods per instruction

(CL_I) = Clock periods used for internal operation

(CL_P) = Clock periods used for program access

(CL_O) = Clock periods used for operand access

Refer to the *CPU16 Reference Manual* (CPU16RM/AD) for more information on this topic.

4.13 Exceptions

An exception is an event that preempts normal instruction processing. Exception processing makes the transition from normal instruction execution to execution of a routine that deals with the exception.

Each exception has an assigned vector that points to an associated handler routine. Exception processing includes all operations required to transfer control to a handler routine, but does not include execution of the handler routine itself. Keep the distinction between exception processing and execution of an exception handler in mind while reading this section.

4.13.1 Exception Vectors

An exception vector is the address of a routine that handles an exception. Exception vectors are contained in a data structure called the exception vector table, which is located in the first 512 bytes of bank 0. Refer to **Table 4-5** for the exception vector table.

All vectors except the reset vector consist of one word and reside in data space. The reset vector consists of four words that reside in program space. Refer to **SECTION 5 SYSTEM INTEGRATION MODULE** for information concerning address space types and the function code outputs. There are 52 predefined or reserved vectors, and 200 user-defined vectors.

Each vector is assigned an 8-bit number. Vector numbers for some exceptions are generated by external devices; others are supplied by the processor. There is a direct mapping of vector number to vector table address. The processor left shifts the vector number one place (multiplies by two) to convert it to an address.

Freescale Semiconductor, Inc.

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

Table 4-5 Exception Vector Table

Vector Number	Vector Address	Address Space	Type of Exception
0	0000	P	Reset — Initial ZK, SK, and PK
	0002	P	Reset — Initial PC
	0004	P	Reset — Initial SP
	0006	P	Reset — Initial IZ (Direct Page)
4	0008	D	Breakpoint
5	000A	D	Bus Error
6	000C	D	Software Interrupt
7	000E	D	Illegal Instruction
8	0010	D	Division by Zero
9 – E	0012 – 001C	D	Unassigned, Reserved
F	001E	D	Uninitialized Interrupt
10	0020	D	Unassigned, Reserved
11	0022	D	Level 1 Interrupt Autovector
12	0024	D	Level 2 Interrupt Autovector
13	0026	D	Level 3 Interrupt Autovector
14	0028	D	Level 4 Interrupt Autovector
15	002A	D	Level 5 Interrupt Autovector
16	002C	D	Level 6 Interrupt Autovector
17	002E	D	Level 7 Interrupt Autovector
18	0030	D	Spurious Interrupt
19 – 37	0032 – 006E	D	Unassigned, Reserved
38 – FF	0070 – 01FE	D	User-Defined Interrupts

4.13.2 Exception Stack Frame

During exception processing, the contents of the program counter and condition code register are stacked at a location pointed to by SK : SP. Unless it is altered during exception processing, the stacked PK : PC value is the address of the next instruction in the current instruction stream, plus $0006. **Figure 4-6** shows the exception stack frame.

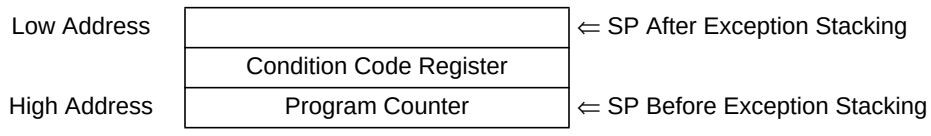


Figure 4-6 Exception Stack Frame Format

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

4.13.3 Exception Processing Sequence

Exception processing is performed in four phases. Priority of all pending exceptions is evaluated and the highest priority exception is processed first. Processor state is stacked, then the CCR PK extension field is cleared. An exception vector number is acquired and converted to a vector address. The content of the vector address is loaded into the PC and the processor jumps to the exception handler routine.

There are variations within each phase for differing types of exceptions. However, all vectors except RESET are 16-bit addresses, and the PK field is cleared during exception processing. Consequently, exception handlers must be located within bank 0 or vectors must point to a jump table in bank 0.

4.13.4 Types of Exceptions

Exceptions can be either internally or externally generated. External exceptions, which are defined as asynchronous, include interrupts, bus errors, breakpoints, and resets. Internal exceptions, which are defined as synchronous, include the software interrupt (SWI) instruction, the background (BGND) instruction, illegal instruction exceptions, and the divide-by-zero exception.

4.13.4.1 Asynchronous Exceptions

Asynchronous exceptions occur without reference to CPU16 or IMB clocks, but exception processing is synchronized. For all asynchronous exceptions except RESET, exception processing begins at the first instruction boundary following recognition of an exception. Refer to **5.8.1 Interrupt Exception Processing** for more information concerning asynchronous exceptions.

Because of pipelining, the stacked return PK : PC value for all asynchronous exceptions, other than reset, is equal to the address of the next instruction in the current instruction stream plus $0006. The RTI instruction, which must terminate all exception handler routines, subtracts $0006 from the stacked value to resume execution of the interrupted instruction stream.

4.13.4.2 Synchronous Exceptions

Synchronous exception processing is part of an instruction definition. Exception processing for synchronous exceptions is always completed, and the first instruction of the handler routine is always executed, before interrupts are detected.

Because of pipelining, the value of PK : PC at the time a synchronous exception executes is equal to the address of the instruction that causes the exception plus $0006. Because RTI always subtracts $0006 upon return, the stacked PK : PC must be adjusted by the instruction that caused the exception so that execution resumes with the following instruction. For this reason, $0002 is added to the PK : PC value before it is stacked.

Freescale Semiconductor, Inc.

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

4.13.5 Multiple Exceptions

Each exception has a hardware priority based upon its relative importance to system operation. Asynchronous exceptions have higher priorities than synchronous exceptions. Exception processing for multiple exceptions is completed by priority, from highest to lowest. Priority governs the order in which exception processing occurs, not the order in which exception handlers are executed.

Unless a bus error, a breakpoint, or a reset occurs during exception processing, the first instruction of all exception handler routines is guaranteed to execute before another exception is processed. Because interrupt exceptions have higher priority than synchronous exceptions, the first instruction in an interrupt handler is executed before other interrupts are sensed.

Bus error, breakpoint, and reset exceptions that occur during exception processing of a previous exception are processed before the first instruction of that exception's handler routine. The converse is not true. If an interrupt occurs during bus error exception processing, for example, the first instruction of the exception handler is executed before interrupts are sensed. This permits the exception handler to mask interrupts during execution.

Refer to **SECTION 5 SYSTEM INTEGRATION MODULE** for detailed information concerning interrupts and system reset. For information concerning processing of specific exceptions, refer to the *CPU16 Reference Manual* (CPU16RM/AD).

4.13.6 RTI Instruction

The return-from-interrupt instruction (RTI) must be the last instruction in all exception handlers except the RESET handler. RTI pulls the exception stack frame that was pushed onto the system stack during exception processing, and restores processor state. Normal program flow resumes at the address of the instruction that follows the last instruction executed before exception processing began.

RTI is not used in the RESET handler because RESET initializes the stack pointer and does not create a stack frame.

4.14 Development Support

The CPU16 incorporates powerful tools for tracking program execution and for system debugging. These tools are deterministic opcode tracking, breakpoint exceptions, and background debug mode. Judicious use of CPU16 capabilities permits in-circuit emulation and system debugging using a bus state analyzer, a simple serial interface, and a terminal.

4.14.1 Deterministic Opcode Tracking

The CPU16 has two multiplexed outputs, IPIPE0 and IPIPE1, that enable external hardware to monitor the instruction pipeline during normal program execution. The signals IPIPE0 and IPIPE1 can be demultiplexed into six pipeline state signals that allow a state analyzer to synchronize with instruction stream activity.

Freescale Semiconductor, Inc.

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

4.14.1.1 IPIPE0/IPIPE1 Multiplexing

Six types of information are required to track pipeline activity. To generate the six state signals, eight pipeline states are encoded and multiplexed into IPIPE0 and IPIPE1. The multiplexed signals have two phases. State signals are active low. **Table 4-6** shows the encoding scheme.

Table 4-6 IPIPE0/IPIPE1 Encoding

Phase	IPIPE1 State	IPIPE0 State	State Signal Name
1	0 0 1 1	0 1 0 1	START and FETCH FETCH START NULL
2	0 0 1 1	0 1 0 1	INVALID ADVANCE EXCEPTION NULL

IPIPE0 and IPIPE1 are timed so that a logic analyzer can capture all six pipeline state signals and address, data, or control bus state in any single bus cycle. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for specifications.

State signals can be latched asynchronously on the falling and rising edges of either address strobe ($\overline{AS}$) or data strobe ($\overline{DS}$). They can also be latched synchronously using the microcontroller CLKOUT signal. Refer to the *CPU16 Reference Manual* (CPU16RM/AD) for more information on the CLKOUT signal, state signals, and state signal demux logic.

4.14.1.2 Combining Opcode Tracking with Other Capabilities

Pipeline state signals are useful during normal instruction execution and execution of exception handlers. The signals provide a complete model of the pipeline up to the point a breakpoint is acknowledged.

Breakpoints are acknowledged after an instruction has executed, when it is in pipeline stage C. A breakpoint can initiate either exception processing or background debug mode. IPIPE0/IPIPE1 are not usable when the CPU16 is in background debug mode.

4.14.2 Breakpoints

Breakpoints are set by assertion of the microcontroller $\overline{BKPT}$ pin. The CPU16 supports breakpoints on any memory access. Acknowledged breakpoints can initiate either exception processing or background debug mode. After BDM has been enabled, the CPU16 will enter BDM when the $\overline{BKPT}$ input is asserted.

- If $\overline{BKPT}$ assertion is synchronized with an instruction prefetch, the instruction is tagged with the breakpoint when it enters the pipeline, and the breakpoint occurs after the instruction executes.
- If $\overline{BKPT}$ assertion is synchronized with an operand fetch, breakpoint processing occurs at the end of the instruction during which $\overline{BKPT}$ is latched.

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

Breakpoints on instructions that are flushed from the pipeline before execution are not acknowledged. Operand breakpoints are always acknowledged. There is no breakpoint acknowledge bus cycle when BDM is entered. Refer to **5.6.4.1 Breakpoint Acknowledge Cycle** for more information about breakpoints.

4.14.3 Opcode Tracking and Breakpoints

Breakpoints are acknowledged after a tagged instruction has executed, that is, when the instruction is copied from pipeline stage B to stage C. Stage C contains the opcode of the previous instruction when execution of the current instruction begins.

When an instruction is tagged, IPIPE0/IPIPE1 reflect the start of execution and the appropriate number of pipeline advances and operand fetches before the breakpoint is acknowledged. If background debug mode is enabled, these signals model the pipeline before BDM is entered.

4.14.4 Background Debug Mode

Microprocessor debugging programs are generally implemented in external software. CPU16 BDM provides a debugger implemented in CPU microcode. BDM incorporates a full set of debug options. Registers can be viewed and altered, memory can be read or written, and test features can be invoked. BDM is an alternate CPU16 operating mode. While the CPU16 is in BDM, normal instruction execution is suspended, and special microcode performs debugging functions under external control. While in BDM, the CPU16 ceases to fetch instructions through the data bus and communicates with the development system through a dedicated serial interface.

4.14.4.1 Enabling BDM

The CPU16 samples the $\overline{\text{BKPT}}$ input during reset to determine whether to enable BDM. When $\overline{\text{BKPT}}$ is asserted at the rising edge of the $\overline{\text{RESET}}$ signal, BDM operation is enabled. BDM remains enabled until the next system reset. If $\overline{\text{BKPT}}$ is at logic level one on the trailing edge of $\overline{\text{RESET}}$, BDM is disabled. $\overline{\text{BKPT}}$ is relatched on each rising transition of $\overline{\text{RESET}}$. $\overline{\text{BKPT}}$ is synchronized internally and must be asserted for at least two clock cycles before negation of $\overline{\text{RESET}}$.

4.14.4.2 BDM Sources

When BDM is enabled, external breakpoint hardware and the BGND instruction can cause the CPU16 to enter BDM. If BDM is not enabled when a breakpoint occurs, a breakpoint exception is processed.

4.14.4.3 Entering BDM

When the CPU16 detects a breakpoint or decodes a BGND instruction when BDM is enabled, it suspends instruction execution and asserts the FREEZE signal. Once FREEZE has been asserted, the CPU16 enables the BDM serial communication hardware and awaits a command. Assertion of FREEZE causes opcode tracking signals IPIPE0 and IPIPE1 to change definition and become serial communication signals DSO and DSI. FREEZE is asserted at the next instruction boundary after the assertion

Freescale Semiconductor, Inc.

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

of $\overline{\text{BKPT}}$ or execution of the BGND instruction. IPIPE0 and IPIPE1 change function before an exception signal can be generated. The development system must use FREEZE assertion as an indication that BDM has been entered. When BDM is exited, FREEZE is negated before initiation of normal bus cycles. IPIPE0 and IPIPE1 are valid when normal instruction prefetch begins.

4.14.4.4 BDM Commands

Commands consist of one 16-bit operation word and can include one or more 16-bit extension words. Each incoming word is read as it is assembled by the serial interface. The microcode routine corresponding to a command is executed as soon as the command is complete. Result operands are loaded into the output shift register to be shifted out as the next command is read. This process is repeated for each command until the CPU returns to normal operating mode. The BDM command set is summarized in **Table 4-7**. Refer to the *CPU16 Reference Manual* (CPU16RM/AD) for a BDM command glossary.

Table 4-7 Command Summary

Command	Mnemonic	Description
Read Registers from Mask	RREGM	Read contents of registers specified by command word register mask
Write Registers from Mask	WREGM	Write to registers specified by command word register mask
Read MAC Registers	RDMAC	Read contents of entire multiply and accumulate register set
Write MAC Registers	WRMAC	Write to entire multiply and accumulate register set
Read PC and SP	RPCSP	Read contents of program counter and stack pointer
Write PC and SP	WPCSP	Write to program counter and stack pointer
Read Data Memory	RDMEM	Read byte from specified 20-bit address in data space
Write Data Memory	WDMEM	Write byte to specified 20-bit address in data space
Read Program Memory	RPMEM	Read word from specified 20-bit address in program space
Write Program Memory	WPMEM	Write word to specified 20-bit address in program space
Execute from Current PK : PC	GO	Instruction pipeline flushed and refilled; instructions executed from current PC – $0006
Null Operation	NOP	Null command performs no operation

4.14.4.5 Returning from BDM

BDM is terminated when a resume execution (GO) command is received. GO refills the instruction pipeline from address (PK : PC – $0006). FREEZE is negated before the first prefetch. Upon negation of FREEZE, the BDM serial subsystem is disabled and the DSO/DSI signals revert to IPIPE0/IPIPE1 functionality.

Freescale Semiconductor, Inc.

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

4.14.4.6 BDM Serial Interface

The BDM serial interface uses a synchronous protocol similar to that of the Freescale serial peripheral interface (SPI). **Figure 4-7** is a diagram of the serial logic required to use BDM with a development system.

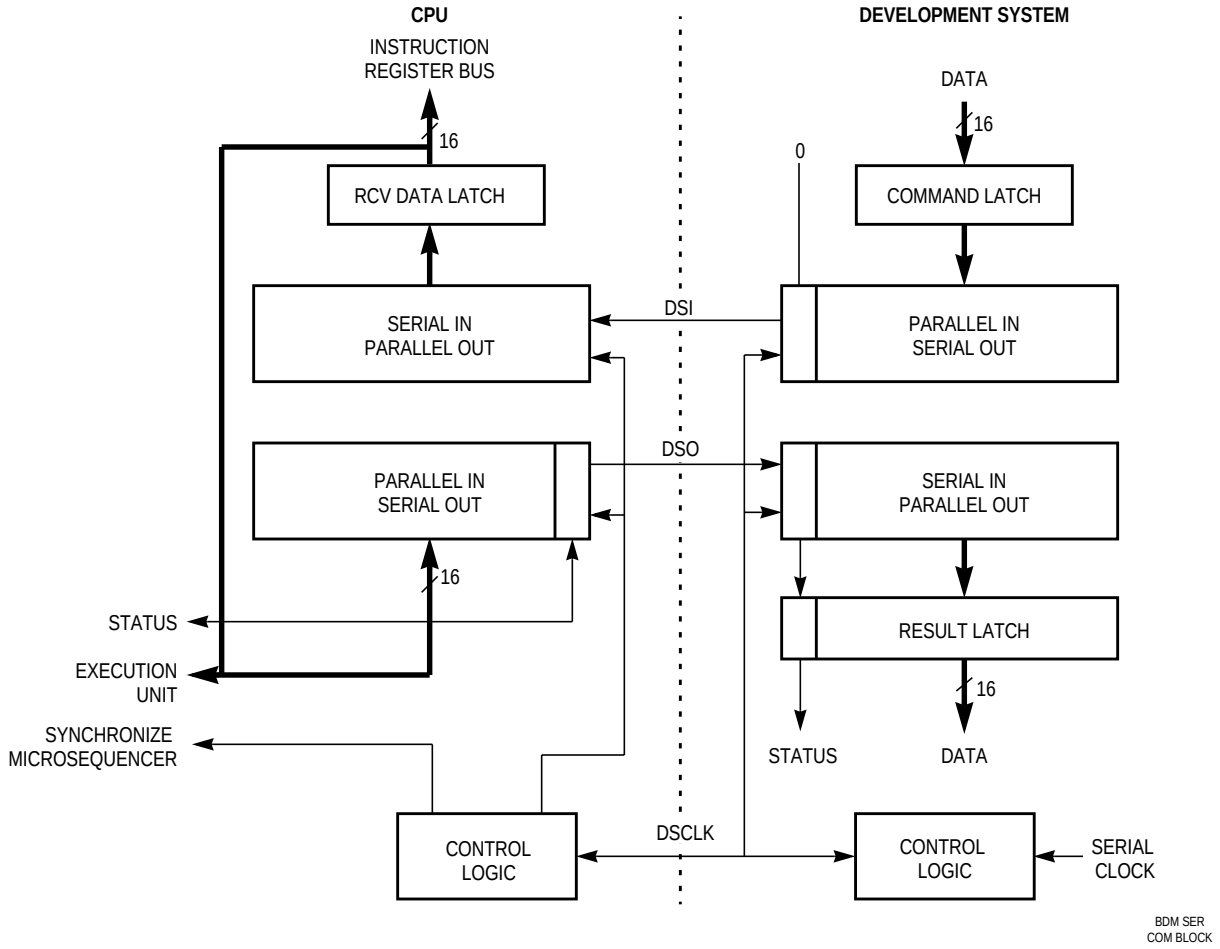


Figure 4-7 BDM Serial I/O Block Diagram

The development system serves as the master of the serial link, and is responsible for the generation of the serial interface clock signal (DSCLK).

Serial clock frequency range is from DC to one-half the CPU16 clock frequency. If DSCLK is derived from the CPU16 system clock, development system serial logic can be synchronized with the target processor.

The serial interface operates in full-duplex mode. Data transfers occur on the falling edge of DSCLK and are stable by the following rising edge of DSCLK. Data is transmitted MSB first, and is latched on the rising edge of DSCLK.

The serial data word is 17 bits wide, which includes 16 data bits and a status/control bit. Bit 16 indicates status of CPU-generated messages.

Command and data transfers initiated by the development system must clear bit 16. All commands that return a result return 16 bits of data plus one status bit.

Freescale Semiconductor, Inc.

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

4.15 Recommended BDM Connection

In order to use BDM development tools when an MCU is installed in a system, Freescale recommends that appropriate signal lines be routed to a male Berg connector or double-row header installed on the circuit board with the MCU. Refer to **Figure 4-8**.

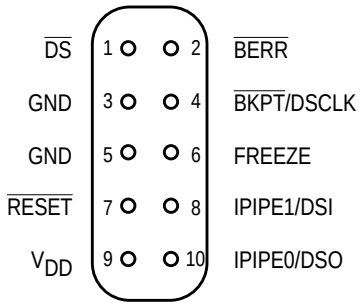


Figure 4-8 BDM Connector Pinout

4.16 Digital Signal Processing

The CPU16 performs low-frequency digital signal processing (DSP) algorithms in real time. The most common DSP operation in embedded control applications is filtering, but the CPU16 can perform several other useful DSP functions. These include auto-correlation (detecting a periodic signal in the presence of noise), cross-correlation (determining the presence of a defined periodic signal), and closed-loop control routines (selective filtration in a feedback path).

Although derivation of DSP algorithms is often a complex mathematical task, the algorithms themselves typically consist of a series of multiply and accumulate (MAC) operations. The CPU16 contains a dedicated set of registers that perform MAC operations. As a group, these registers are called the MAC unit.

DSP operations generally require a large number of MAC iterations. The CPU16 instruction set includes instructions that perform MAC setup and repetitive MAC operations. Other instructions, such as 32-bit load and store instructions, can also be used in DSP routines.

Many DSP algorithms require extensive data address manipulation. To increase throughput, the CPU16 performs effective address calculations and data prefetches during MAC operations. In addition, the MAC unit provides modulo addressing to implement circular DSP buffers efficiently.

Refer to the *CPU16 Reference Manual* (CPU16RM/AD) for detailed information concerning the MAC unit and execution of DSP instructions.

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
SYSTEM INTEGRATION MODULE

This section is an overview of the system integration module (SIM). Refer to the *SIM Reference Manual* (SIMRM/AD) for a comprehensive discussion of SIM capabilities. Refer to **D.2 System Integration Module** for information concerning the SIM address map and register structure.

5.1 General

The SIM consists of six functional blocks. **Figure 5-1** shows a block diagram of the SIM.

The system configuration block controls MCU configuration parameters.

The system clock generates clock signals used by the SIM, other IMB modules, and external devices.

The system protection block provides bus and software watchdog monitors. In addition, it also provides a periodic interrupt timer to support execution of time-critical control routines.

The external bus interface handles the transfer of information between IMB modules and external address space.

The chip-select block provides 12 chip-select signals. Each chip-select signal has an associated base address register and option register that contain the programmable characteristics of that chip-select.

The system test block incorporates hardware necessary for testing the MCU. It is used to perform factory tests, and its use in normal applications is not supported.

Freescale Semiconductor, Inc.

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

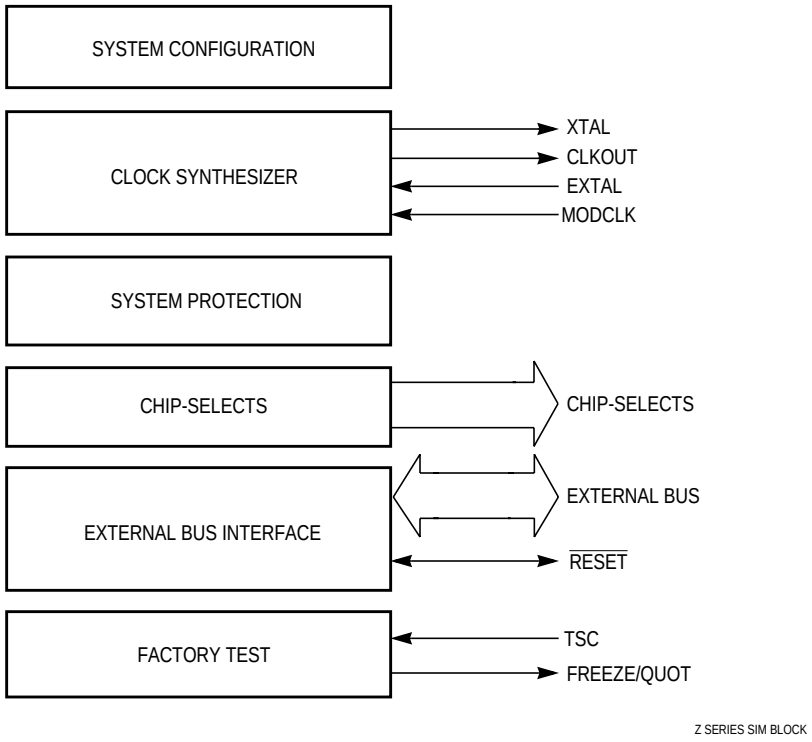


Figure 5-1 System Integration Module Block Diagram

5.2 System Configuration

The SIM configuration register (SIMCR) governs several aspects of system operation. The following paragraphs describe those configuration options controlled by SIMCR.

5.2.1 Module Mapping

Control registers for all the modules in the microcontroller are mapped into a 4-Kbyte block. The state of the module mapping (MM) bit in the SIM configuration register (SIMCR) determines where the control register block is located in the system memory map. When MM = 0, register addresses range from $7FF000 to $7FFFFF; when MM = 1, register addresses range from $FFF000 to $FFFFFF.

In M68HC16 Z-series MCUs, ADDR[23:20] follow the logic state of ADDR19 unless externally driven. MM corresponds to IMB ADDR23. If MM is cleared, the SIM maps IMB modules into address space $7FF000 – $7FFFFF, which is inaccessible to the CPU16. Modules remain inaccessible until reset occurs. The reset state of MM is one, but the bit can be written once. Initialization software should make certain MM remains set.

Freescale Semiconductor, Inc.

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

5.2.2 Interrupt Arbitration

Each module that can request interrupts has an interrupt arbitration (IARB) field. Arbitration between interrupt requests of the same priority is performed by serial contention between IARB field bit values. Contention will take place whenever an interrupt request is acknowledged, even when there is only a single request pending. For an interrupt to be serviced, the appropriate IARB field must have a non-zero value. If an interrupt request from a module with an IARB field value of %0000 is recognized, the CPU16 processes a spurious interrupt exception.

Because the SIM routes external interrupt requests to the CPU16, the SIM IARB field value is used for arbitration between internal and external interrupts of the same priority. The reset value of IARB for the SIM is %1111, and the reset IARB value for all other modules is %0000, which prevents SIM interrupts from being discarded during initialization. Refer to **5.8 Interrupts** for a discussion of interrupt arbitration.

5.2.3 Show Internal Cycles

A show cycle allows internal bus transfers to be monitored externally. The SHEN field in SIMCR determines what the external bus interface does during internal transfer operations. **Table 5-1** shows whether data is driven externally, and whether external bus arbitration can occur. Refer to **5.6.6.1 Show Cycles** for more information.

Table 5-1 Show Cycle Enable Bits

SHEN[1:0]	Action
00	Show cycles disabled, external arbitration enabled
01	Show cycles enabled, external arbitration disabled
10	Show cycles enabled, external arbitration enabled
11	Show cycles enabled, external arbitration enabled; internal activity is halted by a bus grant

5.2.4 Register Access

M68HC16 Z-series MCUs always operate at the supervisor level. The state of the SUPV bit has no meaning.

5.2.5 Freeze Operation

The FREEZE signal halts MCU operations during debugging. FREEZE is asserted internally by the CPU16 if a breakpoint occurs while background mode is enabled. When FREEZE is asserted, only the bus monitor, software watchdog, and periodic interrupt timer are affected. The halt monitor and spurious interrupt monitor continue to operate normally. Setting the freeze bus monitor (FRZBM) bit in SIMCR disables the bus monitor when FREEZE is asserted. Setting the freeze software watchdog (FRZSW) bit disables the software watchdog and the periodic interrupt timer when FREEZE is asserted.

Freescale Semiconductor, Inc.

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

5.3 System Clock

The system clock in the SIM provides timing signals for the IMB modules and for an external peripheral bus. Because the MCU is a fully static design, register and memory contents are not affected when the clock rate changes. System hardware and software support changes in clock rate during operation.

The system clock signal can be generated from one of three sources. An internal phase-locked loop (PLL) can synthesize the clock from a fast reference, a slow reference, or the clock signal can be directly input from an external frequency source.

NOTE

Whether the PLL can use a fast or slow reference is determined by the device. A particular device cannot use both a fast and slow reference.

The fast reference is typically a 4.194-MHz crystal; the slow reference is typically a 32.768-kHz crystal. Each reference frequency may be generated by sources other than a crystal. Keep these sources in mind while reading the rest of this section.

Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for clock specifications.

Figure 5-2 is a block diagram of the clock submodule.

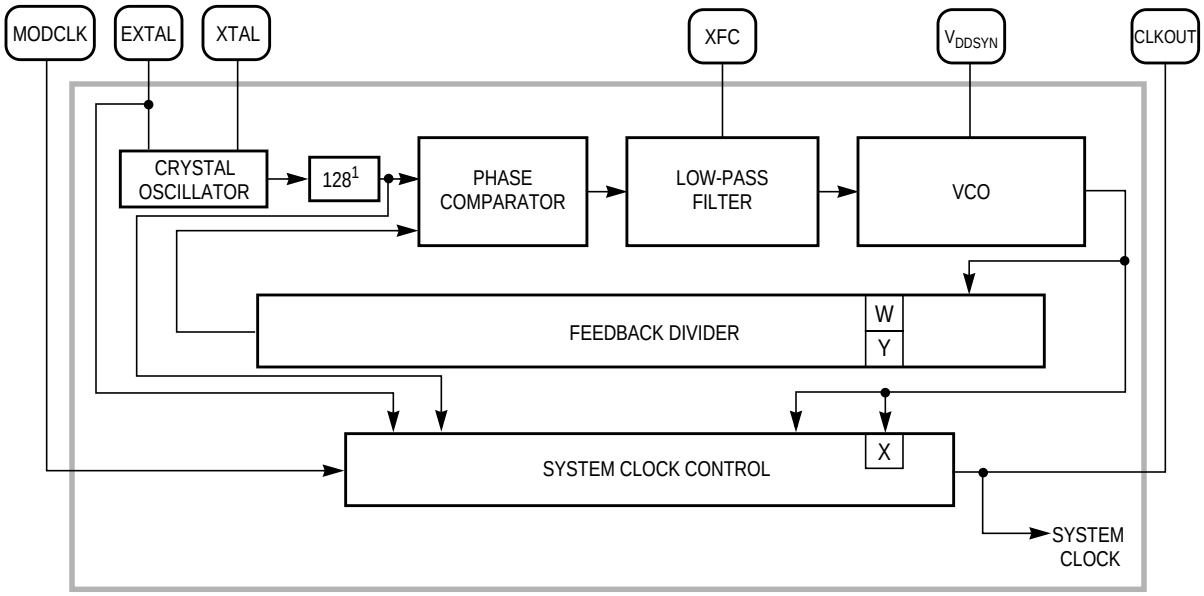


Figure 5-2 System Clock Block Diagram

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

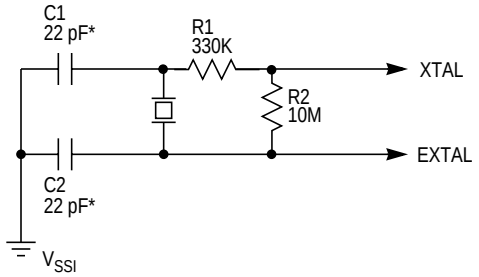
Freescale Semiconductor, Inc.

5.3.1 Clock Sources

The state of the clock mode (MODCLK) pin during reset determines the system clock source. When MODCLK is held high during reset, the clock synthesizer generates a clock signal from an external reference frequency. The clock synthesizer control register (SYNCR) determines operating frequency and mode of operation. When MODCLK is held low during reset, the clock synthesizer is disabled and an external system clock signal must be driven onto the EXTAL pin.

The input clock, referred to as f_{ref}, can be either a crystal or an external clock source. The output of the clock system is referred to as f_{sys}. Ensure that f_{ref} and f_{sys} are within normal operating limits.

To generate a reference frequency using the crystal oscillator, a reference crystal must be connected between the EXTAL and XTAL pins. Typically, a 32.768-kHz crystal is used for a slow reference, but the frequency may vary between 25 kHz to 50 kHz. **Figure 5-3** shows a typical circuit.

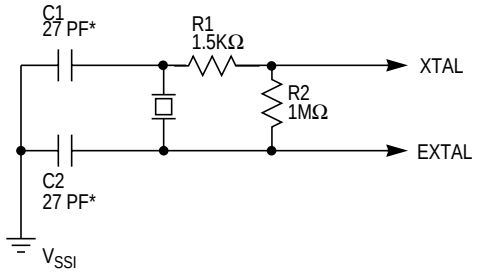


* RESISTANCE AND CAPACITANCE BASED ON A TEST CIRCUIT CONSTRUCTED WITH A DAISHINKU DMX-38 32.768-kHz CRYSTAL. SPECIFIC COMPONENTS MUST BE BASED ON CRYSTAL TYPE. CONTACT CRYSTAL VENDOR FOR EXACT CIRCUIT.

32 OSCILLATOR

Figure 5-3 Slow Reference Crystal Circuit

A 4.194-MHz crystal is typically used for a fast reference, but the frequency may vary between one MHz up to six MHz. **Figure 5-4** shows a typical circuit.



* RESISTANCE AND CAPACITANCE BASED ON A TEST CIRCUIT CONSTRUCTED WITH A KDS041-18 4.194 MHz CRYSTAL. SPECIFIC COMPONENTS MUST BE BASED ON CRYSTAL TYPE. CONTACT CRYSTAL VENDOR FOR EXACT CIRCUIT.

16 OSCILLATOR 4M

Figure 5-4 Fast Reference Crystal Circuit

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

Freescale Semiconductor, Inc.

If a fast or slow reference frequency is provided to the PLL from a source other than a crystal, or an external system clock signal is applied through the EXTAL pin, the XTAL pin must be left floating.

5.3.2 Clock Synthesizer Operation

V_{DDSYN} is used to power the clock circuits when the system clock is synthesized from either a crystal or an externally supplied reference frequency. A separate power source increases MCU noise immunity and can be used to run the clock when the MCU is powered down. A quiet power supply must be used as the V_{DDSYN} source. Adequate external bypass capacitors should be placed as close as possible to the V_{DDSYN} pin to assure a stable operating frequency. When an external system clock signal is applied and the PLL is disabled, V_{DDSYN} should be connected to the V_{DD} supply.

A voltage controlled oscillator (VCO) in the PLL generates the system clock signal. To maintain a 50% clock duty cycle, the VCO frequency (f_{VCO}) is either two or four times the system clock frequency, depending on the state of the X bit in SYNCR. The clock signal is fed back to a divider/counter. The divider controls the frequency of one input to a phase comparator. The other phase comparator input is a reference signal, either from the crystal oscillator or from an external source. The comparator generates a control signal proportional to the difference in phase between the two inputs. This signal is low-pass filtered and used to correct the VCO output frequency.

Filter circuit implementation can vary, depending upon the external environment and required clock stability. **Figure 5-5** shows two recommended system clock filter networks. XFC pin leakage must be kept as low as possible to maintain optimum stability and PLL performance.

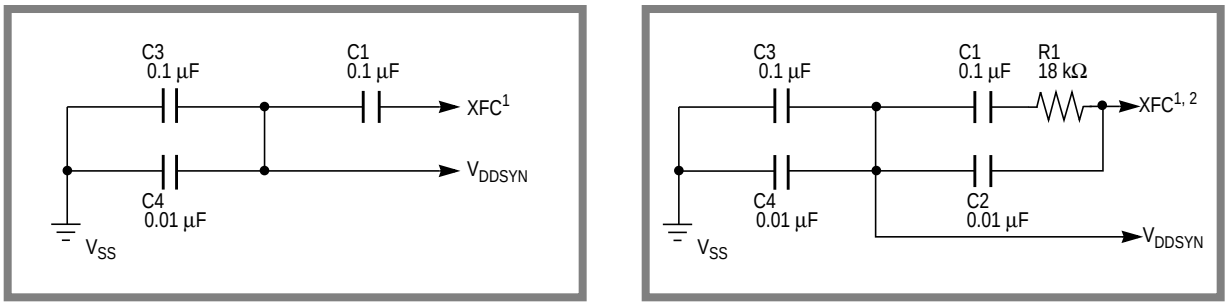
NOTE

The standard filter used in normal operating environments is a single 0.1 µf capacitor, connected from the XFC pin to the V_{DDSYN} supply pin. An alternate filter can be used in high-stability operating environments to reduce PLL jitter under noisy system conditions. Current systems that are operating correctly may not require this filter. If the PLL is not enabled (MODCLK = 0 at reset), the XFC filter is not required. Versions of the SIM that are configured for either slow or fast reference use the same filter component values.

An external filter network connected to the XFC pin is not required when an external system clock signal is applied and the PLL is disabled (MODCLK = 0 at reset). The XFC pin must be left floating in this case.

Freescale Semiconductor, Inc.

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



1. MAINTAIN LOW LEAKAGE ON THE XFC NODE. REFER TO **APPENDIX A ELECTRICAL CHARACTERISTICS** FOR MORE INFORMATION.
2. RECOMMENDED LOOP FILTER FOR REDUCED SENSITIVITY TO LOW FREQUENCY NOISE.

NORMAL/HIGH-STABILITY XFC CONN

Figure 5-5 System Clock Filter Networks

The synthesizer locks when the VCO frequency is equal to f_{ref}. Lock time is affected by the filter time constant and by the amount of difference between the two comparator inputs. Whenever a comparator input changes, the synthesizer must relock. Lock status is shown by the SLOCK bit in SYNCR. During power-up, the MCU does not come out of reset until the synthesizer locks. Crystal type, characteristic frequency, and layout of external oscillator circuitry affect lock time.

When the clock synthesizer is used, SYNCR determines the system clock frequency and certain operating parameters. The W and Y[5:0] bits are located in the PLL feedback path, enabling frequency multiplication by a factor of up to 256. When the W or Y values change, VCO frequency changes, and there is a VCO relock delay. The SYNCR X bit controls a divide-by circuit that is not in the synthesizer feedback loop. When X = 0 (reset state), a divide-by-four circuit is enabled, and the system clock frequency is one-fourth the VCO frequency (f_{VCO}). When X = 1, a divide-by-two circuit is enabled and system clock frequency is one-half the VCO frequency (f_{VCO}). There is no relock delay when clock speed is changed by the X bit.

When a slow reference is used, one W bit and six Y bits are located in the PLL feedback path, enabling frequency multiplication by a factor of up to 256. The X bit is located in the VCO clock output path to enable dividing the system clock frequency by two without disturbing the PLL.

When using a slow reference, the clock frequency is determined by SYNCR bit settings as follows:

$$f_{sys} = 4f_{ref}(Y + 1)(2^{(2W + X)})$$

The reset state of SYNCR ($3F00) results in a power-on f_{sys} of 8.388 MHz when f_{ref} is 32.768 kHz.

Freescale Semiconductor, Inc.

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

When a fast reference is used, three W bits are located in the PLL feedback path, enabling frequency multiplication by a factor from one to eight. Three Y bits and the X bit are located in the VCO clock output path to provide the ability to slow the system clock without disturbing the PLL.

When using a fast reference, the clock frequency is determined by SYNCR bit settings as follows:

$$f_{sys} = \frac{f_{ref}}{128}[4(Y+1)(2^{(2W+X)})]$$

The reset state of SYNCR ($3F00) results in a power-on f_{sys} of 8.388 MHz when f_{ref} is 4.194 MHz.

For the device to perform correctly, both the clock frequency and VCO frequency (selected by the W, X, and Y bits) must be within the limits specified for the MCU. In order for the VCO frequency to be within specifications (less than or equal to the maximum system clock frequency multiplied by two), the X bit must be set for system clock frequencies greater than one-half the maximum specified system clock.

Internal VCO frequency is determined by the following equations:

$$f_{VCO} = 4f_{sys} \text{ if } X = 0$$

or

$$f_{VCO} = 2f_{sys} \text{ if } X = 1$$

On both slow and fast reference devices, when an external system clock signal is applied (MODCLK = 0 during reset), the PLL is disabled. The duty cycle of this signal is critical, especially at operating frequencies close to maximum. The relationship between clock signal duty cycle and clock signal period is expressed as follows:

$$\text{Minimum External Clock Period} = \frac{\text{Minimum External Clock High/Low Time}}{50\% - \text{Percentage Variation of External Clock Input Duty Cycle}}$$

Tables 5-2, **5-3**, and **5-4** show clock control multipliers for all possible combinations of SYNCR bits. To obtain clock frequency, find the counter modulus in the leftmost column, then multiply the reference frequency by the value in the appropriate prescaler cell. Shaded areas indicate which values exceed the specifications for a device rated at a particular operating frequency. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for maximum allowable clock rate.

Tables 5-5, **5-6**, and **5-7** show actual clock frequencies for the same combinations of SYNCR bits. To obtain clock frequency, find the counter modulus in the leftmost column, then refer to appropriate prescaler cell. Shaded areas indicate which values exceed the specifications for a device rated at a particular operating frequency. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for maximum system frequency (f).

Freescale Semiconductor, Inc.

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

Table 5-2 16.78-MHz Clock Control Multipliers

(Shaded cells represent values that exceed 16.78 MHz specifications.)

Modulus	Prescalers							
	[W:X] = 00 (f_{VCO} = 2 × Value)		[W:X] = 01 (f_{VCO} = Value)		[W:X] = 10 (f_{VCO} = 2 × Value)		[W:X] = 11 (f_{VCO} = Value)	
Y	Slow	Fast	Slow	Fast	Slow	Fast	Slow	Fast
000000	4	.03125	8	.625	16	.125	32	.25
000001	8	.0625	16	.125	32	.25	64	.5
000010	12	.09375	24	.1875	48	.375	96	.75
000011	16	.125	32	.25	64	.5	128	1
000100	20	.15625	40	.3125	80	.625	160	1.25
000101	24	.1875	48	.375	96	.75	192	1.5
000110	28	.21875	56	.4375	112	.875	224	1.75
000111	32	.25	64	.5	128	1	256	2
001000	36	.21825	72	.5625	144	1.125	288	2.25
001001	40	.3125	80	.625	160	1.25	320	2.5
001010	44	.34375	88	.6875	176	1.375	352	2.75
001011	48	.375	96	.75	192	1.5	384	3
001100	52	.40625	104	.8125	208	1.625	416	3.25
001101	56	.4375	112	.875	224	1.75	448	3.5
001110	60	.46875	120	.9375	240	1.875	480	3.75
001111	64	.5	128	1	256	2	512	4
010000	68	.53125	136	1.0625	272	2.125	544	4.25
010001	72	.5625	144	1.125	288	2.25	576	4.5
010010	76	.59375	152	1.1875	304	2.375	608	4.75
010011	80	.625	160	1.25	320	2.5	640	5
010100	84	.65625	168	1.3125	336	2.625	672	5.25
010101	88	.6875	176	1.375	352	2.75	704	5.5
010110	92	.71875	184	1.4375	368	2.875	736	5.75
010111	96	.75	192	1.5	384	3	768	6
011000	100	.78125	200	1.5625	400	3.125	800	6.25
011001	104	.8125	208	1.625	416	3.25	832	6.5
011010	108	.84375	216	1.6875	432	3.375	864	6.75
011011	112	.875	224	1.75	448	3.5	896	7
011100	116	.90625	232	1.8125	464	3.625	928	7.25
011101	120	.9375	240	1.875	480	3.75	960	7.5
011110	124	.96875	248	1.9375	496	3.875	992	7.75
011111	128	1	256	2	512	4	1024	8

Freescale Semiconductor, Inc.

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

Table 5-2 16.78-MHz Clock Control Multipliers (Continued)

(Shaded cells represent values that exceed 16.78 MHz specifications.)

Modulus	Prescalers							
	[W:X] = 00 (f_{VCO} = 2 × Value)		[W:X] = 01 (f_{VCO} = Value)		[W:X] = 10 (f_{VCO} = 2 × Value)		[W:X] = 11 (f_{VCO} = Value)	
Y	Slow	Fast	Slow	Fast	Slow	Fast	Slow	Fast
100000	132	1.03125	264	2.0625	528	4.125	1056	8.25
100001	136	1.0625	272	2.125	544	4.25	1088	8.5
100010	140	1.09375	280	2.1875	560	4.375	1120	8.75
100011	144	1.125	288	2.25	576	4.5	1152	9
100100	148	1.15625	296	2.3125	592	4.675	1184	9.25
100101	152	1.1875	304	2.375	608	4.75	1216	9.5
100110	156	1.21875	312	2.4375	624	4.875	1248	9.75
100111	160	1.25	320	2.5	640	5	1280	10
101000	164	1.28125	328	2.5625	656	5.125	1312	10.25
101001	168	1.3125	336	2.625	672	5.25	1344	10.5
101010	172	1.34375	344	2.6875	688	5.375	1376	10.75
101011	176	1.375	352	2.75	704	5.5	1408	11
101100	180	1.40625	360	2.8125	720	5.625	1440	11.25
101101	184	1.4375	368	2.875	736	5.75	1472	11.5
101110	188	1.46875	376	2.9375	752	5.875	1504	11.75
101111	192	1.5	384	3	768	6	1536	12
110000	196	1.53125	392	3.0625	784	6.125	1568	12.25
110001	200	1.5625	400	3.125	800	6.25	1600	12.5
110010	204	1.59375	408	3.1875	816	6.375	1632	12.75
110011	208	1.625	416	3.25	832	6.5	1664	13
110100	212	1.65625	424	3.3125	848	6.625	1696	13.25
110101	216	1.6875	432	3.375	864	6.75	1728	13.5
110110	220	1.71875	440	3.4375	880	6.875	1760	13.75
110111	224	1.75	448	3.5	896	7	1792	14
111000	228	1.78125	456	3.5625	912	7.125	1824	14.25
111001	232	1.8125	464	3.625	928	7.25	1856	14.5
111010	236	1.84375	472	3.6875	944	7.375	1888	14.75
111011	240	1.875	480	3.75	960	7.5	1920	15
111100	244	1.90625	488	3.8125	976	7.625	1952	15.25
111101	248	1.9375	496	3.875	992	7.75	1984	15.5
111110	252	1.96875	504	3.9375	1008	7.875	2016	15.75
111111	256	2	512	4	1024	8	2048	16

Freescale Semiconductor, Inc.

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

Table 5-3 20.97-MHz Clock Control Multipliers

(Shaded cells represent values that exceed 20.97 MHz specifications.)

Modulus	Prescalers							
	[W:X] = 00 (f_{VCO} = 2 × Value)		[W:X] = 01 (f_{VCO} = Value)		[W:X] = 10 (f_{VCO} = 2 × Value)		[W:X] = 11 (f_{VCO} = Value)	
Y	Slow	Fast	Slow	Fast	Slow	Fast	Slow	Fast
000000	4	.03125	8	.625	16	.125	32	.25
000001	8	.0625	16	.125	32	.25	64	.5
000010	12	.09375	24	.1875	48	.375	96	.75
000011	16	.125	32	.25	64	.5	128	1
000100	20	.15625	40	.3125	80	.625	160	1.25
000101	24	.1875	48	.375	96	.75	192	1.5
000110	28	.21875	56	.4375	112	.875	224	1.75
000111	32	.25	64	.5	128	1	256	2
001000	36	.21825	72	.5625	144	1.125	288	2.25
001001	40	.3125	80	.625	160	1.25	320	2.5
001010	44	.34375	88	.6875	176	1.375	352	2.75
001011	48	.375	96	.75	192	1.5	384	3
001100	52	.40625	104	.8125	208	1.625	416	3.25
001101	56	.4375	112	.875	224	1.75	448	3.5
001110	60	.46875	120	.9375	240	1.875	480	3.75
001111	64	.5	128	1	256	2	512	4
010000	68	.53125	136	1.0625	272	2.125	544	4.25
010001	72	.5625	144	1.125	288	2.25	576	4.5
010010	76	.59375	152	1.1875	304	2.375	608	4.75
010011	80	.625	160	1.25	320	2.5	640	5
010100	84	.65625	168	1.3125	336	2.625	672	5.25
010101	88	.6875	176	1.375	352	2.75	704	5.5
010110	92	.71875	184	1.4375	368	2.875	736	5.75
010111	96	.75	192	1.5	384	3	768	6
011000	100	.78125	200	1.5625	400	3.125	800	6.25
011001	104	.8125	208	1.625	416	3.25	832	6.5
011010	108	.84375	216	1.6875	432	3.375	864	6.75
011011	112	.875	224	1.75	448	3.5	896	7
011100	116	.90625	232	1.8125	464	3.625	928	7.25
011101	120	.9375	240	1.875	480	3.75	960	7.5
011110	124	.96875	248	1.9375	496	3.875	992	7.75
011111	128	1	256	2	512	4	1024	8

Freescale Semiconductor, Inc.

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

Table 5-3 20.97-MHz Clock Control Multipliers (Continued)

(Shaded cells represent values that exceed 20.97 MHz specifications.)

Modulus	Prescalers							
	[W:X] = 00 (f_{VCO} = 2 × Value)		[W:X] = 01 (f_{VCO} = Value)		[W:X] = 10 (f_{VCO} = 2 × Value)		[W:X] = 11 (f_{VCO} = Value)	
Y	Slow	Fast	Slow	Fast	Slow	Fast	Slow	Fast
100000	132	1.03125	264	2.0625	528	4.125	1056	8.25
100001	136	1.0625	272	2.125	544	4.25	1088	8.5
100010	140	1.09375	280	2.1875	560	4.375	1120	8.75
100011	144	1.125	288	2.25	576	4.5	1152	9
100100	148	1.15625	296	2.3125	592	4.675	1184	9.25
100101	152	1.1875	304	2.375	608	4.75	1216	9.5
100110	156	1.21875	312	2.4375	624	4.875	1248	9.75
100111	160	1.25	320	2.5	640	5	1280	10
101000	164	1.28125	328	2.5625	656	5.125	1312	10.25
101001	168	1.3125	336	2.625	672	5.25	1344	10.5
101010	172	1.34375	344	2.6875	688	5.375	1376	10.75
101011	176	1.375	352	2.75	704	5.5	1408	11
101100	180	1.40625	360	2.8125	720	5.625	1440	11.25
101101	184	1.4375	368	2.875	736	5.75	1472	11.5
101110	188	1.46875	376	2.9375	752	5.875	1504	11.75
101111	192	1.5	384	3	768	6	1536	12
110000	196	1.53125	392	3.0625	784	6.125	1568	12.25
110001	200	1.5625	400	3.125	800	6.25	1600	12.5
110010	204	1.59375	408	3.1875	816	6.375	1632	12.75
110011	208	1.625	416	3.25	832	6.5	1664	13
110100	212	1.65625	424	3.3125	848	6.625	1696	13.25
110101	216	1.6875	432	3.375	864	6.75	1728	13.5
110110	220	1.71875	440	3.4375	880	6.875	1760	13.75
110111	224	1.75	448	3.5	896	7	1792	14
111000	228	1.78125	456	3.5625	912	7.125	1824	14.25
111001	232	1.8125	464	3.625	928	7.25	1856	14.5
111010	236	1.84375	472	3.6875	944	7.375	1888	14.75
111011	240	1.875	480	3.75	960	7.5	1920	15
111100	244	1.90625	488	3.8125	976	7.625	1952	15.25
111101	248	1.9375	496	3.875	992	7.75	1984	15.5
111110	252	1.96875	504	3.9375	1008	7.875	2016	15.75
111111	256	2	512	4	1024	8	2048	16

Freescale Semiconductor, Inc.

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

Table 5-4 25.17-MHz Clock Control Multipliers

(Shaded cells represent values that exceed 25.17 MHz specifications.)

Modulus	Prescalers							
	[W:X] = 00 (f_{VCO} = 2 × Value)		[W:X] = 01 (f_{VCO} = Value)		[W:X] = 10 (f_{VCO} = 2 × Value)		[W:X] = 11 (f_{VCO} = Value)	
Y	Slow	Fast	Slow	Fast	Slow	Fast	Slow	Fast
000000	4	.03125	8	.625	16	.125	32	.25
000001	8	.0625	16	.125	32	.25	64	.5
000010	12	.09375	24	.1875	48	.375	96	.75
000011	16	.125	32	.25	64	.5	128	1
000100	20	.15625	40	.3125	80	.625	160	1.25
000101	24	.1875	48	.375	96	.75	192	1.5
000110	28	.21875	56	.4375	112	.875	224	1.75
000111	32	.25	64	.5	128	1	256	2
001000	36	.21825	72	.5625	144	1.125	288	2.25
001001	40	.3125	80	.625	160	1.25	320	2.5
001010	44	.34375	88	.6875	176	1.375	352	2.75
001011	48	.375	96	.75	192	1.5	384	3
001100	52	.40625	104	.8125	208	1.625	416	3.25
001101	56	.4375	112	.875	224	1.75	448	3.5
001110	60	.46875	120	.9375	240	1.875	480	3.75
001111	64	.5	128	1	256	2	512	4
010000	68	.53125	136	1.0625	272	2.125	544	4.25
010001	72	.5625	144	1.125	288	2.25	576	4.5
010010	76	.59375	152	1.1875	304	2.375	608	4.75
010011	80	.625	160	1.25	320	2.5	640	5
010100	84	.65625	168	1.3125	336	2.625	672	5.25
010101	88	.6875	176	1.375	352	2.75	704	5.5
010110	92	.71875	184	1.4375	368	2.875	736	5.75
010111	96	.75	192	1.5	384	3	768	6
011000	100	.78125	200	1.5625	400	3.125	800	6.25
011001	104	.8125	208	1.625	416	3.25	832	6.5
011010	108	.84375	216	1.6875	432	3.375	864	6.75
011011	112	.875	224	1.75	448	3.5	896	7
011100	116	.90625	232	1.8125	464	3.625	928	7.25
011101	120	.9375	240	1.875	480	3.75	960	7.5
011110	124	.96875	248	1.9375	496	3.875	992	7.75
011111	128	1	256	2	512	4	1024	8

Freescale Semiconductor, Inc.

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

Table 5-4 25.17-MHz Clock Control Multipliers (Continued)

(Shaded cells represent values that exceed 25.17 MHz specifications.)

Modulus	Prescalers							
	[W:X] = 00 (f_{VCO} = 2 × Value)		[W:X] = 01 (f_{VCO} = Value)		[W:X] = 10 (f_{VCO} = 2 × Value)		[W:X] = 11 (f_{VCO} = Value)	
Y	Slow	Fast	Slow	Fast	Slow	Fast	Slow	Fast
100000	132	1.03125	264	2.0625	528	4.125	1056	8.25
100001	136	1.0625	272	2.125	544	4.25	1088	8.5
100010	140	1.09375	280	2.1875	560	4.375	1120	8.75
100011	144	1.125	288	2.25	576	4.5	1152	9
100100	148	1.15625	296	2.3125	592	4.675	1184	9.25
100101	152	1.1875	304	2.375	608	4.75	1216	9.5
100110	156	1.21875	312	2.4375	624	4.875	1248	9.75
100111	160	1.25	320	2.5	640	5	1280	10
101000	164	1.28125	328	2.5625	656	5.125	1312	10.25
101001	168	1.3125	336	2.625	672	5.25	1344	10.5
101010	172	1.34375	344	2.6875	688	5.375	1376	10.75
101011	176	1.375	352	2.75	704	5.5	1408	11
101100	180	1.40625	360	2.8125	720	5.625	1440	11.25
101101	184	1.4375	368	2.875	736	5.75	1472	11.5
101110	188	1.46875	376	2.9375	752	5.875	1504	11.75
101111	192	1.5	384	3	768	6	1536	12
110000	196	1.53125	392	3.0625	784	6.125	1568	12.25
110001	200	1.5625	400	3.125	800	6.25	1600	12.5
110010	204	1.59375	408	3.1875	816	6.375	1632	12.75
110011	208	1.625	416	3.25	832	6.5	1664	13
110100	212	1.65625	424	3.3125	848	6.625	1696	13.25
110101	216	1.6875	432	3.375	864	6.75	1728	13.5
110110	220	1.71875	440	3.4375	880	6.875	1760	13.75
110111	224	1.75	448	3.5	896	7	1792	14
111000	228	1.78125	456	3.5625	912	7.125	1824	14.25
111001	232	1.8125	464	3.625	928	7.25	1856	14.5
111010	236	1.84375	472	3.6875	944	7.375	1888	14.75
111011	240	1.875	480	3.75	960	7.5	1920	15
111100	244	1.90625	488	3.8125	976	7.625	1952	15.25
111101	248	1.9375	496	3.875	992	7.75	1984	15.5
111110	252	1.96875	504	3.9375	1008	7.875	2016	15.75
111111	256	2	512	4	1024	8	2048	16

Freescale Semiconductor, Inc.

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

Table 5-5 16.78-MHz System Clock Frequencies

(Shaded cells represent values that exceed 16.78 MHz specifications.)

Modulus	Prescaler			
Y	[W:X] = 00 (f_{VCO} = 2 × Value)	[W:X] = 01 (f_{VCO} = Value)	[W:X] = 10 (f_{VCO} = 2 × Value)	[W:X] = 11 (f_{VCO} = Value)
000000	131 kHz	262 kHz	524 kHz	1049 kHz
000001	262	524	1049	2097
000010	393	786	1573	3146
000011	524	1049	2097	4194
000100	655	1311	2621	5243
000101	786	1573	3146	6291
000110	918	1835	3670	7340
000111	1049	2097	4194	8389
001000	1180	2359	4719	9437
001001	1311	2621	5243	10486
001010	1442	2884	5767	11534
001011	1573	3146	6291	12583
001100	1704	3408	6816	13631
001101	1835	3670	7340	14680
001110	1966	3932	7864	15729
001111	2097	4194	8389	16777
010000	2228	4456	8913	17826
010001	2359	4719	9437	18874
010010	2490	4981	9961	19923
010011	2621	5243	10486	20972
010100	2753	5505	11010	22020
010101	2884	5767	11534	23069
010110	3015	6029	12059	24117
010111	3146	6291	12583	25166
011000	3277	6554	13107	26214
011001	3408	6816	13631	27263
011010	3539	7078	14156	28312
011011	3670	7340	14680	29360
011100	3801	7602	15204	30409
011101	3932	7864	15729	31457
011110	4063	8126	16253	32506
011111	4194	8389	16777	33554

Freescale Semiconductor, Inc.

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

Table 5-5 16.78-MHz System Clock Frequencies (Continued)

(Shaded cells represent values that exceed 16.78 MHz specifications.)

Modulus	Prescaler			
Y	[W:X] = 00 (f_{VCO} = 2 × Value)	[W:X] = 01 (f_{VCO} = Value)	[W:X] = 10 (f_{VCO} = 2 × Value)	[W:X] = 11 (f_{VCO} = Value)
100000	4325 kHz	8651 kHz	17302 kHz	34603 kHz
100001	4456	8913	17826	35652
100010	4588	9175	18350	36700
100011	4719	9437	18874	37749
100100	4850	9699	19399	38797
100101	4981	9961	19923	39846
100110	5112	10224	20447	40894
100111	5243	10486	20972	41943
101000	5374	10748	21496	42992
101001	5505	11010	22020	44040
101010	5636	11272	22544	45089
101011	5767	11534	23069	46137
101100	5898	11796	23593	47186
101101	6029	12059	24117	48234
101110	6160	12321	24642	49283
101111	6291	12583	25166	50332
110000	6423	12845	25690	51380
110001	6554	13107	26214	52428
110010	6685	13369	26739	53477
110011	6816	13631	27263	54526
110100	6947	13894	27787	55575
110101	7078	14156	28312	56623
110110	7209	14418	28836	57672
110111	7340	14680	29360	58720
111000	7471	14942	2988	59769
111001	7602	15204	30409	60817
111010	7733	15466	30933	61866
111011	7864	15729	31457	62915
111100	7995	15991	31982	63963
111101	8126	16253	32506	65011
111110	8258	16515	33030	66060
111111	8389	16777	33554	67109

Freescale Semiconductor, Inc.

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

Table 5-6 System Clock Frequencies for a 20.97-MHz System

(Shaded cells represent values that exceed 20.97 MHz specifications.)

Modulus	Prescaler			
Y	[W:X] = 00 (f_{VCO} = 2 × Value)	[W:X] = 01 (f_{VCO} = Value)	[W:X] = 10 (f_{VCO} = 2 × Value)	[W:X] = 11 (f_{VCO} = Value)
000000	131 kHz	262 kHz	524 kHz	1049 kHz
000001	262	524	1049	2097
000010	393	786	1573	3146
000011	524	1049	2097	4194
000100	655	1311	2621	5243
000101	786	1573	3146	6291
000110	918	1835	3670	7340
000111	1049	2097	4194	8389
001000	1180	2359	4719	9437
001001	1311	2621	5243	10486
001010	1442	2884	5767	11534
001011	1573	3146	6291	12583
001100	1704	3408	6816	13631
001101	1835	3670	7340	14680
001110	1966	3932	7864	15729
001111	2097	4194	8389	16777
010000	2228	4456	8913	17826
010001	2359	4719	9437	18874
010010	2490	4981	9961	19923
010011	2621	5243	10486	20972
010100	2753	5505	11010	22020
010101	2884	5767	11534	23069
010110	3015	6029	12059	24117
010111	3146	6291	12583	25166
011000	3277	6554	13107	26214
011001	3408	6816	13631	27263
011010	3539	7078	14156	28312
011011	3670	7340	14680	29360
011100	3801	7602	15204	30409
011101	3932	7864	15729	31457
011110	4063	8126	16253	32506
011111	4194	8389	16777	33554

Freescale Semiconductor, Inc.

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

Table 5-6 System Clock Frequencies for a 20.97-MHz System (Continued)

(Shaded cells represent values that exceed 20.97 MHz specifications.)

Modulus	Prescaler			
Y	[W:X] = 00 (f_{VCO} = 2 × Value)	[W:X] = 01 (f_{VCO} = Value)	[W:X] = 10 (f_{VCO} = 2 × Value)	[W:X] = 11 (f_{VCO} = Value)
100000	4325 kHz	8651 kHz	17302 kHz	34603 kHz
100001	4456	8913	17826	35652
100010	4588	9175	18350	36700
100011	4719	9437	18874	37749
100100	4850	9699	19399	38797
100101	4981	9961	19923	39846
100110	5112	10224	20447	40894
100111	5243	10486	20972	41943
101000	5374	10748	21496	42992
101001	5505	11010	22020	44040
101010	5636	11272	22544	45089
101011	5767	11534	23069	46137
101100	5898	11796	23593	47186
101101	6029	12059	24117	48234
101110	6160	12321	24642	49283
101111	6291	12583	25166	50332
110000	6423	12845	25690	51380
110001	6554	13107	26214	52428
110010	6685	13369	26739	53477
110011	6816	13631	27263	54526
110100	6947	13894	27787	55575
110101	7078	14156	28312	56623
110110	7209	14418	28836	57672
110111	7340	14680	29360	58720
111000	7471	14942	2988	59769
111001	7602	15204	30409	60817
111010	7733	15466	30933	61866
111011	7864	15729	31457	62915
111100	7995	15991	31982	63963
111101	8126	16253	32506	65011
111110	8258	16515	33030	66060
111111	8389	16777	33554	67109

Freescale Semiconductor, Inc.

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

Table 5-7 System Clock Frequencies for a 25.17-MHz System

(Shaded cells represent values that exceed 25.17 MHz specifications.)

Modulus	Prescaler			
Y	[W:X] = 00 (f_{VCO} = 2 × Value)	[W:X] = 01 (f_{VCO} = Value)	[W:X] = 10 (f_{VCO} = 2 × Value)	[W:X] = 11 (f_{VCO} = Value)
000000	131 kHz	262 kHz	524 kHz	1049 kHz
000001	262	524	1049	2097
000010	393	786	1573	3146
000011	524	1049	2097	4194
000100	655	1311	2621	5243
000101	786	1573	3146	6291
000110	918	1835	3670	7340
000111	1049	2097	4194	8389
001000	1180	2359	4719	9437
001001	1311	2621	5243	10486
001010	1442	2884	5767	11534
001011	1573	3146	6291	12583
001100	1704	3408	6816	13631
001101	1835	3670	7340	14680
001110	1966	3932	7864	15729
001111	2097	4194	8389	16777
010000	2228	4456	8913	17826
010001	2359	4719	9437	18874
010010	2490	4981	9961	19923
010011	2621	5243	10486	20972
010100	2753	5505	11010	22020
010101	2884	5767	11534	23069
010110	3015	6029	12059	24117
010111	3146	6291	12583	25166
011000	3277	6554	13107	26214
011001	3408	6816	13631	27263
011010	3539	7078	14156	28312
011011	3670	7340	14680	29360
011100	3801	7602	15204	30409
011101	3932	7864	15729	31457
011110	4063	8126	16253	32506
011111	4194	8389	16777	33554

Freescale Semiconductor, Inc.

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

Table 5-7 System Clock Frequencies for a 25.17-MHz System (Continued)

(Shaded cells represent values that exceed 25.17 MHz specifications.)

Modulus	Prescaler			
Y	[W:X] = 00 (f_{VCO} = 2 × Value)	[W:X] = 01 (f_{VCO} = Value)	[W:X] = 10 (f_{VCO} = 2 × Value)	[W:X] = 11 (f_{VCO} = Value)
100000	4325 kHz	8651 kHz	17302 kHz	34603 kHz
100001	4456	8913	17826	35652
100010	4588	9175	18350	36700
100011	4719	9437	18874	37749
100100	4850	9699	19399	38797
100101	4981	9961	19923	39846
100110	5112	10224	20447	40894
100111	5243	10486	20972	41943
101000	5374	10748	21496	42992
101001	5505	11010	22020	44040
101010	5636	11272	22544	45089
101011	5767	11534	23069	46137
101100	5898	11796	23593	47186
101101	6029	12059	24117	48234
101110	6160	12321	24642	49283
101111	6291	12583	25166	50332
110000	6423	12845	25690	51380
110001	6554	13107	26214	52428
110010	6685	13369	26739	53477
110011	6816	13631	27263	54526
110100	6947	13894	27787	55575
110101	7078	14156	28312	56623
110110	7209	14418	28836	57672
110111	7340	14680	29360	58720
111000	7471	14942	2988	59769
111001	7602	15204	30409	60817
111010	7733	15466	30933	61866
111011	7864	15729	31457	62915
111100	7995	15991	31982	63963
111101	8126	16253	32506	65011
111110	8258	16515	33030	66060
111111	8389	16777	33554	67109

Freescale Semiconductor, Inc.

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

5.3.3 External Bus Clock

The state of the E-clock division bit (EDIV) in SYNCR determines clock rate for the E-clock signal (ECLK) available on pin ADDR23. ECLK is a bus clock for MC6800 devices and peripherals. ECLK frequency can be set to system clock frequency divided by eight or system clock frequency divided by sixteen. The clock is enabled by the CS10PA[1:0] field in chip-select pin assignment register 1 (CSPAR1). ECLK operation during low-power stop is described in the following paragraph. Refer to **5.9 Chip-Selects** for more information about the external bus clock.

5.3.4 Low-Power Operation

Low-power operation is initiated by the CPU16. To reduce power consumption selectively, the CPU can set the STOP bits in each module configuration register. To minimize overall microcontroller power consumption, the CPU can execute the LPSTOP instruction which causes the SIM to turn off the system clock.

When individual module STOP bits are set, clock signals inside each module are turned off, but module registers are still accessible.

When the CPU executes LPSTOP, a special CPU space bus cycle writes a copy of the current interrupt mask into the clock control logic. The SIM brings the MCU out of low-power stop mode when one of the following exceptions occur:

- $\overline{\text{RESET}}$
- Trace
- SIM interrupt of higher priority than the stored interrupt mask

Refer to **5.6.4.2 LPSTOP Broadcast Cycle** for more information.

During a low-power stop mode, unless the system clock signal is supplied by an external source and that source is removed, the SIM clock control logic and the SIM clock signal (SIMCLK) continue to operate. The periodic interrupt timer and input logic for the $\overline{\text{RESET}}$ and $\overline{\text{IRQ}}$ pins are clocked by SIMCLK. The SIM can also continue to generate the CLKOUT signal while in low-power stop mode.

During low-power stop mode, the address bus continues to drive the LPSTOP instruction, and bus control signals are negated. I/O pins configured as outputs continue to hold their previous state; I/O pins configured as inputs will be in a high-impedance state.

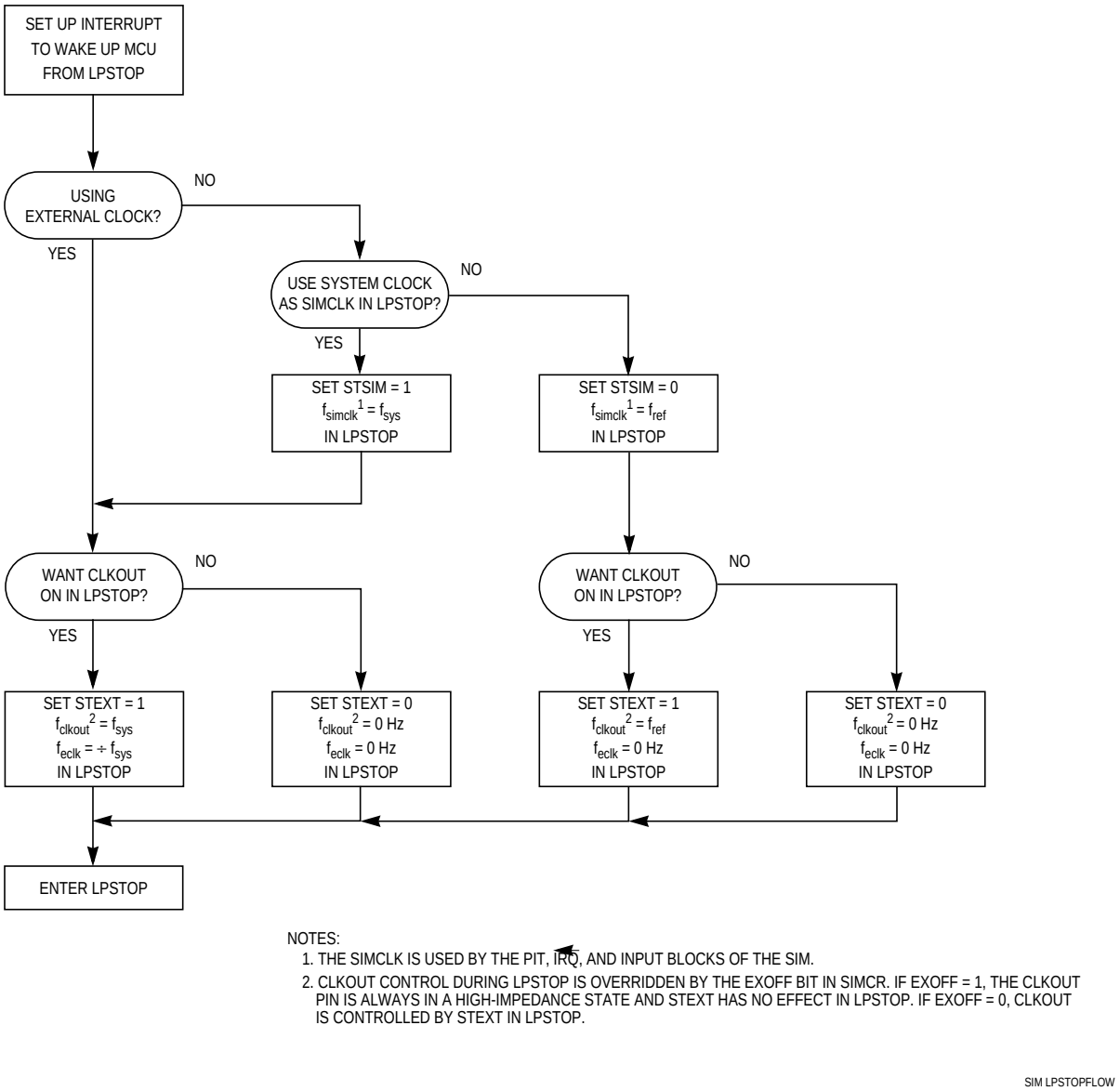
STSIM and STEXT bits in SYNCR determine clock operation during low-power stop mode.

The flowchart shown in **Figure 5-6** summarizes the effects of the STSIM and STEXT bits when MC68HC16Z1, MC68CK16Z1, MC68CM16Z1, MC68HC16Z2, and MC68HC16Z3 MCUs enter normal low-power stop mode. Any clock in the off state is held low. If the synthesizer VCO is turned off during low-power stop mode, there is a PLL relock delay after the VCO is turned back on. **Figure 5-7** summarizes the effects of the STSIM and STEXT bits when MC68HC16Z4 and MC68CK16Z4 MCUs enter normal low-power stop mode.

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

NOTE

The internal oscillator which supplies the input frequency for the PLL always runs when a crystal is used.



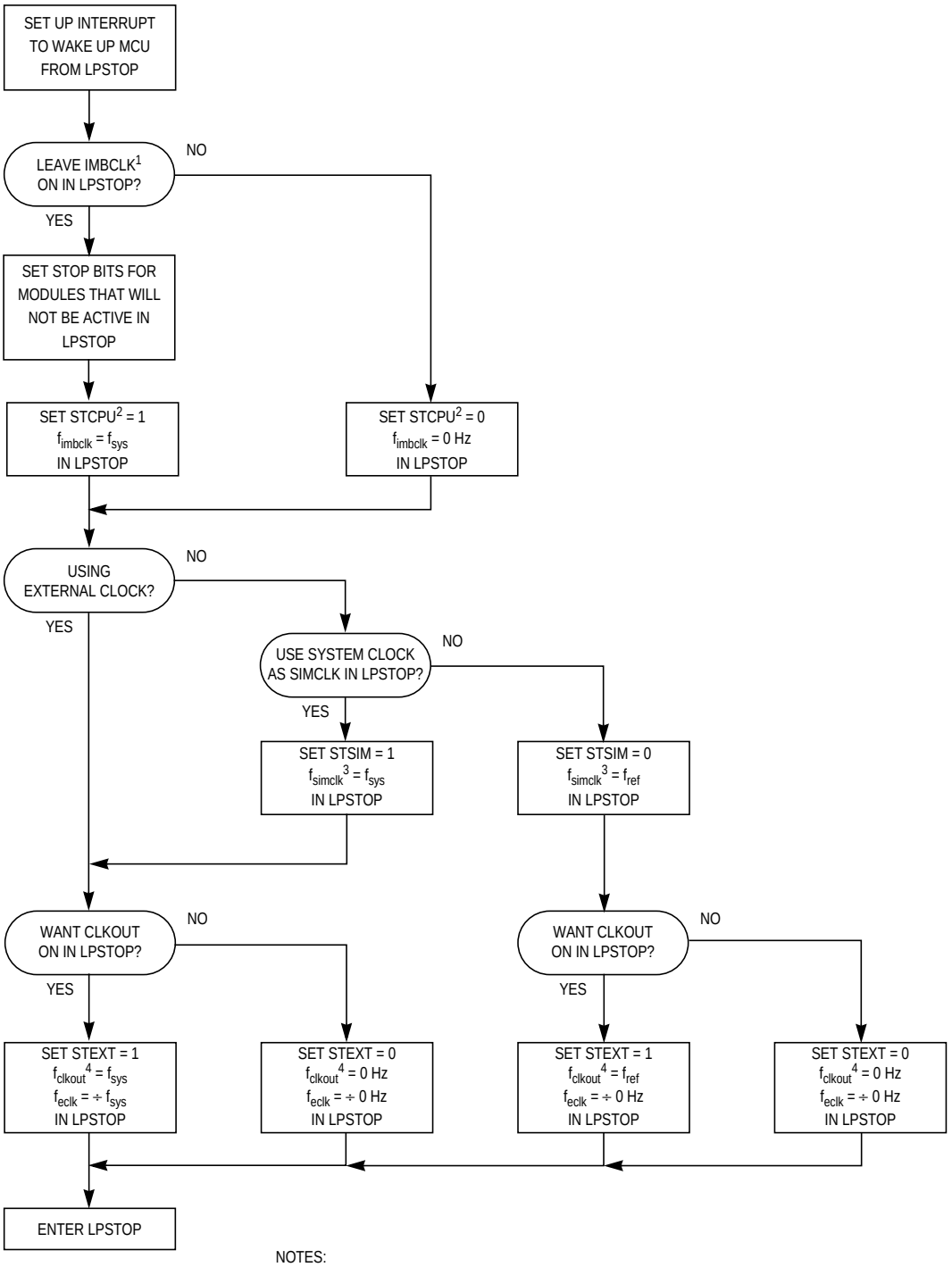
NOTES:
1. THE SIMCLK IS USED BY THE PIT, IRQ, AND INPUT BLOCKS OF THE SIM.
2. CLKOUT CONTROL DURING LPSTOP IS OVERRIDDEN BY THE EXOFF BIT IN SIMCR. IF EXOFF = 1, THE CLKOUT PIN IS ALWAYS IN A HIGH-IMPEDANCE STATE AND STEXT HAS NO EFFECT IN LPSTOP. IF EXOFF = 0, CLKOUT IS CONTROLLED BY STEXT IN LPSTOP.

SIM LPSTOPFLOW

Figure 5-6 SIM LPSTOP Flowchart

Freescale Semiconductor, Inc.

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



NOTES:

1. IMBCLK IS THE CLOCK USED BY THE CPU16L, SIML, ADC, MCCI, AND THE GPT.
2. WHEN STCPU = 1, THE CPU16L IS SHUT DOWN IN LPSTOP. ALL OTHER MODULES WILL REMAIN ACTIVE UNLESS THE STOP BITS IN THEIR MODULE CONFIGURATION REGISTERS ARE SET PRIOR TO ENTERING LPSTOP.
3. THE SIMCLK IS USED BY THE PIT, IRQ, AND INPUT BLOCKS OF THE SIML.
4. CLKOUT CONTROL DURING LPSTOP IS OVERRIDDEN BY THE EXOFF BIT IN SIMCR. IF EXOFF = 1, THE CLKOUT PIN IS ALWAYS IN A HIGH-IMPEDANCE STATE AND STEXT HAS NO EFFECT IN LPSTOP. IF EXOFF = 0, CLKOUT IS CONTROLLED BY STEXT IN LPSTOP. WHEN STCPU = 1, THE CPU16L IS DISABLED IN LPSTOP, BUT ALL OTHER MODULES REMAIN ACTIVE OR STOPPED ACCORDING TO THE SETTING.

SIML LPSTOP FLOWCHART

Figure 5-7 SIML LPSTOP Flowchart

Freescale Semiconductor, Inc.

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

5.4 System Protection

The system protection block preserves reset status, monitors internal activity, and provides periodic interrupt generation. **Figure 5-8** is a block diagram of the submodule.

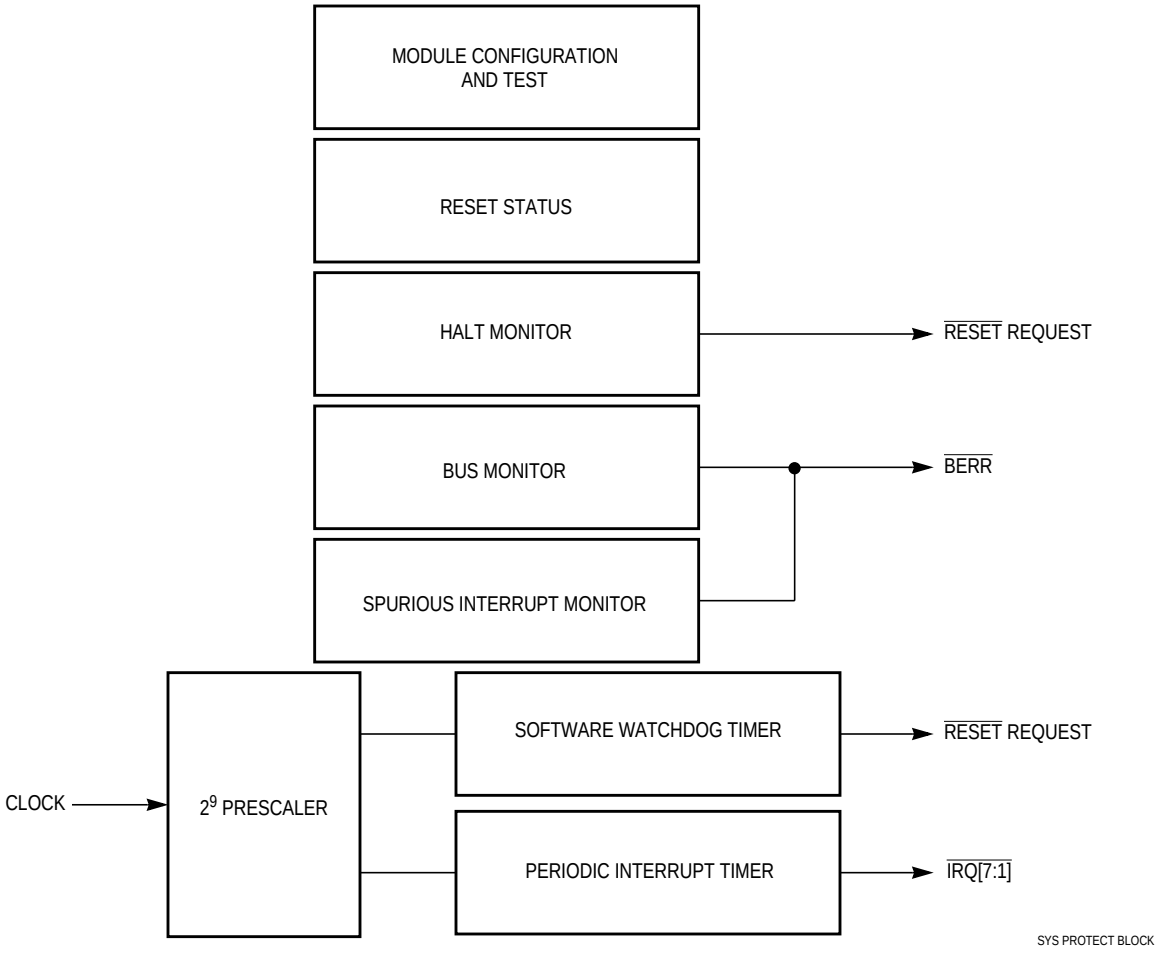


Figure 5-8 System Protection

5.4.1 Reset Status

The reset status register (RSR) latches internal MCU status during reset. Refer to **5.7.10 Reset Status Register** for more information.

5.4.2 Bus Monitor

The internal bus monitor checks data size acknowledge ($\overline{\text{DSACK}}$) or autovector ($\overline{\text{AVEC}}$) signal response times during normal bus cycles. The monitor asserts the internal bus error ($\overline{\text{BERR}}$) signal when the response time is excessively long.

$\overline{\text{DSACK}}$ and $\overline{\text{AVEC}}$ response times are measured in clock cycles. Maximum allowable response time can be selected by setting the bus monitor timing (BMT[1:0]) field in the system protection control register (SYPCR). **Table 5-8** shows the periods allowed.

Freescale Semiconductor, Inc.

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

Table 5-8 Bus Monitor Period

BMT[1:0]	Bus Monitor Time-Out Period
00	64 system clocks
01	32 system clocks
10	16 system clocks
11	8 system clocks

The monitor does not check $\overline{\text{DSACK}}$ response on the external bus unless the CPU16 initiates a bus cycle. The BME bit in SYPCR enables the internal bus monitor for internal to external bus cycles. If a system contains external bus masters, an external bus monitor must be implemented and the internal-to-external bus monitor option must be disabled.

When monitoring transfers to an 8-bit port, the bus monitor does not reset until both byte accesses of a word transfer are completed. Monitor time-out period must be at least twice the number of clocks that a single byte access requires.

5.4.3 Halt Monitor

The halt monitor responds to an assertion of the $\overline{\text{HALT}}$ signal on the internal bus, caused by a double bus fault. A flag in the reset status register (RSR) can indicate that the last reset was caused by the halt monitor. Halt monitor reset can be inhibited by the halt monitor enable (HME) bit in SYPCR. Refer to **5.6.5.2 Double Bus Faults** for more information.

5.4.4 Spurious Interrupt Monitor

During interrupt exception processing, the CPU16 normally acknowledges an interrupt request, arbitrates among various sources of interrupt, recognizes the highest priority source, and then acquires a vector or responds to a request for autovectoring. The spurious interrupt monitor asserts the internal bus error signal ($\overline{\text{BERR}}$) if no interrupt arbitration occurs during interrupt exception processing. The assertion of $\overline{\text{BERR}}$ causes the CPU16 to load the spurious interrupt exception vector into the program counter. The spurious interrupt monitor cannot be disabled. Refer to **5.8 Interrupts** for further information. For detailed information about interrupt exception processing, refer to **4.13 Exceptions**.

5.4.5 Software Watchdog

The software watchdog is controlled by the software watchdog enable (SWE) bit in SYPCR. When enabled, the watchdog requires that a service sequence be written to the software watchdog service register (SWSR) on a periodic basis. If servicing does not take place, the watchdog times out and asserts the $\overline{\text{RESET}}$ signal.

Each time the service sequence is written, the software watchdog timer restarts. The sequence to restart the software watchdog consists of the following steps:

- Write $55 to SWSR.
- Write $AA to SWSR.

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

Both writes must occur before time-out in the order listed. Any number of instructions can be executed between the two writes.

Watchdog clock rate is affected by the software watchdog prescale (SWP) bit and the software watchdog timing (SWT[1:0]) field in SYPCR.

SWP determines system clock prescaling for the watchdog timer and determines that one of two options, either no prescaling or prescaling by a factor of 512, can be selected. The value of SWP is affected by the state of the MODCLK pin during reset, as shown in **Table 5-9**. System software can change SWP value.

Table 5-9 MODCLK Pin and SWP Bit During Reset

MODCLK	SWP
0 (External Clock)	1 (÷ 512)
1 (Internal Clock)	0 (÷ 1)

SWT[1:0] selects the divide ratio used to establish the software watchdog time-out period.

The following equation calculates the time-out period for a slow reference frequency, where f_{ref} is equal to the EXTAL crystal frequency.

$$\text{Time-Out Period} = \frac{\text{Divide Ratio Specified by SWP and SWT[1:0]}}{f_{ref}}$$

The following equation calculates the time-out period for a fast reference frequency, where f_{ref} is equal to the EXTAL crystal frequency.

$$\text{Time-Out Period} = \frac{(128)(\text{Divide Ratio Specified by SWP and SWT[1:0]})}{f_{ref}}$$

The following equation calculates the time-out period for an externally input clock frequency on both slow and fast reference frequency devices, when f_{sys} is equal to the system clock frequency.

$$\text{Time-Out Period} = \frac{\text{Divide Ratio Specified by SWP and SWT[1:0]}}{f_{sys}}$$

Table 5-10 shows the divide ratio for each combination of SWP and SWT[1:0] bits. When SWT[1:0] are modified, a watchdog service sequence must be performed before the new time-out period can take effect.

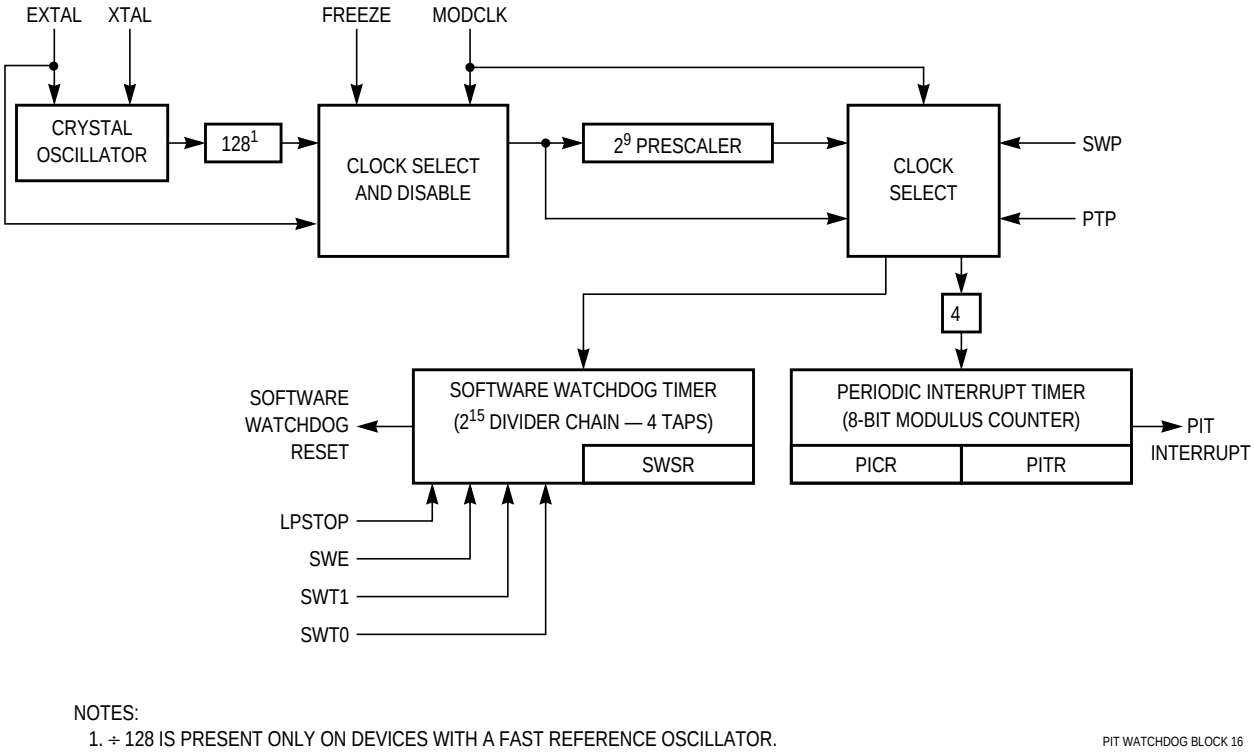
Freescale Semiconductor, Inc.

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

Table 5-10 Software Watchdog Divide Ratio

SWP	SWT[1:0]	Divide Ratio
0	00	2^9
0	01	2^{11}
0	10	2^{13}
0	11	2^{15}
1	00	2^{18}
1	01	2^{20}
1	10	2^{22}
1	11	2^{24}

Figure 5-9 is a block diagram of the watchdog timer and the clock control for the periodic interrupt timer.



NOTES:
1. ÷ 128 IS PRESENT ONLY ON DEVICES WITH A FAST REFERENCE OSCILLATOR.

PIT WATCHDOG BLOCK 16

Figure 5-9 Periodic Interrupt Timer and Software Watchdog Timer

5.4.6 Periodic Interrupt Timer

The periodic interrupt timer (PIT) allows the generation of interrupts of specific priority at predetermined intervals. This capability is often used to schedule control system tasks that must be performed within time constraints. The timer consists of a prescaler, a modulus counter, and registers that determine interrupt timing, priority and vector assignment. Refer to **4.13 Exceptions** for further information about interrupt exception processing.

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

Freescale Semiconductor, Inc.

The periodic interrupt timer modulus counter is clocked by one of two signals. When the PLL is enabled (MODCLK = 1 during reset), f_{ref} is used with a slow reference oscillator; f_{ref} 128 is used with fast reference oscillator. When the PLL is disabled (MODCLK = 0 during reset), f_{ref} is used. The value of the periodic timer prescaler (PTP) bit in the periodic interrupt timer register (PITR) determines system clock prescaling for the periodic interrupt timer. One of two options, either no prescaling, or prescaling by a factor of 512, can be selected. The value of PTP is affected by the state of the MODCLK pin during reset, as shown in **Table 5-11**. System software can change PTP value.

Table 5-11 MODCLK Pin and PTP Bit at Reset

MODCLK	PTP
0 (External Clock)	1 (÷ 512)
1 (Internal Clock)	0 (÷ 1)

Either clock signal selected by the PTP is divided by four before driving the modulus counter. The modulus counter is initialized by writing a value to the periodic interrupt timer modulus (PITM[7:0]) field in PITR. A zero value turns off the periodic timer. When the modulus counter value reaches zero, an interrupt is generated. The modulus counter is then reloaded with the value in PITM[7:0] and counting repeats. If a new value is written to PITR, it is loaded into the modulus counter when the current count is completed.

The following equation calculates the PIT period when a slow reference frequency is used:

$$\text{PIT Period} = \frac{(\text{PITM[7:0]})(1 \text{ if PTP} = 0, 512 \text{ if PTP} = 1)(4)}{f_{ref}}$$

The following equation calculates the PIT period when a fast reference frequency is used:

$$\text{PIT Period} = \frac{(128)(\text{PITM[7:0]})(1 \text{ if PTP} = 0, 512 \text{ if PTP} = 1)(4)}{f_{ref}}$$

The following equation calculates the PIT period for an externally input clock frequency on both slow and fast reference frequency devices.

$$\text{PIT Period} = \frac{(\text{PITM[7:0]})(1 \text{ if PTP} = 0, 512 \text{ if PTP} = 1)(4)}{f_{sys}}$$

5.4.7 Interrupt Priority and Vectoring

Interrupt priority and vectoring are determined by the values of the periodic interrupt request level (PIRQL[2:0]) and periodic interrupt vector (PIV) fields in the periodic interrupt control register (PICR).

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

The PIRQL field is compared to the CPU16 interrupt priority mask to determine whether the interrupt is recognized. **Table 5-12** shows PIRQL[2:0] priority values. Because of SIM hardware prioritization, a PIT interrupt is serviced before an external interrupt request of the same priority. The periodic timer continues to run when the interrupt is disabled.

Table 5-12 Periodic Interrupt Priority

PIRQL[2:0]	Priority Level
000	Periodic Interrupt Disabled
001	Interrupt priority level 1
010	Interrupt priority level 2
011	Interrupt priority level 3
100	Interrupt priority level 4
101	Interrupt priority level 5
110	Interrupt priority level 6
111	Interrupt priority level 7

The PIV field contains the periodic interrupt vector. The vector is placed on the IMB when an interrupt request is made. The vector number is used to calculate the address of the appropriate exception vector in the exception vector table. The reset value of the PIV field is $0F, which corresponds to the uninitialized interrupt exception vector.

5.4.8 Low-Power STOP Operation

When the CPU16 executes the LPSTOP instruction, the current interrupt priority mask is stored in the clock control logic, internal clocks are disabled according to the state of the STSIM bit in the SYNCR, and the MCU enters low-power stop mode. The bus monitor, halt monitor, and spurious interrupt monitor are all inactive during low-power stop.

During low-power stop mode, the clock input to the software watchdog timer is disabled and the timer stops. The software watchdog begins to run again on the first rising clock edge after low-power stop mode ends. The watchdog is not reset by low-power stop mode. A service sequence must be performed to reset the timer.

The periodic interrupt timer does not respond to the LPSTOP instruction, but continues to run during LPSTOP. To stop the periodic interrupt timer, PITR must be loaded with a zero value before the LPSTOP instruction is executed. A PIT interrupt, or an external interrupt request, can bring the MCU out of the low-power stop mode if it has a higher priority than the interrupt mask value stored in the clock control logic when low-power stop mode is initiated. LPSTOP can be terminated by a reset.

5.5 External Bus Interface

The external bus interface (EBI) transfers information between the internal MCU bus and external devices. **Figure 5-10** shows a basic system with external memory and peripherals.

Freescale Semiconductor, Inc.

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

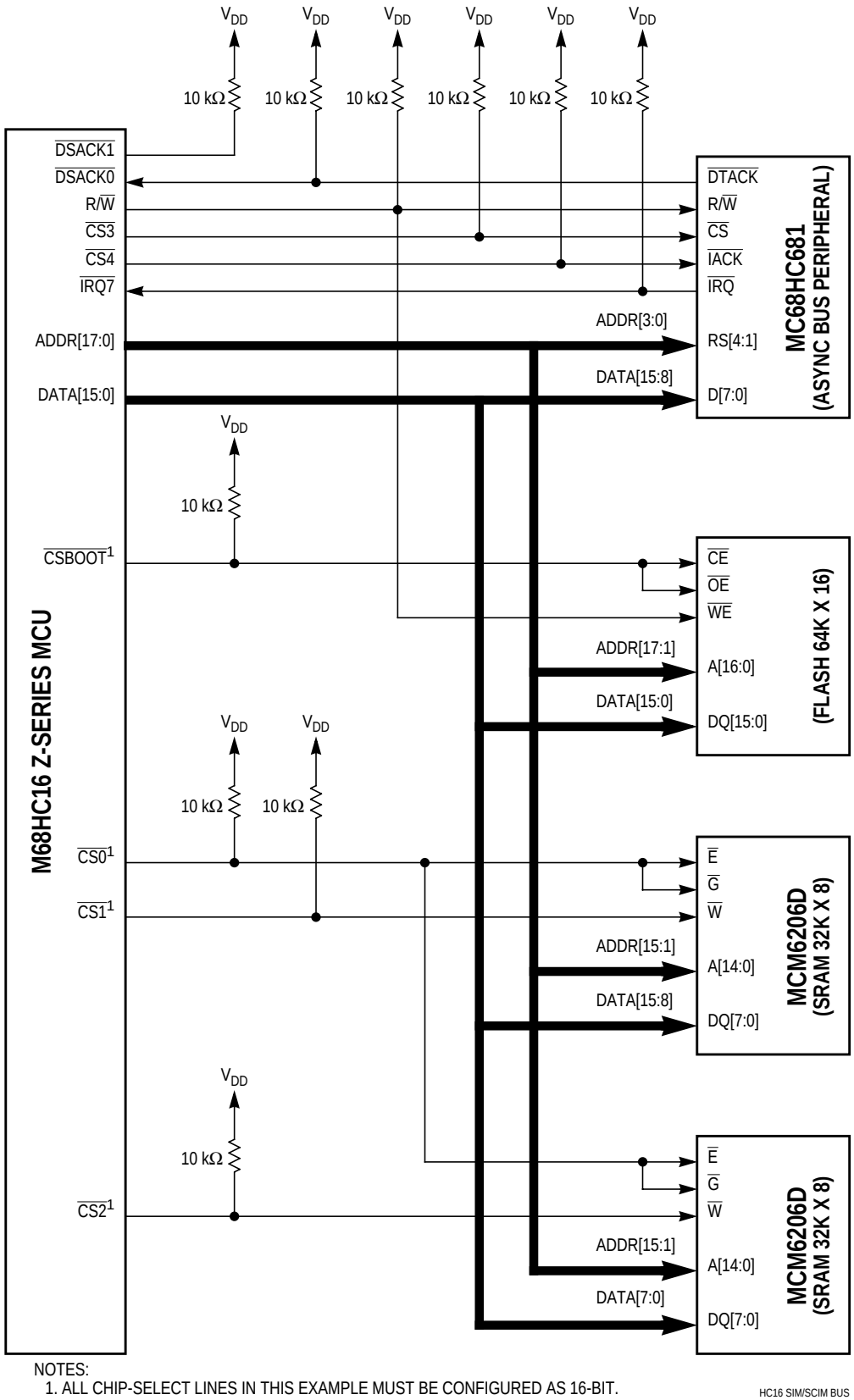


Figure 5-10 MCU Basic System

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

Freescale Semiconductor, Inc.

The external bus has 24 address lines and 16 data lines. ADDR[19:0] are normal address outputs; ADDR[23:20] follow the output state of ADDR19. The EBI provides dynamic sizing between 8-bit and 16-bit data accesses. It supports byte, word, and long-word transfers. Port width is the maximum number of bits accepted or provided by the external memory system during a bus transfer. Widths of eight and sixteen bits are accessed through the use of asynchronous cycles controlled by the size (SIZ1 and SIZ0) and data size acknowledge ($\overline{\text{DSACK1}}$ and $\overline{\text{DSACK0}}$) pins. Multiple bus cycles may be required for dynamically sized transfers.

To add flexibility and minimize the necessity for external logic, MCU chip-select logic is synchronized with EBI transfers. Refer to **5.9 Chip-Selects** for more information.

5.5.1 Bus Control Signals

The address bus provides addressing information to external devices. The data bus transfers 8-bit and 16-bit data between the MCU and external devices. Strobe signals, one for the address bus and another for the data bus, indicate the validity of an address and provide timing information for data.

Control signals indicate the beginning of each bus cycle, the address space, the size of the transfer, and the type of cycle. External devices decode these signals and respond to transfer data and terminate the bus cycle. The EBI can operate in an asynchronous mode for any port width.

5.5.1.1 Address Bus

Bus signals ADDR[19:0] define the address of the byte (or the most significant byte) to be transferred during a bus cycle. The MCU places the address on the bus at the beginning of a bus cycle. The address is valid while $\overline{\text{AS}}$ is asserted.

5.5.1.2 Address Strobe

Address strobe ($\overline{\text{AS}}$) is a timing signal that indicates the validity of an address on the address bus and of many control signals.

5.5.1.3 Data Bus

Signals DATA[15:0] form a bidirectional, non-multiplexed parallel bus that transfers data to or from the MCU. A read or write operation can transfer eight or sixteen bits of data in one bus cycle. For a write cycle, all sixteen bits of the data bus are driven, regardless of the port width or operand size.

5.5.1.4 Data Strobe

Data strobe ($\overline{\text{DS}}$) is a timing signal. For a read cycle, the MCU asserts $\overline{\text{DS}}$ to signal an external device to place data on the bus. $\overline{\text{DS}}$ is asserted at the same time as $\overline{\text{AS}}$ during a read cycle. For a write cycle, $\overline{\text{DS}}$ signals an external device that data on the bus is valid.

Freescale Semiconductor, Inc.

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

5.5.1.5 Read/Write Signal

The read/write signal (R/$\overline{W}$) determines the direction of the transfer during a bus cycle. This signal changes state, when required, at the beginning of a bus cycle, and is valid while $\overline{AS}$ is asserted. R/$\overline{W}$ only transitions when a write cycle is preceded by a read cycle or vice versa. The signal may remain low for two consecutive write cycles.

5.5.1.6 Size Signals

Size signals (SIZ[1:0]) indicate the number of bytes remaining to be transferred during an operand cycle. They are valid while $\overline{AS}$ is asserted. **Table 5-13** shows SIZ0 and SIZ1 encoding.

Table 5-13 Size Signal Encoding

SIZ1	SIZ0	Transfer Size
0	1	Byte
1	0	Word
1	1	3 Byte
0	0	Long Word

5.5.1.7 Function Codes

The CPU generates function code signals (FC[2:0]) to indicate the type of activity occurring on the data or address bus. These signals can be considered address extensions that can be externally decoded to determine which of eight external address spaces is accessed during a bus cycle.

Because the CPU16 always operates in supervisor mode (FC2 = 1), address spaces 0 to 3 are not used. Address space 7 is designated CPU space. CPU space is used for control information not normally associated with read or write bus cycles. Function codes are valid while $\overline{AS}$ is asserted. **Table 5-14** shows address space encoding.

Table 5-14 Address Space Encoding

FC2	FC1	FC0	Address Space
1	0	0	Reserved
1	0	1	Data space
1	1	0	Program space
1	1	1	CPU space

5.5.1.8 Data Size Acknowledge Signals

During normal bus transfers, external devices assert the data size acknowledge signals ($\overline{DSACK[1:0]}$) to indicate port width to the MCU. During a read cycle, these signals tell the MCU to terminate the bus cycle and to latch data. During a write cycle, the signals indicate that an external device has successfully stored data and that the cycle can terminate. $\overline{DSACK[1:0]}$ can also be supplied internally by chip-select logic. Refer to **5.9 Chip-Selects** for more information.

Freescale Semiconductor, Inc.

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

5.5.1.9 Bus Error Signal

The bus error signal (BERR) is asserted when a bus cycle is not properly terminated by DSACK or AVEC assertion. It can also be asserted in conjunction with DSACK to indicate a bus error condition, provided it meets the appropriate timing requirements. Refer to **5.6.5 Bus Exception Control Cycles** for more information.

The internal bus monitor can generate the BERR signal for internal-to-internal and internal-to-external transfers. In systems with an external bus master, the SIM bus monitor must be disabled and external logic must be provided to drive the BERR pin, because the internal BERR monitor has no information about transfers initiated by an external bus master. Refer to **5.6.6 External Bus Arbitration** for more information.

5.5.1.10 Halt Signal

The halt signal (HALT) can be asserted by an external device for debugging purposes to cause single bus cycle operation or (in combination with BERR) a retry of a bus cycle in error. The HALT signal affects external bus cycles only. As a result, a program not requiring use of the external bus may continue executing, unaffected by the HALT signal. When the MCU completes a bus cycle with the HALT signal asserted, DATA[15:0] is placed in a high-impedance state and bus control signals are driven inactive; the address, function code, size, and read/write signals remain in the same state. If HALT is still asserted once bus mastership is returned to the MCU, the address, function code, size, and read/write signals are again driven to their previous states. The MCU does not service interrupt requests while it is halted. Refer to **5.6.5 Bus Exception Control Cycles** for further information.

5.5.1.11 Autovector Signal

The autovector signal (AVEC) can be used to terminate external interrupt acknowledgment cycles. Assertion of AVEC causes the CPU16 to generate vector numbers to locate an interrupt handler routine. If AVEC is continuously asserted, autovectors are generated for all external interrupt requests. AVEC is ignored during all other bus cycles. Refer to **5.8 Interrupts** for more information. AVEC for external interrupt requests can also be supplied internally by chip-select logic. Refer to **5.9 Chip-Selects** for more information. The autovector function is disabled when there is an external bus master. Refer to **5.6.6 External Bus Arbitration** for more information.

5.5.2 Dynamic Bus Sizing

The MCU dynamically interprets the port size of an addressed device during each bus cycle, allowing operand transfers to or from 8-bit and 16-bit ports.

During a bus transfer cycle, an external device signals its port size and indicates completion of the bus cycle to the MCU through the use of the DSACK inputs, as shown in **Table 5-15**. Chip-select logic can generate data size acknowledge signals for an external device. Refer to **5.9 Chip-Selects** for more information.

Freescale Semiconductor, Inc.

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

Table 5-15 Effect of $\overline{\text{DSACK}}$ Signals

$\overline{\text{DSACK1}}$	$\overline{\text{DSACK0}}$	Result
1	1	Insert wait states in current bus cycle
1	0	Complete cycle — Data bus port size is eight bits
0	1	Complete cycle — Data bus port size is sixteen bits
0	0	Reserved

If the CPU is executing an instruction that reads a long-word operand from a 16-bit port, the MCU latches the 16 bits of valid data and then runs another bus cycle to obtain the other 16 bits. The operation for an 8-bit port is similar, but requires four read cycles. The addressed device uses the $\overline{\text{DSACK}}$ signals to indicate the port width. For instance, a 16-bit external device always returns $\overline{\text{DSACK}}$ for a 16-bit port (regardless of whether the bus cycle is a byte or word operation).

Dynamic bus sizing requires that the portion of the data bus used for a transfer to or from a particular port size be fixed. A 16-bit port must reside on data bus bits [15:0], and an 8-bit port must reside on data bus bits [15:8]. This minimizes the number of bus cycles needed to transfer data and ensures that the MCU transfers valid data.

The MCU always attempts to transfer the maximum amount of data on all bus cycles. For a word operation, it is assumed that the port is 16 bits wide when the bus cycle begins.

Operand bytes are designated as shown in **Figure 5-11**. OP[0:3] represent the order of access. For instance, OP0 is the most significant byte of a long-word operand, and is accessed first, while OP3, the least significant byte, is accessed last. The two bytes of a word-length operand are OP0 (most significant) and OP1. The single byte of a byte-length operand is OP0.

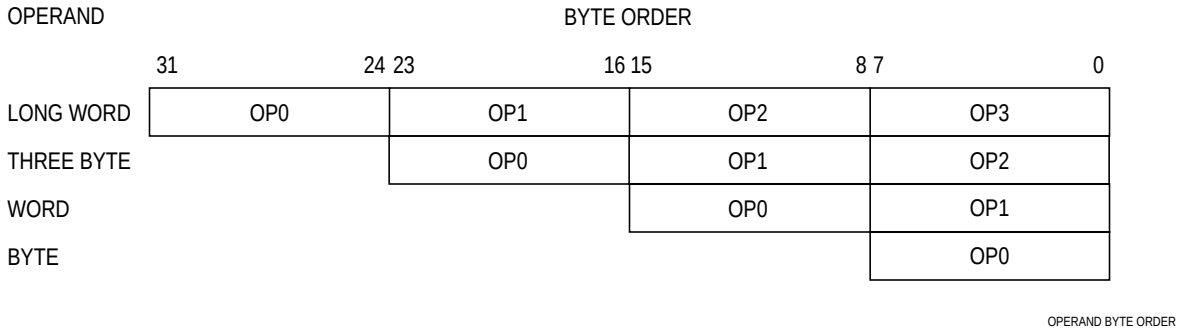


Figure 5-11 Operand Byte Order

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

5.5.3 Operand Alignment

The EBI data multiplexer establishes the necessary connections for different combinations of address and data sizes. The multiplexer takes the two bytes of the 16-bit bus and routes them to their required positions. Positioning of bytes is determined by the size and address outputs. SIZ1 and SIZ0 indicate the number of bytes remaining to be transferred during the current bus cycle. The number of bytes transferred is equal to or less than the size indicated by SIZ1 and SIZ0, depending on port width.

ADDR0 also affects the operation of the data multiplexer. During a bus transfer, ADDR[23:1] indicate the word base address of the portion of the operand to be accessed, and ADDR0 indicates the byte offset from the base.

NOTE

ADDR[23:20] follow the state of ADDR19 in the MCU.

5.5.4 Misaligned Operands

The CPU16 uses a basic operand size of 16 bits. An operand is misaligned when it overlaps a word boundary. This is determined by the value of ADDR0. When ADDR0 = 0 (an even address), the address is on a word and byte boundary. When ADDR0 = 1 (an odd address), the address is on a byte boundary only. A byte operand is aligned at any address; a word or long-word operand is misaligned at an odd address.

The largest amount of data that can be transferred by a single bus cycle is an aligned word. If the MCU transfers a long-word operand through a 16-bit port, the most significant operand word is transferred on the first bus cycle and the least significant operand word is transferred on a following bus cycle.

The CPU16 can perform misaligned word transfers. This capability makes it compatible with the M68HC11 CPU. The CPU16 treats misaligned long-word transfers as two misaligned word transfers.

5.5.5 Operand Transfer Cases

Table 5-16 shows how operands are aligned for various types of transfers. OPn entries are portions of a requested operand that are read or written during a bus cycle and are defined by SIZ1, SIZ0, and ADDR0 for that bus cycle.

Freescale Semiconductor, Inc.

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

Table 5-16 Operand Alignment

Current Cycle	Transfer Case	SIZ1	SIZ0	ADDR0	$\overline{\text{DSACK1}}$	$\overline{\text{DSACK0}}$	DATA [15:8]	DATA [7:0]	Next Cycle
1	Byte to 8-bit port (even)	0	1	0	1	0	OP0	(OP0)[1]	—
2	Byte to 8-bit port (odd)	0	1	1	1	0	OP0	(OP0)	—
3	Byte to 16-bit port (even)	0	1	0	0	1	OP0	(OP0)	—
4	Byte to 16-bit port (odd)	0	1	1	0	1	(OP0)	OP0	—
5	Word to 8-bit port (aligned)	1	0	0	1	0	OP0	(OP1)	2
6	Word to 8-bit port (misaligned)	1	0	1	1	0	OP0	(OP0)	1
7	Word to 16-bit port (aligned)	1	0	0	0	1	OP0	OP1	—
8	Word to 16-bit port (misaligned)	1	0	1	0	1	(OP0)	OP0	3
9	Long word to 8-bit port (aligned)	0	0	0	1	0	OP0	(OP1)	13
10	Long word to 8-bit port (misaligned)[2]	1	0	1	1	0	OP0	(OP0)	1
11	Long word to 16-bit port (aligned)	0	0	0	0	1	OP0	OP1	7
12	Long word to 16-bit port (misaligned)[2]	1	0	1	0	1	(OP0)	OP0	3
13	Three byte to 8-bit port[3]	1	1	1	1	0	OP0	(OP0)	5

NOTES:
1. Operands in parentheses are ignored by the CPU16 during read cycles.
2. The CPU16 treats misaligned long-word transfers as two misaligned-word transfers.
3. Three byte transfer cases occur only as a result of an aligned long word to 8-bit port transfer.

5.6 Bus Operation

Internal microcontroller modules are typically accessed in two system clock cycles. Regular external bus cycles use handshaking between the MCU and external peripherals to manage transfer size and data. These accesses take three system clock cycles, with no wait states. During regular cycles, wait states can be inserted as needed by bus control logic. Refer to **5.6.2 Regular Bus Cycle** for more information.

Fast-termination cycles, which are two-cycle external accesses with no wait states, use chip-select logic to generate handshaking signals internally. Refer to **5.6.3 Fast Termination Cycles** and **5.9 Chip-Selects** for more information. Bus control signal timing, as well as chip-select signal timing, are specified in **APPENDIX A ELECTRICAL CHARACTERISTICS**. Refer to the *SIM Reference Manual* (SIMRM/AD) for more information about each type of bus cycle.

5.6.1 Synchronization to CLKOUT

External devices connected to the MCU bus can operate at a clock frequency different from the frequencies of the MCU as long as the external devices satisfy the interface signal timing constraints. Although bus cycles are classified as asynchronous, they are interpreted relative to the MCU system clock output (CLKOUT).

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

Descriptions are made in terms of individual system clock states, labelled {S0, S1, S2,..., SN}. The designation “state” refers to the logic level of the clock signal, and does not correspond to any implemented machine state. A clock cycle consists of two successive states. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for more information on clock control timing.

Bus cycles terminated by $\overline{\text{DSACK}}$ assertion normally require a minimum of three CLKOUT cycles. To support systems that use CLKOUT to generate $\overline{\text{DSACK}}$ and other inputs, asynchronous input setup time and asynchronous input hold times are specified. When these specifications are met, the MCU is guaranteed to recognize the appropriate signal on a specific edge of the CLKOUT signal.

5.6.2 Regular Bus Cycle

The following paragraphs contain a discussion of cycles that use external bus control logic. Refer to **5.6.3 Fast Termination Cycles** for information about fast termination cycles.

To initiate a transfer, the MCU asserts an address and the SIZ[1:0] signals. The SIZ signals and ADDR0 are externally decoded to select the active portion of the data bus. Refer to **5.5.2 Dynamic Bus Sizing**. When $\overline{\text{AS}}$, $\overline{\text{DS}}$, and R/$\overline{\text{W}}$ are valid, a peripheral device either places data on the bus (read cycle) or latches data from the bus (write cycle), then asserts a $\overline{\text{DSACK}}$[1:0] combination that indicates port size.

The $\overline{\text{DSACK}}$[1:0] signals can be asserted before the data from a peripheral device is valid on a read cycle. To ensure valid data is latched into the MCU, a maximum period between $\overline{\text{DSACK}}$ assertion and $\overline{\text{DS}}$ assertion is specified.

There is no specified maximum for the period between the assertion of $\overline{\text{AS}}$ and $\overline{\text{DSACK}}$. Although the MCU can transfer data in a minimum of three clock cycles when the cycle is terminated with $\overline{\text{DSACK}}$, the MCU inserts wait cycles in clock period increments until either $\overline{\text{DSACK}}$ signal goes low.

If bus termination signals remain unasserted, the MCU will continue to insert wait states, and the bus cycle will never end. If no peripheral responds to an access, or if an access is invalid, external logic should assert the $\overline{\text{BERR}}$ or $\overline{\text{HALT}}$ signals to abort the bus cycle (when $\overline{\text{BERR}}$ and $\overline{\text{HALT}}$ are asserted simultaneously, the CPU16 acts as though only $\overline{\text{BERR}}$ is asserted). When enabled, the SIM bus monitor asserts $\overline{\text{BERR}}$ when $\overline{\text{DSACK}}$ response time exceeds a predetermined limit. The bus monitor time-out period is determined by the BMT[1:0] field in SYPCR. The maximum bus monitor time-out period is 64 system clock cycles.

5.6.2.1 Read Cycle

During a read cycle, the MCU transfers data from an external memory or peripheral device. If the instruction specifies a long-word or word operation, the MCU attempts to read two bytes at once. For a byte operation, the MCU reads one byte. The portion of the data bus from which each byte is read depends on operand size, peripheral address, and peripheral port size. **Figure 5-12** is a flowchart of a word read cycle. Refer to **5.5.2 Dynamic Bus Sizing**, **5.5.4 Misaligned Operands**, and the *SIM Reference Manual* (SIMRM/AD) for more information.

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

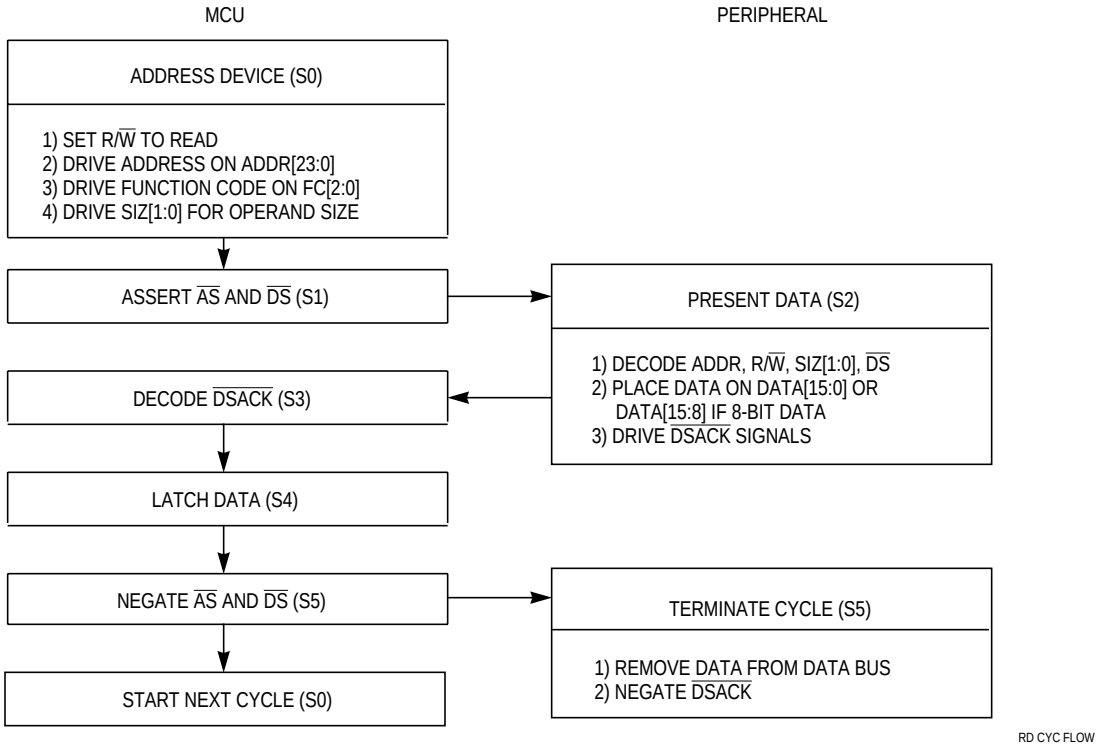


Figure 5-12 Word Read Cycle Flowchart

5.6.2.2 Write Cycle

During a write cycle, the MCU transfers data to an external memory or peripheral device. If the instruction specifies a long-word or word operation, the MCU attempts to write two bytes at once. For a byte operation, the MCU writes one byte. The portion of the data bus upon which each byte is written depends on operand size, peripheral address, and peripheral port size.

Refer to **5.5.2 Dynamic Bus Sizing** and **5.5.4 Misaligned Operands** for more information. **Figure 5-13** is a flowchart of a write-cycle operation for a word transfer. Refer to the *SIM Reference Manual* (SIMRM/AD) for more information.

Freescale Semiconductor, Inc.

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

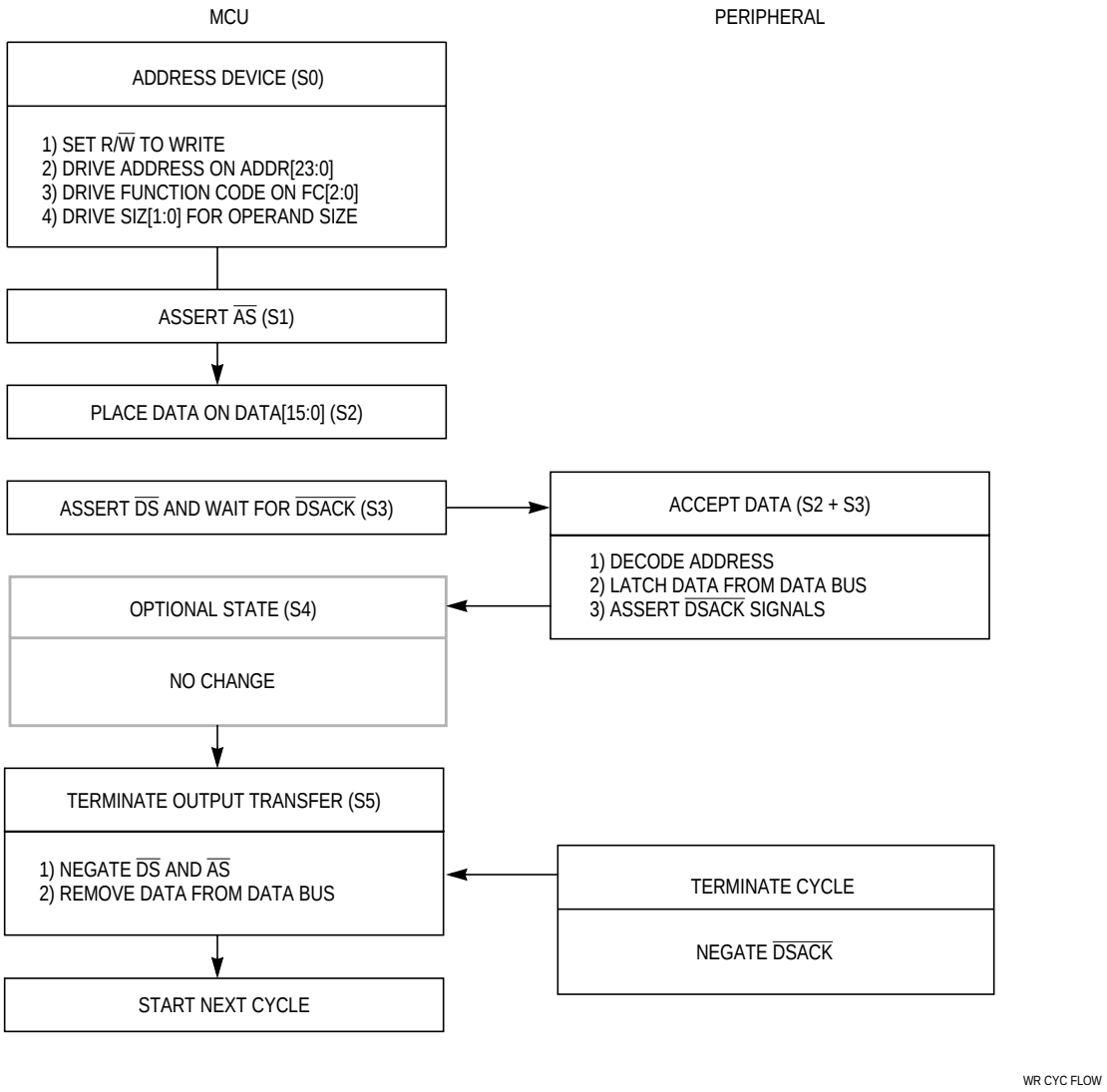


Figure 5-13 Write Cycle Flowchart

5.6.3 Fast Termination Cycles

When an external device can meet fast access timing, an internal chip-select circuit fast termination option can provide a two-cycle external bus transfer. Because the chip-select circuits are driven from the system clock, the bus cycle termination is inherently synchronized with the system clock.

If multiple chip-selects are to be used to provide control signals to a single device and match conditions occur simultaneously, all MODE, STRB, and associated DSACK fields must be programmed to the same value. This prevents a conflict on the internal bus when the wait states are loaded into the DSACK counter shared by all chip-selects.

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

Freescale Semiconductor, Inc.

Fast termination cycles use internal handshaking signals generated by the chip-select logic. To initiate a transfer, the MCU asserts an address and the SIZ[1:0] signals. When AS, DS, and R/W are valid, a peripheral device either places data on the bus (read cycle) or latches data from the bus (write cycle). At the appropriate time, chip-select logic asserts data size acknowledge signals.

The DSACK option fields in the chip-select option registers determine whether internally generated DSACK or externally generated DSACK is used. The external DSACK lines are always active, regardless of the setting of the DSACK field in the chip-select option registers. Thus, an external DSACK can always terminate a bus cycle. Holding a DSACK line low will cause essentially all external bus cycles to be three-cycle (zero wait states) accesses unless the chip-select option register specifies fast accesses.

NOTE

There are certain exceptions to the three-cycle rule when one or both DSACK lines are asserted. Check the current device and mask set errata for details.

For fast termination cycles, the fast termination encoding (%1110) must be used. Refer to **5.9.1 Chip-Select Registers** for information about fast termination setup.

To use fast termination, an external device must be fast enough to have data ready within the specified setup time (for example, by the falling edge of S4). Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for information about fast termination timing.

When fast termination is in use, DS is asserted during read cycles but not during write cycles. The STRB field in the chip-select option register used must be programmed with the address strobe encoding to assert the chip-select signal for a fast termination write.

5.6.4 CPU Space Cycles

Function code signals FC[2:0] designate which of eight external address spaces is accessed during a bus cycle. Address space 7 is designated CPU space. CPU space is used for control information not normally associated with read or write bus cycles. Function codes are valid only while AS is asserted. Refer to **5.5.1.7 Function Codes** for more information on codes and encoding.

During a CPU space access, ADDR[19:16] are encoded to reflect the type of access being made. Three encodings are used by the MCU, as shown in **Figure 5-14**. These encodings represent breakpoint acknowledge (type $0) cycles, low power stop broadcast (type $3) cycles, and interrupt acknowledge (type $F) cycles. Type $0 and type $3 cycles are discussed in the following paragraphs. Refer to **5.8 Interrupts** for information about interrupt acknowledge bus cycles.

Freescale Semiconductor, Inc.

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

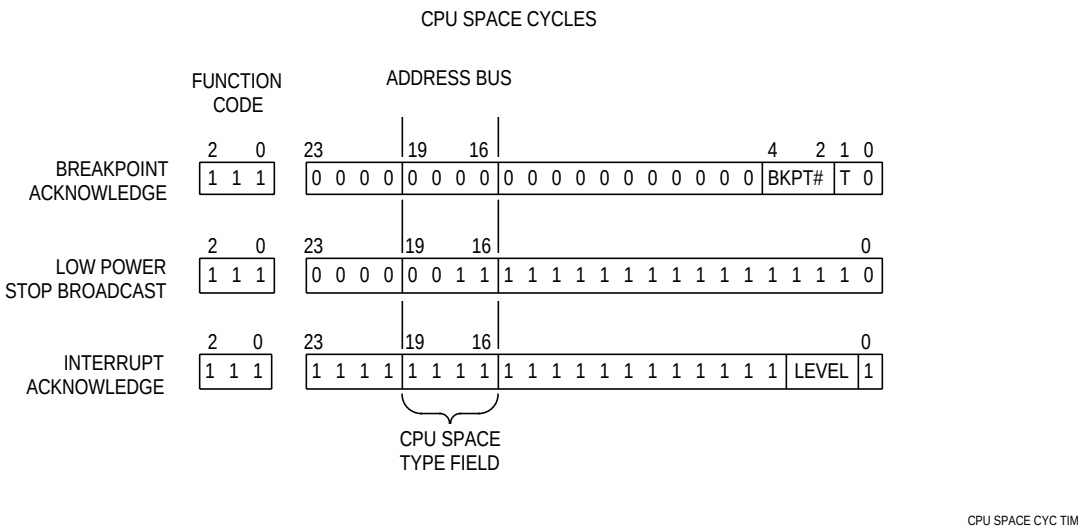


Figure 5-14 CPU Space Address Encoding

5.6.4.1 Breakpoint Acknowledge Cycle

Breakpoints stop program execution at a predefined point during system development. In M68HC16 Z-series MCUs, breakpoints are treated as a type of exception processing. Breakpoints can be used alone or in conjunction with background debug mode.

M68HC16 Z series MCUs have only one source and type of breakpoint. This is a hardware breakpoint initiated by assertion of the $\overline{\text{BKPT}}$ input. Other modular microcontrollers may have more than one source or type. The breakpoint acknowledge cycle discussed here is the bus cycle that occurs as a part of breakpoint exception processing when a breakpoint is initiated while background debug mode is not enabled.

$\overline{\text{BKPT}}$ is sampled on the same clock phase as data. $\overline{\text{BKPT}}$ is valid, the data is tagged as it enters the CPU16 pipeline. When $\overline{\text{BKPT}}$ is asserted while data is valid during an instruction prefetch, the acknowledge cycle occurs immediately after that instruction has executed. When $\overline{\text{BKPT}}$ is asserted while data is valid during an operand fetch, the acknowledge cycle occurs immediately after execution of the instruction during which it is latched. If $\overline{\text{BKPT}}$ is asserted for only one bus cycle and a pipe flush occurs before $\overline{\text{BKPT}}$ is detected by the CPU16, no acknowledge cycle occurs. To ensure detection, $\overline{\text{BKPT}}$ should be asserted until a breakpoint acknowledge cycle is recognized.

When $\overline{\text{BKPT}}$ assertion is acknowledged by the CPU16, the MCU performs a word read from CPU space address $00001E. This corresponds to the breakpoint number field (ADDR[4:2]) and the type bit (T) being set to all ones (source 7, type 1). If this bus cycle is terminated by $\overline{\text{BERR}}$ or by $\overline{\text{DSACK}}$, the MCU performs breakpoint exception processing. Refer to **Figure 5-15** for a flowchart of the breakpoint operation. Refer to the *SIM Reference Manual* (SIMRM/AD) for further information.

Freescale Semiconductor, Inc.

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

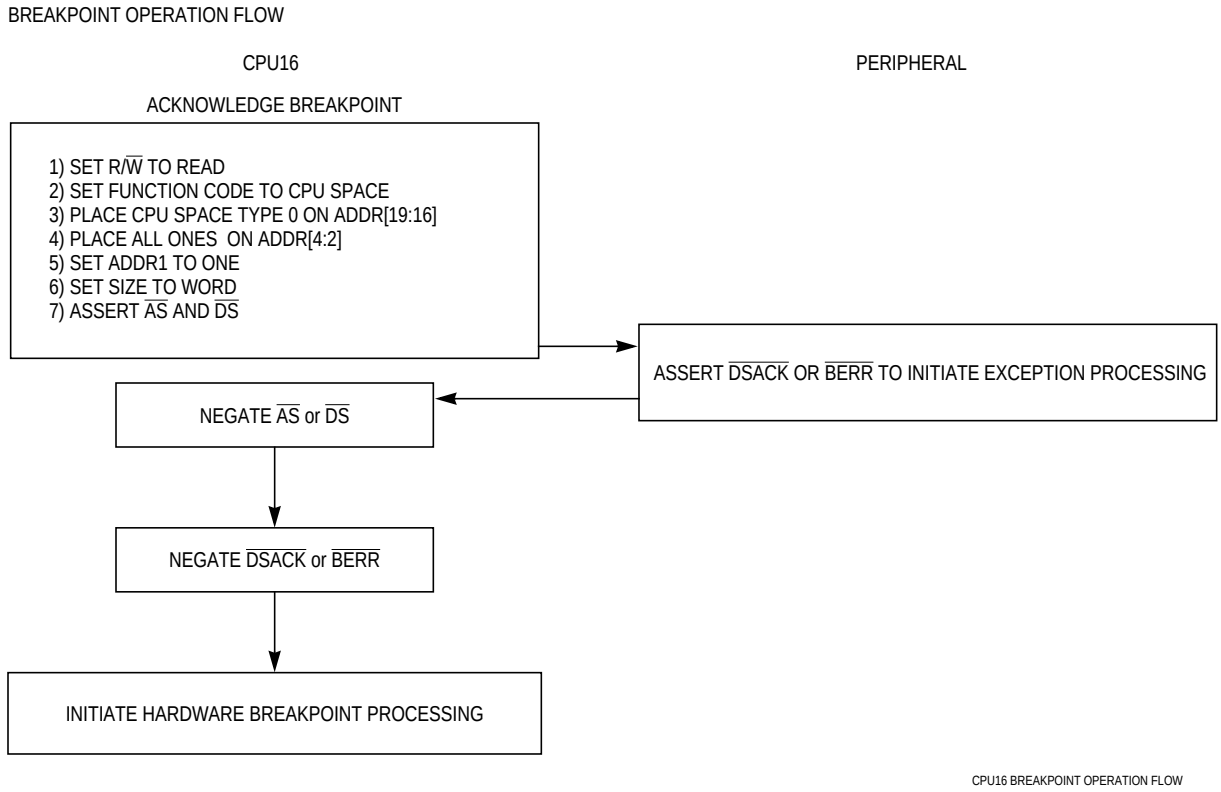


Figure 5-15 Breakpoint Operation Flowchart

5.6.4.2 LPSTOP Broadcast Cycle

Low-power stop mode is initiated by the CPU16. Individual modules can be stopped by setting the STOP bits in each module configuration register. The SIM can turn off system clocks after execution of the LPSTOP instruction. When the CPU16 executes LPSTOP, the LPSTOP broadcast cycle is generated. The SIM brings the MCU out of low-power mode when either an interrupt of higher priority than the interrupt mask level in the CPU16 condition code register or a reset occurs. Refer to **5.3.4 Low-Power Operation** and **SECTION 4 CENTRAL PROCESSOR UNIT** for more information.

During an LPSTOP broadcast cycle, the CPU16 performs a CPU space write to address $3FFFE. This write puts a copy of the interrupt mask value in the clock control logic. The mask is encoded on the data bus as shown in **Figure 5-16**.

The LPSTOP CPU space cycle is shown externally (if the bus is available) as an indication to external devices that the MCU is going into low-power stop mode. The SIM provides an internally generated DSACK response to this cycle. The timing of this bus cycle is the same as for a fast termination write cycle. If the bus is not available (arbitrated away), the LPSTOP broadcast cycle is not shown externally.

NOTE

BERR during the LPSTOP broadcast cycle is ignored.

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

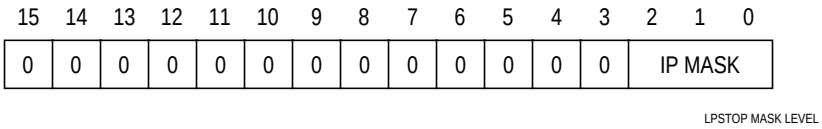


Figure 5-16 LPSTOP Interrupt Mask Level

5.6.5 Bus Exception Control Cycles

An external device or a chip-select circuit must assert at least one of the DSACK[1:0] signals or the AVEC signal to terminate a bus cycle normally. Bus exception control cycles are used when bus cycles are not terminated in the expected manner. There are two sources of bus exception control cycles.

- Bus error signal (BERR)
  - When neither DSACK nor AVEC is asserted within a specified period after assertion of AS, the internal bus monitor asserts internal BERR.
  - The spurious interrupt monitor asserts internal BERR when an interrupt request is acknowledged and no IARB contention occurs. BERR assertion terminates a cycle and causes the MCU to process a bus error exception.
  - External devices can assert BERR to indicate an external bus error.
- Halt signal (HALT)
  - HALT can be asserted by an external device to cause single bus cycle operation. HALT is typically used for debugging purposes.

To control termination of a bus cycle for a bus error condition properly, DSACK, BERR, and HALT must be asserted and negated synchronously with the rising edge of CLKOUT. This ensures that setup time and hold time requirements are met for the same falling edge of the MCU clock when two signals are asserted simultaneously. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for more information. External circuitry that provides these signals must be designed with these constraints in mind, or the internal bus monitor must be used.

Table 5-17 is a summary of the acceptable bus cycle terminations for asynchronous cycles in relation to DSACK assertion.

Freescale Semiconductor, Inc.

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

Table 5-17 $\overline{\text{DSACK}}$, $\overline{\text{BERR}}$, and $\overline{\text{HALT}}$ Assertion Results

Type of Termination	Control Signal	Asserted on Rising Edge of State		Description of Result
		S[1]	S + 2	
NORMAL	$\overline{\text{DSACK}}$ $\overline{\text{BERR}}$ $\overline{\text{HALT}}$	A[2] NA[3] NA	RA[4] NA X[5]	Normal cycle terminate and continue.
HALT	$\overline{\text{DSACK}}$ $\overline{\text{BERR}}$ $\overline{\text{HALT}}$	A NA A/RA	RA NA RA	Normal cycle terminate and halt. Continue when $\overline{\text{HALT}}$ is negated.
BUS ERROR 1	$\overline{\text{DSACK}}$ $\overline{\text{BERR}}$ $\overline{\text{HALT}}$	NA/A A NA	X RA X	Terminate and take bus error exception.
BUS ERROR 2	$\overline{\text{DSACK}}$ $\overline{\text{BERR}}$ $\overline{\text{HALT}}$	A A NA	X RA NA	Terminate and take bus error exception.
BUS ERROR 3	$\overline{\text{DSACK}}$ $\overline{\text{BERR}}$ $\overline{\text{HALT}}$	NA/A A A/S	X RA RA	Terminate and take bus error exception.
BUS ERROR 4	$\overline{\text{DSACK}}$ $\overline{\text{BERR}}$ $\overline{\text{HALT}}$	A NA NA	X A A	Terminate and take bus error exception.

NOTES:
1. S = The number of current even bus state (for example, S2, S4, etc.)
2. A = Signal is asserted in this bus state.
3. NA = Signal is not asserted in this state.
4. RA = Signal was asserted in previous state and remains asserted in this state.
5. X = Don't care

5.6.5.1 Bus Errors

The CPU16 treats bus errors as a type of exception. Bus error exception processing begins when the CPU16 detects assertion of the IMB $\overline{\text{BERR}}$ signal.

$\overline{\text{BERR}}$ assertions do not force immediate exception processing. The signal is synchronized with normal bus cycles and is latched into the CPU16 at the end of the bus cycle in which it was asserted. Because bus cycles can overlap instruction boundaries, bus error exception processing may not occur at the end of the instruction in which the bus cycle begins. Timing of $\overline{\text{BERR}}$ detection/acknowledge is dependent upon several factors:

- Which bus cycle of an instruction is terminated by assertion of $\overline{\text{BERR}}$.
- The number of bus cycles in the instruction during which $\overline{\text{BERR}}$ is asserted.
- The number of bus cycles in the instruction following the instruction in which $\overline{\text{BERR}}$ is asserted.
- Whether $\overline{\text{BERR}}$ is asserted during a program space access or a data space access.

Because of these factors, it is impossible to predict precisely how long after occurrence of a bus error the bus error exception is processed.

Freescale Semiconductor, Inc.

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

NOTE

The external bus interface does not latch data when an external bus cycle is terminated by a bus error. When this occurs during an instruction prefetch, the IMB precharge state (bus pulled high, or $FF) is latched into the CPU16 instruction register, with indeterminate results.

5.6.5.2 Double Bus Faults

Exception processing for bus error exceptions follows the standard exception processing sequence. Refer to **4.13 Exceptions** for more information. However, two special cases of bus error, called double bus faults, can abort exception processing.

$\overline{\text{BERR}}$ assertion is not detected until an instruction is complete. The $\overline{\text{BERR}}$ latch is cleared by the first instruction of the $\overline{\text{BERR}}$ exception handler. Double bus fault occurs in two ways:

1. When bus error exception processing begins, and a second $\overline{\text{BERR}}$ is detected before the first instruction of the exception handler is executed.
2. When one or more bus errors occur before the first instruction after a $\overline{\text{RESET}}$ exception is executed.

Multiple bus errors within a single instruction that can generate multiple bus cycles cause a single bus error exception after the instruction has been executed.

Immediately after assertion of a second $\overline{\text{BERR}}$, the MCU halts and drives the $\overline{\text{HALT}}$ line low. Only a reset can restart a halted MCU. However, bus arbitration can still occur. Refer to **5.6.6 External Bus Arbitration** for more information. A bus error or address error that occurs after exception processing has been completed (during the execution of the exception handler routine, or later) does not cause a double bus fault. The MCU continues to retry the same bus cycle as long as the external hardware requests it.

5.6.5.3 Halt Operation

When $\overline{\text{HALT}}$ is asserted while $\overline{\text{BERR}}$ is not asserted, the MCU halts external bus activity after negation of $\overline{\text{DSACK}}$. The MCU may complete the current word transfer in progress. For a long-word to byte transfer, this could be after S2 or S4. For a word to byte transfer, activity ceases after S2.

Negating and reasserting $\overline{\text{HALT}}$ according to timing requirements provides single-step (bus cycle to bus cycle) operation. The $\overline{\text{HALT}}$ signal affects external bus cycles only, so that a program that does not use external bus can continue executing. During dynamically-sized 8-bit transfers, external bus activity may not stop at the next cycle boundary. Occurrence of a bus error while $\overline{\text{HALT}}$ is asserted causes the CPU16 to process a bus error exception.

When the MCU completes a bus cycle while the $\overline{\text{HALT}}$ signal is asserted, the data bus goes into a high-impedance state and the $\overline{\text{AS}}$ and $\overline{\text{DS}}$ signals are driven to their inactive states. Address, function code, size, and read/write signals remain in the same state.

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

The halt operation has no effect on bus arbitration. However, when external bus arbitration occurs while the MCU is halted, address and control signals go into a high-impedance state. If $\overline{\text{HALT}}$ is still asserted when the MCU regains control of the bus, address, function code, size, and read/write signals revert to the previous driven states. The MCU cannot service interrupt requests while halted.

5.6.6 External Bus Arbitration

The MCU bus design provides for a single bus master at any one time. Either the MCU or an external device can be master. Bus arbitration protocols determine when an external device can become bus master. Bus arbitration requests are recognized during normal processing, $\overline{\text{HALT}}$ assertion, and when the CPU has halted due to a double bus fault.

The bus controller in the MCU manages bus arbitration signals so that the MCU has the lowest priority. External devices that need to obtain the bus must assert bus arbitration signals in the sequences described in the following paragraphs.

Systems that include several devices that can become bus master require external circuitry to assign priorities to the devices, so that when two or more external devices attempt to become bus master at the same time, the one having the highest priority becomes bus master first. The protocol sequence is:

1. An external device asserts the bus request signal ($\overline{\text{BR}}$).
2. The MCU asserts the bus grant signal ($\overline{\text{BG}}$) to indicate that the bus is available.
3. An external device asserts the bus grant acknowledge ($\overline{\text{BGACK}}$) signal to indicate that it has assumed bus mastership.

$\overline{\text{BR}}$ can be asserted during a bus cycle or between cycles. $\overline{\text{BG}}$ is asserted in response to $\overline{\text{BR}}$. To guarantee operand coherency, $\overline{\text{BG}}$ is only asserted at the end of operand transfer.

If more than one external device can be bus master, required external arbitration must begin when a requesting device receives $\overline{\text{BG}}$. An external device must assert $\overline{\text{BGACK}}$ when it assumes mastership, and must maintain $\overline{\text{BGACK}}$ assertion as long as it is bus master.

Two conditions must be met for an external device to assume bus mastership. The device must receive $\overline{\text{BG}}$ through the arbitration process, and $\overline{\text{BGACK}}$ must be inactive, indicating that no other bus master is active. This technique allows the processing of bus requests during data transfer cycles.

$\overline{\text{BG}}$ is negated a few clock cycles after $\overline{\text{BGACK}}$ transition. However, if bus requests are still pending after $\overline{\text{BG}}$ is negated, the MCU asserts $\overline{\text{BG}}$ again within a few clock cycles. This additional $\overline{\text{BG}}$ assertion allows external arbitration circuitry to select the next bus master before the current master has released the bus.

Refer to **Figure 5-17**, which shows bus arbitration for a single device. The flowchart shows $\overline{\text{BR}}$ negated at the same time $\overline{\text{BGACK}}$ is asserted.

Freescale Semiconductor, Inc.

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

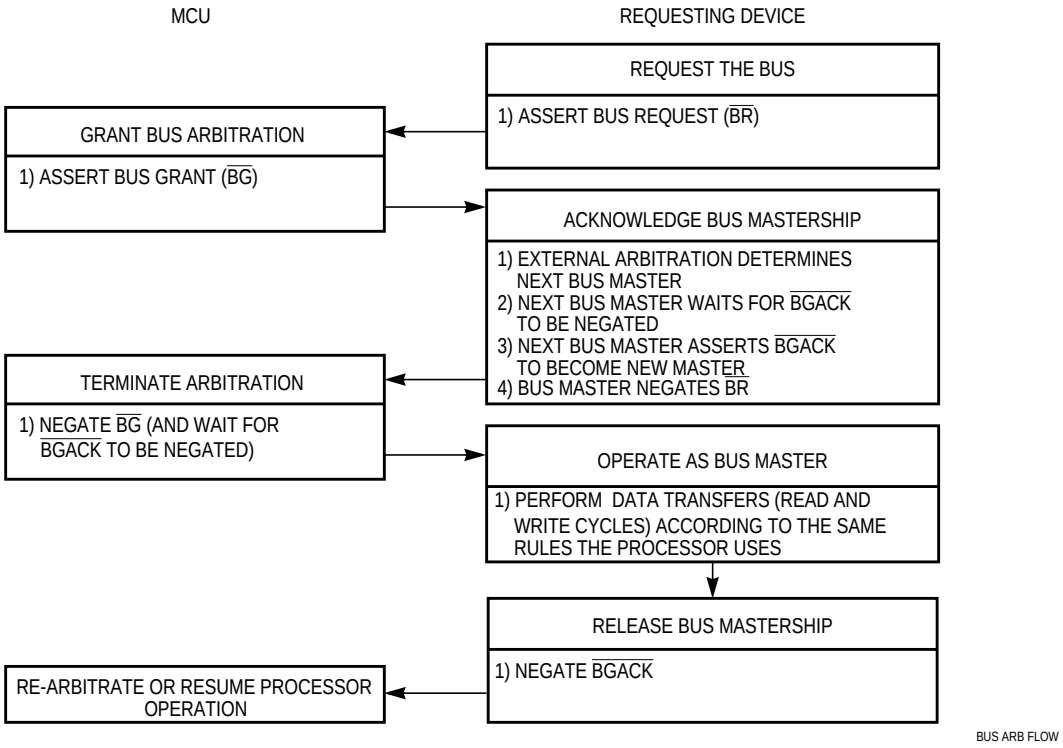


Figure 5-17 Bus Arbitration Flowchart for Single Request

5.6.6.1 Show Cycles

The MCU normally performs internal data transfers without affecting the external bus, but it is possible to show these transfers during debugging. $\overline{AS}$ is not asserted externally during show cycles.

Show cycles are controlled by the SHEN[1:0] in SIMCR. This field is set to %00 by reset. When show cycles are disabled, the address bus, function codes, size, and read/write signals reflect internal bus activity, but $\overline{AS}$ and $\overline{DS}$ are not asserted externally and external data bus pins are in high-impedance state during internal accesses. Refer to **5.2.3 Show Internal Cycles** and the *SIM Reference Manual* (SIMRM/AD) for more information.

When show cycles are enabled, $\overline{DS}$ is asserted externally during internal cycles, and internal data is driven out on the external data bus. Because internal cycles normally continue to run when the external bus is granted, one SHEN[1:0] encoding halts internal bus activity while there is an external master.

SIZ[1:0] signals reflect bus allocation during show cycles. Only the appropriate portion of the data bus is valid during the cycle. During a byte write to an internal address, the portion of the bus that represents the byte that is not written reflects internal bus conditions, and is indeterminate. During a byte write to an external address, the data multiplexer in the SIM causes the value of the byte that is written to be driven out on both bytes of the data bus.

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

5.7 Reset

Reset occurs when an active low logic level on the $\overline{\text{RESET}}$ pin is clocked into the SIM. The $\overline{\text{RESET}}$ input is synchronized to the system clock. If there is no clock when $\overline{\text{RESET}}$ is asserted, reset does not occur until the clock starts. Resets are clocked to allow completion of write cycles in progress at the time the $\overline{\text{RESET}}$ is asserted.

Reset procedures handle system initialization and recovery from catastrophic failure. The MCU performs resets with a combination of hardware and software. The SIM determines whether a reset is valid, asserts control signals, performs basic system configuration and boot ROM selection based on hardware mode-select inputs, then passes control to the CPU16.

5.7.1 Reset Exception Processing

The CPU16 processes resets as a type of asynchronous exception. An exception is an event that preempts normal processing, and can be caused by internal or external events. Exception processing makes the transition from normal instruction execution to execution of a routine that deals with an exception. Each exception has an assigned vector that points to an associated handler routine. These vectors are stored in the exception vector table. The exception vector table consists of 256 four-byte vectors and occupies 512 bytes of address space. The exception vector table can be relocated in memory by changing its base address in the vector base register (VBR). The CPU16 uses vector numbers to calculate displacement into the table. Refer to **4.13 Exceptions** for more information.

Reset is the highest-priority CPU16 exception. Unlike all other exceptions, a reset occurs at the end of a bus cycle, and not at an instruction boundary. Handling resets in this way prevents write cycles in progress at the time the reset signal is asserted from being corrupted. However, any processing in progress is aborted by the reset exception, and cannot be restarted. Only essential reset tasks are performed during exception processing. Other initialization tasks must be accomplished by the exception handler routine. Refer to **5.7.9 Reset Processing Summary** for details on exception processing.

5.7.2 Reset Control Logic

SIM reset control logic determines the cause of a reset, synchronizes request signals to CLKOUT, and asserts reset control signals. Reset control logic can drive three different internal signals.

- EXTRST (external reset) drives the external reset pin.
- CLKRST (clock reset) resets the clock module.
- MSTRST (master reset) goes to all other internal circuits.

All resets are gated by CLKOUT. Asynchronous resets are assumed to be catastrophic. An asynchronous reset can occur on any clock edge. Synchronous resets are timed to occur at the end of bus cycles. The SIM bus monitor is automatically enabled for synchronous resets. When a bus cycle does not terminate normally, the bus monitor terminates it. **Table 5-18** is a summary of reset sources.

Freescale Semiconductor, Inc.

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

Table 5-18 Reset Source Summary

Type	Source	Timing	Cause	Reset Lines Asserted by Controller		
External	External	Synch	$\overline{\text{RESET}}$ pin	MSTRST	CLKRST	EXTRST
Power up	EBI	Asynch	V_{DD}	MSTRST	CLKRST	EXTRST
Software watchdog	Monitor	Asynch	Time out	MSTRST	CLKRST	EXTRST
$\overline{\text{HALT}}$	Monitor	Asynch	Internal $\overline{\text{HALT}}$ assertion (e.g. double bus fault)	MSTRST	CLKRST	EXTRST
Loss of clock	Clock	Synch	Loss of reference	MSTRST	CLKRST	EXTRST
Test	Test	Synch	Test mode	MSTRST	—	EXTRST

Internal single byte or aligned word writes are guaranteed valid for synchronous resets. External writes are also guaranteed to complete, provided the external configuration logic on the data bus is conditioned as shown in **Figure 5-18**.

5.7.3 Reset Mode Selection

The logic states of certain data bus pins during reset determine SIM operating configuration. In addition, the state of the MODCLK pin determines system clock source and the state of the $\overline{\text{BKPT}}$ pin determines what happens during subsequent breakpoint assertions. **Table 5-19** is a summary of reset mode selection options.

Table 5-19 Reset Mode Selection

Mode Select Pin	Default Function (Pin Left High)	Alternate Function (Pin Pulled Low)
DATA0	$\overline{\text{CSBOOT}}$ 16-Bit	$\overline{\text{CSBOOT}}$ 8-Bit
DATA1	$\overline{\text{CS0}}$ $\overline{\text{CS1}}$ $\overline{\text{CS2}}$	$\overline{\text{BR}}$ $\overline{\text{BG}}$ $\overline{\text{BGACK}}$
DATA2	$\overline{\text{CS3}}$ $\overline{\text{CS4}}$ $\overline{\text{CS5}}$	FC0 FC1 FC2
DATA3 DATA4 DATA5 DATA6 DATA7	$\overline{\text{CS6}}$ $\overline{\text{CS}}$[7:6] $\overline{\text{CS}}$[8:6] $\overline{\text{CS}}$[9:6] $\overline{\text{CS}}$[10:6]	ADDR19 ADDR[20:19] ADDR[21:19] ADDR[22:19] ADDR[23:19]
DATA8	$\overline{\text{DSACK}}$[1:0], $\overline{\text{AVEC}}$, $\overline{\text{DS}}$, $\overline{\text{AS}}$, $\overline{\text{SIZ}}$[1:0]	PORTE
DATA9	$\overline{\text{IRQ}}$[7:1] MODCLK	PORTF
DATA11	Normal Operation[1]	Reserved
MODCLK	VCO = System Clock	EXTAL = System Clock
$\overline{\text{BKPT}}$	Background Mode Disabled	Background Mode Enabled

NOTES:
1. DATA11 must remain high during reset to ensure normal operation.

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

5.7.3.1 Data Bus Mode Selection

All data lines have weak internal pull-up devices. When pins are held high by the internal pull-ups, the MCU uses a default operating configuration. However, specific lines can be held low externally during reset to achieve an alternate configuration.

NOTE

External bus loading can overcome the weak internal pull-up drivers on data bus lines and hold pins low during reset.

Use an active device to hold data bus lines low. Data bus configuration logic must release the bus before the first bus cycle after reset to prevent conflict with external memory devices. The first bus cycle occurs ten CLKOUT cycles after $\overline{\text{RESET}}$ is released. If external mode selection logic causes a conflict of this type, an isolation resistor on the driven lines may be required. **Figure 5-18** shows a recommended method for conditioning the mode select signals.

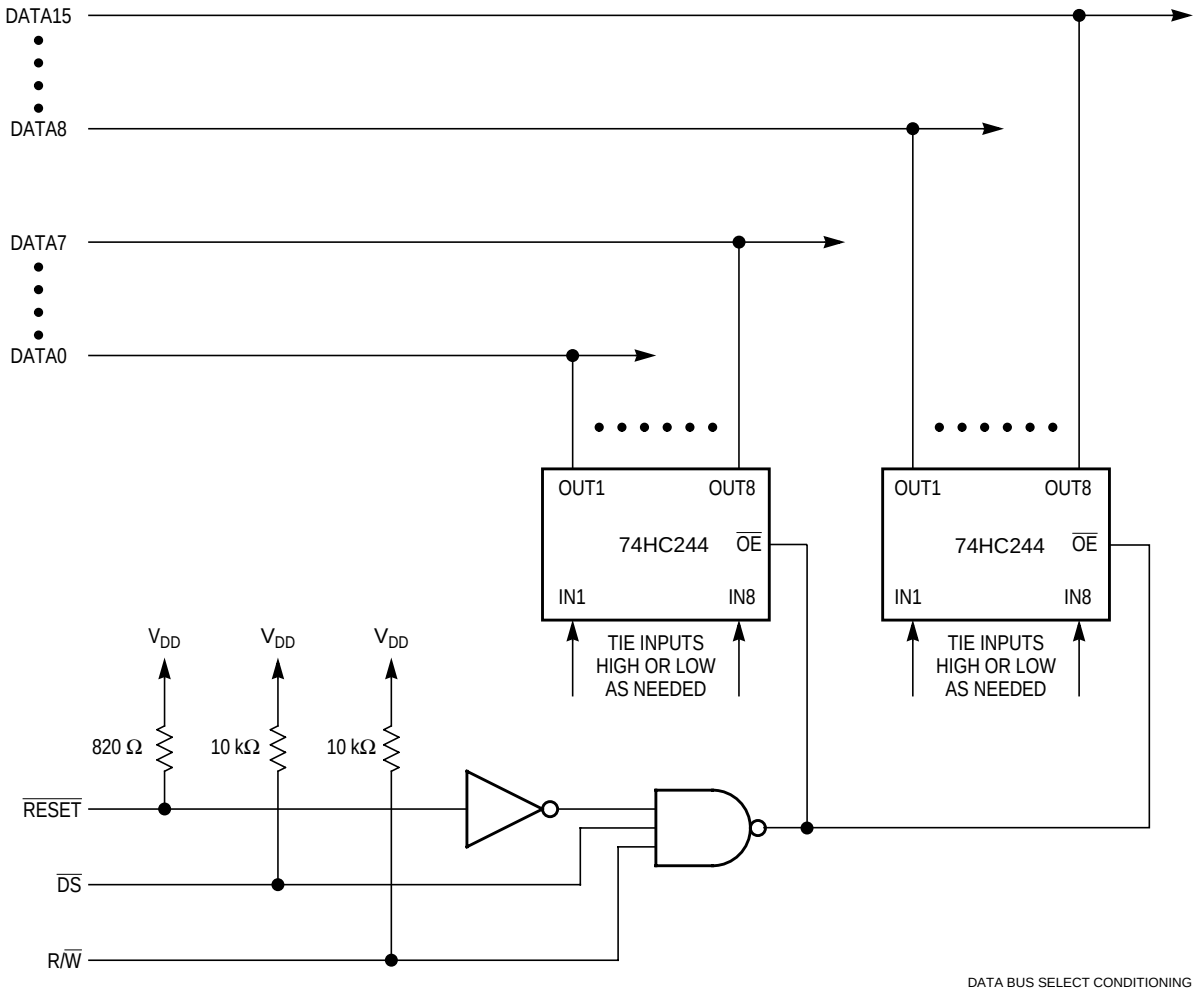


Figure 5-18 Preferred Circuit for Data Bus Mode Select Conditioning

Freescale Semiconductor, Inc.

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

The mode configuration drivers are conditioned with R/$\overline{W}$ and $\overline{DS}$ to prevent conflicts between external devices and the MCU when reset is asserted. If external $\overline{RESET}$ is asserted during an external write cycle, R/$\overline{W}$ conditioning (as shown in **Figure 5-18**) prevents corruption of the data during the write. Similarly, $\overline{DS}$ conditions the mode configuration drivers so that external reads are not corrupted when $\overline{RESET}$ is asserted during an external read cycle.

Alternate methods can be used for driving data bus pins low during reset. **Figure 5-19** shows two of these options. The simplest is to connect a resistor in series with a diode from the data bus pin to the $\overline{RESET}$ line. A bipolar transistor can be used for the same purpose, but an additional current limiting resistor must be connected between the base of the transistor and the $\overline{RESET}$ pin. If a MOSFET is substituted for the bipolar transistor, only the 1 kΩ isolation resistor is required. These simpler circuits do not offer the protection from potential memory corruption during $\overline{RESET}$ assertion as does the circuit shown in **Figure 5-18**.

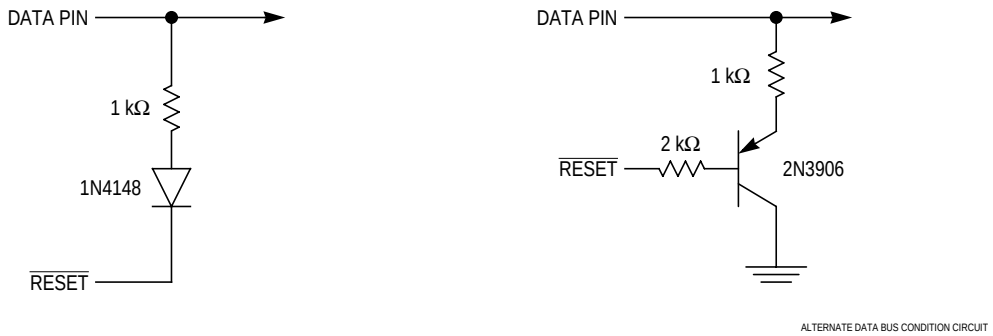


Figure 5-19 Alternate Circuit for Data Bus Mode Select Conditioning

Data bus mode select current is specified in **APPENDIX A ELECTRICAL CHARACTERISTICS**. Do not confuse pin function with pin electrical state. Refer to **5.7.5 Pin State During Reset** for more information.

Unlike other chip-select signals, the boot ROM chip-select ($\overline{CSBOOT}$) is active at the release of $\overline{RESET}$. During reset exception processing, the MCU fetches initialization vectors beginning at address $000000 in supervisor program space. An external memory device containing vectors located at these addresses can be enabled by $\overline{CSBOOT}$ after a reset.

The logic level of DATA0 during reset selects boot ROM port size for dynamic bus allocation. When DATA0 is held low, port size is eight bits; when DATA0 is held high, either by the weak internal pull-up driver or by an external pull-up, port size is 16 bits. Refer to **5.9.4 Chip-Select Reset Operation** for more information.

DATA1 and DATA2 determine the functions of $\overline{CS[2:0]}$ and $\overline{CS[5:3]}$, respectively. DATA[7:3] determine the functions of an associated chip-select and all lower-numbered chip-selects down through $\overline{CS6}$. For example, if DATA5 is pulled low during reset, $\overline{CS[8:6]}$ are assigned alternate function as ADDR[21:19], and $\overline{CS[10:9]}$ remain chip-selects. Refer to **5.9.4 Chip-Select Reset Operation** for more information.

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

DATA8 determines the function of the DSACK[1:0], AVEC, DS, AS, and SIZE pins. If DATA8 is held low during reset, these pins are assigned to I/O port E.

DATA9 determines the function of interrupt request pins IRQ[7:1] and the clock mode select pin (MODCLK). When DATA9 is held low during reset, these pins are assigned to I/O port F.

5.7.3.2 Clock Mode Selection

The state of the clock mode (MODCLK) pin during reset determines what clock source the MCU uses. When MODCLK is held high during reset, the clock signal is generated from a reference frequency using the clock synthesizer. When MODCLK is held low during reset, the clock synthesizer is disabled, and an external system clock signal must be applied. Refer to **5.3 System Clock** for more information.

NOTE

The MODCLK pin can also be used as parallel I/O pin PF0. To prevent inadvertent clock mode selection by logic connected to port F, use an active device to drive MODCLK during reset.

5.7.3.3 Breakpoint Mode Selection

Background debug mode (BDM) is enabled when the breakpoint (BKPT) pin is sampled at a logic level zero at the release of RESET. Subsequent assertion of the BKPT pin or the internal breakpoint signal (for instance, the execution of the CPU16 BKPT instruction) will place the CPU16 in BDM.

If BKPT is sampled at a logic level one at the rising edge of RESET, BDM is disabled. Assertion of the BKPT pin or execution of the BKPT instruction will result in normal breakpoint exception processing.

BDM remains enabled until the next system reset. BKPT is relatched on each rising transition of RESET. BKPT is internally synchronized and must be held low for at least two clock cycles prior to RESET negation for BDM to be enabled. BKPT assertion logic must be designed with special care. If BKPT assertion extends into the first bus cycle following the release of RESET, the bus cycle could inadvertently be tagged with a breakpoint.

Refer to **4.14.4 Background Debug Mode** and the *CPU16 Reference Manual* (CPU16RM/AD) for more information on background debug mode. Refer to the *SIM Reference Manual* (SIMRM/AD) and **APPENDIX A ELECTRICAL CHARACTERISTICS** for more information concerning BKPT signal timing.

5.7.4 MCU Module Pin Function During Reset

Usually, module pins default to port functions and input/output ports are set to input state. This is accomplished by disabling pin functions in the appropriate control registers and by clearing the appropriate port data direction registers. Refer to individual module sections in this manual for more information. **Table 5-20** is a summary of module pin function out of reset. Refer to **APPENDIX D REGISTER SUMMARY** for register function and reset state.

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

Table 5-20 Module Pin Functions

Module[1]	Pin Mnemonic	Function
ADC	PADA[7:0]/AN[7:0]	Discrete input
	V_{RH}	Reference voltage
	V_{RL}	Reference voltage
CPU	DSI/IPIPE1	DSI/IPIPE1
	DSO/IPIPE0	DSO/IPIPE0
	$\overline{BKPT}$/DSCLK	$\overline{BKPT}$/DSCLK
GPT	PGP7/IC4/OC5	Discrete input
	PGP[6:3]/OC[4:1]	Discrete input
	PGP[2:0]/IC[3:1]	Discrete input
	PAI	Discrete input
	PCLK	Discrete input
	PWMA, PWMB	Discrete output
QSM	PQS7/TXD	Discrete input
	PQS[6:4]/PCS[3:1]	Discrete input
	PQS3/PCS0/$\overline{SS}$	Discrete input
	PQS2/SCK	Discrete input
	PQS1/MOSI	Discrete input
	PQS0/MISO	Discrete input
	RXD	RXD
MCCI	PMC7/TXDA	Discrete input
	PMC6/RXDA	Discrete input
	PMC5/TXDB	Discrete input
	PMC4/RXDB	Discrete input
	PMC3/$\overline{SS}$	Discrete input
	PMC2/SCK	Discrete input
	PMC1/MOSI	Discrete input
	PMC0/MISO	Discrete input

NOTES:
1. Module port pins may be in an indeterminate state for up to 15 milliseconds at power-up.

5.7.5 Pin State During Reset

It is important to keep the distinction between pin function and pin electrical state clear. Although control register values and mode select inputs determine pin function, a pin driver can be active, inactive or in high-impedance state while reset occurs. During power-on reset, pin state is subject to the constraints discussed in **5.7.7 Power-On Reset**.

Freescale Semiconductor, Inc.

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

NOTE

Pins that are not used should either be configured as outputs, or (if configured as inputs) pulled to the appropriate inactive state. This decreases additional I_{DD} caused by digital inputs floating near mid-supply level.

5.7.5.1 Reset States of SIM Pins

Generally, while $\overline{\text{RESET}}$ is asserted, SIM pins either go to an inactive high-impedance state or are driven to their inactive states. After $\overline{\text{RESET}}$ is released, mode selection occurs, and reset exception processing begins. Pins configured as inputs must be driven to the desired active state. Pull-up or pull-down circuitry may be necessary. Pins configured as outputs begin to function after $\overline{\text{RESET}}$ is released. **Table 5-21** is a summary of SIM pin states during reset.

Table 5-21 SIM Pin Reset States

Pin(s)	Pin State While $\overline{\text{RESET}}$ Asserted	Pin State After $\overline{\text{RESET}}$ Released			
		Default Function		Alternate Function	
		Pin Function	Pin State	Pin Function	Pin State
$\overline{\text{CS10}}$/ADDR23/ECLK	V_{DD}	$\overline{\text{CS10}}$	V_{DD}	ADDR23	Unknown
$\overline{\text{CS}}$[9:6]/ADDR[22:19]/PC[6:3]	V_{DD}	$\overline{\text{CS}}$[9:6]	V_{DD}	ADDR[22:19]	Unknown
ADDR[18:0]	High-Z	ADDR[18:0]	Unknown	ADDR[18:0]	Unknown
$\overline{\text{AS}}$/PE5	High-Z	$\overline{\text{AS}}$	Output	PE5	Input
$\overline{\text{AVEC}}$/PE2	High-Z	$\overline{\text{AVEC}}$	Input	PE2	Input
$\overline{\text{BERR}}$	High-Z	$\overline{\text{BERR}}$	Input	$\overline{\text{BERR}}$	Input
$\overline{\text{CS1}}$/$\overline{\text{BG}}$	V_{DD}	$\overline{\text{CS1}}$	V_{DD}	$\overline{\text{BG}}$	V_{DD}
$\overline{\text{CS2}}$/$\overline{\text{BGACK}}$	V_{DD}	$\overline{\text{CS2}}$	V_{DD}	$\overline{\text{BGACK}}$	Input
$\overline{\text{CS0}}$/$\overline{\text{BR}}$	V_{DD}	$\overline{\text{CS0}}$	V_{DD}	$\overline{\text{BR}}$	Input
CLKOUT	Output	CLKOUT	Output	CLKOUT	Output
$\overline{\text{CSBOOT}}$	V_{DD}	$\overline{\text{CSBOOT}}$	V_{SS}	$\overline{\text{CSBOOT}}$	V_{SS}
DATA[15:0]	Mode select	DATA[15:0]	Input	DATA[15:0]	Input
$\overline{\text{DS}}$/PE4	High-Z	$\overline{\text{DS}}$	Output	PE4	Input
$\overline{\text{DSACK0}}$/PE0	High-Z	$\overline{\text{DSACK0}}$	Input	PE0	Input
$\overline{\text{DSACK1}}$/PE1	High-Z	$\overline{\text{DSACK1}}$	Input	PE1	Input
$\overline{\text{CS}}$[5:3]/FC[2:0]/PC[2:0]	V_{DD}	$\overline{\text{CS}}$[5:3]	V_{DD}	FC[2:0]	Unknown
$\overline{\text{HALT}}$	High-Z	$\overline{\text{HALT}}$	Input	$\overline{\text{HALT}}$	Input
$\overline{\text{IRQ}}$[7:1]/PF[7:1]	High-Z	$\overline{\text{IRQ}}$[7:1]	Input	PF[7:1]	Input
MODCLK/PF0	Mode Select	MODCLK	Input	PF0	Input
R/$\overline{\text{W}}$	High-Z	R/$\overline{\text{W}}$	Output	R/$\overline{\text{W}}$	Output
$\overline{\text{RESET}}$	Asserted	$\overline{\text{RESET}}$	Input	$\overline{\text{RESET}}$	Input
SIZ[1:0]/PE[7:6]	High-Z	SIZ[1:0]	Unknown	PE[7:6]	Input
TSC	Mode select	TSC	Input	TSC	Input

5.7.5.2 Reset States of Pins Assigned to Other MCU Modules

As a rule, module pins that are assigned to general-purpose I/O ports go into a high-impedance state following reset. However, during power-on reset, module port pins may be in an indeterminate state for a short period. Refer to **5.7.7 Power-On Reset** for more information.

Freescale Semiconductor, Inc.

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

5.7.6 Reset Timing

The $\overline{\text{RESET}}$ input must be asserted for a specified minimum period for reset to occur. External $\overline{\text{RESET}}$ assertion can be delayed internally for a period equal to the longest bus cycle time (or the bus monitor time-out period) in order to protect write cycles from being aborted by reset. While $\overline{\text{RESET}}$ is asserted, SIM pins are either in an inactive, high-impedance state or are driven to their inactive states.

When an external device asserts $\overline{\text{RESET}}$ for the proper period, reset control logic clocks the signal into an internal latch. The control logic drives the $\overline{\text{RESET}}$ pin low for an additional 512 CLKOUT cycles after it detects that the $\overline{\text{RESET}}$ signal is no longer being externally driven to guarantee this length of reset to the entire system.

If an internal source asserts a reset signal, the reset control logic asserts the $\overline{\text{RESET}}$ pin for a minimum of 512 cycles. If the reset signal is still asserted at the end of 512 cycles, the control logic continues to assert the $\overline{\text{RESET}}$ pin until the internal reset signal is negated.

After 512 cycles have elapsed, the $\overline{\text{RESET}}$ pin goes to an inactive, high-impedance state for ten cycles. At the end of this 10-cycle period, the $\overline{\text{RESET}}$ input is tested. When the input is at logic level one, reset exception processing begins. If, however, the $\overline{\text{RESET}}$ input is at logic level zero, reset control logic drives the pin low for another 512 cycles. At the end of this period, the pin again goes to high-impedance state for ten cycles, then it is tested again. The process repeats until external $\overline{\text{RESET}}$ is released.

5.7.7 Power-On Reset

When the SIM clock synthesizer is used to generate system clocks, power-on reset involves special circumstances related to application of the system and the clock synthesizer power. Regardless of clock source, voltage must be applied to clock synthesizer power input pin V_{DDSYN} for the MCU to operate. The following discussion assumes that V_{DDSYN} is applied before and during reset, which minimizes crystal start-up time. When V_{DDSYN} is applied at power-on, start-up time is affected by specific crystal parameters and by oscillator circuit design. V_{DD} ramp-up time also affects pin state during reset. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for voltage and timing specifications.

During power-on reset, an internal circuit in the SIM drives the IMB internal (MSTRST) and external (EXTRST) reset lines. The power-on reset circuit releases the internal reset line as V_{DD} ramps up to the minimum operating voltage, and SIM pins are initialized to the values shown in **Table 5-21**. When V_{DD} reaches the minimum operating voltage, the clock synthesizer VCO begins operation. Clock frequency ramps up to specified limp mode frequency (f_{limp}). The external $\overline{\text{RESET}}$ line remains asserted until the clock synthesizer PLL locks and 512 CLKOUT cycles elapse.

NOTE

V_{DDSYN} and all V_{DD} pins must be powered. Applying power to V_{DDSYN} only will cause errant behavior of the MCU.

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

The SIM clock synthesizer provides clock signals to the other MCU modules. After the clock is running and MSTRST is asserted for at least four clock cycles, these modules reset. V_{DD} ramp time and VCO frequency ramp time determine how long the four cycles take. Worst case is approximately 15 milliseconds. During this period, module port pins may be in an indeterminate state. While input-only pins can be put in a known state by external pull-up resistors, external logic on input/output or output-only pins during this time must condition the lines. Active drivers require high-impedance buffers or isolation resistors to prevent conflict.

Figure 5-20 is a timing diagram for power-on reset. It shows the relationships between $\overline{\text{RESET}}$, V_{DD}, and bus signals.

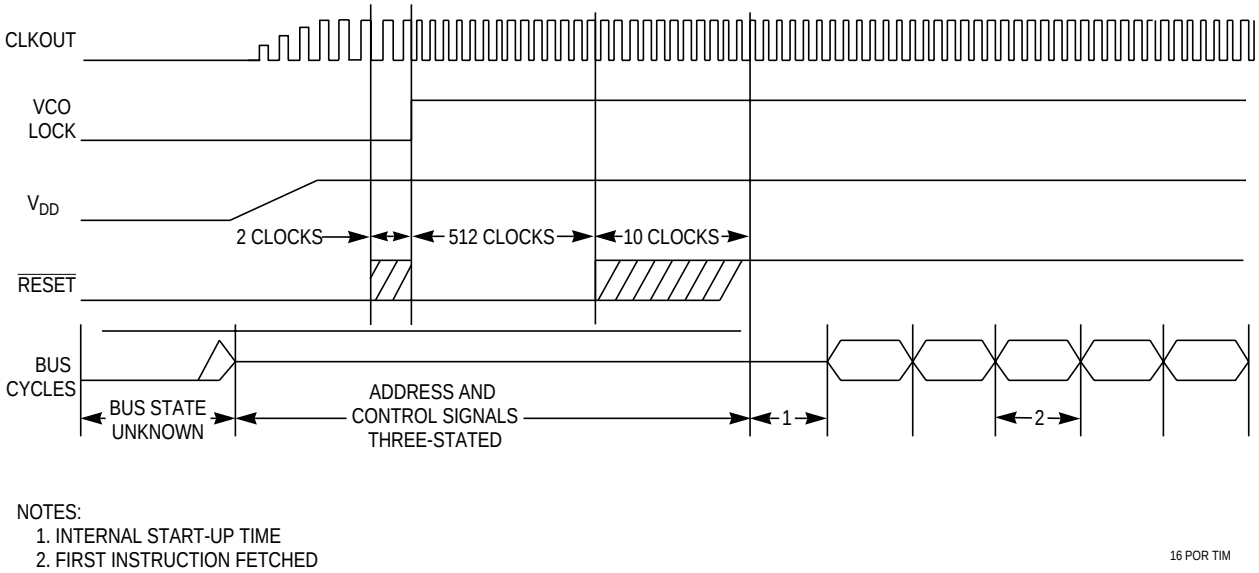


Figure 5-20 Power-On Reset

5.7.8 Use of the Three-State Control Pin

Asserting the three-state control (TSC) input causes the MCU to put all output drivers in a disabled, high-impedance state. The signal must remain asserted for approximately ten clock cycles in order for drivers to change state.

When the internal clock synthesizer is used (MODCLK held high during reset), synthesizer ramp-up time affects how long the ten cycles take. Worst case is approximately 20 milliseconds from TSC assertion.

When an external clock signal is applied (MODCLK held low during reset), pins go to high-impedance state as soon after TSC assertion as approximately ten clock pulses have been applied to the EXTAL pin.

Freescale Semiconductor, Inc.

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

NOTE

When TSC assertion takes effect, internal signals are forced to values that can cause inadvertent mode selection. Once the output drivers change state, the MCU must be powered down and restarted before normal operation can resume.

5.7.9 Reset Processing Summary

To prevent write cycles in progress from being corrupted, a reset is recognized at the end of a bus cycle, and not at an instruction boundary. Any processing in progress at the time a reset occurs is aborted. After SIM reset control logic has synchronized an internal or external reset request, the MSTRST signal is asserted.

The following events take place when MSTRST is asserted.

A. Instruction execution is aborted.
B. The condition code register is initialized.
   1. The IP field is set to $7, disabling all interrupts below priority 7.
   2. The S bit is set, disabling LPSTOP mode.
   3. The SM bit is cleared, disabling MAC saturation mode.
C. The K register is cleared.

NOTE

All CCR bits that are not initialized are not affected by reset. However, out of power-on reset, these bits are indeterminate.

The following events take place when MSTRST is negated after assertion.

A. The CPU16 samples the $\overline{\text{BKPT}}$ input.
B. The CPU16 fetches $\overline{\text{RESET}}$ vectors in the following order:
   1. Initial ZK, SK, and PK extension field values
   2. Initial PC
   3. Initial SP
   4. Initial IZ value

   Vectors can be fetched from internal RAM or from external ROM enabled by the $\overline{\text{CSBOOT}}$ signal.

C. The CPU16 begins fetching instructions pointed to by the initial PK : PC.

5.7.10 Reset Status Register

The reset status register (RSR) contains a bit for each reset source in the MCU. When a reset occurs, a bit corresponding to the reset type is set. When multiple causes of reset occur at the same time, more than one bit in RSR may be set. The reset status register is updated by the reset control logic when the $\overline{\text{RESET}}$ signal is released. Refer to **APPENDIX D REGISTER SUMMARY**.

Freescale Semiconductor, Inc.

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

5.8 Interrupts

Interrupt recognition and servicing involve complex interaction between the SIM, the CPU16, and a device or module requesting interrupt service. This discussion provides an overview of the entire interrupt process. Chip-select logic can also be used to respond to interrupt requests. Refer to **5.9 Chip-Selects** for more information.

5.8.1 Interrupt Exception Processing

The CPU16 handles interrupts as a type of asynchronous exception. An exception is an event that preempts normal processing. Exception processing makes the transition from normal instruction execution to execution of a routine that deals with an exception. Each exception has an assigned vector that points to an associated handler routine. These vectors are stored in a vector table located in the first 512 bytes of address bank 0. The CPU16 uses vector numbers to calculate displacement into the table. Refer to **4.13 Exceptions** for more information.

5.8.2 Interrupt Priority and Recognition

The CPU16 provides for seven levels of interrupt priority (1 – 7), seven automatic interrupt vectors, and 200 assignable interrupt vectors. All interrupts with priorities less than seven can be masked by the interrupt priority (IP) field in the condition code register.

There are seven interrupt request signals ($\overline{\text{IRQ}}$[7:1]). These signals are used internally on the IMB, and there are corresponding pins for external interrupt service requests. The CPU16 treats all interrupt requests as though they come from internal modules; external interrupt requests are treated as interrupt service requests from the SIM. Each of the interrupt request signals corresponds to an interrupt priority level. $\overline{\text{IRQ1}}$ has the lowest priority and $\overline{\text{IRQ7}}$ the highest.

The IP field consists of three bits (CCR[7:5]). Binary values %000 to %111 provide eight priority masks. Masks prevent an interrupt request of a priority less than or equal to the mask value (except for $\overline{\text{IRQ7}}$) from being recognized and processed. When IP contains %000, no interrupt is masked. During exception processing, the IP field is set to the priority of the interrupt being serviced.

Interrupt recognition is determined by interrupt priority level and interrupt priority (IP) mask value. The interrupt priority mask consists of three bits in the CPU16 condition code register (CCR[7:5]). Binary values %000 to %111 provide eight priority masks. Masks prevent an interrupt request of a priority less than or equal to the mask value from being recognized and processed. $\overline{\text{IRQ7}}$, however, is always recognized, even if the mask value is %111.

$\overline{\text{IRQ}}$[7:1] are active-low level-sensitive inputs. The low on the pin must remain asserted until an interrupt acknowledge cycle corresponding to that level is detected.

$\overline{\text{IRQ7}}$ is transition-sensitive as well as level-sensitive: a level-7 interrupt is not detected unless a falling edge transition is detected on the $\overline{\text{IRQ7}}$ line. This prevents redundant servicing and stack overflow. A non-maskable interrupt is generated each time $\overline{\text{IRQ7}}$ is asserted as well as each time the priority mask is written while $\overline{\text{IRQ7}}$ is asserted. If $\overline{\text{IRQ7}}$ is asserted and the IP mask is written to any new value (including %111), $\overline{\text{IRQ7}}$ will be recognized as a new $\overline{\text{IRQ7}}$.

Freescale Semiconductor, Inc.

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

Interrupt requests are sampled on consecutive falling edges of the system clock. Interrupt request input circuitry has hysteresis. To be valid, a request signal must be asserted for at least two consecutive clock periods. Valid requests do not cause immediate exception processing, but are left pending. Pending requests are processed at instruction boundaries or when exception processing of higher-priority interrupts is complete.

The CPU16 does not latch the priority of a pending interrupt request. If an interrupt source of higher priority makes a service request while a lower priority request is pending, the higher priority request is serviced. If an interrupt request with a priority equal to or lower than the current IP mask value is made, the CPU16 does not recognize the occurrence of the request. If simultaneous interrupt requests of different priorities are made, and both have a priority greater than the mask value, the CPU16 recognizes the higher-level request.

5.8.3 Interrupt Acknowledge and Arbitration

When the CPU16 detects one or more interrupt requests of a priority higher than the interrupt priority mask value, it places the interrupt request level on the address bus and initiates a CPU space read cycle. The request level serves two purposes: it is decoded by modules or external devices that have requested interrupt service, to determine whether the current interrupt acknowledge cycle pertains to them, and it is latched into the interrupt priority mask field in the CPU16 condition code register to preclude further interrupts of lower priority during interrupt service.

Modules or external devices that have requested interrupt service must decode the IP mask value placed on the address bus during the interrupt acknowledge cycle and respond if the priority of the service request corresponds to the mask value. However, before modules or external devices respond, interrupt arbitration takes place.

Arbitration is performed by means of serial contention between values stored in individual module interrupt arbitration (IARB) fields. Each module that can make an interrupt service request, including the SIM, has an IARB field in its configuration register. IARB fields can be assigned values from %0000 to %1111. In order to implement an arbitration scheme, each module that can request interrupt service must be assigned a unique, non-zero IARB field value during system initialization. Arbitration priorities range from %0001 (lowest) to %1111 (highest). If the CPU16 recognizes an interrupt service request from a source that has an IARB field value of %0000, a spurious interrupt exception is processed.

WARNING

Do not assign the same arbitration priority to more than one module. When two or more IARB fields have the same nonzero value, the CPU16 interprets multiple vector numbers at the same time, with unpredictable consequences.

Because the EBI manages external interrupt requests, the SIM IARB value is used for arbitration between internal and external interrupt requests. The reset value of IARB for the SIM is %1111, and the reset IARB value for all other modules is %0000.

Freescale Semiconductor, Inc.

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

Although arbitration is intended to deal with simultaneous requests of the same interrupt level, it always takes place, even when a single source is requesting service. This is important for two reasons: the EBI does not transfer the interrupt acknowledge read cycle to the external bus unless the SIM wins contention, and failure to contend causes the interrupt acknowledge bus cycle to be terminated early by a bus error.

When arbitration is complete, the module with both the highest asserted interrupt level and the highest arbitration priority must terminate the bus cycle. Internal modules place an interrupt vector number on the data bus and generate appropriate internal cycle termination signals. In the case of an external interrupt request, after the interrupt acknowledge cycle is transferred to the external bus, the appropriate external device must respond with a vector number, then generate data size acknowledge ($\overline{\text{DSACK}}$) termination signals, or it must assert the autovector ($\overline{\text{AVEC}}$) request signal. If the device does not respond in time, the SIM bus monitor, if enabled, asserts the bus error signal ($\overline{\text{BERR}}$), and a spurious interrupt exception is taken.

Chip-select logic can also be used to generate internal $\overline{\text{AVEC}}$ or $\overline{\text{DSACK}}$ signals in response to interrupt acknowledgment cycles. Refer to **5.9.3 Using Chip-Select Signals for Interrupt Acknowledge** for more information. Chip-select address match logic functions only after the EBI transfers an interrupt acknowledge cycle to the external bus following IARB contention. All interrupts from internal modules have their associated IACK cycles terminated with an internal $\overline{\text{DSACK}}$. Thus, user vectors (instead of autovectors) must always be used for interrupts generated from internal modules. If an internal module makes an interrupt request of a certain priority, and the appropriate chip-select registers are programmed to generate $\overline{\text{AVEC}}$ or $\overline{\text{DSACK}}$ signals in response to an interrupt acknowledge cycle for that priority level, chip-select logic does not respond to the interrupt acknowledge cycle, and the internal module supplies a vector number and generates internal cycle termination signals.

For periodic timer interrupts, the PIRQ[2:0] field in the periodic interrupt control register (PICR) determines PIT priority level. A PIRQ[2:0] value of %000 means that PIT interrupts are inactive. By hardware convention, when the CPU16 receives simultaneous interrupt requests of the same level from more than one SIM source (including external devices), the periodic interrupt timer is given the highest priority, followed by the $\overline{\text{IRQ}}$ pins.

5.8.4 Interrupt Processing Summary

A summary of the entire interrupt processing sequence follows. When the sequence begins, a valid interrupt service request has been detected and is pending.

A. The CPU16 finishes higher priority exception processing or reaches an instruction boundary.
B. Processor state is stacked, then the CCR PK extension field is cleared.
C. The interrupt acknowledge cycle begins:
    1. FC[2:0] are driven to %111 (CPU space) encoding.
    2. The address bus is driven as follows. ADDR[23:20] = %1111; ADDR[19:16] = %1111, which indicates that the cycle is an interrupt acknowledge CPU space cycle; ADDR[15:4] = %111111111111; ADDR[3:1] = the priority of the interrupt request being acknowledged; and ADDR0 = %1.

Freescale Semiconductor, Inc.

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

3. Request priority is latched into the CCR IP field from the address bus.

D. Modules or external peripherals that have requested interrupt service decode the priority value in ADDR[3:1]. If request priority is the same as acknowledged priority, arbitration by IARB contention takes place.

E. After arbitration, the interrupt acknowledge cycle is completed in one of the following ways:
   1. When there is no contention (IARB = %0000), the spurious interrupt monitor asserts $\overline{\text{BERR}}$, and the CPU16 generates the spurious interrupt vector number.
   2. The dominant interrupt source supplies a vector number and $\overline{\text{DSACK}}$ signals appropriate to the access. The CPU16 acquires the vector number.
   3. The $\overline{\text{AVEC}}$ signal is asserted (the signal can be asserted by the dominant interrupt source or the pin can be tied low), and the CPU16 generates an autovector number corresponding to interrupt priority.
   4. The bus monitor asserts $\overline{\text{BERR}}$ and the CPU16 generates the spurious interrupt vector number.

F. The vector number is converted to a vector address.

G. The content of the vector address is loaded into the PC and the processor transfers control to the exception handler routine.

5.8.5 Interrupt Acknowledge Bus Cycles

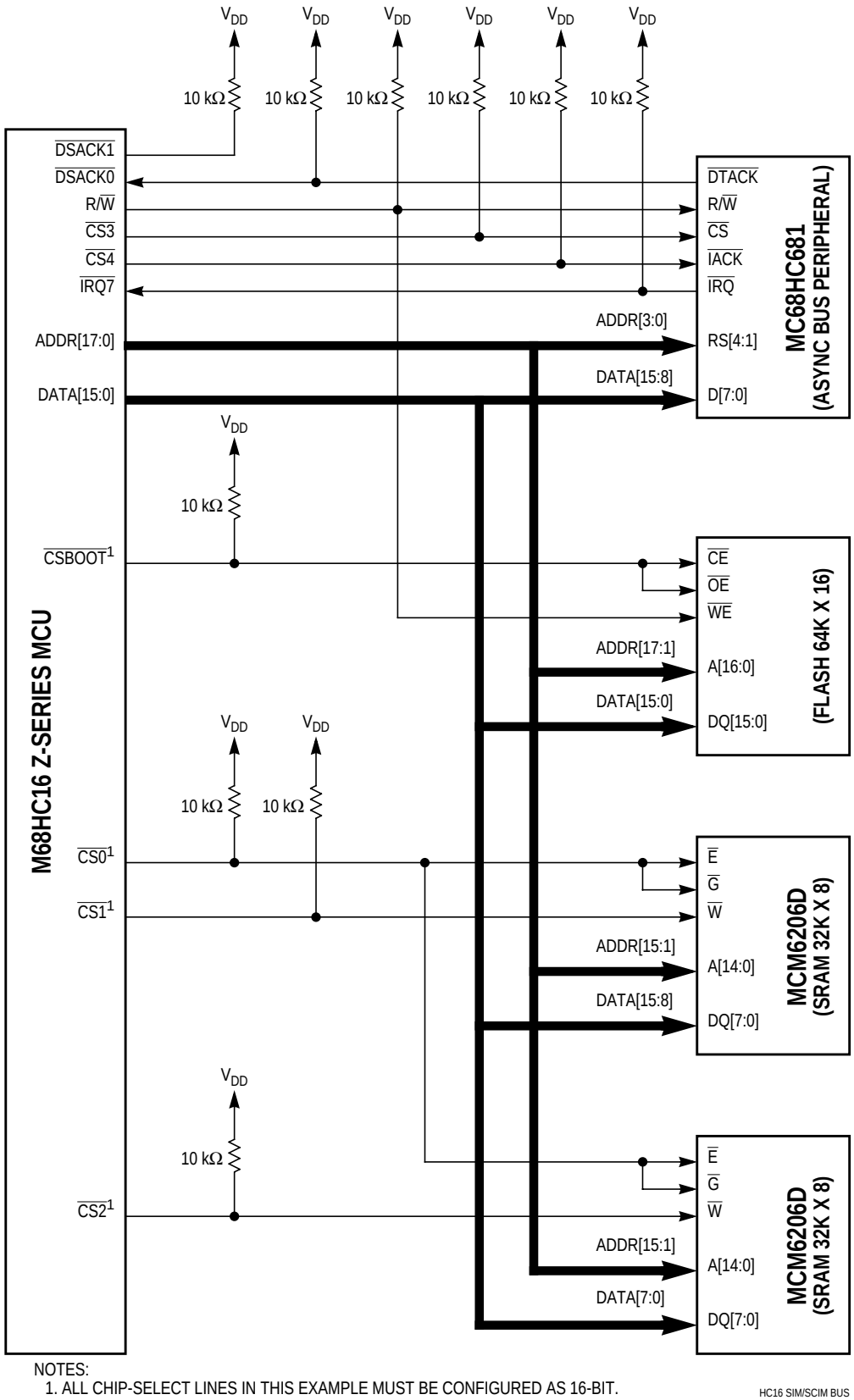
Interrupt acknowledge bus cycles are CPU space cycles that are generated during exception processing. For further information about the types of interrupt acknowledge bus cycles determined by $\overline{\text{AVEC}}$ or $\overline{\text{DSACK}}$, refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** and the *SIM Reference Manual* (SIMRM/AD).

5.9 Chip-Selects

Typical microcontrollers require additional hardware to provide external chip-select signals. The MCU includes 12 programmable chip-select circuits that can provide from two to 16 clock-cycle access to external memory and peripherals. Address block sizes of 2 Kbytes to 512 Kbytes can be selected. However, because ADDR[23:20] follow the state of ADDR19, 512-Kbyte blocks are the largest usable size. **Figure 5-21** is a diagram of a basic system that uses chip-selects.

Freescale Semiconductor, Inc.

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



NOTES:
1. ALL CHIP-SELECT LINES IN THIS EXAMPLE MUST BE CONFIGURED AS 16-BIT.

Figure 5-21 Basic MCU System

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

Freescale Semiconductor, Inc.

Chip-select assertion can be synchronized with bus control signals to provide output enable, read/write strobe, or interrupt acknowledge signals. Logic can also generate DSACK and AVEC signals internally. A single DSACK generator is shared by all chip-selects. Each signal can also be synchronized with the ECLK signal available on ADDR23.

When a memory access occurs, chip-select logic compares address space type, address, type of access, transfer size, and interrupt priority (in the case of interrupt acknowledge) to parameters stored in chip-select registers. If all parameters match, the appropriate chip-select signal is asserted. Select signals are active low. If a chip-select function is given the same address as a microcontroller module or an internal memory array, an access to that address goes to the module or array, and the chip-select signal is not asserted. The external address and data buses do not reflect the internal access.

All chip-select circuits are configured for operation out of reset. However, all chip-select signals except CSBOOT are disabled, and cannot be asserted until the BYTE[1:0] field in the corresponding option register is programmed to a non-zero value to select a transfer size. The chip-select option register must not be written until a base address has been written to a proper base address register. Alternate functions for chip-select pins are enabled if appropriate data bus pins are held low at the release of RESET. Refer to **5.7.3.1 Data Bus Mode Selection** for more information. **Figure 5-22** is a functional diagram of a single chip-select circuit.

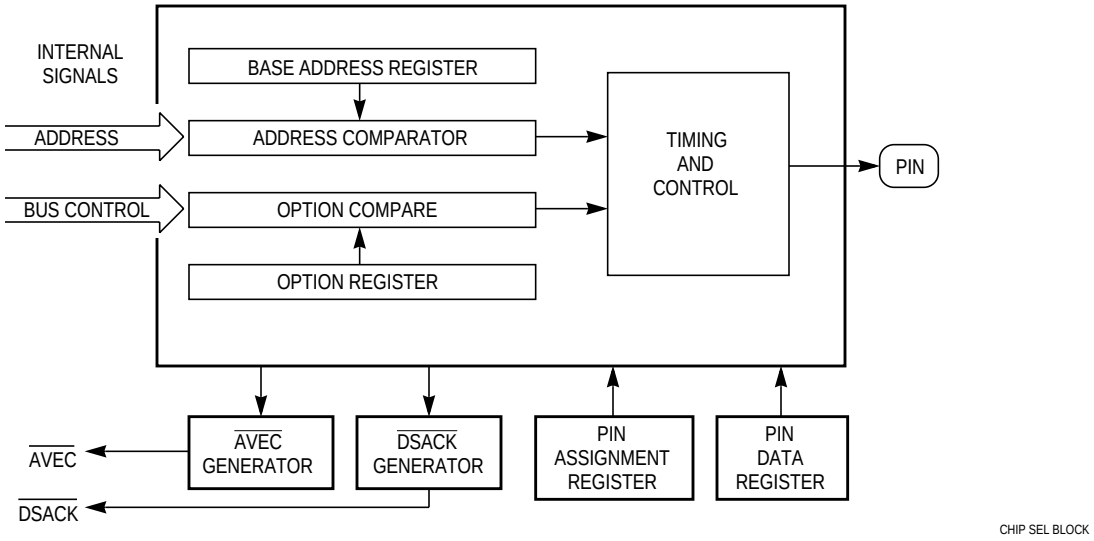


Figure 5-22 Chip-Select Circuit Block Diagram

5.9.1 Chip-Select Registers

Each chip-select pin can have one or more functions. Chip-select pin assignment registers CSPAR[1:0] determine functions of the pins. Pin assignment registers also determine port size (8- or 16-bit) for dynamic bus allocation. A pin data register (PORTC) latches data for chip-select pins that are used for discrete output.

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

Freescale Semiconductor, Inc.

Blocks of addresses are assigned to each chip-select function. Block sizes of 2 Kbytes to 1 Mbyte can be selected by writing values to the appropriate base address register (CSBAR[10:0] and CSBARBT). However, because the logic state of ADDR20 is always the same as the state of ADDR19 in the MCU, the largest usable block size is 512 Kbytes. Multiple chip-selects assigned to the same block of addresses must have the same number of wait states.

Chip-select option registers (CSORBT and CSOR[0:10]) determine timing of and conditions for assertion of chip-select signals. Eight parameters, including operating mode, access size, synchronization, and wait state insertion can be specified.

Initialization software usually resides in a peripheral memory device controlled by the chip-select circuits. A set of special chip-select functions and registers (CSORBT and CSBARBT) is provided to support bootstrap operation.

Comprehensive address maps and register diagrams are provided in **APPENDIX D REGISTER SUMMARY**.

5.9.1.1 Chip-Select Pin Assignment Registers

The pin assignment registers contain twelve 2-bit fields that determine the functions of the chip-select pins. Each pin has two or three possible functions, as shown in **Table 5-22**.

Table 5-22 Chip-Select Pin Functions

Chip-Select	Alternate Function	Discrete Output
$\overline{\text{CSBOOT}}$	$\overline{\text{CSBOOT}}$	—
$\overline{\text{CS0}}$	$\overline{\text{BR}}$	—
$\overline{\text{CS1}}$	$\overline{\text{BG}}$	—
$\overline{\text{CS2}}$	$\overline{\text{BGACK}}$	—
$\overline{\text{CS3}}$	FC0	PC0
$\overline{\text{CS4}}$	FC1	PC1
$\overline{\text{CS5}}$	FC2	PC2
$\overline{\text{CS6}}$	ADDR19	PC3
$\overline{\text{CS7}}$	ADDR20	PC4
$\overline{\text{CS8}}$	ADDR21	PC5
$\overline{\text{CS9}}$	ADDR22	PC6
$\overline{\text{CS10}}$	ADDR23	ECLK

Table 5-23 shows pin assignment field encoding. Pins that have no discrete output function must not use the %00 encoding as this will cause the alternate function to be selected. For instance, %00 for $\overline{\text{CS0}}$/$\overline{\text{BR}}$ will cause the pin to perform the $\overline{\text{BR}}$ function.

Table 5-23 Pin Assignment Field Encoding

CSxPA[1:0]	Description
00	Discrete output
01	Alternate function
10	Chip-select (8-bit port)
11	Chip-select (16-bit port)

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

Port size determines the way in which bus transfers to an external address are allocated. Port size of eight bits or sixteen bits can be selected when a pin is assigned as a chip-select. Port size and transfer size affect how the chip-select signal is asserted. Refer to **5.9.1.3 Chip-Select Option Registers** for more information.

Out of reset, chip-select pin function is determined by the logic level on a corresponding data bus pin. The data bus pins have weak internal pull-up drivers, but can be held low by external devices. Refer to **5.7.3.1 Data Bus Mode Selection** for more information. Either 16-bit chip-select function (%11) or alternate function (%01) can be selected during reset. All pins except the boot ROM select pin ($\overline{\text{CSBOOT}}$) are disabled out of reset. There are twelve chip-select functions and only eight associated data bus pins. There is not a one-to-one correspondence. Refer to **5.9.4 Chip-Select Reset Operation** for more detailed information.

The $\overline{\text{CSBOOT}}$ signal is enabled out of reset. The state of the DATA0 line during reset determines what port width $\overline{\text{CSBOOT}}$ uses. If DATA0 is held high (either by the weak internal pull-up driver or by an external pull-up device), 16-bit port size is selected. If DATA0 is held low, 8-bit port size is selected.

A pin programmed as a discrete output drives an external signal to the value specified in the port C register. No discrete output function is available on $\overline{\text{CSBOOT}}$, $\overline{\text{BR}}$, $\overline{\text{BG}}$, or $\overline{\text{BGACK}}$. ADDR23 provides the ECLK output rather than a discrete output signal.

When a pin is programmed for discrete output or alternate function, internal chip-select logic still functions and can be used to generate $\overline{\text{DSACK}}$ or $\overline{\text{AVEC}}$ internally on an address and control signal match.

5.9.1.2 Chip-Select Base Address Registers

Each chip-select has an associated base address register. A base address is the lowest address in the block of addresses enabled by a chip-select. Block size is the extent of the address block above the base address. Block size is determined by the value contained in BLKSZ[2:0]. Multiple chip-selects assigned to the same block of addresses must have the same number of wait states.

BLKSZ[2:0] determines which bits in the base address field are compared to corresponding bits on the address bus during an access. Provided other constraints determined by option register fields are also satisfied, when a match occurs, the associated chip-select signal is asserted. **Table 5-24** shows BLKSZ[2:0] encoding.

Table 5-24 Block Size Encoding

BLKSZ[2:0]	Block Size	Address Lines Compared[1]
000	2 Kbytes	ADDR[23:11]
001	8 Kbytes	ADDR[23:13]
010	16 Kbytes	ADDR[23:14]
011	64 Kbytes	ADDR[23:16]
100	128 Kbytes	ADDR[23:17]
101	256 Kbytes	ADDR[23:18]
110	512 Kbytes	ADDR[23:19]
111	512 Kbytes	ADDR[23:20]

NOTES:
1. ADDR[23:20] are the same logic level as ADDR19 during normal operation.

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

The chip-select address compare logic uses only the most significant bits to match an address within a block. The value of the base address must be an integer multiple of the block size.

Because the logic state of ADDR[23:20] follows that of ADDR19 in the CPU16, maximum block size is 512 Kbytes, and addresses from $080000 to $F7FFFF are inaccessible.

After reset, the MCU fetches the initialization routine from the address contained in the reset vector, located beginning at address $000000 of program space. To support bootstrap operation from reset, the base address field in the boot chip-select base address register (CSBARBT) has a reset value of $000, which corresponds to a base address of $000000 and a block size of 512 Kbytes. A memory device containing the reset vector and initialization routine can be automatically enabled by $\overline{\text{CSBOOT}}$ after a reset. Refer to **5.9.4 Chip-Select Reset Operation** for more information.

5.9.1.3 Chip-Select Option Registers

Option register fields determine timing of and conditions for assertion of chip-select signals. To assert a chip-select signal, and to provide $\overline{\text{DSACK}}$ or autovector support, other constraints set by fields in the option register and in the base address register must also be satisfied. The following paragraphs summarize option register functions. Refer to **D.2.21 Chip-Select Option Registers** for register and bit field information.

The MODE bit determines whether chip-select assertion simulates an asynchronous bus cycle, or is synchronized to the M6800-type bus clock signal ECLK available on ADDR23. Refer to **5.3 System Clock** for more information on ECLK.

BYTE[1:0] controls bus allocation for chip-select transfers. Port size, set when a chip-select is enabled by a pin assignment register, affects signal assertion. When an 8-bit port is assigned, any BYTE field value other than %00 enables the chip-select signal. When a 16-bit port is assigned, however, BYTE field value determines when the chip-select is enabled. The BYTE fields for $\overline{\text{CS}}$[10:0] are cleared during reset. However, both bits in the boot ROM chip-select option register (CSORBT) BYTE field are set (%11) when the $\overline{\text{RESET}}$ signal is released.

R/$\overline{\text{W}}$[1:0] causes a chip-select signal to be asserted only for a read, only for a write, or for both read and write. Use this field in conjunction with the STRB bit to generate asynchronous control signals for external devices.

The STRB bit controls the timing of a chip-select assertion in asynchronous mode. Selecting address strobe causes a chip-select signal to be asserted synchronized with the address strobe. Selecting data strobe causes a chip-select signal to be asserted synchronized with the data strobe. This bit has no effect in synchronous mode.

$\overline{\text{DSACK}}$[3:0] specifies the source of $\overline{\text{DSACK}}$ in asynchronous mode. It also allows the user to optimize bus speed in a particular application by controlling the number of wait states that are inserted.

NOTE

The external $\overline{\text{DSACK}}$ pins are always active.

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

SPACE[1:0] determines the address space in which a chip-select is asserted. An access must have the space type represented by the SPACE[1:0] encoding in order for a chip-select signal to be asserted.

IPL[2:0] contains an interrupt priority mask that is used when chip-select logic is set to trigger on external interrupt acknowledge cycles. When SPACE[1:0] is set to %00 (CPU space), interrupt priority (ADDR[3:1]) is compared to the IPL field. If the values are the same, and other option register constraints are satisfied, a chip-select signal is asserted. This field only affects the response of chip-selects and does not affect interrupt recognition by the CPU. Encoding %000 in the IPL field causes a chip-select signal to be asserted regardless of interrupt acknowledge cycle priority, provided all other constraints are met.

The $\overline{\text{AVEC}}$ bit is used to make a chip-select respond to an interrupt acknowledge cycle. If the $\overline{\text{AVEC}}$ bit is set, an autovector will be selected for the particular external interrupt being serviced. If $\overline{\text{AVEC}}$ is zero, the interrupt acknowledge cycle will be terminated with $\overline{\text{DSACK}}$, and an external vector number must be supplied by an external device.

5.9.1.4 PORTC Data Register

The PORTC data register latches data for PORTC pins programmed as discrete outputs. When a pin is assigned as a discrete output, the value in this register appears at the output. PC[6:0] correspond to $\overline{\text{CS}}$[9:3]. Bit 7 is not used. Writing to this bit has no effect, and it always reads zero.

5.9.2 Chip-Select Operation

When the MCU makes an access, enabled chip-select circuits compare the following items:

- Function codes to SPACE fields, and to the IP mask if the SPACE field encoding is not for CPU space.
- Appropriate address bus bits to base address fields.
- Read/write status to R/$\overline{\text{W}}$ fields.
- ADDR0 and/or SIZ[1:0] bits to BYTE field (16-bit ports only).
- Priority of the interrupt being acknowledged (ADDR[3:1]) to IPL fields (when the access is an interrupt acknowledge cycle).

When a match occurs, the chip-select signal is asserted. Assertion occurs at the same time as $\overline{\text{AS}}$ or $\overline{\text{DS}}$ assertion in asynchronous mode. Assertion is synchronized with ECLK in synchronous mode. In asynchronous mode, the value of the $\overline{\text{DSACK}}$ field determines whether $\overline{\text{DSACK}}$ is generated internally. $\overline{\text{DSACK}}$[3:0] also determines the number of wait states inserted before internal $\overline{\text{DSACK}}$ assertion.

The speed of an external device determines whether internal wait states are needed. Normally, wait states are inserted into the bus cycle during S3 until a peripheral asserts $\overline{\text{DSACK}}$. If a peripheral does not generate $\overline{\text{DSACK}}$, internal $\overline{\text{DSACK}}$ generation must be selected and a predetermined number of wait states can be programmed into the chip-select option register. Refer to the *SIM Reference Manual* (SIMRM/AD) for further information.

Freescale Semiconductor, Inc.

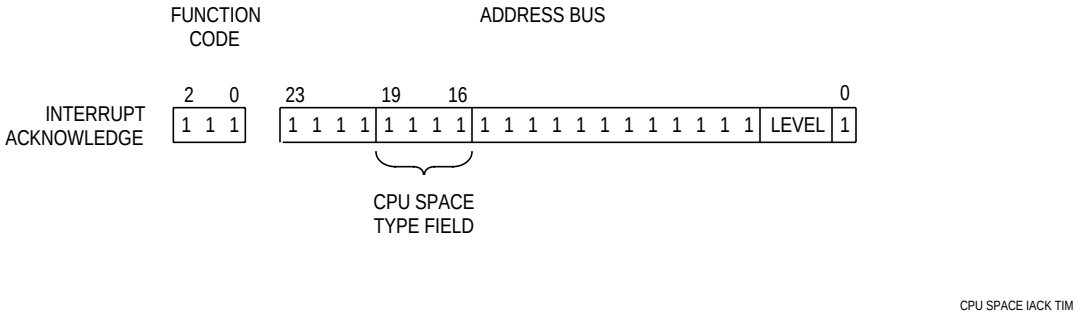
For More Information On This Product,
Go to: www.freescale.com

5.9.3 Using Chip-Select Signals for Interrupt Acknowledge

Ordinary bus cycles use supervisor or user space access, but interrupt acknowledge bus cycles use CPU space access. Refer to **5.6.4 CPU Space Cycles** and **5.8 Interrupts** for more information. There are no differences in flow for chip selects in each type of space, but base and option registers must be properly programmed for each type of external bus cycle.

During a CPU space cycle, bits [15:3] of the appropriate base register must be configured to match ADDR[23:11], as the address is compared to an address generated by the CPU. ADDR[23:20] follow the state of ADDR19 in this MCU. The states of base register bits [15:12] must match that of bit 11.

Figure 5-23 shows CPU space encoding for an interrupt acknowledge cycle. FC[2:0] are set to %111, designating CPU space access. ADDR[3:1] indicate interrupt priority, and the space type field (ADDR[19:16]) is set to %1111, the interrupt acknowledge code. The rest of the address lines are set to one.


FUNCTION CODE
ADDRESS BUS
INTERRUPT ACKNOWLEDGE
2 0
1 1 1
23 19 16 0
1 1 1

4. Set the BYTE field to lower byte when using a 16-bit port, as the external vector for a 16-bit port is fetched from the lower byte. Set the BYTE field to upper byte when using an 8-bit port.

If an interrupting device does not provide a vector number, an autovector acknowledge must be generated, either by asserting the AVEC pin or by generating AVEC internally using the chip-select option register. This terminates the bus cycle.

5.9.4 Chip-Select Reset Operation

The least significant bit of each of the 2-bit chip-select pin assignment fields in CSPAR0 and CSPAR1 each have a reset value of one. The reset values of the most significant bits of each field are determined by the states of DATA[7:1] during reset. There are weak internal pull-up drivers for each of the data lines so that chip-select operation is selected by default out of reset. However, the internal pull-up drivers can be overcome by bus loading effects.

To ensure a particular configuration out of reset, use an active device to put the data lines in a known state during reset. The base address fields in chip-select base address registers CSBAR[0:10] and chip-select option registers CSOR[0:10] have the reset values shown in **Table 5-25**. The BYTE fields of CSOR[0:10] have a reset value of “disable”, so that a chip-select signal cannot be asserted until the base and option registers are initialized.

Table 5-25 Chip-Select Base and Option Register Reset Values

Fields	Reset Values
Base address	$000000
Block size	2 Kbyte
Async/sync mode	Asynchronous mode
Upper/lower byte	Disabled
Read/write	Disabled
AS/DS	AS
DSACK	No wait states
Address space	CPU space
IPL	Any level
Autovector	External interrupt vector

Following reset, the MCU fetches the initial stack pointer and program counter values from the exception vector table, beginning at $000000 in supervisor program space. The CSBOOT chip-select signal is used to select an external boot device mapped to a base address of $000000.

The MSB of the CSBTPA field in CSPAR0 has a reset value of one, so that chip-select function is selected by default out of reset. The BYTE field in chip-select option register CSORBT has a reset value of “both bytes” so that the select signal is enabled out of reset. The LSB of the CSBOOT field, determined by the logic level of DATA0 during reset, selects the boot ROM port size. When DATA0 is held low during reset, port size is eight bits. When DATA0 is held high during reset, port size is 16 bits. DATA0 has a weak internal pull-up driver, so that a 16-bit port is selected by default

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

Freescale Semiconductor, Inc.

However, the internal pull-up driver can be overcome by bus loading effects. To ensure a particular configuration out of reset, use an active device to put DATA0 in a known state during reset.

The base address field in the boot chip-select base address register CSBARBT has a reset value of all zeros, so that when the initial access to address $000000 is made, an address match occurs, and the $\overline{\text{CSBOOT}}$ signal is asserted. The block size field in CSBARBT has a reset value of 512 Kbytes. **Table 5-26** shows $\overline{\text{CSBOOT}}$ reset values.

Table 5-26 $\overline{\text{CSBOOT}}$ Base and Option Register Reset Values

Fields	Reset Values
Base address	$000000
Block size	512 Kbyte
Async/sync mode	Asynchronous mode
Upper/lower byte	Both bytes
Read/write	Read/write
$\overline{\text{AS}}$/$\overline{\text{DS}}$	$\overline{\text{AS}}$
$\overline{\text{DSACK}}$	13 wait states
Address space	Supervisor space
IPL[1]	Any level
Autovector	Interrupt vector externally

NOTES:
1. These fields are not used unless “Address space” is set to CPU space.

5.10 Parallel Input/Output Ports

Sixteen SIM pins can be configured for general-purpose discrete input and output. Although these pins are organized into two ports, port E and port F, function assignment is by individual pin. PE3 is not connected to a pin. PE3 returns zero when read and writes have no effect. Pin assignment registers, data direction registers, and data registers are used to implement discrete I/O.

5.10.1 Pin Assignment Registers

Bits in the port E and port F pin assignment registers (PEPAR and PFPAR) control the functions of the pins on each port. Any bit set to one defines the corresponding pin as a bus control signal. Any bit cleared to zero defines the corresponding pin as an I/O pin. PEPA3 returns one when read, and writes have no effect.

5.10.2 Data Direction Registers

Bits in the port E and port F data direction registers (DDRE and DDRF) control the direction of the pin drivers when the pins are configured as I/O. Any bit in a register set to one configures the corresponding pin as an output. Any bit in a register cleared to zero configures the corresponding pin as an input. These registers can be read or written at any time. DDE3 returns zero when read. Writes have no effect.

Freescale Semiconductor, Inc.

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

5.10.3 Data Registers

A write to the port E and port F data registers (PORTE[0:1] and PORTF[0:1]) is stored in an internal data latch, and if any pin in the corresponding port is configured as an output, the value stored for that bit is driven out on the pin. A read of a data register returns the value at the pin only if the pin is configured as a discrete input. Otherwise, the value read is the value stored in the register. Both data registers can be accessed in two locations and can be read or written at any time.

5.11 Factory Test

The test submodule supports scan-based testing of the various MCU modules. It is integrated into the SIM to support production test. Test submodule registers are intended for Freescale use only. Register names and addresses are provided in **APPENDIX D REGISTER SUMMARY** to show the user that these addresses are occupied. The QUOT pin is also used for factory test.

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 6
STANDBY RAM MODULE

The standby RAM (SRAM) module consists of a fixed-location control register block and an array of fast (two clock) static RAM that may be mapped to a user specified location in the system memory map. Array size depends on the M68HC16, M68CK16, and M68CM16 Z-series version. Refer to **Table 6-1** for appropriate SRAM array size. The SRAM is especially useful for system stacks and variable storage.

Table 6-1 SRAM Configuration

Z-Series Device	Array Size
MC68HC16Z1 MC68CK16Z1 MC68CM16Z1 MC68HC16Z4 MC68CK16Z4	1 Kbyte
MC68HC16Z2	2 Kbytes
MC68HC16Z3	4 Kbytes

The SRAM can be mapped to any address that is a multiple of the array size so long as SRAM boundaries do not overlap the module control registers (overlap makes the registers inaccessible). Data can be read/written in bytes, words or long words. SRAM is powered by V_{DD} in normal operation. During power-down, SRAM contents can be maintained by power from the V_{STBY} input. Power switching between sources is automatic.

6.1 SRAM Register Block

There are four SRAM control registers: the RAM module configuration register (RAMMCR), the RAM test register (RAMTST), and the RAM array base address registers (RAMBAH/RAMBAL).

The module mapping bit (MM) in the SIM configuration register (SIMCR) defines the most significant bit (ADDR23) of the IMB address for each M68HC16, M68CK16, and M68CM16 Z-series module. Because ADDR[23:20] are driven to the same value as ADDR19, MM must be set to one. If MM is cleared, IMB modules are inaccessible. For more information about how the state of MM affects the system, refer to **5.2.1 Module Mapping**.

The SRAM control register consists of eight bytes, but not all locations are implemented. Unimplemented register addresses are read as zeros, and writes have no effect. Refer to **D.3 Standby RAM Module** for the register block address map and register bit/field definitions.

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

Freescale Semiconductor, Inc.

6.2 SRAM Array Address Mapping

Base address registers RAMBAH and RAMBAL are used to specify the SRAM array base address in the memory map. RAMBAH and RAMBAL can only be written while the SRAM is in low-power stop mode (RAMMCR STOP = 1) and the base address lock (RAMMCR RLCK = 0) is disabled. RLCK can be written once only to a value of one; subsequent writes are ignored. This prevents accidental remapping of the array.

NOTE

In the CPU16, ADDR[23:20] follow the logic state of ADDR19. The SRAM array must not be mapped to addresses $080000–$7FFFFF, which are inaccessible to the CPU16. If mapped to these addresses, the array remains inaccessible until a reset occurs, or it is remapped outside of this range.

6.3 SRAM Array Address Space Type

The RASP[1:0] in RAMMCR determine the SRAM array address space type. The SRAM module can respond to both program and data space accesses or to program space accesses only. Because the CPU16 operates in supervisor mode only, RASP1 has no effect. **Table 6-2** shows RASP[1:0] encodings.

Table 6-2 SRAM Array Address Space Type

RASP[1:0]	Space
X0	Program and data accesses
X1	Program access only

Refer to **5.5.1.7 Function Codes** for more information concerning address space types and program/data space access. Refer to **4.6 Addressing Modes** for more information on addressing modes.

6.4 Normal Access

The array can be accessed by byte, word, or long word. A byte or aligned word access takes one bus cycle or two system clocks. A long word or misaligned word access requires two bus cycles. Refer to **5.6 Bus Operation** for more information concerning access times.

6.5 Standby and Low-Power Stop Operation

Standby and low-power modes should not be confused. Standby mode maintains the RAM array when the main MCU power supply is turned off. Low-power stop mode allows the CPU16 to control MCU power consumption by disabling unused modules.

Relative voltage levels of the MCU V_{DD} and V_{STBY} pins determine whether the SRAM is in standby mode. SRAM circuitry switches to the standby power source when V_{DD} drops below specified limits. If specified standby supply voltage levels are maintained during the transition, there is no loss of memory when switching occurs. The RAM array cannot be accessed while the SRAM module is powered from V_{STBY}. If standby operation is not desired, connect the V_{STBY} pin to V_{SS}.

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

I_{SB} (SRAM standby current) values may vary while V_{DD} transitions occur. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for standby switching and power consumption specifications.

6.6 Reset

Reset places the SRAM in low-power stop mode, enables program space access, and clears the base address registers and the register lock bit. These actions make it possible to write a new base address into the ROMBAH and ROMBAL registers.

When a synchronous reset occurs while a byte or word SRAM access is in progress, the access is completed. If reset occurs during the first word access of a long-word operation, only the first word access is completed. If reset occurs during the second word access of a long-word operation, the entire access is completed. Data being read from or written to the RAM may be corrupted by an asynchronous reset. For more information, refer to **5.7 Reset**.

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 7
MASKED ROM MODULE

The masked ROM module (MRM) is only available with the MC68HC16Z2 and the MC68HC16Z3. The MRM consists of a fixed-location control register block and an 8-Kbyte mask-programmed read-only memory array that can be mapped to any 8-Kbyte boundary in the system memory map. The MRM can be programmed to insert wait states to match slower external development memory. Access time depends upon the number of wait states specified, but can be as fast as two clock cycles. The MRM can be used for program accesses only, or for program and data accesses. Data can be read in bytes, words or long words. The MRM can be configured to support system bootstrap during reset.

7.1 MRM Register Block

There are three MRM control registers: the masked ROM module configuration register (MRMCR), and the ROM array base address registers (ROMBAH and ROMBAL). In addition, the MRM register block contains the signature registers (RSIGHI and RSIGLO), and ROM bootstrap words (ROMBS[0:3]).

The module mapping bit (MM) in the SIM configuration register (SIMCR) defines the most significant bit (ADDR23) of the IMB address for each M68HC16, M68CK16, and M68CM16 Z-series module. Because ADDR[23:20] are driven to the same value as ADDR19, MM must be set to one. If MM is cleared, IMB modules are inaccessible. For more information about how the state of MM affects the system, refer to **5.2.1 Module Mapping**.

The MRM control register block consists of 32 bytes, but not all locations are implemented. Unimplemented register addresses are read as zeros, and writes have no effect. Refer to **D.4 Masked ROM Module** for the register block address map and register bit/field definitions.

7.2 MRM Array Address Mapping

Base address registers ROMBAH and ROMBAL are used to specify the ROM array base address in the memory map. Although the base address loaded into ROMBAH and ROMBAL during reset is mask-programmed as user-specified, these registers can be written after reset to change the default array address if the base address lock bit (LOCK in MRMCR) is not masked to a value of one.

NOTE

In the CPU16, ADDR[23:20] follow the logic state of ADDR19. The MRM array must not be mapped to addresses $7FF000–$7FFFFF, which are inaccessible to the CPU16. If mapped to these addresses, the array remains inaccessible until a reset occurs, or it is remapped outside of this range.

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

The MRM array can be mapped to any 8-Kbyte boundary in the memory map, but must not overlap other module control registers (overlap makes the registers inaccessible). If the array overlaps the MRM register block, addresses in the register block are accessed instead of the corresponding ROM array addresses.

ROMBAH and ROMBAL can only be written while the ROM is in low-power stop mode (MRMCR STOP = 1) and the base address lock (MRMCR LOCK = 0) is disabled. LOCK can be written once only to a value of one; subsequent writes are ignored. This prevents accidental remapping of the array.

7.3 MRM Array Address Space Type

ASPC[1:0] in MRMCR determines ROM array address space type. The module can respond to both program and data space accesses or to program space accesses only. The default value of ASPC[1:0] is established during mask programming, but the value can be changed after reset if the LOCK bit in the MRMCR has not been masked to a value of one. Because the CPU16 operates in supervisor mode only, ASPC1 has no effect.

Table 7-1 shows ASPC[1:0] field encodings.

Table 7-1 ROM Array Space Field

ASPC[1:0]	State Specified
X0	Program and data accesses
X1	Program access only

Refer to **5.5.1.7 Function Codes** for more information concerning address space types and program/data space access. Refer to **4.6 Addressing Modes** for more information on addressing modes.

7.4 Normal Access

The array can be accessed by byte, word, or long word. A byte or aligned word access takes a minimum of one bus cycle (two system clocks). A long word or misaligned word access requires a minimum of two bus cycles.

Access time can be optimized for a particular application by inserting wait states into each access. The number of wait states inserted is determined by the value of WAIT[1:0] in the MRMCR. Two, three, four, or five clock accesses can be specified. The default value WAIT[1:0] is established during mask programming, but field value can be changed after reset if the LOCK bit in the MRMCR has not been masked to a value of one.

Table 7-2 shows WAIT[1:0] field encodings.

Freescale Semiconductor, Inc.

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

Table 7-2 Wait States Field

WAIT[1:0]	Number of Wait States	Clocks per Transfer
00	0	3
01	1	4
10	2	5
11	–1	2

Refer to **5.6 Bus Operation** for more information concerning access times.

7.5 Low-Power Stop Mode Operation

Low-power stop mode minimizes MCU power consumption. Setting the STOP bit in MRMCR places the MRM in low-power stop mode. In low-power stop mode, the array cannot be accessed. The reset state of STOP is the complement of the logic state of DATA14 during reset. Low-power stop mode is exited by clearing STOP.

7.6 ROM Signature

Signature registers RSIGHI and RSIGLO contain a user-specified mask-programmed signature pattern. A user-specified signature algorithm provides the capability to verify ROM array contents.

7.7 Reset

The state of the MRM following reset is determined by the default values programmed into the MRMCR $\overline{\text{BOOT}}$, LOCK, ASPC[1:0], and WAIT[1:0] bits. The default array base address is determined by the values programmed into ROMBAL and ROMBAH.

When the mask programmed value of the MRMCR $\overline{\text{BOOT}}$ bit is zero, the contents of MRM bootstrap words ROMBS[0:3] are used as reset vectors. When the mask programmed value of the MRMCR $\overline{\text{BOOT}}$ bit is one, reset vectors are fetched from external memory, and system integration module chip-select logic is used to assert the boot ROM select signal $\overline{\text{CSBOOT}}$. Refer to **5.9.4 Chip-Select Reset Operation** for more information concerning external boot ROM selection.

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 8
ANALOG-TO-DIGITAL CONVERTER

This section is an overview of the analog-to-digital converter module (ADC). Refer to the *ADC Reference Manual* (ADCRM/AD) for a comprehensive discussion of ADC capabilities. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for ADC timing and electrical specifications. Refer to **D.5 Analog-to-Digital Converter Module** for register address mapping and bit/field definitions.

8.1 General

The ADC is a unipolar, successive-approximation converter with eight modes of operation. It has selectable 8- or 10-bit resolution. Monotonicity is guaranteed in both modes.

A bus interface unit handles communication between the ADC and other microcontroller modules, and supplies IMB timing signals to the ADC. Special operating modes and test functions are controlled by a module configuration register (ADCMCR) and a factory test register (ADCTST).

ADC module conversion functions can be grouped into three basic subsystems: an analog front end, a digital control section, and result storage. **Figure 8-1** is a functional block diagram of the ADC module.

In addition to use as multiplexer inputs, the eight analog inputs can be used as a general-purpose digital input port (port ADA), provided signals are within logic level specification. A port data register (PORTADA) is used to access input data.

8.2 External Connections

The ADC uses 12 pins on the MCU package. Eight pins are analog inputs (which can also be used as digital inputs), two pins are dedicated analog reference connections (V_{RH} and V_{RL}), and two pins are analog supply connections (V_{DDA} and V_{SSA}).

Freescale Semiconductor, Inc.

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

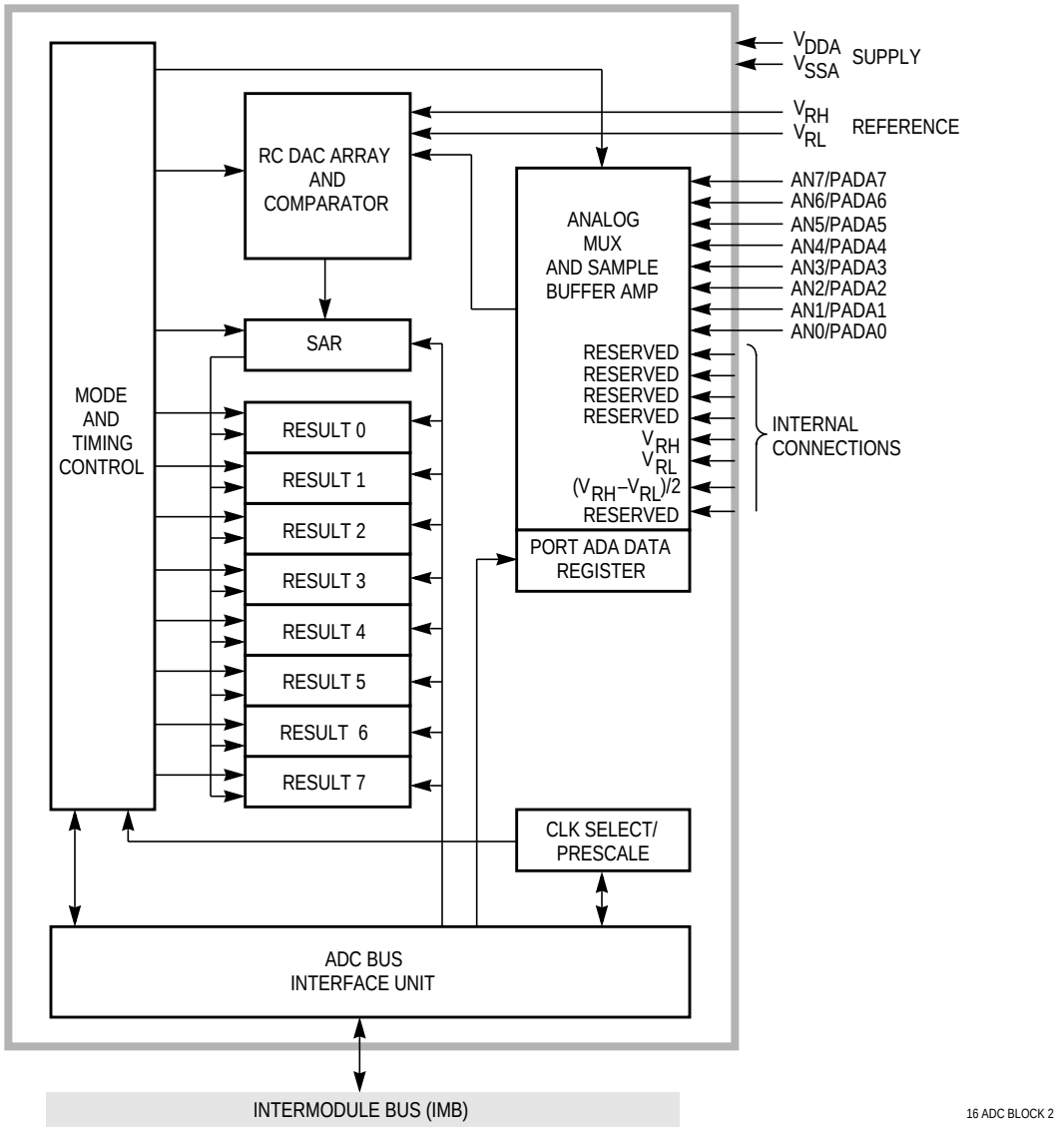


Figure 8-1 ADC Block Diagram

8.2.1 Analog Input Pins

Each of the eight analog input pins (AN[7:0]) is connected to a multiplexer in the ADC. The multiplexer selects an analog input for conversion to digital data.

Analog input pins can also be read as digital inputs, provided the applied voltage meets V_{IH} and V_{IL} specifications. When used as digital inputs, the pins are organized into an 8-bit port (PORTADA), and referred to as PADA[7:0]. There is no data direction register because port pins are input only.

Freescale Semiconductor, Inc.

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

8.2.2 Analog Reference Pins

Separate high (V_{RH}) and low (V_{RL}) analog reference voltages are connected to the analog reference pins. The pins permit connection of regulated and filtered supplies that allow the ADC to achieve its highest degree of accuracy.

8.2.3 Analog Supply Pins

Pins V_{DDA} and V_{SSA} supply power to analog circuitry associated with the RC DAC. Other circuitry in the ADC is powered from the digital power bus (pins V_{DDI} and V_{SSI}). Dedicated analog power supplies are necessary to isolate sensitive ADC circuitry from noise on the digital power bus.

8.3 Programmer's Model

The ADC module is mapped into 32 words of address space. Five words are control/status registers, one word is digital port data, and 24 words provide access to the results of AD conversion (eight addresses for each type of converted data). Two words are reserved for expansion.

The ADC module base address is determined by the value of the MM bit in the SIM configuration register (SIMCR). The base address is normally $FFF700.

Internally, the ADC has both a differential data bus and a buffered IMB data bus. Registers not directly associated with conversion functions, such as the configuration register, the test register, and the port data register, reside on the buffered bus, while conversion registers and result registers reside on the differential bus.

Registers that reside on the buffered bus are updated immediately when written. However, writes to ADC control registers abort any conversion in progress.

8.4 ADC Bus Interface Unit

The ADC is designed to act as a slave device on the intermodule bus. The ADC bus interface unit (ABIU) provides IMB bus cycle termination and synchronizes internal ADC signals with IMB signals. The ABIU also manages data bus routing to accommodate the three conversion data formats, and controls the interface to the module differential data bus.

8.5 Special Operating Modes

Low-power stop mode and freeze mode are ADC operating modes associated with assertion of IMB signals by other microcontroller modules or by external sources. These modes are controlled by the values of bits in the ADC module configuration register (ADCMCR).

8.5.1 Low-Power Stop Mode

When the STOP bit in ADCMCR is set, the IMB clock signal to the ADC is disabled. This places the module in an idle state, and power consumption is minimized. The ABIU does not shut down and ADC registers are still accessible. If a conversion is in progress when STOP is set, it is aborted.

Freescale Semiconductor, Inc.

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

STOP is set during system reset, and must be cleared before the ADC can be used. Because analog circuit bias currents are turned off during low-power stop mode, the ADC requires recovery time after STOP is cleared.

Execution of the CPU16 LPSTOP command places the entire modular microcontroller in low-power stop mode. Refer to **5.3.4 Low-Power Operation** for more information.

8.5.2 Freeze Mode

When the CPU16 in the modular microcontroller enters background debug mode, the FREEZE signal is asserted. The type of response is determined by the value of the FRZ[1:0] field in the ADCMCR. **Table 8-1** shows the different ADC responses to FREEZE assertion.

Table 8-1 FRZ Field Selection

FRZ[1:0]	Response
00	Ignore FREEZE, continue conversions
01	Reserved
10	Finish conversion in process, then freeze
11	Freeze immediately

When the ADC freezes, the ADC clock stops and all sequential activity ceases. Contents of control and status registers remain valid while frozen. When the FREEZE signal is negated, ADC activity resumes.

If the ADC freezes during a conversion, activity resumes with the next step in the conversion sequence. However, capacitors in the analog conversion circuitry discharge while the ADC is frozen; as a result, the conversion will be inaccurate.

Refer to **4.14.4 Background Debug Mode** for more information.

8.6 Analog Subsystem

The analog subsystem consists of a multiplexer, sample capacitors, a buffer amplifier, an RC DAC array, and a high-gain comparator. Comparator output sequences the successive approximation register (SAR). The interface between the comparator and the SAR is the boundary between ADC analog and digital subsystems.

8.6.1 Multiplexer

The multiplexer selects one of 16 sources for conversion. Eight sources are internal and eight are external. Multiplexer operation is controlled by channel selection field CD:CA in register ADCTL1. **Table 8-2** shows the different multiplexer channel sources. The multiplexer contains positive and negative stress protection circuitry. This circuitry prevents voltages on other input channels from affecting the current conversion.

Freescale Semiconductor, Inc.

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

Table 8-2 Multiplexer Channel Sources

[CD:CA] Value	Input Source
0000	AN0
0001	AN1
0010	AN2
0011	AN3
0100	AN4
0101	AN5
0110	AN6
0111	AN7
1000	Reserved
1001	Reserved
1010	Reserved
1011	Reserved
1100	V_{RH}
1101	V_{RL}
1110	$(V_{RH} - V_{RL}) / 2$
1111	Test/Reserved

8.6.2 Sample Capacitor and Buffer Amplifier

Each of the eight external input channels is associated with a sample capacitor and share a single sample buffer amplifier. After a conversion is initiated, the multiplexer output is connected to the sample capacitor at the input of the sample buffer amplifier for the first two ADC clock cycles of the sampling period. The sample amplifier buffers the input channel from the relatively large capacitance of the RC DAC array.

During the second two clock cycles of a sampling period, the sample capacitor is disconnected from the multiplexer, and the sample buffer amplifier charges the RC DAC array with the value stored in the sample capacitor.

During the third portion of a sampling period, both sample capacitor and buffer amplifier are bypassed, and multiplexer input charges the DAC array directly. The length of this third portion of a sampling period is determined by the value of the STS field in ADCTL0.

8.6.3 RC DAC Array

The RC DAC array consists of binary-weighted capacitors and a resistor-divider chain. The array performs two functions: it acts as a sample hold circuit during conversion, and it provides each successive digital-to-analog comparison voltage to the comparator. Conversion begins with MSB comparison and ends with LSB comparison. Array switching is controlled by the digital subsystem.

Freescale Semiconductor, Inc.

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

8.6.4 Comparator

The comparator indicates whether each approximation output from the RC DAC array during resolution is higher or lower than the sampled input voltage. Comparator output is fed to the digital control logic, which sets or clears each bit in the successive approximation register in sequence, MSB first.

8.7 Digital Control Subsystem

The digital control subsystem includes control and status registers, clock and prescaler control logic, channel and reference select logic, conversion sequence control logic, and the successive approximation register.

The subsystem controls the multiplexer and the output of the RC array during sample and conversion periods, stores the results of comparison in the successive-approximation register, then transfers results to the result registers.

8.7.1 Control/Status Registers

There are two control registers (ADCTL0, ADCTL1) and one status register (ADCSTAT). ADCTL0 controls conversion resolution, sample time, and clock/prescaler value. ADCTL1 controls analog input selection, conversion mode, and initiation of conversion. A write to ADCTL0 aborts the current conversion sequence and halts the ADC. Conversion must be restarted by writing to ADCTL1. A write to ADCTL1 aborts the current conversion sequence and starts a new sequence with parameters altered by the write. ADCSTAT shows conversion sequence status, conversion channel status, and conversion completion status.

The following paragraphs are a general discussion of control function. **D.5 Analog-to-Digital Converter Module** shows the ADC address map and discusses register bits and fields.

8.7.2 Clock and Prescaler Control

The ADC clock is derived from the system clock by a programmable prescaler. ADC clock period is determined by the value of the PRS field in ADCTL0. The prescaler has two stages. The first stage is a 5-bit modulus counter. It divides the system clock by any value from two to 32 (PRS[4:0] = %00000 to %11111). The second stage is a divide-by-two circuit. **Table 8-3** shows prescaler output values.

Freescale Semiconductor, Inc.

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

Table 8-3 Prescaler Output

PRS[4:0]	ADC Clock	Minimum System Clock	Maximum System Clock
%00000	Reserved	—	—
%00001	System Clock/4	2.0 MHz	8.4 MHz
%00010	System Clock/6	3.0 MHz	12.6 MHz
%00011	System Clock/8	4.0 MHz	16.8 MHz
…	…	…	…
%11101	System Clock/60	30.0 MHz	—
%11110	System Clock/62	31.0 MHz	—
%11111	System Clock/64	32.0 MHz	—

ADC clock speed must be between 0.5 MHz and 2.1 MHz. The reset value of the PRS field is %00011, which divides a nominal 16.78 MHz system clock by eight, yielding maximum ADC clock frequency. There are a minimum of four IMB clock cycles for each ADC clock cycle.

8.7.3 Sample Time

The first two portions of all sample periods require four ADC clock cycles. During the third portion of a sample period, the selected channel is connected directly to the RC DAC array for a specified number of clock cycles. The value of the STS field in ADCTL0 determines the number of cycles. Refer to **Table 8-4**. The number of clock cycles required for a sample period is the value specified by STS plus four. Sample time is determined by PRS value.

Table 8-4 TS Field Selection

STS[1:0]	Sample Time
00	2 A/D clock periods
01	4 A/D clock periods
10	8 A/D clock periods
11	16 A/D clock periods

8.7.4 Resolution

ADC resolution can be either eight or ten bits. Resolution is determined by the state of the RES10 bit in ADCTL0. Both 8-bit and 10-bit conversion results are automatically aligned in the result registers.

8.7.5 Conversion Control Logic

Analog-to-digital conversions are performed in sequences. Sequences are initiated by any write to ADCTL1. If a conversion sequence is already in progress, a write to either control register will abort it and reset the SCF and CCF flags in the A/D status register. There are eight conversion modes. Conversion mode is determined by ADCTL1 control bits. Each conversion mode affects the bits in status register ADCSTAT differently. Result storage differs from mode to mode.

Freescale Semiconductor, Inc.

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

8.7.5.1 Conversion Parameters

Table 8-5 describes the conversion parameters controlled by bits in ADCTL1.

Table 8-5 Conversion Parameters Controlled by ADCTL1

Conversion Parameter	Description
Conversion channel	The value of the channel selection field (CD:CA) in ADCTL1 determines which multiplexer inputs are used in a conversion sequence. There are 16 possible inputs. Seven inputs are external pins (AN[6:0]), and nine are internal.
Length of sequence	A conversion sequence consists of either four or eight conversions. The number of conversions in a sequence is determined by the state of the S8CM bit in ADCTL1.
Single or continuous conversion	Conversion can be limited to a single sequence or a sequence can be performed continuously. The state of the SCAN bit in ADCTL1 determines whether single or continuous conversion is performed.
Single or multiple channel conversion	Conversion sequence(s) can be run on a single channel or on a block of four or eight channels. Channel conversion is controlled by the state of the MULT bit in ADCTL1.

8.7.5.2 Conversion Modes

Conversion modes are defined by the state of the SCAN, MULT, and S8CM bits in ADCTL1. Table 8-6 shows mode numbering.

Table 8-6 ADC Conversion Modes

SCAN	MULT	S8CM	Mode
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

The following paragraphs describe each type of conversion mode:

Mode 0 — A single four-conversion sequence is performed on a single input channel specified by the value in CD:CA. Each result is stored in a separate result register (RSLT0 to RSLT3). The appropriate CCF bit in ADCSTAT is set as each register is filled. The SCF bit in ADCSTAT is set when the conversion sequence is complete.

Mode 1 — A single eight-conversion sequence is performed on a single input channel specified by the value in CD:CA. Each result is stored in a separate result register (RSLT0 to RSLT7). The appropriate CCF bit in ADCSTAT is set as each register is filled. The SCF bit in ADCSTAT is set when the conversion sequence is complete.

Freescale Semiconductor, Inc.

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

Mode 2 — A single conversion is performed on each of four sequential input channels, starting with the channel specified by the value in CD:CA. Each result is stored in a separate result register (RSLT0 to RSLT3). The appropriate CCF bit in ADCSTAT is set as each register is filled. The SCF bit in ADCSTAT is set when the last conversion is complete.

Mode 3 — A single conversion is performed on each of eight sequential input channels, starting with the channel specified by the value in CD:CA. Each result is stored in a separate result register (RSLT0 to RSLT7). The appropriate CCF bit in ADCSTAT is set as each register is filled. The SCF bit in ADCSTAT is set when the last conversion is complete.

Mode 4 — Continuous four-conversion sequences are performed on a single input channel specified by the value in CD:CA. Each result is stored in a separate result register (RSLT0 to RSLT3). Previous results are overwritten when a sequence repeats. The appropriate CCF bit in ADCSTAT is set as each register is filled. The SCF bit in ADCSTAT is set when the first four-conversion sequence is complete.

Mode 5 — Continuous eight-conversion sequences are performed on a single input channel specified by the value in CD:CA. Each result is stored in a separate result register (RSLT0 to RSLT7). Previous results are overwritten when a sequence repeats. The appropriate CCF bit in ADCSTAT is set as each register is filled. The SCF bit in ADCSTAT is set when the first eight-conversion sequence is complete.

Mode 6 — Continuous conversions are performed on each of four sequential input channels, starting with the channel specified by the value in CD:CA. Each result is stored in a separate result register (RSLT0 to RSLT3). The appropriate CCF bit in ADCSTAT is set as each register is filled. The SCF bit in ADCSTAT is set when the first four-conversion sequence is complete.

Mode 7 — Continuous conversions are performed on each of eight sequential input channels, starting with the channel specified by the value in CD:CA. Each result is stored in a separate result register (RSLT0 to RSLT7). The appropriate CCF bit in ADCSTAT is set as each register is filled. The SCF bit in ADCSTAT is set when the first eight-conversion sequence is complete.

Table 8-7 is a summary of ADC operation when MULT is cleared (single-channel modes). **Table 8-8** is a summary of ADC operation when MULT is set (multi-channel modes). Number of conversions per channel is determined by SCAN. Channel numbers are given in order of conversion.

Freescale Semiconductor, Inc.

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

Table 8-7 Single-Channel Conversions (MULT = 0)

S8CM	CD	CC	CB	CA	Input	Result Register[1]
0	0	0	0	0	AN0	RSLT[0:3]
0	0	0	0	1	AN1	RSLT[0:3]
0	0	0	1	0	AN2	RSLT[0:3]
0	0	0	1	1	AN3	RSLT[0:3]
0	0	1	0	0	AN4	RSLT[0:3]
0	0	1	0	1	AN5	RSLT[0:3]
0	0	1	1	0	AN6	RSLT[0:3]
0	0	1	1	1	AN7	RSLT[0:3]
0	1	0	0	0	Reserved	RSLT[0:3]
0	1	0	0	1	Reserved	RSLT[0:3]
0	1	0	1	0	Reserved	RSLT[0:3]
0	1	0	1	1	Reserved	RSLT[0:3]
0	1	1	0	0	V_{RH}	RSLT[0:3]
0	1	1	0	1	V_{RL}	RSLT[0:3]
0	1	1	1	0	$(V_{RH} - V_{RL}) / 2$	RSLT[0:3]
0	1	1	1	1	Test/Reserved	RSLT[0:3]
1	0	0	0	0	AN0	RSLT[0:7]
1	0	0	0	1	AN1	RSLT[0:7]
1	0	0	1	0	AN2	RSLT[0:7]
1	0	0	1	1	AN3	RSLT[0:7]
1	0	1	0	0	AN4	RSLT[0:7]
1	0	1	0	1	AN5	RSLT[0:7]
1	0	1	1	0	AN6	RSLT[0:7]
1	0	1	1	1	AN7	RSLT[0:7]
1	1	0	0	0	Reserved	RSLT[0:7]
1	1	0	0	1	Reserved	RSLT[0:7]
1	1	0	1	0	Reserved	RSLT[0:7]
1	1	0	1	1	Reserved	RSLT[0:7]
1	1	1	0	0	V_{RH}	RSLT[0:7]
1	1	1	0	1	V_{RL}	RSLT[0:7]
1	1	1	1	0	$(V_{RH} - V_{RL}) / 2$	RSLT[0:7]
1	1	1	1	1	Test/Reserved	RSLT[0:7]

NOTES:
1. Result register (RSLT) is either RJURRX, LJSRRX, or LJURRX, depending on the address read.

Freescale Semiconductor, Inc.

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

Table 8-8 Multiple-Channel Conversions (MULT = 1)

S8CM	CD	CC	CB	CA	Input	Result Register[1]
0	0	0	X	X	AN0	RSLT0
					AN1	RSLT1
					AN2	RSLT2
					AN3	RSLT3
0	0	1	X	X	AN4	RSLT0
					AN5	RSLT1
					AN6	RSLT2
					AN7	RSLT3
0	1	0	X	X	Reserved	RSLT0
					Reserved	RSLT1
					Reserved	RSLT2
					Reserved	RSLT3
0	1	1	X	X	V_{RH}	RSLT0
					V_{RL}	RSLT1
					$(V_{RH} - V_{RL}) / 2$	RSLT2
					Test/Reserved	RSLT3
1	0	X	X	X	AN0	RSLT0
					AN1	RSLT1
					AN2	RSLT2
					AN3	RSLT3
					AN4	RSLT4
					AN5	RSLT5
					AN6	RSLT6
					AN7	RSLT7
1	1	X	X	X	Reserved	RSLT0
					Reserved	RSLT1
					Reserved	RSLT2
					Reserved	RSLT3
					V_{RH}	RSLT4
					V_{RL}	RSLT5
					$(V_{RH} - V_{RL}) / 2$	RSLT6
					Test/Reserved	RSLT7

NOTES:
1. Result register (RSLT) is either RJURRX, LJSRRX, or LJURRX, depending on the address read.

Freescale Semiconductor, Inc.

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

8.7.6 Conversion Timing

Total conversion time is made up of initial sample time, transfer time, final sample time, and resolution time. Initial sample time is the time during which a selected input channel is connected to the sample buffer amplifier through a sample capacitor. During transfer time, the sample capacitor is disconnected from the multiplexer, and the RC DAC array is driven by the sample buffer amp. During final sampling time, the sample capacitor and amplifier are bypassed, and the multiplexer input charges the RC DAC array directly. During resolution time, the voltage in the RC DAC array is converted to a digital value, and the value is stored in the SAR.

Initial sample time and transfer time are fixed at two ADC clock cycles each. Final sample time can be 2, 4, 8, or 16 ADC clock cycles, depending on the value of the STS field in ADCTL0. Resolution time is ten cycles for 8-bit conversion and twelve cycles for 10-bit conversion.

Transfer and resolution require a minimum of 16 ADC clocks (8 µs with a 2.1 MHz ADC clock) for 8-bit resolution or 18 ADC clocks (9 µs with a 2.1 MHz ADC clock) for 10-bit resolution. If maximum final sample time (16 ADC clocks) is used, total conversion time is 15 µs for an 8-bit conversion or 16 µs for a 10-bit conversion (with a 2.1 MHz ADC clock).

Figures 8-2 and **8-3** illustrate the timing for 8- and 10-bit conversions, respectively. These diagrams assume a final sampling period of two ADC clocks.

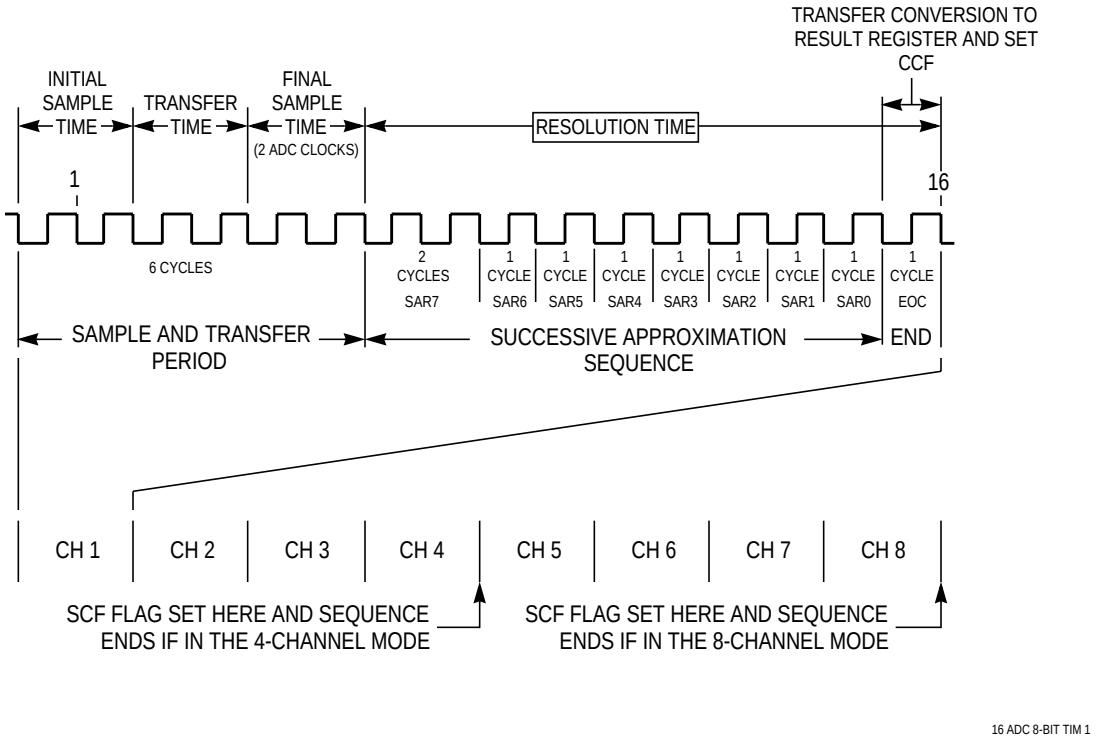


Figure 8-2 8-Bit Conversion Timing

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

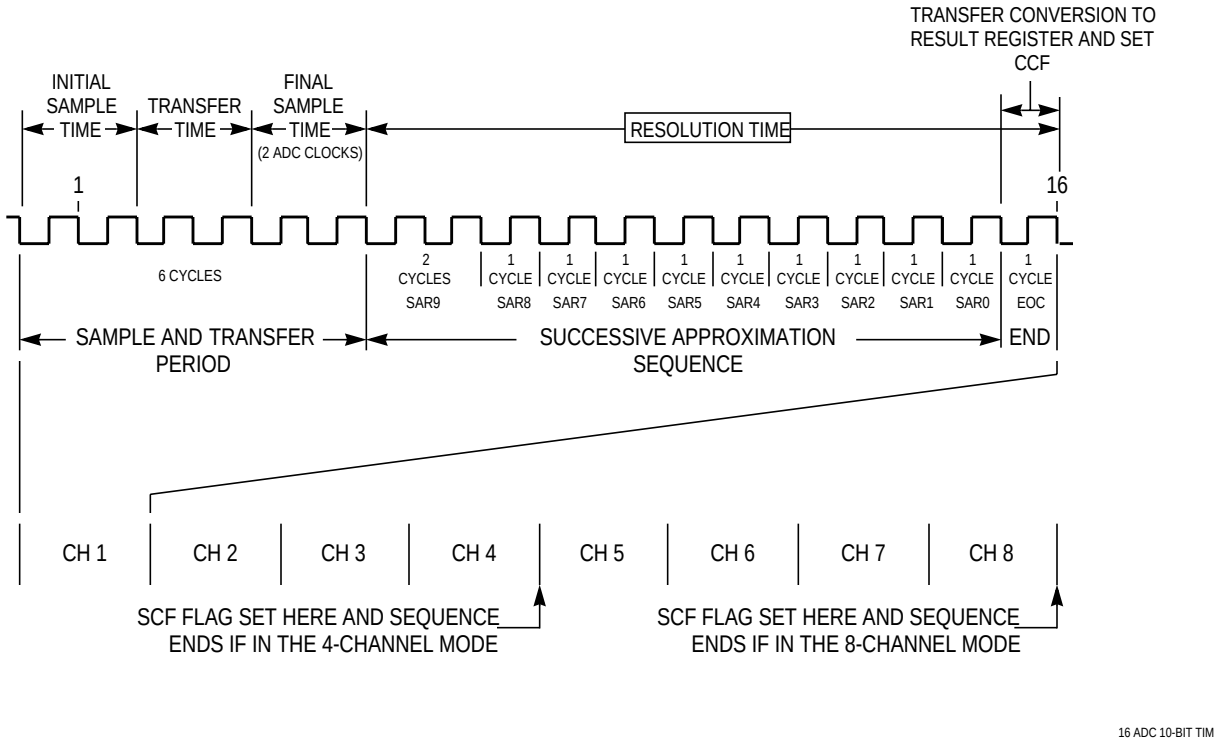


Figure 8-3 10-Bit Conversion Timing

8.7.7 Successive Approximation Register

The successive approximation register (SAR) accumulates the result of each conversion one bit at a time, starting with the most significant bit.

At the start of the resolution period, the MSB of the SAR is set, and all less significant bits are cleared. Depending on the result of the first comparison, the MSB is either left set or cleared. Each successive bit is set or left cleared in descending order until all eight or ten bits have been resolved.

When conversion is complete, the content of the SAR is transferred to the appropriate result register. Refer to **APPENDIX D REGISTER SUMMARY** for register mapping and configuration.

8.7.8 Result Registers

Result registers are used to store data after conversion is complete. The registers can be accessed from the IMB under ABIU control. Each register can be read from three different addresses in the ADC memory map. The format of the result data depends on the address from which it is read. **Table 8-9** shows the three types of formats.

Freescale Semiconductor, Inc.

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

Table 8-9 Result Register Formats

Result Data Format	Description
Unsigned right-justified format	Conversion result is unsigned right-justified data. Bits [9:0] are used for 10-bit resolution, bits [7:0] are used for 8-bit conversion (bits [9:8] are zero). Bits [15:10] always return zero when read.
Signed left-justified format	Conversion result is signed left-justified data. Bits [15:6] are used for 10-bit resolution, bits [15:8] are used for 8-bit conversion (bits [7:6] are zero). Although the ADC is unipolar, it is assumed that the zero point is $(V_{RH} - V_{RL}) / 2$ when this format is used. The value read from the register is an offset two's-complement number; for positive input, bit 15 equals zero, for negative input, bit 15 equals one. Bits [5:0] always return zero when read.
Unsigned left-justified format	Conversion result is unsigned left-justified data. Bits [15:6] are used for 10-bit resolution, bits [15:8] are used for 8-bit conversion (bits [7:6] are zero). Bits [5:0] always return zero when read.

Refer to **APPENDIX D REGISTER SUMMARY** for register mapping and configuration.

8.8 Pin Considerations

The ADC requires accurate, noise-free input signals for proper operation. The following sections discuss the design of external circuitry to maximize ADC performance.

8.8.1 Analog Reference Pins

No A/D converter can be more accurate than its analog reference. Any noise in the reference can result in at least that much error in a conversion. The reference for the ADC, supplied by pins V_{RH} and V_{RL}, should be low-pass filtered from its source to obtain a noise-free, clean signal. In many cases, simple capacitive bypassing may suffice. In extreme cases, inductors or ferrite beads may be necessary if noise or RF energy is present. Series resistance is not advisable since there is an effective DC current requirement from the reference voltage by the internal resistor string in the RC DAC array. External resistance may introduce error in this architecture under certain conditions. Any series devices in the filter network should contain a minimum amount of DC resistance.

For accurate conversion results, the analog reference voltages must be within the limits defined by V_{DDA} and V_{SSA}, as explained in the following subsection.

8.8.2 Analog Power Pins

The analog supply pins (V_{DDA} and V_{SSA}) define the limits of the analog reference voltages (V_{RH} and V_{RL}) and of the analog multiplexer inputs. **Figure 8-4** is a diagram of the analog input circuitry.

Freescale Semiconductor, Inc.

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

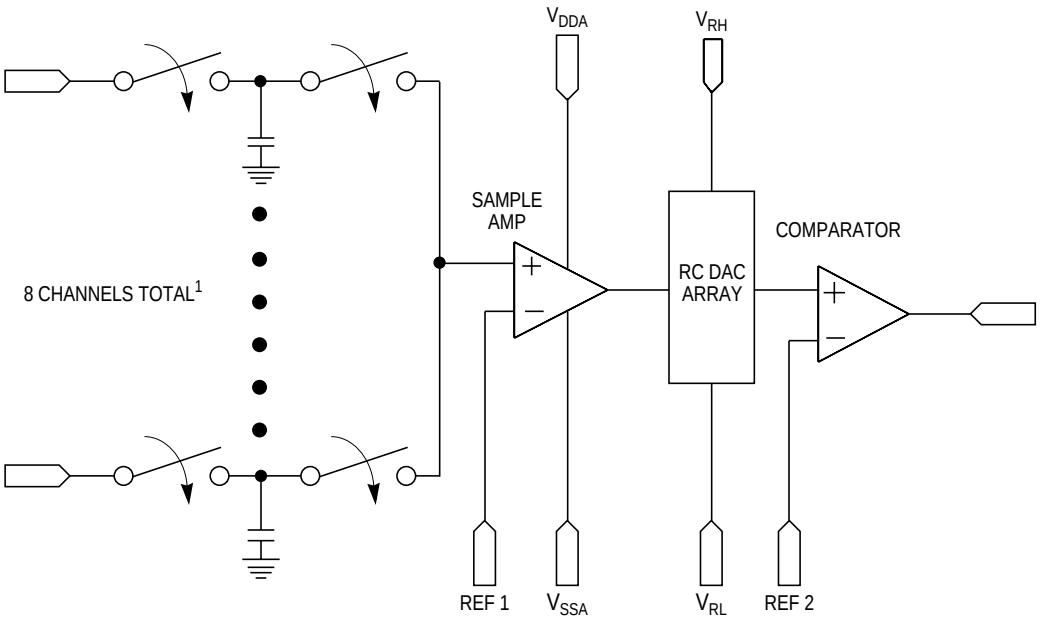


Figure 8-4 Analog Input Circuitry

Since the sample amplifier is powered by V_{DDA} and V_{SSA}, it can accurately transfer input signal levels up to but not exceeding V_{DDA} and down to but not below V_{SSA}. If the input signal is outside of this range, the output from the sample amplifier is clipped.

In addition, V_{RH} and V_{RL} must be within the range defined by V_{DDA} and V_{SSA}. As long as V_{RH} is less than or equal to V_{DDA}, and V_{RL} is greater than or equal to V_{SSA}, and the sample amplifier has accurately transferred the input signal, resolution is ratiometric within the limits defined by V_{RL} and V_{RH}. If V_{RH} is greater than V_{DDA}, the sample amplifier can never transfer a full-scale value. If V_{RL} is less than V_{SSA}, the sample amplifier can never transfer a zero value.

Figure 8-5 shows the results of reference voltages outside the range defined by V_{DDA} and V_{SSA}. At the top of the input signal range, V_{DDA} is 10 mV lower than V_{RH}. This results in a maximum obtainable 10-bit conversion value of 3FE. At the bottom of the signal range, V_{SSA} is 15 mV higher than V_{RL}, resulting in a minimum obtainable 10-bit conversion value of three.

Freescale Semiconductor, Inc.

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

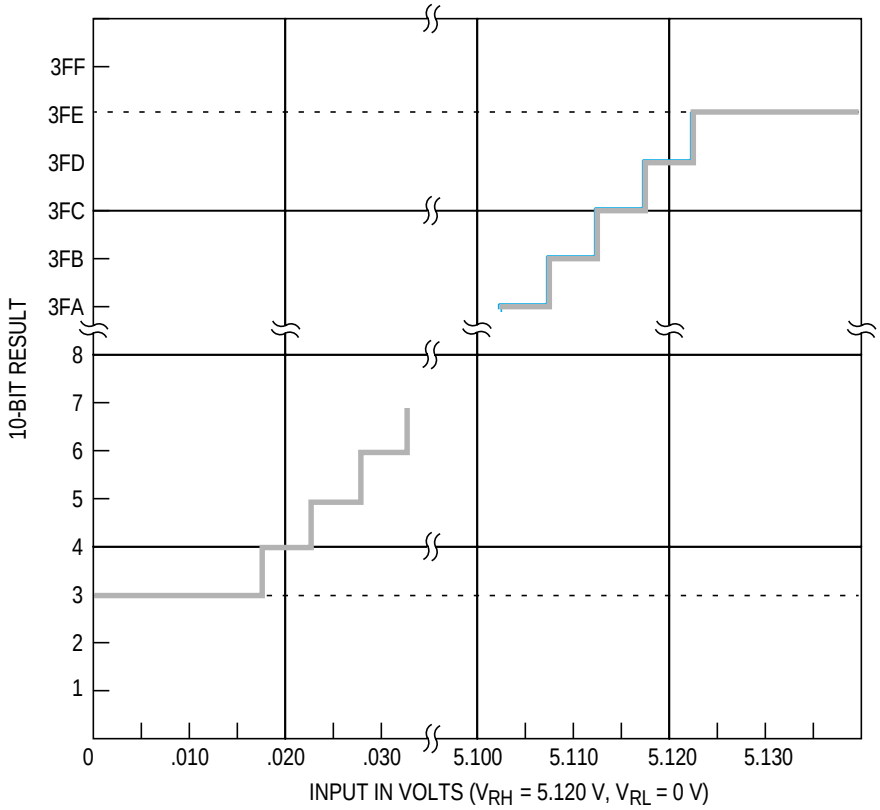


Figure 8-5 Errors Resulting from Clipping

8.8.3 Analog Supply Filtering and Grounding

Two important factors influencing performance in analog integrated circuits are supply filtering and grounding. Generally, digital circuits use bypass capacitors on every V_{DD}/V_{SS} pin pair. This applies to analog subsystems or submodules also. Equally important as bypassing is the distribution of power and ground.

Analog supplies should be isolated from digital supplies as much as possible. This necessity stems from the higher performance requirements often associated with analog circuits. Therefore, deriving an analog supply from a local digital supply is not recommended. However, if for economic reasons digital and analog power are derived from a common regulator, filtering of the analog power is recommended in addition to the bypassing of the supplies already mentioned. For example, an RC low-pass filter could be used to isolate the digital and analog supplies when generated by a common regulator. If multiple high precision analog circuits are locally employed (such as two A/D converters), the analog supplies should be isolated from each other, as sharing supplies introduces the potential for interference between analog circuits.

Freescale Semiconductor, Inc.

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

Grounding is the most important factor influencing analog circuit performance in mixed signal systems (or in stand-alone analog systems). Close attention must be paid to avoid introducing additional sources of noise into the analog circuitry. Common sources of noise include ground loops, inductive coupling, and combining digital and analog grounds together inappropriately.

The problem of how and when to combine digital and analog grounds arises from the large transients which the digital ground must handle. If the digital ground is not able to handle the large transients, the current from the large transients can return to ground through the analog ground. It is the excess current overflowing into the analog ground which causes performance degradation by developing a differential voltage between the true analog ground and the microcontroller's ground pin. The end result is that the ground observed by the analog circuit is no longer true ground and often ends in skewed results.

Two similar approaches designed to improve or eliminate the problems associated with grounding excess transient currents involve star-point ground systems. One approach is to star-point the different grounds at the power supply origin, thus keeping the ground isolated. Refer to **Figure 8-6**.

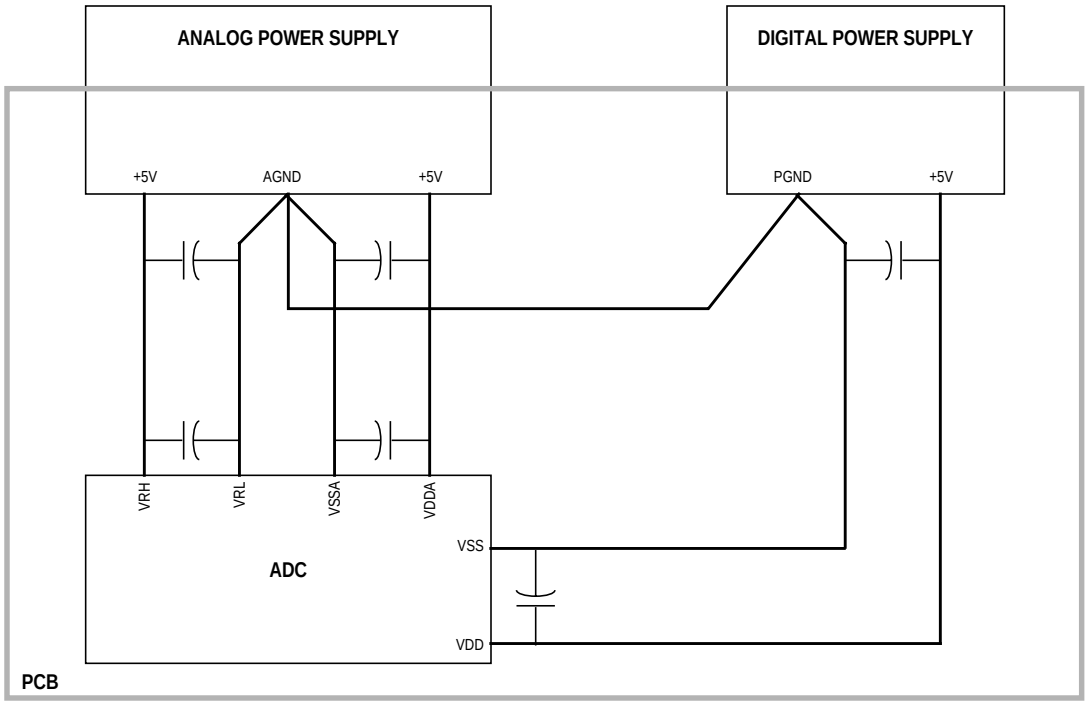


Figure 8-6 Star-Ground at the Point of Power Supply Origin

Another approach is to star-point the different grounds near the analog ground pin on the microcontroller by using small traces for connecting the non-analog grounds to the analog ground. The small traces are meant only to accommodate DC differences, not AC transients.

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

Freescale Semiconductor, Inc.

NOTE

This star-point scheme still requires adequate grounding for digital and analog subsystems in addition to the star-point ground.

Other suggestions for PCB layout in which the ADC is employed include the following:

- The analog ground must be low impedance to all analog ground points in the circuit.
- Bypass capacitors should be as close to the power pins as possible.
- The analog ground should be isolated from the digital ground. This can be done by cutting a separate ground plane for the analog ground.
- Non-minimum traces should be utilized for connecting bypass capacitors and filters to their corresponding ground/power points.
- Minimum distance for trace runs when possible.

8.8.4 Accommodating Positive/Negative Stress Conditions

Positive or negative stress refers to conditions which exceed nominally defined operating limits. Examples include applying a voltage exceeding the normal limit on an input (for example, voltages outside of the suggested supply/reference ranges) or causing currents into or out of the pin which exceed normal limits. ADC specific considerations are voltages greater than V_{DDA}, V_{RH} or less than V_{SSA} applied to an analog input which cause excessive currents into or out of the input. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** on exact magnitudes.

Both stress conditions can potentially disrupt conversion results on neighboring inputs. Parasitic devices, associated with CMOS processes, can cause an immediate disruptive influence on neighboring pins. Common examples of parasitic devices are diodes to substrate and bipolar devices with the base terminal tied to substrate (V_{SSI}/V_{SSA} ground). Under stress conditions, current introduced on an adjacent pin can cause errors on adjacent channels by developing a voltage drop across the adjacent external channel source impedances.

Figure 8-7 shows an active parasitic bipolar when an input pin is subjected to negative stress conditions. Positive stress conditions do not activate a similar parasitic device.

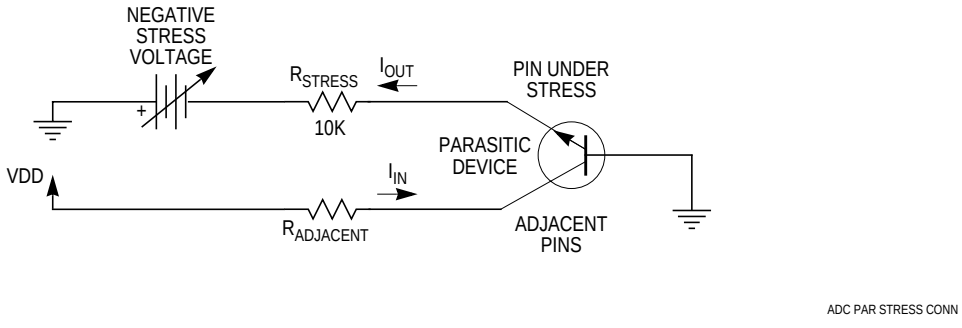


Figure 8-7 Input Pin Subjected to Negative Stress

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

The current out of the pin (I_{OUT}) under negative stress is determined by the following equation:

$$I_{OUT} = \frac{|V_{STRESS} - V_{BE}|}{R_{STRESS}}$$

where:

V_{STRESS} = Adjustable voltage source

V_{BE} = Parasitic bipolar base/emitter voltage (refer to $V_{NEGCLAMP}$ in **APPENDIX A ELECTRICAL CHARACTERISTICS**)

R_{STRESS} = Source impedance (10K resistor in **Figure 8-7** on stressed channel)

The current into (I_{IN}) the neighboring pin is determined by the $1/K_N$ (Gain) of the parasitic bipolar transistor ($1/K_N << 1$).

One way to minimize the impact of stress conditions on the ADC is to apply voltage limiting circuits such as diodes to supply and ground. However, leakage from such circuits and the potential influence on the sampled voltage to be converted must be considered. Refer to **Figure 8-8**.

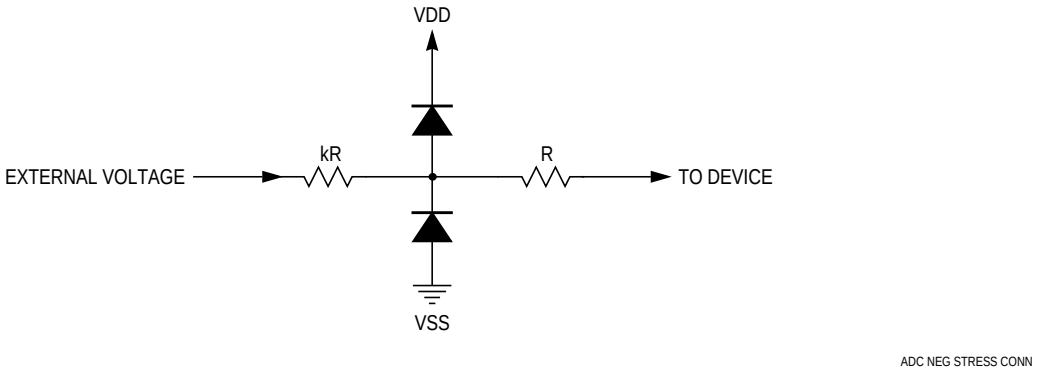


Figure 8-8 Voltage Limiting Diodes in a Negative Stress Circuit

Another method for minimizing the impact of stress conditions on the ADC is to strategically allocate ADC inputs so that the lower accuracy inputs are adjacent to the inputs most likely to see stress conditions.

Finally, suitable source impedances should be selected to meet design goals and minimize the effect of stress conditions.

8.8.5 Analog Input Considerations

The source impedance of the analog signal to be measured and any intermediate filtering should be considered whether external multiplexing is used or not. **Figure 8-9** shows the connection of eight typical analog signal sources to one ADC analog input pin through a separate multiplexer chip. Also, an example of an analog signal source connected directly to a ADC analog input channel is displayed.

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

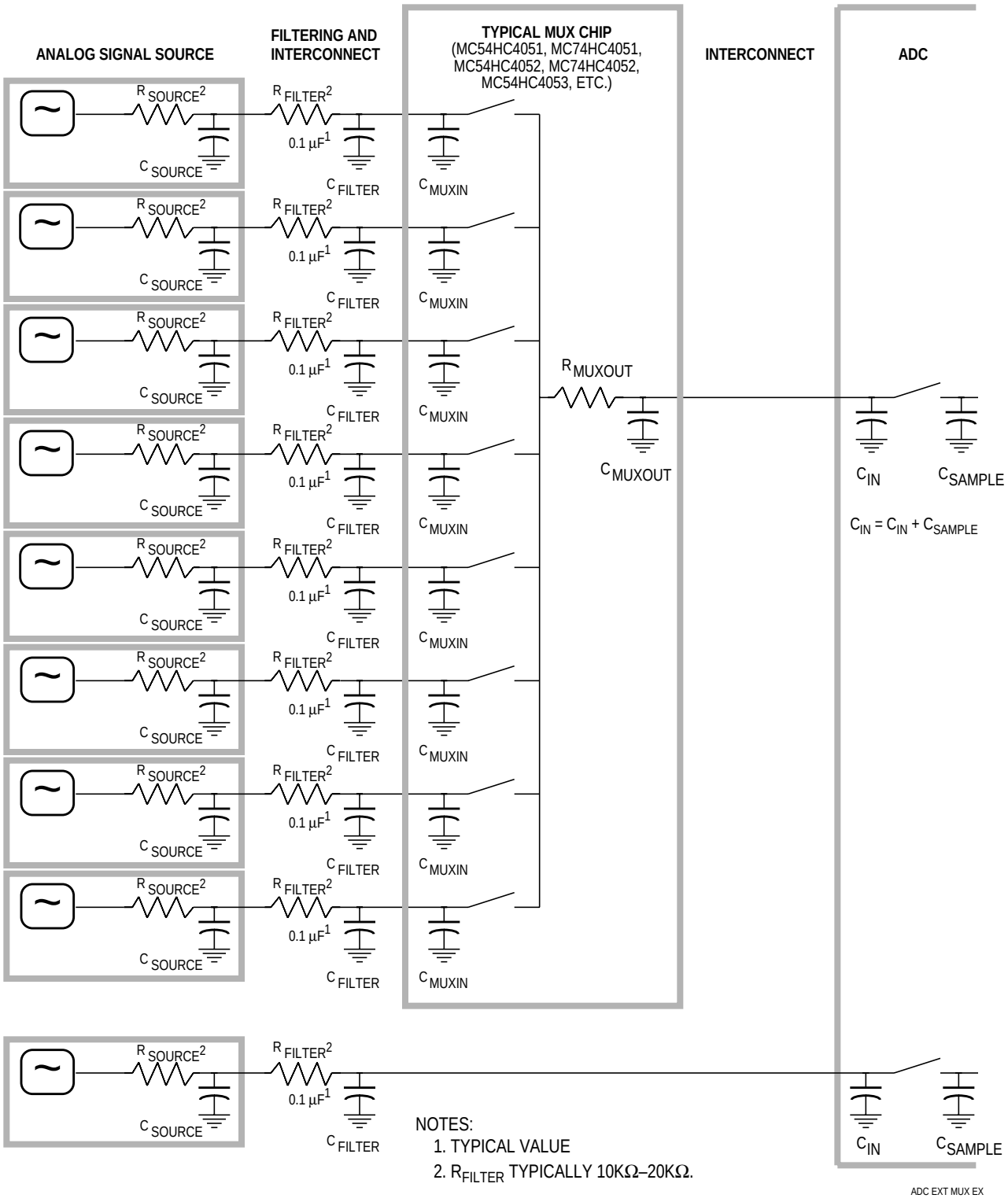


Figure 8-9 External Multiplexing of Analog Signal Sources

Freescale Semiconductor, Inc.

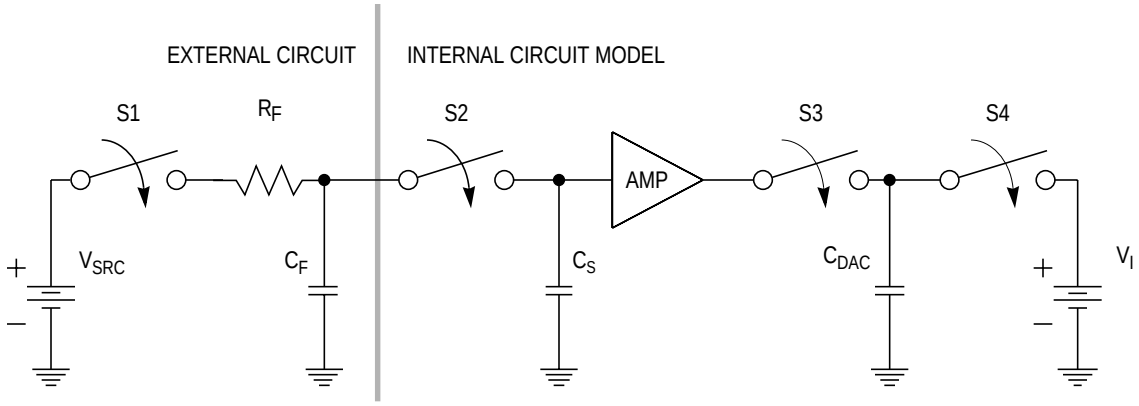
For More Information On This Product,
Go to: www.freescale.com

8.8.6 Analog Input Pins

Analog inputs should have low AC impedance at the pins. Low AC impedance can be realized by placing a capacitor with good high frequency characteristics at the input pin of the part. Ideally, that capacitor should be as large as possible (within the practical range of capacitors that still have good high frequency characteristics). This capacitor has two effects. First, it helps attenuate any noise that may exist on the input. Second, it sources charge during the sample period when the analog signal source is a high-impedance source.

Series resistance can be used with the capacitor on an input pin to implement a simple RC filter. The maximum level of filtering at the input pins is application dependent and is based on the bandpass characteristics required to accurately track the dynamic characteristics of an input. Simple RC filtering at the pin may be limited by the source impedance of the transducer or circuit supplying the analog signal to be measured. Refer to **8.8.6.2 Error Resulting from Leakage**. In some cases, the size of the capacitor at the pin may be very small.

Figure 8-10 is a simplified model of an input channel. Refer to this model in the following discussion of the interaction between the user's external circuitry and the circuitry inside the ADC.



V_{SRC} = SOURCE VOLTAGE
R_F = FILTER IMPEDANCE (SOURCE IMPEDANCE INCLUDED)
C_F = FILTER CAPACITOR
C_S = INTERNAL CAPACITANCE (FOR A BYPASSED CHANNEL, THIS IS THE C_{DAC} CAPACITANCE)
C_{DAC} = DAC CAPACITOR ARRAY
V_I = INTERNAL VOLTAGE SOURCE FOR PRECHARGE ($V_{DDA}/2$)

ADC SAMPLE AMP MODEL

Figure 8-10 Electrical Model of an A/D Input Pin

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

In **Figure 8-10**, R_F and C_F comprise the user's external filter circuit. C_S is the internal sample capacitor. Each channel has its own capacitor. C_S is never precharged; it retains the value of the last sample. V_l is an internal voltage source used to precharge the DAC capacitor array (C_{DAC}) before each sample. The value of this supply is V_{DDA}/2, or 2.5 volts for 5-volt operation.

The following paragraphs provide a simplified description of the interaction between the ADC and the user's external circuitry. This circuitry is assumed to be a simple RC low-pass filter passing a signal from a source to the ADC input pin. The following simplifying assumptions are made:

- The source impedance is included with the series resistor of the RC filter.
- The external capacitor is perfect (no leakage, no significant dielectric absorption characteristics, etc.)
- All parasitic capacitance associated with the input pin is included in the value of the external capacitor.
- Inductance is ignored.
- The “on” resistance of the internal switches is zero ohms and the “off” resistance is infinite.

8.8.6.1 Settling Time for the External Circuit

The values for R_F and C_F in the user's external circuitry determine the length of time required to charge C_F to the source voltage level (V_{SRC}).

At time t = 0, S1 in **Figure 8-10** closes. S2 is open, disconnecting the internal circuitry from the external circuitry. Assume that the initial voltage across CF is zero. As CF charges, the voltage across it is determined by the following equation, where t is the total charge time:

$$V_{CF} = V_{SRC}(1 - e^{-t/R_F C_F})$$

When t = 0, the voltage across C_F = 0. As t approaches infinity, V_{CF} will equal V_{SRC}. (This assumes no internal leakage.) With 10-bit resolution, 1/2 of a count is equal to 1/2048 full-scale value. Assuming worst case (V_{SRC} = full scale), **Table 8-10** shows the required time for C_F to charge to within 1/2 of a count of the actual source voltage during 10-bit conversions. **Table 8-10** is based on the RC network in **Figure 8-10**.

NOTE

The following times are completely independent of the A/D converter architecture (assuming the ADC is not affecting the charging).

Freescale Semiconductor, Inc.

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

Table 8-10 External Circuit Settling Time (10-Bit Conversions)

Filter Capacitor (C_F)	Source Resistance (R_F)			
	100 Ω	1 kΩ	10 kΩ	100 kΩ
1 µF	760 µs	7.6 ms	76 ms	760 ms
.1 µF	76 µs	760 µs	7.6 ms	76 ms
.01 µF	7.6 µs	76 µs	760 µs	7.6 ms
.001 µF	760 ns	7.6 µs	76 µs	760 µs
100 pF	76 ns	760 ns	7.6 µs	76 µs

The external circuit described in **Table 8-10** is a low-pass filter. A user interested in measuring an AC component of the external signal must take the characteristics of this filter into account.

8.8.6.2 Error Resulting from Leakage

A series resistor limits the current to a pin, therefore input leakage acting through a large source impedance can degrade A/D accuracy. The maximum input leakage current is specified in **APPENDIX A ELECTRICAL CHARACTERISTICS**. Input leakage is greatest at high operating temperatures and as a general rule decreases by one half for each 10°C decrease in temperature.

Assuming $V_{RH} - V_{RL}$ = 5.12 V, 1 count (assuming 10-bit resolution) corresponds to 5 mV of input voltage. A typical input leakage of 50 nA acting through 100 kΩ of external series resistance results in an error of less than 1 count (5.0 mV). If the source impedance is 1 MΩ and a typical leakage of 50 nA is present, an error of 10 counts (50 mV) is introduced.

In addition to internal junction leakage, external leakage (e.g., if external clamping diodes are used) and charge sharing effects with internal capacitors also contribute to the total leakage current. **Table 8-11** illustrates the effect of different levels of total leakage on accuracy for different values of source impedance. The error is listed in terms of 10-bit counts.

CAUTION

Leakage from the part of 10 nA is obtainable only within a limited temperature range.

Table 8-11 Error Resulting From Input Leakage (IOFF)

Source Impedance	Leakage Value (10-Bit Conversions)			
	10 nA	50 nA	100 nA	1000 nA
1 kΩ	—	—	—	0.2 counts
10 kΩ	—	0.1 counts	0.2 counts	2 counts
100 kΩ	0.2 counts	1 count	2 counts	20 counts

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
QUEUED SERIAL MODULE

This section is an overview of the queued serial module (QSM). Refer to the *QSM Reference Manual* (QSMRM/AD) for complete information about the QSM.

9.1 General

The QSM contains two serial interfaces: the queued serial peripheral interface (QSPI) and the serial communication interface (SCI). **Figure 9-1** is a block diagram of the QSM. The QSM is present on the MC68HC16Z1, MC68CK16Z1, MC68CM16Z1, MC68HC16Z2, and MC68HC16Z3 microcontrollers.

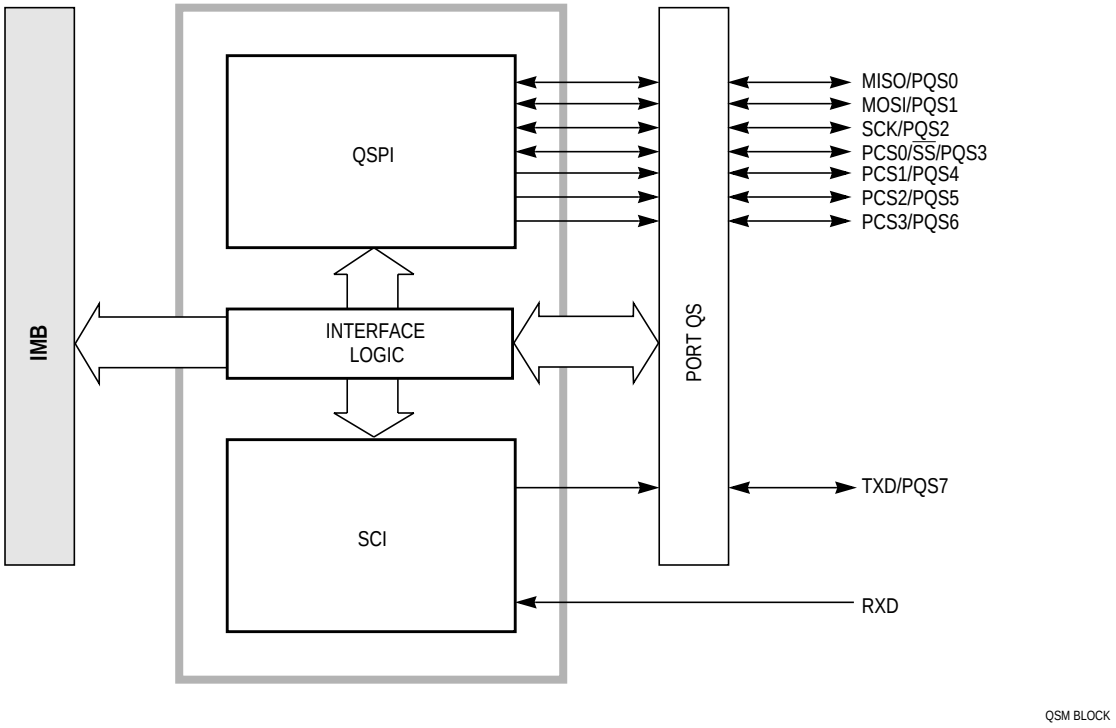


Figure 9-1 QSM Block Diagram

The QSPI provides peripheral expansion or interprocessor communication through a full-duplex, synchronous, three-line bus. Four programmable peripheral chip-selects can select up to sixteen peripheral devices by using an external 1 of 16 line selector. A self-contained RAM queue allows up to sixteen serial transfers of eight to sixteen bits each or continuous transmission of up to a 256-bit data stream without CPU16 intervention. A special wrap-around mode supports continuous transmission/reception of data.

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

Freescale Semiconductor, Inc.

The SCI provides a standard non-return to zero (NRZ) mark/space format. It operates in either full- or half-duplex mode. There are separate transmitter and receiver enable bits and dual data buffers. A modulus-type baud rate generator provides rates from 110 baud to 781 kbaud with a 25.17 MHz system clock. Word length of either eight or nine bits is software selectable. Optional parity generation and detection provide either even or odd parity check capability. Advanced error detection circuitry catches glitches of up to 1/16 of a bit time in duration. Wake-up functions allow the CPU16 to run uninterrupted until meaningful data is available.

9.2 QSM Registers and Address Map

There are four types of QSM registers: QSM global registers, QSM pin control registers, QSPI registers, and SCI registers. Refer to **9.2.1 QSM Global Registers** and **9.2.2 QSM Pin Control Registers** for a discussion of global and pin control registers. Refer to **9.3.1 QSPI Registers** and **9.4.1 SCI Registers** for further information about QSPI and SCI registers. Writes to unimplemented register bits have no effect, and reads of unimplemented bits always return zero.

Refer to **D.6 Queued Serial Module** for a QSM address map and register bit and field definitions. Refer to **5.2.1 Module Mapping** for more information about how the state of MM affects the system.

9.2.1 QSM Global Registers

The QSM configuration register (QSMCR) controls the interface between the QSM and the intermodule bus. The QSM test register (QTEST) is used during factory test of the QSM. The QSM interrupt level register (QILR) determines the priority of interrupts requested by the QSM and the vector used when an interrupt is acknowledged. The QSM interrupt vector register (QIVR) contains the interrupt vector for both QSM submodules. QILR and QIVR are 8-bit registers located at the same word address.

9.2.1.1 Low-Power Stop Mode Operation

When the STOP bit in QSMCR is set, the system clock input to the QSM is disabled and the module enters low-power stop mode. QSMCR is the only register guaranteed to be readable while STOP is asserted. The QSPI RAM is not readable in low-power stop mode. However, writes to RAM or any register are guaranteed valid while STOP is asserted. STOP can be set by the CPU16 and by reset.

The QSPI and SCI must be brought to an orderly stop before asserting STOP to avoid data corruption. To accomplish this, disable QSM interrupts or set the interrupt priority level mask in the CPU16 condition code register to a value higher than the $\overline{\text{IRQ}}$ level requested by the QSM. The SCI receiver and transmitter should be disabled after transfers in progress are complete. The QSPI can be halted by setting the HALT bit in SPCR3 and then setting STOP after the HALTA flag is set. Refer to **5.3.4 Low-Power Operation** for more information about low-power stop mode.

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

9.2.1.2 Freeze Operation

The freeze FRZ[1:0] bits in QSMCR are used to determine what action is taken by the QSM when the IMB FREEZE signal is asserted. FREEZE is asserted when the CPU16 enters background debug mode. At the present time, FRZ0 has no effect; setting FRZ1 causes the QSPI to halt on the first transfer boundary following FREEZE assertion. Refer to **4.14.4 Background Debug Mode** for more information about background debug mode.

9.2.1.3 QSM Interrupts

Both the QSPI and SCI can generate interrupt requests. Each has a separate interrupt request priority register. A single vector register is used to generate exception vector numbers.

The values of the ILQSPI and ILSCI fields in QILR determine the priority of QSPI and SCI interrupt requests. The values in these fields correspond to internal interrupt request signals $\overline{IRQ}[7:1]$. A value of %111 causes $\overline{IRQ7}$ to be asserted when a QSM interrupt request is made. Lower field values cause correspondingly lower-numbered interrupt request signals to be asserted. Setting the ILQSPI or ILSCI field values to %000 disables interrupts for the QSPI and the SCI respectively. If ILQSPI and ILSCI have the same non-zero value, and the QSPI and SCI make simultaneous interrupt requests, the QSPI has priority.

When the CPU16 acknowledges an interrupt request, it places the value in the condition code register interrupt priority (IP) mask on ADDR[3:1]. The QSM compares the IP mask value to the priority of the interrupt request to determine whether it should contend for arbitration. QSM arbitration priority is determined by the value of the IARB field in QSMCR. Each module that can generate interrupt requests must have a non-zero IARB value, otherwise the CPU16 will identify any such interrupt requests as spurious and take a spurious interrupt exception. Arbitration is performed by means of serial contention between values stored in individual module IARB fields.

When the QSM wins interrupt arbitration, it responds to the CPU16 interrupt acknowledge cycle by placing an interrupt vector number on the data bus. The vector number is used to calculate displacement into the CPU16 exception vector table. SCI and QSPI vector numbers are generated from the value in the QIVR INTV field. The values of bits INTV[7:1] are the same for both the QSPI and the SCI. The value of INTV0 is supplied by the QSM when an interrupt request is made. INTV0 = 0 for SCI interrupt requests; INTV0 = 1 for QSPI interrupt requests.

At reset, INTV[7:0] is initialized to $0F, the uninitialized interrupt vector number. To enable interrupt-driven serial communication, a user-defined vector number must be written to QIVR, and interrupt handler routines must be located at the addresses pointed to by the corresponding vector. Writes to INTV0 have no effect. Reads of INTV0 return a value of one.

Refer to **SECTION 4 CENTRAL PROCESSOR UNIT** and **SECTION 5 SYSTEM INTEGRATION MODULE** for more information about exceptions and interrupts.

Freescale Semiconductor, Inc.

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

9.2.2 QSM Pin Control Registers

The QSM uses nine pins. Eight of the pins can be used for serial communication or for parallel I/O. Clearing a bit in the port QS pin assignment register (PQSPAR) assigns the corresponding pin to general-purpose I/O; setting a bit assigns the pin to the QSPI. PQSPAR does not affect operation of the SCI.

The port QS data direction register (DDRQS) determines whether pins are inputs or outputs. Clearing a bit makes the corresponding pin an input; setting a bit makes the pin an output. DDRQS affects both QSPI function and I/O function. DDQS7 determines the direction of the TXD pin only when the SCI transmitter is disabled. When the SCI transmitter is enabled, the TXD pin is an output.

The port QS data register (PORTQS) latches I/O data. PORTQS writes drive pins defined as outputs. PORTQS reads return data present on the pins. To avoid driving undefined data, first write PORTQS, then configure DDRQS.

PQSPAR and DDRQS are 8-bit registers located at the same word address. Refer to **Table 9-1** for a summary of QSM pin functions.

Table 9-1 Effect of DDRQS on QSM Pin Function

QSM Pin	QSPI Mode	DDRQS Bit	Bit State	Pin Function
MISO	Master	DDQS0	0	Serial data input to QSPI
			1	Disables data input
	Slave		0	Disables data output
			1	Serial data output from QSPI
MOSI	Master	DDQS1	0	Disables data output
			1	Serial data output from QSPI
	Slave		0	Serial data input to QSPI
			1	Disables data input
SCK[1]	Master	DDQS2	—	Clock output from QSPI
	Slave		—	Clock input to QSPI
PCS0/$\overline{SS}$	Master	DDQS3	0	Assertion causes mode fault
			1	Chip-select output
	Slave		0	QSPI slave select input
			1	Disables slave select input
PCS[1:3]	Master	DDQS[4:6]	0	Disables chip-select output
			1	Chip-select output
	Slave		0	Inactive
			1	Inactive
TXD[2]	—	DDQS7	X	Serial data output from SCI
RXD	—	None	NA	Serial data input to SCI

NOTES:
1. PQS2 is a digital I/O pin unless the SPI is enabled (SPE set in SPCR1), in which case it becomes the QSPI serial clock SCK.
2. PQS7 is a digital I/O pin unless the SCI transmitter is enabled (TE set in SCCR1), in which case it becomes the SCI serial data output TXD.

Freescale Semiconductor, Inc.

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

9.3 Queued Serial Peripheral Interface

The queued serial peripheral interface (QSPI) is used to communicate with external devices through a synchronous serial bus. The QSPI is fully compatible with SPI systems found on other Freescale products, but has enhanced capabilities. The QSPI can perform full duplex three-wire or half duplex two-wire transfers. A variety of transfer rates, clocking, and interrupt-driven communication options is available.

Figure 9-2 displays a block diagram of the QSPI.

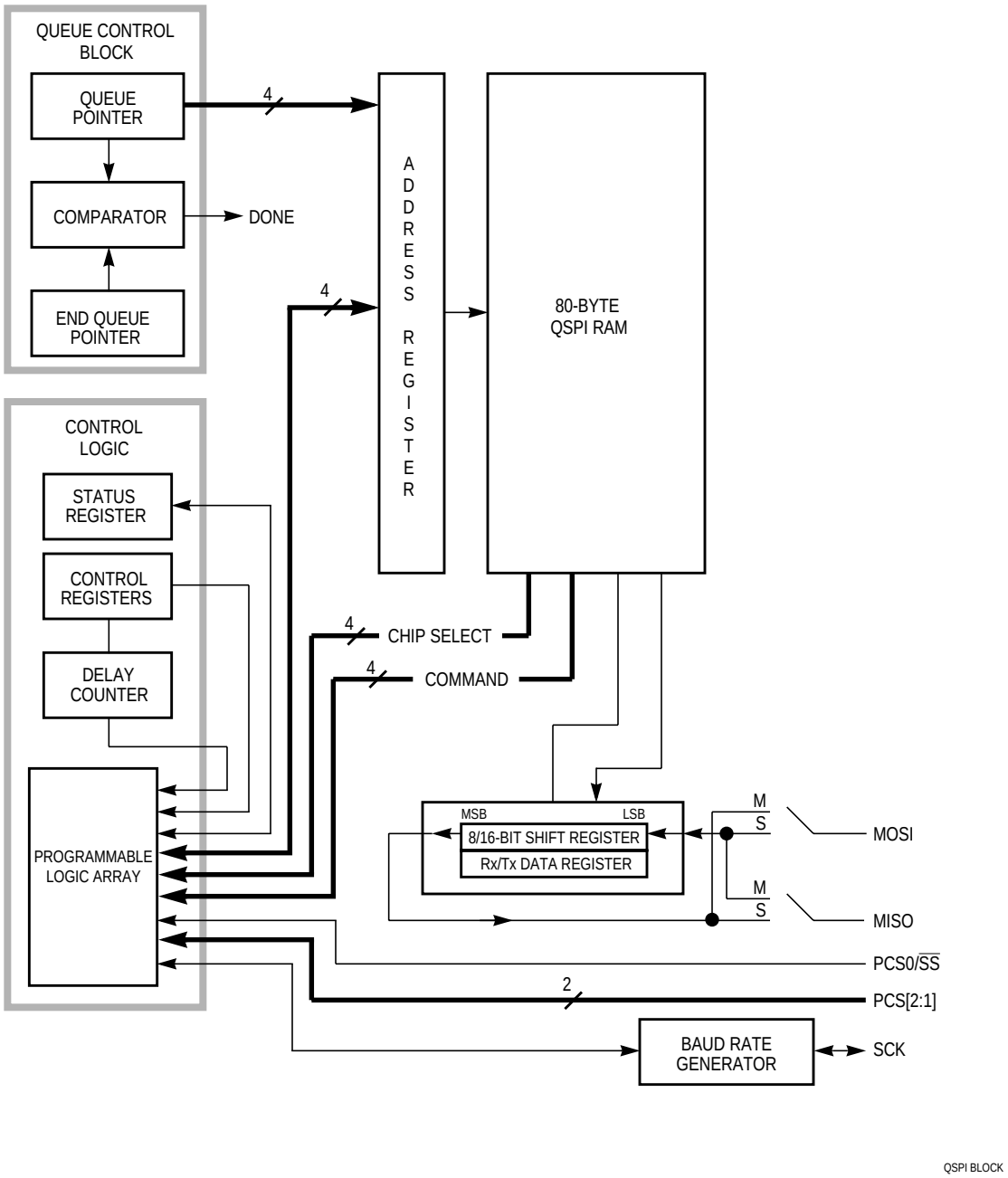


Figure 9-2 QSPI Block Diagram

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

The serial transfer length is programmable from eight to sixteen bits, inclusive. An inter-transfer delay of 17 to 8192 system clocks can be specified (default is 17 system clocks).

A dedicated 80-byte RAM is used to store received data, data to be transmitted, and a queue of commands. The CPU16 can access these locations directly.

The command queue allows the QSPI to perform up to 16 serial transfers without CPU16 intervention. Each queue entry contains all the information needed by the QSPI to independently complete one serial transfer.

A pointer identifies the queue location containing the data and command for the next serial transfer. Normally, the pointer address is incremented after each serial transfer, but the CPU16 can change the pointer value at any time. Support for multiple-tasks can be provided by segmenting the queue.

The QSPI has four peripheral chip-select pins. The chip-select signals simplify interfacing by reducing CPU16 intervention. If the chip-select signals are externally decoded, 16 independent select signals can be generated.

Wrap-around mode allows continuous execution of queued commands. In wrap-around mode, newly received data replaces previously received data in the receive RAM. Wrap-around mode can simplify the interface with A/D converters by continuously updating conversion values stored in the RAM.

Continuous transfer mode allows an uninterrupted bit stream of eight to 256 bits in length to be transferred without CPU16 intervention. Longer transfers are possible, but minimal intervention is required to prevent loss of data. A standard delay of 17 system clocks is inserted between the transfer of each queue entry.

9.3.1 QSPI Registers

The programmer's model for the QSPI consists of the QSM global and pin control registers, four QSPI control registers (SPCR[0:3]), the status register (SPSR), and the 80-byte QSPI RAM. Registers and RAM can be read and written by the CPU16. Refer to **D.6 Queued Serial Module** for register bit and field definitions.

9.3.1.1 Control Registers

Control registers contain parameters for configuring the QSPI and enabling various modes of operation. The CPU16 has read and write access to all control registers. The QSM has read access only to all bits except the SPE bit in SPCR1. Control registers must be initialized before the QSPI is enabled to ensure proper operation. SPCR1 must be written last because it contains the QSPI enable bit (SPE).

Writing a new value to any control register except SPCR2 while the QSPI is enabled disrupts operation. SPCR2 is buffered. New SPCR2 values become effective after completion of the current serial transfer. Rewriting NEWQP in SPCR2 causes execution to restart at the designated location. Reads of SPCR2 return the current value of the register, not of the buffer. Writing the same value into any control register except SPCR2 while the QSPI is enabled has no effect on QSPI operation.

Freescale Semiconductor, Inc.

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

9.3.1.2 Status Register

SPSR contains information concerning the current serial transmission. Only the QSPI can set the bits in this register. The CPU16 reads SPSR to obtain QSPI status information and writes SPSR to clear status flags.

9.3.2 QSPI RAM

The QSPI contains an 80-byte block of dual-ported static RAM that can be accessed by both the QSPI and the CPU16. The RAM is divided into three segments: receive data RAM, transmit data RAM, and command data RAM. Receive data is information received from a serial device external to the MCU. Transmit data is information stored for transmission to an external device. Command control data defines transfer parameters. Refer to **Figure 9-3**, which shows RAM organization.

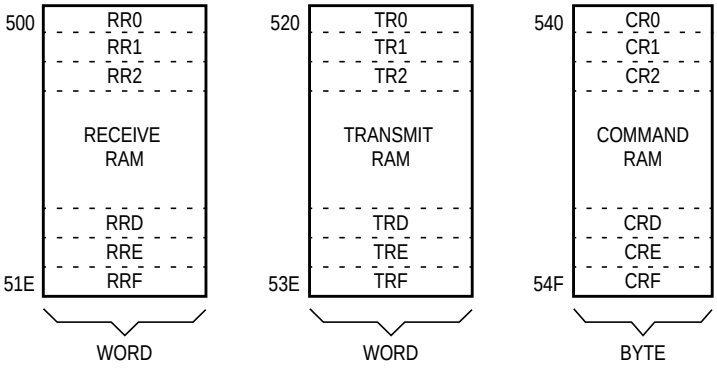


Figure 9-3 QSPI RAM

9.3.2.1 Receive RAM

Data received by the QSPI is stored in this segment to be read by the CPU16. Data stored in the receive RAM is right-justified. Unused bits in a receive queue entry are set to zero by the QSPI upon completion of the individual queue entry. The CPU16 can access the data using byte, word, or long-word transfers.

The CPTQP value in SPSR shows which queue entries have been executed. The CPU16 can use this information to determine which locations in receive RAM contain valid data before reading them.

9.3.2.2 Transmit RAM

Data that is to be transmitted by the QSPI is stored in this segment and must be written by the CPU16 in right-justified form. The QSPI cannot modify information in the transmit RAM. The QSPI copies the information to its data serializer for transmission. Information remains in the transmit RAM until overwritten.

Freescale Semiconductor, Inc.

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

9.3.2.3 Command RAM

Command RAM is used by the QSPI in master mode. The CPU16 writes one byte of control information to this segment for each QSPI command to be executed. The QSPI cannot modify information in command RAM.

Command RAM consists of 16 bytes. Each byte is divided into two fields. The peripheral chip-select field enables peripherals for transfer. The command control field provides transfer options.

A maximum of 16 commands can be in the queue. Queue execution by the QSPI proceeds from the address in NEWQP through the address in ENDQP (both of these fields are in SPCR2).

9.3.3 QSPI Pins

The QSPI uses seven pins. These pins can be configured for general-purpose I/O when not needed for QSPI application.

Table 9-2 shows QSPI input and output pins and their functions.

Table 9-2 QSPI Pins

Pin Names	Mnemonics	Mode	Function
Master In Slave Out	MISO	Master Slave	Serial data input to QSPI Serial data output from QSPI
Master Out Slave In	MOSI	Master Slave	Serial data output from QSPI Serial data input to QSPI
Serial Clock	SCK	Master Slave	Clock output from QSPI Clock input to QSPI
Peripheral Chip Selects	PCS[3:1]	Master	Select peripherals
Slave Select	PCS0/$\overline{SS}$	Master Master Slave	Selects peripherals Causes mode fault Initiates serial transfer

9.3.4 QSPI Operation

The QSPI uses a dedicated 80-byte block of static RAM accessible by both the QSPI and the CPU16 to perform queued operations. The RAM is divided into three segments. There are 16 command bytes, 16 transmit data words, and 16 receive data words. QSPI RAM is organized so that one byte of command data, one word of transmit data, and one word of receive data correspond to one queue entry, $0–$F.

The CPU16 initiates QSPI operation by setting up a queue of QSPI commands in command RAM, writing transmit data into transmit RAM, then enabling the QSPI. The QSPI executes the queued commands, sets a completion flag (SPIF), and then either interrupts the CPU16 or waits for intervention.

There are four queue pointers. The CPU16 can access three of them through fields in QSPI registers. The new queue pointer (NEWQP), contained in SPCR2, points to the first command in the queue. An internal queue pointer points to the command currently being executed. The completed queue pointer (CPTQP), contained in SPSR, points to the last command executed. The end queue pointer (ENDQP), contained in SPCR2, points to the final command in the queue.

Freescale Semiconductor, Inc.

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

The internal pointer is initialized to the same value as NEWQP. During normal operation, the command pointed to by the internal pointer is executed, the value in the internal pointer is copied into CPTQP, the internal pointer is incremented, and then the sequence repeats. Execution continues at the internal pointer address unless the NEWQP value is changed. After each command is executed, ENDQP and CPTQP are compared. When a match occurs, the SPIF flag is set and the QSPI stops and clears SPE, unless wrap-around mode is enabled.

At reset, NEWQP is initialized to $0. When the QSPI is enabled, execution begins at queue address $0 unless another value has been written into NEWQP. ENDQP is initialized to $0 at reset, but should be changed to show the last queue entry before the QSPI is enabled. NEWQP and ENDQP can be written at any time. When NEWQP changes, the internal pointer value also changes. However, if NEWQP is written while a transfer is in progress, the transfer is completed normally. Leaving NEWQP and ENDQP set to $0 transfers only the data in transmit RAM location $0.

9.3.5 QSPI Operating Modes

The QSPI operates in either master or slave mode. Master mode is used when the MCU initiates data transfers. Slave mode is used when an external device initiates transfers. Switching between these modes is controlled by MSTR in SPCR0. Before entering either mode, the appropriate QSM and QSPI registers must be initialized properly.

In master mode, the QSPI executes a queue of commands defined by control bits in each command RAM queue entry. Chip-select pins are activated, data is transmitted from the transmit RAM and received by the receive RAM.

In slave mode, operation proceeds in response to $\overline{SS}$ pin activation by an external SPI bus master. Operation is similar to master mode, but no peripheral chip selects are generated, and the number of bits transferred is controlled in a different manner. When the QSPI is selected, it automatically executes the next queue transfer to exchange data with the external device correctly.

Although the QSPI inherently supports multi-master operation, no special arbitration mechanism is provided. A mode fault flag (MODF) indicates a request for SPI master arbitration. System software must provide arbitration. Note that unlike previous SPI systems, MSTR is not cleared by a mode fault being set nor are the QSPI pin output drivers disabled. The QSPI and associated output drivers must be disabled by clearing SPE in SPCR1.

Figure 9-4 shows QSPI initialization. **Figures 9-5** through **9-9** show QSPI master and slave operation. The CPU16 must initialize the QSM global and pin registers and the QSPI control registers before enabling the QSPI for either mode of operation. The command queue must be written before the QSPI is enabled for master mode operation. Any data to be transmitted should be written into transmit RAM before the QSPI is enabled. During wrap-around operation, data for subsequent transmissions can be written at any time.

Freescale Semiconductor, Inc.

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

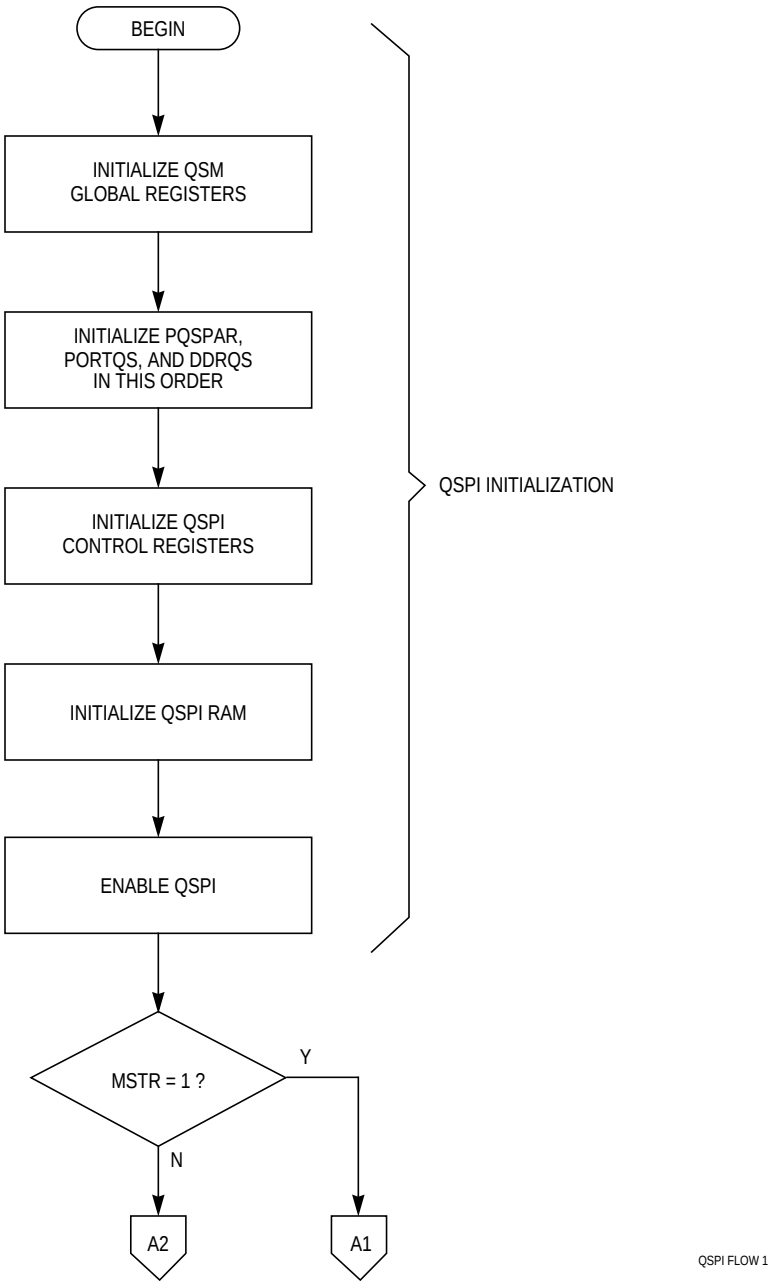


Figure 9-4 Flowchart of QSPI Initialization Operation

Freescale Semiconductor, Inc.

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

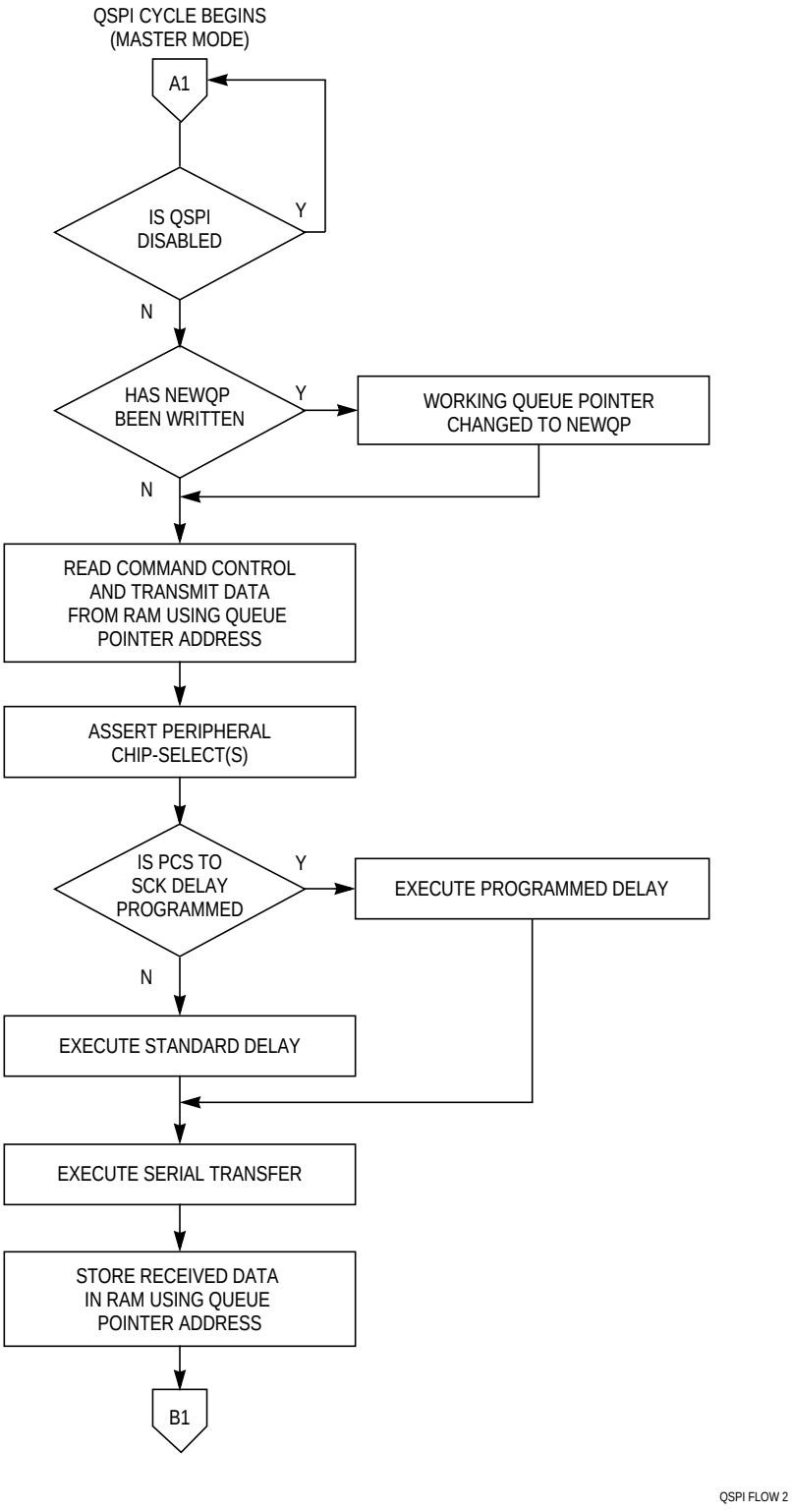


Figure 9-5 Flowchart of QSPI Master Operation (Part 1)

Freescale Semiconductor, Inc.

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

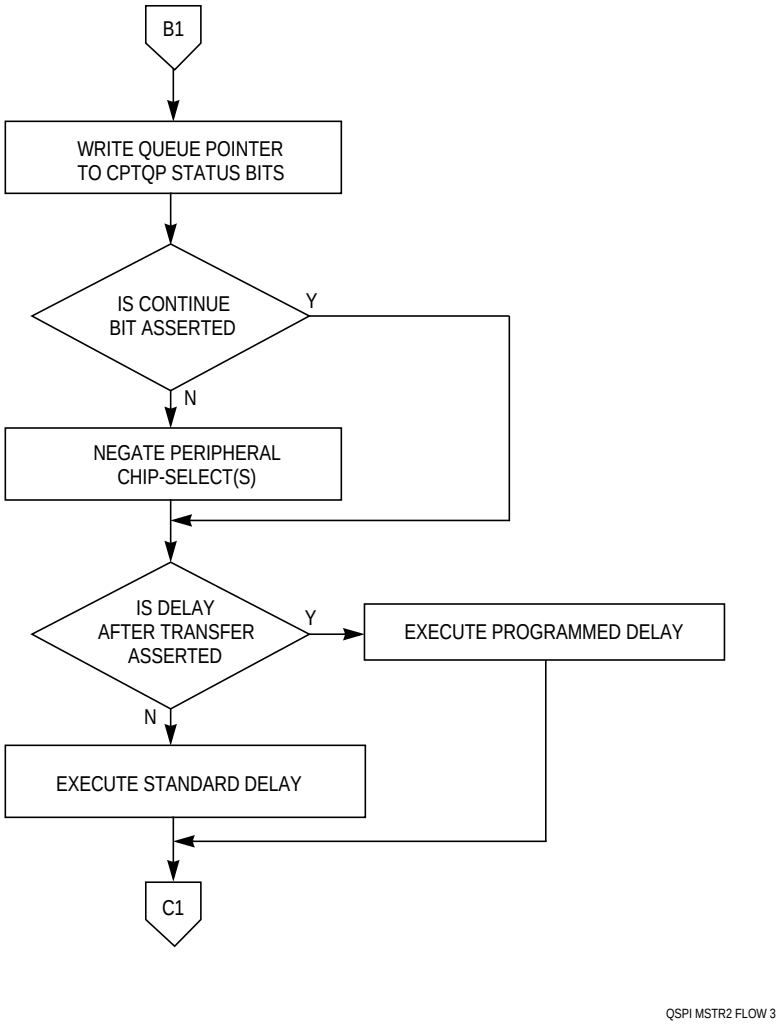


Figure 9-6 Flowchart of QSPI Master Operation (Part 2)

Freescale Semiconductor, Inc.

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

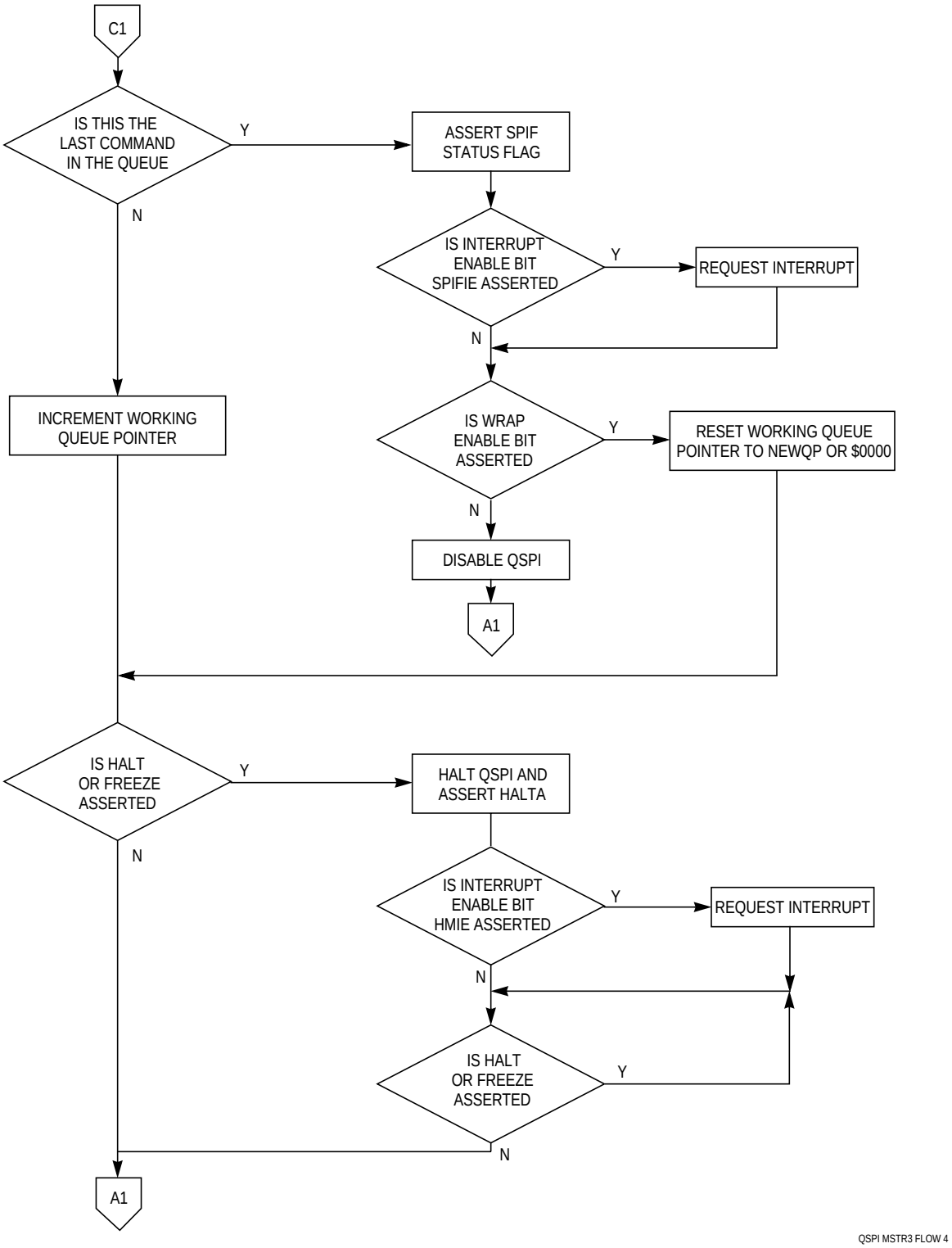


Figure 9-7 Flowchart of QSPI Master Operation (Part 3)

Freescale Semiconductor, Inc.

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

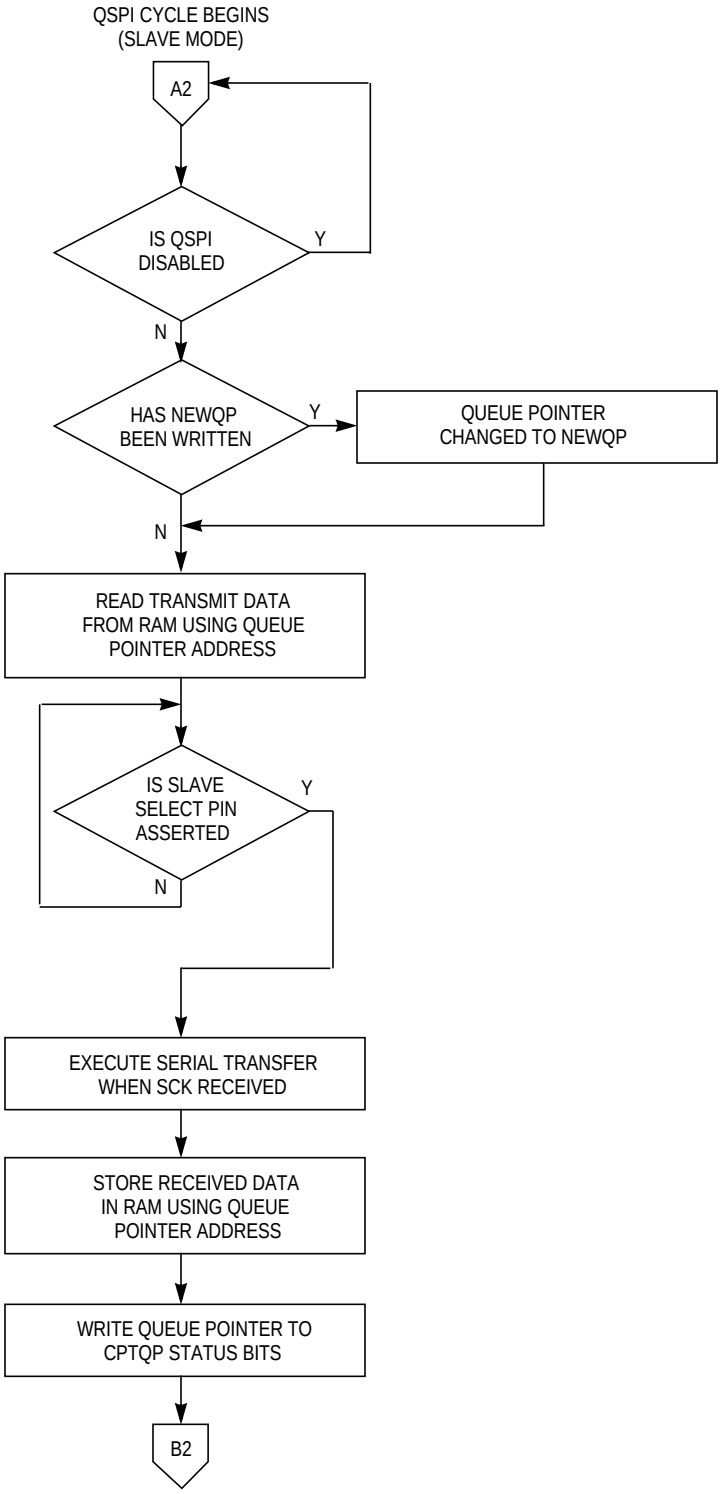


Figure 9-8 Flowchart of QSPI Slave Operation (Part 1)

Freescale Semiconductor, Inc.

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

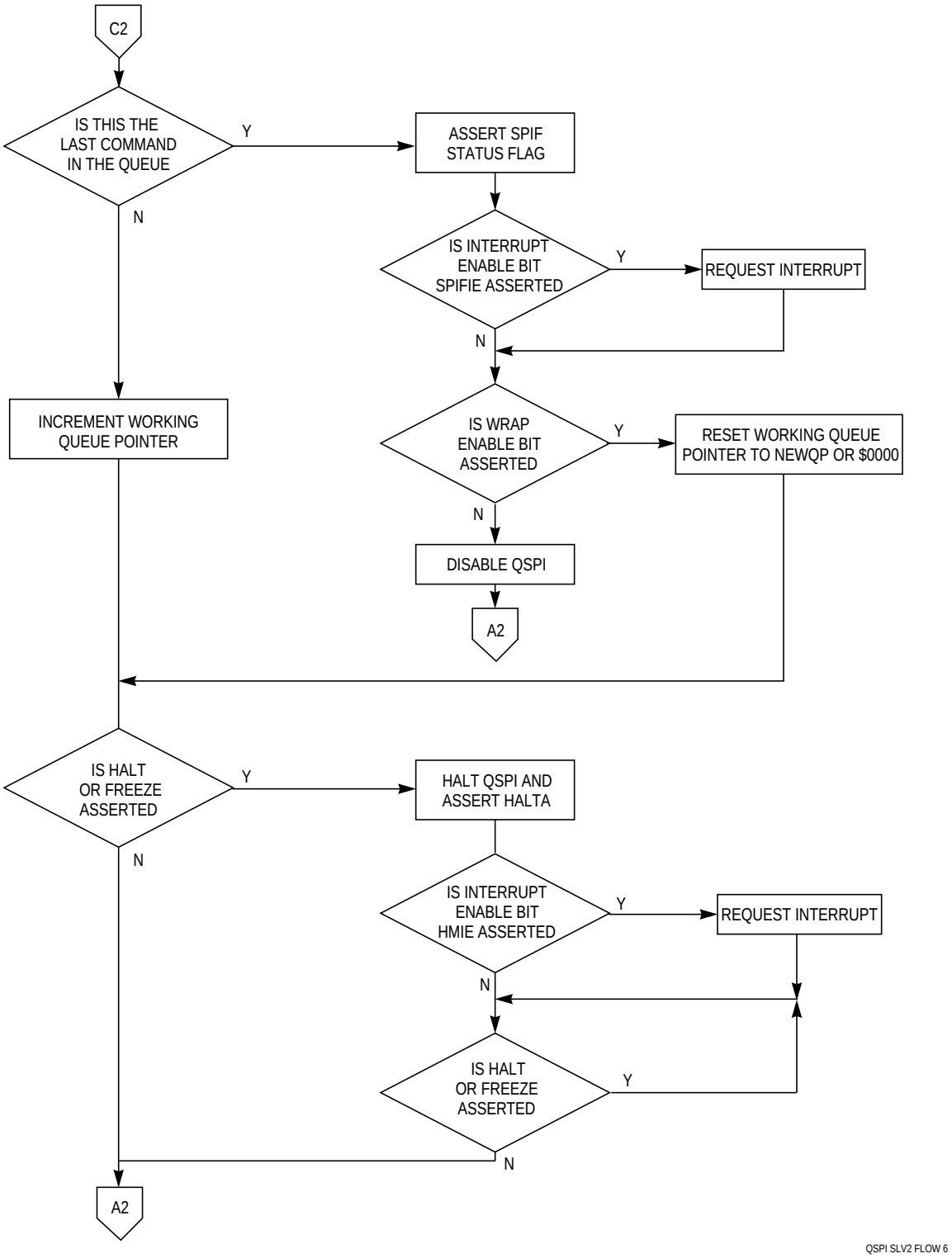


Figure 9-9 Flowchart of QSPI Slave Operation (Part 2)

Freescale Semiconductor, Inc.

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

Normally, the SPI bus performs synchronous bidirectional transfers. The serial clock on the SPI bus master supplies the clock signal SCK to time the transfer of data. Four possible combinations of clock phase and polarity can be specified by the CPHA and CPOL bits in SPCR0.

Data is transferred with the most significant bit first. The number of bits transferred per command defaults to eight, but can be set to any value from eight to sixteen bits inclusive by writing a value into the BITS[3:0] field in SPCR0 and setting BITSE in the command RAM.

Typically, SPI bus outputs are not open-drain unless multiple SPI masters are in the system. If needed, the WOMQ bit in SPCR0 can be set to provide wired-OR, open-drain outputs. An external pull-up resistor should be used on each output line. WOMQ affects all QSPI pins regardless of whether they are assigned to the QSPI or used as general-purpose I/O.

9.3.5.1 Master Mode

Setting the MSTR bit in SPCR0 selects master mode operation. In master mode, the QSPI can initiate serial transfers, but cannot respond to externally initiated transfers. When the slave select input of a device configured for master mode is asserted, a mode fault occurs.

Before QSPI operation begins, QSM register PQSPAR must be written to assign the necessary pins to the QSPI. The pins necessary for master mode operation are MISO, MOSI, SCK, and one or more of the chip-select pins. MISO is used for serial data input in master mode, and MOSI is used for serial data output. Either or both may be necessary, depending on the particular application. SCK is the serial clock output in master mode and must be assigned to the QSPI for proper operation.

The PORTQS data register must next be written with values that make the PQS2/SCK and PQS[6:3]/PCS[3:0] outputs inactive when the QSPI completes a series of transfers. Pins allocated to the QSPI by PQSPAR are controlled by PORTQS when the QSPI is inactive. PORTQS I/O pins driven to states opposite those of the inactive QSPI signals can generate glitches that momentarily enable or partially clock a slave device.

For example, if a slave device operates with an inactive SCK state of logic one (CPOL = 1) and uses active low peripheral chip-select PCS0, the PQS[3:2] bits in PORTQS must be set to %11. If PQS[3:2] = %00, falling edges will appear on PQS2/SCK and PQS3/PCS0 as the QSPI relinquishes control of these pins and PORTQS drives them to logic zero from the inactive SCK and PCS0 states of logic one.

Before master mode operation is initiated, QSM register DDRQS is written last to direct the data flow on the QSPI pins used. Configure the SCK, MOSI and appropriate chip-select pins PCS[3:0] as outputs. The MISO pin must be configured as an input.

After pins are assigned and configured, write appropriate data to the command queue. If data is to be transmitted, write the data to transmit RAM. Initialize the queue pointers as appropriate.

Freescale Semiconductor, Inc.

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

Data transfer is synchronized with the internally-generated serial clock SCK. Control bits, CPHA and CPOL, in SPCR0, control clock phase and polarity. Combinations of CPHA and CPOL determine upon which SCK edge to drive outgoing data from the MOSI pin and to latch incoming data from the MISO pin.

Baud rate is selected by writing a value from two to 255 into SPBR[7:0] in SPCR0. The QSPI uses a modulus counter to derive the SCK baud rate from the MCU system clock.

The following expressions apply to the SCK baud rate:

$$\text{SCK Baud Rate} = \frac{f_{sys}}{2 \times \text{SPBR[7:0]}}$$

or

$$\text{SPBR[7:0]} = \frac{f_{sys}}{2 \times \text{SCK Baud Rate Desired}}$$

Giving SPBR[7:0] a value of zero or one disables the baud rate generator and SCK assumes its inactive state.

The DSCK bit in each command RAM byte inserts either a standard (DSCK = 0) or user-specified (DSCK = 1) delay from chip-select assertion until the leading edge of the serial clock. The DSCKL field in SPCR1 determines the length of the user-defined delay before the assertion of SCK. The following expression determines the actual delay before SCK:

$$\text{PCS to SCK Delay} = \frac{\text{DSCKL[6:0]}}{f_{sys}}$$

where DSCKL[6:0] equals {1, 2, 3,..., 127}.

When DSCK equals zero, DSCKL[6:0] is not used. Instead, the PCS valid-to-SCK transition is one-half the SCK period.

There are two transfer length options. The user can choose a default value of eight bits, or a programmed value from eight to sixteen bits, inclusive. The programmed value must be written into BITS[3:0] in SPCR0. The BITSE bit in each command RAM byte determines whether the default value (BITSE = 0) or the BITS[3:0] value (BITSE = 1) is used. **Table 9-3** shows BITS[3:0] encoding.

Freescale Semiconductor, Inc.

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

Table 9-3 Bits Per Transfer

BITS[3:0]	Bits Per Transfer
0000	16
0001	Reserved
0010	Reserved
0011	Reserved
0100	Reserved
0101	Reserved
0110	Reserved
0111	Reserved
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Delay after transfer can be used to provide a peripheral deselect interval. A delay can also be inserted between consecutive transfers to allow serial A/D converters to complete conversion. Writing a value to DTL[7:0] in SPCR1 specifies a delay period. The DT bit in each command RAM byte determines whether the standard delay period (DT = 0) or the user-specified delay period (DT = 1) is used. The following expression is used to calculate the delay:

$$\text{Delay after Transfer} = \frac{32 \times \text{DTL}[7:0]}{f_{sys}} \text{ if DT = 1}$$

where DTL equals {1, 2, 3,..., 255}.

A zero value for DTL[7:0] causes a delay-after-transfer value of $8192/f_{sys}$.

$$\text{Standard Delay after Transfer} = \frac{17}{f_{sys}} \text{ if DT = 0}$$

Adequate delay between transfers must be specified for long data streams because the QSPI requires time to load a transmit RAM entry for transfer. Receiving devices need at least the standard delay between successive transfers. If the system clock is operating at a slower rate, the delay between transfers must be increased proportionately.

Freescale Semiconductor, Inc.

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

QSPI operation is initiated by setting the SPE bit in SPCR1. Shortly after SPE is set, the QSPI executes the command at the command RAM address pointed to by NEWQP. Data at the pointer address in transmit RAM is loaded into the data serializer and transmitted. Data that is simultaneously received is stored at the pointer address in receive RAM.

When the proper number of bits have been transferred, the QSPI stores the working queue pointer value in CPTQP, increments the working queue pointer, and loads the next data for transfer from transmit RAM. The command pointed to by the incremented working queue pointer is executed next, unless a new value has been written to NEWQP. If a new queue pointer value is written while a transfer is in progress, that transfer is completed normally.

When the CONT bit in a command RAM byte is set, PCS pins are continuously driven to specified states during and between transfers. If the chip-select pattern changes during or between transfers, the original pattern is driven until execution of the following transfer begins. When CONT is cleared, the data in register PORTQS is driven between transfers. The data in PORTQS must match the inactive states of SCK and any peripheral chip-selects used.

When the QSPI reaches the end of the queue, it sets the SPIF flag. SPIF is set during the final transfer before it is complete. If the SPIFIE bit in SPCR2 is set, an interrupt request is generated when SPIF is asserted. At this point, the QSPI clears SPE and stops unless wrap-around mode is enabled.

9.3.5.2 Master Wrap-Around Mode

Wrap-around mode is enabled by setting the WREN bit in SPCR2. The queue can wrap to pointer address $0 or to the address pointed to by NEWQP, depending on the state of the WRTO bit in SPCR2.

In wrap-around mode, the QSPI cycles through the queue continuously, even while the QSPI is requesting interrupt service. SPE is not cleared when the last command in the queue is executed. New receive data overwrites previously received data in receive RAM. Each time the end of the queue is reached, the SPIF flag is set. SPIF is not automatically reset. If interrupt-driven QSPI service is used, the service routine must clear the SPIF bit to end the current interrupt request. Additional interrupt requests during servicing can be prevented by clearing SPIFIE, but SPIFIE is buffered. Clearing it does not end the current request.

Wrap-around mode is exited by clearing the WREN bit or by setting the HALT bit in SPCR3. Exiting wrap-around mode by clearing SPE is not recommended, as clearing SPE may abort a serial transfer in progress. The QSPI sets SPIF, clears SPE, and stops the first time it reaches the end of the queue after WREN is cleared. After HALT is set, the QSPI finishes the current transfer, then stops executing commands. After the QSPI stops, SPE can be cleared.

Freescale Semiconductor, Inc.

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

9.3.5.3 Slave Mode

Clearing the MSTR bit in SPCR0 selects slave mode operation. In slave mode, the QSPI is unable to initiate serial transfers. Transfers are initiated by an external SPI bus master. Slave mode is typically used on a multi-master SPI bus. Only one device can be bus master (operate in master mode) at any given time.

Before QSPI operation is initiated, QSM register PQSPAR must be written to assign necessary pins to the QSPI. The pins necessary for slave mode operation are MISO, MOSI, SCK, and PCS0/$\overline{SS}$. MISO is used for serial data output in slave mode, and MOSI is used for serial data input. Either or both may be necessary, depending on the particular application. SCK is the serial clock input in slave mode and must be assigned to the QSPI for proper operation. Assertion of the active-low slave select signal ($\overline{SS}$) initiates slave mode operation.

Before slave mode operation is initiated, DDRQS must be written to direct data flow on the QSPI pins used. Configure the MOSI, SCK and PCS0/$\overline{SS}$ pins as inputs. The MISO pin must be configured as an output.

After pins are assigned and configured, write data to be transmitted into transmit RAM. Command RAM is not used in slave mode, and does not need to be initialized. Set the queue pointers, as appropriate.

When SPE is set and MSTR is clear, a low state on the slave select PCS0/$\overline{SS}$ pin begins slave mode operation at the address indicated by NEWQP. Data that is received is stored at the pointer address in receive RAM. Data is simultaneously loaded into the data serializer from the pointer address in transmit RAM and transmitted. Transfer is synchronized with the externally generated SCK. The CPHA and CPOL bits determine upon which SCK edge to latch incoming data from the MISO pin and to drive outgoing data from the MOSI pin.

Because the command RAM is not used in slave mode, the CONT, BITSE, DT, DSCK, and peripheral chip-select bits have no effect. The PCS0/$\overline{SS}$ pin is used only as an input.

The SPBR, DT and DSCKL fields in SPCR0 and SPCR1 bits are not used in slave mode. The QSPI drives neither the clock nor the chip-select pins and thus cannot control clock rate or transfer delay.

Because the BITSE option is not available in slave mode, the BITS field in SPCR0 specifies the number of bits to be transferred for all transfers in the queue. When the number of bits designated by BITS[3:0] has been transferred, the QSPI stores the working queue pointer value in CPTQP, increments the working queue pointer, and loads new transmit data from transmit RAM into the data serializer. The working queue pointer address is used the next time PCS0/$\overline{SS}$ is asserted, unless the CPU16 writes to NEWQP first.

The QSPI shifts one bit for each pulse of SCK until the slave select input goes high. If $\overline{SS}$ goes high before the number of bits specified by the BITS field is transferred, the QSPI resumes operation at the same pointer address the next time $\overline{SS}$ is asserted. The maximum value that the BITS field can have is 16. If more than 16 bits are transmitted before $\overline{SS}$ is negated, pointers are incremented and operation continues.

Freescale Semiconductor, Inc.

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

The QSPI transmits as many bits as it receives at each queue address, until the BITS[3:0] value is reached or $\overline{SS}$ is negated. $\overline{SS}$ does not need to go high between transfers as the QSPI transfers data until reaching the end of the queue, whether $\overline{SS}$ remains low or is toggled between transfers.

When the QSPI reaches the end of the queue, it sets the SPIF flag. If the SPIFIE bit in SPCR2 is set, an interrupt request is generated when SPIF is asserted. At this point, the QSPI clears SPE and stops unless wrap-around mode is enabled.

9.3.5.4 Slave Wrap-Around Mode

Slave wrap-around mode is enabled by setting the WREN bit in SPCR2. The queue can wrap to pointer address $0 or to the address pointed to by NEWQP, depending on the state of the WRTO bit in SPCR2. Slave wrap-around operation is identical to master wrap-around operation.

9.3.6 Peripheral Chip Selects

Peripheral chip-select signals are used to select an external device for serial data transfer. Chip-select signals are asserted when a command in the queue is executed. Signals are asserted at a logic level corresponding to the value of the PCS[3:0] bits in each command byte. More than one chip-select signal can be asserted at a time, and more than one external device can be connected to each PCS pin, provided proper fanout is observed. PCS0 shares a pin with the slave select ($\overline{SS}$) signal, which initiates slave mode serial transfer. If $\overline{SS}$ is taken low when the QSPI is in master mode, a mode fault occurs.

To configure a peripheral chip-select, set the appropriate bit in PQSPAR, then configure the chip-select pin as an output by setting the appropriate bit in DDRQS. The value of the bit in PORTQS that corresponds to the chip-select pin determines the base state of the chip-select signal. If base state is zero, chip-select assertion must be active high (PCS bit in command RAM must be set); if base state is one, assertion must be active low (PCS bit in command RAM must be cleared). PORTQS bits are cleared during reset. If no new data is written to PORTQS before pin assignment and configuration as an output, the base state of chip-select signals is zero and chip-select pins should thus be driven active-high.

9.4 Serial Communication Interface

The serial communication interface (SCI) communicates with external devices through an asynchronous serial bus. The SCI uses a standard non-return to zero (NRZ) transmission format. The SCI is fully compatible with other Freescale SCI systems, such as those on M68HC11 and M68HC05 devices. **Figure 9-10** is a block diagram of the SCI transmitter. **Figure 9-11** is a block diagram of the SCI receiver.

Freescale Semiconductor, Inc.

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

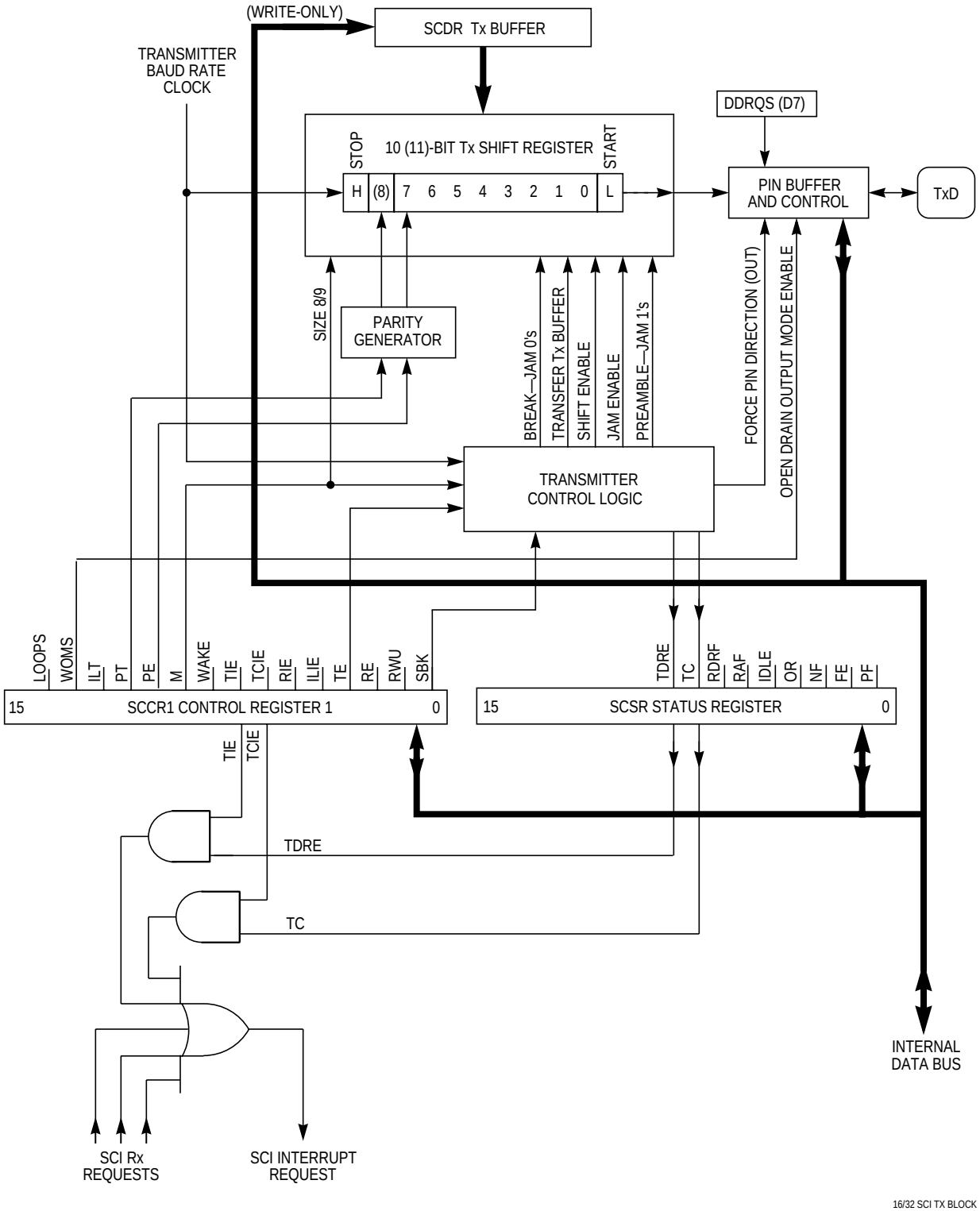


Figure 9-10 SCI Transmitter Block Diagram

Freescale Semiconductor, Inc.

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

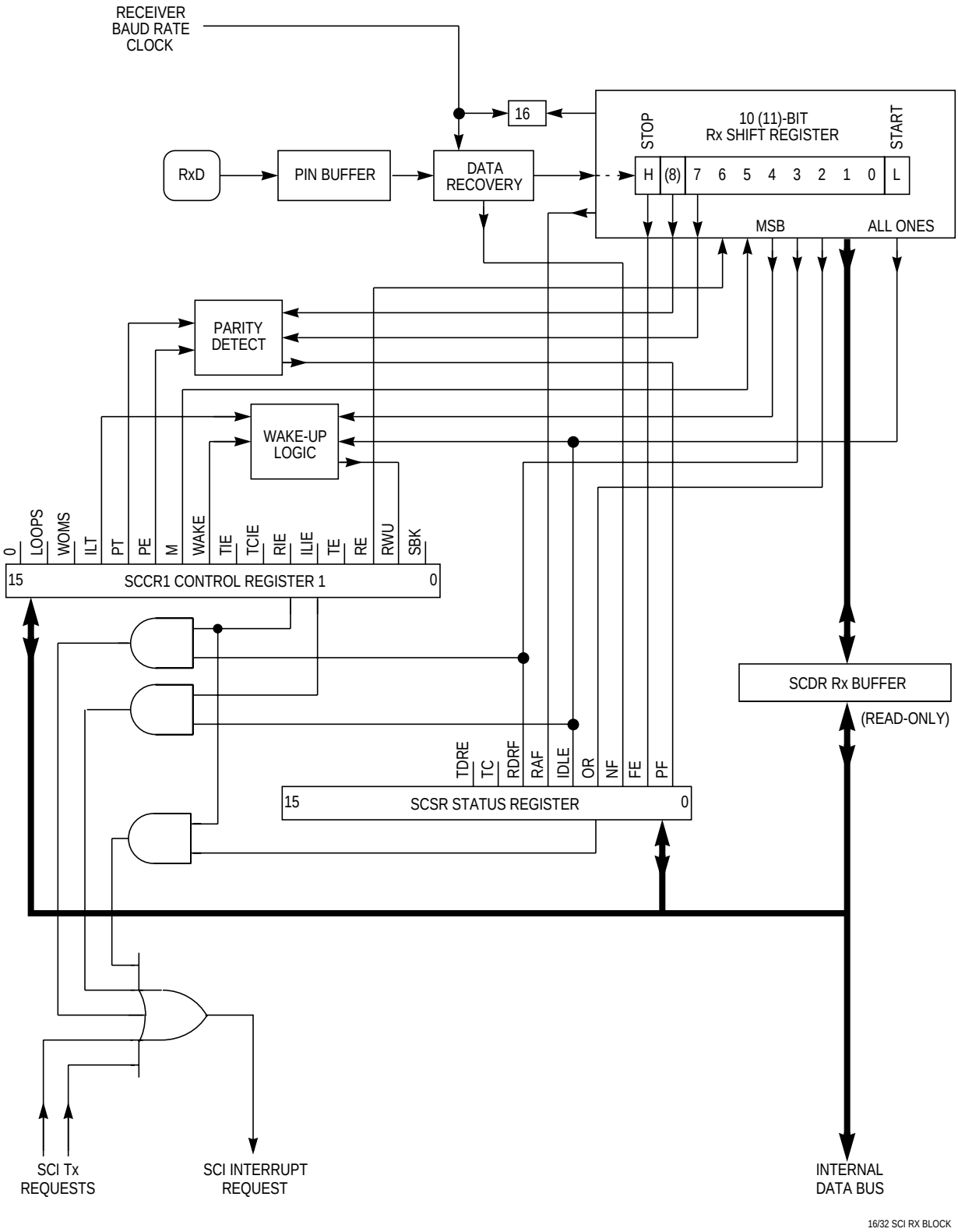


Figure 9-11 SCI Receiver Block Diagram

Freescale Semiconductor, Inc.

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

9.4.1 SCI Registers

The SCI programming model includes the QSM global and pin control registers, and four SCI registers. There are two SCI control registers (SCCR0 and SCCR1), one status register (SCSR), and one data register (SCDR). Refer to **D.6 Queued Serial Module** for register bit and field definitions.

9.4.1.1 Control Registers

SCCR0 contains the baud rate selection field. Baud rate must be set before the SCI is enabled. This register can be read or written.

SCCR1 contains a number of SCI configuration parameters, including transmitter and receiver enable bits, interrupt enable bits, and operating mode enable bits. This register can be read or written at any time. The SCI can modify the RWU bit under certain circumstances.

Changing the value of SCI control bits during a transfer may disrupt operation. Before changing register values, allow the SCI to complete the current transfer, then disable the receiver and transmitter.

9.4.1.2 Status Register

SCSR contains flags that show SCI operating conditions. These flags are cleared either by SCI hardware or by reading SCSR, then reading or writing SCDR. A long-word read can consecutively access both SCSR and SCDR. This action clears receiver status flag bits that were set at the time of the read, but does not clear TDRE or TC flags.

If an internal SCI signal for setting a status bit comes after reading the asserted status bits, but before reading or writing SCDR, the newly set status bit is not cleared. SCSR must be read again with the bit set, and SCDR must be read or written before the status bit is cleared.

Reading either byte of SCSR causes all 16 bits to be accessed, and any status bit already set in either byte is cleared on a subsequent read or write of SCDR.

9.4.1.3 Data Register

SCDR contains two data registers at the same address. The receive data register (RDR) is a read-only register that contains data received by the SCI. Data enters the receive serial shifter and is transferred to RDR. The transmit data register (TDR) is a write-only register that contains data to be transmitted. Data is first written to TDR, then transferred to the transmit serial shifter, where additional format bits are added before transmission. R[7:0]/T[7:0] contain either the first eight data bits received when SCDR is read, or the first eight data bits to be transmitted when SCDR is written. R8/T8 are used when the SCI is configured for 9-bit operation. When the SCI is configured for 8-bit operation, they have no meaning or effect.

Freescale Semiconductor, Inc.

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

9.4.2 SCI Pins

Two unidirectional pins, TXD (transmit data) and RXD (receive data), are associated with the SCI. TXD can be used by the SCI or for general-purpose I/O. TXD function is controlled by PQSPA7 in the port QS pin assignment register (PQSPAR) and TE in SCI control register 1 (SCCR1). The receive data (RXD) pin is dedicated to the SCI.

9.4.3 SCI Operation

The SCI can operate in polled or interrupt-driven mode. Status flags in SCSR reflect SCI conditions regardless of the operating mode chosen. The TIE, TCIE, RIE, and ILIE bits in SCCR1 enable interrupts for the conditions indicated by the TDRE, TC, RDRF, and IDLE bits in SCSR, respectively.

9.4.3.1 Definition of Terms

- Bit-Time — The time required to transmit or receive one bit of data, which is equal to one cycle of the baud frequency.
- 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 and be preceded by at least three receive time samples of logic one.
- Stop Bit — One bit-time of logic one that indicates the end of a data frame.
- Frame — A complete unit of serial information. The SCI can use 10-bit or 11-bit frames.
- Data Frame — A start bit, a specified number of data or information bits, and at least one stop bit.
- Idle Frame — A frame that consists of consecutive ones. An idle frame has no start bit.
- Break Frame — A frame that consists of consecutive zeros. A break frame has no stop bits.

9.4.3.2 Serial Formats

All data frames must have a start bit and at least one stop bit. Receiving and transmitting devices must use the same data frame format. The SCI provides hardware support for both 10-bit and 11-bit frames. The M bit in SCCR1 specifies the number of bits per frame.

The most common data frame format for NRZ serial interfaces is one start bit, eight data bits (LSB first), and one stop bit; a total of ten bits. The most common 11-bit data frame contains one start bit, eight data bits, a parity or control bit, and one stop bit. Ten-bit and 11-bit frames are shown in **Table 9-4**.

Freescale Semiconductor, Inc.

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

Table 9-4 Serial Frame Formats

10-Bit Frames			
Start	**Data**	**Parity/Control**	**Stop**
1	7	—	2
1	7	1	1
1	8	—	1
11-Bit Frames			
Start	**Data**	**Parity/Control**	**Stop**
1	7	1	2
1	8	1	1

9.4.3.3 Baud Clock

The SCI baud rate is programmed by writing a 13-bit value to the SCBR field in SCI control register 0 (SCCR0). The baud rate is derived from the MCU system clock by a modulus counter. Writing a value of zero to SCBR[12:0] disables the baud rate generator. Baud rate is calculated as follows:

$$\text{SCI Baud Rate} = \frac{f_{sys}}{32 \times \text{SCBR[12:0]}}$$

or

$$\text{SCBR[12:0]} = \frac{f_{sys}}{32 \times \text{SCI Baud Rate Desired}}$$

where SCBR[12:0] is in the range {1, 2, 3, ..., 8191}.

The SCI receiver operates asynchronously. An internal clock is necessary to synchronize with an incoming data stream. The SCI baud rate generator produces a receive time sampling clock with a frequency 16 times that of the SCI baud rate. The SCI determines the position of bit boundaries from transitions within the received waveform, and adjusts sampling points to the proper positions within the bit period.

9.4.3.4 Parity Checking

The PT bit in SCCR1 selects either even (PT = 0) or odd (PT = 1) parity. PT affects received and transmitted data. The PE bit in SCCR1 determines whether parity checking is enabled (PE = 1) or disabled (PE = 0). When PE is set, the MSB of data in a frame is used for the parity function. For transmitted data, a parity bit is generated; for received data, the parity bit is checked. When parity checking is enabled, the PF bit in the SCI status register (SCSR) is set if a parity error is detected.

Enabling parity affects the number of data bits in a frame, which can in turn affect frame size. **Table 9-5** shows possible data and parity formats.

Freescale Semiconductor, Inc.

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

Table 9-5 Effect of Parity Checking on Data Size

M	PE	Result
0	0	8 data bits
0	1	7 data bits, 1 parity bit
1	0	9 data bits
1	1	8 data bits, 1 parity bit

9.4.3.5 Transmitter Operation

The transmitter consists of a serial shifter and a parallel data register (TDR) located in the SCI data register (SCDR). The serial shifter cannot be directly accessed by the CPU16. The transmitter is double-buffered, which means that data can be loaded into TDR while other data is shifted out. The TE bit in SCCR1 enables (TE = 1) and disables (TE = 0) the transmitter.

The shifter output is connected to the TXD pin while the transmitter is operating (TE = 1, or TE = 0 and transmission in progress). Wired-OR operation should be specified when more than one transmitter is used on the same SCI bus. The WOMS bit in SCCR1 determines whether TXD is an open-drain (wired-OR) output or a normal CMOS output. An external pull-up resistor on TXD is necessary for wired-OR operation. WOMS controls TXD function whether the pin is used by the SCI or as a general-purpose I/O pin.

Data to be transmitted is written to SCDR, then transferred to the serial shifter. The transmit data register empty (TDRE) flag in SCSR shows the status of TDR. When TDRE = 0, TDR contains data that has not been transferred to the shifter. Writing to SCDR again overwrites the data. TDRE is set when the data in TDR is transferred to the shifter. Before new data can be written to SCDR, however, the processor must clear TDRE by writing to SCSR. If new data is written to SCDR without first clearing TDRE, the data will not be transmitted.

The transmission complete (TC) flag in SCSR shows transmitter shifter state. When TC = 0, the shifter is busy. TC is set when all shifting operations are completed. TC is not automatically cleared. The processor must clear it by first reading SCSR while TC is set, then writing new data to SCDR.

The state of the serial shifter is checked when the TE bit is set. If TC = 1, an idle frame is transmitted as a preamble to the following data frame. If TC = 0, the current operation continues until the final bit in the frame is sent, then the preamble is transmitted. The TC bit is set at the end of preamble transmission.

The SBK bit in SCCR1 is used to insert break frames in a transmission. A non-zero integer number of break frames is transmitted while SBK is set. Break transmission begins when SBK is set, and ends with the transmission in progress at the time either SBK or TE is cleared. If SBK is set while a transmission is in progress, that transmission finishes normally before the break begins. To assure the minimum break time, toggle SBK quickly to one and back to zero. The TC bit is set at the end of break transmission. After break transmission, at least one bit-time of logic level one (mark idle) is transmitted to ensure that a subsequent start bit can be detected.

Freescale Semiconductor, Inc.

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

If TE remains set, after all pending idle, data and break frames are shifted out, TDRE and TC are set and TXD is held at logic level one (mark).

When TE is cleared, the transmitter is disabled after all pending idle, data, and break frames are transmitted. The TC flag is set, and control of the TXD pin reverts to PQSPAR and DDRQS. Buffered data is not transmitted after TE is cleared. To avoid losing data in the buffer, do not clear TE until TDRE is set.

Some serial communication systems require a mark on the TXD pin even when the transmitter is disabled. Configure the TXD pin as an output, then write a one to PQS7. When the transmitter releases control of the TXD pin, it will revert to driving a logic one output.

To insert a delimiter between two messages, to place non-listening receivers in wake-up mode between transmissions, or to signal a retransmission by forcing an idle line, clear and then set TE before data in the serial shifter has shifted out. The transmitter finishes the transmission, then sends a preamble. After the preamble is transmitted, if TDRE is set, the transmitter will mark idle. Otherwise, normal transmission of the next sequence will begin.

Both TDRE and TC have associated interrupts. The interrupts are enabled by the transmit interrupt enable (TIE) and transmission complete interrupt enable (TCIE) bits in SCCR1. Service routines can load the last byte of data in a sequence into SCDR, then terminate the transmission when a TDRE interrupt occurs.

9.4.3.6 Receiver Operation

The RE bit in SCCR1 enables (RE = 1) and disables (RE = 0) the receiver. The receiver contains a receive serial shifter and a parallel receive data register (RDR) located in the SCI data register (SCDR). The serial shifter cannot be directly accessed by the CPU16. The receiver is double-buffered, allowing data to be held in RDR while other data is shifted in.

Receiver bit processor logic drives a state machine that determines the logic level for each bit-time. This state machine controls when the bit processor logic is to sample the RXD pin and also controls when data is to be passed to the receive serial shifter. A receive time clock is used to control sampling and synchronization. Data is shifted into the receive serial shifter according to the most recent synchronization of the receive time clock with the incoming data stream. From this point on, data movement is synchronized with the MCU system clock. Operation of the receiver state machine is detailed in the *QSM Reference Manual* (QSMRM/AD).

The number of bits shifted in by the receiver depends on the serial format. However, all frames must end with at least one stop bit. When the stop bit is received, the frame is considered to be complete, and the received data in the serial shifter is transferred to RDR. The receiver data register flag (RDRF) is set when the data is transferred.

Noise errors, parity errors, and framing errors can be detected while a data stream is being received. Although error conditions are detected as bits are received, the noise flag (NF), the parity flag (PF), and the framing error (FE) flag in SCSR are not set until data is transferred from the serial shifter to RDR.

Freescale Semiconductor, Inc.

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

RDRF must be cleared before the next transfer from the shifter can take place. If RDRF is set when the shifter is full, transfers are inhibited and the overrun error (OR) flag in SCSR is set. OR indicates that RDR needs to be serviced faster. When OR is set, the data in RDR is preserved, but the data in the serial shifter is lost. Because framing, noise, and parity errors are detected while data is in the serial shifter, FE, NF, and PF cannot occur at the same time as OR.

When the CPU16 reads SCSR and SCDR in sequence, it acquires status and data, and also clears the status flags. Reading SCSR acquires status and arms the clearing mechanism. Reading SCDR acquires data and clears SCSR.

When RIE in SCCR1 is set, an interrupt request is generated whenever RDRF is set. Because receiver status flags are set at the same time as RDRF, they do not have separate interrupt enables.

9.4.3.7 Idle-Line Detection

During a typical serial transmission, frames are transmitted isochronally and no idle time occurs between frames. Even when all the data bits in a frame are logic ones, the start bit provides one logic zero bit-time during the frame. An idle line is a sequence of contiguous ones equal to the current frame size. Frame size is determined by the state of the M bit in SCCR1.

The SCI receiver has both short and long idle-line detection capability. Idle-line detection is always enabled. The idle line type (ILT) bit in SCCR1 determines which type of detection is used. When an idle line condition is detected, the IDLE flag in SCSR is set.

For short idle-line detection, the receiver bit processor counts contiguous logic one bit-times whenever they occur. Short detection provides the earliest possible recognition of an idle line condition, because the stop bit and contiguous logic ones before and after it are counted. For long idle-line detection, the receiver counts logic ones after the stop bit is received. Only a complete idle frame causes the IDLE flag to be set.

In some applications, software overhead can cause a bit-time of logic level one to occur between frames. This bit-time does not affect content, but if it occurs after a frame of ones when short detection is enabled, the receiver flags an idle line.

When the ILIE bit in SCCR1 is set, an interrupt request is generated when the IDLE flag is set. The flag is cleared by reading SCSR and SCDR in sequence. IDLE is not set again until after at least one frame has been received (RDRF = 1). This prevents an extended idle interval from causing more than one interrupt.

9.4.3.8 Receiver Wake-Up

The receiver wake-up function allows a transmitting device to direct a transmission to a single receiver or to a group of receivers by sending an address frame at the start of a message. Hardware activates each receiver in a system under certain conditions. Resident software must process address information and enable or disable receiver operation.

Freescale Semiconductor, Inc.

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

A receiver is placed in wake-up mode by setting the RWU bit in SCCR1. While RWU is set, receiver status flags and interrupts are disabled. Although the CPU16 can clear RWU, it is normally cleared by hardware during wake-up.

The WAKE bit in SCCR1 determines which type of wake-up is used. When WAKE = 0, idle-line wake-up is selected. When WAKE = 1, address-mark wake-up is selected. Both types require a software-based device addressing and recognition scheme.

Idle-line wake-up allows a receiver to sleep until an idle line is detected. When an idle-line is detected, the receiver clears RWU and wakes up. The receiver waits for the first frame of the next transmission. The byte is received normally, transferred to RDR, and the RDRF flag is set. If software does not recognize the address, it can set RWU and put the receiver back to sleep. For idle-line wake-up to work, there must be a minimum of one frame of idle line between transmissions. There must be no idle time between frames within a transmission.

Address-mark wake-up uses a special frame format to wake up the receiver. When the MSB of an address-mark frame is set, that frame contains address information. The first frame of each transmission must be an address frame. When the MSB of a frame is set, the receiver clears RWU and wakes up. The byte is received normally, transferred to the RDR, and the RDRF flag is set. If software does not recognize the address, it can set RWU and put the receiver back to sleep. Address-mark wake-up allows idle time between frames and eliminates idle time between transmissions. However, there is a loss of efficiency because of an additional bit-time per frame.

9.4.3.9 Internal Loop Mode

The LOOPS bit in SCCR1 controls a feedback path in the data serial shifter. When LOOPS is set, the SCI transmitter output is fed back into the receive serial shifter. TXD is asserted (idle line). Both transmitter and receiver must be enabled before entering loop mode.

Freescale Semiconductor, Inc.

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

SECTION 10
MULTICHANNEL COMMUNICATION INTERFACE

This section is an overview of the multichannel communication interface (MCCI) module. Refer to the *MCCI Reference Manual* (MCCIRM/AD) for more information on MCCI capabilities. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for MCCI timing and electrical specifications. Refer to **D.7 Multichannel Communication Interface Module** for register address mapping and bit/field definitions.

10.1 General

The MCCI contains three serial interfaces: a serial peripheral interface (SPI) and two serial communication interfaces (SCI). **Figure 10-1** is a block diagram of the MCCI. The MCCI is present only on MC68HC16Z4 and MC68CK16Z4 microcontrollers.

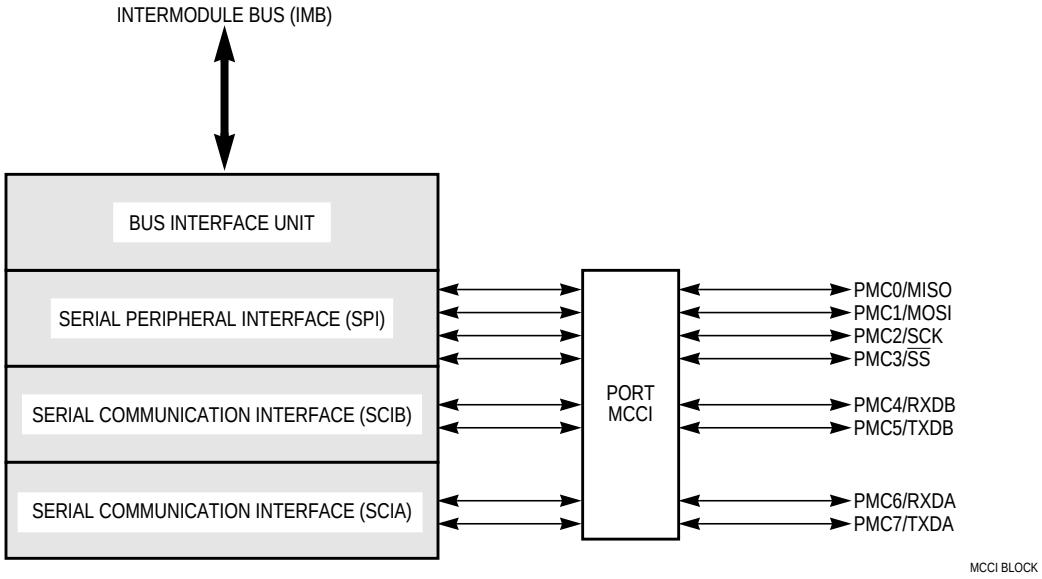


Figure 10-1 MCCI Block Diagram

The SPI provides easy peripheral expansion or interprocessor communication via a full-duplex, synchronous, three-line bus: data in, data out, and a serial clock. Serial transfer of eight or sixteen bits can begin with the most significant bit (MSB) or least significant bit (LSB). The MCCI module can be configured as a master or slave device. Clock control logic allows a selection of clock polarity and a choice of two clocking protocols to accommodate most available synchronous serial peripheral devices. When the SPI is configured as a master, software selects one of 254 different bit rates for the serial clock.

Freescale Semiconductor, Inc.

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

The SCI is a universal asynchronous receiver transmitter (UART) serial interface with a standard non-return to zero (NRZ) mark/space format. It operates in either full- or half-duplex mode. It also contains separate transmit and receive enable bits and a double-transmit buffer. A modulus-type baud rate generator provides rates from 64 baud to 524 kbaud with a 16.78-MHz system clock. Word length of either eight or nine bits is software selectable. Optional parity generation and detection provide either even or odd parity check capability. Advanced error detection circuitry catches glitches of up to 1/16 of a bit time in duration. Wakeup functions allow the CPU to run uninterrupted until meaningful data is received.

10.2 MCCI Registers and Address Map

The MCCI address map occupies 64 bytes from address $YFFC00 to $YFFC3F. It consists of MCCI global registers and SPI and SCI control, status, and data registers. Writes to unimplemented register bits have no effect, and reads of unimplemented bits always return zero.

The MM bit in the system integration module configuration register (SIMCR) defines the most significant bit (ADDR23) of the IMB address for each module. Because ADDR[23:20] are driven to the same bit as ADDR19, MM must be set to one. If MM is cleared, IMB modules are inaccessible. Refer to **5.2.1 Module Mapping** for more information about how the state of MM affects the system.

10.2.1 MCCI Global Registers

The MCCI module configuration register (MMCR) contains bits and fields to place the MCCI in low-power operation, establish the privilege level required to access MCCI registers, and establish the priority of the MCCI during interrupt arbitration. The MCCI test register (MTEST) is used only during factory test of the MCCI. The SCI interrupt level register (ILSCI) determines the level of interrupts requested by each SCI. Separate fields hold the interrupt-request levels for SCIA and SCIB. The MCCI interrupt vector register (MIVR) determines which three vectors in the exception vector table are to be used for MCCI interrupts. The SPI and both SCI interfaces have separate interrupt vectors adjacent to one another. The SPI interrupt level register (ILSPI) determines the priority level of interrupts requested by the SPI. The MCCI port data registers (PORTMC and PORTMCP) are used to configure port MCCI for general-purpose I/O. The MCCI pin assignment register (MPAR) determines which of the SPI pins (with the exception of SCK) are used by the SPI, and which pins are available for general-purpose I/O. The MCCI data direction register (MDDR) configures each pin as an input or output.

10.2.1.1 Low-Power Stop Mode

When the STOP bit in the MMCR is set, the IMB clock signal to most of the MCCI module is disabled. This places the module in an idle state and minimizes power consumption.

Freescale Semiconductor, Inc.

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

To ensure that the MCCI stops in a known state, assert the STOP bit before executing the CPU LPSTOP instruction. Before asserting the STOP bit, disable the SPI (clear the SPE bit) and disable the SCI receivers and transmitters (clear the RE and TE bits). Complete transfers in progress before disabling the SPI and SCI interfaces.

Once the STOP bit is asserted, it can be cleared by system software or by reset.

10.2.1.2 Privilege Levels

The supervisor bit (SUPV) in the MMCR has no effect since the CPU16 operates only in the supervisor mode.

10.2.1.3 MCCI Interrupts

The interrupt request level of each of the three MCCI interfaces can be programmed to a value of zero (interrupts disabled) through seven (highest priority). These levels are selected by the ILSCIA and ILSCIB fields in the SCI interrupt level register (ILSCI) and the ILSPI field in the SPI interrupt level register (ILSPI). In case two or more MCCI submodules request an interrupt simultaneously and are assigned the same interrupt request level, the SPI submodule is given the highest priority and SCIB is given the lowest.

When an interrupt is requested which is at a higher level than the interrupt mask in the CPU status register, the CPU initiates an interrupt acknowledge cycle. During this cycle, the MCCI compares its interrupt request level to the level recognized by the CPU. If a match occurs, arbitration with other modules begins.

Interrupting modules present their arbitration number on the IMB, and the module with the highest number wins. The arbitration number for the MCCI is programmed into the interrupt arbitration (IARB) field of the MMCR. Each module should be assigned a unique arbitration number. The reset value of the IARB field is $0, which prevents the MCCI from arbitrating during an interrupt acknowledge cycle. The IARB field should be initialized by system software to a value from $F (highest priority) through $1 (lowest priority). Otherwise, the CPU identifies any interrupts generated as spurious and takes a spurious-interrupt exception.

If the MCCI wins the arbitration, it generates an interrupt vector that uniquely identifies the interrupting serial interface. The six MSBs are read from the interrupt vector (INTV) field in the MCCI interrupt vector register (MIVR). The two LSBs are assigned by the MCCI according to the interrupting serial interface, as indicated in **Table 10-1**.

Table 10-1 MCCI Interrupt Vectors

Interface	INTV[1:0]
SCIA	00
SCIB	01
SPI	10

Freescale Semiconductor, Inc.

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

Select a value for INTV so that each MCCI interrupt vector corresponds to one of the user-defined vectors ($40–$FF). Refer to the *CPU16 Reference Manual* (CPU16RM/AD) for additional information on interrupt vectors.

10.2.2 Pin Control and General-Purpose I/O

The eight pins used by the SPI and SCI subsystems have alternate functions as general-purpose I/O pins. Configuring the MCCI submodule includes programming each pin for either general-purpose I/O or its serial interface function. In either function, each pin must also be programmed as input or output.

The MCCI data direction register (MDDR) assigns each MCCI pin as either input or output. The MCCI pin assignment register (MPAR) assigns the MOSI, MISO, and $\overline{SS}$ pins as either SPI pins or general-purpose I/O. (The fourth pin, SCK, is automatically assigned to the SPI whenever the SPI is enabled, for example, when the SPE bit in the SPI control register is set.) The receiver enable (RE) and transmitter enable (TE) bits in the SCI control registers (SCCR0A, SCCR0B) automatically assign the associated pin as an SCI pin when set or general-purpose I/O when cleared. **Table 10-2** summarizes how pin function and direction are assigned.

Table 10-2 Pin Assignments

Pin	Function Assigned By	Direction Assigned By
TXDA/PMC7	TE bit in SCCR0A	MMDR7
RXDA/PMC6	RE bit in SCCR0A	MMDR6
TXDB/PMC5	TE bit in SCCR0B	MMDR5
RXDB/PMC4	RE bit in SCCR0B	MMDR4
$\overline{SS}$/PMC3	$\overline{SS}$ bit in MPAR	MMDR3
SCK/PMC2	SPE bit in SPCR	MMDR2
MOSI/PMC1	MOSI bit in MPAR	MMDR1
MISO/PMC0	MISO bit in MPAR	MMDR0

10.3 Serial Peripheral Interface (SPI)

The SPI submodule communicates with external peripherals and other MCUs via a synchronous serial bus. The SPI is fully compatible with the serial peripheral interface systems found on othe Freescale devices, such as the M68HC11 and M68HC05 families. The SPI can perform full duplex three-wire or half duplex two-wire transfers. Serial transfer of eight or sixteen bits can begin with the MSB or LSB. The system can be configured as a master or slave device.

Figure 10-2 shows a block diagram of the SPI.

Freescale Semiconductor, Inc.

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

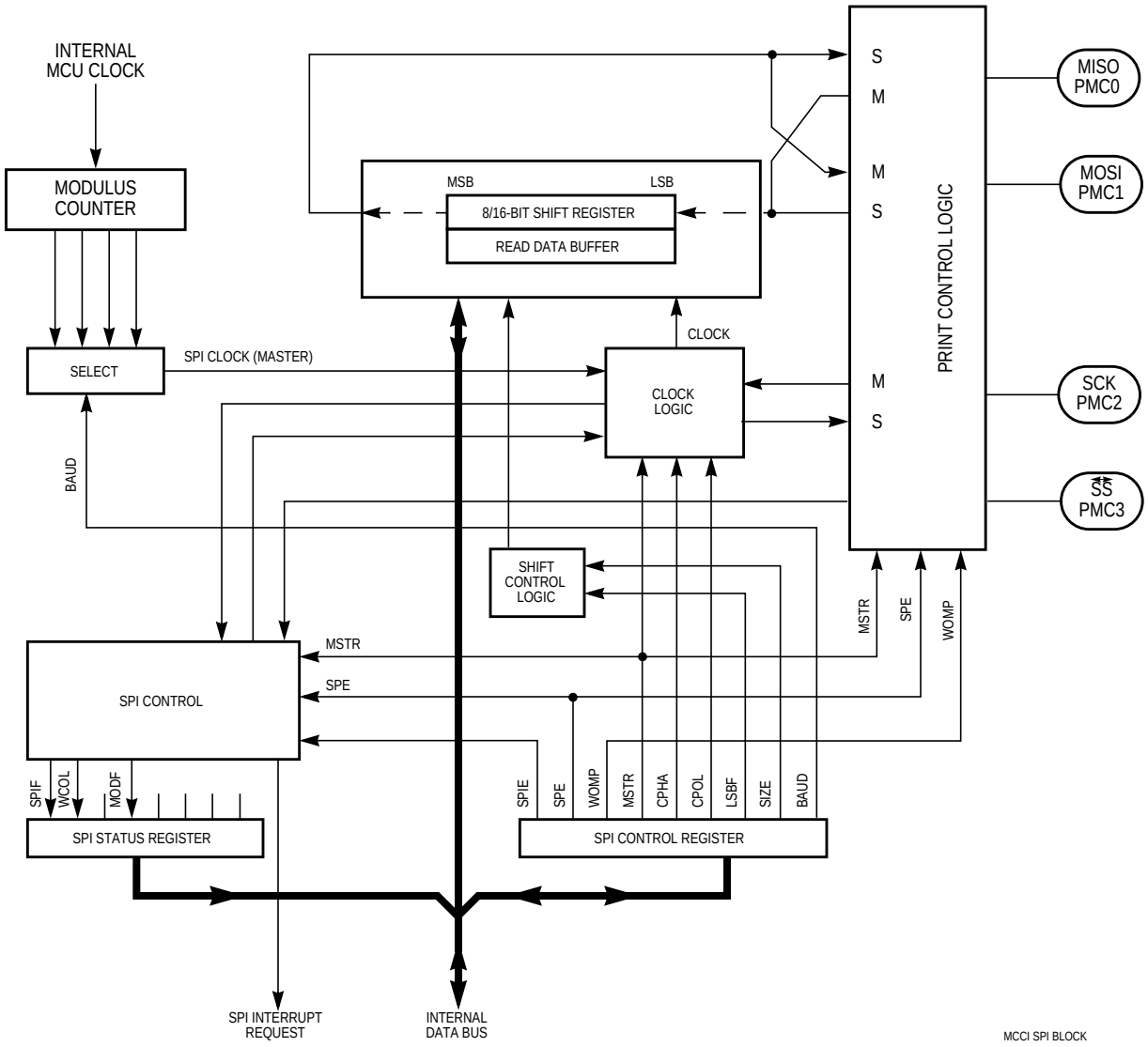


Figure 10-2 SPI Block Diagram

Clock control logic allows a selection of clock polarity and a choice of two clocking protocols to accommodate most available synchronous serial peripheral devices. When the SPI is configured as a master, software selects one of 254 different bit rates for the serial clock.

During an SPI transfer, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). A serial clock line synchronizes shifting and sampling of the information on the two serial data lines. A slave-select line allows individual selection of a slave SPI device. Slave devices which are not selected do not interfere with SPI bus activities. On a master SPI device the slave-select line can optionally be used to indicate a multiple-master bus contention.

Freescale Semiconductor, Inc.

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

Error-detection logic is included to support interprocessor interfacing. A write-collision detector indicates when an attempt is made to write data to the serial shift register while a transfer is in progress. A multiple-master mode-fault detector automatically disables SPI output drivers if more than one MCU simultaneously attempts to become bus master.

10.3.1 SPI Registers

SPI control registers include the SPI control register (SPCR), the SPI status register (SPSR), and the SPI data register (SPDR). Refer to **D.7.13 SPI Control Register**, **D.7.14 SPI Status Register**, and **D.7.15 SPI Data Register** for register bit and field definitions.

10.3.1.1 SPI Control Register (SPCR)

The SPCR contains parameters for configuring the SPI. The register can be read or written at any time.

10.3.1.2 SPI Status Register (SPSR)

The SPSR contains SPI status information. Only the SPI can set the bits in this register. The CPU reads the register to obtain status information.

10.3.1.3 SPI Data Register (SPDR)

The SPDR is used to transmit and receive data on the serial bus. A write to this register in the master device initiates transmission or reception of another byte or word. After a byte or word of data is transmitted, the SPIF status bit is set in both the master and slave devices.

A read of the SPDR actually reads a buffer. If the first SPIF is not cleared by the time a second transfer of data from the shift register to the read buffer is initiated, an overrun condition occurs. In cases of overrun the byte or word causing the overrun is lost.

A write to the SPDR is not buffered and places data directly into the shift register for transmission.

10.3.2 SPI Pins

Four bidirectional pins are associated with the SPI. The MPAR configures each pin for either SPI function or general-purpose I/O. The MDDR assigns each pin as either input or output. The WOMP bit in the SPI control register (SPCR) determines whether each SPI pin that is configured for output functions as an open-drain output or a normal CMOS output. The MDDR and WOMP assignments are valid regardless of whether the pins are configured for SPI use or general-purpose I/O.

The operation of pins configured for SCI use depends on whether the SCI is operating as a master or a slave, determined by the MSTR bit in the SPCR.

Table 10-3 shows SPI pins and their functions.

Freescale Semiconductor, Inc.

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

Table 10-3 SPI Pin Functions

Pin Name	Mode	Function
Master in, slave out (MISO)	Master	Provides serial data input to the SPI
	Slave	Provides serial data output from the SPI
Master out, slave in (MOSI)	Master	Provides serial output from the SPI
	Slave	Provides serial input to the SPI
Serial clock (SCK)	Master	Provides clock output from the SPI
	Slave	Provides clock input to the SPI
Slave select ($\overline{SS}$)	Master	Detects bus-master mode fault
	Slave	Selects the SPI for an externally-initiated serial transfer

10.3.3 SPI Operating Modes

The SPI operates in either master or slave mode. Master mode is used when the MCU originates data transfers. Slave mode is used when an external device initiates serial transfers to the MCU. The MSTR bit in SPCR selects master or slave operation.

10.3.3.1 Master Mode

Setting the MSTR bit in SPCR selects master mode operation. In master mode, the SPI can initiate serial transfers but cannot respond to externally initiated transfers. When the slave-select input of a device configured for master mode is asserted, a mode fault occurs.

When using the SPI in master mode, include the following steps:

1. Write to the MMCR, MIVR, and ILSPI. Refer to **10.5 MCCI Initialization** for more information.
2. Write to the MPAR to assign the following pins to the SPI: MISO, MOSI, and (optionally) $\overline{SS}$. MISO is used for serial data input in master mode, and MOSI is used for serial data output. Either or both may be necessary, depending on the particular application. $\overline{SS}$ is used to generate a mode fault in master mode. If this SPI is the only possible master in the system, the $\overline{SS}$ pin may be used for general-purpose I/O.
3. Write to the MDDR to direct the data flow on SPI pins. Configure the SCK (serial clock) and MOSI pins as outputs. Configure MISO and (optionally) $\overline{SS}$ as inputs.
4. Write to the SPCR to assign values for BAUD, CPHA, CPOL, SIZE, LSBF, WOMP, and SPIE. Set the MSTR bit to select master operation. Set the SPE bit to enable the SPI.
5. Enable the slave device.
6. Write appropriate data to the SPI data register to initiate the transfer.

When the SPI reaches the end of the transmission, it sets the SPIF flag in the SPSR. If the SPIE bit in the SPCR is set, an interrupt request is generated when SPIF is asserted. After the SPSR is read with SPIF set, and then the SPDR is read or written to, the SPIF flag is automatically cleared.

Freescale Semiconductor, Inc.

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

Data transfer is synchronized with the internally-generated serial clock (SCK). Control bits CPHA and CPOL in SPCR control clock phase and polarity. Combinations of CPHA and CPOL determine the SCK edge on which the master MCU drives outgoing data from the MOSI pin and latches incoming data from the MISO pin.

10.3.3.2 Slave Mode

Clearing the MSTR bit in SPCR selects slave mode operation. In slave mode, the SPI is unable to initiate serial transfers. Transfers are initiated by an external bus master. Slave mode is typically used on a multimaster SPI bus. Only one device can be bus master (operate in master mode) at any given time.

When using the SPI in slave mode, include the following steps:

1. Write to the MMCR and interrupt registers. Refer to **10.5 MCCI Initialization** for more information.
2. Write to the MPAR to assign the following pins to the SPI: MISO, MOSI, and $\overline{SS}$. MISO is used for serial data output in slave mode, and MOSI is used for serial data input. Either or both may be necessary, depending on the particular application. SCK is the input serial clock. $\overline{SS}$ selects the SPI when asserted.
3. Write to the MDDR to direct the data flow on SPI pins. Configure the SCK, MOSI, and $\overline{SS}$ pins as inputs. Configure MISO as an output.
4. Write to the SPCR to assign values for CPHA, CPOL, SIZE, LSBF, WOMP, and SPIE. Set the MSTR bit to select master operation. Set the SPE bit to enable the SPI. (The BAUD field in the SPCR of the slave device has no effect on SPI operation.)

When SPE is set and MSTR is clear, a low state on the $\overline{SS}$ pin initiates slave mode operation. The $\overline{SS}$ pin is used only as an input.

After a byte or word of data is transmitted, the SPI sets the SPIF flag. If the SPIE bit in SPCR is set, an interrupt request is generated when SPIF is asserted.

Transfer is synchronized with the externally generated SCK. The CPHA and CPOL bits determine the SCK edge on which the slave MCU latches incoming data from the MOSI pin and drives outgoing data from the MISO pin.

10.3.4 SPI Clock Phase and Polarity Controls

Two bits in the SPCR determine SCK phase and polarity. The clock polarity (CPOL) bit selects clock polarity (high true or low true clock). The clock phase control bit (CPHA) selects one of two transfer formats and affects the timing of the transfer. The clock phase and polarity should be the same for the master and slave devices. In some cases, the phase and polarity may be changed between transfers to allow a master device to communicate with slave devices with different requirements. The flexibility of the SPI system allows it to be directly interfaced to almost any existing synchronous serial peripheral.

Freescale Semiconductor, Inc.

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

10.3.4.1 CPHA = 0 Transfer Format

Figure 10-3 is a timing diagram of an 8-bit, MSB-first SPI transfer in which CPHA equals zero. Two waveforms are shown for SCK: one for CPOL equal to zero and another for CPOL equal to one. The diagram may be interpreted as a master or slave timing diagram since the SCK, MISO and MOSI pins are directly connected between the master and the slave. The MISO signal shown is the output from the slave and the MOSI signal shown is the output from the master. The $\overline{SS}$ line is the chip-select input to the slave.

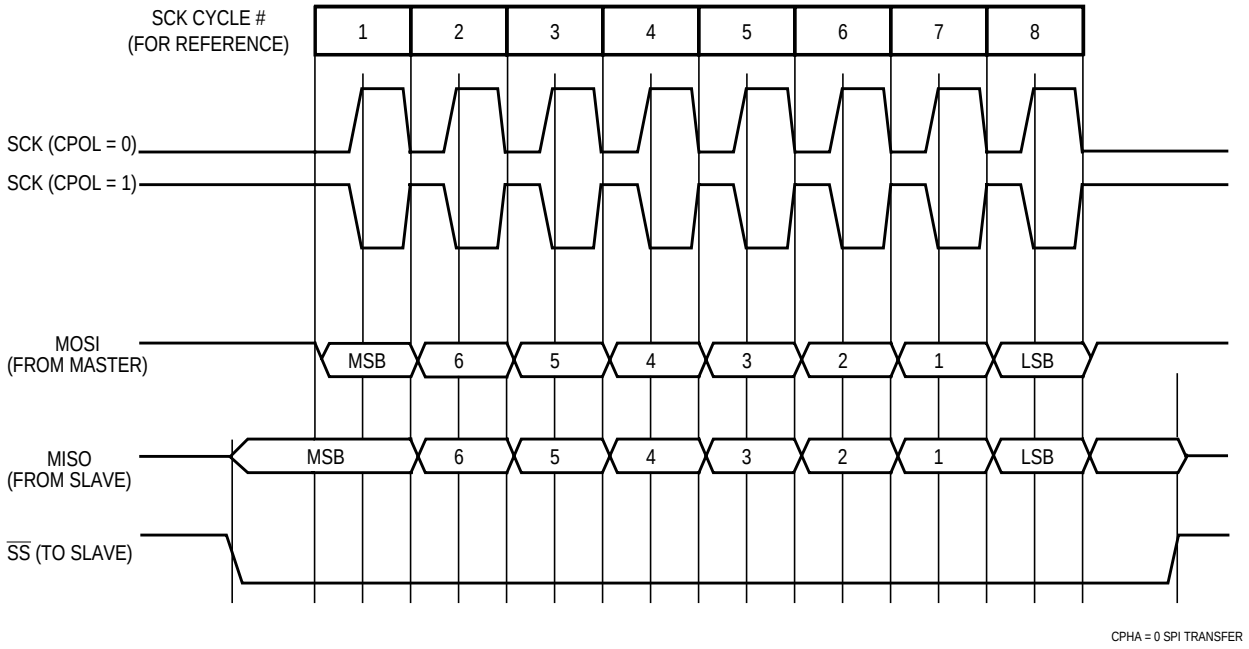


Figure 10-3 CPHA = 0 SPI Transfer Format

For a master, writing to the SPDR initiates the transfer. For a slave, the falling edge of $\overline{SS}$ indicates the start of a transfer. The SCK signal remains inactive for the first half of the first SCK cycle. Data is latched on the first and each succeeding odd clock edge, and the SPI shift register is left-shifted on the second and succeeding even clock edges. SPIF is set at the end of the eighth SCK cycle.

When CPHA equals zero, the $\overline{SS}$ line must be negated and reasserted between each successive serial byte. If the slave writes data to the SPI data register while $\overline{SS}$ is asserted (low), a write collision error results. To avoid this problem, the slave should read bit three of PORTMCP, which indicates the state of the $\overline{SS}$ pin, before writing to the SPDR again.

Freescale Semiconductor, Inc.

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

10.3.4.2 CPHA = 1 Transfer Format

Figure 10-4 is a timing diagram of an 8-bit, MSB-first SPI transfer in which CPHA equals one. Two waveforms are shown for SCK, one for CPOL equal to zero and another for CPOL equal to one. The diagram may be interpreted as a master or slave timing diagram since the SCK, MISO and MOSI pins are directly connected between the master and the slave. The MISO signal shown is the output from the slave and the MOSI signal shown is the output from the master. The $\overline{SS}$ line is the slave select input to the slave.

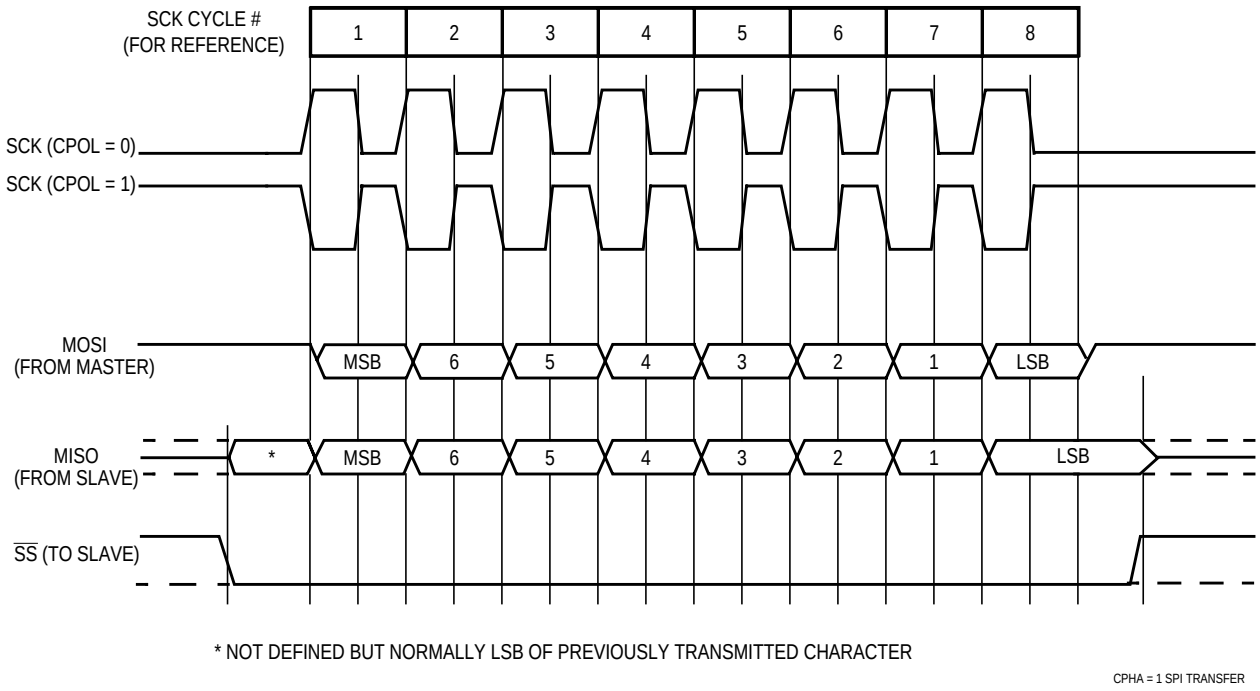


Figure 10-4 CPHA = 1 SPI Transfer Format

For a master, writing to the SPDR initiates the transfer. For a slave, the first edge of SCK indicates the start of a transfer. The SPI is left-shifted on the first and each succeeding odd clock edge, and data is latched on the second and succeeding even clock edges.

SCK is inactive for the last half of the eighth SCK cycle. For a master, SPIF is set at the end of the eighth SCK cycle (after the seventeenth SCK edge). Since the last SCK edge occurs in the middle of the eighth SCK cycle, however, the slave has no way of knowing when the end of the last SCK cycle occurs. The slave therefore considers the transfer complete after the last bit of serial data has been sampled, which corresponds to the middle of the eighth SCK cycle.

When CPHA is one, the $\overline{SS}$ line may remain at its active low level between transfers. This format is sometimes preferred in systems having a single fixed master and only one slave that needs to drive the MISO data line.

Freescale Semiconductor, Inc.

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

10.3.5 SPI Serial Clock Baud Rate

Baud rate is selected by writing a value from two to 255 into SPBR[7:0] in the SPCR of the master MCU. Writing an SPBR[7:0] value into the SPCR of the slave device has no effect. The SPI uses a modulus counter to derive SCK baud rate from the MCU system clock.

The following expressions apply to SCK baud rate:

$$\text{SCK Baud Rate} = \frac{f_{sys}}{2 \times \text{SPBR[7:0]}}$$

or

$$\text{SPBR[7:0]} = \frac{f_{sys}}{2 \times \text{SCK Baud Rate Desired}}$$

Giving SPBR[7:0] a value of zero or one disables the baud rate generator. SCK is disabled and assumes its inactive state value.

SPBR[7:0] has 254 active values. **Table 10-4** lists several possible baud values and the corresponding SCK frequency based on a 16.78-MHz system clock.

Table 10-4 SCK Frequencies

System Clock Frequency	Required Division Ratio	Value of SPBR	Actual SCK Frequency
16.78 MHz	4	2	4.19 MHz
	8	4	2.10 MHz
	16	8	1.05 MHz
	34	17	493 kHz
	168	84	100 kHz
	510	255	33 kHz

10.3.6 Wired-OR Open-Drain Outputs

Typically, SPI bus outputs are not open-drain unless multiple SPI masters are in the system. If needed, the WOMP bit in SPCR can be set to provide wired-OR, open-drain outputs. An external pull-up resistor should be used on each output line. WOMP affects all SPI pins regardless of whether they are assigned to the SPI or used as general-purpose I/O.

10.3.7 Transfer Size and Direction

The SIZE bit in the SPCR selects a transfer size of eight (SIZE = 0) or sixteen (SIZE = 1) bits. The LSBF bit in the SPCR determines whether serial shifting to and from the data register begins with the LSB (LSBF = 1) or MSB (LSBF = 0).

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

10.3.8 Write Collision

A write collision occurs if an attempt is made to write the SPDR while a transfer is in progress. Since the SPDR is not double buffered in the transmit direction, a successful write to SPDR would cause data to be written directly into the SPI shift register. Because this would corrupt any transfer in progress, a write collision error is generated instead. The transfer continues undisturbed, the data that caused the error is not written to the shifter, and the WCOL bit in SPSR is set. No SPI interrupt is generated.

A write collision is normally a slave error because a slave has no control over when a master initiates a transfer. Since a master is in control of the transfer, software can avoid a write collision error generated by the master. The SPI logic can, however, detect a write collision in a master as well as in a slave.

What constitutes a transfer in progress depends on the SPI configuration. For a master, a transfer starts when data is written to the SPDR and ends when SPIF is set. For a slave, the beginning and ending points of a transfer depend on the value of CPHA. When CPHA = 0, the transfer begins when $\overline{SS}$ is asserted and ends when it is negated. When CPHA = 1, a transfer begins at the edge of the first SCK cycle and ends when SPIF is set. Refer to **10.3.4 SPI Clock Phase and Polarity Controls** for more information on transfer periods and on avoiding write collision errors.

When a write collision occurs, the WCOL bit in the SPSR is set. To clear WCOL, read the SPSR while WCOL is set, and then either read the SPDR (either before or after SPIF is set) or write the SPDR after SPIF is set. (Writing the SPDR before SPIF is set results in a second write collision error.) This process clears SPIF as well as WCOL.

10.3.9 Mode Fault

When the SPI system is configured as a master and the $\overline{SS}$ input line is asserted, a mode fault error occurs, and the MODF bit in the SPSR is set. Only an SPI master can experience a mode fault error, caused when a second SPI device becomes a master and selects this device as if it were a slave.

To avoid latchup caused by contention between two pin drivers, the MCU does the following when it detects a mode fault error:

1. Forces the MSTR control bit to zero to reconfigure the SPI as a slave.
2. Forces the SPE control bit to zero to disable the SPI system.
3. Sets the MODF status flag and generates an SPI interrupt if SPIE = 1.
4. Clears the appropriate bits in the MDDR to configure all SPI pins except the $\overline{SS}$ pin as inputs.

After correcting the problems that led to the mode fault, clear MODF by reading the SPSR while MODF is set and then writing to the SPCR. Control bits SPE and MSTR may be restored to their original set state during this clearing sequence or after the MODF bit has been cleared. Hardware does not allow the user to set the SPE and MSTR bits while MODF is a logic one except during the proper clearing sequence.

Freescale Semiconductor, Inc.

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

10.4 Serial Communication Interface (SCI)

The SCI submodule contains two independent SCI systems. Each is a full-duplex universal asynchronous receiver transmitter (UART). This SCI system is fully compatible with SCI systems found on other Freescale devices, such as the M68HC11 and M68HC05 families.

The SCI uses a standard non-return to zero (NRZ) transmission format. An on-chip baud-rate generator derives standard baud-rate frequencies from the MCU oscillator. Both the transmitter and the receiver are double buffered, so that back-to-back characters can be handled easily even if the CPU is delayed in responding to the completion of an individual character. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate.

Figure 10-5 shows a block diagram of the SCI transmitter. **Figure 10-6** shows a block diagram of the SCI receiver.

The two independent SCI systems are called SCIA and SCIB. These SCIs are identical in register set and hardware configuration, providing an application with full flexibility in using the dual SCI system. References to SCI registers in this section do not always distinguish between the two SCI systems. A reference to SCCR1, for example, applies to both SCCR1A (SCIA control register 1) and SCCR1B (SCIB control register 1).

10.4.1 SCI Registers

The SCI programming model includes the MCCI global and pin control registers and eight SCI registers. Each of the two SCI units contains two SCI control registers, one status register, and one data register. Refer to **D.7.9 SCI Control Register 0**, **D.7.11 SCI Status Register**, and **D.7.12 SCI Data Register** for register bit and field definitions.

All registers may be read or written at any time by the CPU. Rewriting the same value to any SCI register does not disrupt operation; however, writing a different value into an SCI register when the SCI is running may disrupt operation. To change register values, the receiver and transmitter should be disabled with the transmitter allowed to finish first. The status flags in the SCSR may be cleared at any time.

When initializing the SCI, set the transmitter enable (TE) and receiver enable (RE) bits in SCCR1 last. A single word write to SCCR1 can be used to initialize the SCI and enable the transmitter and receiver.

10.4.1.1 SCI Control Registers

SCCR0 contains the baud rate selection field. The baud rate must be set before the SCI is enabled. The CPU16 can read and write this register at any time.

Freescale Semiconductor, Inc.

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

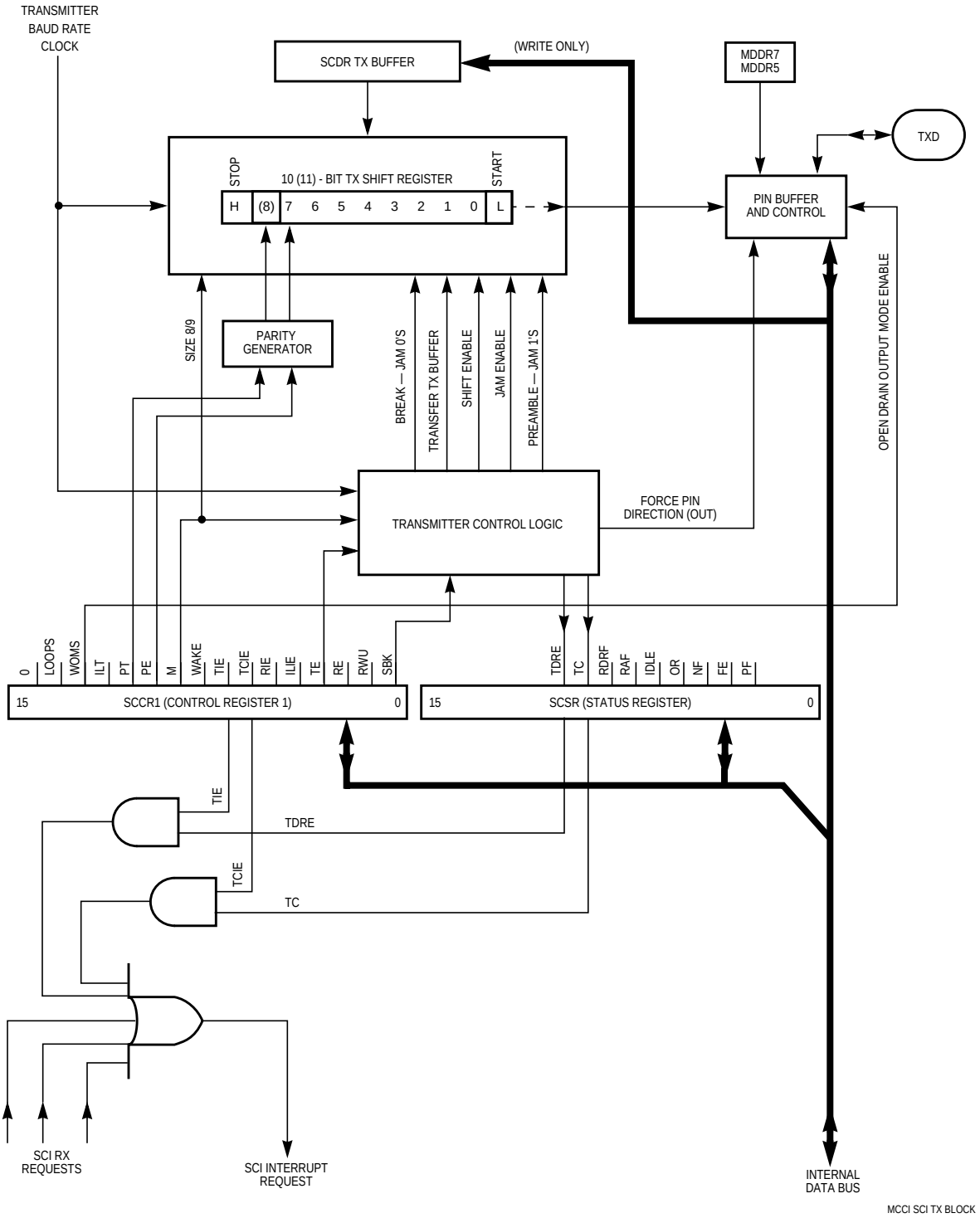


Figure 10-5 SCI Transmitter Block Diagram

Freescale Semiconductor, Inc.

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

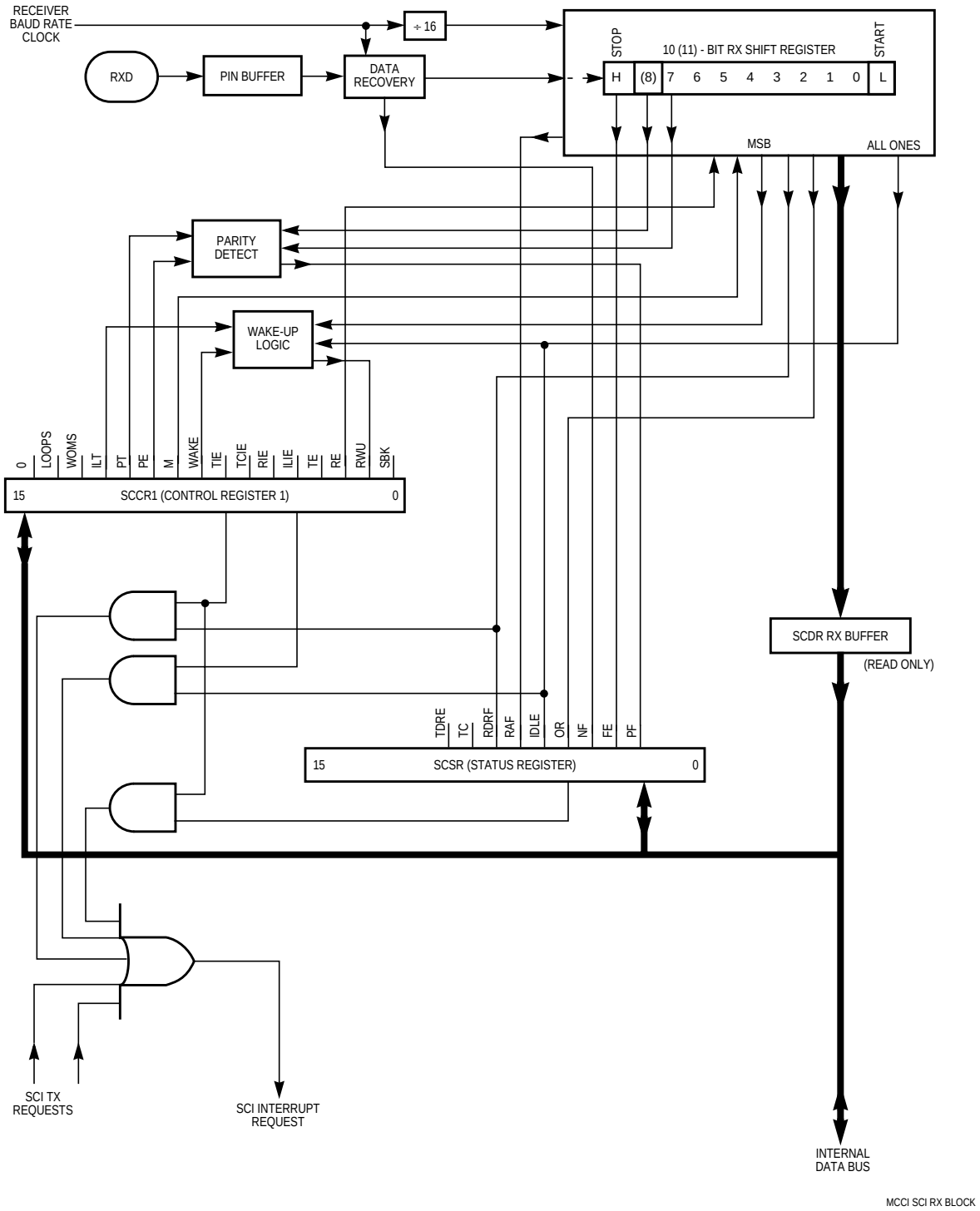


Figure 10-6 SCI Receiver Block Diagram

Freescale Semiconductor, Inc.

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

SCCR1 contains a number of SCI configuration parameters, including transmitter and receiver enable bits, interrupt enable bits, and operating mode enable bits. The CPU16 can read and write this register at any time. The SCI can modify the RWU bit under certain circumstances.

Changing the value of SCI control bits during a transfer may disrupt operation. Before changing register values, allow the SCI to complete the current transfer, then disable the receiver and transmitter.

10.4.1.2 SCI Status Register

The SCSR contains flags that show SCI operating conditions. These flags are cleared either by SCI hardware or by a read/write sequence. To clear SCI transmitter flags, read the SCSR and then write to the SCDR. To clear SCI receiver flags, read the SCSR and then read the SCDR. A long-word read can consecutively access both the SCSR and the SCDR. This action clears receiver status flag bits that were set at the time of the read, but does not clear TDRE or TC flags.

If an internal SCI signal for setting a status bit comes after the CPU has read the asserted status bits, but before the CPU has written or read the SCDR, the newly set status bit is not cleared. The SCSR must be read again with the bit set, and the SCDR must be written to or read before the status bit is cleared.

Reading either byte of the SCSR causes all 16 bits to be accessed, and any status bit already set in either byte will be cleared on a subsequent read or write of the SCDR.

10.4.1.3 SCI Data Register

The SCDR contains two data registers at the same address. The RDR is a read-only register that contains data received by the SCI serial interface. The data comes into the receive serial shifter and is transferred to the RDR. The TDR is a write-only register that contains data to be transmitted. The data is first written to the TDR, then transferred to the transmit serial shifter, where additional format bits are added before transmission.

10.4.2 SCI Pins

Four pins are associated with the SCI: TXDA, TXDB, RXDA, and RXDB. The state of the TE or RE bit in SCI control register 1 of each SCI submodule (SCCR1A, SCCR1B) determines whether the associated pin is configured for SCI operation or general-purpose I/O. The MDDR assigns each pin as either input or output. The WOMC bit in SCCR1A or SCCR1B determines whether the associated RXD and TXD pins, when configured as outputs, function as open-drain output pins or normal CMOS outputs. The MDDR and WOMC assignments are valid regardless of whether the pins are configured for SPI use or general-purpose I/O.

SCI pins are listed in **Table 10-5**.

Freescale Semiconductor, Inc.

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

Table 10-5 SCI Pins

Pin	Mode	SCI Function	Port I/O Signal
Transmit data	TXDA	Serial data output from SCIA (TE = 1)	PMC7
	TXDB	Serial data output from SCIB (TE = 1)	PMC5
Receive data	RXDA	Serial data input to SCIA (RE = 1)	PMC6
	RXDB	Serial data input to SCIB (RE = 1)	PMC4

10.4.3 Receive Data Pins (RXDA, RXDB)

RXDA and RXDB are the serial data inputs to the SCIA and SCIB interfaces, respectively. Each pin is also available as a general-purpose I/O pin when the RE bit in SCCR1 of the associated SCI submodule is cleared. When used for general-purpose I/O, RXDA and RXDB may be configured either as input or output as determined by the RXDA and RXDB bits in the MDDR.

10.4.4 Transmit Data Pins (TXDA, TXDB)

When used for general-purpose I/O, TXDA and TXDB can be configured either as input or output as determined by the TXDA and TXDB bits in the MDDR. The TXDA and TXDB pins are enabled for SCI use by setting the TE bit in SCCR1 of each SCI interface.

10.4.5 SCI Operation

SCI operation can be polled by means of status flags in the SCSR, or interrupt-driven operation can be employed by means of the interrupt-enable bits in SCCR1.

10.4.5.1 Definition of Terms

Data can be transmitted and received in a number of formats. The following terms concerning data format are used in this section:

- Bit-Time — The time required to transmit or receive one bit of data, which is equal to one cycle of the baud frequency.
- 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 and be preceded by at least three receive time samples of logic one.
- Stop Bit — One bit-time of logic one that indicates the end of a data frame.
- Frame — A complete unit of serial information. The SCI can use 10-bit or 11-bit frames.
- Data Frame — A start bit, a specified number of data or information bits, and at least one stop bit.
- Idle Frame — A frame that consists of consecutive ones. An idle frame has no start bit.
- Break Frame — A frame that consists of consecutive zeros. A break frame has no stop bits.

Freescale Semiconductor, Inc.

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

10.4.5.2 Serial Formats

All data frames must have a start bit and at least one stop bit. Receiving and transmitting devices must use the same data frame format. The SCI provides hardware support for both 10-bit and 11-bit frames. The M bit in SCCR1 specifies the number of bits per frame.

The most common data frame format for NRZ serial interfaces is one start bit, eight data bits (LSB first), and one stop bit; a total of ten bits. The most common 11-bit data frame contains one start bit, eight data bits, a parity or control bit, and one stop bit. Ten-bit and eleven-bit frames are shown in **Table 10-6**.

Table 10-6 Serial Frame Formats

10-Bit Frames			
Start	**Data**	**Parity/Control**	**Stop**
1	7	—	2
1	7	1	1
1	8	—	1
11-Bit Frames			
Start	**Data**	**Parity/Control**	**Stop**
1	7	1	2
1	8	1	1

10.4.5.3 Baud Clock

The SCI baud rate is programmed by writing a 13-bit value to the SCBR field in SCI control register zero (SCCR0). The baud rate is derived from the MCU system clock by a modulus counter. Writing a value of zero to SCBR[12:0] disables the baud rate generator. Baud rate is calculated as follows:

$$\text{SCI Baud Rate} = \frac{f_{sys}}{32 \times \text{SCBR[12:0]}}$$

or

$$\text{SCBR[12:0]} = \frac{f_{sys}}{32 \times \text{SCI Baud Rate Desired}}$$

where SCBR[12:0] is in the range {1, 2, 3, ..., 8191}.

The SCI receiver operates asynchronously. An internal clock is necessary to synchronize with an incoming data stream. The SCI baud rate generator produces a receive time sampling clock with a frequency 16 times that of the SCI baud rate. The SCI determines the position of bit boundaries from transitions within the received waveform, and adjusts sampling points to the proper positions within the bit period.

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

10.4.5.4 Parity Checking

The PT bit in SCCR1 selects either even (PT = 0) or odd (PT = 1) parity. PT affects received and transmitted data. The PE bit in SCCR1 determines whether parity checking is enabled (PE = 1) or disabled (PE = 0). When PE is set, the MSB of data in a frame is used for the parity function. For transmitted data, a parity bit is generated for received data; the parity bit is checked. When parity checking is enabled, the PF bit in the SCI status register (SCSR) is set if a parity error is detected.

Enabling parity affects the number of data bits in a frame, which can in turn affect frame size. **Table 10-7** shows possible data and parity formats.

Table 10-7 Effect of Parity Checking on Data Size

M	PE	Result
0	0	8 data bits
0	1	7 data bits, 1 parity bit
1	0	9 data bits
1	1	8 data bits, 1 parity bit

10.4.5.5 Transmitter Operation

The transmitter consists of a serial shifter and a parallel data register (TDR) located in the SCI data register (SCDR). The serial shifter cannot be directly accessed by the CPU16. The transmitter is double-buffered, which means that data can be loaded into the TDR while other data is shifted out. The TE bit in SCCR1 enables (TE = 1) and disables (TE = 0) the transmitter.

Shifter output is connected to the TXD pin while the transmitter is operating (TE = 1, or TE = 0 and transmission in progress). Wired-OR operation should be specified when more than one transmitter is used on the same SCI bus. The WOMS bit in SCCR1 determines whether TXD is an open-drain (wired-OR) output or a normal CMOS output. An external pull-up resistor on TXD is necessary for wired-OR operation. WOMS controls TXD function whether the pin is used by the SCI or as a general-purpose I/O pin.

Data to be transmitted is written to SCDR, then transferred to the serial shifter. The transmit data register empty (TDRE) flag in SCSR shows the status of TDR. When TDRE = 0, the TDR contains data that has not been transferred to the shifter. Writing to SCDR again overwrites the data. TDRE is set when the data in the TDR is transferred to the shifter. Before new data can be written to the SCDR, however, the processor must clear TDRE by writing to SCSR. If new data is written to the SCDR without first clearing TDRE, the data will not be transmitted.

The transmission complete (TC) flag in SCSR shows transmitter shifter state. When TC = 0, the shifter is busy. TC is set when all shifting operations are completed. TC is not automatically cleared. The processor must clear it by first reading SCSR while TC is set, then writing new data to SCDR.

Freescale Semiconductor, Inc.

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

The state of the serial shifter is checked when the TE bit is set. If TC = 1, an idle frame is transmitted as a preamble to the following data frame. If TC = 0, the current operation continues until the final bit in the frame is sent, then the preamble is transmitted. The TC bit is set at the end of preamble transmission.

The SBK bit in SCCR1 is used to insert break frames in a transmission. A non-zero integer number of break frames is transmitted while SBK is set. Break transmission begins when SBK is set, and ends with the transmission in progress at the time either SBK or TE is cleared. If SBK is set while a transmission is in progress, that transmission finishes normally before the break begins. To assure the minimum break time, toggle SBK quickly to one and back to zero. The TC bit is set at the end of break transmission. After break transmission, at least one bit-time of logic level one (mark idle) is transmitted to ensure that a subsequent start bit can be detected.

If TE remains set, after all pending idle, data and break frames are shifted out, TDRE and TC are set and TXD is held at logic level one (mark).

When TE is cleared, the transmitter is disabled after all pending idle, data, and break frames are transmitted. The TC flag is set, and control of the TXD pin reverts to MPAR and MDDR. Buffered data is not transmitted after TE is cleared. To avoid losing data in the buffer, do not clear TE until TDRE is set.

Some serial communication systems require a mark on the TXD pin even when the transmitter is disabled. Configure the TXD pin as an output, then write a one to PQS7. When the transmitter releases control of the TXD pin, it reverts to driving a logic one output.

To insert a delimiter between two messages, to place non-listening receivers in wake-up mode between transmissions, or to signal a retransmission by forcing an idle line, clear and then set TE before data in the serial shifter has shifted out. The transmitter finishes the transmission, then sends a preamble. After the preamble is transmitted, if TDRE is set, the transmitter will mark idle. Otherwise, normal transmission of the next sequence will begin.

Both TDRE and TC have associated interrupts. The interrupts are enabled by the transmit interrupt enable (TIE) and transmission complete interrupt enable (TCIE) bits in SCCR1. Service routines can load the last byte of data in a sequence into SCDR, then terminate the transmission when a TDRE interrupt occurs.

10.4.5.6 Receiver Operation

The RE bit in SCCR1 enables (RE = 1) and disables (RE = 0) the receiver. The receiver contains a receive serial shifter and a parallel receive data register (RDR) located in the SCI data register (SCDR). The serial shifter cannot be directly accessed by the CPU16. The receiver is double-buffered, allowing data to be held in the RDR while other data is shifted in.

Receiver bit processor logic drives a state machine that determines the logic level for each bit-time. This state machine controls when the bit processor logic is to sample the RXD pin and also controls when data is to be passed to the receive serial shifter.

Freescale Semiconductor, Inc.

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

A receive time clock is used to control sampling and synchronization. Data is shifted into the receive serial shifter according to the most recent synchronization of the receive time clock with the incoming data stream. From this point on, data movement is synchronized with the MCU system clock.

The number of bits shifted in by the receiver depends on the serial format. However, all frames must end with at least one stop bit. When the stop bit is received, the frame is considered to be complete, and the received data in the serial shifter is transferred to the RDR. The receiver data register flag (RDRF) is set when the data is transferred.

Noise errors, parity errors, and framing errors can be detected while a data stream is being received. Although error conditions are detected as bits are received, the noise flag (NF), the parity flag (PF), and the framing error flag (FE) in SCSR are not set until data is transferred from the serial shifter to the RDR.

RDRF must be cleared before the next transfer from the shifter can take place. If RDRF is set when the shifter is full, transfers are inhibited and the overrun error (OR) flag in SCSR is set. OR indicates that the RDR needs to be serviced faster. When OR is set, the data in the RDR is preserved, but the data in the serial shifter is lost. Because framing, noise, and parity errors are detected while data is in the serial shifter, FE, NF, and PF cannot occur at the same time as OR.

When the CPU16 reads SCSR and SCDR in sequence, it acquires status and data, and also clears the status flags. Reading SCSR acquires status and arms the clearing mechanism. Reading SCDR acquires data and clears SCSR.

When RIE in SCCR1 is set, an interrupt request is generated whenever RDRF is set. Because receiver status flags are set at the same time as RDRF, they do not have separate interrupt enables.

10.4.5.7 Idle-Line Detection

During a typical serial transmission, frames are transmitted isochronally and no idle time occurs between frames. Even when all the data bits in a frame are logic ones, the start bit provides one logic zero bit-time during the frame. An idle line is a sequence of contiguous ones equal to the current frame size. Frame size is determined by the state of the M bit in SCCR1.

The SCI receiver has both short and long idle-line detection capability. Idle-line detection is always enabled. The idle line type (ILT) bit in SCCR1 determines which type of detection is used. When an idle line condition is detected, the IDLE flag in SCSR is set.

For short idle-line detection, the receiver bit processor counts contiguous logic one bit-times whenever they occur. Short detection provides the earliest possible recognition of an idle line condition, because the stop bit and contiguous logic ones before and after it are counted. For long idle-line detection, the receiver counts logic ones after the stop bit is received. Only a complete idle frame causes the IDLE flag to be set.

Freescale Semiconductor, Inc.

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

In some applications, software overhead can cause a bit-time of logic level one to occur between frames. This bit-time does not affect content, but if it occurs after a frame of ones when short detection is enabled, the receiver flags an idle line.

When the ILIE bit in SCCR1 is set, an interrupt request is generated when the IDLE flag is set. The flag is cleared by reading SCSR and SCDR in sequence. IDLE is not set again until after at least one frame has been received (RDRF = 1). This prevents an extended idle interval from causing more than one interrupt.

10.4.5.8 Receiver Wake-Up

The receiver wake-up function allows a transmitting device to direct a transmission to a single receiver or to a group of receivers by sending an address frame at the start of a message. Hardware activates each receiver in a system under certain conditions. Resident software must process address information and enable or disable receiver operation.

A receiver is placed in wake-up mode by setting the RWU bit in SCCR1. While RWU is set, receiver status flags and interrupts are disabled. Although the CPU32 can clear RWU, it is normally cleared by hardware during wake-up.

The WAKE bit in SCCR1 determines which type of wake-up is used. When WAKE = 0, idle-line wake-up is selected. When WAKE = 1, address-mark wake-up is selected. Both types require a software-based device addressing and recognition scheme.

Idle-line wake-up allows a receiver to sleep until an idle line is detected. When an idle-line is detected, the receiver clears RWU and wakes up. The receiver waits for the first frame of the next transmission. The byte is received normally, transferred to the RDR, and the RDRF flag is set. If software does not recognize the address, it can set RWU and put the receiver back to sleep. For idle-line wake-up to work, there must be a minimum of one frame of idle line between transmissions. There must be no idle time between frames within a transmission.

Address-mark wake-up uses a special frame format to wake up the receiver. When the MSB of an address-mark frame is set, that frame contains address information. The first frame of each transmission must be an address frame. When the MSB of a frame is set, the receiver clears RWU and wakes up. The byte is received normally, transferred to the RDR, and the RDRF flag is set. If software does not recognize the address, it can set RWU and put the receiver back to sleep. Address-mark wake-up allows idle time between frames and eliminates idle time between transmissions. However, there is a loss of efficiency because of an additional bit-time per frame.

10.4.5.9 Internal Loop

The LOOPS bit in SCCR1 controls a feedback path in the data serial shifter. When LOOPS is set, the SCI transmitter output is fed back into the receive serial shifter. TXD is asserted (idle line). Both transmitter and receiver must be enabled before entering loop mode.

Freescale Semiconductor, Inc.

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

10.5 MCCI Initialization

After reset, the MCCI remains in an idle state. Several registers must be initialized before serial operations begin. A general sequence guide for initialization follows.

A. Global
   1. Configure MMCR
      a. Write an interrupt arbitration number greater than zero into the IARB field.
      b. Clear the STOP bit if it is not already cleared.
   2. Interrupt vector and interrupt level registers (MIVR, ILSPI, and ILSCI)
      a. Write the SPI/SCI interrupt vector into MIVR.
      b. Write the SPI interrupt request level into the ILSPI and the interrupt request levels for the two SCI interfaces into the ILSCI.
   3. Port data register
      a. Write a data word to PORTMC.
      b. Read a port pin state from PORTMCP.
   4. Pin control registers
      a. Establish the direction of MCCI pins by writing to the MDDR.
      b. Assign pin functions by writing to the MPAR.

B. Serial Peripheral Interface
   1. Configure SPCR
      a. Write a transfer rate value into the BAUD field.
      b. Determine clock phase (CPHA) and clock polarity (CPOL).
      c. Specify an 8- or 16-bit transfer (SIZE) and MSB- or LSB-first transfer mode (LSBF).
      d. Select master or slave operating mode (MSTR).
      e. Enable or disable wired-OR operation (WOMP).
      f. Enable or disable SPI interrupts (SPIE).
      g. Enable the SPI by setting the SPE bit.

C. Serial Communication Interface (SCIA/SCIB)
   1. To transmit, read the SCSR, and then write transmit data to the SCDR. This clears the TDRE and TC indicators in the SCSR.
      a. SCI control register 0 (SCCR0)
      b. Write a baud rate value into the BR field.
   2. Configure SCCR1
      a. Select 8- or 9-bit frame format (M).
      b. Determine use (PE) and type (PT) of parity generation or detection.
      c. To receive, set the RE and RIE bits in SCCR1. Select use (RWU) and type (WAKE) of receiver wakeup. Select idle-line detection type (ILT) and enable or disable idle-line interrupt (ILIE).
      d. To transmit, set TE and TIE bits in SCCR1, and enable or disable WOMC and TCIE bits. Disable break transmission (SBK) for 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

SECTION 11
GENERAL-PURPOSE TIMER

This section is an overview of the general-purpose timer (GPT) function. Refer to the *GPT Reference Manual* (GPTRM/AD) for complete information about the GPT module.

11.1 General

The 11-channel general-purpose timer (GPT) is used in systems where a moderate level of CPU control is required. The GPT consists of a capture/compare unit, a pulse accumulator, and two pulse-width modulators. A bus interface unit connects the GPT to the intermodule bus (IMB). **Figure 11-1** is a block diagram of the GPT.

The capture/compare unit features three input capture channels, four output compare channels, and one channel that can be selected as input capture or output compare. These channels share a 16-bit free-running counter (TCNT) that derives its clock from a nine-stage prescaler or from the external clock input signal, PCLK.

Pulse accumulator channel logic includes an 8-bit counter. The pulse accumulator can operate in either event counting mode or gated time accumulation mode.

Pulse-width modulator outputs are periodic waveforms whose duty cycles can be independently selected and modified by user software. The PWM circuits share a 16-bit free-running counter that can be clocked by the same nine-stage prescaler used by the capture/compare unit or by the PCLK input.

All GPT pins can also be used for general-purpose input/output. The input capture and output compare pins form a bidirectional 8-bit parallel port (port GP). PWM pins are outputs only. PAI and PCLK pins are inputs only.

Freescale Semiconductor, Inc.

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

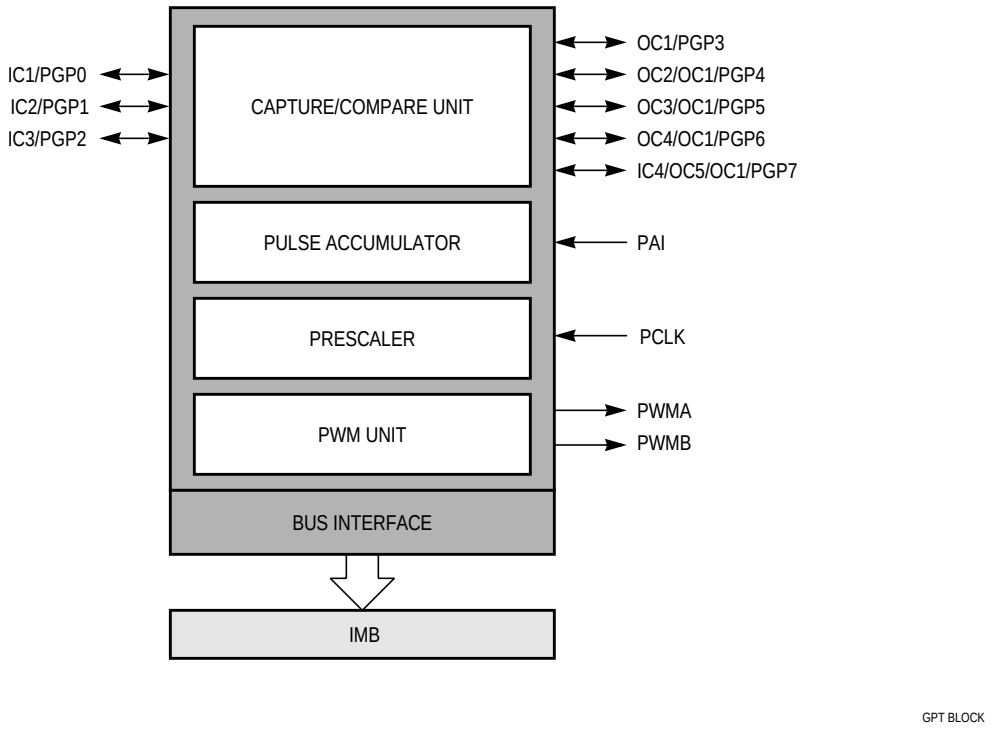


Figure 11-1 GPT Block Diagram

11.2 GPT Registers and Address Map

The GPT programming model consists of a configuration register (GPTMCR), parallel I/O registers (DDRGP, PORTGP), capture/compare registers (TCNT, TCTL1, TCTL2, TIC[1:3], TOC[1:4], TI4/O5, CFORC), pulse accumulator registers (PACNT, PACTL), pulse-width modulation registers (PWMA, PWMB, PWMC, PWMCNT, PWMBUFA, PWMBUFB), status registers (TFLG1, TFLG2) and interrupt control registers (TMSK1, TMSK2). Functions of the module configuration register are discussed in **11.3 Special Modes of Operation** and **11.4 Polled and Interrupt-Driven Operation**. Other register functions are discussed in the appropriate sections.

All registers can be accessed using byte or word operations. Certain capture/compare registers and pulse-width modulation registers must be accessed by word operations to ensure coherency. If byte accesses are used to read a register such as the timer counter register (TCNT), there is a possibility that data in the byte not being accessed will change while the other byte is read. Both bytes must be accessed at the same time.

The modmap (MM) bit in the system integration module configuration register (SIMCR) defines the most significant bit (ADDR23) of the IMB address for each register in the MCU. Because the CPU16 drives ADDR[23:20] to the same logic state as ADDR[19:0], MM must equal one.

Freescale Semiconductor, Inc.

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

Refer to **D.8 General-Purpose Timer** for a GPT address map and register bit/field descriptions. Refer to **5.2.1 Module Mapping** for more information about how the state of MM affects the system.

11.3 Special Modes of Operation

The GPT module configuration register (GPTMCR) is used to control special GPT operating modes. These include low-power stop mode, freeze mode, single-step mode, and test mode. Normal GPT operation can be polled or interrupt-driven. Refer to **11.4 Polled and Interrupt-Driven Operation** for more information.

11.3.1 Low-Power Stop Mode

Low-power stop operation is initiated by setting the STOP bit in GPTMCR. In stop mode the system clock to the module is turned off. The clock remains off until STOP is negated or a reset occurs. All counters and prescalers within the timer stop counting while the STOP bit is set. Only the module configuration register (GPTMCR) and the interrupt configuration register (ICR) should be accessed while in the stop mode. Accesses to other GPT registers cause unpredictable behavior. Low-power stop can also be used to disable module operation during debugging.

11.3.2 Freeze Mode

The freeze (FRZ[1:0]) bits in GPTMCR are used to determine what action is taken by the GPT when the IMB FREEZE signal is asserted. FREEZE is asserted when the CPU enters background debug mode. At the present time, FRZ1 is not implemented; FRZ0 causes the GPT to enter freeze mode. Refer to **4.14.4 Background Debug Mode** for more information.

Freeze mode freezes the current state of the timer. The prescaler and the pulse accumulator do not increment and changes to the pins are ignored (input pin synchronizers are not clocked). All of the other timer functions that are controlled by the CPU operate normally. For example, registers can be written to change pin directions, force output compares, and read or write I/O pins.

While the FREEZE signal is asserted, the CPU has write access to registers and bits that are normally read-only or write-once. The write-once bits can be written to as often as needed. The prescaler and the pulse accumulator remain stopped and the input pins are ignored until the FREEZE signal is negated (the CPU is no longer in BDM), the FRZ0 bit is cleared, or the MCU is reset.

Activities that are in progress before FREEZE assertion are completed. For example, if an input edge on an input capture pin is detected just as the FREEZE signal is asserted, the capture occurs and the corresponding interrupt flag is set.

Freescale Semiconductor, Inc.

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

11.3.3 Single-Step Mode

Two bits in GPTMCR support GPT debugging without using BDM. When the STOPP bit is asserted, the prescaler and the pulse accumulator stop counting and changes at input pins are ignored. Reads of the GPT pins return the state of the pin when STOPP was set. After STOPP is set, the INCP bit can be set to increment the prescaler and clock the input synchronizers once. The INCP bit is self-negating after the prescaler is incremented. INCP can be set repeatedly. The INCP bit has no effect when the STOPP bit is not set.

11.3.4 Test Mode

Test mode is used during Freescale factory testing. The GPT has no dedicated test-mode control register; all GPT testing is done under control of the system integration module.

11.4 Polled and Interrupt-Driven Operation

Normal GPT function can be polled or interrupt-driven. All GPT functions have an associated status flag and an associated interrupt. The timer interrupt flag registers (TFLG1 and TFLG2) contain status flags used for polled and interrupt-driven operation. The timer mask registers (TMSK1 and TMSK2) contain interrupt control bits. Control routines can monitor GPT operation by polling the status registers. When an event occurs, the control routine transfers control to a service routine that handles that event. If interrupts are enabled for an event, the GPT requests interrupt service when the event occurs. Using interrupts does not require continuously polling the status flags to see if an event has taken place. However, to disable the interrupt request, status flags must be cleared after an interrupt is serviced.

11.4.1 Polled Operation

When an event occurs in the GPT, that event sets a status flag in TFLG1 or TFLG2. The GPT sets the flags; they cannot be set by the CPU. TFLG1 and TFLG2 are 8-bit registers that can be accessed individually or as one 16-bit register. The registers are initialized to zero at reset. **Table 11-1** shows status flag assignment.

Freescale Semiconductor, Inc.

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

Table 11-1 GPT Status Flags

Flag Mnemonic	Register Assignment	Source
IC1F	TFLG1	Input capture 1
IC2F	TFLG1	Input capture 2
IC3F	TFLG1	Input capture 3
OC1F	TFLG1	Output compare 1
OC2F	TFLG1	Output compare 2
OC3F	TFLG1	Output compare 3
OC4F	TFLG1	Output compare 4
I4/O5F	TFLG1	Input capture 4/output compare 5
TOF	TFLG2	Timer overflow
PAOVF	TFLG2	Pulse accumulator overflow
PAIF	TFLG2	Pulse accumulator input

For each bit in TFLG1 and TFLG2 there is a corresponding bit in TMSK1 and TMSK2 in the same bit position. If a mask bit is set and an associated event occurs, a hardware interrupt request is generated.

In order to re-enable a status flag after an event occurs, the status flags must be cleared. Status registers are cleared in a particular sequence. The register must first be read for set flags, then zeros must be written to the flags that are to be cleared. If a new event occurs between the time that the register is read and the time that it is written, the associated flag is not cleared.

11.4.2 GPT Interrupts

The GPT has 11 internal sources that can cause it to request interrupt service (refer to **Table 11-2**). Setting bits in TMSK1 and TMSK2 enables specific interrupt sources. TMSK1 and TMSK2 are 8-bit registers that can be addressed individually or as one 16-bit register. The registers are initialized to zero at reset. For each bit in TMSK1 and TMSK2 there is a corresponding bit in TFLG1 and TFLG2 in the same bit position. TMSK2 also controls the operation of the timer prescaler. Refer to **11.7 Prescaler** for more information.

The value of the interrupt priority level (IPL[2:0]) field in the interrupt control register (ICR) determines the priority of GPT interrupt requests. IPL[2:0] values correspond to MCU interrupt request signals $\overline{\text{IRQ}}$[7:1]. $\overline{\text{IRQ7}}$ is the highest priority interrupt request signal; $\overline{\text{IRQ1}}$ is the lowest-priority signal. A value of %111 causes $\overline{\text{IRQ7}}$ to be asserted when a GPT interrupt request is made; lower field values cause corresponding lower-priority interrupt request signals to be asserted. Setting field value to %000 disables interrupts.

Freescale Semiconductor, Inc.

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

Table 11-2 GPT Interrupt Sources

Name	Source Number	Source	Vector Number
—	0000	Adjusted channel	IVBA : 0000
IC1	0001	Input capture 1	IVBA : 0001
IC2	0010	Input capture 2	IVBA : 0010
IC3	0011	Input capture 3	IVBA : 0011
OC1	0100	Output compare 1	IVBA : 0100
OC2	0101	Output compare 2	IVBA : 0101
OC3	0110	Output compare 3	IVBA : 0110
OC4	0111	Output compare 4	IVBA : 0111
IC4/OC5	1000	Input capture 4/output compare 5	IVBA : 1000
TO	1001	Timer overflow	IVBA : 1001
PAOV	1010	Pulse accumulator overflow	IVBA : 1010
PAI	1011	Pulse accumulator input	IVBA : 1011

The CPU16 recognizes only interrupt request signals of a priority greater than the condition code register interrupt priority (IP) mask value. When the CPU acknowledges an interrupt request, the priority of the acknowledged request is written to the IP mask and driven out on the IMB address lines.

When the IP mask value driven out on the address lines is the same as the IRL value, the GPT contends for arbitration priority. GPT arbitration priority is determined by the value of IARB[3:0] in GPTMCR. Each MCU module that can make interrupt requests must be assigned a non-zero IARB value to implement an arbitration scheme. Arbitration is performed by serial assertion of IARB[3:0] bit values.

When the GPT wins interrupt arbitration, it responds to the CPU interrupt acknowledge cycle by placing an interrupt vector number on the data bus. The vector number is used to calculate displacement into the CPU16 exception vector table. Vector numbers are formed by concatenating the value in ICR IVBA[3:0] with a 4-bit value supplied by the GPT when an interrupt request is made. Hardware prevents the vector number from changing while it is being driven out on the IMB. Vector number assignment is shown in **Table 11-2**.

At reset, IVBA[3:0] is initialized to $0. To enable interrupt-driven timer operation, the upper nibble of a user-defined vector number ($40 – $FF) must be written to IVBA, and interrupt handler routines must be located at the addresses pointed to by the corresponding vector.

NOTE

IVBA[3:0] must be written before GPT interrupts are enabled, or the GPT could supply a vector number ($00 to $0F) that corresponds to an assigned or reserved exception vector.

The internal GPT interrupt priority hierarchy is shown in **Table 11-2**. The lower the interrupt source number, the higher the priority. A single GPT interrupt source can be given priority over all other GPT interrupt sources by assigning the priority adjust field (IPA[3:0]) in the ICR a value equal to its source number.

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

Interrupt requests are asserted until associated status flags are cleared. Status flags must be cleared in a particular sequence. The status register must first be read for set flags, then zeros must be written to the flags that are to be cleared. If a new event occurs between the time that the register is read and the time that it is written, the associated flag is not cleared.

For more information on interrupts, refer to **5.8 Interrupts**. For more information on exceptions, refer to **4.13.4 Types of Exceptions**.

11.5 Pin Descriptions

The GPT uses 12 of the MCU pins. Each pin can perform more than one function. Descriptions of GPT pins divided into functional groups follow.

11.5.1 Input Capture Pins

Each input capture pin is associated with a single GPT input capture function. Each pin has hysteresis. Any pulse longer than two system clocks is guaranteed to be valid and any pulse shorter than one system clock is ignored. Each pin has an associated 16-bit capture register that holds the captured counter value. These pins can also be used for general-purpose I/O. Refer to **11.8.2 Input Capture Functions** for more information.

11.5.2 Input Capture/Output Compare Pin

The input capture/output compare pin can be configured for use by either an input capture or an output compare function. It has an associated 16-bit register that is used for holding either the input capture value or the output match value. When used for input capture the pin has the same hysteresis as other input capture pins. The pin can be used for general-purpose I/O. Refer to **11.8.2 Input Capture Functions** and **11.8.3 Output Compare Functions** for more information.

11.5.3 Output Compare Pins

Output compare pins are used for GPT output compare functions. Each pin has an associated 16-bit compare register and a 16-bit comparator. Pins OC2, OC3, and OC4 are associated with a specific output compare function. The OC1 function can affect the output of all compare pins. If the OC1 pin is not needed for an output compare function it can be used to output the clock selected for the timer counter register. Any of these pins can also be used for general-purpose I/O. Refer to **11.8.3 Output Compare Functions** for more information.

11.5.4 Pulse Accumulator Input Pin

The pulse accumulator input (PAI) pin connects a discrete signal to the pulse accumulator for timed or gated pulse accumulation. PAI has hysteresis. Any pulse longer than two system clocks is guaranteed to be valid and any pulse shorter than one system clock is ignored. It can be used as a general-purpose input pin. Refer to **11.10 Pulse Accumulator** for more information.

Freescale Semiconductor, Inc.

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

11.5.5 Pulse-Width Modulation

Pulse-width modulation (PWMA/B) pins carry pulse-width modulator outputs. The modulators can be programmed to generate a periodic waveform of variable frequency and duty cycle. PWMA can be used to output the clock selected as the input to the PWM counter. These pins can also be used for general-purpose output. Refer to **11.11 Pulse-Width Modulation Unit** for more information.

11.5.6 Auxiliary Timer Clock Input

The auxiliary timer clock input (PCLK) pin connects an external clock to the GPT. The external clock can be used as the clock source for the capture/compare unit or the PWM unit in place of one of the prescaler outputs. PCLK has hysteresis. Any pulse longer than two system clocks is guaranteed to be valid and any pulse shorter than one system clock is ignored. This pin can also be used as a general-purpose input pin. Refer to **11.7 Prescaler** for more information.

11.6 General-Purpose I/O

Any GPT pin can be used for general-purpose I/O when it is not used for another purpose. Capture/compare pins are bidirectional, others can be used only for output or input. I/O direction is controlled by a data direction bit in the port GP data direction register (DDRGP).

Parallel data is read from and written to the port GP data register (PORTGP). Pin data can be read even when pins are configured for a timer function. Data read from PORTGP always reflects the state of the external pin, while data written to PORTGP may not always affect the external pin.

Data written to PORTGP does not immediately affect pins used for output compare functions, but the data is latched. When an output compare function is disabled, the last data written to PORTGP is driven out on the associated pin if it is configured as an output. Data written to PORTGP can cause input captures if the corresponding pin is configured for input capture function.

The pulse accumulator input (PAI) and the external clock input (PCLK) pins provide general-purpose input. The state of these pins can be read by accessing the PAIS and PCLKS bits in the pulse accumulator control register (PACTL).

Pulse-width modulation A and B (PWMA/B) output pins can serve as general-purpose outputs. The force PWM value (FPWMx) and the force logic one (F1x) bits in the compare force (CFORC) and PWM control (PWMC) registers, respectively, control their operation.

11.7 Prescaler

Capture/compare and PWM units have independent 16-bit free-running counters as a main timing component. These counters derive their clocks from the prescaler or from the PCLK input. **Figure 11-2** is a prescaler block diagram.

Freescale Semiconductor, Inc.

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

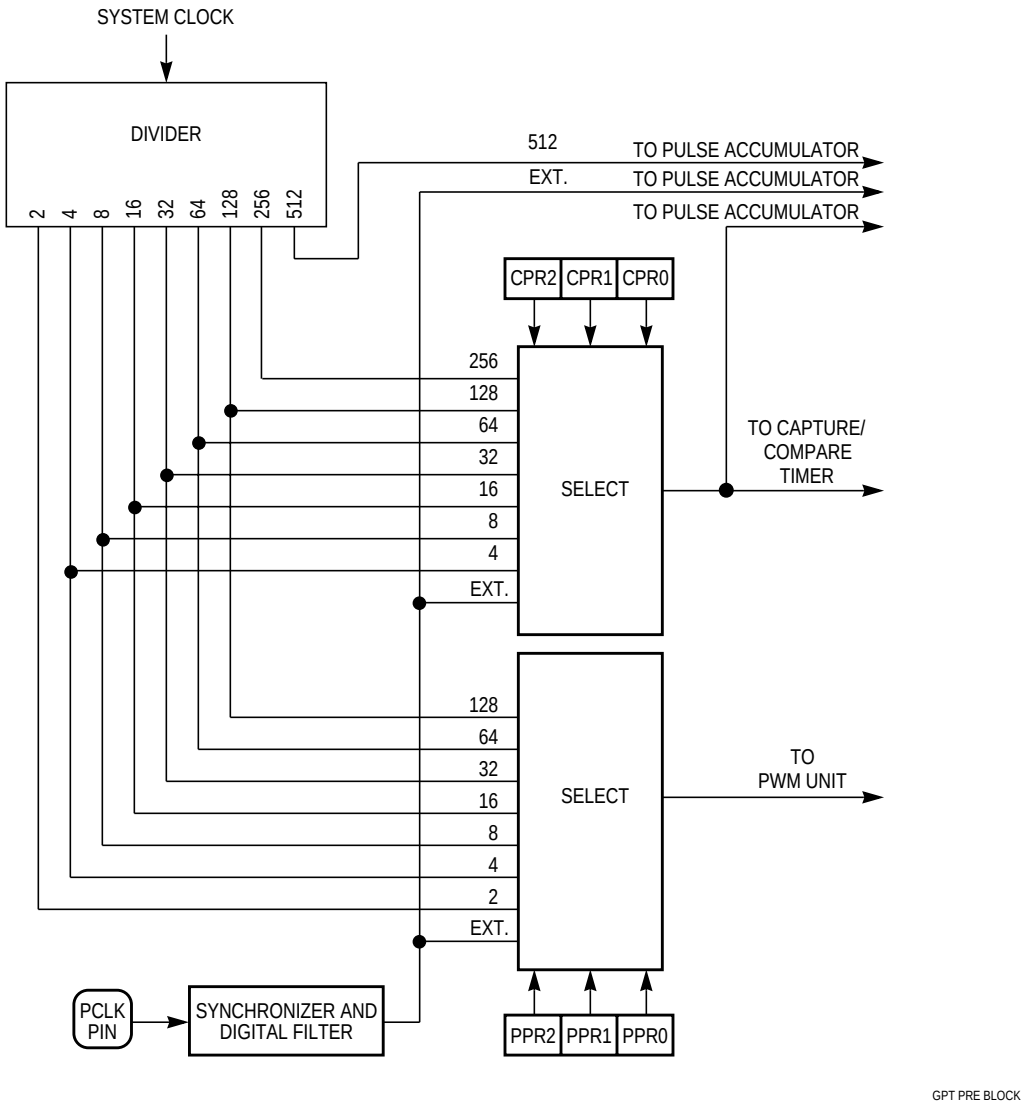


Figure 11-2 Prescaler Block Diagram

In the prescaler, the system clock is divided by a nine-stage divider chain. Prescaler outputs equal to system clock divided by 2, 4, 8, 16, 32, 64, 128, 256 and 512 are provided. Connected to these outputs are two multiplexers, one for the capture/compare unit, the other for the PWM unit.

Multiplexers can each select one of seven prescaler taps or an external input from the PCLK pin. Multiplexer output for the timer counter (TCNT) is selected by bits CPR[2:0] in timer interrupt mask register 2 (TMSK2). Multiplexer output for the PWM counter (PWMCNT) is selected by bits PPR[2:0] in PWM control register C (PWMC). After reset, the GPT is configured to use system clock divided by four for TCNT and system clock divided by two for PWMCNT. Initialization software can change the division factor. The PPR bits can be written at any time, but the CPR bits can only be written once after reset, unless the GPT is in test or freeze mode.

Freescale Semiconductor, Inc.

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

The prescaler can be read at any time. In freeze mode the prescaler can also be written. Word accesses must be used to ensure coherency. If coherency is not needed byte accesses can be used. The prescaler value is contained in bits [8:0] while bits [15:9] are unimplemented and are read as zeros.

Multiplexer outputs (including the PCLK signal) can be connected to external pins. The CPROUT bit in the TMSK2 register configures the OC1pin to output the TCNT clock and the PPROUT bit in the PWMC register configures the PWMA pin to output the PWMC clock. CPROUT and PPROUT can be written at any time. Clock signals on OC1 and PWMA do not have a 50% duty cycle. They have the period of the selected clock but are high for only one system clock time.

The prescaler also supplies three clock signals to the pulse accumulator clock select mux. These are the system clock divided by 512, the external clock signal from the PCLK pin and the capture/compare clock signal.

11.8 Capture/Compare Unit

The capture/compare unit contains the timer counter (TCNT), the input capture (IC) functions and the output compare (OC) functions. **Figure 11-3** is a block diagram of the capture/compare unit.

11.8.1 Timer Counter

The timer counter (TCNT) is the key timing component in the capture/compare unit. The timer counter is a 16-bit free-running counter that starts counting after the processor comes out of reset. The counter cannot be stopped during normal operation. After reset, the GPT is configured to use the system clock divided by four as the input to the counter. The prescaler divides the system clock and provides selectable input frequencies. User software can configure the system to use one of seven prescaler outputs or an external clock.

The counter can be read any time without affecting its value. Because the GPT is interfaced to the IMB, and the IMB supports a 16-bit bus, a word read gives a coherent value. If coherency is not needed, byte accesses can be made. The counter is set to $0000 during reset and is normally a read-only register. In test mode and freeze mode, any value can be written to the timer counter.

When the counter rolls over from $FFFF to $0000, the timer overflow flag (TOF) in timer interrupt flag register 2 (TFLG2) is set. An interrupt can be enabled by setting the corresponding interrupt enable bit (TOI) in timer interrupt mask register 2 (TMSK2). Refer to **11.4.2 GPT Interrupts** for more information.

11.8.2 Input Capture Functions

All GPT input capture functions use the same 16-bit timer counter (TCNT). Each input capture pin has a dedicated 16-bit latch and input edge-detection/selection logic. Each input capture function has an associated status flag, and can cause the GPT to make an interrupt service request.

When a selected edge transition occurs on an input capture pin, the associated 16-bit latch captures the content of TCNT and sets the appropriate status flag. An interrupt request can be generated when the transition is detected.

Freescale Semiconductor, Inc.

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

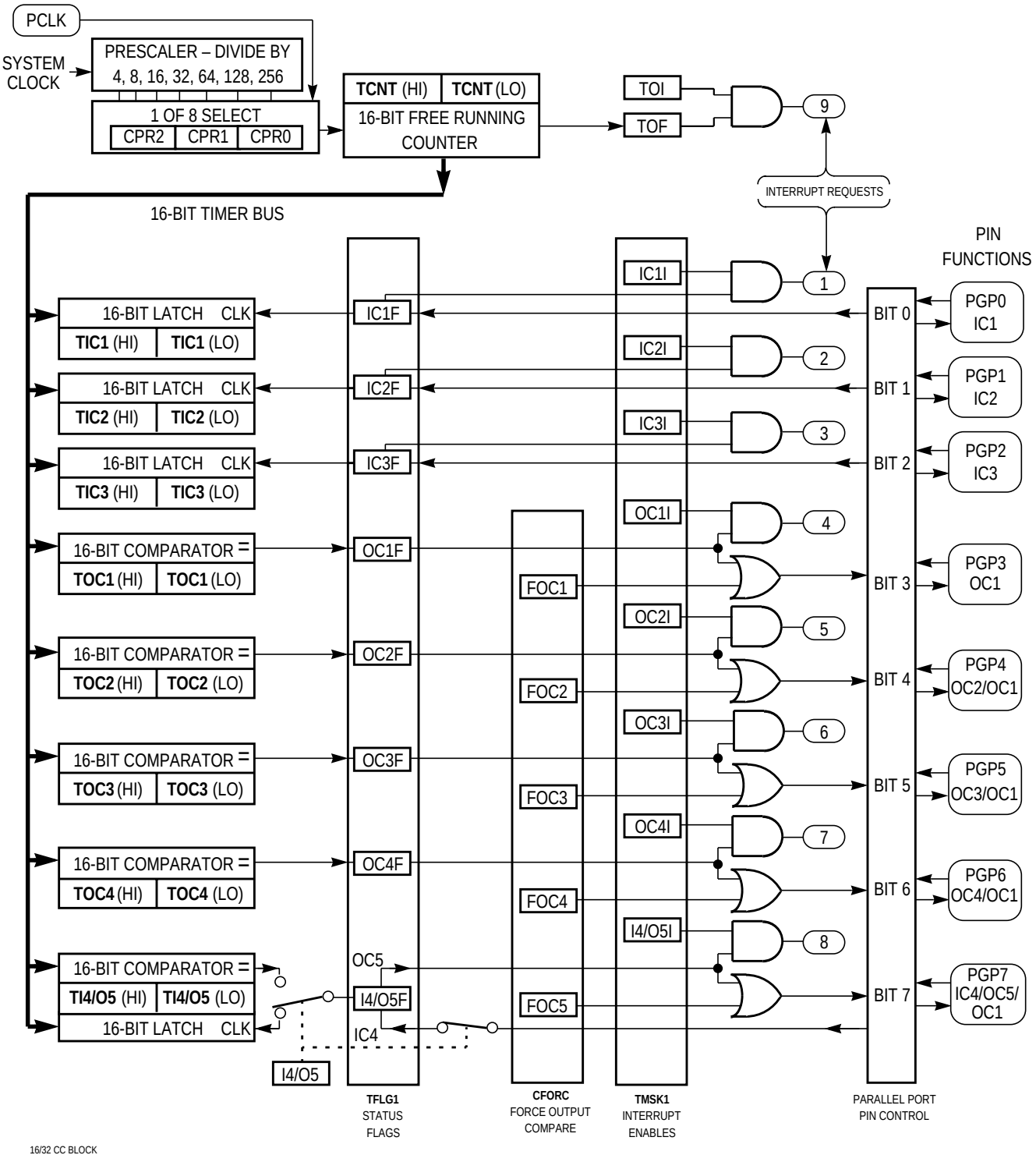


Figure 11-3 Capture/Compare Unit Block Diagram

Freescale Semiconductor, Inc.

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

Edge-detection logic consists of control bits that enable edge detection and select a transition to detect. The EDGExA/B bits in timer control register 2 (TCTL2) determine whether the input capture functions detect rising edges only, falling edges only, or both rising and falling edges. Clearing both bits disables the input capture function. Input capture functions operate independently of each other and can capture the same TCNT value if individual input edges are detected within the same timer count cycle.

Input capture interrupt logic includes a status flag, which indicates that an edge has been detected, and an interrupt enable bit. An input capture event sets the ICxF bit in the timer interrupt flag register 1 (TFLG1) and causes the GPT to make an interrupt request if the corresponding ICxI bit is set in the timer interrupt mask register 1 (TMSK1). If the ICxI bit is cleared, software must poll the status flag to determine that an event has occurred. Refer to **11.4 Polled and Interrupt-Driven Operation** for more information.

Input capture events are generally asynchronous to the timer counter. Because of this, input capture signals are conditioned by a synchronizer and digital filter. Events are synchronized with the system clock and digital filter. Events are synchronized with the system clock so that latching of TCNT content and counter incrementation occur on opposite half-cycles of the system clock. Inputs have hysteresis. Capture of any transition longer than two system clocks is guaranteed; any transition shorter than one system clock has no effect.

Figure 11-4 shows the relationship of system clock to synchronizer output. The value latched into the capture register is the value of the counter several system clock cycles after the transition that triggers the edge detection logic. There can be up to one clock cycle of uncertainty in latching of the input transition. Maximum time is determined by the system clock frequency.

The input capture register is a 16-bit register. A word access is required to ensure coherency. If coherency is not required, byte accesses can be used to read the register. Input capture registers can be read at any time without affecting their values.

Freescale Semiconductor, Inc.

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

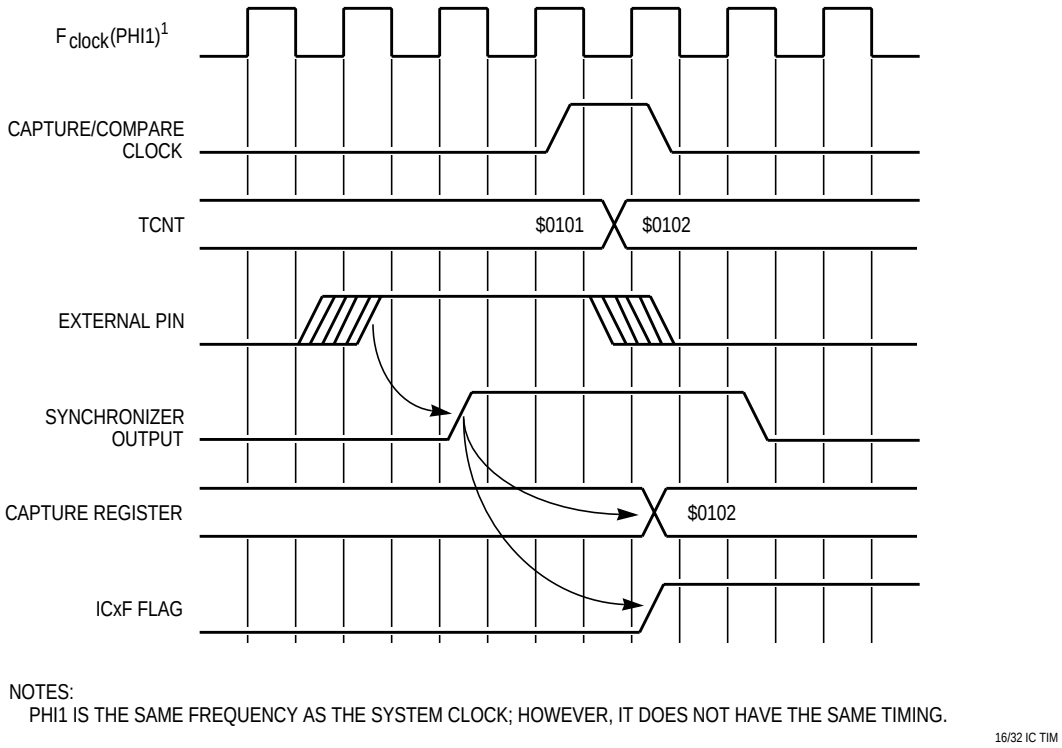


Figure 11-4 Input Capture Timing Example

An input capture occurs every time a selected edge is detected, even when the input capture status flag is set. This means that the value read from the input capture register corresponds to the most recent edge detected, which may not be the edge that caused the status flag to be set.

11.8.3 Output Compare Functions

Each GPT output compare pin has an associated 16-bit compare register and a 16-bit comparator. Each output compare function has an associated status flag, and can cause the GPT to make an interrupt service request. Output compare logic is designed to prevent false compares during data transition times.

When the programmed content of an output compare register matches the value in TCNT, an output compare status flag (OCxF) bit in TFLG1 is set. If the appropriate interrupt enable bit (OCxI) in TMSK1 is set, an interrupt request is made when a match occurs. Refer to **11.4.2 GPT Interrupts** for more information.

Operation of output compare 1 differs from that of the other output compare functions. OC1 control logic can be programmed to make state changes on other OC pins when an OC1 match occurs. Control bits in the timer compare force register (CFORC) allow for early forced compares.

Freescale Semiconductor, Inc.

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

11.8.3.1 Output Compare 1

Output compare 1 can affect any or all of OC[5:1] when an output match occurs. In addition to allowing generation of multiple control signals from a single comparison operation, this function makes it possible for two or more output compare functions to control the state of a single OC pin. Output pulses as short as one timer count can be generated in this way.

The OC1 action mask register (OC1M) and the OC1 action data register (OC1D) control OC1 function. Setting a bit in OC1M selects a corresponding bit in the GPT parallel data port. Bits in OC1D determine whether selected bits are to be set or cleared when an OC1 match occurs. Pins must be configured as outputs in order for the data in the register to be driven out on the corresponding pin. If an OC1 match and another output match occur at the same time and both attempt to alter the same pin, the OC1 function controls the state of the pin.

11.8.3.2 Forced Output Compare

Timer compare force register (CFORC) is used to make forced compares. The action taken as a result of a forced compare is the same as when an output compare match occurs, except that status flags are not set. Forced channels take programmed actions immediately after the write to CFORC.

The CFORC register is implemented as the upper byte of a 16-bit register which also contains the PWM control register C (PWMC). It can be accessed as eight bits or a word access can be used. Reads of force compare bits (FOC) have no meaning and always return zeros. These bits are self-negating.

11.9 Input Capture 4/Output Compare 5

The IC4/OC5 pin can be used for input capture, output compare, or general-purpose I/O. A function enable bit (I4/O5) in the pulse accumulator control register (PACTL) configures the pin for input capture (IC4) or output compare function (OC5). Both bits are cleared during reset, configuring the pin as an input, but also enabling the OC5 function. IC4/OC5 I/O functions are controlled by DDGP7 in the port GP data direction register (DDRGP).

The 16-bit register (TI4/O5) used with the IC4/OC5 function acts as an input capture register or as an output compare register depending on which function is selected. When used as the input capture 4 register, it cannot be written to except in test or freeze mode.

11.10 Pulse Accumulator

The pulse accumulator counter (PACNT) is an 8-bit read/write up-counter. PACNT can operate in external event counting or gated time accumulation modes. **Figure 11-5** is a block diagram of the pulse accumulator.

Freescale Semiconductor, Inc.

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

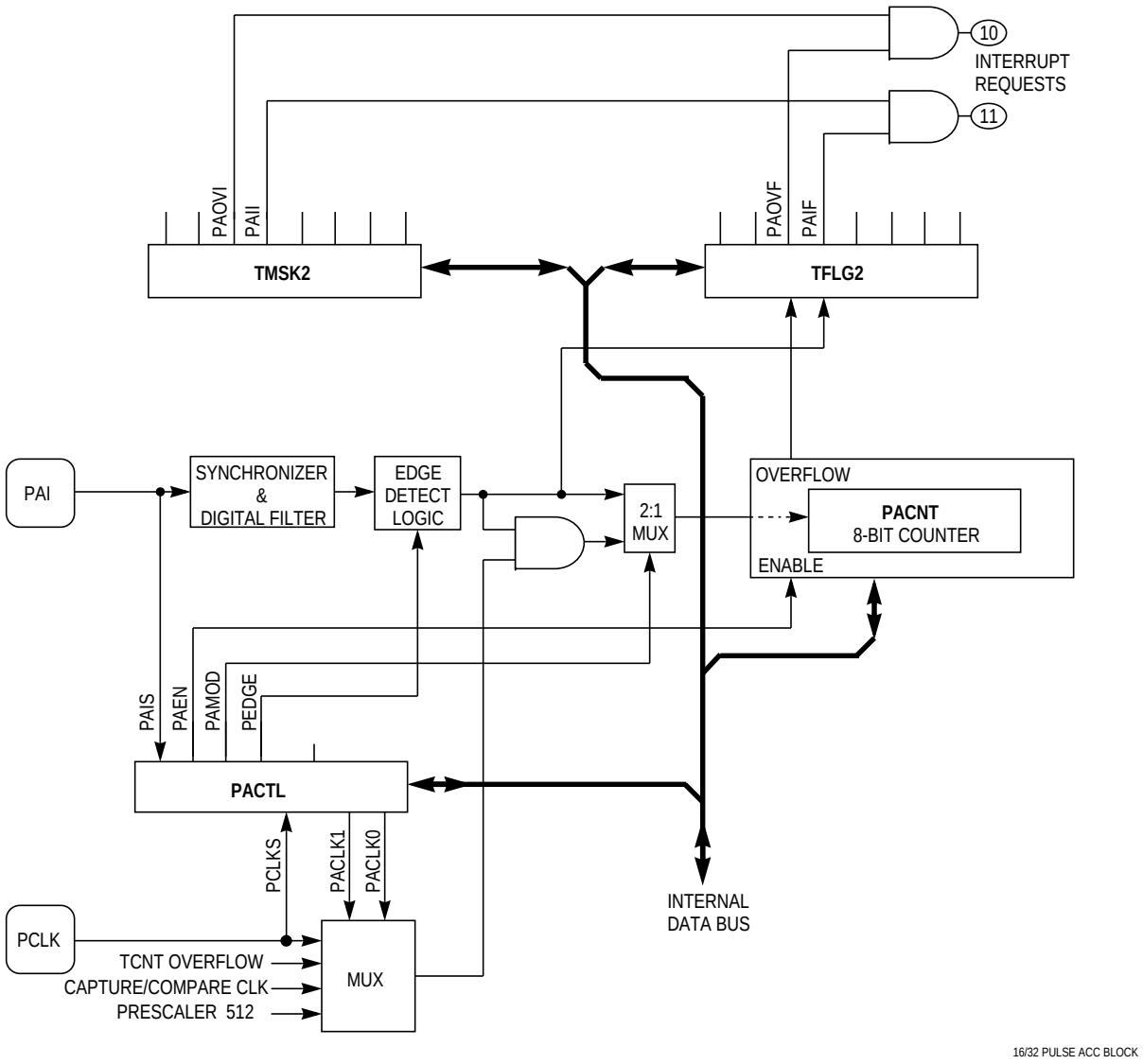


Figure 11-5 Pulse Accumulator Block Diagram

In event counting mode, the counter increments each time a selected transition of the pulse accumulator input (PAI) pin is detected. The maximum clocking rate is the system clock divided by four.

In gated time accumulation mode a clock increments PACNT while the PAI pin is in the active state. There are four possible clock sources.

Two bits in the TFLG2 register show pulse accumulator status. The pulse accumulator flag (PAIF) indicates that a selected edge has been detected at the PAI pin. The pulse accumulator overflow flag (PAOVF) indicates that the pulse accumulator count has rolled over from $FF to $00. This can be used to extend the range of the counter beyond eight bits.

Freescale Semiconductor, Inc.

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

An interrupt request can be made when each of the status flags is set. However, operation of the PAI interrupt depends on operating mode. In event counting mode, an interrupt is requested when the edge being counted is detected. In gated mode, the request is made when the PAI input changes from active to inactive state. Interrupt requests are enabled by the PAOVI and PAII bits in the TMSK2 register.

Bits in the pulse accumulator control register (PACTL) control the operation of PACNT. The PAMOD bit selects event counting or gated operation. In event counting mode, the PEDGE control bit determines whether a rising or falling edge is detected. In gated mode, PEDGE specifies the active state of the gate signal. Bits PACLK[1:0] select the clock source used in gated mode.

PACTL and PACNT are implemented as one 16-bit register, but can be accessed with byte or word access cycles. Both registers are cleared at reset, but the PAIS and PCLKS bits show the state of the PAI and PCLK pins.

The PAI pin can also be used for general-purpose input. The logic state of the PAIS bit in PACTL shows the state of the pin.

11.11 Pulse-Width Modulation Unit

The pulse-width modulation (PWM) unit has two output channels, PWMA and PWMB. A single clock output from the prescaler multiplexer drives a 16-bit counter that is used to control both channels. **Figure 11-6** is a block diagram of the pulse-width modulation unit.

Freescale Semiconductor, Inc.

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

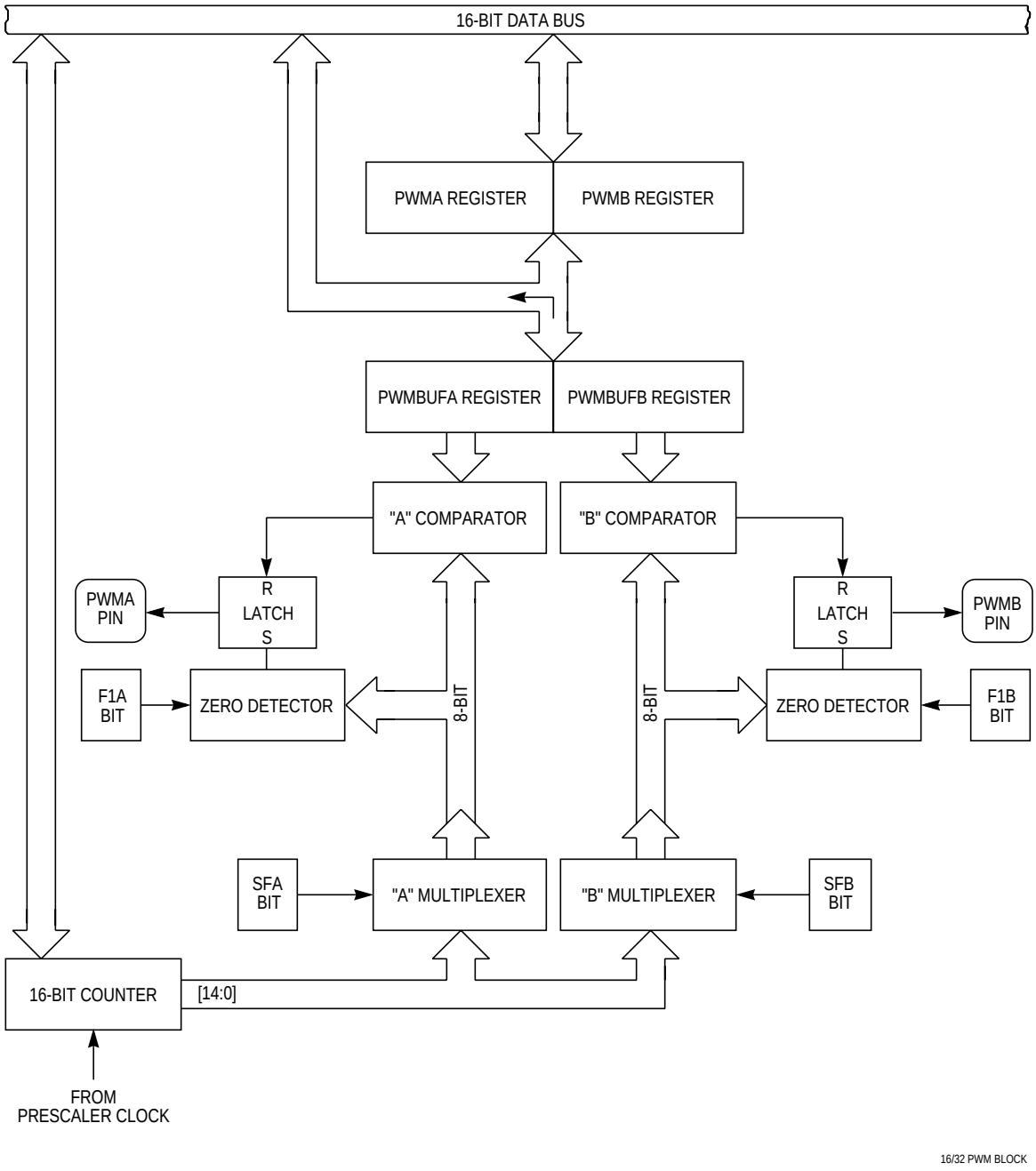


Figure 11-6 PWM Block Diagram

The PWM unit has two operational modes. Fast mode uses a clocking rate that equals 1/256 of the prescaler output rate; slow mode uses a rate equal to 1/32768 of the prescaler output rate. The duty cycle ratios of the two PWM channels can be individually controlled by software. The PWMA pin can also output the clock that drives the PWM counter. PWM pins can also be used as output pins.

Freescale Semiconductor, Inc.

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

11.11.1 PWM Counter

The 16-bit counter in the PWM unit is similar to the timer counter in the capture/compare unit. During reset, the GPT is configured to use the system clock divided by two to drive the counter. Initialization software can reconfigure the counter to use one of seven prescaler outputs or an external clock input from the PCLK pin.

The PWM count register (PWMCNT) can be read at any time without affecting its value. A read must be a word access to ensure coherence, but byte accesses can be made if coherence is not needed. The counter is cleared to $0000 during reset and is a read-only register except in freeze or test mode.

Fifteen of the sixteen counter bits are output to multiplexers A and B. The multiplexers provide the fast and slow modes of the PWM unit. Mode for PWMA is selected by the SFA bit in the PWM control register C (PWMC). Mode for PWMB is selected by the SFB bit in the same register.

PWMA, PWMB, and PPR[2:0] bits in PWMC control PWM output frequency. In fast mode, bits [7:0] of PWMCNT are used to clock the PWM logic; in slow mode, bits [14:7] are used. The period of a PWM output in slow mode is 128 times longer than the fast mode period. **Table 11-3** shows a range of PWM output frequencies using 16.78 MHz, 20.97 MHz, and 25.17 MHz system clocks.

Table 11-3 PWM Frequency Ranges

PPR [2:0]	Prescaler Tap			SFA/B = 0			SFA/B = 1		
	16.78 MHz	20.97 MHz	25.17 MHz	16.78 MHz	20.97 MHz	25.17 MHz	16.78 MHz	20.97 MHz	25.17 MHz
000	Div 2 = 8.39 MHz	Div 2 = 10.5 MHz	Div 2 = 12.6 MHz	32.8 kHz	41 kHz	49.2 kHz	256 Hz	320 Hz	384 Hz
001	Div 4 = 4.19 MHz	Div 4 = 5.25 MHz	Div 4 = 6.29 MHz	16.4 kHz	20.5 kHz	24.6 kHz	128 Hz	160 Hz	192 Hz
010	Div 8 = 2.10 MHz	Div 8 = 2.62 MHz	Div 8 = 3.15 MHz	8.19 kHz	10.2 kHz	12.3 kHz	64.0 Hz	80.0 Hz	96 Hz
011	Div 16 = 1.05 MHz	Div 16 = 1.31 MHz	Div 16 = 1.57 MHz	4.09 kHz	5.15 kHz	6.13 kHz	32.0 Hz	40.0 Hz	48 Hz
100	Div 32 = 524 kHz	Div 32 = 655 kHz	Div 32 = 787 kHz	2.05 kHz	2.56 kHz	3.07 kHz	16.0 Hz	20.0 Hz	24 Hz
101	Div 64 = 262 kHz	Div 64 = 328 kHz	Div 64 = 393 kHz	1.02 kHz	1.28 kHz	1.54 kHz	8.0 Hz	10.0 Hz	12 Hz
110	Div 128 = 131 kHz	Div 128 = 164 kHz	Div 128 = 197 kHz	512 Hz	641 Hz	770 Hz	4.0 Hz	5.0 Hz	6 Hz
111	PCLK	PCLK	PCLK	PCLK/256	PCLK/256	PCLK/256	PCLK/ 32768	PCLK/ 32768	PCLK/ 32768

11.11.2 PWM Function

The pulse width values of the PWM outputs are determined by control registers PWMA and PWMB. PWMA and PWMB are 8-bit registers implemented as two bytes of a 16-bit register. PWMA and PWMB can be accessed as separate bytes or as one 16-bit register. A value of $00 loaded into either register causes the corresponding output pin to output a continuous logic level zero signal. A value of $80 causes the corresponding output signal to have a 50% duty cycle, and so on, to the maximum value of $FF, which corresponds to an output which is at logic level one for 255/256 of the cycle.

Setting the F1A (for PWMA) or F1B (for PWMB) bits in the CFORC register causes the corresponding pin to output a continuous logic level one signal. The logic level of the associated pin does not change until the end of the current cycle. F1A and F1B are the lower two bits of CFORC, but can be accessed at the same word address as PWMC.

Freescale Semiconductor, Inc.

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

Data written to PWMA and PWMB is not used until the end of a complete cycle. This prevents spurious short or long pulses when register values are changed. The current duty cycle value is stored in the appropriate PWM buffer register (PWMBUFA or PWMBUFB). The new value is transferred from the PWM register to the buffer register at the end of the current cycle.

Registers PWMA, PWMB, and PWMC are reset to $00 during reset. These registers may be written or read at any time. PWMC is implemented as the lower byte of a 16-bit register. The upper byte is the CFORC register. The buffer registers, PWMBUFA and PWMBUFB, are read-only at all times and may be accessed as separate bytes or as one 16-bit register.

Pins PWMA and PWMB can also be used for general-purpose output. The values of the F1A and F1B bits in PWMC are driven out on the corresponding PWM pins when normal PWM operation is disabled. When read, the F1A and F1B bits reflect the states of the PWMA and PWMB pins.

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

APPENDIX A
ELECTRICAL CHARACTERISTICS

Table A-1 Maximum Ratings

Num	Rating	Symbol	Value	Unit
1	Supply Voltage[1, 2, 3]	V_{DD}	– 0.3 to +6.5	V
2	Input Voltage [1, 2, 3, 4, 5, 7]	V_{IN}	– 0.3 to +6.5	V
3	Instantaneous Maximum Current Single Pin Limit (applies to all pins)[1, 3, 5, 6]	I_D	25	mA
4	Operating Maximum Current Digital Input Disruptive Current[3, 5, 6, 7, 8] $V_{NEGCLAMP} \approx -0.3$ V $V_{POSCLAMP} \approx V_{DD} + 0.3$ V	Ii_D	– 500 to +500	µA
5	Operating Temperature Range “C” Suffix “V” Suffix “M” Suffix	T_A	T_L to T_H – 40 to +85 – 40 to +105 – 40 to +125	°C
6	Storage Temperature Range	T_{stg}	– 55 to +150	°C

NOTES:

1. Permanent damage can occur if maximum ratings are exceeded. Exposure to voltages or currents in excess of recommended values affects device reliability. Device modules may not operate normally while being exposed to electrical extremes.
2. Although sections of the device contain circuitry to protect against damage from high static voltages or electrical fields, take normal precautions to avoid exposure to voltages higher than maximum-rated voltages.
3. This parameter is periodically sampled rather than 100% tested.
4. All pins except TSC.
5. Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values for positive and negative clamp voltages, then use the larger of the two values.
6. Power supply must maintain regulation within operating V_{DD} range during instantaneous and operating maximum current.
7. All functional non-supply pins are internally clamped to V_{SS}. All functional pins except EXTAL, TSC, and XFC are internally clamped to V_{DD}.
8. Total input current for all digital input-only and all digital input/output pins must not exceed 10 mA. Exceeding this limit can cause disruption of normal operation.

Freescale Semiconductor, Inc.

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

Table A-2 Typical Ratings, 2.7 to 3.6V, 16.78-MHz Operation

Num	Rating	Symbol	Value	Unit
1	Supply Voltage	V_{DD}	3.0	V
2	Operating Temperature	T_A	25	°C
3	V_{DD} Supply Current RUN LPSTOP, VCO off LPSTOP, External clock, max f_{sys}	I_{DD}	 38 70 1	 mA µA mA
4	Clock Synthesizer Operating Voltage	V_{DDSYN}	3.0	V
5	V_{DDSYN} Supply Current 4.194 MHz VCO on, maximum f_{sys} 32.768 kHz VCO on, maximum f_{sys} 4.194 MHz External Clock, maximum f_{sys} 32.768 kHz External Clock, maximum f_{sys} 4.194 MHz LPSTOP, VCO off 32.768 kHz LPSTOP, VCO off 4.194 MHz V_{DD} powered down 32.768 kHz V_{DD} powered down	I_{DDSYN}	 TBD 200 TBD 1 TBD 20 TBD 10	 mA µA µA mA mA µA µA µA
6	RAM Standby Voltage	V_{SB}	3	V
7	RAM Standby Current Normal RAM operation Standby operation	I_{SB}	 3 3	 µA µA
8	Power Dissipation	P_D	120	mW

Freescale Semiconductor, Inc.

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

Table A-3 Typical Ratings, 5V, 16.78-MHz Operation

Num	Rating	Symbol	Value	Unit
1	Supply Voltage	V_{DD}	5.0	V
2	Operating Temperature	T_A	25	°C
3	V_{DD} Supply Current RUN LPSTOP, VCO off LPSTOP, External clock, maximum f_{sys}	I_{DD}	 65 125 3	 mA µA mA
4	Clock Synthesizer Operating Voltage	V_{DDSYN}	5.0	V
5	V_{DDSYN} Supply Current VCO on, maximum f_{sys} External Clock, maximum f_{sys} LPSTOP, VCO off V_{DD} powered down	I_{DDSYN}	 1.0 3.0 100 50	 mA mA µA µA
6	RAM Standby Voltage	V_{SB}	5.0	V
7	RAM Standby Current Normal RAM operation Standby operation	I_{SB}	 1.0 1.0	 µA µA
8	Power Dissipation	P_D	380	mW

Table A-4 Typical Ratings, 20.97-MHz Operation

Num	Rating	Symbol	Value	Unit
1	Supply Voltage	V_{DD}	5.0	V
2	Operating Temperature	T_A	25	°C
3	V_{DD} Supply Current RUN LPSTOP, VCO off LPSTOP, External clock, maximum f_{sys}	I_{DD}	 95 125 3.75	 mA µA mA
4	Clock Synthesizer Operating Voltage	V_{DDSYN}	5.0	V
5	V_{DDSYN} Supply Current VCO on, maximum f_{sys} External Clock, maximum f_{sys} LPSTOP, VCO off V_{DD} powered down	I_{DDSYN}	 1.0 4.0 100 50	 mA mA µA µA
6	RAM Standby Voltage	V_{SB}	5.0	V
7	RAM Standby Current Normal RAM operation Standby operation	I_{SB}	 1.0 1.0	 µA µA
8	Power Dissipation	P_D	480	mW

Freescale Semiconductor, Inc.

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

Table A-5 Typical Ratings, 25.17-MHz

Num	Rating	Symbol	Value	Unit
1	Supply Voltage	V_{DD}	5.0	V
2	Operating Temperature	T_A	25	°C
3	V_{DD} Supply Current RUN LPSTOP, VCO off LPSTOP, External clock, max f_{sys}	I_{DD}	 110 125 3.75	 mA µA mA
4	Clock Synthesizer Operating Voltage	V_{DDSYN}	5.0	V
5	V_{DDSYN} Supply Current VCO on, maximum f_{sys} External Clock, maximum f_{sys} LPSTOP, VCO off V_{DD} powered down	I_{DDSYN}	 1.0 5.0 100 50	 mA mA µA µA
6	RAM Standby Voltage	V_{SB}	5.0	V
7	RAM Standby Current Normal RAM operation Standby operation	I_{SB}	 1.0 1.0	 µA µA
8	Power Dissipation	P_D	555	mW

Freescale Semiconductor, Inc.

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

Table A-6 Thermal Characteristics

Num	Characteristic	Symbol	Value	Unit
1	Thermal Resistance Plastic 132-Pin Surface Mount Plastic 144-Pin Surface Mount	Θ_{JA}	 38 49	°C/W

The average chip-junction temperature (T_J) in C can be obtained from:

$$T_J = T_A + (P_D \times \Theta_{JA}) \quad (1)$$

where:

T_A= Ambient Temperature, °C

Θ_{JA}= Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{DD} \times V_{DD}$, Watts — Chip Internal Power

$P_{I/O}$= Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected. An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \times (T_A + 273°C) + \Theta_{JA} \times P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

Freescale Semiconductor, Inc.

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

Table A-7 Low Voltage Clock Control Timing

(V_{DD} and V_{DDSYN} = 2.7 to 3.6 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
1	PLL Reference Frequency Range[1] MC68CM16Z1	f_{ref}	 3.2	 4.2	 MHz
2	PLL Reference Frequency Range[1] MC68CK16Z1 MC68CK16Z4	f_{ref}	 20 20	 50 50	 kHz kHz
3	System Frequency[2] On-Chip PLL System Frequency Slow On-Chip PLL System Frequency Fast On-Chip PLL System Frequency External Clock Operation	f_{sys}	dc 4 (f_{ref}) 4 (f_{ref}) /128 dc	16.78 16.78 16.78 16.78	MHz
4	PLL Lock Time[1, 7, 8, 9] Changing W or Y in SYNCR or exiting from LPSTOP[3] Warm Start-Up[4] Cold Start-Up (fast reference option only)[5]	t_{lpll}	—	 20 50 75	ms
5	VCO Frequency[6]	f_{VCO}	—	2 (f_{sys} max)	ms
6	Limp Mode Clock Frequency SYNCR X bit = 0 SYNCR X bit = 1	f_{limp}	 — —	 f_{sys} max /2 f_{sys} max	 MHz
7	CLKOUT Jitter[1, 7, 8, 9, 10] Short term (5 µs interval) Long term (500 µs interval)	J_{clk}	 –0.5 –0.05	 0.5 0.05	 %

NOTES:
1. Refer to notes in **Table A-10**.

Freescale Semiconductor, Inc.

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

Table A-8 16.78-MHz Clock Control Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Minimum	Maximum	Unit
1	PLL Reference Frequency Range[1] MC68HC16Z1 MC68HC16Z2 MC68HC16Z3	f_{ref}	 25 3.2 3.2	 50 4.2 4.2	 kHz MHz MHz
2	System Frequency[2] On-Chip PLL System Frequency Slow On-Chip PLL System Frequency Fast On-Chip PLL System Frequency External Clock Operation	f_{sys}	dc 4 (f_{ref}) 4 (f_{ref}) /128 dc	16.78 16.78 16.78 16.78	MHz
3	PLL Lock Time[1, 7, 8, 9] Changing W or Y in SYNCR or exiting from LPSTOP[3] Warm Start-Up[4] Cold Start-Up (fast reference option only)[5]	t_{lpll}	—	 20 50 75	ms
4	VCO Frequency[6]	f_{VCO}	—	2 (f_{sys} max)	ms
5	Limp Mode Clock Frequency SYNCR X bit = 0 SYNCR X bit = 1	f_{limp}	 — —	 f_{sys} max /2 f_{sys} max	 MHz
6	CLKOUT Jitter[1, 7, 8, 9, 10] Short term (5 µs interval) Long term (500 µs interval)	J_{clk}	 –0.5 –0.05	 0.5 0.05	 %

NOTES:
1. Refer to notes in **Table A-10**.

Freescale Semiconductor, Inc.

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

Table A-9 20.97-MHz Clock Control Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Minimum	Maximum	Unit
1	PLL Reference Frequency Range[1] MC68HC16Z1 MC68HC16Z2 MC68HC16Z3	f_{ref}	 20 3.2 3.2	 50 5.2 5.2	 kHz MHz MHz
2	System Frequency[2] On-Chip PLL System Frequency Slow On-Chip PLL System Frequency Fast On-Chip PLL System Frequency External Clock Operation	f_{sys}	dc 4 (f_{ref}) 4 (f_{ref}) /128 dc	20.97 20.97 20.97 20.97	MHz
3	PLL Lock Time[1, 7, 8, 9] Changing W or Y in SYNCR or exiting from LPSTOP[3] Warm Start-Up[4] Cold Start-Up (fast reference option only)[5]	t_{lpll}	—	 20 50 75	ms
4	VCO Frequency[6]	f_{VCO}	—	2 (f_{sys} max)	MHz
5	Limp Mode Clock Frequency SYNCR X bit = 0 SYNCR X bit = 1	f_{limp}	 — —	 f_{sys} max /2 f_{sys} max	MHz
6	CLKOUT Jitter[1, 7, 8, 9, 10] Short term (5 µs interval) Long term (500 µs interval)	J_{clk}	 –1.0 –0.5	 1.0 0.5	%

NOTES:
1. Refer to notes in **Table A-10**

Freescale Semiconductor, Inc.

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

Table A-10 25.17-MHz Clock Control Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
1	PLL Reference Frequency Range[1] MC68HC16Z1 MC68HC16Z2 MC68HC16Z3	f_{ref}	 20 3.2 3.2	 50 5.2 5.2	 kHz MHz MHz
2	System Frequency[2] On-Chip PLL System Frequency Slow On-Chip PLL System Frequency Fast On-Chip PLL System Frequency External Clock Operation	f_{sys}	dc 4 (f_{ref}) 4 (f_{ref}) /128 dc	25.17 25.17 25.17 25.17	 MHz
3	PLL Lock Time[1, 7, 8, 9] Changing W or Y in SYNCR or exiting from LPSTOP[3] Warm Start-Up[4] Cold Start-Up (fast reference option only)[5]	t_{lpll}	 — — —	 20 50 75	 ms
4	VCO Frequency[6]	f_{VCO}	—	2 (f_{sys} max)	MHz
5	Limp Mode Clock Frequency SYNCR X bit = 0 SYNCR X bit = 1	f_{limp}	 — —	 f_{sys} max/2 f_{sys} max	 MHz
6	CLKOUT Jitter[1, 7, 8, 9, 10] Short term (5 µs interval) Long term (500 µs interval)	J_{clk}	 –1.0 –0.05	 1.0 0.5	 %

NOTES:

1. The base configuration of the MC68HC16Z1, MC68CK16Z1, MC68HC16Z4, and the MC68CK16Z4 requires a 32.768 kHz crystal reference. The base configuration of the MC68CM16Z1, M68HC16Z2, and the MC68HC16Z3 requires a 4.194 MHz crystal reference.
2. All internal registers retain data at 0 Hz.
3. Assumes that V_{DDSYN} and V_{DD} are stable, that an external filter is attached to the XFC pin, and that the crystal oscillator is stable.
4. Assumes that V_{DDSYN} is stable, that an external filter is attached to the XFC pin, and that the crystal oscillator is stable, followed by V_{DD} ramp-up. Lock time is measured from V_{DD} at specified minimum to $\overline{\text{RESET}}$ negated.
5. Cold start is measured from V_{DDSYN} and V_{DD} at specified minimum to $\overline{\text{RESET}}$ negated.
6. Internal VCO frequency (f_{VCO}) is determined by SYNCR W and Y bit values. The SYNCR X bit controls a divide-by-two circuit that is not in the synthesizer feedback loop.
   When X = 0, the divider is enabled, and $f_{sys} = f_{VCO} \div 4$.
   When X = 1, the divider is disabled, and $f_{sys} = f_{VCO} \div 2$.
   X must equal one when operating at maximum specified f_{sys}.
7. This parameter is periodically sampled rather than 100% tested.
8. Assumes that a low-leakage external filter network is used to condition clock synthesizer input voltage. Total external resistance from the XFC pin due to external leakage must be greater than 15 MΩ to guarantee this specification. Filter network geometry can vary depending upon operating environment.
9. Proper layout procedures must be followed to achieve specifications.
10. Jitter is the average deviation from the programmed frequency measured over the specified interval at maximum f_{sys}. Measurements are made with the device powered by filtered supplies and clocked by a stable external clock signal. Noise injected into the PLL circuitry via V_{DDSYN} and V_{SS} and variation in crystal oscillator frequency increase the J_{clk} percentage for a given interval. When jitter is a critical constraint on control system operation, this parameter should be measured during functional testing of the final system.

Freescale Semiconductor, Inc.

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

Table A-11 Low Voltage 16.78-MHz DC Characteristics

(V_{DD} and V_{DDSYN} = 2.7 to 3.6Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
1	Input High Voltage	V_{IH}	0.7 (V_{DD})	V_{DD} + 0.3	V
2	Input Low Voltage	V_{IL}	V_{SS} – 0.3	0.2 (V_{DD})	V
3	Input Hysteresis[1]	V_{HYS}	0.5	—	V
4	Input Leakage Current [2] V_{in} = V_{DD} or V_{SS} Input-only pins	I_{in}	–2.5	2.5	µA
5	High Impedance (Off-State) Leakage Current[2] V_{in} = V_{DD} or V_{SS} All input/output and output pins	I_{OZ}	–2.5	2.5	µA
6	CMOS Output High Voltage[2, 3] I_{OH} = –10.0 µA Group 1, 2, 4 input/output and output pins	V_{OH}	V_{DD} –0.2	—	V
7	CMOS Output Low Voltage[2] I_{OL} = 10.0 µA Group 1, 2, 4 input/output and output pins	V_{OL}	—	0.2	V
8	Output High Voltage[2, 3] I_{OH} = –0.4 mA Group 1, 2, 4 input/output and output pins	V_{OH}	V_{DD} –0.5	—	V
9	Output Low Voltage[2] I_{OL} = 0.8 mA Group 1 I/O pins, CLKOUT, FREEZE/QUOT, IPIPE0 I_{OL} = 2.6 mA Group 2 and group 4 I/O pins, $\overline{CSBOOT}$, $\overline{BG}/\overline{CS}$ I_{OL} = 6 mA Group 3	V_{OL}	— — —	0.4 0.4 0.4	V
10	Three State Control Input High Voltage	V_{IHTSC}	7.2	9.1	V
11	Data Bus Mode Select Pull-up Current [4] V_{in} = V_{IL} V_{in} = V_{IH}	I_{MSP}	— –8	–95 —	µA
12	V_{DD} Supply Current[5] Run[6] LPSTOP, 4.194 MHz crystal, VCO Off (STSIM = 0)[7] LPSTOP, 32.768 kHz crystal, VCO Off (STSIM = 0)[7] LPSTOP, external clock input frequency = max f_{sys} WAIT[8]	I_{DD} S_{IDD} S_{IDD} S_{IDD} W_{IDD}	 — — — —	50 2 260 3.0 23	mA mA µA mA mA
13	Clock Synthesizer Operating Voltage	V_{DDSYN}	2.7	3.6	V
14	MC68CM16Z1V_{DDSYN} Supply Current[4] VCO on, crystal reference, maximum f_{sys}[7] External clock, maximum f_{sys} LPSTOP, 4.194 MHz crystal reference, VCO off (STSIM = 0)[7] V_{DD} powered down	I_{DDSYN}	— — — —	2 2.5 2 2	mA mA mA mA
14A	MC68CK16Z1/Z4 V_{DDSYN} Supply Current[4] VCO on, crystal reference, maximum f_{sys}[7] External clock, maximum f_{sys} LPSTOP, 32.768 kHz crystal reference, VCO off (STSIM = 0)[7] 32.768 kHz, V_{DD} powered down	I_{DDSYN}	— — — —	655 2.5 150 70	µA mA µA µA
15	RAM Standby Voltage[9] Specified V_{DD} applied V_{DD} = V_{SS}	V_{SB}	0.0 2.7	V_{DD} 3.6	V
16	MC68CK16Z1/Z4 RAM Standby Current[4, 9, 10] Normal RAM operation $V_{DD} > V_{SB} – 0.5$ V Transient condition $V_{SB} – 0.5$ V ≥ V_{DD} ≥ V_{SS} + 0.5 V Standby operation $V_{DD} < V_{SS}$ + 0.5 V	I_{SB}	— — —	10 3 50	µA mA µA

Freescale Semiconductor, Inc.

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

Table A-11 Low Voltage 16.78-MHz DC Characteristics (Continued)

(V_{DD} and V_{DDSYN} = 2.7 to 3.6Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
17	MC68CM16Z1/Z4 Power Dissipation[11]	P_D	—	191	mW
18	Input Capacitance[2, 7] All input-only pins All input/output pins	C_{in}	— —	10 20	pF
19	Load Capacitance[2] Group 1 I/O Pins, CLKOUT, FREEZE/QUOT, IPIPE0 Group 2 I/O Pins and CSBOOT, BG/CS Group 3 I/O Pins Group 4 I/O Pins	C_L	— — — —	90 100 100 100	pF

NOTES:

1. Applies to:
   Port ADA [7:0] — AN[7:0]
   Port E [7:4] — SIZ[1:0], AS, DS
   Port F [7:0] — IRQ[7:1], MODCLK
   Port GP[7:0] — IC4/OC5/OC1, IC[3:1], OC[4:1]/OC1
   Port MCCI[7:0] — TXD, PCS[3:1], PCS0/SS, SCK, MOSI, MISO
   BKPT/DSCLK, DSI/IPIPE1, PAI, PCLK, RESET, RXD, TSC
2. Input-Only Pins: EXTAL, TSC, BKPT/DSCLK, PAI, PCLK, RXD
   Output-Only Pins: CSBOOT, BG/CS1, CLKOUT, FREEZE/QUOT, DSO/IPIPE0, PWMA, PWMB
   Input/Output Pins:
   Group 1: Port GP[7:0] — IC4/OC5/OC1, IC[3:1], OC[4:1]/OC1, DATA[15:0], DSI/IPIPE1
   Group 2: Port C[6:0] — ADDR[22:19]/CS[9:6], FC[2:0]/CS[5:3]
   Port E[7:0] — SIZ[1:0], AS, DS, AVEC, DSACK[1:0]
   Port F[7:0] — IRQ[7:1], MODCLK
   Port MCCI[7:3] — TXD, PCS[3:1], PCS0/SS, ADDR23/CS10/ECLK
   Group 3: HALT, RESET
   Group 4: MISO, MOSI, SCK
3. Does not apply to HALT and RESET because they are open drain pins.
   Does not apply to port MCCI[7:0] (TXD, PCS[3:1], PCS0/SS, SCK, MOSI, MISO) in wired-OR mode.
4. Use of an active pulldown device is recommended.
5. Total operating current is the sum of the appropriate I_{DD}, I_{DDSYN}, I_{SB}, and I_{DDA}.
6. Current measured with system clock frequency of 16.78 MHz, all modules active.
7. This parameter is periodically sampled rather than 100% tested.
8. CPU16 in WAIT, all other modules inactive.
9. The RAM module will not switch into standby mode as long as V_{SB} does not exceed V_{DD} by more than 0.5 Volt. The RAM array cannot be accessed while the module is in standby mode.
10. When V_{DD} is transitioning during a power up or power down sequence, and V_{SB} is applied, current flows between the V_{STBY} and V_{DD} pins, which causes standby current to increase toward the maximum transient condition specification. System noise on the V_{DD} and V_{STBY} pins can contribute to this condition.
11. Power dissipation measured at specified system clock frequency, all modules active. Power dissipation can be calculated using the expression:
    P_D = Maximum V_{DD} (I_{DD} + I_{DDSYN} + I_{SB}) + Maximum V_{DDA} (I_{DDA})
    I_{DD} includes supply currents for all device modules powered by V_{DD} pins.

Freescale Semiconductor, Inc.

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

Table A-12 16.78-MHz DC Characteristics

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
1	Input High Voltage	V_{IH}	0.7 (V_{DD})	V_{DD} + 0.3	V
2	Input Low Voltage	V_{IL}	V_{SS} – 0.3	0.2 (V_{DD})	V
3	Input Hysteresis[1, 2]	V_{HYS}	0.5	—	V
4	Input Leakage Current[3, 4] V_{in} = V_{DD} or V_{SS}	I_{IN}	–2.5	2.5	µA
5	High Impedance (Off-State) Leakage Current [4, 5] V_{in} = V_{DD} or V_{SS}	I_{OZ}	–2.5	2.5	µA
6	CMOS Output High Voltage [4, 6, 7] I_{OH} = –10.0 µA	V_{OH}	V_{DD} –0.2	—	V
7	CMOS Output Low Voltage [4, 8] I_{OL} = 10.0 µA	V_{OL}	—	0.2	V
8	Output High Voltage [4, 6, 7] I_{OH} = –0.8 mA	V_{OH}	V_{DD} –0.8	—	V
9	Output Low Voltage[4, 8] I_{OL} = 1.6 mA I_{OL} = 5.3 mA I_{OL} = 12 mA	V_{OL}	— — —	0.4 0.4 0.4	V
10	Three State Control Input High Voltage	V_{IIHTSC}	1.6 (V_{DD})	9.1	V
11	Data Bus Mode Select Pull-Up Current[9, 10] V_{in} = V_{IL} V_{in} = V_{IH}	I_{MSP}	— –15	–120 —	µA
12	MC68HC16Z1 V_{DD} Supply Current[11, 12, 13] Run LPSTOP, crystal, VCO Off (STSIM = 0) LPSTOP, external clock input frequency = maximum f_{sys}	I_{DD}	— — —	110 350 5	mA µA mA
12A	MC68HC16Z2/Z3 V_{DD} Supply Current[11, 12, 13] Run LPSTOP, crystal, VCO Off (STSIM = 0) LPSTOP, external clock input frequency = maximum f_{sys}	I_{DD}	— — —	113 2 10	mA mA mA
13	Clock Synthesizer Operating Voltage	V_{DDSYN}	4.5	5.5	V
14	MC68HC16Z1 V_{DDSYN} Supply Current[11, 13] VCO on, crystal reference, maximum f_{sys} External clock, maximum f_{sys} LPSTOP, crystal reference, VCO off (STSIM = 0) V_{DD} powered down	I_{DDSYN}	— — — —	1 5 150 100	mA mA µA µA
14A	MC68HC16Z2/Z3 V_{DDSYN} Supply Current[11, 13] VCO on, crystal reference, maximum f_{sys} External clock, maximum f_{sys} LPSTOP, crystal reference, VCO off (STSIM = 0) V_{DD} powered down	I_{DDSYN}	— — — —	2 7 2 2	mA mA mA mA
15	RAM Standby Voltage[14] Specified V_{DD} applied V_{DD} = V_{SS}	V_{SB}	0.0 3.0	5.5 5.5	V

Freescale Semiconductor, Inc.

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

Table A-12 16.78-MHz DC Characteristics (Continued)

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
16	MC68HC16Z1 RAM Standby Current[12] Normal RAM operation[15] $V_{DD} > V_{SB} - 0.5$ V Transient condition $V_{SB} - 0.5$ V ≥ V_{DD} ≥ V_{SS} + 0.5 V Standby operation[14] $V_{DD} < V_{SS} + 0.5$ V	I_{SB}	— — —	TBD 2.5 50	µA mA µA
16A	MC68HC16Z2/Z3 RAM Standby Current[12] Normal RAM operation[15] $V_{DD} > V_{SB} - 0.5$ V Transient condition $V_{SB} - 0.5$ V ≥ V_{DD} ≥ V_{SS} + 0.5 V Standby operation[14] $V_{DD} < V_{SS} + 0.5$ V	I_{SB}	— — —	TBD 2.5 100	µA mA µA
17	MC68HC16Z1 Power Dissipation[16]	P_D	—	639	mW
17A	MC68HC16Z2/Z3 Power Dissipation[16]	P_D	—	666	mW
18	Input Capacitance[3, 7, 13] All input-only pins except ADC pins All input/output pins	C_{IN}	— —	10 20	pF
19	Load Capacitance[4] Group 1 I/O Pins, CLKOUT, FREEZE/QUOT, IPIPE0 Group 2 I/O Pins and $\overline{CSBOOT}$, $\overline{BG}/\overline{CS}$ Group 3 I/O Pins Group 4 I/O Pins	C_L	— — — —	90 100 130 200	pF

NOTES:
1. Refer to notes in **Table A-14**.

Freescale Semiconductor, Inc.

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

Table A-13 20.97-MHz DC Characteristics

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
1	Input High Voltage	V_{IH}	0.7 (V_{DD})	V_{DD} + 0.3	V
2	Input Low Voltage	V_{IL}	V_{SS} – 0.3	0.2 (V_{DD})	V
3	Input Hysteresis[1, 2]	V_{HYS}	0.5	—	V
4	Input Leakage Current[3, 4] V_{in} = V_{DD} or V_{SS}	I_{IN}	–2.5	2.5	µA
5	High Impedance (Off-State) Leakage Current[4, 5] V_{in} = V_{DD} or V_{SS}	I_{OZ}	–2.5	2.5	µA
6	CMOS Output High Voltage[4, 6, 7] I_{OH} = –10.0 µA	V_{OH}	V_{DD} –0.2	—	V
7	CMOS Output Low Voltage[4, 8] I_{OL} = 10.0 µA	V_{OL}	—	0.2	V
8	Output High Voltage[4, 6, 7] I_{OH} = –0.8 mA	V_{OH}	V_{DD} –0.8	—	V
9	Output Low Voltage[4, 8] I_{OL} = 1.6 mA I_{OL} = 5.3 mA I_{OL} = 12 mA	V_{OL}	— — —	0.4 0.4 0.4	V
10	Three State Control Input High Voltage	V_{IIHTSC}	1.6 (V_{DD})	9.1	V
11	Data Bus Mode Select Pull-Up Current[9, 10] V_{in} = V_{IL} V_{in} = V_{IH}	I_{MSP}	— –15	–120 —	µA
12	MC68HC16Z1 V_{DD} Supply Current[11, 12, 13] Run LPSTOP, crystal, VCO Off (STSIM = 0) LPSTOP, external clock input frequency = maximum f_{sys}	I_{DD}	— — —	140 350 5	mA µA mA
12A	MC68HC16Z2/Z3 V_{DD} Supply Current[11, 12, 13] Run LPSTOP, crystal, VCO Off (STSIM = 0) LPSTOP, external clock input frequency = maximum f_{sys}	I_{DD}	— — —	140 2 10	mA mA mA
13	Clock Synthesizer Operating Voltage	V_{DDSYN}	4.75	5.25	V
14	MC68HC16Z1 V_{DDSYN} Supply Current[11, 13] VCO on, crystal reference, maximum f_{sys} External clock, maximum f_{sys} LPSTOP, crystal reference, VCO off (STSIM = 0) V_{DD} powered down	I_{DDSYN}	— — — —	2 6 150 100	mA mA µA µA
14A	MC68HC16Z2/Z3 V_{DDSYN} Supply Current[11, 13] VCO on, crystal reference, maximum f_{sys} External clock, maximum f_{sys} LPSTOP, crystal reference, VCO off (STSIM = 0) V_{DD} powered down	I_{DDSYN}	— — — —	2.5 8.75 2 2	mA mA mA mA
15	RAM Standby Voltage[14] Specified V_{DD} applied V_{DD} = V_{SS}	V_{SB}	0.0 3.0	5.25 5.25	V

Freescale Semiconductor, Inc.

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

Table A-13 20.97-MHz DC Characteristics

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
16	MC68HC16Z1 RAM Standby Current[12] Normal RAM operation[15] $V_{DD} > V_{SB} - 0.5$ V Transient condition $V_{SB} - 0.5$ V ≥ V_{DD} ≥ V_{SS} + 0.5 V Standby operation[14] $V_{DD} < V_{SS} + 0.5$ V	I_{SB}	— — —	10 3 50	µA mA µA
16A	MC68HC16Z2/Z3 RAM Standby Current[12] Normal RAM operation[15] $V_{DD} > V_{SB} - 0.5$ V Transient condition $V_{SB} - 0.5$ V ≥ V_{DD} ≥ V_{SS} + 0.5 V Standby operation[14] $V_{DD} < V_{SS} + 0.5$ V	I_{SB}	— — —	10 3 100	µA mA µA
17	MC68HC16Z1 Power Dissipation[16]	P_D	—	772	mW
17A	MC68HC16Z2/Z3 Power Dissipation[16]	P_D	—	787	mW
18	Input Capacitance[3, 7, 13] All input-only pins except ADC pins All input/output pins	C_{IN}	— —	10 20	pF
19	Load Capacitance[4] Group 1 I/O Pins, CLKOUT, FREEZE/QUOT, IPIPE0 Group 2 I/O Pins and $\overline{\text{CSBOOT}}$, $\overline{\text{BG}}/\overline{\text{CS}}$ Group 3 I/O Pins Group 4 I/O Pins	C_L	— — — —	90 100 130 200	pF

NOTES:
1. Refer to notes in **Table A-14**.

Freescale Semiconductor, Inc.

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

Table A-14 25.17-MHz DC Characteristics

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
1	Input High Voltage	V_{IH}	0.7 (V_{DD})	V_{DD} + 0.3	V
2	Input Low Voltage	V_{IL}	V_{SS} – 0.3	0.2 (V_{DD})	V
3	Input Hysteresis[1, 2]	V_{HYS}	0.5	—	V
4	Input Leakage Current[3, 4] V_{in} = V_{DD} or V_{SS}	I_{IN}	–2.5	2.5	µA
5	High Impedance (Off-State) Leakage Current[4, 5] V_{in} = V_{DD} or V_{SS}	I_{OZ}	–2.5	2.5	µA
6	CMOS Output High Voltage[4, 6, 7] I_{OH} = –10.0 µA	V_{OH}	V_{DD} –0.2	—	V
7	CMOS Output Low Voltage[4, 8] I_{OL} = 10.0 µA	V_{OL}	—	0.2	V
8	Output High Voltage[4, 6, 7] I_{OH} = –0.8 mA	V_{OH}	V_{DD} –0.8	—	V
9	Output Low Voltage[4, 8] I_{OL} = 1.6 mA I_{OL} = 5.3 mA I_{OL} = 12 mA	V_{OL}	— — —	0.4 0.4 0.4	V
10	Three State Control Input High Voltage	V_{IIHTSC}	1.6 (V_{DD})	9.1	V
11	Data Bus Mode Select Pull-Up Current[9, 10] V_{in} = V_{IL} V_{in} = V_{IH}	I_{MSP}	— –15	–120 —	µA
12	MC68HC16Z1 V_{DD} Supply Current[11, 12, 13] Run LPSTOP, crystal, VCO Off (STSIM = 0) LPSTOP, external clock input frequency = maximum f_{sys}	I_{DD}	— — —	140 350 5	mA µA mA
12A	MC68HC16Z2/Z3 V_{DD} Supply Current [11, 12, 13] Run LPSTOP, crystal, VCO Off (STSIM = 0) LPSTOP, external clock input frequency = maximum f_{sys}	I_{DD}	— — —	140 2 10	mA mA mA
13	Clock Synthesizer Operating Voltage	V_{DDSYN}	4.75	5.25	V
14	MC68HC16Z1 V_{DDSYN} Supply Current[11, 13] VCO on, crystal reference, maximum f_{sys} External clock, maximum f_{sys} LPSTOP, crystal reference, VCO off (STSIM = 0) V_{DD} powered down	I_{DDSYN}	— — — —	2 7 150 100	mA mA µA µA
14A	MC68HC16Z2/Z3 V_{DDSYN} Supply Current[11, 13] VCO on, crystal reference, maximum f_{sys} External clock, maximum f_{sys} LPSTOP, crystal reference, VCO off (STSIM = 0) V_{DD} powered down	I_{DDSYN}	— — — —	2.5 8.75 2 2	mA mA mA mA
15	RAM Standby Voltage[14] Specified V_{DD} applied V_{DD} = V_{SS}	V_{SB}	0.0 3.0	5.25 5.25	V

Freescale Semiconductor, Inc.

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

Table A-14 25.17-MHz DC Characteristics (Continued)

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
16	MC68HC16Z1 RAM Standby Current[12] Normal RAM operation[15] $V_{DD} > V_{SB} - 0.5$ V Transient condition $V_{SB} - 0.5$ V ≥ V_{DD} ≥ V_{SS} + 0.5 V Standby operation[14] $V_{DD} < V_{SS} + 0.5$ V	I_{SB}	— — —	10 3 50	µA mA µA
16A	MC68HC16Z2/Z3 RAM Standby Current[12] Normal RAM operation[15] $V_{DD} > V_{SB} - 0.5$ V Transient condition $V_{SB} - 0.5$ V ≥ V_{DD} ≥ V_{SS} + 0.5 V Standby operation[14] $V_{DD} < V_{SS} + 0.5$ V	I_{SB}	— — —	10 3 100	µA mA µA
17	MC68HC16Z1 Power Dissipation[16]	P_D	—	777	mW
17A	MC68HC16Z2/Z3 Power Dissipation[16]	P_D	—	787	mW
18	Input Capacitance[3, 7, 13] All input-only pins except ADC pins All input/output pins	C_{IN}	— —	10 20	pF
19	Load Capacitance[4] Group 1 I/O Pins, CLKOUT, FREEZE/QUOT, IPIPE0 Group 2 I/O Pins and $\overline{CSBOOT}$, BG/$\overline{CS}$ Group 3 I/O Pins Group 4 I/O Pins	C_L	— — — —	90 100 130 200	pF

NOTES:

1. Applies to :
   Port ADA[7:0] — AN[7:0]
   Port E[7:4] — SIZ[1:0], $\overline{AS}$, $\overline{DS}$
   Port F[7:0] — $\overline{IRQ}$[7:1], MODCLK
   Port GP[7:0] — IC4/OC5/OC1, IC[3:1], OC[4:1]/OC1
   Port QS[7:0] — TXD, PCS[3:1], PCS0/$\overline{SS}$, SCK, MOSI, MISO
   $\overline{BKPT}$/DSCLK, DSI/IPIPE1, PAI, PCLK, $\overline{RESET}$, RXD, TSC
   EXTAL (when PLL enabled)
2. This parameter is periodically sampled rather than 100% tested.
3. Applies to all input-only pins except ADC pins.
4. Input-Only Pins: EXTAL, TSC, $\overline{BKPT}$/DSCLK, PAI, PCLK, RXD
   Input/Output: $\overline{CSBOOT}$, BG/$\overline{CS1}$, CLKOUT, FREEZE/QUOT, DSO/IPIPE0, PWMA, PWMB
   Output-Only Pins:
   Group 1: Port GP[7:0] — IC4/OC5/OC1, IC[3:1], OC[4:1]/OC1, DATA[15:0], DSI/IPIPE1
   Group 2: Port C[6:0] — ADDR[22:19]/$\overline{CS[9:6]}$, FC[2:0]/$\overline{CS[5:3]}$,
   Port E[7:0] — SIZ[1:0], $\overline{AS}$, $\overline{DS}$, $\overline{AVEC}$, $\overline{DSACK}$[1:0]
   Port F[7:0] — $\overline{IRQ}$[7:1], MODCLK
   Port QS[7:3] — TXD, PCS[3:1], PCS0/$\overline{SS}$
   ADDR23/$\overline{CS10}$/ECLK, ADDR[18:0], R/$\overline{W}$, $\overline{BERR}$, $\overline{BR}$/$\overline{CS0}$, $\overline{BGACK}$/$\overline{CS2}$
   Group 3: $\overline{HALT}$, $\overline{RESET}$
   Group 3: MISO, MOSI, SCK
5. Applies to all input/output and output pins.
6. Does not apply to $\overline{HALT}$ and $\overline{RESET}$ because they are open drain pins. Does not apply to port QS[7:0] (TXD, PCS[3:1], PCS0/$\overline{SS}$, SCK, MOSI, MISO) in wired-OR mode.
7. Applies to Group 1, 2, 4 input/output and all output pins.
8. Applies to Group 1, 2, 3, 4 input/output pins, $\overline{BG}$/$\overline{CS}$, CLKOUT, $\overline{CSBOOT}$, FREEZE/QUOT, and IPIPE0.
9. Applies to DATA[15:0].
10. Use of an active pulldown device is recommended.
11. Total operating current is the sum of the appropriate I_{DD}, I_{DDSYN}, I_{SB}, and I_{DDA}.
12. Current measured at maximum system clock frequency, all modules active.

Freescale Semiconductor, Inc.

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

13. The base configuration of the MC68HC16Z1, MC68CK16Z1, MC68HC16Z4, and the MC68CK16Z4 requires a 32.768 kHz crystal reference. The base configuration of the MC68CM16Z1, MC68HC16Z2, and the MC68HC16Z3 requires a 4.194 MHz crystal reference.
14. The RAM module will not switch into standby mode as long as V_{SB} does not exceed V_{DD} by more than 0.5 volts. The RAM array cannot be accessed while the module is in standby mode.
15. When V_{SB} is more than 0.3 V greater than V_{DD}, current flows between the V_{STBY} and V_{DD} pins, which causes standby current to increase toward the maximum transient condition specification. System noise on the V_{DD} and V_{STBY} pin can contribute to this condition.
16. Power dissipation is measured with the appropriate system clock frequency, all modules active. Power dissipation can be calculated using the following expression:

$$P_D = \text{Maximum } V_{DD} (I_{DD} + I_{DDSYN} + I_{SB}) + \text{Maximum } V_{DDA} (I_{DDA})$$

I_{DD} includes supply currents for all device modules powered by V_{DD} pins.

Freescale Semiconductor, Inc.

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

Table A-15 Low Voltage 16.78-MHz AC Timing

(V_{DD} and V_{DDSYN} = 2.7 to 3.6Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
F1	Frequency of Operation	f	—	16.78	MHz
1	Clock Period	t_{cyc}	59.6	—	ns
1A	ECLK Period	t_{Ecyc}	476	—	ns
1B	External Clock Input Period[2]	t_{Xcyc}	64	—	ns
2, 3	Clock Pulse Width[3]	t_{CW}	24	—	ns
2A, 3A	ECLK Pulse Width	t_{ECW}	236	—	ns
2B, 3B	External Clock Input High/Low Time[2]	t_{XCHL}	32	—	ns
4, 5	CLKOUT Rise and Fall Time	t_{Crf}	—	9	ns
4A, 5A	Rise and Fall Time (All outputs except CLKOUT)	t_{rf}	0	8	ns
4B, 5B	External Clock Input Rise and Fall Time[3]	t_{XCrf}	0	5	ns
6	Clock High to ADDR, FC, SIZ Valid[4]	t_{CHAV}	0	35	ns
7	Clock High to ADDR, Data, FC, SIZ High Impedance	t_{CHAZx}	2	59	ns
8	Clock High to ADDR, FC, SIZ Invalid	t_{CHAZn}	0	—	ns
9	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Asserted[4]	t_{CLSA}	2	25	ns
9A	$\overline{AS}$ to $\overline{DS}$ or $\overline{CS}$ Asserted (Read)[5]	t_{STSA}	–15	15	ns
11	ADDR, FC, SIZ Valid to $\overline{AS}$, $\overline{CS}$, (and $\overline{DS}$ Read) Asserted	t_{AVSA}	15	—	ns
12	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated	t_{CLSN}	2	29	ns
13	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to ADDR, FC, SIZ Invalid (Address Hold)	t_{SNAI}	15	—	ns
14	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted	t_{SWA}	110	—	ns
14A	$\overline{DS}$, $\overline{CS}$ Width Asserted (Write)	t_{SWAW}	45	—	ns
14B	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted (Fast Cycle)	t_{SWDW}	40	—	ns
15	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Width Negated[6]	t_{SN}	40	—	ns
16	Clock High to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ High Impedance	t_{CHSZ}	0	59	ns
17	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to R/$\overline{W}$ High	t_{SNRN}	15	—	ns
18	Clock High to R/$\overline{W}$ High	t_{CHRH}	0	30	ns
20	Clock High to R/$\overline{W}$ Low	t_{CHRL}	0	30	ns
21	R/$\overline{W}$ High to $\overline{AS}$, $\overline{CS}$ Asserted	t_{RAAA}	15	—	ns
22	R/$\overline{W}$ Low to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{RASA}	70	—	ns
23	Clock High to Data Out Valid	t_{CHDO}	—	30	ns
24	Data Out Valid to Negating Edge of $\overline{AS}$, $\overline{CS}$ (Fast Write Cycle)	t_{DVASN}	15	—	ns
25	$\overline{DS}$, $\overline{CS}$ Negated to Data Out Invalid (Data Out Hold)	t_{SNDOI}	15	—	ns
26	Data Out Valid to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{DVSA}	15	—	ns
27	Data In Valid to Clock Low (Data Setup)[4]	t_{DICL}	5	—	ns
27A	Late $\overline{BERR}$, $\overline{HALT}$ Asserted to Clock Low (Setup Time)	t_{BELCL}	20	—	ns
28	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{DSACK}$[1:0], $\overline{BERR}$, $\overline{HALT}$, $\overline{AVEC}$ Negated	t_{SNDN}	0	80	ns
29	$\overline{DS}$, $\overline{CS}$ Negated to Data In Invalid (Data In Hold)[7]	t_{SNDI}	0	—	ns
29A	$\overline{DS}$, $\overline{CS}$ Negated to Data In High Impedance[7, 8]	t_{SHDI}	—	55	ns

Freescale Semiconductor, Inc.

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

Table A-15 Low Voltage 16.78-MHz AC Timing (Continued)

(V_{DD} and V_{DDSYN} = 2.7 to 3.6Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
30	CLKOUT Low to Data In Invalid (Fast Cycle Hold)[7]	t_{CLDI}	15	—	ns
30A	CLKOUT Low to Data In High Impedance[7]	t_{CLDH}	—	90	ns
31	$\overline{DSACK[1:0]}$ Asserted to Data In Valid[9]	t_{DADI}	—	50	ns
33	Clock Low to $\overline{BG}$ Asserted/Negated	t_{CLBAN}	—	30	ns
35	$\overline{BR}$ Asserted to $\overline{BG}$ Asserted[10]	t_{BRAGA}	1	—	t_{cyc}
37	$\overline{BGACK}$ Asserted to $\overline{BG}$ Negated	t_{GAGN}	1	2	t_{cyc}
39	$\overline{BG}$ Width Negated	t_{GH}	2	—	t_{cyc}
39A	$\overline{BG}$ Width Asserted	t_{GA}	1	—	t_{cyc}
46	R/$\overline{W}$ Width Asserted (Write or Read)	t_{RWA}	150	—	ns
46A	R/$\overline{W}$ Width Asserted (Fast Write or Read Cycle)	t_{RWAS}	90	—	ns
47A	Asynchronous Input Setup Time $\overline{BR}$, $\overline{BGACK}$, $\overline{DSACK[1:0]}$, $\overline{BERR}$, $\overline{AVEC}$, $\overline{HALT}$	t_{AIST}	15	—	ns
47B	Asynchronous Input Hold Time	t_{AIHT}	15	—	ns
48	$\overline{DSACK[1:0]}$ Asserted to $\overline{BERR}$, $\overline{HALT}$ Asserted[12]	t_{DABA}	—	30	ns
53	Data Out Hold from Clock High	t_{DOCH}	0	—	ns
54	Clock High to Data Out High Impedance	t_{CHDH}	—	28	ns
55	R/$\overline{W}$ Asserted to Data Bus Impedance Change	t_{RADC}	40	—	ns
70	Clock Low to Data Bus Driven (Show Cycle)	t_{SCLDD}	0	30	ns
71	Data Setup Time to Clock Low (Show Cycle)	t_{SCLDS}	15	—	ns
72	Data Hold from Clock Low (Show Cycle)	t_{SCLDH}	10	—	ns
73	$\overline{BKPT}$ Input Setup Time	t_{BKST}	20	—	ns
74	$\overline{BKPT}$ Input Hold Time	t_{BKHT}	15	—	ns
75	Mode Select Setup Time, DATA[15:0], MODCLK, $\overline{BKPT}$ pins	t_{MSS}	20	—	t_{cyc}
76	Mode Select Hold Time, DATA[15:0], MODCLK, $\overline{BKPT}$ pins	t_{MSH}	0	—	ns
77	$\overline{RESET}$ Assertion Time[12]	t_{RSTA}	4	—	t_{cyc}
78	$\overline{RESET}$ Rise Time[13]	t_{RSTR}	—	10	t_{cyc}
100	CLKOUT High to Phase 1 Asserted[14]	t_{CHP1A}	3	40	ns
101	CLKOUT High to Phase 2 Asserted[14]	t_{CHP2A}	3	40	ns
102	Phase 1 Valid to $\overline{AS}$ or $\overline{DS}$ Asserted[14]	t_{P1VSN}	10	—	ns
103	Phase 2 Valid to $\overline{AS}$ or $\overline{DS}$ Negated[14]	t_{P2VSN}	10	—	ns
104	$\overline{AS}$ or $\overline{DS}$ Valid to Phase 1 Negated[14]	t_{SAP1N}	10	—	ns
105	$\overline{AS}$ or $\overline{DS}$ Negated to Phase 2 Negated[14]	t_{SNP2N}	10	—	ns

NOTES:
1. Refer to notes in **Table A-18**.

Freescale Semiconductor, Inc.

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

Table A-16 16.78-MHz AC Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 10 %, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
F1	Frequency of Operation	f	—	16.78	MHz
1	Clock Period	t_{cyc}	59.6	—	ns
1A	ECLK Period	t_{Ecyc}	476	—	ns
1B	External Clock Input Period[2]	t_{Xcyc}	59.6	—	ns
2, 3	Clock Pulse Width[3]	t_{CW}	24	—	ns
2A, 3A	ECLK Pulse Width	t_{ECW}	236	—	ns
2B, 3B	External Clock Input High/Low Time[2]	t_{XCHL}	29.8	—	ns
4, 5	CLKOUT Rise and Fall Time	t_{Crf}	—	5	ns
4A, 5A	Rise and Fall Time (All Outputs except CLKOUT)	t_{rf}	—	8	ns
4B, 5B	External Clock Input Rise and Fall Time[3]	t_{XCrf}	—	5	ns
6	Clock High to ADDR, FC, SIZE Valid[4]	t_{CHAV}	0	29	ns
7	Clock High to ADDR, Data, FC, SIZE, High Impedance	t_{CHAZx}	0	59	ns
8	Clock High to ADDR, FC, SIZE, Invalid	t_{CHAZn}	0	—	ns
9	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Asserted[4]	t_{CLSA}	2	24	ns
9A	$\overline{AS}$ to $\overline{DS}$ or $\overline{CS}$ Asserted (Read)[5]	t_{STSA}	-15	15	ns
11	ADDR, FC, SIZE Valid to $\overline{AS}$, $\overline{CS}$, (and $\overline{DS}$ Read) Asserted	t_{AVSA}	15	—	ns
12	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated	t_{CLSN}	2	29	ns
13	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to ADDR, FC SIZE Invalid (Address Hold)	t_{SNAI}	15	—	ns
14	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted	t_{SWA}	100	—	ns
14A	$\overline{DS}$, $\overline{CS}$ Width Asserted (Write)	t_{SWAW}	45	—	ns
14B	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted (Fast Cycle)	t_{SWDW}	40	—	ns
15	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Width Negated[6]	t_{SN}	40	—	ns
16	Clock High to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ High Impedance	t_{CHSZ}	—	59	ns
17	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to R/$\overline{W}$ High	t_{SNRN}	15	—	ns
18	Clock High to R/$\overline{W}$ High	t_{CHRH}	0	29	ns
20	Clock High to R/$\overline{W}$ Low	t_{CHRL}	0	29	ns
21	R/$\overline{W}$ High to $\overline{AS}$, $\overline{CS}$ Asserted	t_{RAAA}	15	—	ns
22	R/$\overline{W}$ Low to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{RASA}	70	—	ns
23	Clock High to Data Out Valid	t_{CHDO}	—	29	ns
24	Data Out Valid to Negating Edge of $\overline{AS}$, $\overline{CS}$ (Fast Write Cycle)	t_{DVASN}	15	—	ns
25	$\overline{DS}$, $\overline{CS}$ Negated to Data Out Invalid (Data Out Hold)	t_{SNDOI}	15	—	ns
26	Data Out Valid to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{DVSA}	15	—	ns
27	Data In Valid to Clock Low (Data Setup)[4]	t_{DICL}	5	—	ns
27A	Late $\overline{BERR}$, $\overline{HALT}$ Asserted to Clock Low (Setup Time)	t_{BELCL}	20	—	ns
28	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{DSACK}$[1:0], $\overline{BERR}$, $\overline{HALT}$, $\overline{AVEC}$ Negated	t_{SNDN}	0	80	ns
29	$\overline{DS}$, $\overline{CS}$ Negated to Data In Invalid (Data In Hold)[7]	t_{SNDI}	0	—	ns
29A	$\overline{DS}$, $\overline{CS}$ Negated to Data In High Impedance[7, 8]	t_{SHDI}	—	55	ns

Freescale Semiconductor, Inc.

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

Table A-16 16.78-MHz AC Timing (Continued)

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 10 %, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
30	CLKOUT Low to Data In Invalid (Fast Cycle Hold)[7]	t_{CLDI}	15	—	ns
30A	CLKOUT Low to Data In High Impedance[7]	t_{CLDH}	—	90	ns
31	$\overline{DSACK[1:0]}$ Asserted to Data In Valid[9]	t_{DADI}	—	50	ns
33	Clock Low to $\overline{BG}$ Asserted/Negated	t_{CLBAN}	—	29	ns
35	$\overline{BR}$ Asserted to $\overline{BG}$ Asserted[10]	t_{BRAGA}	1	—	t_{cyc}
37	$\overline{BGACK}$ Asserted to $\overline{BG}$ Negated	t_{GAGN}	1	2	t_{cyc}
39	$\overline{BG}$ Width Negated	t_{GH}	2	—	t_{cyc}
39A	$\overline{BG}$ Width Asserted	t_{GA}	1	—	t_{cyc}
46	R/$\overline{W}$ Width Asserted (Write or Read)	t_{RWA}	150	—	ns
46A	R/$\overline{W}$ Width Asserted (Fast Write or Read Cycle)	t_{RWAS}	90	—	ns
47A	Asynchronous Input Setup Time $\overline{BR}$, $\overline{BGACK}$, $\overline{DSACK[1:0]}$, $\overline{BERR}$, $\overline{AVEC}$, $\overline{HALT}$	t_{AIST}	5	—	ns
47B	Asynchronous Input Hold Time	t_{AIHT}	15	—	ns
48	$\overline{DSACK[1:0]}$ Asserted to $\overline{BERR}$, $\overline{HALT}$ Asserted[11]	t_{DABA}	—	30	ns
53	Data Out Hold from Clock High	t_{DOCH}	0	—	ns
54	Clock High to Data Out High Impedance	t_{CHDH}	—	28	ns
55	R/$\overline{W}$ Asserted to Data Bus Impedance Change	t_{RADC}	40	—	ns
70	Clock Low to Data Bus Driven (Show Cycle)	t_{SCLDD}	0	29	ns
71	Data Setup Time to Clock Low (Show Cycle)	t_{SCLDS}	15	—	ns
72	Data Hold from Clock Low (Show Cycle)	t_{SCLDH}	10	—	ns
73	$\overline{BKPT}$ Input Setup Time	t_{BKST}	15	—	ns
74	$\overline{BKPT}$ Input Hold Time	t_{BKHT}	10	—	ns
75	Mode Select Setup Time, DATA[15:0], MODCLK, $\overline{BKPT}$ pins	t_{MSS}	20	—	t_{cyc}
76	Mode Select Hold Time, DATA[15:0], MODCLK, $\overline{BKPT}$ pins	t_{MSH}	0	—	ns
77	$\overline{RESET}$ Assertion Time[12]	t_{RSTA}	4	—	t_{cyc}
78	$\overline{RESET}$ Rise Time[13]	t_{RSTR}	—	10	t_{cyc}
100	CLKOUT High to Phase 1 Asserted[14]	t_{CHP1A}	3	40	ns
101	CLKOUT High to Phase 2 Asserted[14]	t_{CHP2A}	3	40	ns
102	Phase 1 Valid to $\overline{AS}$ or $\overline{DS}$ Asserted[14]	t_{P1VSA}	10	—	ns
103	Phase 2 Valid to $\overline{AS}$ or $\overline{DS}$ Asserted[14]	t_{P2VSN}	10	—	ns
104	$\overline{AS}$ or $\overline{DS}$ Valid to Phase 1 Negated[14]	t_{SAP1N}	10	—	ns
105	$\overline{AS}$ or $\overline{DS}$ Negated to Phase 2 Negated[14]	t_{SNP2N}	10	—	ns

NOTES:
1. Refer to notes in **Table A-18**.

Freescale Semiconductor, Inc.

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

Table A-17 20.97-MHz AC Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
F1	Frequency of Operation	f	—	20.97	MHz
1	Clock Period	t_{cyc}	47.7	—	ns
1A	ECLK Period	t_{Ecyc}	381	—	ns
1B	External Clock Input Period[2]	t_{Xcyc}	47.7	—	ns
2, 3	Clock Pulse Width[3]	t_{CW}	18.8	—	ns
2A, 3A	ECLK Pulse Width	t_{ECW}	183	—	ns
2B, 3B	External Clock Input High/Low Time[2]	t_{XCHL}	23.8	—	ns
4, 5	CLKOUT Rise and Fall Time	t_{Crf}	—	5	ns
4A, 5A	Rise and Fall Time (All Outputs except CLKOUT)	t_{rf}	—	8	ns
4B, 5B	External Clock Input Rise and Fall Time[3]	t_{XCrf}	—	5	ns
6	Clock High to ADDR, FC, SIZE Valid[4]	t_{CHAV}	0	23	ns
7	Clock High to ADDR, Data, FC, SIZE, High Impedance	t_{CHAZx}	0	47	ns
8	Clock High to ADDR, FC, SIZE, Invalid	t_{CHAZn}	0	—	ns
9	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Asserted[4]	t_{CLSA}	0	23	ns
9A	$\overline{AS}$ to $\overline{DS}$ or $\overline{CS}$ Asserted (Read)[5]	t_{STSA}	-10	10	ns
11	ADDR, FC, SIZE Valid to $\overline{AS}$, $\overline{CS}$, (and $\overline{DS}$ Read) Asserted	t_{AVSA}	10	—	ns
12	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated	t_{CLSN}	2	23	ns
13	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to ADDR, FC SIZE Invalid (Address Hold)	t_{SNAI}	10	—	ns
14	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted	t_{SWA}	80	—	ns
14A	$\overline{DS}$, $\overline{CS}$ Width Asserted (Write)	t_{SWAW}	36	—	ns
14B	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted (Fast Cycle)	t_{SWDW}	32	—	ns
15	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Width Negated[6]	t_{SN}	32	—	ns
16	Clock High to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ High Impedance	t_{CHSZ}	—	47	ns
17	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to R/$\overline{W}$ High	t_{SNRN}	10	—	ns
18	Clock High to R/$\overline{W}$ High	t_{CHRH}	0	23	ns
20	Clock High to R/$\overline{W}$ Low	t_{CHRL}	0	23	ns
21	R/$\overline{W}$ High to $\overline{AS}$, $\overline{CS}$ Asserted	t_{RAAA}	10	—	ns
22	R/$\overline{W}$ Low to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{RASA}	54	—	ns
23	Clock High to Data Out Valid	t_{CHDO}	—	23	ns
24	Data Out Valid to Negating Edge of $\overline{AS}$, $\overline{CS}$ (Fast Write Cycle)	t_{DVASN}	10	—	ns
25	$\overline{DS}$, $\overline{CS}$ Negated to Data Out Invalid (Data Out Hold)	t_{SNDOI}	10	—	ns
26	Data Out Valid to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{DVSA}	10	—	ns
27	Data In Valid to Clock Low (Data Setup)[4]	t_{DICL}	5	—	ns
27A	Late $\overline{BERR}$, $\overline{HALT}$ Asserted to Clock Low (Setup Time)	t_{BELCL}	15	—	ns
28	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{DSACK}$[1:0], $\overline{BERR}$, $\overline{HALT}$, $\overline{AVEC}$ Negated	t_{SNDN}	0	60	ns
29	$\overline{DS}$, $\overline{CS}$ Negated to Data In Invalid (Data In Hold)[7]	t_{SNDI}	0	—	ns
29A	$\overline{DS}$, $\overline{CS}$ Negated to Data In High Impedance[7, 8]	t_{SHDI}	—	48	ns

Freescale Semiconductor, Inc.

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

Table A-17 20.97-MHz AC Timing (Continued)

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
30	CLKOUT Low to Data In Invalid (Fast Cycle Hold)[7]	t_{CLDI}	10	—	ns
30A	CLKOUT Low to Data In High Impedance[7]	t_{CLDH}	—	72	ns
31	$\overline{DSACK[1:0]}$ Asserted to Data In Valid[9]	t_{DADI}	—	46	ns
33	Clock Low to $\overline{BG}$ Asserted/Negated	t_{CLBAN}	—	23	ns
35	$\overline{BR}$ Asserted to $\overline{BG}$ Asserted[10]	t_{BRAGA}	1	—	t_{cyc}
37	$\overline{BGACK}$ Asserted to $\overline{BG}$ Negated	t_{GAGN}	1	2	t_{cyc}
39	$\overline{BG}$ Width Negated	t_{GH}	2	—	t_{cyc}
39A	$\overline{BG}$ Width Asserted	t_{GA}	1	—	t_{cyc}
46	R/$\overline{W}$ Width Asserted (Write or Read)	t_{RWA}	115	—	ns
46A	R/$\overline{W}$ Width Asserted (Fast Write or Read Cycle)	t_{RWAS}	70	—	ns
47A	Asynchronous Input Setup Time $\overline{BR}$, $\overline{BGACK}$, $\overline{DSACK[1:0]}$, $\overline{BERR}$, $\overline{AVEC}$, $\overline{HALT}$	t_{AIST}	5	—	ns
47B	Asynchronous Input Hold Time	t_{AIHT}	12	—	ns
48	$\overline{DSACK[1:0]}$ Asserted to $\overline{BERR}$, $\overline{HALT}$ Asserted[11]	t_{DABA}	—	30	ns
53	Data Out Hold from Clock High	t_{DOCH}	0	—	ns
54	Clock High to Data Out High Impedance	t_{CHDH}	—	23	ns
55	R/$\overline{W}$ Asserted to Data Bus Impedance Change	t_{RADC}	32	—	ns
70	Clock Low to Data Bus Driven (Show Cycle)	t_{SCLDD}	0	23	ns
71	Data Setup Time to Clock Low (Show Cycle)	t_{SCLDS}	10	—	ns
72	Data Hold from Clock Low (Show Cycle)	t_{SCLDH}	10	—	ns
73	$\overline{BKPT}$ Input Setup Time	t_{BKST}	10	—	ns
74	$\overline{BKPT}$ Input Hold Time	t_{BKHT}	10	—	ns
75	Mode Select Setup Time, DATA[15:0], MODCLK, $\overline{BKPT}$ pins	t_{MSS}	20	—	t_{cyc}
76	Mode Select Hold Time, DATA[15:0], MODCLK, $\overline{BKPT}$ pins	t_{MSH}	0	—	ns
77	$\overline{RESET}$ Assertion Time[12]	t_{RSTA}	4	—	t_{cyc}
78	$\overline{RESET}$ Rise Time[13]	t_{RSTR}	—	10	t_{cyc}
100	CLKOUT High to Phase 1 Asserted[14]	t_{CHP1A}	3	40	ns
101	CLKOUT High to Phase 2 Asserted[14]	t_{CHP2A}	3	40	ns
102	Phase 1 Valid to $\overline{AS}$ or $\overline{DS}$ Asserted[14]	t_{P1VSA}	10	—	ns
103	Phase 2 Valid to $\overline{AS}$ or $\overline{DS}$ Asserted[14]	t_{P2VSN}	10	—	ns
104	$\overline{AS}$ or $\overline{DS}$ Valid to Phase 1 Negated[14]	t_{SAP1N}	10	—	ns
105	$\overline{AS}$ or $\overline{DS}$ Negated to Phase 2 Negated[14]	t_{SNP2N}	10	—	ns

NOTES:
1. Refer to notes in **Table A-18**.

Freescale Semiconductor, Inc.

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

Table A-18 25.17-MHz AC Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
F1	Frequency of Operation	f	—	25.166	MHz
1	Clock Period	t_{cyc}	39.7	—	ns
1A	ECLK Period	t_{Ecyc}	318	—	ns
1B	External Clock Input Period[2]	t_{Xcyc}	39.7	—	ns
2, 3	Clock Pulse Width[3]	t_{CW}	15	—	ns
2A, 3A	ECLK Pulse Width	t_{ECW}	155	—	ns
2B, 3B	External Clock Input High/Low Time[2]	t_{XCHL}	19.8	—	ns
4, 5	CLKOUT Rise and Fall Time	t_{Crf}	—	5	ns
4A, 5A	Rise and Fall Time (All Outputs except CLKOUT)	t_{rf}	—	8	ns
4B, 5B	External Clock Input Rise and Fall Time[3]	t_{XCrf}	—	4	ns
6	Clock High to ADDR, FC, SIZ Valid[4]	t_{CHAV}	0	19	ns
7	Clock High to ADDR, Data, FC, SIZ, High Impedance	t_{CHAZx}	0	39	ns
8	Clock High to ADDR, FC, SIZ, Invalid	t_{CHAZn}	0	—	ns
9	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Asserted[4]	t_{CLSA}	2	19	ns
9A	$\overline{AS}$ to $\overline{DS}$ or $\overline{CS}$ Asserted (Read)[5]	t_{STSA}	–10	15	ns
11	ADDR, FC, SIZE Valid to $\overline{AS}$, $\overline{CS}$, (and $\overline{DS}$ Read) Asserted	t_{AVSA}	8	—	ns
12	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated	t_{CLSN}	2	19	ns
13	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to ADDR, FC, SIZ Invalid (Address Hold)	t_{SNAI}	8	—	ns
14	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted	t_{SWA}	65	—	ns
14A	$\overline{DS}$, $\overline{CS}$ Width Asserted (Write)	t_{SWAW}	25	—	ns
14B	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted (Fast Cycle)	t_{SWDW}	22	—	ns
15	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Width Negated[6]	t_{SN}	22	—	ns
16	Clock High to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ High Impedance	t_{CHSZ}	—	39	ns
17	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to R/$\overline{W}$ High	t_{SNRN}	10	—	ns
18	Clock High to R/$\overline{W}$ High	t_{CHRH}	0	19	ns
20	Clock High to R/$\overline{W}$ Low	t_{CHRL}	0	19	ns
21	R/$\overline{W}$ High to $\overline{AS}$, $\overline{CS}$ Asserted	t_{RAAA}	10	—	ns
22	R/$\overline{W}$ Low to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{RASA}	40	—	ns
23	Clock High to Data Out Valid	t_{CHDO}	—	19	ns
24	Data Out Valid to Negating Edge of $\overline{AS}$, $\overline{CS}$ (Fast Write Cycle)	t_{DVASN}	7	—	ns
25	$\overline{DS}$, $\overline{CS}$ Negated to Data Out Invalid (Data Out Hold)	t_{SNDOI}	5	—	ns
26	Data Out Valid to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{DVSA}	8	—	ns
27	Data In Valid to Clock Low (Data Setup)[4]	t_{DICL}	5	—	ns
27A	Late $\overline{BERR}$, $\overline{HALT}$ Asserted to Clock Low (Setup Time)	t_{BELCL}	10	—	ns
28	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{DSACK}$[1:0], $\overline{BERR}$, $\overline{HALT}$, $\overline{AVEC}$ Negated	t_{SNDN}	0	50	ns
29	$\overline{DS}$, $\overline{CS}$ Negated to Data In Invalid (Data In Hold)[7]	t_{SNDI}	0	—	ns
29A	$\overline{DS}$, $\overline{CS}$ Negated to Data In High Impedance[7, 8]	t_{SHDI}	—	45	ns

Freescale Semiconductor, Inc.

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

Table A-18 25.17-MHz AC Timing (Continued)

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
30	CLKOUT Low to Data In Invalid (Fast Cycle Hold)[7]	t_{CLDI}	8	—	ns
30A	CLKOUT Low to Data In High Impedance[7]	t_{CLDH}	—	60	ns
31	$\overline{DSACK}$[1:0] Asserted to Data In Valid[9]	t_{DADI}	—	35	ns
33	Clock Low to $\overline{BG}$ Asserted/Negated	t_{CLBAN}	—	19	ns
35	$\overline{BR}$ Asserted to $\overline{BG}$ Asserted[10]	t_{BRAGA}	1	—	t_{cyc}
37	$\overline{BGACK}$ Asserted to $\overline{BG}$ Negated	t_{GAGN}	1	2	t_{cyc}
39	$\overline{BG}$ Width Negated	t_{GH}	2	—	t_{cyc}
39A	$\overline{BG}$ Width Asserted	t_{GA}	1	—	t_{cyc}
46	R/$\overline{W}$ Width Asserted (Write or Read)	t_{RWA}	90	—	ns
46A	R/$\overline{W}$ Width Asserted (Fast Write or Read Cycle)	t_{RWAS}	55	—	ns
47A	Asynchronous Input Setup Time $\overline{BR}$, $\overline{BGACK}$, $\overline{DSACK}$[1:0], $\overline{BERR}$, $\overline{AVEC}$, $\overline{HALT}$	t_{AIST}	5	—	ns
47B	Asynchronous Input Hold Time	t_{AIHT}	10	—	ns
48	$\overline{DSACK}$[1:0] Asserted to $\overline{BERR}$, $\overline{HALT}$ Asserted[11]	t_{DABA}	—	27	ns
53	Data Out Hold from Clock High	t_{DOCH}	0	—	ns
54	Clock High to Data Out High Impedance	t_{CHDH}	—	23	ns
55	R/$\overline{W}$ Asserted to Data Bus Impedance Change	t_{RADC}	25	—	ns
70	Clock Low to Data Bus Driven (Show Cycle)	t_{SCLDD}	0	19	ns
71	Data Setup Time to Clock Low (Show Cycle)	t_{SCLDS}	8	—	ns
72	Data Hold from Clock Low (Show Cycle)	t_{SCLDH}	8	—	ns
73	$\overline{BKPT}$ Input Setup Time	t_{BKST}	10	—	ns
74	$\overline{BKPT}$ Input Hold Time	t_{BKHT}	10	—	ns
75	Mode Select Setup Time (DATA[15:0], MODCLK, $\overline{BKPT}$)	t_{MSS}	20	—	t_{cyc}
76	Mode Select Hold Time (DATA[15:0], MODCLK, $\overline{BKPT}$)	t_{MSH}	0	—	ns
77	$\overline{RESET}$ Assertion Time[12]	t_{RSTA}	4	—	t_{cyc}
78	$\overline{RESET}$ Rise Time[13]	t_{RSTR}	—	10	t_{cyc}
100	CLKOUT High to Phase 1 Asserted[14]	t_{CHP1A}	3	34	ns
101	CLKOUT High to Phase 2 Asserted[14]	t_{CHP2A}	3	34	ns
102	Phase 1 Valid to $\overline{AS}$ or $\overline{DS}$ Asserted[14]	t_{P1VSA}	9	—	ns
103	Phase 2 Valid to $\overline{AS}$ or $\overline{DS}$ Asserted[14]	t_{P2VSN}	9	—	ns
104	$\overline{AS}$ or $\overline{DS}$ Valid to Phase 1 Negated[14]	t_{SAP1N}	9	—	ns
105	$\overline{AS}$ or $\overline{DS}$ Negated to Phase 2 Negated[14]	t_{SNP2N}	9	—	ns

NOTES:

1. All AC timing is shown with respect to V_{IH}/V_{IL} levels unless otherwise noted.
2. When an external clock is used, minimum high and low times are based on a 50% duty cycle. The minimum allowable t_{Xcyc} period is reduced when the duty cycle of the external clock varies. The relationship between external clock input duty cycle and minimum t_{Xcyc} is expressed:

   Minimum t_{Xcyc} period = minimum t_{XCHL} / (50% – external clock input duty cycle tolerance).

Freescale Semiconductor, Inc.

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

3. Parameters for an external clock signal applied while the internal PLL is disabled (MODCLK pin held low during reset) do not pertain to an external reference applied while the PLL is enabled (MODCLK pin held high during reset). When the PLL is enabled, the clock synthesizer detects successive transitions of the reference signal. If transitions occur within the correct clock period, rise/fall times and duty cycle are not critical.
4. Address access time = (2.5 + WS) $t_{cyc} - t_{CHAV} - t_{DICL}$
   Chip select access time = (2 + WS) $t_{cyc} - t_{CLSA} - t_{DICL}$
   Where: WS = number of wait states. When fast termination is used (2 clock bus) WS = –1.
5. Specification 9A is the worst-case skew between $\overline{AS}$ and $\overline{DS}$ or $\overline{CS}$. The amount of skew depends on the relative loading of these signals. When loads are kept within specified limits, skew will not cause $\overline{AS}$ and $\overline{DS}$ to fall outside the limits shown in specification 9.
6. If multiple-chip selects are used, $\overline{CS}$ width negated (specification 15) applies to the time from the negation of a heavily loaded chip-select to the assertion of a lightly loaded chip select. The $\overline{CS}$ width negated specification between multiple chip-selects does not apply to chip selects being used for synchronous ECLK cycles.
7. Hold times are specified with respect to $\overline{DS}$ or $\overline{CS}$ on asynchronous reads and with respect to CLKOUT on fast cycle reads. The user is free to use either hold time.
8. Maximum value is equal to (t_{cyc} / 2) + 25 ns.
9. If the asynchronous setup time (specification 47A) requirements are satisfied, the $\overline{DSACK[1:0]}$ low to data setup time (specification 31) and $\overline{DSACK[1:0]}$ low to $\overline{BERR}$ low setup time (specification 48) can be ignored. The data must only satisfy the data-in to clock low setup time (specification 27) for the following clock cycle. $\overline{BERR}$ must satisfy only the late $\overline{BERR}$ low to clock low setup time (specification 27A) for the following clock cycle.
10. To ensure coherency during every operand transfer, $\overline{BG}$ is not asserted in response to $\overline{BR}$ until after all cycles of the current operand transfer are complete.
11. In the absence of $\overline{DSACK[1:0]}$, $\overline{BERR}$ is an asynchronous input using the asynchronous setup time (specification 47A).
12. After external $\overline{RESET}$ negation is detected, a short transition period (approximately 2 t_{cyc}) elapses, then the SIM drives $\overline{RESET}$ low for 512 t_{cyc}.
13. External logic must pull $\overline{RESET}$ high during this period in order for normal MCU operation to begin.
14. Eight pipeline states are multiplexed into IPIPE[1:0]. The multiplexed signals have two phases.

Freescale Semiconductor, Inc.

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

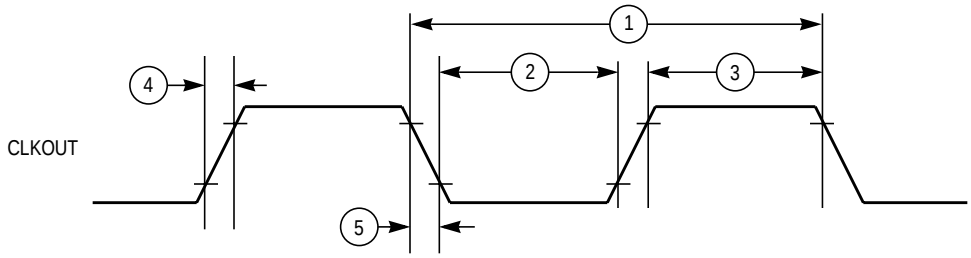
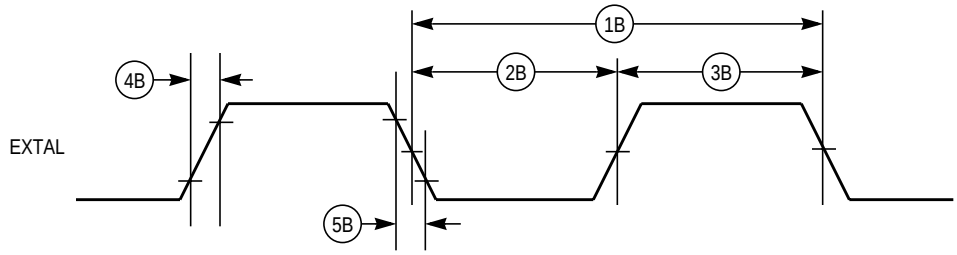


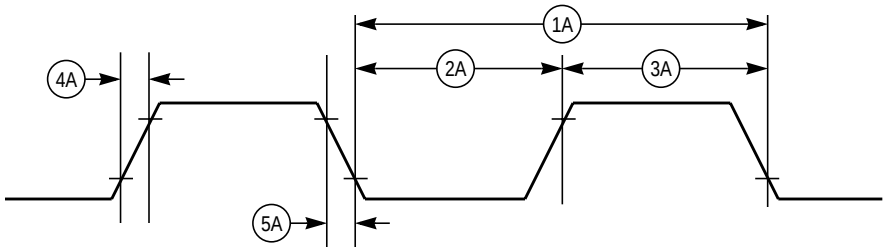
Figure A-1 CLKOUT Output Timing Diagram



NOTE: TIMING SHOWN WITH RESPECT TO V_{IH}/V_{IL} LEVELS.
PULSE WIDTH SHOWN WITH RESPECT TO 50% V_{DD}.

16 EXT CLK INPUT TIM

Figure A-2 External Clock Input Timing Diagram



NOTE: TIMING SHOWN WITH RESPECT TO V_{IH}/V_{IL} LEVELS.

16 ECLK OUTPUT TIM

Figure A-3 ECLK Output Timing Diagram

Freescale Semiconductor, Inc.

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

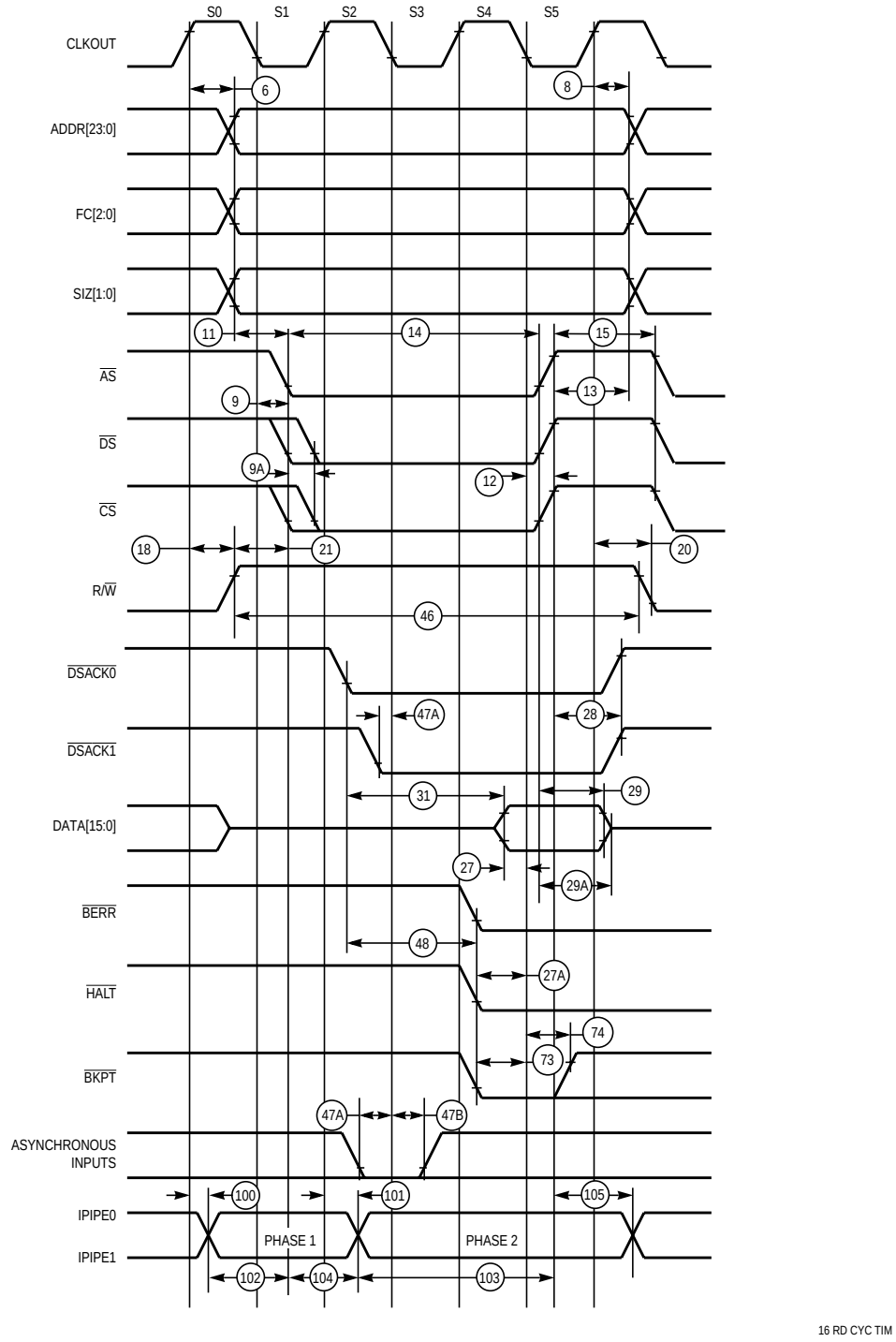


Figure A-4 Read Cycle Timing Diagram

Freescale Semiconductor, Inc.

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

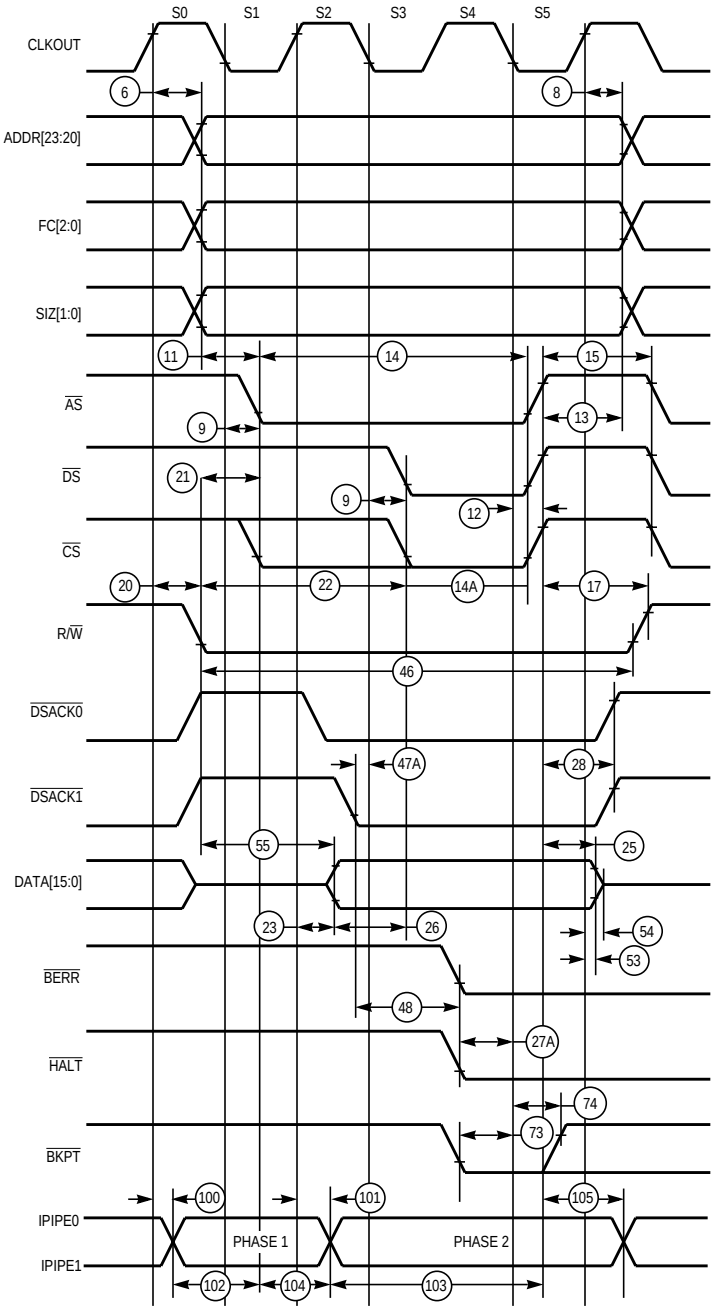


Figure A-5 Write Cycle Timing Diagram

Freescale Semiconductor, Inc.

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

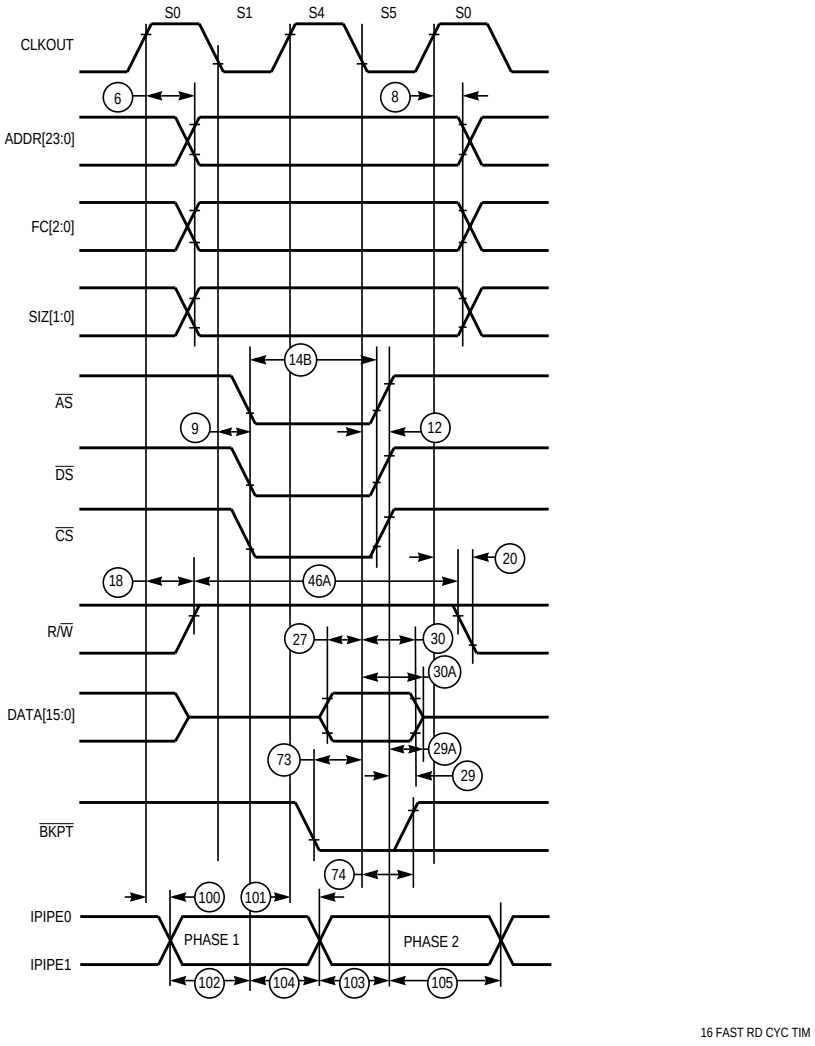


Figure A-6 Fast Termination Read Cycle Timing Diagram

Freescale Semiconductor, Inc.

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

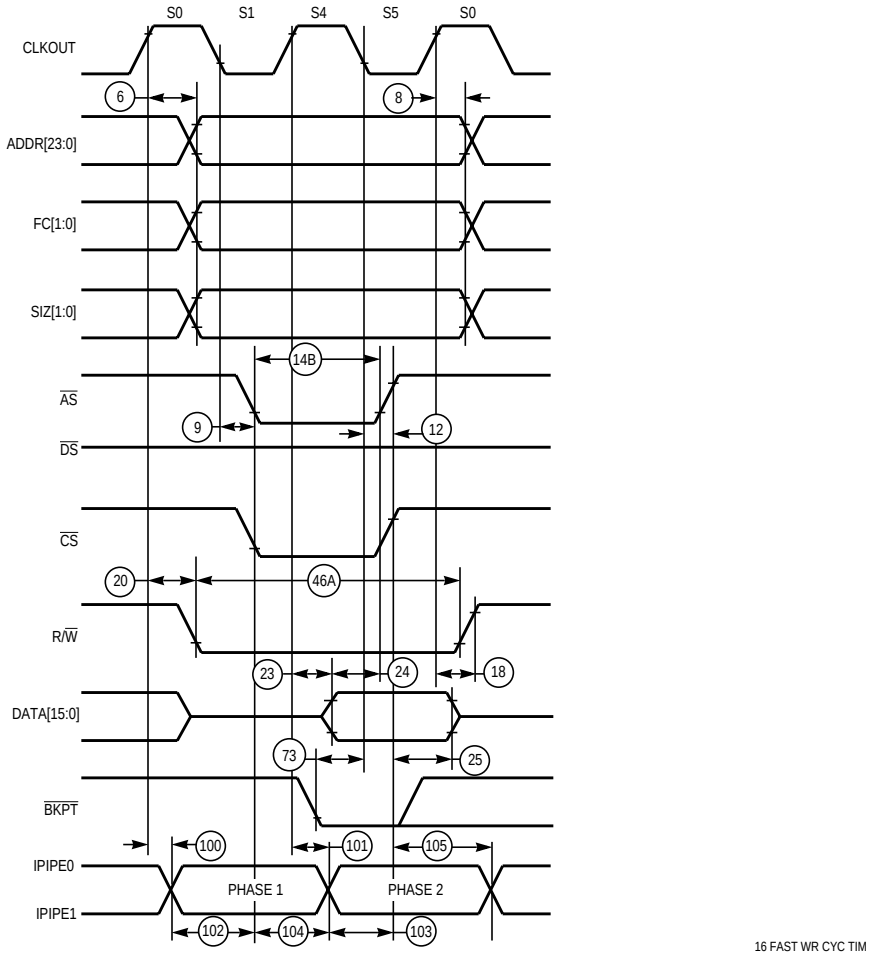


Figure A-7 Fast Termination Write Cycle Timing Diagram

Freescale Semiconductor, Inc.

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

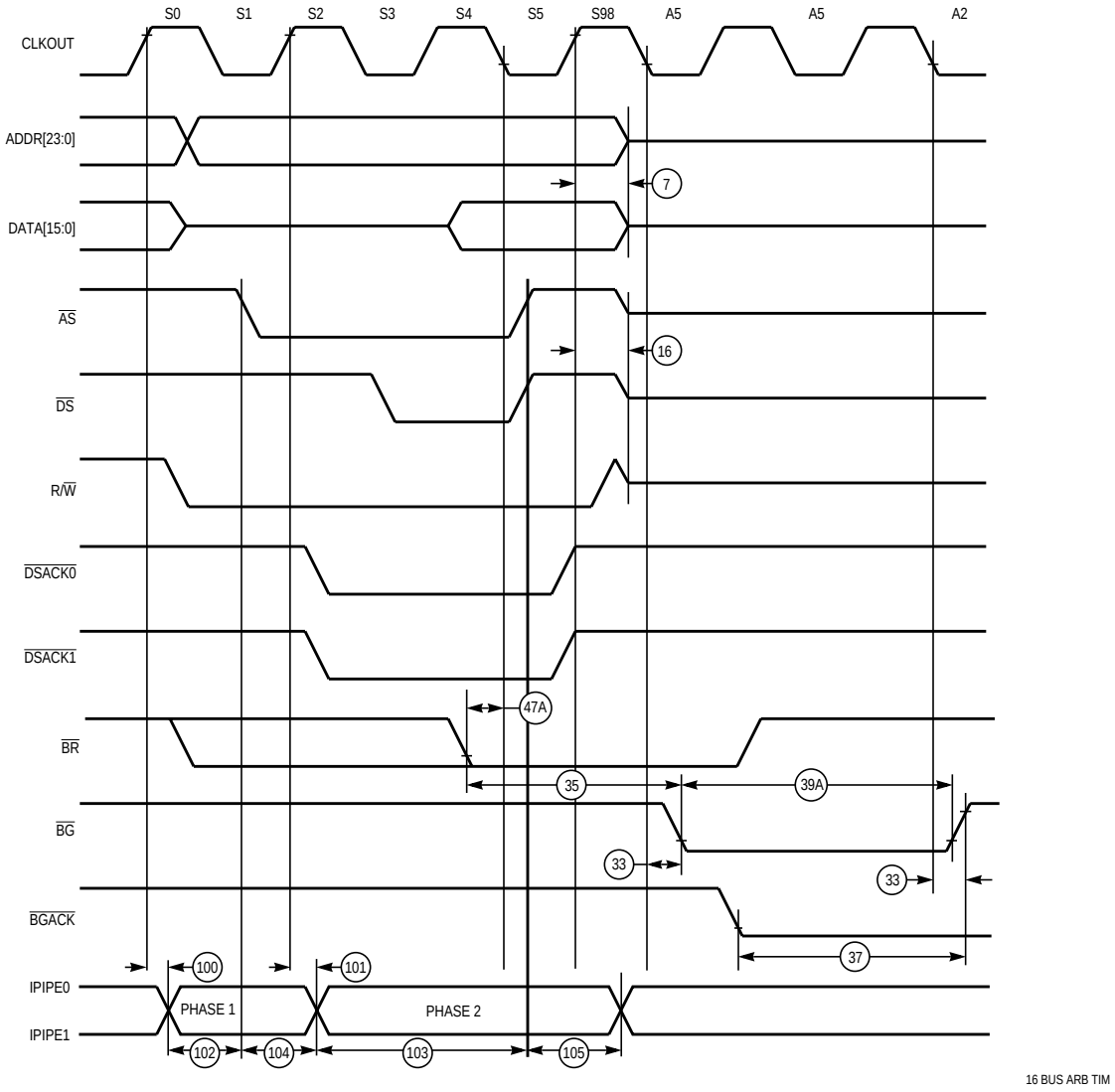


Figure A-8 Bus Arbitration Timing Diagram — Active Bus Case

Freescale Semiconductor, Inc.

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

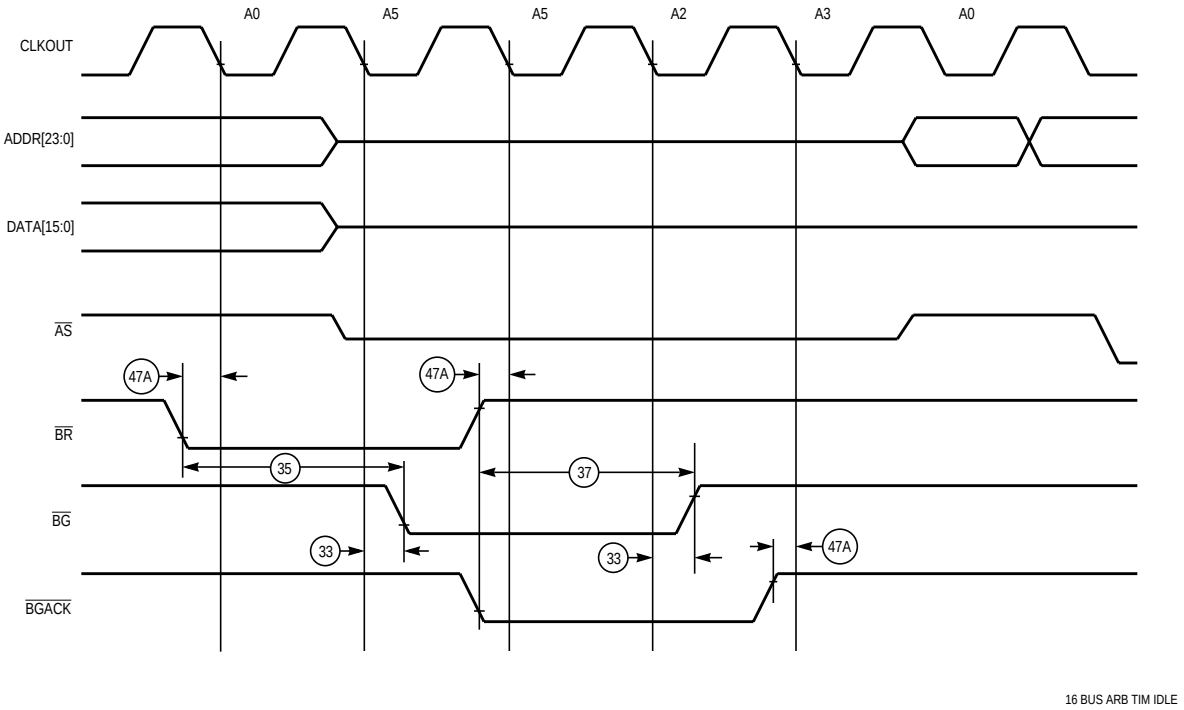
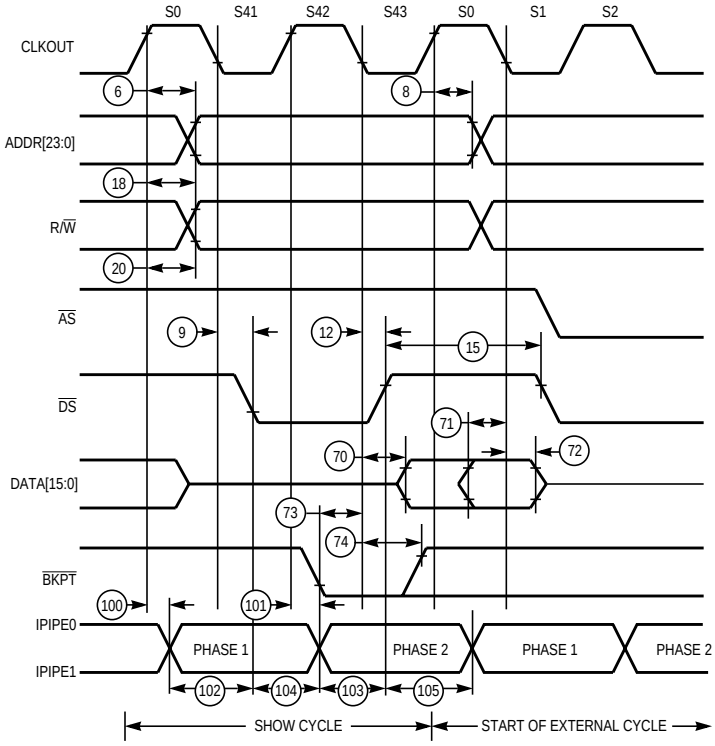


Figure A-9 Bus Arbitration Timing Diagram — Idle Bus Case

Freescale Semiconductor, Inc.



NOTE:
SHOW CYCLES CAN STRETCH DURING CLOCK PHASE S42 WHEN BUS ACCESSES TAKE LONGER THAN TWO CYCLES DUE TO IMB MODULE WAIT-STATE INSERTION.

16 SHW CYC TIM

Figure A-10 Show Cycle Timing Diagram

Freescale Semiconductor, Inc.

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

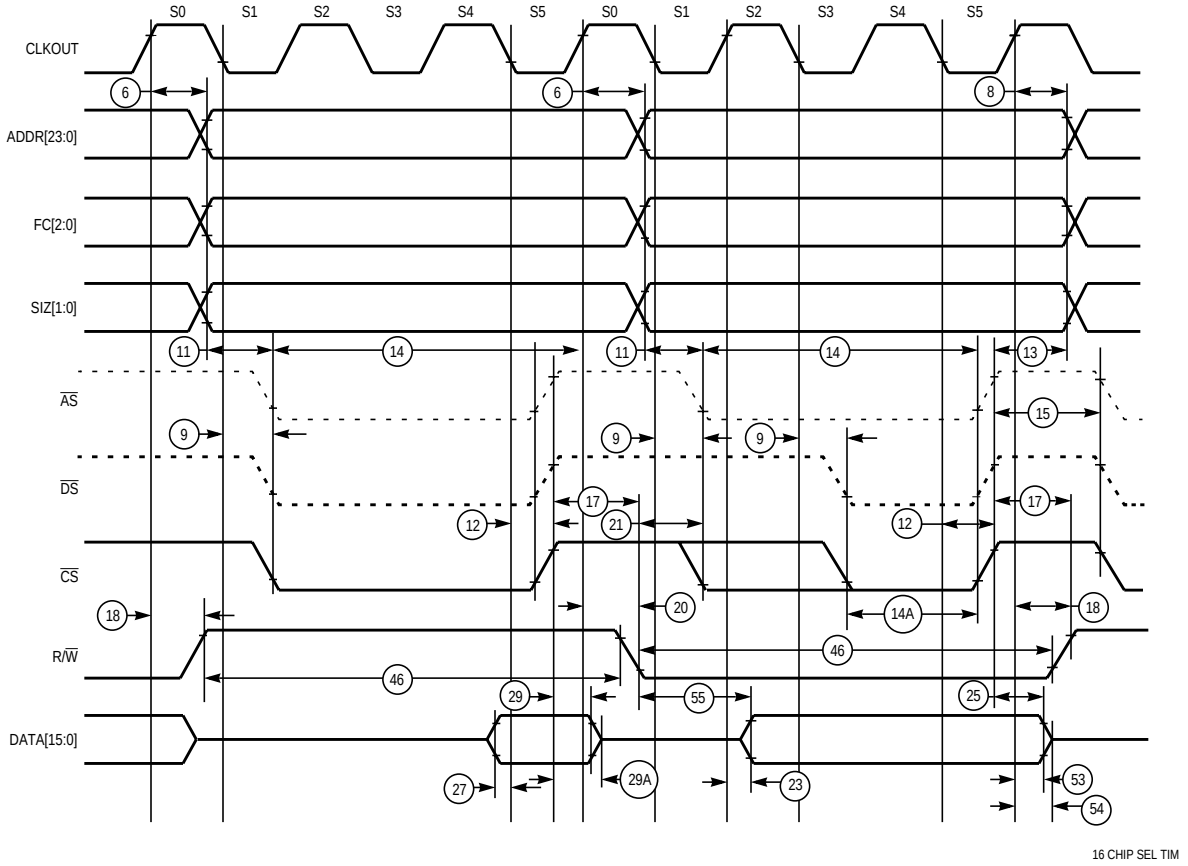


Figure A-11 Chip-Select Timing Diagram

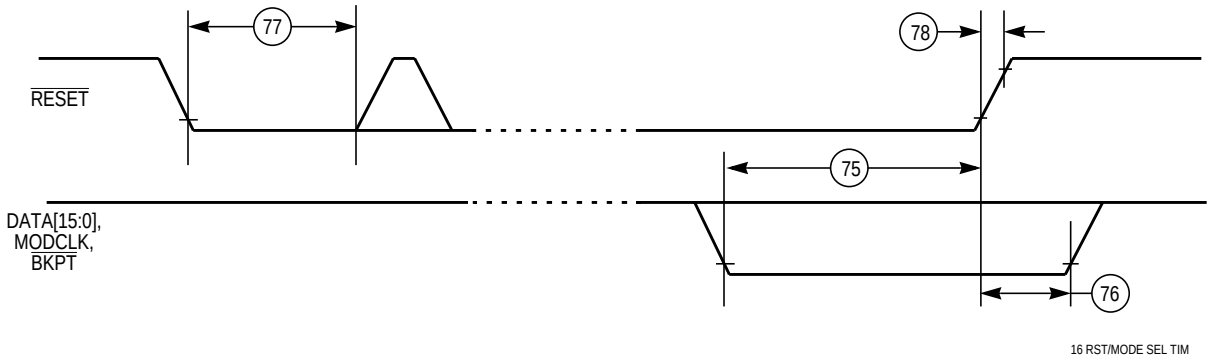


Figure A-12 Reset and Mode Select Timing Diagram

Freescale Semiconductor, Inc.

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

Table A-19 Low Voltage 16.78-MHz Background Debug Mode Timing

(V_{DD} and V_{DDSYN} = 2.7 to 3.6Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
B0	DSI Input Setup Time	t_{DSISU}	15	—	ns
B1	DSI Input Hold Time	t_{DSIH}	15	—	ns
B2	DSCLK Setup Time	t_{DSCSU}	15	—	ns
B3	DSCLK Hold Time	t_{DSCH}	15	—	ns
B4	DSO Delay Time	t_{DSOD}	—	35	ns
B5	DSCLK Cycle Time	t_{DSCCYC}	2	—	t_{cyc}
B6	CLKOUT High to FREEZE Asserted/Negated	t_{FRZAN}	—	50	ns
B7	CLKOUT High to IPIPE1 High Impedance	t_{IFZ}	—	50	ns
B8	CLKOUT High to IPIPE1 Valid	t_{IF}	—	50	ns
B9	DSCLK Low Time	t_{DSCLO}	1	—	t_{cyc}
B10	IPIPE1 High Impedance to FREEZE Asserted	t_{IPFA}	TBD	—	t_{cyc}
B11	FREEZE Negated to IPIPE[0:1] Active	t_{FRIP}	TBD	—	t_{cyc}

NOTES:
1. All AC timing is shown with respect to V_{IH}/V_{IL} levels unless otherwise noted.

Table A-20 16.78-MHz Background Debug Mode Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
B0	DSI Input Setup Time	t_{DSISU}	15	—	ns
B1	DSI Input Hold Time	t_{DSIH}	10	—	ns
B2	DSCLK Setup Time	t_{DSCSU}	15	—	ns
B3	DSCLK Hold Time	t_{DSCH}	10	—	ns
B4	DSO Delay Time	t_{DSOD}	—	25	ns
B5	DSCLK Cycle Time	t_{DSCCYC}	2	—	t_{cyc}
B6	CLKOUT High to FREEZE Asserted/Negated	t_{FRZAN}	—	50	ns
B7	CLKOUT High to IPIPE1 High Impedance	t_{IFZ}	—	TBD	ns
B8	CLKOUT High to IPIPE1 Valid	t_{IF}	—	TBD	ns
B9	DSCLK Low Time	t_{DSCLO}	1	—	t_{cyc}
B10	IPIPE1 High Impedance to FREEZE Asserted	t_{IPFA}	TBD	—	t_{cyc}
B11	FREEZE Negated to IPIPE[0:1] Active	t_{FRIP}	TBD	—	t_{cyc}

NOTES:
1. All AC timing is shown with respect to V_{IH}/V_{IL} levels unless otherwise noted.

Freescale Semiconductor, Inc.

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

Table A-21 20.97-MHz Background Debug Mode Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
B0	DSI Input Setup Time	t_{DSISU}	15	—	ns
B1	DSI Input Hold Time	t_{DSIH}	10	—	ns
B2	DSCLK Setup Time	t_{DSCSU}	15	—	ns
B3	DSCLK Hold Time	t_{DSCH}	10	—	ns
B4	DSO Delay Time	t_{DSOD}	—	25	ns
B5	DSCLK Cycle Time	t_{DSCCYC}	2	—	t_{cyc}
B6	CLKOUT High to FREEZE Asserted/Negated	t_{FRZAN}	—	50	ns
B7	CLKOUT High to IPIPE1 High Impedance	t_{IFZ}	—	50	ns
B8	CLKOUT High to IPIPE1 Valid	t_{IF}	—	50	ns
B9	DSCLK Low Time	t_{DSCLO}	1	—	t_{cyc}
B10	IPIPE1 High Impedance to FREEZE Asserted	t_{IPFA}	TBD	—	t_{cyc}
B11	FREEZE Negated to IPIPE[0:1] Active	t_{FRIP}	TBD	—	t_{cyc}

NOTES:
1. All AC timing is shown with respect to V_{IH}/V_{IL} levels unless otherwise noted.

Table A-22 25.17-MHz Background Debug Mode Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
B0	DSI Input Setup Time	t_{DSISU}	10	—	ns
B1	DSI Input Hold Time	t_{DSIH}	5	—	ns
B2	DSCLK Setup Time	t_{DSCSU}	10	—	ns
B3	DSCLK Hold Time	t_{DSCH}	5	—	ns
B4	DSO Delay Time	t_{DSOD}	—	20	ns
B5	DSCLK Cycle Time	t_{DSCCYC}	2	—	t_{cyc}
B6	CLKOUT High to FREEZE Asserted/Negated	t_{FRZAN}	—	20	ns
B7	CLKOUT High to IPIPE1 High Impedance	t_{IFZ}	—	20	ns
B8	CLKOUT High to IPIPE1 Valid	t_{IF}	—	20	ns
B9	DSCLK Low Time	t_{DSCLO}	1	—	t_{cyc}
B10	IPIPE1 High Impedance to FREEZE Asserted	t_{IPFA}	TBD	—	t_{cyc}
B11	FREEZE Negated to IPIPE[0:1] Active	t_{FRIP}	TBD	—	t_{cyc}

NOTES:
1. All AC timing is shown with respect to V_{IH}/V_{IL} levels unless otherwise noted.

Freescale Semiconductor, Inc.

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

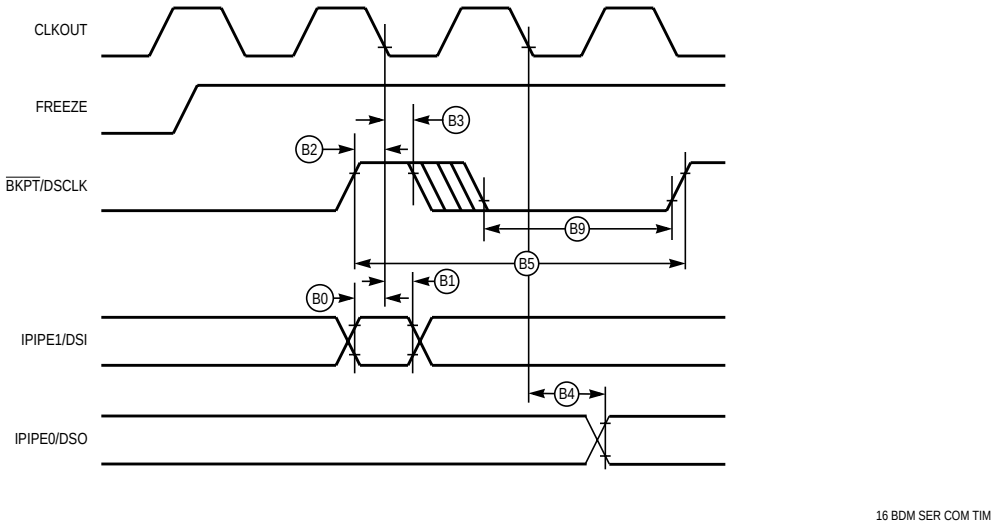


Figure A-13 Background Debug Mode Timing Diagram (Serial Communication)

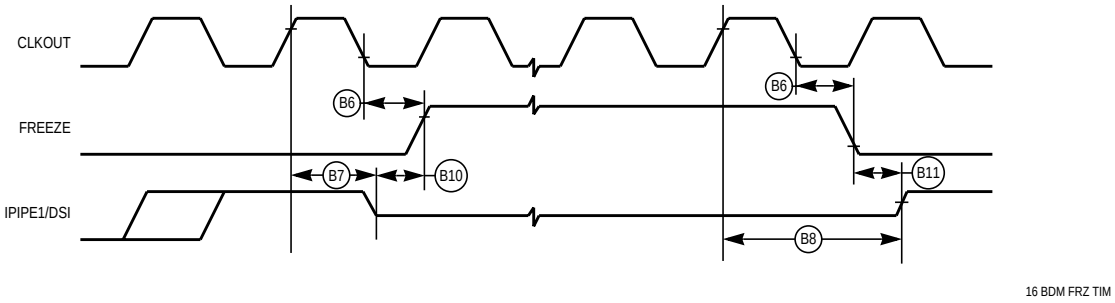


Figure A-14 Background Debug Mode Timing Diagram (Freeze Assertion)

Freescale Semiconductor, Inc.

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

Table A-23 Low Voltage ECLK Bus Timing

(V_{DD} and V_{DDSYN} = 2.7 to 3.6 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
E1	ECLK Low to Address Valid[2]	t_{EAD}	—	60	ns
E2	ECLK Low to Address Hold	t_{EAH}	10	—	ns
E3	ECLK Low to $\overline{CS}$ Valid ($\overline{CS}$ Delay)	t_{ECSD}	—	150	ns
E4	ECLK Low to $\overline{CS}$ Hold	t_{ECSH}	15	—	ns
E5	$\overline{CS}$ Negated Width	t_{ECSN}	30	—	ns
E6	Read Data Setup Time	t_{EDSR}	30	—	ns
E7	Read Data Hold Time	t_{EDHR}	15	—	ns
E8	ECLK Low to Data High Impedance	t_{EDHZ}	—	60	ns
E9	$\overline{CS}$ Negated to Data Hold (Read)	t_{ECDH}	0	—	ns
E10	$\overline{CS}$ Negated to Data High Impedance	t_{ECDZ}	—	1	t_{cyc}
E11	ECLK Low to Data Valid (Write)	t_{EDDW}	—	2	t_{cyc}
E12	ECLK Low to Data Hold (Write)	t_{EDHW}	5	—	ns
E13	$\overline{CS}$ Negated to Data Hold (Write)	t_{ECHW}	0	—	ns
E14	Address Access Time (Read)[3]	t_{EACC}	386	—	ns
E15	Chip-Select Access Time (Read)[4]	t_{EACS}	296	—	ns
E16	Address Setup Time	t_{EAS}	—	1/2	t_{cyc}

NOTES:
1. Refer to notes in **Table A-26**.

Freescale Semiconductor, Inc.

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

Table A-24 16.78-MHz ECLK Bus Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
E1	ECLK Low to Address Valid[2]	t_{EAD}	—	60	ns
E2	ECLK Low to Address Hold	t_{EAH}	10	—	ns
E3	ECLK Low to $\overline{CS}$ Valid ($\overline{CS}$ Delay)	t_{ECSD}	—	150	ns
E4	ECLK Low to $\overline{CS}$ Hold	t_{ECSH}	15	—	ns
E5	$\overline{CS}$ Negated Width	t_{ECSN}	30	—	ns
E6	Read Data Setup Time	t_{EDSR}	30	—	ns
E7	Read Data Hold Time	t_{EDHR}	15	—	ns
E8	ECLK Low to Data High Impedance	t_{EDHZ}	—	60	ns
E9	$\overline{CS}$ Negated to Data Hold (Read)	t_{ECDH}	0	—	ns
E10	$\overline{CS}$ Negated to Data High Impedance	t_{ECDZ}	—	1	t_{cyc}
E11	ECLK Low to Data Valid (Write)	t_{EDDW}	—	2	t_{cyc}
E12	ECLK Low to Data Hold (Write)	t_{EDHW}	5	—	ns
E13	$\overline{CS}$ Negated to Data Hold (Write)	t_{ECHW}	0	—	ns
E14	Address Access Time (Read)[3]	t_{EACC}	386	—	ns
E15	Chip-Select Access Time (Read)[4]	t_{EACS}	296	—	ns
E16	Address Setup Time	t_{EAS}	—	1/2	t_{cyc}

NOTES:
1. Refer to notes in **Table A-26**.

Freescale Semiconductor, Inc.

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

Table A-25 20.97-MHz ECLK Bus Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
E1	ECLK Low to Address Valid[2]	t_{EAD}	—	48	ns
E2	ECLK Low to Address Hold	t_{EAH}	10	—	ns
E3	ECLK Low to $\overline{CS}$ Valid ($\overline{CS}$ Delay)	t_{ECSD}	—	120	ns
E4	ECLK Low to $\overline{CS}$ Hold	t_{ECSH}	10	—	ns
E5	$\overline{CS}$ Negated Width	t_{ECSN}	25	—	ns
E6	Read Data Setup Time	t_{EDSR}	25	—	ns
E7	Read Data Hold Time	t_{EDHR}	5	—	ns
E8	ECLK Low to Data High Impedance	t_{EDHZ}	—	48	ns
E9	$\overline{CS}$ Negated to Data Hold (Read)	t_{ECDH}	0	—	ns
E10	$\overline{CS}$ Negated to Data High Impedance	t_{ECDZ}	—	1	t_{cyc}
E11	ECLK Low to Data Valid (Write)	t_{EDDW}	—	2	t_{cyc}
E12	ECLK Low to Data Hold (Write)	t_{EDHW}	10	—	ns
E13	$\overline{CS}$ Negated to Data Hold (Write)	t_{ECHW}	0	—	ns
E14	Address Access Time (Read)[3]	t_{EACC}	308	—	ns
E15	Chip-Select Access Time (Read)[4]	t_{EACS}	236	—	ns
E16	Address Setup Time	t_{EAS}	1/2	—	t_{cyc}

NOTES:
1. Refer to notes in **Table A-26**.

Freescale Semiconductor, Inc.

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

Table A-26 25.17-MHz ECLK Bus Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
E1	ECLK Low to Address Valid[2]	t_{EAD}	—	40	ns
E2	ECLK Low to Address Hold	t_{EAH}	10	—	ns
E3	ECLK Low to $\overline{CS}$ Valid ($\overline{CS}$ Delay)	t_{ECSD}	—	100	ns
E4	ECLK Low to $\overline{CS}$ Hold	t_{ECSH}	10	—	ns
E5	$\overline{CS}$ Negated Width	t_{ECSN}	20	—	ns
E6	Read Data Setup Time	t_{EDSR}	25	—	ns
E7	Read Data Hold Time	t_{EDHR}	5	—	ns
E8	ECLK Low to Data High Impedance	t_{EDHZ}	—	40	ns
E9	$\overline{CS}$ Negated to Data Hold (Read)	t_{ECDH}	0	—	ns
E10	$\overline{CS}$ Negated to Data High Impedance	t_{ECDZ}	—	1	t_{cyc}
E11	ECLK Low to Data Valid (Write)	t_{EDDW}	—	2	t_{cyc}
E12	ECLK Low to Data Hold (Write)	t_{EDHW}	5	—	ns
E13	$\overline{CS}$ Negated to Data Hold (Write)	t_{ECHW}	0	—	ns
E14	Address Access Time (Read)[3]	t_{EACC}	255	—	ns
E15	Chip-Select Access Time (Read)[4]	t_{EACS}	195	—	ns
E16	Address Setup Time	t_{EAS}	—	1/2	t_{cyc}

NOTES:

1. All AC timing is shown with respect to V_{IH}/V_{IL} levels unless otherwise noted.
2. When previous bus cycle is not an ECLK cycle, the address may be valid before ECLK goes low.
3. Address access time = $t_{Ecyc} - t_{EAD} - t_{EDSR}$.
4. Chip select access time = $t_{Ecyc} - t_{ECSD} - t_{EDSR}$.

Freescale Semiconductor, Inc.

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

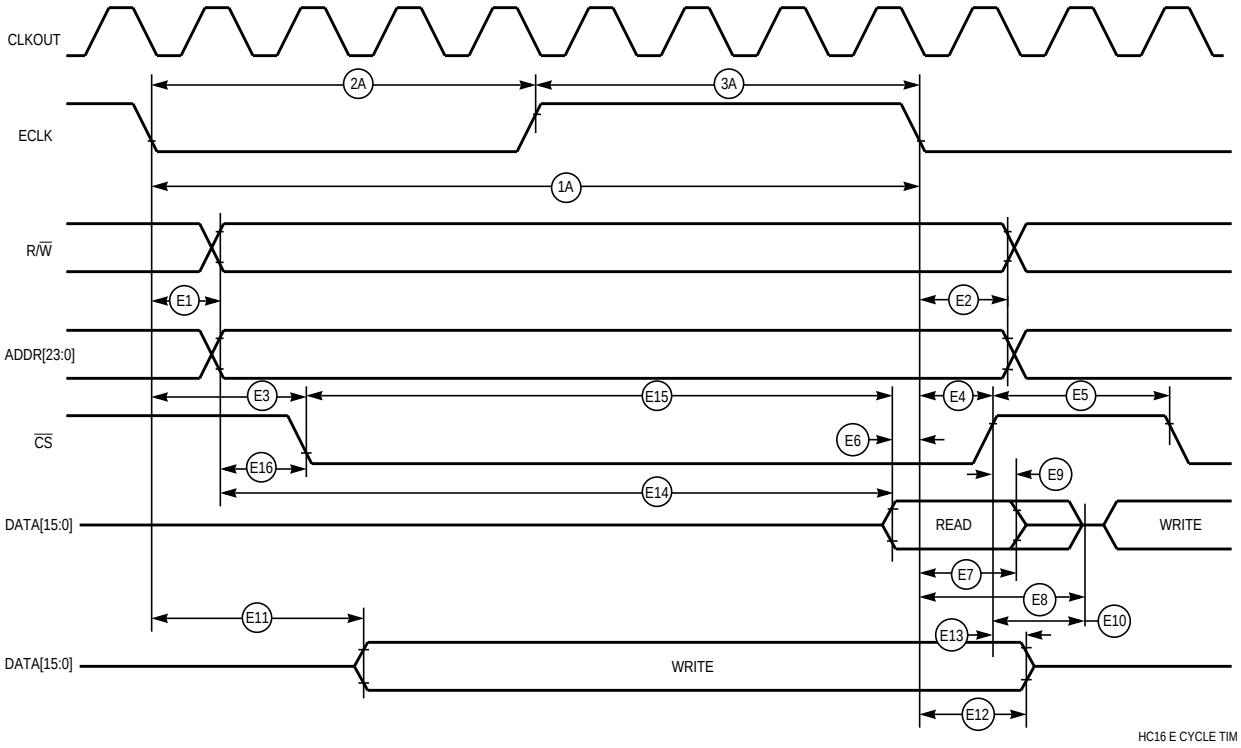


Figure A-15 ECLK Timing Diagram

Freescale Semiconductor, Inc.

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

Table A-27 Low Voltage QSPI Timing

(V_{DD} and V_{DDSYN} = 2.7 to 3.6 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Function	Symbol	Min	Max	Unit
1	Operating Frequency Master Slave	f_{op}	DC DC	1/4 1/4	f_{sys} f_{sys}
2	Cycle Time Master Slave	t_{qcyc}	4 4	510 —	t_{cyc} t_{cyc}
3	Enable Lead Time Master Slave	t_{lead}	2 2	128 —	t_{cyc} t_{cyc}
4	Enable Lag Time Master Slave	t_{lag}	— 2	1/2 —	SCK t_{cyc}
5	Clock (SCK) High or Low Time Master Slave[2]	t_{sw}	2 t_{cyc} – 60 2 t_{cyc} – n	255 t_{cyc} —	ns ns
6	Sequential Transfer Delay Master Slave (Does Not Require Deselect)	t_{td}	17 13	8192 —	t_{cyc} t_{cyc}
7	Data Setup Time (Inputs) Master Slave	t_{su}	20 20	— —	ns ns
8	Data Hold Time (Inputs) Master Slave	t_{hi}	30 20	— —	ns ns
9	Slave Access Time	t_a	—	1	t_{cyc}
10	Slave MISO Disable Time	t_{dis}	—	2	t_{cyc}
11	Data Valid (after SCK Edge) Master Slave	t_v	— —	50 50	ns ns
12	Data Hold Time (Outputs) Master Slave	t_{ho}	0 0	— —	ns ns
13	Rise Time Input Output	 t_{ri} t_{ro}	 — —	 2 30	 µs ns
14	Fall Time Input Output	 t_{fi} t_{fo}	 — —	 2 30	 µs ns

NOTES:
1. Refer to notes in **Table A-28**.

Freescale Semiconductor, Inc.

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

Table A-28 QSPI Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5% for 16.78 MHz, 10% for 20/25 MHz, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Function	Symbol	Min	Max	Unit
1	Operating Frequency Master Slave	f_{op}	DC DC	1/4 1/4	f_{sys} f_{sys}
2	Cycle Time Master Slave	t_{qcyc}	4 4	510 —	t_{cyc} t_{cyc}
3	Enable Lead Time Master Slave	t_{lead}	2 2	128 —	t_{cyc} t_{cyc}
4	Enable Lag Time Master Slave	t_{lag}	— 2	1/2 —	SCK t_{cyc}
5	Clock (SCK) High or Low Time Master Slave[2]	t_{sw}	2 t_{cyc} – 60 2 t_{cyc} – n	255 t_{cyc} —	ns ns
6	Sequential Transfer Delay Master Slave (Does Not Require Deselect)	t_{td}	17 13	8192 —	t_{cyc} t_{cyc}
7	Data Setup Time (Inputs) Master Slave	t_{su}	30 20	— —	ns ns
8	Data Hold Time (Inputs) Master Slave	t_{hi}	0 20	— —	ns ns
9	Slave Access Time	t_a	—	1	t_{cyc}
10	Slave MISO Disable Time	t_{dis}	—	2	t_{cyc}
11	Data Valid (after SCK Edge) Master Slave	t_v	— —	50 50	ns ns
12	Data Hold Time (Outputs) Master Slave	t_{ho}	0 0	— —	ns ns
13	Rise Time Input Output	 t_{ri} t_{ro}	 — —	 2 30	 µs ns
14	Fall Time Input Output	 t_{fi} t_{fo}	 — —	 2 30	 µs ns

NOTES:
1. All AC timing is shown with respect to V_{IH}/V_{IL} levels unless otherwise noted.
2. For high time, n = External SCK rise time; for low time, n = External SCK fall time.

Freescale Semiconductor, Inc.

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

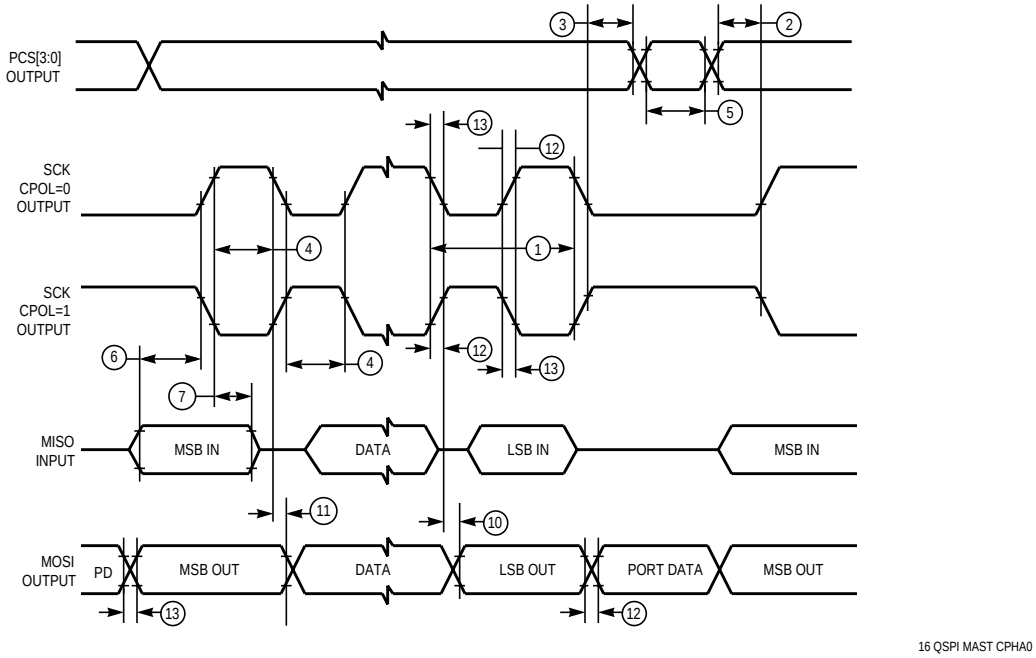


Figure A-16 QSPI Timing — Master, CPHA = 0

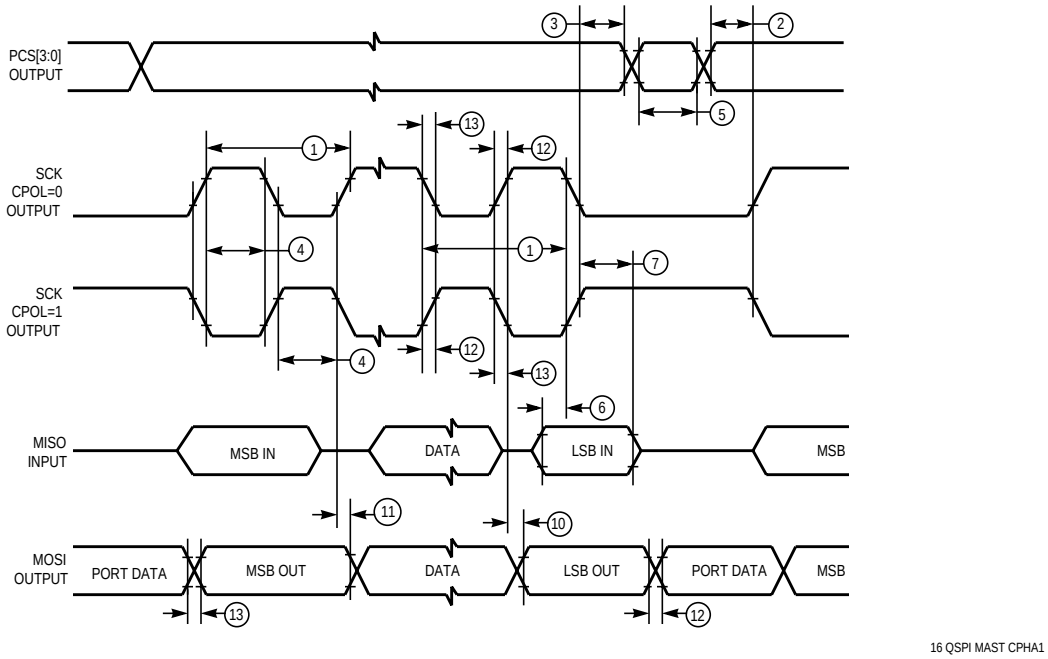


Figure A-17 QSPI Timing — Master, CPHA = 1

Freescale Semiconductor, Inc.

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

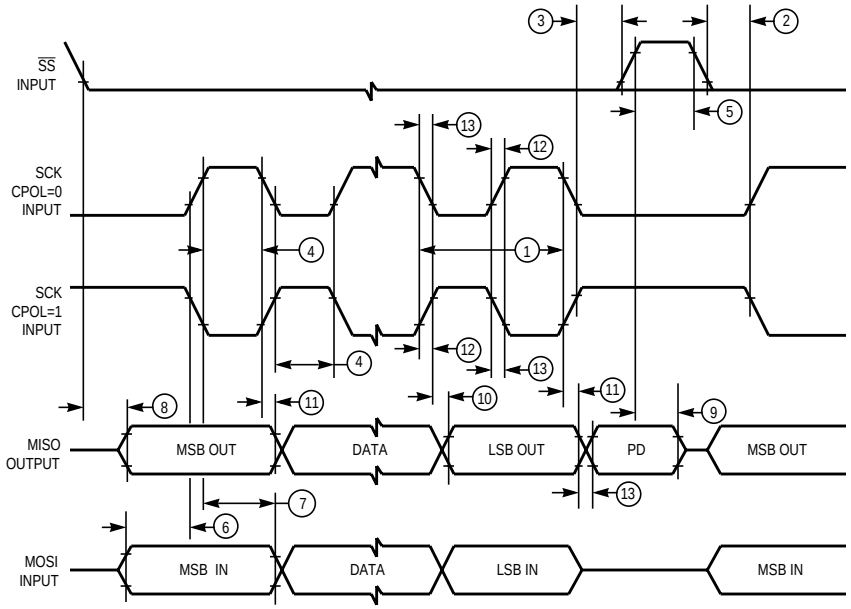


Figure A-18 QSPI Timing — Slave, CPHA = 0

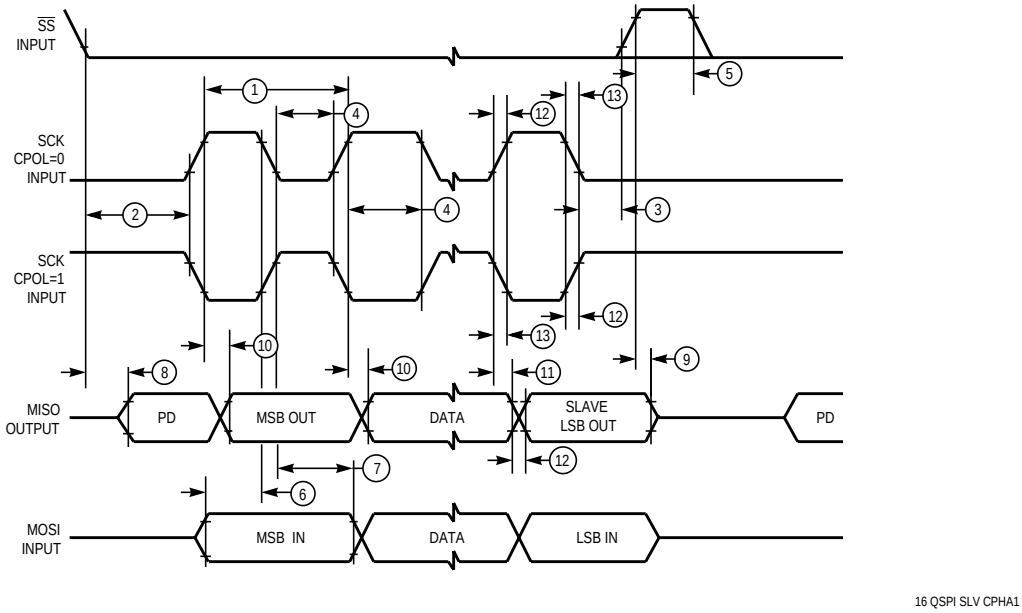


Figure A-19 QSPI Timing — Slave, CPHA = 1

Freescale Semiconductor, Inc.

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

Table A-29 Low Voltage SPI Timing

(V_{DD} and V_{DDSYN} = 2.7 to 3.6 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Function	Symbol	Min	Max	Unit
1	Operating Frequency Master Slave	f_{op}	DC DC	1/4 1/4	f_{sys} f_{sys}
2	Cycle Time Master Slave	t_{qcyc}	4 4	510 —	t_{cyc} t_{cyc}
3	Enable Lead Time Master Slave	t_{lead}	2 2	128 —	t_{cyc} t_{cyc}
4	Enable Lag Time Master Slave	t_{lag}	— 2	1/2 —	SCK t_{cyc}
5	Clock (SCK) High or Low Time Master Slave[2]	t_{sw}	2 t_{cyc} – 60 2 t_{cyc} – n	255 t_{cyc} —	ns ns
6	Sequential Transfer Delay Master Slave (Does Not Require Deselect)	t_{td}	17 13	8192 —	t_{cyc} t_{cyc}
7	Data Setup Time (Inputs) Master Slave	t_{su}	20 20	— —	ns ns
8	Data Hold Time (Inputs) Master Slave	t_{hi}	30 20	— —	ns ns
9	Slave Access Time	t_a	—	1	t_{cyc}
10	Slave MISO Disable Time	t_{dis}	—	2	t_{cyc}
11	Data Valid (after SCK Edge) Master Slave	t_v	— —	50 50	ns ns
12	Data Hold Time (Outputs) Master Slave	t_{ho}	0 0	— —	ns ns
13	Rise Time Input Output	 t_{ri} t_{ro}	 — —	 2 30	 µs ns
14	Fall Time Input Output	 t_{fi} t_{fo}	 — —	 2 30	 µs ns

NOTES:
1. Refer to notes in **Table A-30**.

Freescale Semiconductor, Inc.

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

Table A-30 SPI Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 10% for 16.78 MHz, 5% for 20/25 MHz, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Function	Symbol	Min	Max	Unit
1	Operating Frequency Master Slave	f_{op}	DC DC	1/4 1/4	f_{sys} f_{sys}
2	Cycle Time Master Slave	t_{qcyc}	4 4	510 —	t_{cyc} t_{cyc}
3	Enable Lead Time Master Slave	t_{lead}	2 2	128 —	t_{cyc} t_{cyc}
4	Enable Lag Time Master Slave	t_{lag}	— 2	1/2 —	SCK t_{cyc}
5	Clock (SCK) High or Low Time Master Slave[2]	t_{sw}	2 t_{cyc} – 60 2 t_{cyc} – n	255 t_{cyc} —	ns ns
6	Sequential Transfer Delay Master Slave (Does Not Require Deselect)	t_{td}	17 13	8192 —	t_{cyc} t_{cyc}
7	Data Setup Time (Inputs) Master Slave	t_{su}	30 20	— —	ns ns
8	Data Hold Time (Inputs) Master Slave	t_{hi}	0 20	— —	ns ns
9	Slave Access Time	t_a	—	1	t_{cyc}
10	Slave MISO Disable Time	t_{dis}	—	2	t_{cyc}
11	Data Valid (after SCK Edge) Master Slave	t_v	— —	50 50	ns ns
12	Data Hold Time (Outputs) Master Slave	t_{ho}	0 0	— —	ns ns
13	Rise Time Input Output	 t_{ri} t_{ro}	 — —	 2 30	 µs ns
14	Fall Time Input Output	 t_{fi} t_{fo}	 — —	 2 30	 µs ns

NOTES:
1. All AC timing is shown with respect to V_{IH}/V_{IL} levels unless otherwise noted.
2. For high time, n = External SCK rise time; for low time, n = External SCK fall time.

Freescale Semiconductor, Inc.

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

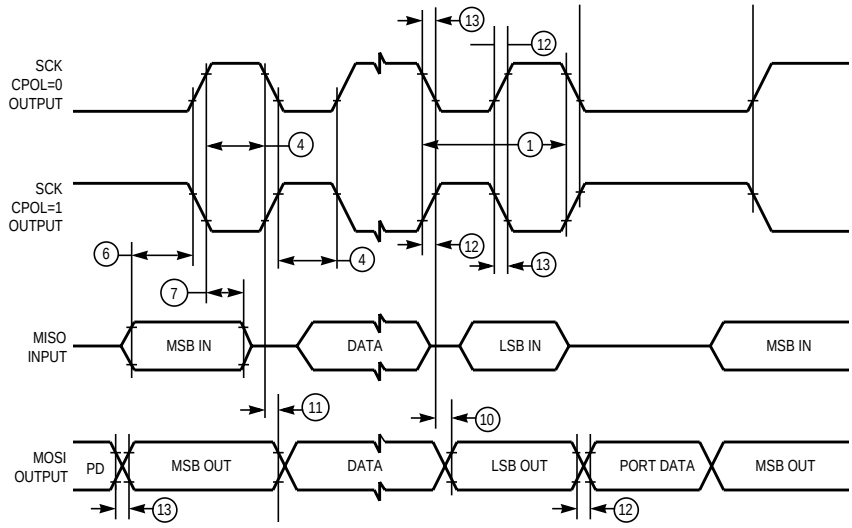


Figure A-20 SPI Timing — Master, CPHA = 0

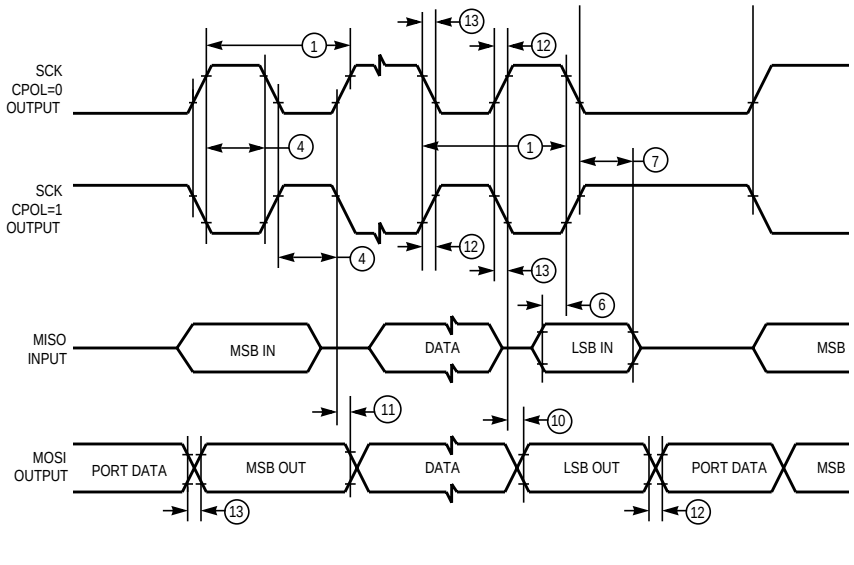


Figure A-21 SPI Timing — Master, CPHA = 1

Freescale Semiconductor, Inc.

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

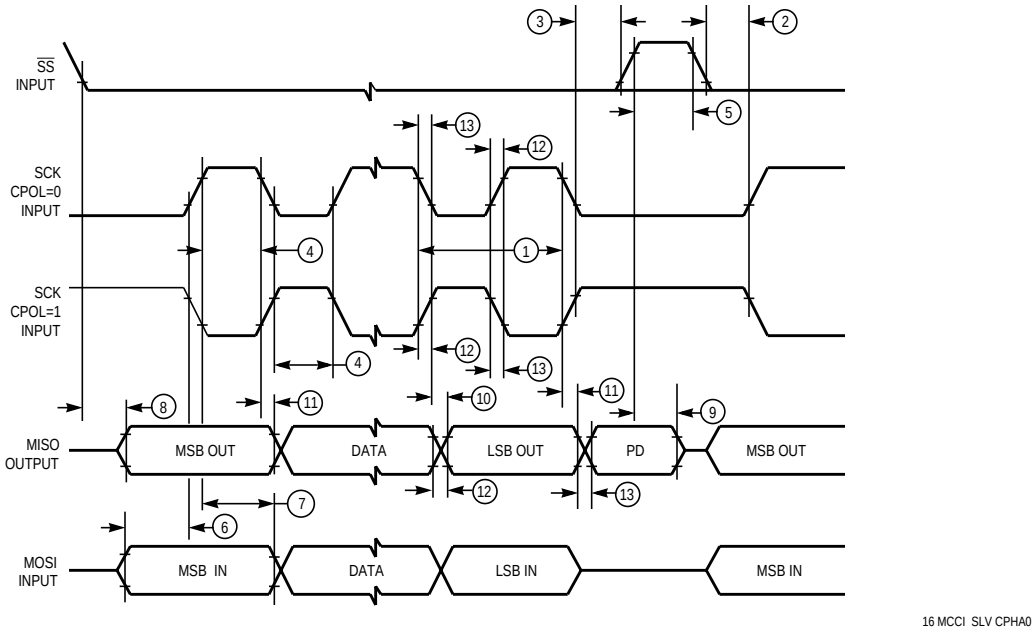


Figure A-22 SPI Timing — Slave, CPHA = 0

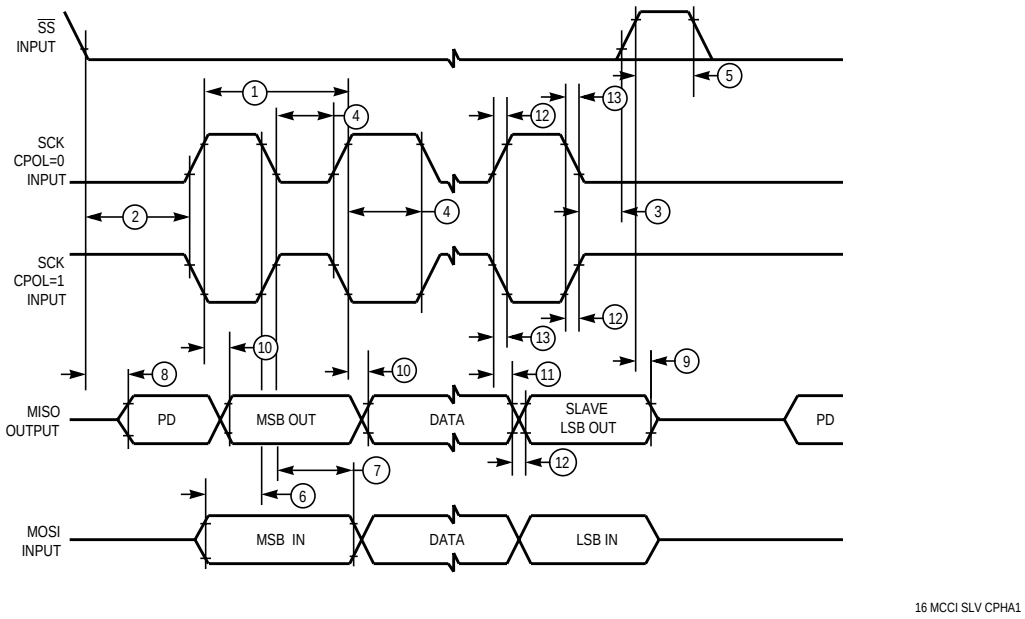


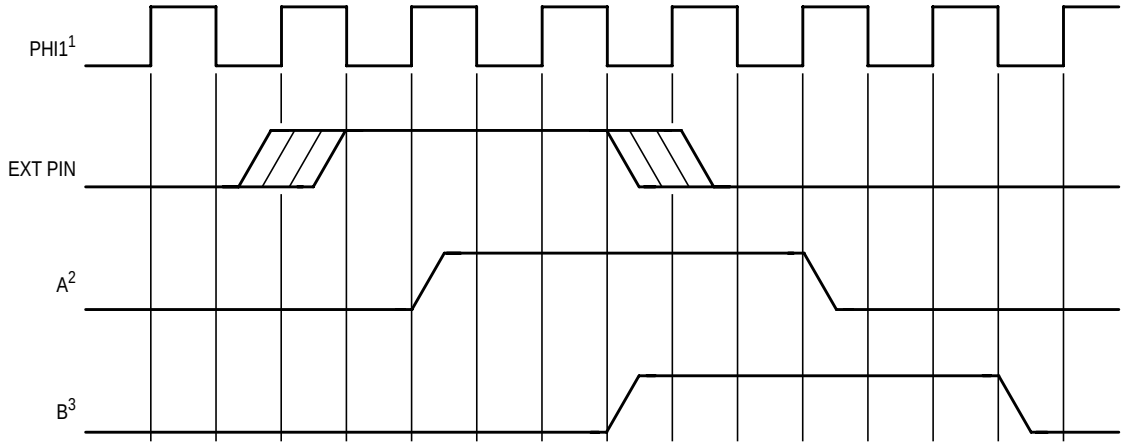
Figure A-23 SPI Timing — Slave, CPHA = 1

Freescale Semiconductor, Inc.

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

Table A-31 General-Purpose Timer AC Characteristics

Num	Parameter	Symbol	Min	Max	Unit
1	Operating Frequency	Fclock	0	16.78	MHz
2	PCLK Frequency	Fpclk	0	1/4 Fclock	MHz
3	Pulse Width Input Capture	PWtim	2/Fclock	—	—
4	PWM Resolution	—	2/Fclock	—	—
5	IC/OC Resolution	—	4/Fclock	—	—
6	PCLK Width (PWM)	—	4/Fclock	—	—
7	PCLK Width (IC/OC)	—	4/Fclock	—	—
8	PAI Pulse Width	—	2/Fclock	—	—



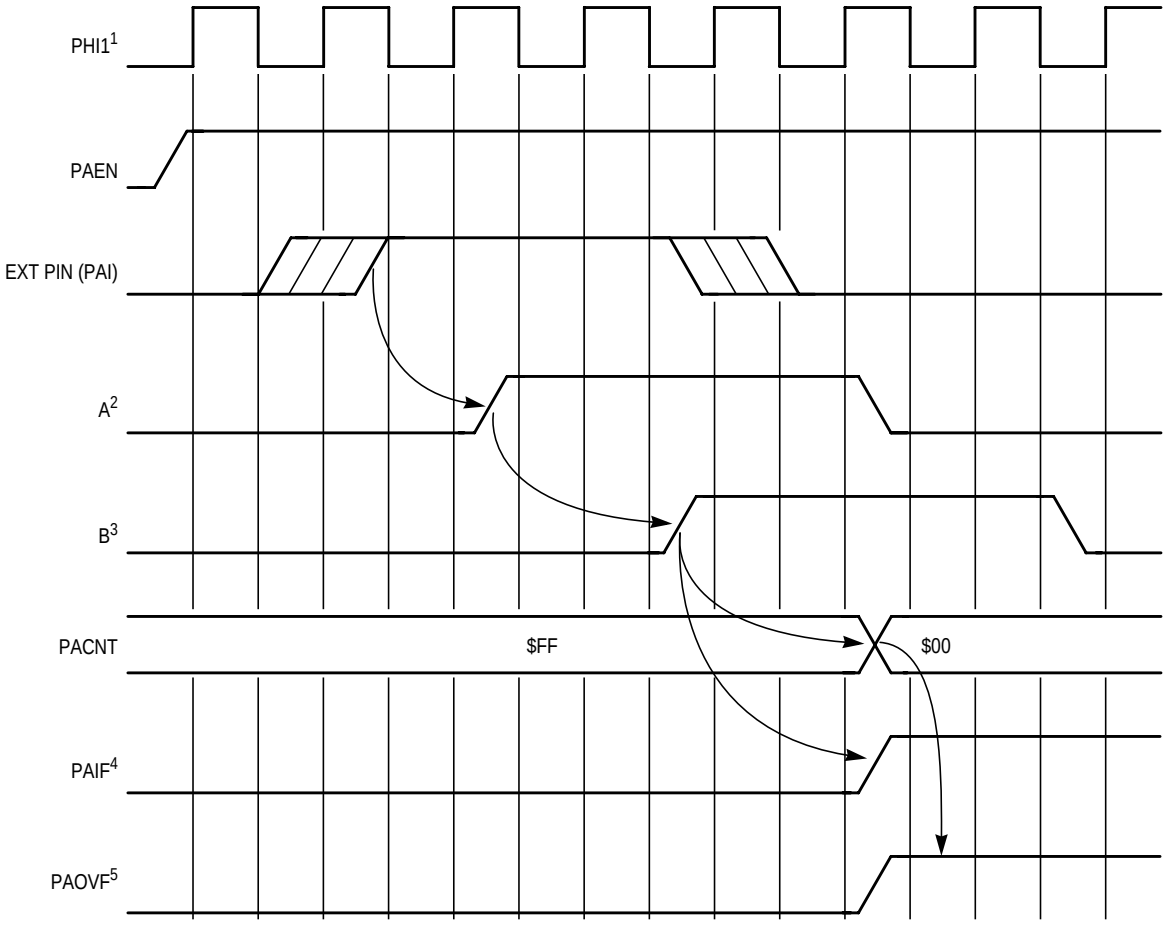
NOTES:
1. PHI1 IS THE SAME FREQUENCY AS THE SYSTEM CLOCK; HOWEVER, IT DOES NOT HAVE THE SAME TIMING.
2. A = INPUT SIGNAL AFTER THE SYNCHRONIZER.
3. B = “A” AFTER THE DIGITAL FILTER.

INPUT SIG CONDITIONER TIM

Figure A-24 Input Signal Conditioner Timing

Freescale Semiconductor, Inc.

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



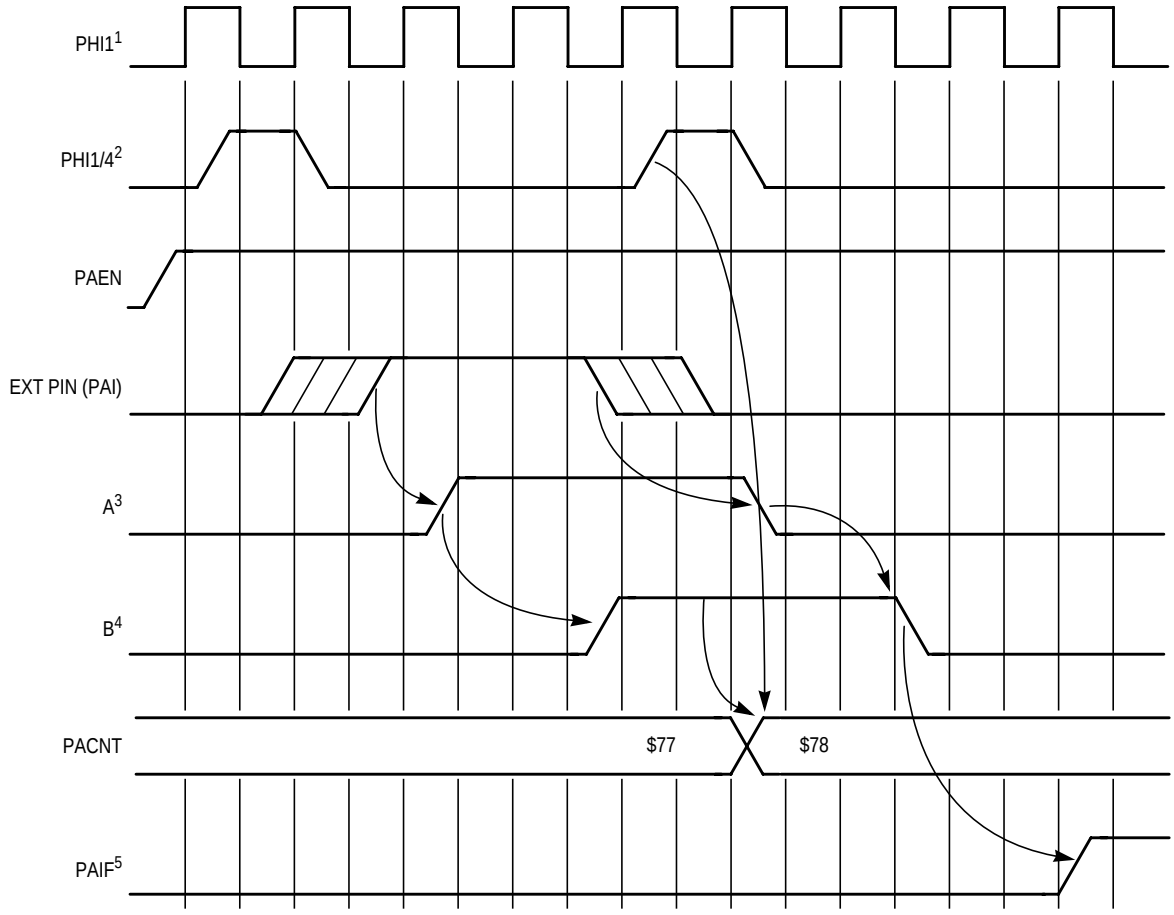
NOTES:
1. PHI1 IS THE SAME FREQUENCY AS THE SYSTEM CLOCK; HOWEVER, IT DOES NOT HAVE THE SAME TIMING.
2. A = PAI SIGNAL AFTER THE SYNCHRONIZER.
3. B = "A" AFTER THE DIGITAL FILTER.
4. THE EXTERNAL LEADING EDGE CAUSES THE PULSE ACCUMULATOR TO INCREMENT AND THE PAIF FLAG TO BE SET.
5. THE COUNTER TRANSITION FROM $FF TO $00 CAUSES THE PAOVF FLAG TO BE SET.

PULSE ACCUM ECM LEAD EDGE

Figure A-25 Pulse Accumulator — Event Counting Mode (Leading Edge)

Freescale Semiconductor, Inc.

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



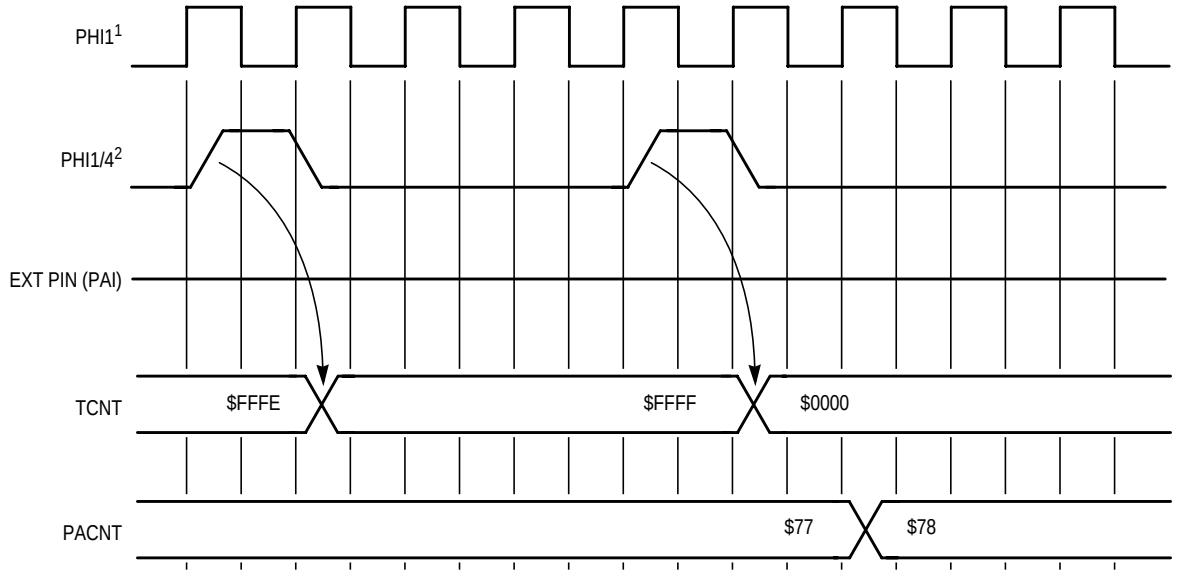
NOTES:
1. PHI1 HAS THE SAME FREQUENCY AS THE SYSTEM CLOCK; HOWEVER, IT DOES NOT HAVE THE SAME TIMING.
2. PHI1/4 CLOCKS PACNT WHEN GT-PAIF IS ASSERTED.
3. A = PAI SIGNAL AFTER THE SYNCHRONIZER.
4. B = "A" AFTER THE DIGITAL FILTER.
5. PAIF IS ASSERTED WHEN PAI IS NEGATED.

PULSE ACCUM GATED MODE

Figure A-26 Pulse Accumulator — Gated Mode (Count While Pin High)

Freescale Semiconductor, Inc.

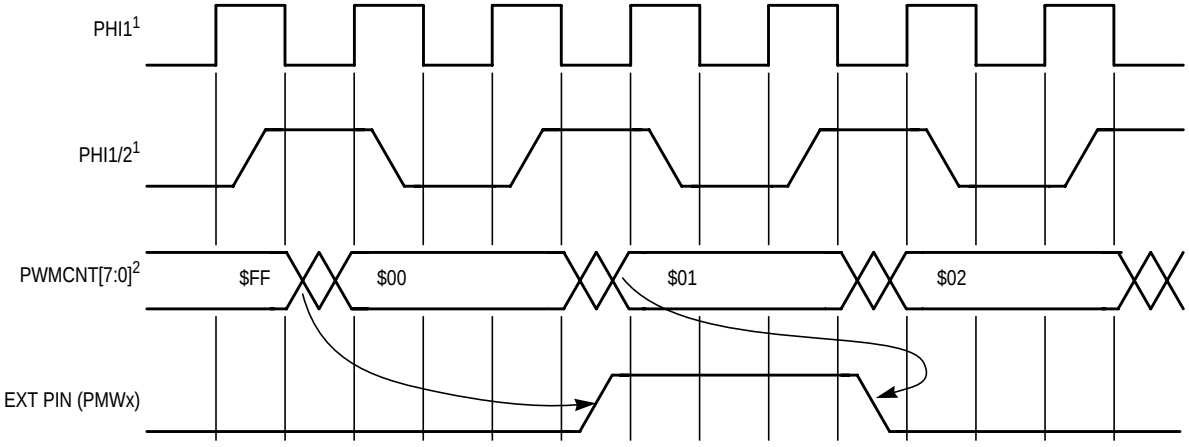
For More Information On This Product,
Go to: www.freescale.com



NOTES:
1. PHI1 HAS THE SAME FREQUENCY AS THE SYSTEM CLOCK; HOWEVER, IT DOES NOT HAVE THE SAME TIMING.
2. TCNT COUNTS AS A RESULT OF PHI1/4; PACNT COUNTS WHEN TCNT OVERFLOWS FROM $FFFF TO $0000 AND THE CONDITIONED PAI SIGNAL IS ASSERTED.

PULSE ACCUM TOF GATED MODE

Figure A-27 Pulse Accumulator — Using TOF as Gated Mode Clock



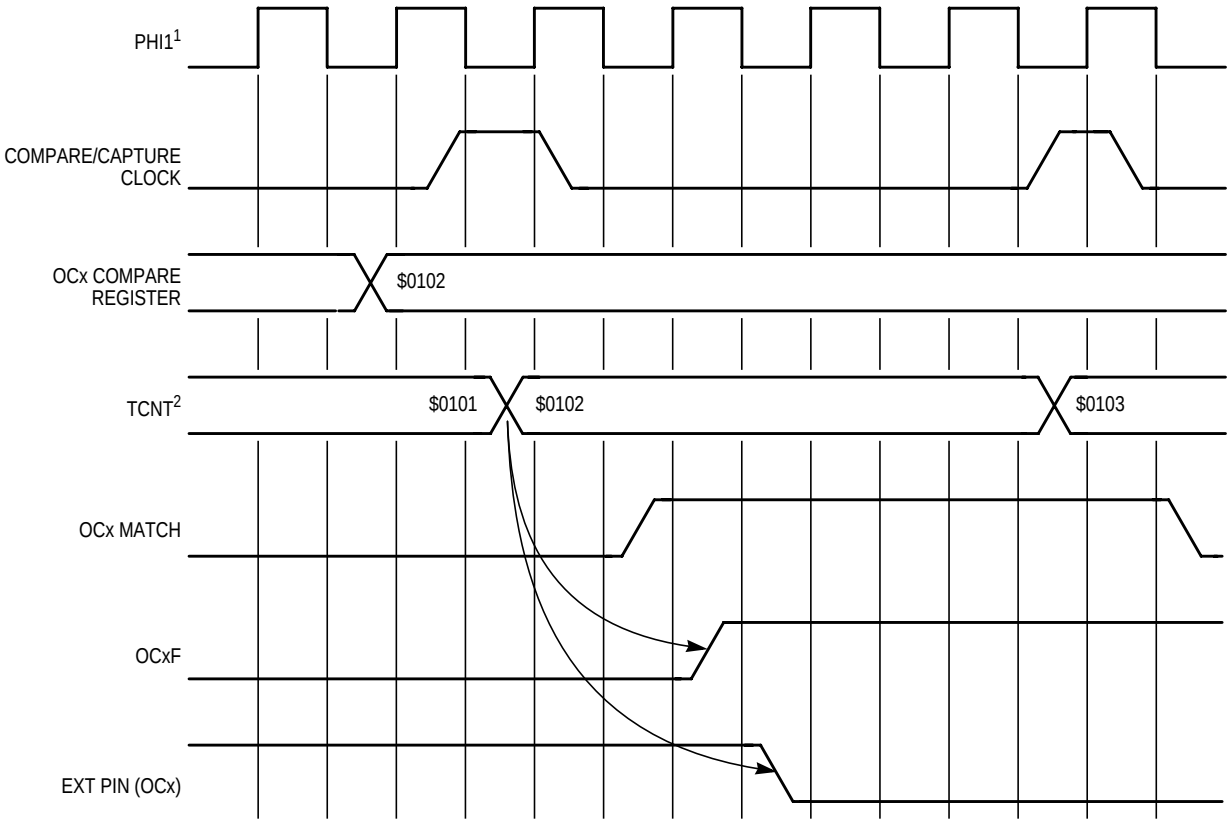
NOTES:
1. PHI1 IS THE SAME FREQUENCY AS THE SYSTEM CLOCK; HOWEVER, IT DOES NOT HAVE THE SAME TIMING.
2. WHEN THE COUNTER ROLLS OVER FROM $FF TO $00, THE PWM PIN IS SET TO LOGIC LEVEL ONE.
WHEN THE COUNTER EQUALS THE PWM REGISTER, THE PWM PIN IS CLEARED TO A LOGIC LEVEL ZERO.

PWMx FAST MODE

Figure A-28 PWMx (PWMx Register = 01, Fast Mode)

Freescale Semiconductor, Inc.

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



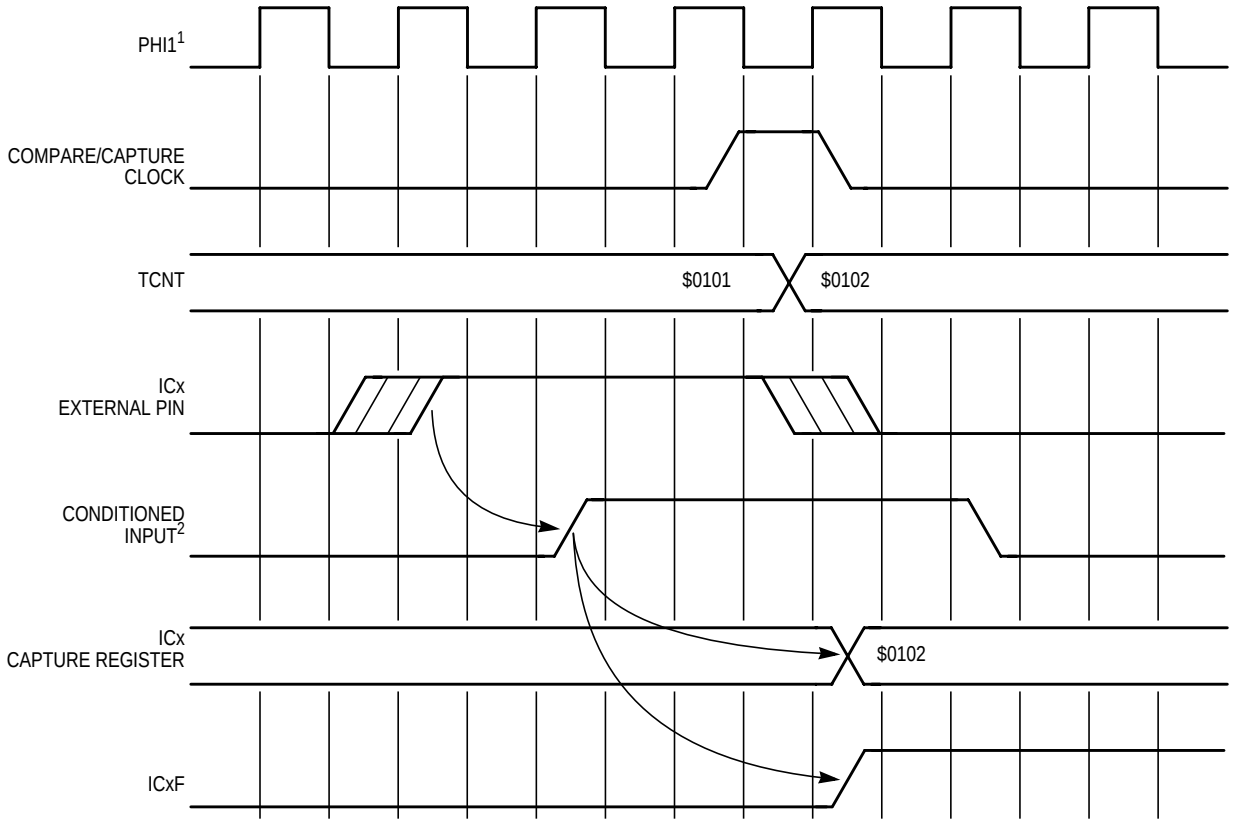
NOTES:
1. PHI1 IS THE SAME FREQUENCY AS THE SYSTEM CLOCK; HOWEVER, IT DOES NOT HAVE THE SAME TIMING.
2. WHEN THE TCNT MATCHES THE OCx COMPARE REGISTER, THE OCx FLAG IS SET FOLLOWED BY THE OCx PIN CHANGING STATE.

OUTPUT COMPARE

Figure A-29 Output Compare (Toggle Pin State)

Freescale Semiconductor, Inc.

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



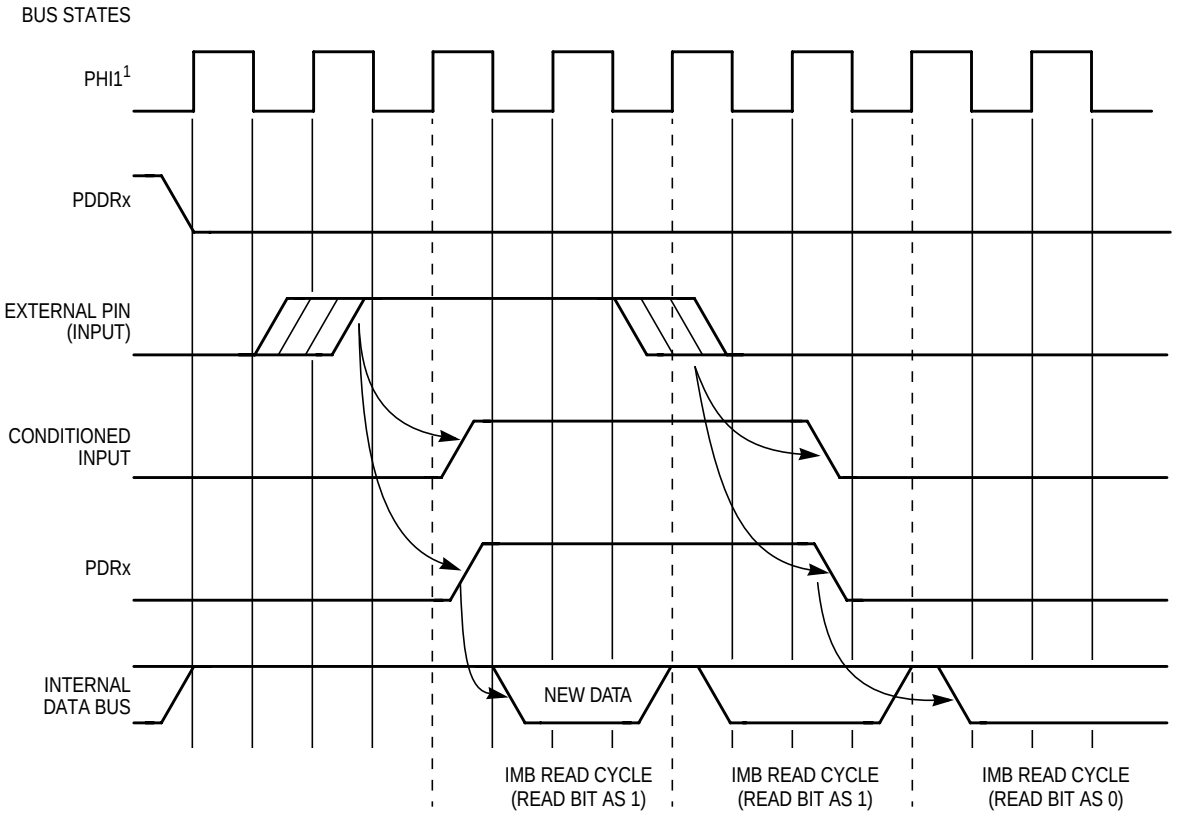
NOTES:
1. PHI1 IS THE SAME FREQUENCY AS THE SYSTEM CLOCK; HOWEVER, IT DOES NOT HAVE THE SAME TIMING.
2. THE CONDITIONED INPUT SIGNAL CAUSES THE CURRENT VALUE OF THE TCNT TO BE LATCHED BY THE ICx CAPTURE REGISTER. THE ICxF FLAG IS SET AT THE SAME TIME.

INPUT CAPTURE

Figure A-30 Input Capture (Capture on Rising Edge)

Freescale Semiconductor, Inc.

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



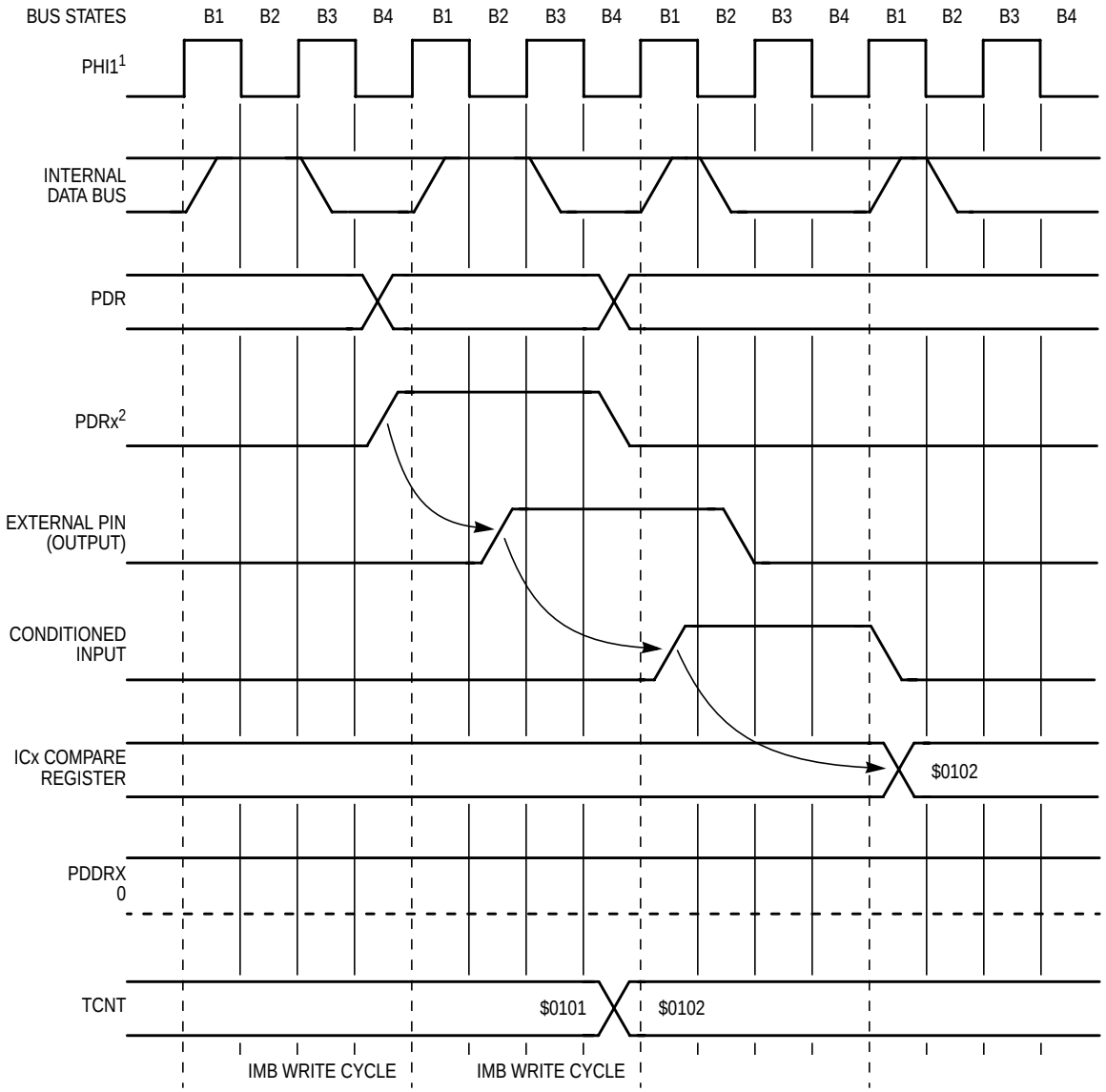
NOTES:
1. PHI1 IS THE SAME FREQUENCY AS THE SYSTEM CLOCK; HOWEVER, IT DOES NOT HAVE THE SAME TIMING.

GENERAL PURPOSE INPUT

Figure A-31 General-Purpose Input

Freescale Semiconductor, Inc.

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



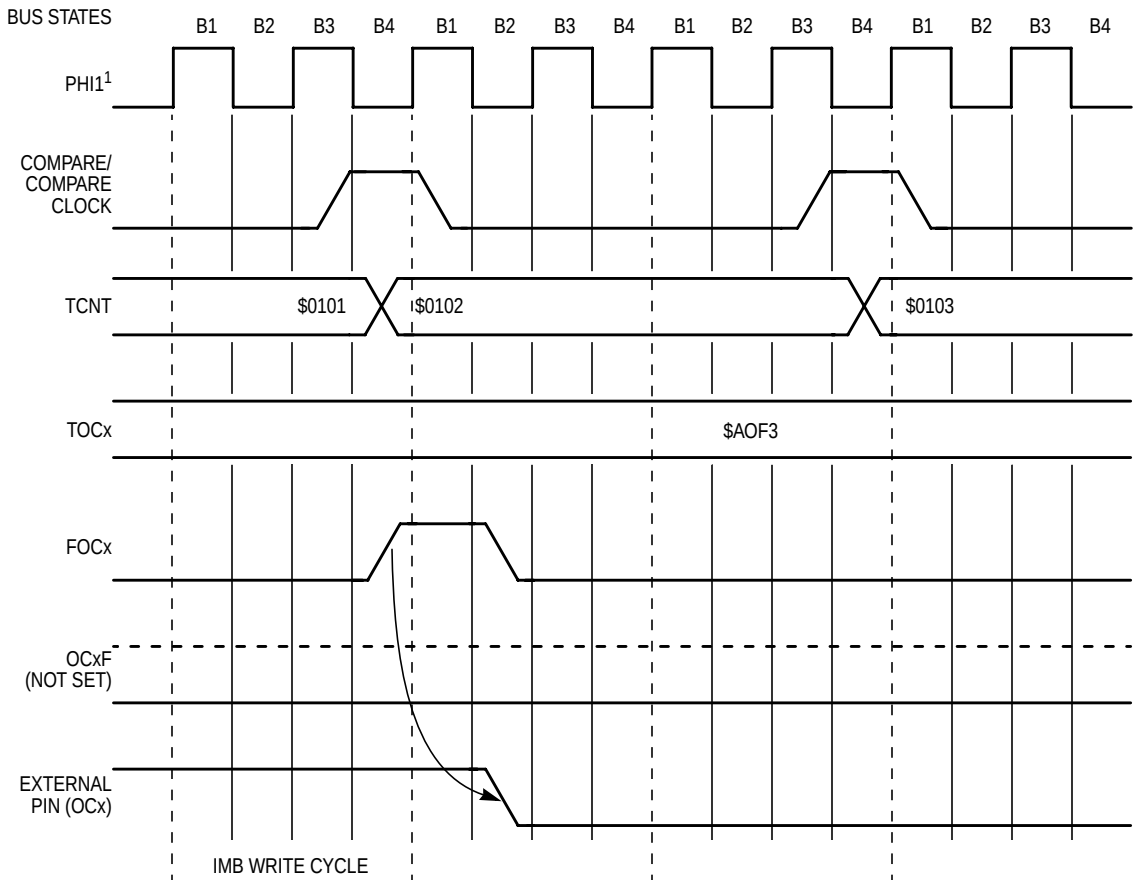
NOTES:
1. PHI1 IS THE SAME FREQUENCY AS THE SYSTEM CLOCK; HOWEVER, IT DOES NOT HAVE THE SAME TIMING.
2. WHEN THE BIT VALUE IS DRIVEN ON THE PIN, THE INPUT CIRCUIT SEES THE SIGNAL. AFTER IT IS CONDITIONED, IT CAUSES THE CONTENTS OF THE TCNT TO BE LATCHED INTO THE ICx COMPARE REGISTER.

GENERAL PURPOSE OUTPUT

Figure A-32 General-Purpose Output (Causes Input Capture)

Freescale Semiconductor, Inc.

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



NOTES:
1. PHI1 IS THE SAME FREQUENCY AS THE SYSTEM CLOCK; HOWEVER, IT DOES NOT HAVE THE SAME TIMING.

FORCE COMPARE

Figure A-33 Force Compare (CLEAR)

Freescale Semiconductor, Inc.

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

Table A-32 ADC Maximum Ratings

Num	Parameter	Symbol	Min	Max	Unit
1	Analog Supply	V_{DDA}	–0.3	6.5	V
2	Internal Digital Supply, with reference to V_{SSI}	V_{DDI}	–0.3	6.5	V
3	Reference Supply, with reference to V_{SSI}	V_{RH}, V_{RL}	–0.3	6.5	V
4	V_{SS} Differential Voltage	$V_{SSI} - V_{SSA}$	–0.1	0.1	V
5	V_{DD} Differential Voltage	$V_{DDI} - V_{DDA}$	–6.5	6.5	V
6	V_{REF} Differential Voltage	$V_{RH} - V_{RL}$	–6.5	6.5	V
7	V_{RH} to V_{DDA} Differential Voltage	$V_{RH} - V_{DDA}$	–6.5	6.5	V
8	V_{RL} to V_{SSA} Differential Voltage	$V_{RL} - V_{SSA}$	–6.5	6.5	V
9	Disruptive Input Current[1, 2, 3, 4, 5, 6, 7] $V_{NEGCLAMP} \cong -0.3$ V $V_{POSCLAMP} \cong 8$ V	I_{NA}	–500	500	µA
10	Positive Overvoltage Current Coupling Ratio[1,5,6,8]	K_P	2000	—	—
11	Negative Overvoltage Current Coupling Ratio[1,5,6,8]	K_N	500	—	—
12	Maximum Input Current [3,4,6] $V_{NEGCLAMP} \cong -0.3$ V $V_{POSCLAMP} \cong 8$ V	I_{MA}	–25	25	mA

NOTES:

1. Below disruptive current conditions, a stressed channel will store the maximum conversion value for analog inputs greater than V_{RH} and the minimum conversion value for inputs less than V_{RL}. This assumes that $V_{RH} \leq V_{DDA}$ and $V_{RL} \geq V_{SSA}$ due to the presence of the sample amplifier. Other channels are not affected by non-disruptive conditions.
2. Input signals with large slew rates or high frequency noise components cannot be converted accurately. These signals also interfere with conversion of other channels.
3. Exceeding limit may cause conversion error on stressed channels and on unstressed channels. Transitions within the limit do not affect device reliability or cause permanent damage.
4. Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values using positive and negative clamp values, then use the larger of the calculated values.
5. This parameter is periodically sampled rather than 100% tested.
6. Applies to single pin only.
7. The values of external system components can change the maximum input current value, and affect operation. A voltage drop may occur across the external source impedances of the adjacent pins, impacting conversions on these adjacent pins. The actual maximum may need to be determined by testing the complete design.
8. Current coupling is the ratio of the current induced from overvoltage (positive or negative, through an external series coupling resistor), divided by the current induced on adjacent pins. A voltage drop may occur across the external source impedances of the adjacent pins, impacting conversions on these adjacent pins.

Freescale Semiconductor, Inc.

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

Table A-33 Low Voltage ADC DC Electrical Characteristics (Operating)

(V_{SS} = 0 Vdc, ADCLK = 1.05 MHz, T_A within operating temperature range)

Num	Parameter	Symbol	Min	Max	Unit
1	Analog Supply[1]	V_{DDA}	2.7	3.6	V
2	Internal Digital Supply[1]	V_{DDI}	2.7	3.6	V
3	V_{SS} Differential Voltage	$V_{SSI} - V_{SSA}$	– 1.0	1.0	mV
4	V_{DD} Differential Voltage	$V_{DDI} - V_{DDA}$	– 0.6	0.6	V
5	Reference Voltage Low[2, 3]	V_{RL}	V_{SSA}	V_{DDA} / 2	V
6	Reference Voltage High[2, 3]	V_{RH}	V_{DDA} / 2	V_{DDA}	V
7	V_{REF} Differential Voltage[3]	$V_{RH} - V_{RL}$	2.7	3.6	V
8	Input Voltage[2]	V_{INDC}	V_{SSA}	V_{DDA}	V
9	Input High, Port ADA	V_{IH}	0.7 (V_{DDA})	V_{DDA} + 0.3	V
10	Input Low, Port ADA	V_{IL}	V_{SSA} – 0.3	0.2 (V_{DDA})	V
11	Analog Supply Current Normal Operation[4] Low-Power Stop	I_{DDA}	 — —	 1.0 200	 mA µA
12	Reference Supply Current	I_{REF}	—	120	µA
13	Input Current, Off Channel[5]	I_{OFF}	—	150	nA
14	Total Input Capacitance, Not Sampling	C_{INN}	—	10	pF
15	Total Input Capacitance, Sampling	C_{INS}	—	15	pF

NOTES:

1. Refers to operation over full temperature and frequency range.
2. To obtain full-scale, full-range results, $V_{SSA} \le V_{RL} \le V_{INDC} \le V_{RH} \le V_{DDA}$.
3. Accuracy tested and guaranteed at $V_{RH} - V_{RL}$ = 2.7 V ± 3.6 Vdc.
4. Current measured at maximum system clock frequency with ADC active.
5. Maximum leakage occurs at maximum operating temperature. Current decreases by approximately one-half for each 10°C decrease from maximum temperature.

Table A-34 Low Voltage ADC AC Characteristics (Operating)

(V_{DD} and V_{DDA} = 2.7 to 3.6 Vdc, V_{SS} = 0 Vdc, T_A within operating temperature range)

Num	Parameter	Symbol	Min	Max	Unit
1	ADC Clock Frequency	f_{ADCLK}	0.5	1.05	MHz
2	8-Bit Conversion Time[1] f_{ADCLK} = 1.05 MHz	t_{CONV}	15.2	—	µs
3	10-Bit Conversion Time[1] f_{ADCLK} = 1.05 MHz	t_{CONV}	17.1	—	µs
4	Stop Recovery Time	t_{SR}	—	50	µs

NOTES:

1. Conversion accuracy varies with f_{ADCLK} rate. Reduced conversion accuracy occurs at maximum.

Freescale Semiconductor, Inc.

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

Table A-35 5V ADC DC Electrical Characteristics (Operating)

(V_{SS} = 0 Vdc, ADCLK = 2.1 MHz, $T_A = T_L$ to T_H)

Num	Parameter	Symbol	Min	Max	Unit
1	Analog Supply[1]	V_{DDA}	4.5	5.5	V
2	Internal Digital Supply[1]	V_{DDI}	4.5	5.5	V
3	V_{SS} Differential Voltage	$V_{SSI} - V_{SSA}$	– 1.0	1.0	mV
4	V_{DD} Differential Voltage	$V_{DDI} - V_{DDA}$	– 1.0	1.0	V
5	Reference Voltage Low[2,3]	V_{RL}	V_{SSA}	V_{DDA} / 2	V
6	Reference Voltage High[2,3]	V_{RH}	V_{DDA} / 2	V_{DDA}	V
7	V_{REF} Differential Voltage[3]	$V_{RH} - V_{RL}$	4.5	5.5	V
8	Input Voltage[2]	V_{INDC}	V_{SSA}	V_{DDA}	V
9	Input High, Port ADA	V_{IH}	0.7 (V_{DDA})	V_{DDA} + 0.3	V
10	Input Low, Port ADA	V_{IL}	V_{SSA} – 0.3	0.2 (V_{DDA})	V
11	Analog Supply Current Normal Operation[4] Low-Power Stop	I_{DDA}	 — —	 1.0 200	 mA µA
12	Reference Supply Current	I_{REF}	—	250	µA
13	Input Current, Off Channel[5]	I_{OFF}	—	150	nA
14	Total Input Capacitance, Not Sampling	C_{INN}	—	10	pF
15	Total Input Capacitance, Sampling	C_{INS}	—	15	pF

NOTES:

1. Refers to operation over full temperature and frequency range.
2. To obtain full-scale, full-range results, $V_{SSA} \le V_{RL} \le V_{INDC} \le V_{RH} \le V_{DDA}$.
3. Accuracy tested and guaranteed at $V_{RH} - V_{RL}$ = 5.0 V ± 5% for 20/25 MHz, ± 10% for 16 MHz.
4. Current measured at maximum system clock frequency with ADC active.
5. Maximum leakage occurs at maximum operating temperature. Current decreases by approximately one-half for each 10°C decrease from maximum temperature.

Freescale Semiconductor, Inc.

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

Table A-36 ADC AC Characteristics (Operating)

(V_{DD} and V_{DDA} = 5.0 Vdc ± 5% for 20/25 MHz, ± 10% for 16 MHz, V_{SS} = 0 Vdc, T_A within operating temperature range)

Num	Parameter	Symbol	Min	Max	Unit
1	ADC Clock Frequency	f_{ADCLK}	0.5	2.1	MHz
2	8-Bit Conversion Time[1] f_{ADCLK} = 1.0 MHz f_{ADCLK} = 2.1 MHz	t_{CONV}	15.2 7.6	—	µs
3	10-Bit Conversion Time[1] f_{ADCLK} = 1.0 MHz f_{ADCLK} = 2.1 MHz	t_{CONV}	17.1 8.6	—	µs
4	Stop Recovery Time	t_{SR}	—	10	µs

NOTES:
1. Conversion accuracy varies with f_{ADCLK} rate. Reduced conversion accuracy occurs at maximum.

Freescale Semiconductor, Inc.

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

Table A-37 Low Voltage ADC Conversion Characteristics (Operating)

(V_{DD} and V_{DDA} = 2.7 to 3.6 Vdc , V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, f_{ADCLK} = 1.05 MHz)

Num	Parameter	Symbol	Min	Typical	Max	Unit
1	8-Bit Resolution[1]	1 Count	—	12	—	mV
2	8-Bit Differential Nonlinearity	DNL	–0.5	—	0.5	Counts
3	8-Bit Integral Nonlinearity	INL	–1.0	—	1.0	Counts
4	8-Bit Absolute Error[2]	AE	–1.5	—	1.5	Counts
5	10-Bit Resolution[1]	1 Count	—	3	—	mV
6	10-Bit Differential Nonlinearity[3]	DNL	–1	—	1	Counts
7	10-Bit Integral Nonlinearity[3]	INL	–2.0	—	2.0	Counts
8	10-Bit Absolute Error[3,4]	AE	–4	—	4.0	Counts
9	Source Impedance at Input[5]	R_S	—	20	—	kΩ

NOTES:

1. At $V_{RH} - V_{RL}$= 3.072 V, one 10-bit count = 3 mV and one 8-bit count = 12 mV.
2. 8-bit absolute error of 1.5 counts (18 mV) includes 1/2 count (6 mV) inherent quantization error and 1 count (12 mV) circuit (differential, integral, and offset) error.
3. Conversion accuracy varies with f_{ADCLK} rate. Reduced conversion accuracy occurs at maximum f_{ADCLK}. Assumes that minimum sample time (2 ADC clocks) is selected.
4. 10-bit absolute error of 4.0 counts (12 mV) includes 1/2 count (1.5 mV) inherent quantization error and 3.5 counts (10.5 mV) circuit (differential, integral, and offset) error.
5. Maximum source impedance is application-dependent. Error resulting from pin leakage depends on junction leakage into the pin and on leakage due to charge-sharing with internal capacitance.
   Error from junction leakage is a function of external source impedance and input leakage current. Expected error in result value due to junction leakage is expressed in voltage (V_{ERRJ}):

$$V_{ERRJ} = R_S \times I_{OFF}$$

   where I_{OFF} is a function of operating temperature, as shown in **Table A-35**.
   Charge-sharing leakage is a function of input source impedance, conversion rate, change in voltage between successive conversions, and the size of the decoupling capacitor used. Error levels are best determined empirically. In general, continuous conversion of the same channel may not be compatible with high source impedance.

Freescale Semiconductor, Inc.

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

Table A-38 ADC Conversion Characteristics (Operating)

(V_{DD} and V_{DDA} = 5.0 Vdc ± 5% for 20/25 MHz, ± 10% for 16 MHz, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, 0.5 MHz ≤ f_{ADCLK} ≤ 1.0 MHz, 2 clock input sample time)

Num	Parameter	Symbol	Min	Typical	Max	Unit
1	8-Bit Resolution[1]	1 Count	—	20	—	mV
2	8-Bit Differential Nonlinearity	DNL	–0.5	—	0.5	Counts
3	8-Bit Integral Nonlinearity	INL	–1	—	1	Counts
4	8-Bit Absolute Error[2]	AE	–1	—	1	Counts
5	10-Bit Resolution[1]	1 Count	—	5	—	mV
6	10-Bit Differential Nonlinearity[3]	DNL	–1	—	1	Counts
7	10-Bit Integral Nonlinearity[3]	INL	–2.0	—	2.0	Counts
8	10-Bit Absolute Error[3,4]	AE	–2.5	—	2.5	Counts
9	Source Impedance at Input[5]	R_S	—	20	—	kΩ

NOTES:

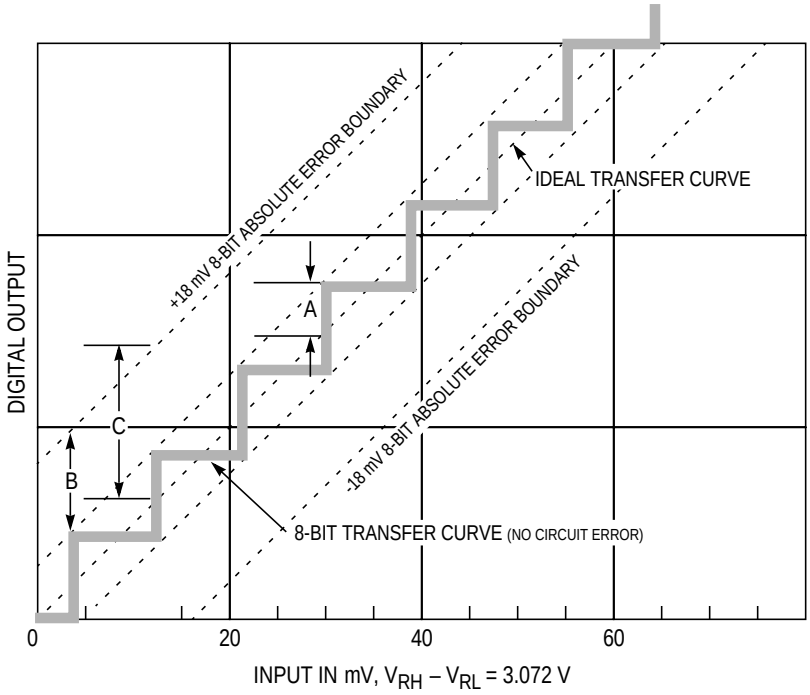
1. At $V_{RH} - V_{RL}$ = 5.12 V, one 10-bit count = 5 mV and one 8-bit count = 20 mV.
2. 8-bit absolute error of 1 count (20 mV) includes 1/2 count (10 mV) inherent quantization error and 1/2 count (10 mV) circuit (differential, integral, and offset) error.
3. Conversion accuracy varies with f_{ADCLK} rate. Reduced conversion accuracy occurs at maximum f_{ADCLK}. Assumes that minimum sample time (2 ADC clocks) is selected.
4. 10-bit absolute error of 2.5 counts (12.5 mV) includes 1/2 count (2.5 mV) inherent quantization error and 2 counts (10 mV) circuit (differential, integral, and offset) error.
5. Maximum source impedance is application-dependent. Error resulting from pin leakage depends on junction leakage into the pin and on leakage due to charge-sharing with internal capacitance.
   Error from junction leakage is a function of external source impedance and input leakage current. Expected error in result value due to junction leakage is expressed in voltage (V_{ERRJ}):

$$V_{ERRJ} = R_S \times I_{OFF}$$

   where I_{OFF} is a function of operating temperature, as shown in **Table A-35**.
   Charge-sharing leakage is a function of input source impedance, conversion rate, change in voltage between successive conversions, and the size of the decoupling capacitor used. Error levels are best determined empirically. In general, continuous conversion of the same channel may not be compatible with high source impedance.

Freescale Semiconductor, Inc.

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



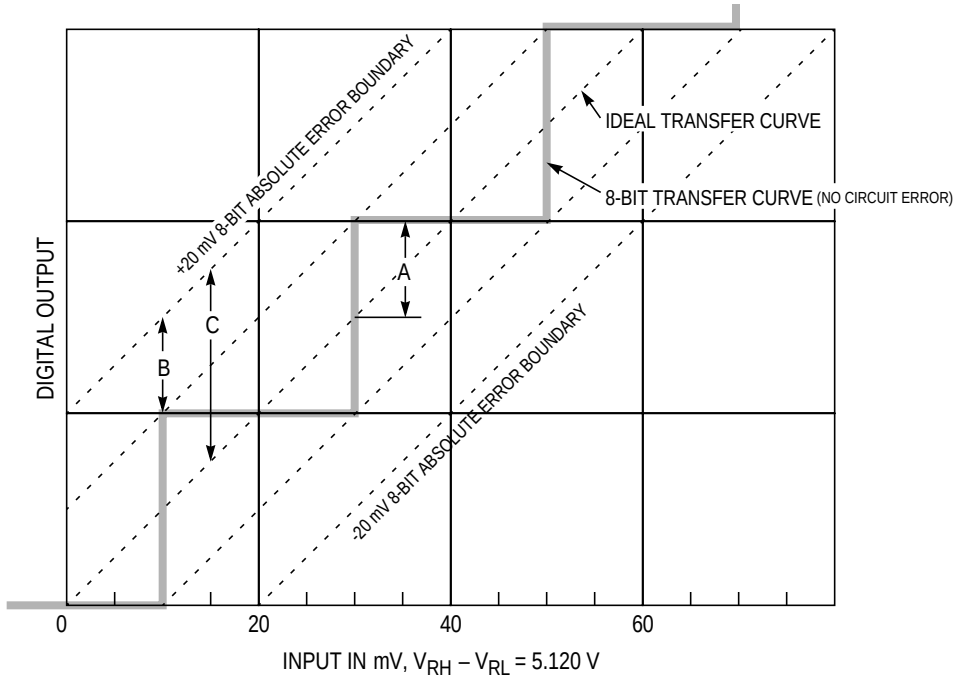
A – +1/2 COUNT (6 mV) INHERENT QUANTIZATION ERROR
B – CIRCUIT-CONTRIBUTED +12 mV ERROR
C – +18 mV ABSOLUTE ERROR (1.5 8-BIT COUNTS)

ADC 8-BIT ACCURACY LV

Figure A-34 Low Voltage 8-Bit ADC Conversion Accuracy

Freescale Semiconductor, Inc.

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



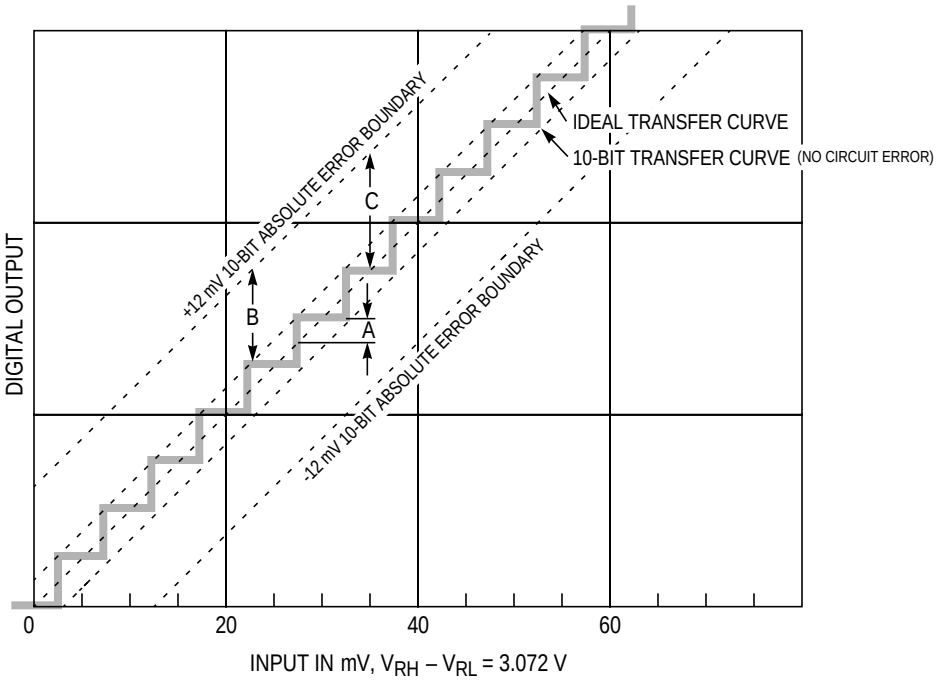
A – +1/2 COUNT (10 mV) INHERENT QUANTIZATION ERROR
B – CIRCUIT-CONTRIBUTED +10 mV ERROR
C – + 20 mV ABSOLUTE ERROR (ONE 8-BIT COUNT)

ADC 8-BIT ACCURACY

Figure A-35 8-Bit ADC Conversion Accuracy

Freescale Semiconductor, Inc.

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



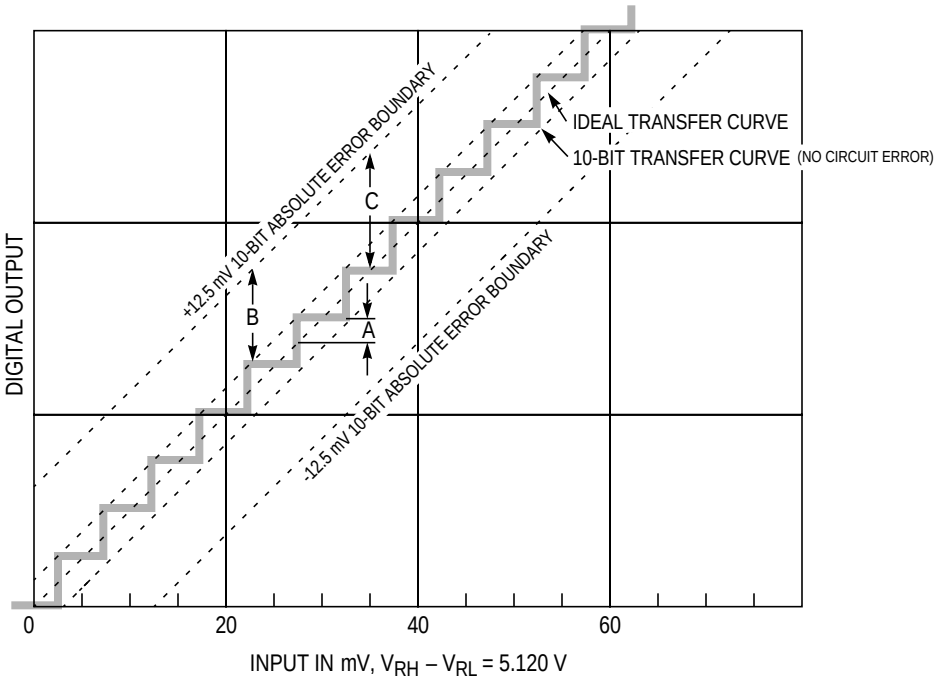
A – +.5 COUNT (1.5 mV) INHERENT QUANTIZATION ERROR
B – CIRCUIT-CONTRIBUTED +10.5 mV ERROR
C – +12 mV ABSOLUTE ERROR (4 10-BIT COUNTS)

ADC 10-BIT ACCURACY LV

Figure A-36 Low Voltage 10-Bit ADC Conversion Accuracy

Freescale Semiconductor, Inc.

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



A – +.5 COUNT (2.5 mV) INHERENT QUANTIZATION ERROR
B – CIRCUIT-CONTRIBUTED +10 mV ERROR
C – +12.5 mV ABSOLUTE ERROR (2.5 10-BIT COUNTS)

ADC 10-BIT ACCURACY

Figure A-37 10-Bit ADC Conversion Accuracy

Freescale Semiconductor, Inc.

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

Freescale Semiconductor, Inc.

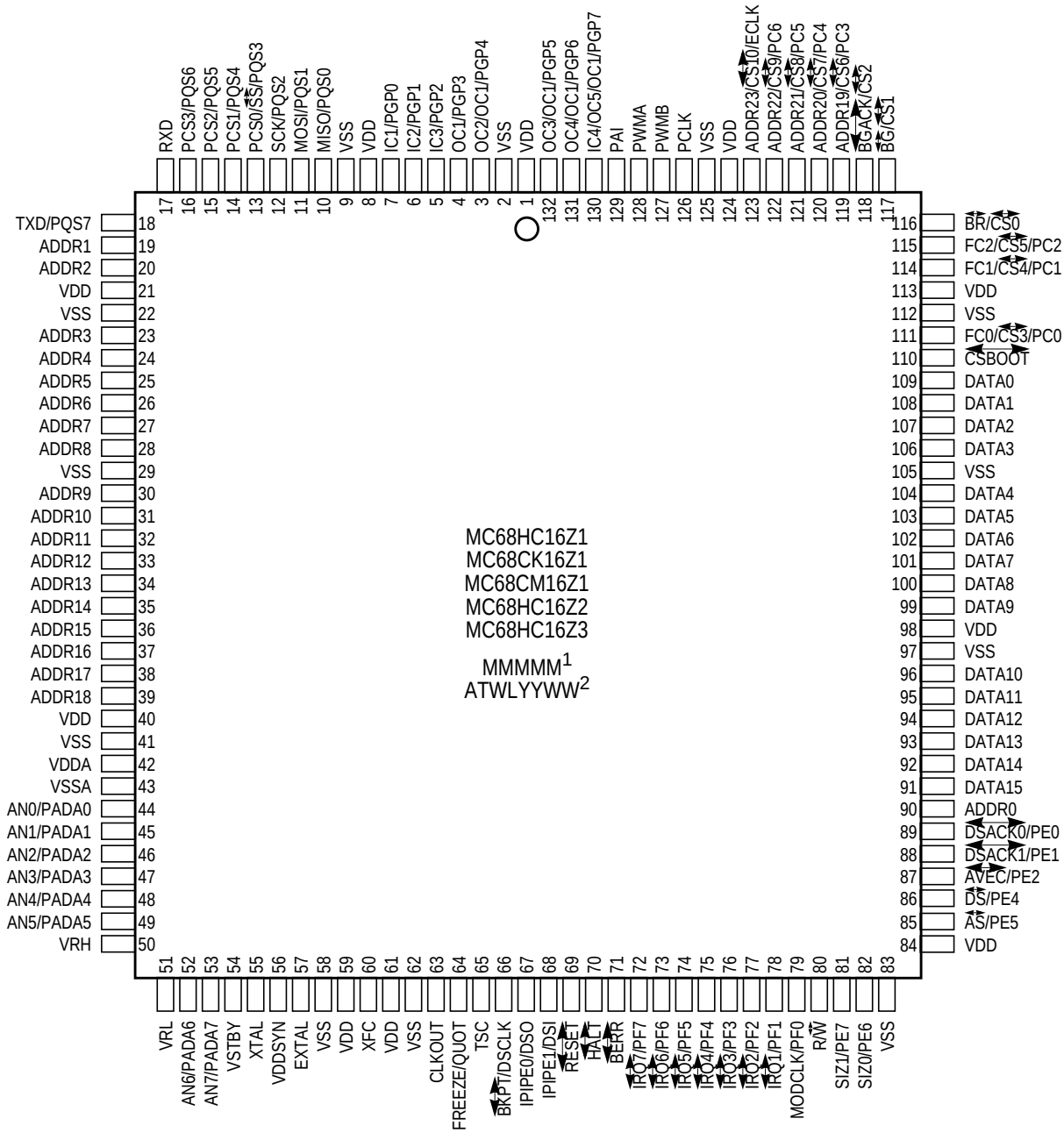
For More Information On This Product,
Go to: www.freescale.com

APPENDIX B
MECHANICAL DATA AND ORDERING INFORMATION

M68HC16 Z-series microcontrollers are available in both 132- and 144-pin packages. This appendix provides package pin assignment drawings, dimensional drawings, and ordering information.

Freescale Semiconductor, Inc.

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

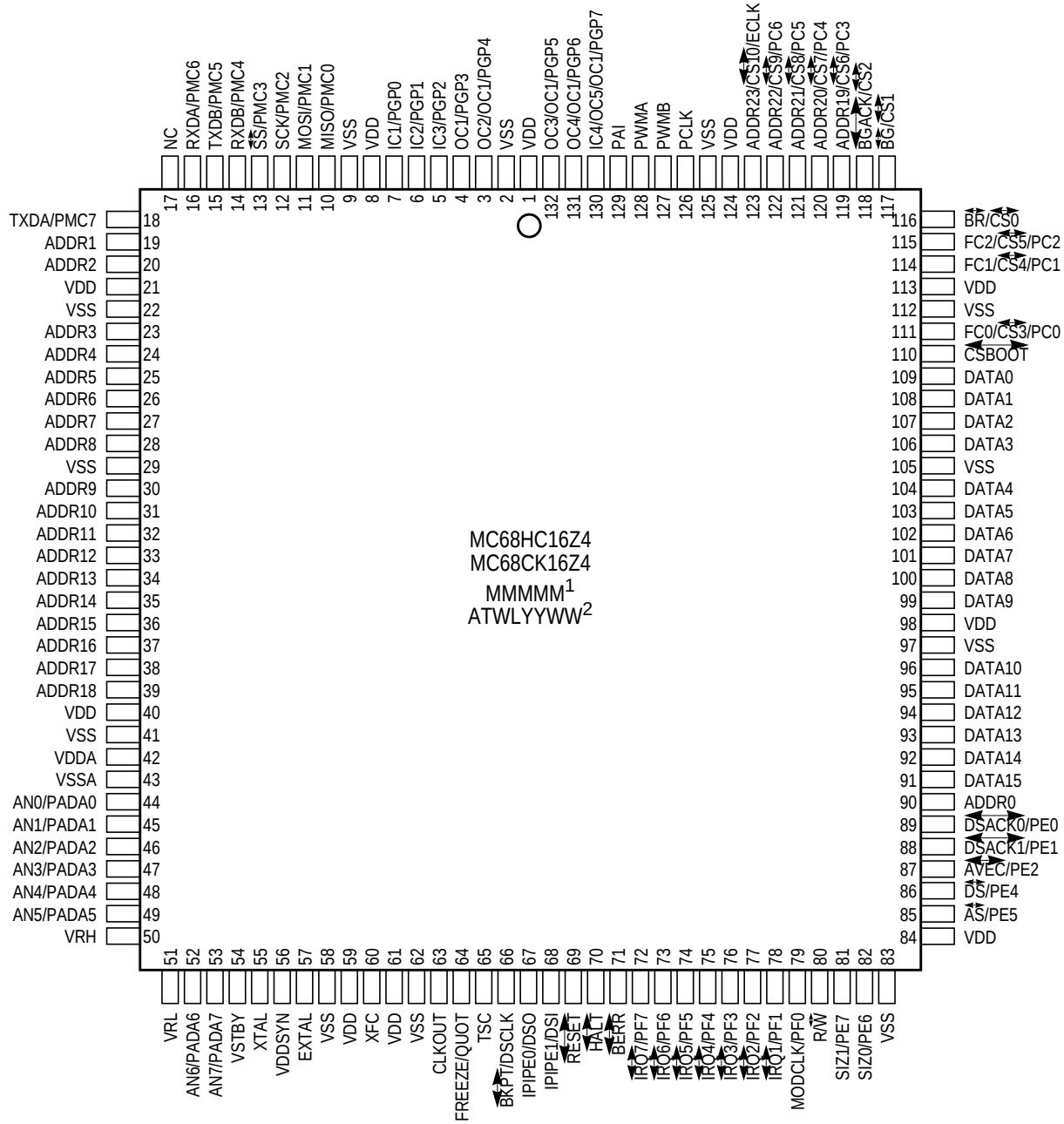


NOTES:
1. MMMMM = MASK OPTION NUMBER
2. ATWLYYWW = ASSEMBLY TEST LOCATION/YEAR, WEEK

Figure B-1 MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 Pin Assignments for 132-Pin Package

Freescale Semiconductor, Inc.

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

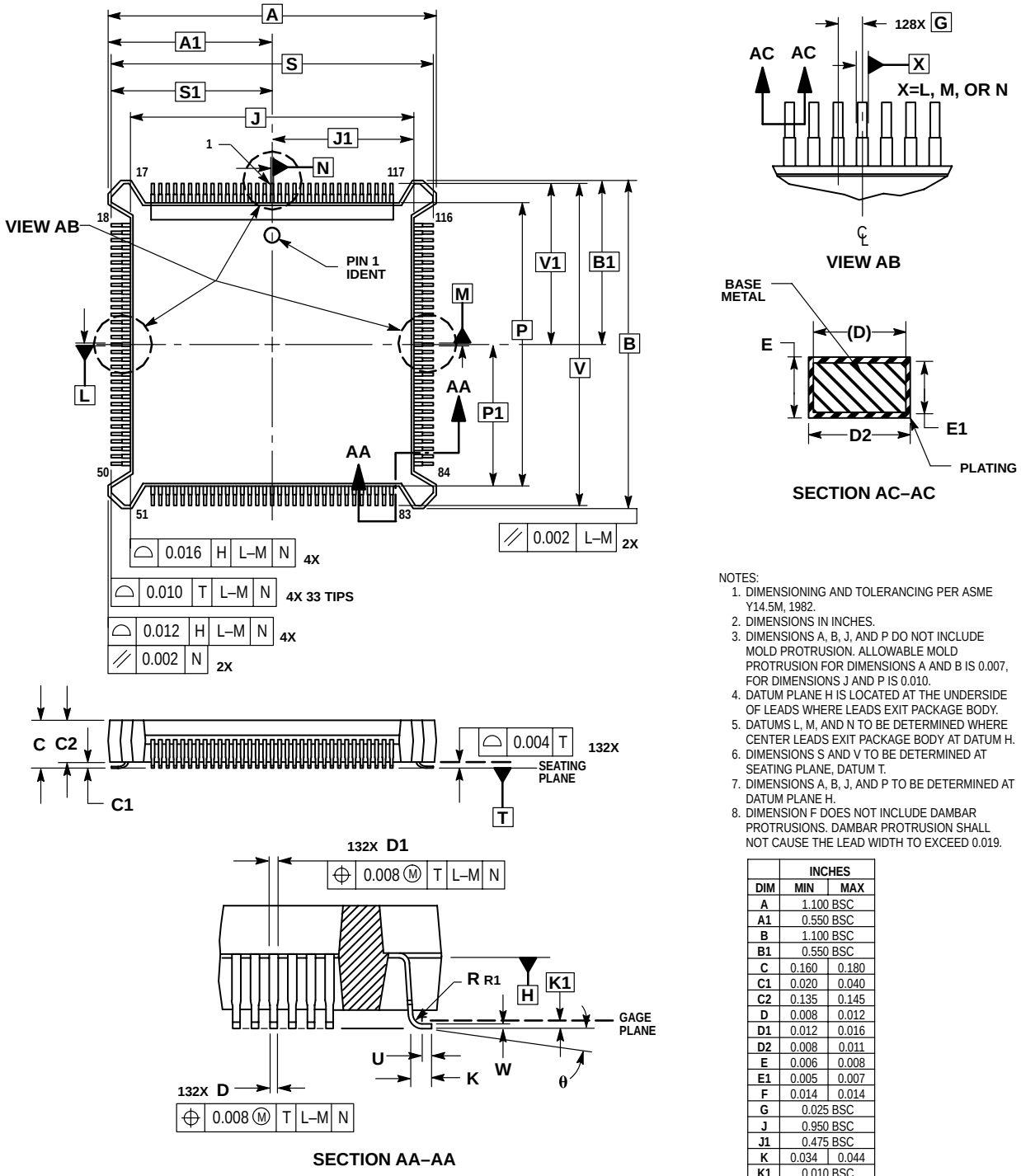


NOTES:
1. MMMMM = MASK OPTION NUMBER
2. ATWLYYWW = ASSEMBLY TEST LOCATION/YEAR, WEEK

HC16Z4/CK16Z4 132-PIN QFP

Figure B-2 MC68HC16Z4/CKZ4 Pin Assignments for 132-Pin Package

Freescale Semiconductor, Inc.



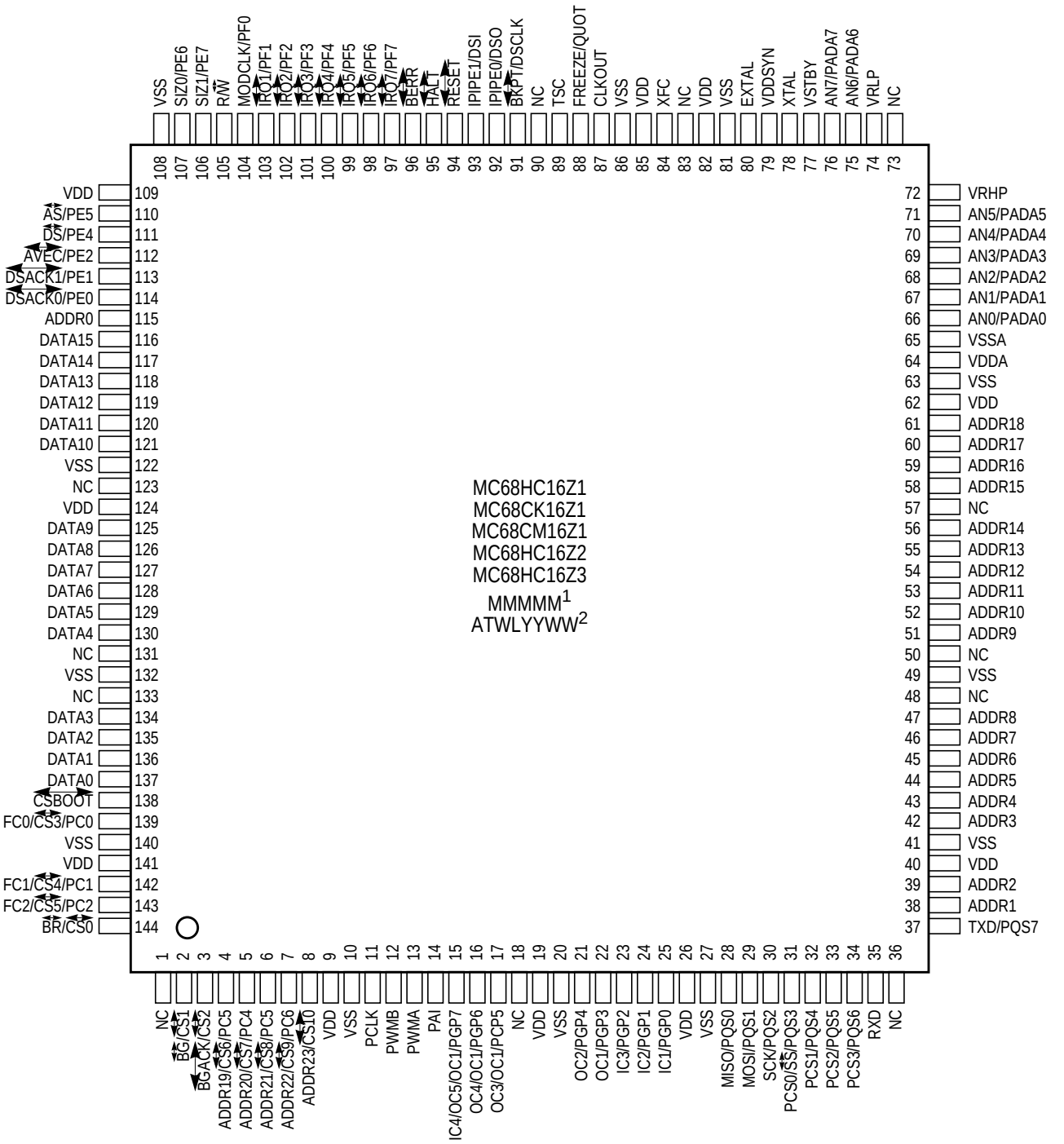
NOTES:
1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1982.
2. DIMENSIONS IN INCHES.
3. DIMENSIONS A, B, J, AND P DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION FOR DIMENSIONS A AND B IS 0.007, FOR DIMENSIONS J AND P IS 0.010.
4. DATUM PLANE H IS LOCATED AT THE UNDERSIDE OF LEADS WHERE LEADS EXIT PACKAGE BODY.
5. DATUMS L, M, AND N TO BE DETERMINED WHERE CENTER LEADS EXIT PACKAGE BODY AT DATUM H.
6. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE, DATUM T.
7. DIMENSIONS A, B, J, AND P TO BE DETERMINED AT DATUM PLANE H.
8. DIMENSION F DOES NOT INCLUDE DAMBAR PROTRUSIONS. DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED 0.019.

	INCHES	
DIM	MIN	MAX
A	1.100 BSC	
A1	0.550 BSC	
B	1.100 BSC	
B1	0.550 BSC	
C	0.160	0.180
C1	0.020	0.040
C2	0.135	0.145
D	0.008	0.012
D1	0.012	0.016
D2	0.008	0.011
E	0.006	0.008
E1	0.005	0.007
F	0.014	0.014
G	0.025 BSC	
J	0.950 BSC	
J1	0.475 BSC	
K	0.034	0.044
K1	0.010 BSC	
P	0.950 BSC	
P1	0.475 BSC	
R1	0.013 REF	
S	1.080 BSC	
S1	0.540 BSC	
U	0.025 REF	
V	1.080 BSC	
V1	0.540 BSC	
W	0.006	0.008
θ	0°	8°

Figure B-3 Case 831A-01 — 132-Pin Package Dimensions

Freescale Semiconductor, Inc.

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



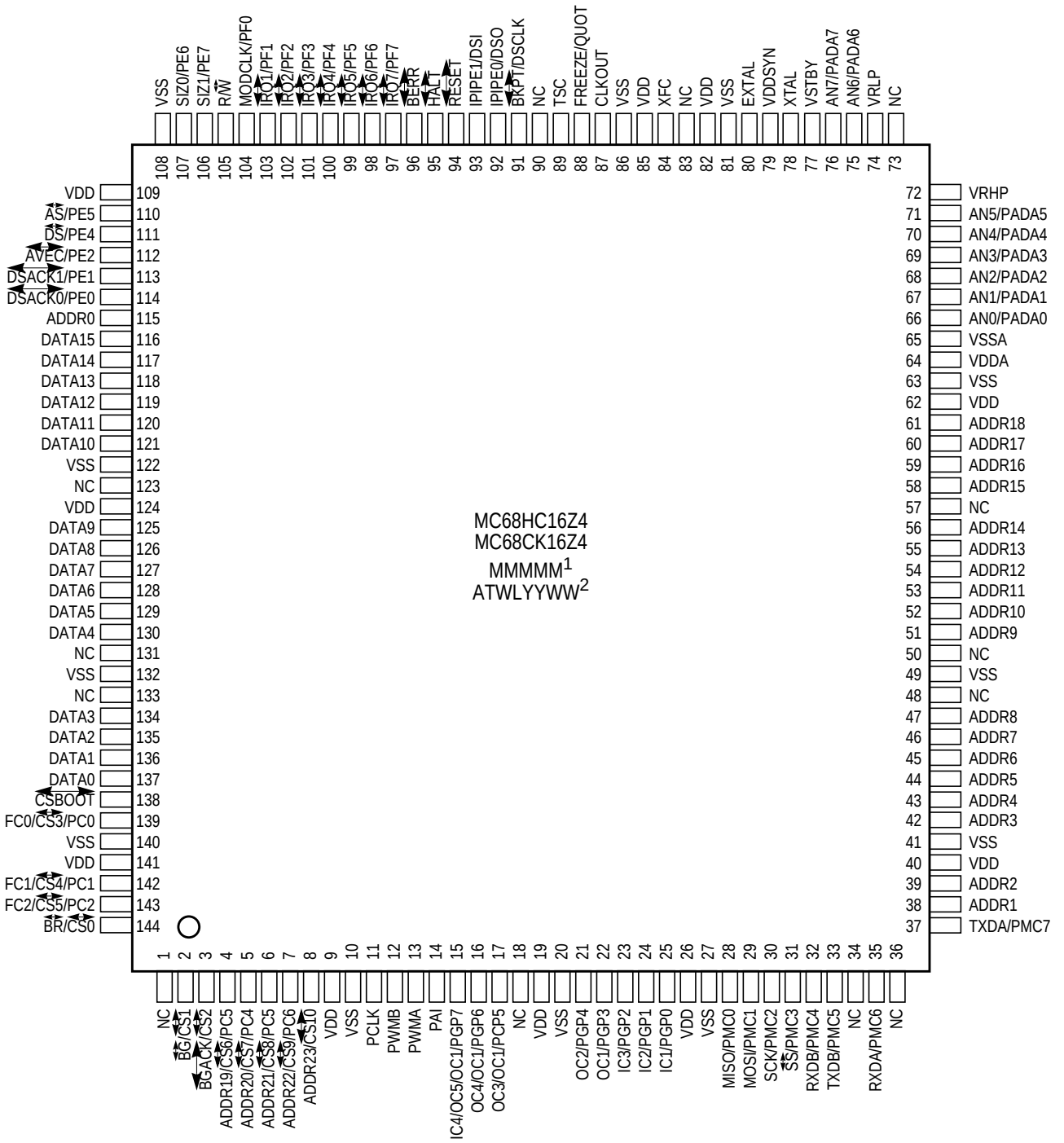
NOTES:
1. MMMMM = MASK OPTION NUMBER
2. ATWLYYWW = ASSEMBLY TEST LOCATION/YEAR, WEEK

HC16Z1/CKZ1/CMZ1/Z2/Z3 144-PIN QFP

Figure B-4 MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 Pin Assignments for 144-Pin Package

Freescale Semiconductor, Inc.

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



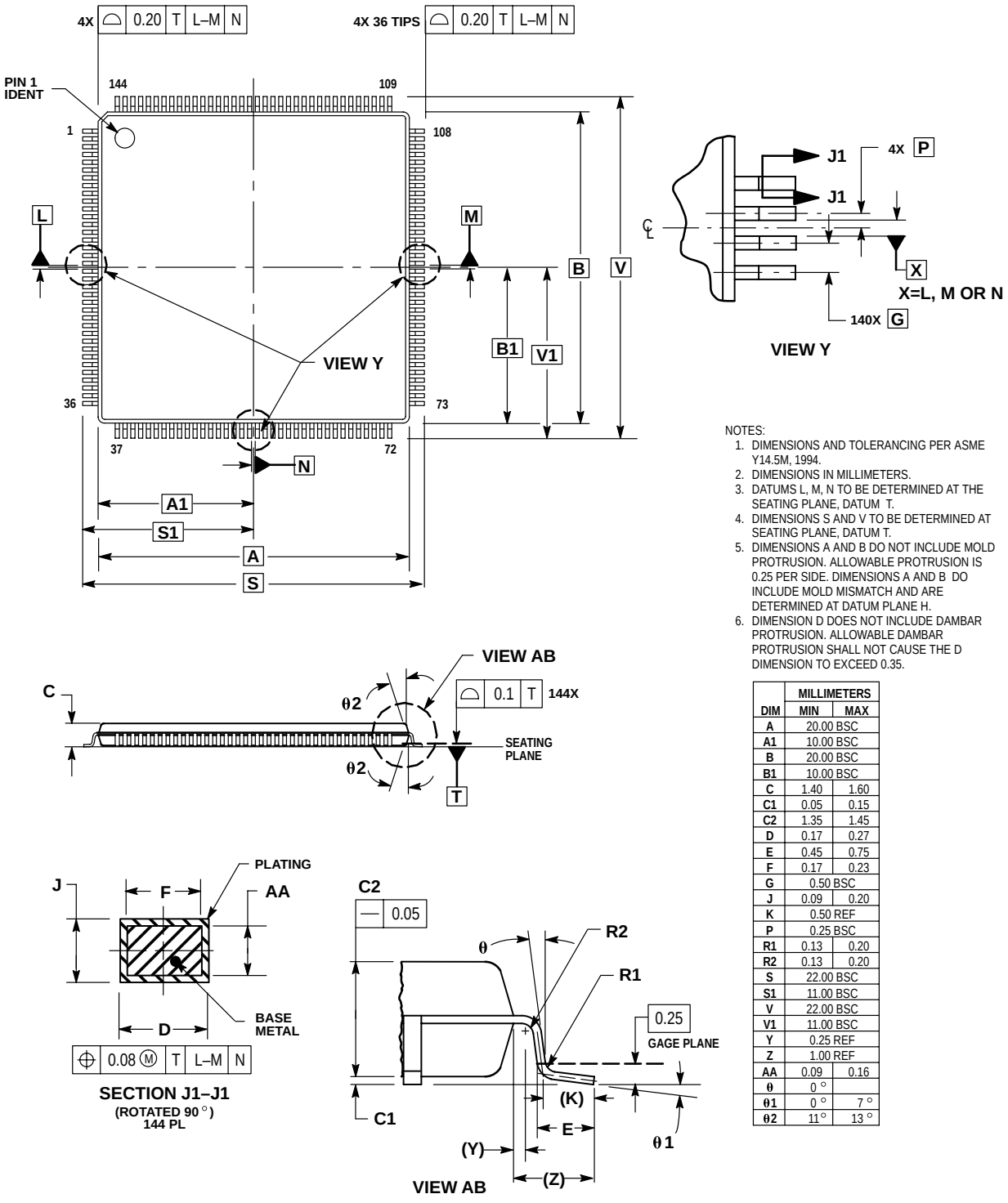
NOTES:
1. MMMMM = MASK OPTION NUMBER
2. ATWLYYWW = ASSEMBLY TEST LOCATION/YEAR, WEEK

HC16Z4/CK16Z4 144-PIN QFP

Figure B-5 MC68HC16Z4/CKZ4 Pin Assignments for 144-Pin Package

Freescale Semiconductor, Inc.

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



DIM	MILLIMETERS MIN	MAX
A	20.00 BSC	
A1	10.00 BSC	
B	20.00 BSC	
B1	10.00 BSC	
C	1.40	1.60
C1	0.05	0.15
C2	1.35	1.45
D	0.17	0.27
E	0.45	0.75
F	0.17	0.23
G	0.50 BSC	
J	0.09	0.20
K	0.50 REF	
P	0.25 BSC	
R1	0.13	0.20
R2	0.13	0.20
S	22.00 BSC	
S1	11.00 BSC	
V	22.00 BSC	
V1	11.00 BSC	
Y	0.25 REF	
Z	1.00 REF	
AA	0.09	0.16
θ	0 °	
θ1	0 °	7 °
θ2	11°	13 °

Figure B-6 Case 918 — 144-Pin Package Dimensions

Freescale Semiconductor, Inc.

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

B.1 Obtaining Updated M68HC16 Z-Series MCU Mechanical Information

Although all devices manufactured by Freescale conform to current JEDEC standards, complete mechanical information regarding M68HC16 Z-series microcontrollers is available through Motorola's Design-Net.

To download updated package specifications, perform the following steps:

1. Visit the Design-Net case outline database search engine at http://design-net.com/cgi-bin/cases.
2. Enter the case outline number, located in **Figure B-3** without the revision code (for example, 831A, not 831A-01) in the field next to the search button.
3. Download the file with the new package diagram.

B.2 Ordering Information

Use the information in **Table B-1** to specify the appropriate device when placing an order.

Table B-1 M68HC16 Z-Series Ordering Information

(Shaded cells indicate preliminary part numbers)

Device	Crystal Input	Operating Voltage	Package Type	Temperature	Frequency (MHz)	Package Order Quantity	Order Number
MC68HC16Z1	32 kHz	5 V	132-Pin PQFP	–40 to +85°C	16 MHz	2	SPMCK16Z1CFC16
						36	MCK68HC16Z1CFC16
						180	MCK16Z1CFC16B1
					20 MHz	2	SPMCK16Z1CFC20
						36	MCK68HC16Z1CFC20
						180	MCK16Z1CFC20B1
					25 MHz	2	SPMCK16Z1CFC25
						36	MCK68HC16Z1CFC25
						180	MCK16Z1CFC25B1
				–40 to +105°C	16 MHz	2	SPMCK16Z1VFC16
						36	MCK68HC16Z1VFC16
						180	MCK16Z1VFC16B1
					20 MHz	2	SPMCK16Z1VFC20
						36	MCK68HC16Z1VFC20
						180	MCK16Z1VFC20B1
					25 MHz	2	SPMCK16Z1VFC25
						36	MCK68HC16Z1VFC25
						180	MCK16Z1VFC25B1

Freescale Semiconductor, Inc.

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

Table B-1 M68HC16 Z-Series Ordering Information (Continued)

(Shaded cells indicate preliminary part numbers)

Device	Crystal Input	Operating Voltage	Package Type	Temperature	Frequency (MHz)	Package Order Quantity	Order Number
MC68HC16Z1	32 kHz	5 V	132-Pin PQFP	–40 to +125°C	16 MHz	2	SPMCK16Z1MFC16
						36	MCK68HC16Z1MFC16
						180	MCK16Z1MFC16B1
					20 MHz	2	SPMCK16Z1MFC20
						36	MCK68HC16Z1MFC20
						180	MCK16Z1MFC20B1
MC68HC16Z1	32 kHz	5 V	144-Pin TQFP	–40 to +85°C	16 MHz	2	SPMCK16Z1CPV16
						60	MCK68HC16Z1CPV16
						300	MCK16Z1CPV16B1
					20 MHz	2	SPMCK16Z1CPV20
						60	MCK68HC16Z1CPV20
						300	MCK16Z1CPV20B1
					25 MHz	2	SPMCK16Z1CPV25
						60	MCK68HC16Z1CPV25
						300	MCK16Z1CPV25B1
				–40 to +105°C	16 MHz	2	SPMCK16Z1VPV16
						60	MCK68HC16Z1VPV16
						300	MCK16Z1VPV16B1
					20 MHz	2	SPMCK16Z1VPV20
						60	MCK68HC16Z1VPV20
						300	MCK16Z1VPV20B1
					25 MHz	2	SPMCK16Z1VPV25
						60	MCK68HC16Z1VPV25
						300	MCK16Z1VPV25B1
				–40 to +125°C	16 MHz	2	SPMCK16Z1MPV16
						60	MCK68HC16Z1MPV16
						300	MCK16Z1MPV16B1
					20 MHz	2	SPMCK16Z1MPV20
						60	MCK68HC16Z1MPV20
						300	MCK16Z1MPV20B1
MC68HC16Z1	32 kHz	2.7 V	132-Pin PQFP	–40 to +85°C	16 MHz	2	SPMCCK16Z1CFC16
						36	MC68CK16Z1CFC16
						180	MCCK16Z1CFC16B1
			144-Pin TQFP	–40 to +85°C	16 MHz	2	SPMCCK16Z1CPV16
						60	MC68CK16Z1CPV16
						300	MCCK16Z1CPV16B1
MC68HC16Z1	4 MHz	2.7 V	132-Pin PQFP	–40 to +85°C	16 MHz	2	SPMCCM16Z1CFC16
						36	MC68CM16Z1CFC16
						180	MCCM16Z1CFC16B1

Freescale Semiconductor, Inc.

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

Table B-1 M68HC16 Z-Series Ordering Information (Continued)

(Shaded cells indicate preliminary part numbers)

Device	Crystal Input	Operating Voltage	Package Type	Temperature	Frequency (MHz)	Package Order Quantity	Order Number
MC68HC16Z1	4 MHz	2.7 V	144-Pin TQFP	–40 to +85°C	16 MHz	2	SPMCCM16Z1CPV16
						60	MC68CM16Z1CPV16
						300	MCCM16Z1CPV16B1
MC68HC16Z2 (ROM)	4 MHz or 32 kHz	5 V	132-Pin PQFP	–40 to +85°C	16 MHz	2	NA
						36	MC68HC16Z2CFC16
						180	NA
					20 MHz	2	NA
						36	MC68HC16Z2CFC20
						180	NA
					25 MHz	2	NA
						36	MC68HC16Z2CFC25
						180	NA
				–40 to +105°C	16 MHz	2	NA
						36	MC68HC16Z2VFC16
						180	NA
					20 MHz	2	NA
						36	MC68HC16Z2VFC20
						180	NA
					25 MHz	2	NA
						36	MC68HC16Z2VFC25
						180	NA
				–40 to +125°C	16 MHz	2	NA
						36	MC68HC16Z2MFC16
						180	NA
					20 MHz	2	NA
						36	MC68HC16Z2MFC20
						180	NA
			144-Pin TQFP	–40 to +85°C	16 MHz	2	NA
						60	MC68HC16Z2CPV16
						300	NA
					20 MHz	2	NA
						60	MC68HC16Z2CPV20
						300	NA
					25 MHz	2	NA
						60	MC68HC16Z2CPV25
						300	NA

Freescale Semiconductor, Inc.

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

Table B-1 M68HC16 Z-Series Ordering Information (Continued)

(Shaded cells indicate preliminary part numbers)

Device	Crystal Input	Operating Voltage	Package Type	Temperature	Frequency (MHz)	Package Order Quantity	Order Number
MC68HC16Z2 (ROM)	4 MHz or 32 kHz	5 V	144-Pin TQFP	–40 to +105°C	16 MHz	2	NA
						60	MC68HC16Z2VPV16
						300	NA
					20 MHz	2	NA
						60	MC68HC16Z2VPV20
						300	NA
					25 MHz	2	NA
						60	MC68HC16Z2VPV25
						300	NA
				–40 to +125°C	16 MHz	2	NA
						60	MC68HC16Z2MPV16
						300	NA
					20 MHz	2	NA
						60	MC68HC16Z2MPV20
						300	NA
MC68HC16Z2 (No ROM)	4 MHz	5 V	132-Pin PQFP	–40 to +85°C	16 MHz	2	SPMCM16Z2BCFC16
						36	MCM16Z2BCFC16
						180	MCM16Z2BCFC16B1
					20 MHz	2	SPMCM16Z2BCFC20
						36	MCM16Z2BCFC20
						180	MCM16Z2BCFC20B1
					25 MHz	2	SPMCM16Z2BCFC25
						36	MCM16Z2BCFC25
						180	MCM16Z2BCFC25B1
				–40 to +105°C	16 MHz	2	SPMCM16Z2BVFC16
						36	MCM16Z2BVFC16
						180	MCM16Z2BVFC16B1
					20 MHz	2	SPMCM16Z2BVFC20
						36	MCM16Z2BVFC20
						180	MCM16Z2BVFC20B1
					25 MHz	2	SPMCM16Z2BVFC25
						36	MCM16Z2BVFC25
						180	MCM16Z2BVFC25B1
				–40 to +125°C	16 MHz	2	SPMCM16Z2BMFC16
						36	MCM16Z2BMFC16
						180	MCM16Z2BMFC16B1
					20 MHz	2	SPMCM16Z2BMFC20
						36	MCM16Z2BMFC20
						180	MCM16Z2BMFC20B1

Freescale Semiconductor, Inc.

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

Table B-1 M68HC16 Z-Series Ordering Information (Continued)

(Shaded cells indicate preliminary part numbers)

Device	Crystal Input	Operating Voltage	Package Type	Temperature	Frequency (MHz)	Package Order Quantity	Order Number
MC68HC16Z2 (No ROM)	4 MHz	5 V	144-Pin TQFP	–40 to +85°C	16 MHz	2	SPMCM16Z2BCPV16
						60	MCM16Z2BCPV16
						300	MCM16Z2BCPV16B1
					20 MHz	2	SPMCM16Z2BCPV20
						60	MCM16Z2BCPV20
						300	MCM16Z2BCPV20B1
					25 MHz	2	SPMCM16Z2BCPV25
						60	MCM16Z2BCPV25
						300	MCM16Z2BCPV25B1
				–40 to +105°C	16 MHz	2	SPMCM16Z2BVPV16
						60	MCM16Z2BVPV16
						300	MCM16Z2BVPV16B1
					20 MHz	2	SPMCM16Z2BVPV20
						60	MCM16Z2BVPV20
						300	MCM16Z2BVPV20B1
					25 MHz	2	SPMCM16Z2BVPV25
						60	MCM16Z2BVPV25
						300	MCM16Z2BVPV25B1
				–40 to +125°C	16 MHz	2	SPMCM16Z2BMPV16
						60	MCM16Z2BMPV16
						300	MCM16Z2BMPV16B1
					20 MHz	2	SPMCM16Z2BMPV20
						60	MCM16Z2BMPV20
						300	MCM16Z2BMPV20B1
MC68HC16Z3 (ROM)	4 MHz or 32 kHz	5 V	132-Pin PQFP	–40 to +85°C	16 MHz	2	NA
						36	MC68HC16Z3CFC16
						180	NA
					20 MHz	2	NA
						36	MC68HC16Z3CFC20
						180	NA
					25 MHz	2	NA
						36	MC68HC16Z3CFC25
						180	NA

Freescale Semiconductor, Inc.

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

Table B-1 M68HC16 Z-Series Ordering Information (Continued)

(Shaded cells indicate preliminary part numbers)

Device	Crystal Input	Operating Voltage	Package Type	Temperature	Frequency (MHz)	Package Order Quantity	Order Number
MC68HC16Z3 (ROM)	4 MHz or 32 kHz	5 V	132-Pin PQFP	–40 to +105°C	16 MHz	2	NA
						36	MC68HC16Z3CFC16
						180	NA
					20 MHz	2	NA
						36	MC68HC16Z3VFC20
						180	NA
					25 MHz	2	NA
						36	MC68HC16Z3VFC25
						180	NA
				–40 to +125°C	16 MHz	2	NA
						36	MC68HC16Z3MFC16
						180	NA
					20 MHz	2	NA
						36	MC68HC16Z3MFC20
						180	NA
			144-Pin TQFP	–40 to +85°C	16 MHz	2	NA
						60	MC68HC16Z3CPV16
						300	NA
					20 MHz	2	NA
						60	MC68HC16Z3CPV20
						300	NA
					25 MHz	2	NA
						60	MC68HC16Z3CPV25
						300	NA
				–40 to +105°C	16 MHz	2	NA
						60	MC68HC16Z3VPV16
						300	NA
					20 MHz	2	NA
						60	MC68HC16Z3VPV20
						300	NA
					25 MHz	2	NA
						60	MC68HC16Z3VPV25
						300	NA
				–40 to +125°C	16 MHz	2	NA
						60	MC68HC16Z3MPV16
						300	NA
					20 MHz	2	NA
						60	MC68HC16Z3MPV20
						300	NA

Freescale Semiconductor, Inc.

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

Table B-1 M68HC16 Z-Series Ordering Information (Continued)

(Shaded cells indicate preliminary part numbers)

Device	Crystal Input	Operating Voltage	Package Type	Temperature	Frequency (MHz)	Package Order Quantity	Order Number
MC68HC16Z3 (RTOS)	4 MHz	5 V	132-Pin PQFP	–40 to +85°C	16 MHz	2	SPMCM16Z3RCFC16
						36	MCM16Z3RCFC16
						180	MCM16Z3RCFC16B1
					20 MHz	2	SPMCM16Z3RCFC20
						36	MCM16Z3RCFC20
						180	MCM16Z3RCFC20B1
					25 MHz	2	SPMCM16Z3RCFC25
						36	MCM16Z3RCFC25
						180	MCM16Z3RCFC25B1
				–40 to +105°C	16 MHz	2	SPMCM16Z3RVFC16
						36	MCM16Z3RVFC16
						180	MCM16Z3RVFC16B1
					20 MHz	2	SPMCM16Z3RVFC20
						36	MCM16Z3RVFC20
						180	MCM16Z3RVFC20B1
					25 MHz	2	SPMCM16Z3RVFC25
						36	MCM16Z3RVFC25
						180	MCM16Z3RVFC25B1
				–40 to +125°C	16 MHz	2	SPMCM16Z3RMFC16
						36	MCM16Z3RMFC16
						180	MCM16Z3RMFC16B1
					20 MHz	2	SPMCM16Z3RMFC20
						36	MCM16Z3RMFC20
						180	MCM16Z3RMFC20B1
			144-Pin TQFP	–40 to +85°C	16 MHz	2	SPMCM16Z3RCPV16
						60	MCM16Z3RCPV16
						300	MCM16Z3RCPV16B1
					20 MHz	2	SPMCM16Z3RCPV20
						60	MCM16Z3RCPV20
						300	MCM16Z3RCPV20B1
					25 MHz	2	SPMCM16Z3RCPV25
						60	MCM16Z3RCPV25
						300	MCM16Z3RCPV25B1

Freescale Semiconductor, Inc.

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

Table B-1 M68HC16 Z-Series Ordering Information (Continued)

(Shaded cells indicate preliminary part numbers)

Device	Crystal Input	Operating Voltage	Package Type	Temperature	Frequency (MHz)	Package Order Quantity	Order Number
MC68HC16Z3 (RTOS)	4 MHz	5 V	144-Pin TQFP	–40 to +105°C	16 MHz	2	SPMCM16Z3RVPV16
						60	MCM16Z3RVPV16
						300	MCM16Z3RVPV16B1
					20 MHz	2	SPMCM16Z3RVPV20
						60	MCM16Z3RVPV20
						300	MCM16Z3RVPV20B1
					25 MHz	2	SPMCM16Z3RVPV25
						60	MCM16Z3RVPV25
						300	MCM16Z3RVPV25B1
				–40 to +125°C	16 MHz	2	SPMCM16Z3RMPV16
						60	MCM16Z3RMPV16
						300	MCM16Z3RMPV16B1
					20 MHz	2	SPMCM16Z3RMPV20
						60	MCM16Z3RMPV20
						300	MCM16Z3RMPV20B1
MC68HC16Z4	32 kHz	5 V	132-Pin PQFP	–40 to +85°C	16 MHz	2	SPMCK16Z4CFC16
						36	MCK68HC16Z4CFC16
						180	MCK16Z4CFC16B1
					20 MHz	2	SPMCK16Z4CFC20
						36	MCK68HC16Z4CFC20
						180	MCK16Z4CFC20B1
					25 MHz	2	SPMCK16Z4CFC25
						36	MCK68HC16Z4CFC25
						180	MCK16Z4CFC25B1
				–40 to +105°C	16 MHz	2	SPMCK16Z4VFC16
						36	MCK68HC16Z4VFC16
						180	MCK16Z4VFC16B1
					20 MHz	2	SPMCK16Z4VFC20
						36	MCK68HC16Z4VFC20
						180	MCK16Z1VFC20B1
					25 MHz	2	SPMCK16Z4VFC25
						36	MCK68HC16Z4VFC25
						180	MCK16Z4VFC25B1
				–40 to +125°C	16 MHz	2	SPMCK16Z4MFC16
						36	MCK68HC16Z4MFC16
						180	MCK16Z4MFC16B1
					20 MHz	2	SPMCK16Z4MFC20
						36	MCK68HC16Z4MFC20
						180	MCK16Z4MFC20B1

Freescale Semiconductor, Inc.

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

Table B-1 M68HC16 Z-Series Ordering Information (Continued)

(Shaded cells indicate preliminary part numbers)

Device	Crystal Input	Operating Voltage	Package Type	Temperature	Frequency (MHz)	Package Order Quantity	Order Number
MC68HC16Z4	32 kHz	5 V	144-Pin TQFP	–40 to +85°C	16 MHz	2	SPMCK16Z4CPV16
						60	MCK68HC16Z4CPV16
						300	MCK16Z4CPV16B1
					20 MHz	2	SPMCK16Z4CPV20
						60	MCK68HC16Z4CPV20
						300	MCK16Z4CPV20B1
					25 MHz	2	SPMCK16Z4CPV25
						60	MCK68HC16Z4CPV25
						300	MCK16Z4CPV25B1
				–40 to +105°C	16 MHz	2	SPMCK16Z4VPV16
						60	MCK68HC16Z4VPV16
						300	MCK16Z4VPV16B1
					20 MHz	2	SPMCK16Z4VPV20
						60	MCK68HC16Z4VPV20
						300	MCK16Z4VPV20B1
					25 MHz	2	SPMCK16Z4VPV25
						60	MCK68HC16Z4VPV25
						300	MCK16Z4VPV25B1
				–40 to +125°C	16 MHz	2	SPMCK16Z4MPV16
						60	MCK68HC16Z4MPV16
						300	MCK16Z4MPV16B1
					20 MHz	2	SPMCK16Z4MPV20
						60	MCK68HC16Z4MPV20
						300	MCK16Z4MPV20B1
		2.7 V	132-Pin PQFP	–40 to +85°C	16 MHz	2	SPMCCK16Z4CFC16
						36	MC68CK16Z4CFC16
						180	MCCK16Z4CFC16B1
			144-Pin TQFP	–40 to +85°C	16 MHz	2	SPMCCK16Z4CPV16
						60	MC68CK16Z4CPV16
						300	MCCK16Z4CPV16B1

Freescale Semiconductor, Inc.

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

APPENDIX C
DEVELOPMENT SUPPORT

This section serves as a brief reference to Freescale development tools for M68HC16 Z-series microcontrollers.

Information provided is complete as of the time of publication, but new systems and software are continually being developed. In addition, there is a growing number of third-party tools available. The Freescale *Microcontroller Development Tools Directory* (MCUDEVTLDIR/D Revision. 3) provides an up-to-date list of development tools. Contact your Freescale representative for further information.

C.1 M68MMDS1632 Modular Development System

The M68MMDS1632 Freescale modular development system (MMDS) is a development tool for evaluating M68HC16 and M68300 MCU-based systems. The MMDS1632 is an in-circuit emulator, which includes a station module and active probe. A separately purchased MPB and PPB completes MMDS functionality with regard to a particular MCU or MCU family. The many MPBs and PPBs available let the MMDS emulate a variety of different MCUs. Contact your Freescale sales representative, who will assist you in selecting and configuring the modular system that fits your needs. A full-featured development system, the MMDS provides both in-circuit emulation and bus analysis capabilities, including:

- Real-time in-circuit emulation at maximum speed of 16 MHz
- Built-in emulation memory
  - — 1-Mbyte main emulation memory (three-clock bus cycle)
  - — 256-Kbyte fast termination (two-clock bus cycle)
  - — 4-Kbyte dual-port emulation memory (three-clock bus cycle)
- Real-time bus analysis
  - — Instruction disassembly
  - — State-machine-controlled triggering
- Four hardware breakpoints; bitwise masking
- Analog/digital emulation
- Synchronized signal output
- Built-in AC power supply, 90 – 264 V, 50 – 60 Hz, FCC and EC EMI compliant
- RS-232 connection to host capable of communicating at 1200, 2400, 4800, 9600, 19200, 38400, or 57600 baud

Freescale Semiconductor, Inc.

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

C.2 M68MEVB1632 Modular Evaluation Board

The M68MEVB1632 Modular Evaluation Board (MEVB) is a development tool for evaluating M68HC16 and M68300 MCU-based systems. The MEVB consists of the M68MPFB1632 modular platform board, an MCU personality board (MPB), an in-circuit debugger (ICD16 or ICD32), and development software. MEVB features include:

- An economical means of evaluating target systems incorporating M68HC16 and M68300 HCMOS MCU devices
- Expansion memory sockets for installing RAM, EPROM, or EEPROM
  - — Data RAM: 32K x 16, 128K x 16, or 512K x 16
  - — EPROM/EEPROM: 32K x 16, 64K x 16, 128K x 16, 256K x 16, or 512K x 16
  - — Fast RAM: 32K x 16 or 128K x 16
- Background-mode operation, for detailed operation from a personal computer platform without an on-board monitor
- Integrated assembly/editing/evaluation/programming environment for easy development
- As many as seven software breakpoints
- Re-usable ICD hardware for your target application debug or control
- Two RS-232C terminal input/output (I/O) ports for user evaluation of the serial communication interface
- Logic analyzer pod connectors
- Port replacement unit (PRU) to rebuild I/O ports lost to address/data/control
- On-board V_{PP} (+12 VDC) generation for MCU and flash EEPROM programming.
- On-board wire-wrap area

NOTE

The MC68HC16Z1 and the MC68HC16Z2/Z3 both utilize the M68HC16MPFB, however, each MCU uses a different personality board (M68MPB16Z1 on the MC68HC16Z1; M68MPB16Z2/Z3 on the MC68HC16Z2/Z3).

Freescale Semiconductor, Inc.

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

APPENDIX D
REGISTER SUMMARY

This appendix contains address maps, register diagrams, and bit/field definitions for M68HC16 Z-series MCUs. More detailed information about register function is provided in the appropriate sections of the manual.

Except for central processing unit resources, information is presented in the intermodule bus address order shown in **Table D-1**.

Table D-1 Module Address Map

Module	Size (Bytes)	Base Address
SIM	128	$YFFA00
SRAM	8	$YFFB00
MRM (MC68HC16Z2/MC68HC16Z3 only)	32	$YFF820
ADC	64	$YFF700
QSM	512	$YFFC00
MCCI (MC68HC16Z4, MC68CK16Z4 only)	64	$YFFC00
GPT	64	$YFF900

Control registers for all the modules in the microcontroller are mapped into a 4-Kbyte block. The state of the module mapping (MM) bit in the SIM module configuration register (SIMCR) determines where the control registers block is located in the system memory map. When MM = 0, register addresses range from $7FF000 to $7FFFFF; when MM = 1, register addresses range from $FFF000 to $FFFFFF.

With the CPU16, ADDR[23:20] follow the logic state of ADDR19 unless driven externally. MM corresponds to IMB ADDR23. If it is cleared, the SIM maps IMB modules into address space $7FF000 – $7FFFFF, which is inaccessible to the CPU16. Modules remain inaccessible until reset occurs. The reset state of MM is one, but the bit is can be written once. Initialization software should make certain it remains set.

D.1 Central Processing Unit

CPU16 registers are not part of the module address map. **Figure D-1** is a functional representation of CPU resources.

Freescale Semiconductor, Inc.

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

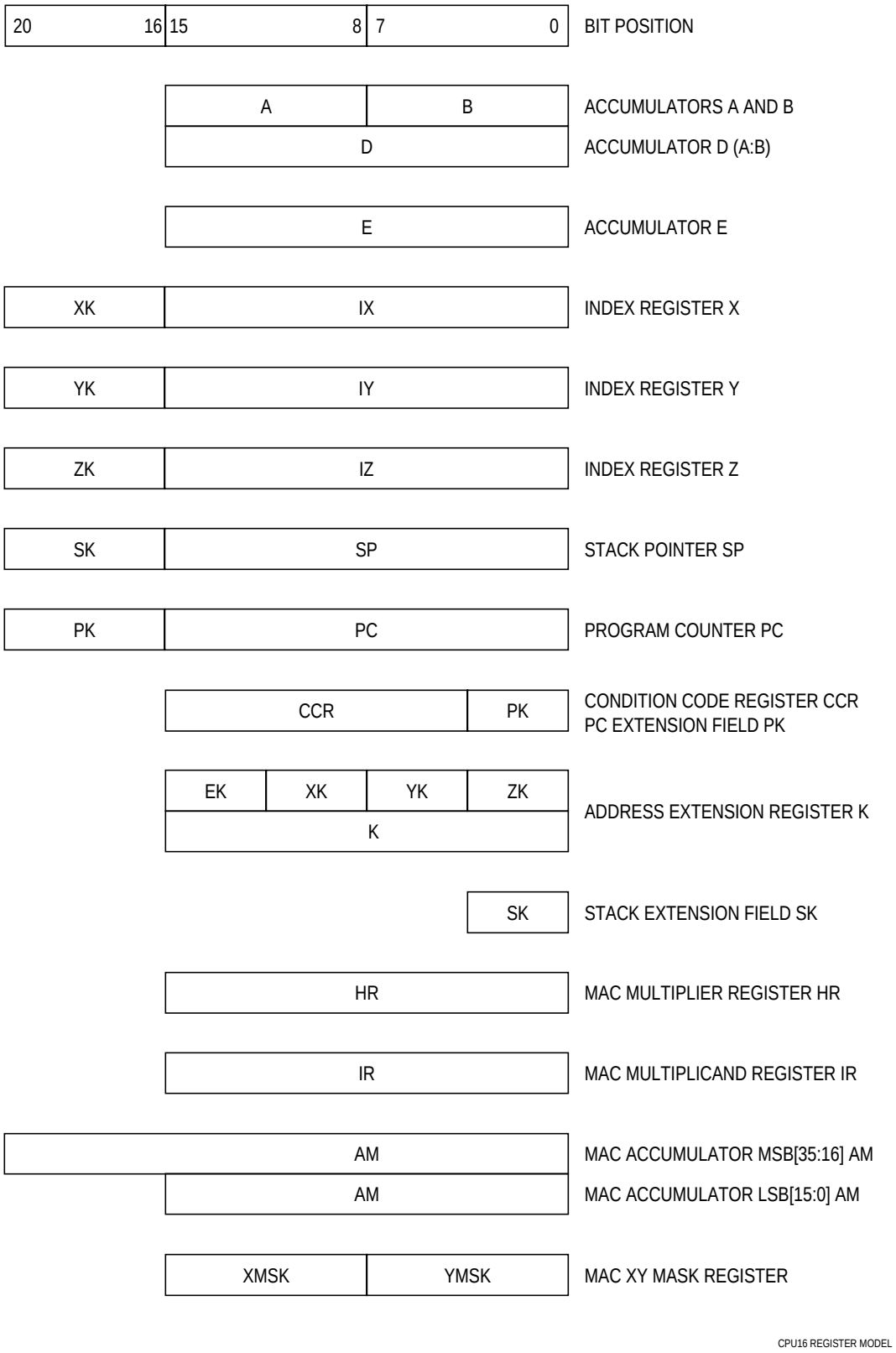


Figure D-1 CPU16 Register Model

Freescale Semiconductor, Inc.

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

D.1.1 Condition Code Register

CCR — Condition Code Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S	MV	H	EV	N	Z	V	C	IP[2:0]			SM	PK[3:0]			

The CCR contains processor status flags, the interrupt priority field, and the program counter address extension field. The CPU16 has a special set of instructions that manipulate the CCR.

S — STOP Enable
0 = Stop CPU16 clocks when LPSTOP instruction is executed.
1 = Perform NOPs when LPSTOP instruction is executed.

MV — Accumulator M overflow flag
Set when overflow into AM35 has occurred.

H — Half Carry Flag
Set when a carry from A3 or B3 occurs during BCD addition.

EV — Accumulator M Extension Overflow Flag
EV is set when an overflow into AM31 has occurred.

N — Negative Flag
N is set under the following conditions:
- When the MSB is set in the operand of a read operation.
- When the MSB is set in the result of a logic or arithmetic operation.

Z — Zero Flag
Z is set under the following conditions:
- When all bits are zero in the operand of a read operation.
- When all bits are zero in the result of a logic or arithmetic operation.

V — Overflow Flag
Set when two's complement overflow occurs as the result of an operation.

C — Carry Flag
Set when carry or borrow occurs during arithmetic operation. Also used during shifts and rotates.

IP[2:0] — Interrupt Priority Field
The priority value in this field (0 to 7) is used to mask low priority interrupts.

SM — Saturate Mode Bit
When SM is set, if either EV or MV is set, data read from AM using TMER or TMET is given maximum positive or negative value, depending on the state of the AM sign bit before overflow.

PK[3:0] — Program Counter Address Extension Field
This field is concatenated with the program counter to form a 20-bit address.

Freescale Semiconductor, Inc.

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

D.2 System Integration Module

Table D-2 shows the SIM address map.

Table D-2 SIM Address Map

Address[1]	15 8	7 0
$YFFA00	SIM Module Configuration Register (SIMCR)	
$YFFA02	SIM Test Register (SIMTR)	
$YFFA04	Clock Synthesizer Control Register (SYNCR)	
$YFFA06	Not Used	Reset Status Register (RSR)
$YFFA08	SIM Test Register E (SIMTRE)	
$YFFA0A	Not Used	Not Used
$YFFA0C	Not Used	Not Used
$YFFA0E	Not Used	Not Used
$YFFA10	Not Used	Port E Data Register 0 (PORTE0)
$YFFA12	Not Used	Port E Data Register 1(PORTE1)
$YFFA14	Not Used	Port E Data Direction Register (DDRE)
$YFFA16	Not Used	Port E Pin Assignment Register (PEPAR)
$YFFA18	Not Used	Port F Data Register 0 (PORTF0)
$YFFA1A	Not Used	Port F Data Register 1 (PORTF1)
$YFFA1C	Not Used	Port F Data Direction Register (DDRF)
$YFFA1E	Not Used	Port F Pin Assignment Register (PFPAR)
$YFFA20	Not Used	System Protection Control Register (SYPCR)
$YFFA22	Periodic Interrupt Control Register (PICR)	
$YFFA24	Periodic Interrupt Timer Register (PITR)	
$YFFA26	Not Used	Software Watchdog Service Register (SWSR)
$YFFA28	Not Used	
$YFFA2A	Not Used	
$YFFA2C	Not Used	
$YFFA2E	Not Used	
$YFFA30	Test Module Master Shift A Register (TSTMSRA)	
$YFFA32	Test Module Master Shift B Register (TSTMSRB)	
$YFFA34	Test Module Shift Count Register (TSTSC)	
$YFFA36	Test Module Repetition Counter Register (TSTRC)	
$YFFA38	Test Module Control Register (CREG)	
$YFFA3A	Test Module Distributed Register (DREG)	
$YFFA3C	Not Used	
$YFFA3E	Not Used	
$YFFA40	Not Used	Port C Data Register (PORTC)
$YFFA42	Not Used	Not Used
$YFFA44	Chip-Select Pin Assignment Register 0 (CSPAR0)	
$YFFA46	Chip-Select Pin Assignment Register 1 (CSPAR1)	
$YFFA48	Chip-Select Base Address Register Boot (CSBARBT)	
$YFFA4A	Chip-Select Option Register Boot (CSORBT)	
	Chip-Select Base Address Register 0 (CSBAR0)	

Freescale Semiconductor, Inc.

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

Table D-2 SIM Address Map (Continued)

Address[1]	15 8	7 0
$YFFA4E	Chip-Select Option Address Register 0 (CSOR0)	
$YFFA50	Chip-Select Base Address Register 1 (CSBAR1)	
$YFFA52	Chip-Select Option Address Register 1 (CSOR1)	
$YFFA54	Chip-Select Base Address Register 2 (CSBAR2)	
$YFFA56	Chip-Select Option Address Register 2 (CSOR2)	
$YFFA58	Chip-Select Base Address Register 3 (CSBAR3)	
$YFFA5A	Chip-Select Option Address Register 3 (CSOR3)	
$YFFA5C	Chip-Select Base Address Register 4 (CSBAR4)	
$YFFA5E	Chip-Select Option Address Register 4 (CSOR4)	
$YFFA60	Chip-Select Base Address Register 5 (CSBAR5)	
$YFFA62	Chip-Select Option Address Register 5 (CSOR5)	
$YFFA64	Chip-Select Base Address Register 6 (CSBAR6)	
$YFFA66	Chip-Select Option Address Register 6 (CSOR6)	
$YFFA68	Chip-Select Base Address Register 7 (CSBAR7)	
$YFFA6A	Chip-Select Option Address Register 7 (CSOR7)	
$YFFA6C	Chip-Select Base Address Register 8 (CSBAR8)	
$YFFA6E	Chip-Select Option Address Register 8 (CSOR8)	
$YFFA70	Chip-Select Base Address Register 9 (CSBAR9)	
$YFFA72	Chip-Select Option Address Register 9 (CSOR9)	
$YFFA74	Chip-Select Base Address Register 10 (CSBAR10)	
$YFFA76	Chip-Select Option Address Register 10 (CSOR10)	
$YFFA78	Not Used	
$YFFA7A	Not Used	
$YFFA7C	Not Used	
$YFFA7E	Not Used	

NOTES:
1. Y = M111, where M is the logic state of the module mapping (MM) bit in the SIMCR.

Freescale Semiconductor, Inc.

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

D.2.1 SIM Module Configuration Register

SIMCR — SIM Module Configuration Register **$YFFA00**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXOFF	FRZSW	FRZBM	0	RSVD[1]	0	SHEN[1:0]		SUPV	MM	0	0	IARB[3:0]			
RESET:															
0	1	1	0	$\overline{\text{DATA11}}$	0	0	0	1	1	0	0	1	1	1	1

NOTES:
1. This bit must be left at zero. Pulling $\overline{\text{DATA11}}$ high during reset ensures this bit remains zero. A one in this bit could allow the MCU to enter an unsupported operating mode.

SIMCR controls system configuration. SIMCR can be read or written at any time, except for the module mapping (MM) bit, which can only be written once after reset, and the reserved bit, which is read-only. Write has no effect.

EXOFF — External Clock Off

0 = The CLKOUT pin is driven during normal operation.
1 = The CLKOUT pin is placed in a high-impedance state.

FRZSW — Freeze Software Enable

0 = When FREEZE is asserted, the software watchdog and periodic interrupt timer continue to operate, allowing interrupts during background debug mode.
1 = When FREEZE is asserted, the software watchdog and periodic interrupt timer are disabled, preventing interrupts during background debug mode.

FRZBM — Freeze Bus Monitor Enable

0 = When FREEZE is asserted, the bus monitor continues to operate.
1 = When FREEZE is asserted, the bus monitor is disabled.

SHEN[1:0] — Show Cycle Enable

The SHEN field determines how the external bus is driven during internal transfer operations. A show cycle allows internal transfers to be monitored externally.

Table D-3 indicates whether show cycle data is driven externally, and whether external bus arbitration can occur. To prevent bus conflict, external devices must not be selected during show cycles.

Table D-3 Show Cycle Enable Bits

SHEN[1:0]	Action
00	Show cycles disabled, external arbitration enabled
01	Show cycles enabled, external arbitration disabled
10	Show cycles enabled, external arbitration enabled
11	Show cycles enabled, external arbitration enabled; internal activity is halted by a bus grant

SUPV — Supervisor/User Data Space

This bit has no effect because the CPU16 always operates in the supervisor mode.

Freescale Semiconductor, Inc.

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

MM — Module Mapping

0 = Internal modules are addressed from $7FF000 – $7FFFFF.

1 = Internal modules are addressed from $FFF000 – $FFFFFF.

The logic state of the MM determines the value of ADDR23 for IMB module addresses. Because ADDR[23:20] are driven to the same state as ADDR19, MM must be set to one. If MM is cleared, IMB modules are inaccessible to the CPU16. This bit can be written only once after reset.

IARB[3:0] — Interrupt Arbitration ID

Each module that can generate interrupts, including the SIM, has an IARB field. Each IARB field can be assigned a value from $0 to $F. During an interrupt acknowledge cycle, IARB permits arbitration among simultaneous interrupts of the same priority level. The reset value of the SIM IARB field is $F, the highest priority. This prevents SIM interrupts from being discarded during system initialization.

D.2.2 System Integration Test Register

SIMTR — System Integration Test Register **$YFFA02**

Used for factory test only.

D.2.3 Clock Synthesizer Control Register

SYNCR — Clock Synthesizer Control Register **$YFFA04**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	X	Y[5:0]						EDIV	0	0	RSVD[1]	SLOCK	RSVD[1]	STSIM	STEXT
RESET:															
0	0	1	1	1	1	1	1	0	0	0	0	U	0	0	0

NOTES:

1. Ensure that the software does not change the value of these bits. They should always be zero.

This register determines system clock operating frequency and operation during low-power stop mode. With a slow reference frequency between 25 and 50 kHz (typically a 32.768-kHz crystal), the clock frequency is determined by the following equation:

$$f_{sys} = f_{ref}[4(Y + 1)(2^{(2W + X)})]$$

With a fast reference frequency between 1 and 6 MHz (typically a 4.194-MHz crystal), the reference frequency is divided by 128 before it is passed to the PLL system. The clock frequency is determined by the following equation:

$$f_{sys} = \frac{f_{ref}}{128}[4(Y + 1)(2^{(2W + X)})]$$

Freescale Semiconductor, Inc.

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

W — Frequency Control (VCO)

This bit controls a prescaler tap in the synthesizer feedback loop. Setting this bit increases the VCO speed by a factor of four. VCO relock delay is required.

X — Frequency Control (Prescaler)

This bit controls a divide by two prescaler that is not in the synthesizer feedback loop. Setting the bit doubles clock speed without changing the VCO speed. No VCO relock delay is required.

Y[5:0] — Frequency Control (Counter)

The Y field controls the modulus down counter in the synthesizer feedback loop, causing it to divide by a value of Y + 1. VCO relock delay is required.

EDIV — E Clock Divide Rate

0 = ECLK frequency is system clock divided by eight.
1 = ECLK frequency is system clock divided by sixteen.

SLOCK — Synthesizer Lock Flag

0 = VCO is enabled, but has not locked.
1 = VCO has locked on the desired frequency or VCO is disabled.

The MCU remains in reset until the synthesizer locks, but SLOCK does not indicate synthesizer lock status until after the user first writes to SYNCR.

STSIM — Stop Mode SIM Clock

0 = When LPSTOP is executed, the SIM clock is driven from the external crystal oscillator and the VCO is turned off to conserve power.
1 = When LPSTOP is executed, the SIM clock is driven from the internal VCO.

STEXT — Stop Mode External Clock

0 = When LPSTOP is executed, the CLKOUT signal is held negated to conserve power.
1 = When LPSTOP is executed and EXOFF 1 in SIMCR, the CLKOUT signal is driven from the SIM clock, as determined by the state of the STSIM bit.

D.2.4 Reset Status Register

RSR — Reset Status Register **$YFFA06**

15 … 8	7	6	5	4	3	2	1	0
NOT USED	EXT	POW	SW	HLT	0	RSVD	SYS	TST

RSR contains a status bit for each reset source in the MCU. RSR is updated when the MCU comes out of reset. A set bit indicates what type of reset occurred. If multiple sources assert reset signals at the same time, more than one bit in RSR may be set. This register can be read at any time; a write has no effect. Bits [15:8] are unimplemented and always read zero.

EXT — External Reset

Reset caused by the $\overline{\text{RESET}}$ pin.

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

POW — Power-Up Reset
Reset caused by the power-up reset circuit.

SW — Software Watchdog Reset
Reset caused by the software watchdog circuit.

HLT — Halt Monitor Reset
Reset caused by the halt monitor.

SYS — System Reset
The CPU16 does not support this function. This bit will never be set.

TST — Test Submodule Reset
Reset caused by the test submodule. Used during factory test reserved operating mode only.

D.2.5 System Integration Test Register E

SIMTRE — System Integration Test Register E **$YFFA08**

Used for factory test only.

D.2.6 Port E Data Register

PORTE0 — Port E0 Data Register **$YFFA10**
PORTE1 — Port E1 Data Register **$YFFA12**

15 … 8	7	6	5	4	3	2	1	0
NOT USED	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
RESET:	U	U	U	U	U	U	U	U

This register can be accessed in two locations and can be read or written at any time. A write to this register is stored in an internal data latch, and if any pin in the corresponding port is configured as an output, the value stored for that bit is driven out on the pin. A read of this data register returns the value at the pin only if the pin is configured as a discrete input. Otherwise, the value read is the value stored in the register. Bits [15:8] are unimplemented and will always read zero.

D.2.7 Port E Data Direction Register

DDRE — Port E Data Direction Register **$YFFA14**

15 … 8	7	6	5	4	3	2	1	0
NOT USED	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0
RESET:	0	0	0	0	0	0	0	0

Bits in this register control the direction of the port E pin drivers when pins are configured for I/O. Setting a bit configures the corresponding pin as an output; clearing a bit configures the corresponding pin as an input. This register can be read or written at any time. Bits [15:8] are unimplemented and will always read zero.

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

NOTE

When changing a port E pin from an output to an input, the pin will drive high for approximately four milliseconds. This ensures that the shared bus control signal will be in a negated state before the pin becomes an input.

D.2.8 Port E Pin Assignment Register

PEPAR — Port E Pin Assignment **$YFFA16**

15 … 8	7	6	5	4	3	2	1	0
NOT USED	PEPA7	PEPA6	PEPA5	PEPA4	PEPA3	PEPA2	PEPA1	PEPA0
RESET:	DATA8	DATA8	DATA8	DATA8	DATA8	DATA8	DATA8	DATA8

This register determines the function of port E pins. Setting a bit assigns the corresponding pin to a bus control signal; clearing a bit assigns the pin to I/O port E. PE3 is not connected to a pin. PEPA3 can be read and written, but has no function. Bits [15:8] are unimplemented and will always read zero.

Table D-4 displays port E pin assignments.

Table D-4 Port E Pin Assignments

PEPAR Bit	Port E Signal	Bus Control Signal
PEPA7	PE7	SIZ1
PEPA6	PE6	SIZ0
PEPA5	PE5	$\overline{AS}$
PEPA4	PE4	$\overline{DS}$
PEPA3	PE3	—[1]
PEPA2	PE2	$\overline{AVEC}$
PEPA1	PE1	$\overline{DSACK1}$
PEPA0	PE0	$\overline{DSACK0}$

NOTES:
1. The CPU16 does not support the $\overline{RMC}$ function for this pin. This bit is not connected to a pin for I/O usage.

D.2.9 Port F Data Register

PORTF0 — Port F Data Register 0 **$YFFA18**
PORTF1 — Port F Data Register 1 **$YFFA1A**

15 … 8	7	6	5	4	3	2	1	0
NOT USED	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
RESET:	U	U	U	U	U	U	U	U

This register can be accessed in two locations and can be read or written at any time. A write to this register is stored in an internal data latch, and if any pin in the corre-

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

sponding port is configured as an output, the value stored for that bit is driven out on the pin. A read of this data register returns the value at the pin only if the pin is configured as a discrete input. Otherwise, the value read is the value stored in the register. Bits [15:8] are unimplemented and will always read zero.

D.2.10 Port F Data Direction Register

DDRF — Port F Data Direction Register **$YFFA1C**

15 … 8	7	6	5	4	3	2	1	0
NOT USED	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0
RESET:	0	0	0	0	0	0	0	0

This register controls the direction of the port F pin drivers when pins are configured for I/O. Setting a bit configures the corresponding pin as an output; clearing a bit configures the corresponding pin as an input. This register can be read or written at any time. Bits [15:8] are unimplemented and will always read zero.

D.2.11 Port F Pin Assignment Register

PFPAR — Port F Pin Assignment Register **$YFFA1E**

15 … 8	7	6	5	4	3	2	1	0
NOT USED	PFPA7	PFPA6	PFPA5	PFPA4	PFPA3	PFPA2	PFPA1	PFPA0
RESET:	DATA9	DATA9	DATA9	DATA9	DATA9	DATA9	DATA9	DATA9

This register determines the function of port F pins. Setting a bit assigns the corresponding pin to a control signal; clearing a bit assigns the pin to port F. Bits [15:8] are unimplemented and will always read zero. Refer to **Table D-5**.

Table D-5 Port F Pin Assignments

PFPAR Field	Port F Signal	Alternate Signal
PFPA7	PF7	$\overline{\text{IRQ7}}$
PFPA6	PF6	$\overline{\text{IRQ6}}$
PFPA5	PF5	$\overline{\text{IRQ5}}$
PFPA4	PF4	$\overline{\text{IRQ4}}$
PFPA3	PF3	$\overline{\text{IRQ3}}$
PFPA2	PF2	$\overline{\text{IRQ2}}$
PFPA1	PF1	$\overline{\text{IRQ1}}$
PFPA0	PF0	$\overline{\text{MODCLK}}$

Freescale Semiconductor, Inc.

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

D.2.12 System Protection Control Register

SYPCR — System Protection Control Register **$YFFA20**

15 … 8	7	6	5	4	3	2	1	0
NOT USED	SWE	SWP	SWT[1:0]		HME	BME	BMT[1:0]	
RESET:	1	$\overline{\text{MODCLK}}$	0	0	0	0	0	0

This register controls system monitor functions, software watchdog clock prescaling, and bus monitor timing. This register can be written once following power-on or reset. Bits [15:8] are unimplemented and will always read zero.

SWE — Software Watchdog Enable
- 0 = Software watchdog is disabled.
- 1 = Software watchdog is enabled.

SWP — Software Watchdog Prescaler

This bit controls the value of the software watchdog prescaler.
- 0 = Software watchdog clock is not prescaled.
- 1 = Software watchdog clock is prescaled by 512.

The reset value of SWP is the complement of the state of the MODCLK pin during reset.

SWT[1:0] — Software Watchdog Timing

This field selects the divide ratio used to establish the software watchdog time-out period. Refer to **Table D-6**.

Table D-6 Software Watchdog Divide Ratio

SWP	SWT[1:0]	Divide Ratio
0	00	2^9
0	01	2^{11}
0	10	2^{13}
0	11	2^{15}
1	00	2^{18}
1	01	2^{20}
1	10	2^{22}
1	11	2^{24}

The following equation calculates the time-out period for a slow reference frequency, where f_{ref} is equal to the EXTAL crystal frequency.

$$\text{Time-out Period} = \frac{\text{Divide Ratio Specified by SWP and SWT[1:0]}}{f_{ref}}$$

The following equation calculates the time-out period for a fast reference frequency, where f_{ref} is equal to the EXTAL crystal frequency.

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

$$\text{Time-out Period} = \frac{(128)(\text{Divide Ratio Specified by SWP and SWT[1:0]})}{f_{ref}}$$

The following equation calculates the time-out period for an externally input clock frequency on both slow and fast reference frequency devices, when f_{sys} is equal to the system clock frequency.

$$\text{Time-out Period} = \frac{\text{Divide Ratio Specified by SWP and SWT[1:0]}}{f_{sys}}$$

HME — Halt Monitor Enable
0 = Halt monitor is disabled.
1 = Halt monitor is enabled.

BME — Bus Monitor External Enable
0 = Disable bus monitor for external bus cycles.
1 = Enable bus monitor for external bus cycles.

BMT[1:0] — Bus Monitor Timing
This field selects the bus monitor time-out period. Refer to **Table D-7**.

Table D-7 Bus Monitor Time-Out Period

BMT[1:0]	Bus Monitor Time-Out Period
00	64 system clocks
01	32 system clocks
10	16 system clocks
11	8 system clocks

D.2.13 Periodic Interrupt Control Register

PICR — Periodic Interrupt Control Register **$YFFA22**

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

PICR sets the interrupt level and vector number for the periodic interrupt timer (PIT). Bits [10:0] can be read or written at any time. Bits [15:11] are unimplemented and always read zero.

PIRQL[2:0] — Periodic Interrupt Request Level
This field determines the priority of periodic interrupt requests. A value of %000 disables PIT interrupts.

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

PIV[7:0] — Periodic Interrupt Vector

This field specifies the periodic interrupt vector number supplied by the SIM when the CPU16 acknowledges an interrupt request.

D.2.14 Periodic Interrupt Timer Register

PITR — Periodic Interrupt Timer Register **$YFFA24**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PTP	PITM[7:0]							
RESET:															
0	0	0	0	0	0	0	$\overline{\text{MODCLK}}$	0	0	0	0	0	0	0	0

Contains the count value for the periodic timer. This register can be read or written at any time.

PTP — Periodic Timer Prescaler

0 = Periodic timer clock not prescaled.

1 = Periodic timer clock prescaled by a value of 512.

PITM[7:0] — Periodic Interrupt Timing Modulus

This field determines the periodic interrupt rate. Use the following equations to calculate timer period.

The following equation calculates the PIT period when a slow reference frequency is used:

$$\text{PIT Period} = \frac{(\text{PITM[7:0]})(1 \text{ if PTP} = 0, 512 \text{ if PTP} = 1)(4)}{f_{ref}}$$

The following equation calculates the PIT period when a fast reference frequency is used:

$$\text{PIT Period} = \frac{(128)(\text{PITM[7:0]})(1 \text{ if PTP} = 0, 512 \text{ if PTP} = 1)(4)}{f_{ref}}$$

The following equation calculates the PIT period for an externally input clock frequency on both slow and fast reference frequency devices.

$$\text{PIT Period} = \frac{(\text{PITM[7:0]})(1 \text{ if PTP} = 0, 512 \text{ if PTP} = 1)(4)}{f_{sys}}$$

Freescale Semiconductor, Inc.

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

D.2.15 Software Watchdog Service Register

SWSR — Software Watchdog Service Register[1] **$YFFA26**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								SWSR[7:0]							
RESET:								0	0	0	0	0	0	0	0

NOTES:
1. This register is shown with a read value.

This register can be read or written at any time. Bits [15:8] are unimplemented and will always read zero.

To reset the software watchdog:

1. Write $55 to SWSR.
2. Write $AA to SWSR.

Both writes must occur in the order specified before the software watchdog times out, but any number of instructions can occur between the two writes.

D.2.16 Port C Data Register

PORTC — Port C Data Register **$YFFA40**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								0	PC6	PC5	PC4	PC3	PC2	PC1	PC0
RESET:								0	1	1	1	1	1	1	1

This register latches data for chip-select pins configured as discrete outputs. This register can be read or written at any time. Bits [15:8] are unimplemented and will always read zero.

D.2.17 Chip-Select Pin Assignment Registers

CSPAR0 — Chip-Select Pin Assignment Register 0 **$YFFA44**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	CS5PA[1:0]		CS4PA[1:0]		CS3PA[1:0]		CS2PA[1:0]		CS1PA[1:0]		CS0PA[1:0]		CSBTPA[1:0]	
RESET:															
0	0	DATA2	1	DATA2	1	DATA2	1	DATA1	1	DATA1	1	DATA1	1	1	DATA0

Chip-select pin assignment registers configure the chip-select pins for discrete I/O, an alternate function, or as an 8-bit or 16-bit chip-select. The possible encodings for each 2-bit field in CSPAR[0:1] (except for CSBTPA[1:0]) are shown in **Table D-8**.

Freescale Semiconductor, Inc.

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

Table D-8 Pin Assignment Field Encoding

CSxPA[1:0]	Description
00	Discrete output[1]
01	Alternate function[1]
10	Chip-select (8-bit port)
11	Chip-select (16-bit port)

NOTES:
1. Does not apply to the $\overline{\text{CSBOOT}}$ field.

This register contains seven 2-bit fields that determine the function of corresponding chip-select pins. Bits [15:14] are not used. These bits always read zero; writes have no effect. CSPAR0 bit 1 always reads one; writes to CSPAR0 bit 1 have no effect. The alternate functions can be enabled by data bus mode selection during reset. This register may be read or written at any time. After reset, software may enable one or more pins as discrete outputs.

Table D-9 shows CSPAR0 pin assignments.

Table D-9 CSPAR0 Pin Assignments

CSPAR0 Field	Chip-Select Signal	Alternate Signal	Discrete Output
CS5PA[1:0]	CS5	FC2	PC2
CS4PA[1:0]	CS4	FC1	PC1
CS3PA[1:0]	CS3	FC0	PC0
CS2PA[1:0]	CS2	$\overline{\text{BGACK}}$	—
CS1PA[1:0]	CS1	$\overline{\text{BG}}$	—
CS0PA[1:0]	CS0	$\overline{\text{BR}}$	—
CSBTPA[1:0]	$\overline{\text{CSBOOT}}$	—	—

CSPAR1 — Chip-Select Pin Assignment Register 1 **$YFFA46**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	CS10PA[1:0]		CS9PA[1:0]		CS8PA[1:0]		CS7PA[1:0]		CS6PA[1:0]	
RESET:															
0	0	0	0	0	0	DATA7[1]	1	DATA [7:6][1]	1	DATA [7:5][1]	1	DATA [7:4][1]	1	DATA [7:3][1]	1

NOTES:
1. Refer to **Table D-11** for CSPAR1 reset state information.

CSPAR1 contains five 2-bit fields that determine the functions of corresponding chip-select pins. Bits [15:10] are not used. These bits always read zero; writes have no effect. **Table D-10** shows CSPAR1 pin assignments, including alternate functions that can be enabled by data bus mode selection during reset.

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

Table D-10 CSPAR1 Pin Assignments

CSPAR1 Field	Chip-Select Signal	Alternate Signal	Discrete Output
CS10PA[1:0]	$\overline{CS10}$	ADDR23[1]	ECLK
CS9PA[1:0]	$\overline{CS9}$	ADDR22[1]	PC6
CS8PA[1:0]	$\overline{CS8}$	ADDR21[1]	PC5
CS7PA[1:0]	$\overline{CS7}$	ADDR20[1]	PC4
CS6PA[1:0]	$\overline{CS6}$	ADDR19	PC3

NOTES:
1. On the CPU16, ADDR[23:20] follow the logic state of ADDR19 unless externally driven.

The reset state of DATA[7:3] determines whether pins controlled by CSPAR1 are initially configured as high-order address lines or chip-selects. **Table D-11** shows the correspondence between DATA[7:3] and the reset configuration of $\overline{CS}$[10:6]/ ADDR[23:19]. This register may be read or written at any time. After reset, software may enable one or more pins as discrete outputs.

Table D-11 Reset Pin Function of $\overline{CS}$[10:6]

Data Bus Pins at Reset					Chip-Select/Address Bus Pin Function				
DATA7	DATA6	DATA5	DATA4	DATA3	$\overline{CS10}$/ ADDR23	$\overline{CS9}$/ ADDR22	$\overline{CS8}$/ ADDR21	$\overline{CS7}$/ ADDR20	$\overline{CS6}$/ ADDR19
1	1	1	1	1	$\overline{CS10}$	$\overline{CS9}$	$\overline{CS8}$	$\overline{CS7}$	$\overline{CS6}$
1	1	1	1	0	$\overline{CS10}$	$\overline{CS9}$	$\overline{CS8}$	$\overline{CS7}$	ADDR19
1	1	1	0	X	$\overline{CS10}$	$\overline{CS9}$	$\overline{CS8}$	ADDR20	ADDR19
1	1	0	X	X	$\overline{CS10}$	$\overline{CS9}$	ADDR21	ADDR20	ADDR19
1	0	X	X	X	$\overline{CS10}$	ADDR22	ADDR21	ADDR20	ADDR19
0	X	X	X	X	ADDR23	ADDR22	ADDR21	ADDR20	ADDR19

D.2.18 Chip-Select Base Address Register Boot

CSBARBT — Chip-Select Base Address Register Boot **$YFFA48**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR 23	ADDR 22	ADDR 21	ADDR 20	ADDR 19	ADDR 18	ADDR 17	ADDR 16	ADDR 15	ADDR 14	ADDR 13	ADDR 12	ADDR 11	BLKSZ[2:0]		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

D.2.19 Chip-Select Base Address Registers

CSBAR[0:10] — Chip-Select Base Address Registers **$YFFA4C–$YFFA74**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR 23	ADDR 22	ADDR 21	ADDR 20	ADDR 19	ADDR 18	ADDR 17	ADDR 16	ADDR 15	ADDR 14	ADDR 13	ADDR 12	ADDR 11	BLKSZ[2:0]		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Freescale Semiconductor, Inc.

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

Each chip-select pin has an associated base address register. A base address is the lowest address in the block of addresses enabled by a chip select. CSBARBT contains the base address for selection of a boot memory device. Bit and field definitions for CSBARBT and CSBAR[0:10] are the same, but reset block sizes differ. These registers may be read or written at any time.

ADDR[23:11] — Base Address

This field sets the starting address of a particular chip-select's address space. The address compare logic uses only the most significant bits to match an address within a block. The value of the base address must be an integer multiple of the block size. Base address register diagrams show how base register bits correspond to address lines.

BLKSZ[2:0] — Block Size Field

This field determines the size of the block that is enabled by the chip-select.

Table D-12 shows bit encoding for the base address registers block size field.

Table D-12 Block Size Field Bit Encoding

BLKSZ[2:0]	Block Size	Address Lines Compared[1]
000	2 Kbytes	ADDR[23:11]
001	8 Kbytes	ADDR[23:13]
010	16 Kbytes	ADDR[23:14]
011	64 Kbytes	ADDR[23:16]
100	128 Kbytes	ADDR[23:17]
101	256 Kbytes	ADDR[23:18]
110	512 Kbytes	ADDR[23:19]
111	512 Kbytes	ADDR[23:20]

NOTES:
1. ADDR[23:20] are the same logic level as ADDR19 during normal operation.

D.2.20 Chip-Select Option Register Boot

CSORBT — Chip-Select Option Register Boot **$YFFA4A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	BYTE[1:0]		R/$\overline{W}$[1:0]		STRB	$\overline{DSACK}$[3:0]				SPACE[1:0]		IPL[2:0]			$\overline{AVEC}$
RESET:															
0	1	1	1	1	0	1	1	0	1	1	1	0	0	0	0

D.2.21 Chip-Select Option Registers

CSOR[0:10] — Chip-Select Option Registers **$YFFA4E–YFFA76**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	BYTE[1:0]		R/$\overline{W}$[1:0]		STRB	$\overline{DSACK}$[3:0]				SPACE[1:0]		IPL[2:0]			$\overline{AVEC}$
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

CSORBT and CSOR[0:10] contain parameters that support operations from external memory devices. Bit and field definitions for CSORBT and CSOR[0:10] are the same.

MODE — Asynchronous/Synchronous Mode

0 = Asynchronous mode is selected.

1 = Synchronous mode is selected, and used with ECLK peripherals.

In asynchronous mode, chip-select assertion is synchronized with $\overline{AS}$ and $\overline{DS}$.

In synchronous mode, the chip-select signal is asserted with ECLK.

BYTE[1:0] — Upper/Lower Byte Option

This field is used only when the chip-select 16-bit port option is selected in the pin assignment register. This allows the usage of two external 8-bit memory devices to be concatenated to form a 16-bit memory. **Table D-13** shows upper/lower byte options.

Table D-13 BYTE Field Bit Encoding

BYTE[1:0]	Description
00	Disable
01	Lower byte
10	Upper byte
11	Both bytes

R/$\overline{W}$[1:0]— Read/Write

This field causes a chip-select to be asserted only for a read, only for a write, or for both read and write. **Table D-14** shows the options.

Table D-14 Read/Write Field Bit Encoding

R/$\overline{W}$[1:0]	Description
00	Disable
01	Read only
10	Write only
11	Read/Write

STRB — Address Strobe/Data Strobe

This bit controls the timing for assertion of a chip-select in asynchronous mode only. Selecting address strobe causes the chip-select to be asserted synchronized with address strobe. Selecting data strobe causes the chip-select to be asserted synchronized with data strobe. Data strobe timing is used to create a write strobe when needed.

0 = Address strobe

1 = Data strobe

$\overline{DSACK}$[3:0] — Data Strobe Acknowledge

This field specifies the source of $\overline{DSACK}$ in asynchronous mode as internally generated or externally supplied. It also allows the user to adjust bus timing with internal $\overline{DSACK}$ generation by controlling the number of wait states that are inserted to optimize bus speed in a particular application. **Table D-15** shows the $\overline{DSACK}$[3:0] field encoding. The fast termination encoding (%1110) effectively corresponds to –1 wait states.

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

Table D-15 $\overline{\text{DSACK}}$ Field Encoding

$\overline{\text{DSACK}}$[3:0]	Clock Cycles Required Per Access	Wait States Inserted Per Access
0000	3	0
0001	4	1
0010	5	2
0011	6	3
0100	7	4
0101	8	5
0110	9	6
0111	10	7
1000	11	8
1001	12	9
1010	13	10
1011	14	11
1100	15	12
1101	16	13
1110	2	Fast Termination
1111	—	External $\overline{\text{DSACK}}$

External memories are purchased with guaranteed access times on speed (in nanoseconds). Table D-16 relates wait states selected by $\overline{\text{DSACK}}$[3:0] to the memory device access time.

NOTE

Table D-16 assumes a system configuration that minimizes power consumption and the number of chip-selects employed. Other access techniques can provide the same access times with slower memory devices, but require more chip-selects to be used and will subsequently increase system power consumption.

Table D-16 Memory Access Times at 16.78, 20.97, and 25.17 MHz

Speed	t_{cyc}	Fast Termination Access Time	0 Wait State	1 Wait State
16.78 MHz	62.5 ns	30.0 ns	95.0 ns	155.0 ns
20.97 MHz	50.0 ns	20.0 ns	70.0 ns	120.0 ns
25.17 MHz	40.0 ns	15.0 ns	55.0 ns	95.0 ns

Freescale Semiconductor, Inc.

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

SPACE[1:0] — Address Space Select

Use this option field to select an address space for chip-select assertion or to configure a chip-select as an interrupt acknowledge strobe for an external device. The CPU16 normally operates in supervisor mode only, but interrupt acknowledge cycles take place in CPU space. **Table D-17** shows address space bit encodings.

Table D-17 Address Space Bit Encodings

SPACE[1:0]	Address Space
00	CPU Space
01	User Space
10	Supervisor Space
11	Supervisor/User Space

IPL[2:0] — Interrupt Priority Level

When SPACE[1:0] is set for CPU space (%00), chip-select logic can be used as an interrupt acknowledge strobe for an external device. During an interrupt acknowledge cycle, the interrupt priority level is driven on address lines ADDR[3:1] and is then compared to the value in IPL[2:0]. If the values match, an interrupt acknowledge strobe will be generated on the particular chip-select pin, provided other option register conditions are met. **Table D-18** shows IPL[2:0] field encoding.

Table D-18 Interrupt Priority Level Field Encoding

IPL[2:0]	Interrupt Priority Level
000	Any Level[1]
001	1
010	2
011	3
100	4
101	5
110	6
111	7

NOTES:
1. Any level means that chip-select is asserted regardless of the level of the interrupt acknowledge cycle.

$\overline{\text{AVEC}}$ — Autovector Enable

This field selects one of two methods of acquiring an interrupt vector during an interrupt acknowledge cycle. This field is not applicable when SPACE[1:0] = %00.

0 = External interrupt vector enabled

1 = Autovector enabled

If the chip select is configured to trigger on an interrupt acknowledge cycle (SPACE[1:0] = %00) and the $\overline{\text{AVEC}}$ field is set to one, the chip-select automatically generates $\overline{\text{AVEC}}$ and completes the interrupt acknowledge cycle. Otherwise, the vector must be supplied by the requesting external device to complete the IACK read cycle.

Freescale Semiconductor, Inc.

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

D.2.22 Master Shift Registers

TSTMSRA — Test Module Master Shift Register A **$YFFA30**

Used for factory test only.

TSTMSRB — Test Module Master Shift Register B **$YFFA32**

Used for factory test only.

D.2.23 Test Module Shift Count Register

TSTSC — Test Module Shift Count **$YFFA34**

Used for factory test only.

D.2.24 Test Module Repetition Count Register

TSTRC — Test Module Repetition Count **$YFFA36**

Used for factory test only.

D.2.25 Test Module Control Register

CREG — Test Module Control Register **$YFFA38**

Used for factory test only.

D.2.26 Test Module Distributed Register

DREG — Test Module Distributed Register **$YFFA3A**

Used for factory test only.

Freescale Semiconductor, Inc.

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

D.3 Standby RAM Module

Table D-19 shows the SRAM address map.

Table D-19 SRAM Address Map

Address[1]	15 0
$YFFB00	RAM Module Configuration Register (RAMMCR)
$YFFB02	RAM Test Register (RAMTST)
$YFFB04	RAM Array Base Address Register High (RAMBAH)
$YFFB06	RAM Array Base Address Register Low (RAMBAL)

NOTES:
1. Y = M111, where M is the logic state of the module mapping (MM) bit in the SIMCR.

D.3.1 RAM Module Configuration Register

RAMMCR — RAM Module Configuration Register **$YFFB00**

15				11		9	8		0
STOP	0	0	0	RLCK	0	RASP[1:0]		NOT USED	
RESET:									
1	0	0	0	0	0	1	1		

STOP — Low-Power Stop Mode Enable
0 = SRAM operates normally.
1 = SRAM enters low-power stop mode.
This bit controls whether SRAM operates normally or enters low-power stop mode. In low-power stop mode, the array retains its contents, but cannot be read or written. This bit can be read or written at any time.

RLCK — RAM Base Address Lock
0 = SRAM base address registers can be written.
1 = SRAM base address registers are locked and cannot be modified.
RLCK defaults to zero on reset; it can be written once to a one, and may be read at any time.

RASP[1:0] — RAM Array Space
The RASP field limits access to the SRAM array in microcontrollers that support separate user and supervisor operating modes. RASP1 has no effect because the CPU16 operates in supervisor mode only. This bit may be read or written at any time. Refer to Table D-20.

Table D-20 SRAM Array Address Space Type

RASP[1:0]	Space
X0	Program and data accesses
X1	Program access only

Freescale Semiconductor, Inc.

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

D.3.2 RAM Test Register

RAMTST — RAM Test Register **$YFFB02**

Used for factory test only.

D.3.3 Array Base Address Register High

RAMBAH — Array Base Address Register High (Z1, Z2, Z3, and Z4) **$YFFB04**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								ADDR 23	ADDR 22	ADDR 21	ADDR 20	ADDR 19	ADDR 18	ADDR 17	ADDR 16
RESET:								0	0	0	0	0	0	0	0

D.3.4 Array Base Address Register Low

RAMBAL — Array Base Address Register Low (1K SRAM — Z1/Z4) **$YFFB06**

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

RAMBAL — Array Base Address Register Low (2K SRAM — Z2) **$YFFB06**

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

RAMBAL — Array Base Address Register Low (4K SRAM — Z3) **$YFFB06**

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

RAMBAH and RAMBAL specify the SRAM array base address in the system memory map. They can only be written while the SRAM is in low-power stop mode (STOP = 1, the default out of reset) and the base address lock is disabled (RLCK = 0, the default out of reset). This prevents accidental remapping of the array. Because the CPU16 drives ADDR[23:20] to the same logic level as ADDR19, the values of RAMBAH ADDR[23:20] must match the value of ADDR19 for the array to be accessible. These registers may be read at any time. RAMBAH[15:8] are unimplemented and will always read zero.

Freescale Semiconductor, Inc.

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

D.4 Masked ROM Module

The MRM is used only in the MC68HC16Z2 and the MC68HC16Z3. **Table D-21** shows the MRM address map.

The reset states shown for the MRM registers are for the generic (blank ROM) versions of the device. Several MRM register bit fields can be user-specified on a custom masked ROM device. Contact a Freescale sales representative for information on ordering a custom ROM device.

Table D-21 MRM Address Map

Address	15 0
$YFF820	Masked ROM Module Configuration Register (MRMCR)
$YFF822	Not Used
$YFF824	ROM Array Base Address Register High (ROMBAH)
$YFF826	ROM Array Base Address Register Low (ROMBAL)
$YFF828	Signature Register High (SIGHI)
$YFF82A	Signature Register Low (SIGLO)
$YFF82C	Not Used
$YFF82E	Not Used
$YFF830	ROM Bootstrap Word 0 (ROMBS0)
$YFF832	ROM Bootstrap Word 1 (ROMBS1)
$YFF834	ROM Bootstrap Word 2 (ROMBS2)
$YFF836	ROM Bootstrap Word 3 (ROMBS3)
$YFF838	Not Used
$YFF83A	Not Used
$YFF83C	Not Used
$YFF83E	Not Used

D.4.1 Masked ROM Module Configuration Register

MRMCR — Masked ROM Module Configuration Register **$YFF820**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP	0	0	$\overline{\text{BOOT}}$	LOCK	EMUL	ASPC[1:0]		WAIT[1:0]		NOT USED					
RESET:															
$\overline{\text{DATA14}}$	0	0	1	0	0	1	1	1	1						

STOP — Low-Power Stop Mode Enable

The reset state of the STOP bit is the complement of DATA14 state during reset. The ROM array base address cannot be changed unless the STOP bit is set.

0 = ROM array operates normally.

1 = ROM array operates in low-power stop mode. The ROM array cannot be read in this mode.

This bit may be read or written at any time.

Freescale Semiconductor, Inc.

$\overline{\text{BOOT}}$ — Boot ROM Control

Reset state of $\overline{\text{BOOT}}$ is specified at mask time. This is a read-only bit.

0 = ROM responds to bootstrap word locations during reset vector fetch.

1 = ROM does not respond to bootstrap word locations during reset vector fetch.

Bootstrap operation is overridden if STOP = 1 at reset.

LOCK — Lock Registers

The reset state of LOCK is specified at mask time. If the reset state of the LOCK is zero, it can be set once after reset to allow protection of the registers after initialization. Once the LOCK bit is set, it cannot be cleared again until after a reset. LOCK protects the ASPC and WAIT fields, as well as the ROMBAL and ROMBAH registers. ASPC, ROMBAL and ROMBAH are also protected by the STOP bit.

0 = Write lock disabled. Protected registers and fields can be written.

1 = Write lock enabled. Protected registers and fields cannot be written.

EMUL — Emulation Mode Control

0 = Normal ROM operation

1 = Accesses to the ROM array are forced external, allowing memory selected by the CSM pin to respond to the access.

Because the MC68HC16Z2 and the MC68HC16Z3 do not support ROM emulation mode, this bit should never be set.

ASPC[1:0] — ROM Array Space

The ASPC field limits access to the SRAM array in microcontrollers that support separate user and supervisor operating modes. ASPC1 has no effect because the CPU16 operates in supervisor mode only. This bit may be read or written at any time. The reset state of ASPC[1:0] is specified at mask time. **Table D-22** shows ASPC[1:0] encoding.

Table D-22 ROM Array Space Field

ASPC[1:0]	State Specified
X0	Program and data accesses
X1	Program access only

WAIT[1:0] — Wait States Field

WAIT[1:0] specifies the number of wait states inserted by the MRM during ROM array accesses. The reset state of WAIT[1:0] is user specified. The field can be written only if LOCK = 0 and STOP = 1. **Table D-23** shows the wait states field.

Table D-23 Wait States Field

WAIT[1:0]	Number of Wait States	Clocks per Transfer
00	0	3
01	1	4
10	2	5
11	–1	2

Freescale Semiconductor, Inc.

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

D.4.2 ROM Array Base Address Registers

ROMBAH — ROM Array Base Address Register High **$YFF824**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	ADDR 23	ADDR 22	ADDR 21	ADDR 20	ADDR 19	ADDR 18	ADDR 17	ADDR 16
RESET:								1	1	1	1	1	1	1	1

The reset value of the shaded bits is user specified, but the bits can be written after reset to change the base address. If the values of ROMBAH bits ADDR[23:20] do not match that of ADDR19, the CPU16 cannot access the ROM array.

ROMBAL — ROM Array Base Address Register Low **$YFF826**

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

ROMBAH and ROMBAL specify the ROM array base address. The reset state of these registers is specified at mask time. They can only be written when STOP = 1 and LOCK = 0. This prevents accidental remapping of the array. Because the 8-Kbyte ROM array in the MC68HC16Z2 and the MC68HC16Z3 must be mapped to an 8-Kbyte boundary, ROMBAL bits [12:0] always contain $0000. ROMBAH ADDR[15:8] read zero.

D.4.3 ROM Signature Registers High

RSIGHI — ROM Signature Register High **$YFF828**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED													RSP18	RSP17	RSP16
RESET:															
													0	0	0

RSIGLO — ROM Signature Register Low **$YFF82A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSP15	RSP14	RSP13	RSP12	RSP11	RSP10	RSP9	RSP8	RSP7	RSP6	RSP5	RSP4	RSP3	RSP2	RSP1	RSP0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Signature registers RSIGHI and RSIGLO contain a user-specified mask-programmed signature pattern. A user-specified signature algorithm provides the capability to verify ROM array contents.

Freescale Semiconductor, Inc.

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

D.4.4 ROM Bootstrap Words

ROMBS0 — ROM Bootstrap Word 0 **$YFF830**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED				ZK[3:0]				SK[3:0]				PK[3:0]			

ROMBS1 — ROM Bootstrap Word 1 **$YFF832**

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

ROMBS2 — ROM Bootstrap Word 2 **$YFF834**

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

ROMBS3 — ROM Bootstrap Word 3 **$YFF836**

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

Typically, CPU16 reset vectors reside in non-volatile memory and are fetched when the CPU16 comes out of reset. These four words can be used as reset vectors with the contents specified at mask time. The content of these words cannot be changed. On generic (blank ROM) MC68HC16Z2 and MC68HC16Z3 devices, ROMBS[0:3] are masked to $0000. When the ROM on the MC68HC16Z2 and MC68HC16Z3 is masked with customer specific code, ROMBS[0:3] respond to system addresses $00000 to $00006 during the reset vector fetch if $\overline{\text{BOOT}}$ = 0.

Freescale Semiconductor, Inc.

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

D.5 Analog-to-Digital Converter Module

Table D-24 ADC Module Address Map

Address[1]	15 8	7 0
$YFF700	ADC Module Configuration Register (ADCMCR)	
$YFF702	ADC Test Register (ADCTEST)	
$YFF704	Not Used	
$YFF706	Not Used	Port ADA Data Register (PORTADA)
$YFF708	Not Used	
$YFF70A	Control Register 0 (ADCTL0)	
$YFF70C	Control Register 1 (ADCTL1)	
$YFF70E	Status Register (ADCSTAT)	
$YFF710	Right-Justified Unsigned Result Register 0 (RJURR0)	
$YFF712	Right-Justified Unsigned Result Register 1 (RJURR1)	
$YFF714	Right-Justified Unsigned Result Register 2 (RJURR2)	
$YFF716	Right-Justified Unsigned Result Register 3 (RJURR3)	
$YFF718	Right-Justified Unsigned Result Register 4 (RJURR4)	
$YFF71A	Right-Justified Unsigned Result Register 5 (RJURR5)	
$YFF71C	Right-Justified Unsigned Result Register 6 (RJURR6)	
$YFF71E	Right-Justified Unsigned Result Register 7 (RJURR7)	
$YFF720	Left-Justified Signed Result Register 0 (LJSRR0)	
$YFF722	Left-Justified Signed Result Register 1 (LJSRR1)	
$YFF724	Left-Justified Signed Result Register 2 (LJSRR2)	
$YFF726	Left-Justified Signed Result Register 3 (LJSRR3)	
$YFF728	Left-Justified Signed Result Register 4 (LJSRR4)	
$YFF72A	Left-Justified Signed Result Register 5 (LJSRR5)	
$YFF72C	Left-Justified Signed Result Register 6 (LJSRR6)	
$YFF72E	Left-Justified Signed Result Register 7 (LJSRR7)	
$YFF730	Left-Justified Unsigned Result Register 0 (LJURR0)	
$YFF732	Left-Justified Unsigned Result Register 1 (LJURR1)	
$YFF734	Left-Justified Unsigned Result Register 2 (LJURR2)	
$YFF736	Left-Justified Unsigned Result Register 3 (LJURR3)	
$YFF738	Left-Justified Unsigned Result Register 4 (LJURR4)	
$YFF73A	Left-Justified Unsigned Result Register 5 (LJURR5)	
$YFF73C	Left-Justified Unsigned Result Register 6 (LJURR6)	
$YFF73E	Left-Justified Unsigned Result Register 7 (LJURR7)	

NOTES:
1. Y = M111, where M is the logic state of the MM bit in the SIMCR

Freescale Semiconductor, Inc.

D.5.1 ADC Module Configuration Register

ADCMCR — ADC Module Configuration Register **$YFF700**

15	14	13	12	8	7	6	0
STOP	FRZ		NOT USED		SUPV	NOT USED	
RESET:							
1	0	0			1		

ADCMCR controls ADC operation during low-power stop mode, background debug mode, and freeze mode.

STOP — Low-Power Stop Mode Enable

0 = Normal operation

1 = Low-power operation

STOP places the ADC in low-power state. Setting STOP aborts any conversion in progress. STOP is set to logic level one during reset, and may be cleared to logic level zero by the CPU16. Clearing STOP enables normal ADC operation. However, because analog circuitry bias current has been turned off, there is a period of recovery before output stabilization.

FRZ[1:0] — Freeze Assertion Response

The FRZ field determines ADC response to assertion of the FREEZE signal when the device is placed in background debug mode. Refer to **Table D-25**.

Table D-25 Freeze Encoding

FRZ[1:0]	Response
00	Ignore FREEZE, continue conversions
01	Reserved
10	Finish conversion in process, then freeze
11	Freeze immediately

SUPV — Supervisor/Unrestricted

This bit has no effect because the CPU16 always operates in supervisor mode.

D.5.2 ADC Test Register

ADCTEST — ADC Test Register **$YFF702**

Used for factory test only.

D.5.3 Port ADA Data Register

PORTADA — Port ADA Data Register **$YFF706**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PADA7	PADA6	PADA5	PADA4	PADA3	PADA2	PADA1	PADA0
RESET:															
								REFLECTS STATE OF THE INPUT PINS							

Port ADA is an input port that shares pins with the A/D converter inputs.

Freescale Semiconductor, Inc.

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

PADA[7:0] — Port ADA Data Pins

A read of PADA[7:0] returns the logic level of the port ADA pins. If an input is not at an appropriate logic level (that is, outside the defined levels), the read is indeterminate. Use of a port ADA pin for digital input does not preclude its simultaneous use as an analog input.

D.5.4 ADC Control Register 0

ADCTL0 — ADC Control Register 0 **$YFF70A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								RES10	STS[1:0]		PRS[4:0]				
RESET:								0	0	0	0	0	0	1	1

ADCTL0 is used to select 8- or 10-bit conversions, sample time, and ADC clock frequency. Writes to it have immediate effect.

RES10 — 10-Bit Resolution

0 = 8-bit conversion
1 = 10-bit conversion

Conversion results are appropriately aligned in result registers to reflect the number of bits.

STS[1:0] — Sample Time Selection

Total conversion time is the sum of initial sample time, transfer time, final sample time, and resolution time. Initial sample time is fixed at two ADC clocks. Transfer time is fixed at two ADC clocks. Resolution time is fixed at ten ADC clocks for an 8-bit conversion and twelve ADC clocks for a 10-bit conversion. Final sample time is determined by the STS[1:0] field. Refer to **Table D-26**.

Table D-26 Sample Time Selection

STS[1:0]	Sample Time
00	2 ADC Clock Periods
01	4 ADC Clock Periods
10	8 ADC Clock Periods
11	16 ADC Clock Periods

PRS[4:0] — Prescaler Rate Selection

The ADC clock is derived from the system clock by a programmable prescaler. ADC clock period is determined by the value of the PRS field in ADCTL0. The prescaler has two stages. The first stage is a 5-bit modulus counter. It divides the system clock by any value from two to 32 (PRS[4:0] = %00000 to %11111). The second stage is a divide-by-two circuit. Refer to **Table D-27**.

Freescale Semiconductor, Inc.

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

Table D-27 Prescaler Output

PRS[4:0]	ADC Clock	Minimum System Clock	Maximum System Clock
%00000	Reserved	—	—
%00001	System Clock/4	2.0 MHz	8.4 MHz
%00010	System Clock/6	3.0 MHz	12.6 MHz
%00011	System Clock/8	4.0 MHz	16.8 MHz
…	…	…	…
%11101	System Clock/60	30.0 MHz	—
%11110	System Clock/62	31.0 MHz	—
%11111	System Clock/64	32.0 MHz	—

D.5.5 ADC Control Register 1

ADCTL1 — ADC Control Register 1 **$YFF70C**

15 … 7	6	5	4	3	2	1	0
NOT USED	SCAN	MULT	S8CM	CD	CC	CB	CA
RESET:	0	0	0	0	0	0	0

ADCTL1 is used to initiate an A/D conversion and to select conversion modes and a conversion channel or channels. It can be read or written at any time. A write to ADCTL1 initiates a conversion sequence. If a conversion sequence is already in progress, a write to ADCTL1 aborts it and resets the SCF and CCF flags in the ADC status register.

SCAN — Scan Mode Selection
0 = Single conversion
1 = Continuous conversions
Length of conversion sequence(s) is determined by S8CM.

MULT — Multichannel Conversion
0 = Conversion sequence(s) run on a single channel selected by [CD:CA].
1 = Sequential conversions of four or eight channels selected by [CD:CA].
Length of conversion sequence(s) is determined by S8CM.

S8CM — Select Eight-Conversion Sequence Mode
0 = Four-conversion sequence
1 = Eight-conversion sequence
This bit determines the number of conversions in a conversion sequence. **Table D-28** displays the different ADC conversion modes.

Freescale Semiconductor, Inc.

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

Table D-28 ADC Conversion Mode

SCAN	MULT	S8CM	MODE
0	0	0	Single 4-Conversion Single-Channel Sequence
0	0	1	Single 8-Conversion Single-Channel Sequence
0	1	0	Single 4-Conversion Multichannel Sequence
0	1	1	Single 8-Conversion Multichannel Sequence
1	0	0	Multiple 4-Conversion Single-Channel Sequences
1	0	1	Multiple 8-Conversion Single-Channel Sequences
1	1	0	Multiple 4-Conversion Multichannel Sequences
1	1	1	Multiple 8-Conversion Multichannel Sequences

CD:CA — Channel Selection

Bits in this field select input channel or channels for A/D conversion.

Conversion mode determines which channel or channels are selected for conversion and which result registers are used to store conversion results. **Tables D-29** and **D-30** contain a summary of the effects of ADCTL1 bits and fields.

Freescale Semiconductor, Inc.

Table D-29 Single-Channel Conversions (MULT = 0)

S8CM	CD	CC	CB	CA	Input	Result Register[1]
0	0	0	0	0	AN0	RSLT[0:3]
0	0	0	0	1	AN1	RSLT[0:3]
0	0	0	1	0	AN2	RSLT[0:3]
0	0	0	1	1	AN3	RSLT[0:3]
0	0	1	0	0	AN4	RSLT[0:3]
0	0	1	0	1	AN5	RSLT[0:3]
0	0	1	1	0	AN6	RSLT[0:3]
0	0	1	1	1	AN7	RSLT[0:3]
0	1	0	0	0	Reserved	RSLT[0:3]
0	1	0	0	1	Reserved	RSLT[0:3]
0	1	0	1	0	Reserved	RSLT[0:3]
0	1	0	1	1	Reserved	RSLT[0:3]
0	1	1	0	0	V_{RH}	RSLT[0:3]
0	1	1	0	1	V_{RL}	RSLT[0:3]
0	1	1	1	0	$(V_{RH} - V_{RL}) / 2$	RSLT[0:3]
0	1	1	1	1	Test/Reserved	RSLT[0:3]
1	0	0	0	0	AN0	RSLT[0:7]
1	0	0	0	1	AN1	RSLT[0:7]
1	0	0	1	0	AN2	RSLT[0:7]
1	0	0	1	1	AN3	RSLT[0:7]
1	0	1	0	0	AN4	RSLT[0:7]
1	0	1	0	1	AN5	RSLT[0:7]
1	0	1	1	0	AN6	RSLT[0:7]
1	0	1	1	1	AN7	RSLT[0:7]
1	1	0	0	0	Reserved	RSLT[0:7]
1	1	0	0	1	Reserved	RSLT[0:7]
1	1	0	1	0	Reserved	RSLT[0:7]
1	1	0	1	1	Reserved	RSLT[0:7]
1	1	1	0	0	V_{RH}	RSLT[0:7]
1	1	1	0	1	V_{RL}	RSLT[0:7]
1	1	1	1	0	$(V_{RH} - V_{RL}) / 2$	RSLT[0:7]
1	1	1	1	1	Test/Reserved	RSLT[0:7]

NOTES:
1. Result register (RSLT) is either RJURRX, LJSRRX, or LJURRX, depending on the address read.

Freescale Semiconductor, Inc.

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

Table D-30 Multiple-Channel Conversions (MULT = 1)

S8CM	CD	CC	CB	CA	Input	Result Register[1]
0	0	0	X	X	AN0	RSLT0
					AN1	RSLT1
					AN2	RSLT2
					AN3	RSLT3
0	0	1	X	X	AN4	RSLT0
					AN5	RSLT1
					AN6	RSLT2
					AN7	RSLT3
0	1	0	X	X	Reserved	RSLT0
					Reserved	RSLT1
					Reserved	RSLT2
					Reserved	RSLT3
0	1	1	X	X	V_{RH}	RSLT0
					V_{RL}	RSLT1
					$(V_{RH} - V_{RL}) / 2$	RSLT2
					Test/Reserved	RSLT3
1	0	X	X	X	AN0	RSLT0
					AN1	RSLT1
					AN2	RSLT2
					AN3	RSLT3
					AN4	RSLT4
					AN5	RSLT5
					AN6	RSLT6
					AN7	RSLT7
1	1	X	X	X	Reserved	RSLT0
					Reserved	RSLT1
					Reserved	RSLT2
					Reserved	RSLT3
					V_{RH}	RSLT4
					V_{RL}	RSLT5
					$(V_{RH} - V_{RL}) / 2$	RSLT6
					Test/Reserved	RSLT7

NOTES:
1. Result register (RSLT) is either RJURRX, LJSRRX, or LJURRX, depending on the address read.

Freescale Semiconductor, Inc.

D.5.6 ADC Status Register

ADCSTAT — ADC Status Register **$YFF70E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCF	NOT USED				CCTR[2:0]			CCF[7:0]							
RESET:															
0					0	0	0	0	0	0	0	0	0	0	0

ADCSTAT contains information related to the status of a conversion sequence.

SCF — Sequence Complete Flag

0 = Sequence not complete

1 = Sequence complete

SCF is set at the end of the conversion sequence when SCAN is cleared, and at the end of the first conversion sequence when SCAN is set. SCF is cleared when ADCTL1 is written and a new conversion sequence begins.

CCTR[2:0] — Conversion Counter

This field reflects the contents of the conversion counter pointer in either four or eight count conversion sequence. The value corresponds to the number of the next result register to be written, and thus indicates which channel is being converted.

CCF[7:0] — Conversion Complete Flags

Each bit in this field corresponds to an A/D result register (for example, CCF7 to RSLT7). A bit is set when conversion for the corresponding channel is complete, and remains set until the associated result register is read.

D.5.7 Right Justified, Unsigned Result Register

RJURR — Right-Justified, Unsigned Result Register **$YFF710–$YFF71F**

15	10	9	8	7	6	5	4	3	2	1	0
NOT USED		10	10	8/10	8/10	8/10	8/10	8/10	8/10	8/10	8/10

Conversion result is unsigned right-justified data. Bits [9:0] are used for 10-bit resolution. For 8-bit conversions, bits [7:0] contain data and bits [9:8] are zero. Bits [15:10] always return zero when read.

D.5.8 Left Justified, Signed Result Register

LJSRR — Left Justified, Signed Result Register **$YFF720–$YFF72F**

15	14	13	12	11	10	9	8	7	6	5	0
8/10	8/10	8/10	8/10	8/10	8/10	8/10	8/10	10	10	NOT USED	

Conversion result is signed left-justified data. Bits [15:6] are used for 10-bit resolution. For 8-bit conversions, bits [15:8] contain data and bits [7:6] are zero. Although the ADC is unipolar, it is assumed that the zero point is halfway between low and high reference when this format is used ($V_{RH} – V_{RL}/2$). For positive input, bit 15 = 0. For negative input, bit 15 = 1. Bits [5:0] always return zero when read.

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

D.5.9 Left Justified, Unsigned Result Register

LJURR — Left Justified, Unsigned Result Register **$YFF730–$YFF73F**

15	14	13	12	11	10	9	8	7	6	5 … 0
8/10	8/10	8/10	8/10	8/10	8/10	8/10	8/10	10	10	NOT USED

Conversion result is unsigned left-justified data. Bits [15:6] are used for 10-bit resolution. For 8-bit conversions, bits [15:8] contain data and bits [7:6] are zero. Bits [5:0] always return zero when read.

Freescale Semiconductor, Inc.

D.6 Queued Serial Module

Table D-31 QSM Address Map

Address[1]	15 8	7 0
$YFFC00	QSM Module Configuration Register (QSMCR)	
$YFFC02	QSM Test Register (QTEST)	
$YFFC04	QSM Interrupt Level Register (QILR)	QSM Interrupt Vector Register (QIVR)
$YFFC06	Not Used	
$YFFC08	SCI Control 0 Register (SCCR0)	
$YFFC0A	SCI Control 1 Register (SCCR1)	
$YFFC0C	SCI Status Register (SCSR)	
$YFFC0E	SCI Data Register (SCDR)	
$YFFC10	Not Used	
$YFFC12	Not Used	
$YFFC14	Not Used	Port QS Data Register (PORTQS)
$YFFC16	Port QS Pin Assignment Register (PQSPAR)	Port QS Data Direction Register (DDRQS)
$YFFC18	SPI Control Register 0 (SPCR0)	
$YFFC1A	SPI Control Register 1 (SPCR1)	
$YFFC1C	SPI Control Register 2 (SPCR2)	
$YFFC1E	SPI Control Register 3 (SPCR3)	SPI Status Register (SPSR)
$YFFC20 – $YFFCFF	Not Used	
$YFFD00 – $YFFD1F	Receive RAM (RR[0:F])	
$YFFD20 – $YFFD3F	Transmit RAM (TR[0:F])	
$YFFD40 – $YFFD4F	Command RAM (CR[0:F])	

NOTES:
1. Y = M111, where M is the logic state of the module mapping (MM) bit in the SIMCR.

D.6.1 QSM Configuration Register

QSMCR — QSM Configuration Register **$YFFC00**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP	FRZ1	FRZ0	NOT USED					SUPV	NOT USED			IARB[3:0]			
RESET:															
0	0	0						1				0	0	0	0

QSMCR bits enable stop and freeze modes, and determine the arbitration priority of QSM interrupt requests.

Freescale Semiconductor, Inc.

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

STOP — Low-Power Stop Mode Enable

0 = QSM clock operates normally.

1 = QSM clock is stopped.

When STOP is set, the QSM enters low-power stop mode. The system clock input to the module is disabled. While STOP is set, only QSMCR reads and writes are guaranteed to be valid, but only writes to the QSPI RAM and other QSM registers are guaranteed valid. The SCI receiver and transmitter and the QSPI should be disabled before STOP is set. To stop the QSPI, set the HALT bit in SPCR3, wait until the HALTA flag is set, then set STOP. To stop the SCI, clear the TS and RE bits in SCCR1.

FRZ1 — FREEZE Assertion Response

FRZ1 determines what action is taken by the QSPI when the IMB FREEZE signal is asserted.

0 = Ignore the IMB FREEZE signal.

1 = Halt the QSPI on a transfer boundary.

FRZ0 — Not Implemented

Bits [12:8] — Not Implemented

SUPV — Supervisor/Unrestricted

This bit has no effect because the CPU16 in the MCU operates only in supervisor mode.

Bits [6:4] — Not Implemented

IARB[3:0] — Interrupt Arbitration ID

The IARB field is used to arbitrate between simultaneous interrupt requests of the same priority. Each module that can generate interrupt requests must be assigned a unique, non-zero IARB field value in order to request an interrupt.

D.6.2 QSM Test Register

QTEST — QSM Test Register **$YFFC02**

Used for factory test only.

D.6.3 QSM Interrupt Level Register/Interrupt Vector Register

QILR — QSM Interrupt Levels Register **$YFFC04**
QIVR — QSM Interrupt Vector Register **$YFFC05**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED		ILQSPI[2:0]			ILSCI[2:0]			INTV[7:0]							
RESET:															
		0	0	0	0	0	0	0	0	0	0	1	1	1	1

The values of ILQSPI[2:0] and ILSCI[2:0] in QILR determine the priority of QSPI and SCI interrupt requests. QIVR determines the value of the interrupt vector number the QSM supplies when it responds to an interrupt acknowledge cycle.

Freescale Semiconductor, Inc.

ILQSPI[2:0] — Interrupt Level for QSPI

When an interrupt request is made, the ILQSPI value determines the priority level of all QSPI interrupts. When a request is acknowledged, the QSM compares this value to a mask value supplied by the CPU16 to determine whether to respond. ILQSPI must have a value in the range $0 (interrupts disabled) to $7 (highest priority).

ILSCI[2:0] — Interrupt Level for SCI

When an interrupt request is made, the ILSCI value determines the priority level of all SCI interrupts. When a request is acknowledged, the QSM compares this value to a mask value supplied by the CPU16 to determine whether to respond. The field must have a value in the range $0 (interrupts disabled) to $7 (highest priority).

If ILQSPI[2:0] and ILSCI[2:0] have the same non-zero value, and both submodules simultaneously request interrupt service, the QSPI takes priority over the SCI.

INTV[7:0] — Interrupt Vector Number

The value of INTV[7:1] is used for both QSPI and SCI interrupt requests; the value of INTV0 used during an interrupt acknowledge cycle is supplied by the QSM. INTV0 is at logic level zero during an SCI interrupt and at logic level one during a QSPI interrupt. A write to INTV0 has no effect. Reads of INTV0 return a value of one. At reset, QIVR is initialized to $0F, the uninitialized interrupt vector number. To use interrupt-driven serial communication, a user-defined vector number must be written to QIVR.

D.6.4 SCI Control Register

SCCR0 — SCI Control Register 0 **$YFFC08**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED			SCBR[12:0]												
RESET:			0	0	0	0	0	0	0	0	0	0	1	0	0

SCCR0 contains the SCI baud rate selection field. Baud rate must be set before the SCI is enabled. The CPU16 can read and write SCCR0 at any time. Changing the value of SCCR0 bits during a transfer operation disrupts operation.

Bits [15:13] — Not Implemented

SCBR[12:0] — SCI Baud Rate

SCI baud rate is programmed by writing a 13-bit value to this field. Writing a value of zero to SCBR disables the baud rate generator. The baud clock rate is calculated as follows:

$$\text{SCI Baud Rate} = \frac{f_{sys}}{32 \times \text{SCBR[12:0]}}$$

or

$$\text{SCBR[12:0]} = \frac{f_{sys}}{32 \times \text{SCI Baud Rate Desired}}$$

where SCBR[12:0] is in the range of 1 to 8191.

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

Writing a value of zero to SCBR disables the baud rate generator. There are 8191 different bauds available. The baud value depends on the value for SCBR and the system clock, as used in the equation above. **Table D-32** shows possible baud rates for a 16.78 MHz system clock. The maximum baud rate with this system clock speed is 524 kbaud.

Table D-32 Examples of SCI Baud Rates

Nominal Baud Rate	Actual Baud Rate	Percent Error	Value of SCBR
500,00.00	524,288.00	4.86	1
38,400.00	37,449.14	–2.48	14
32,768.00	32,768.00	0.00	16
19,200.00	19,418.07	1.14	27
9,600.00	9,532.51	–0.70	55
4,800.00	4,809.98	0.21	109
2,400.00	2,404.99	0.21	218
1,200.00	1,199.74	–0.02	437
600.00	599.87	–0.02	874
300.00	299.94	–0.02	1,748
110.00	110.01	0.01	4,766
64.00	64.00	0.01	8,191

More accurate baud rates can be obtained by varying the system clock frequency with the VCO synthesizer. Each VCO speed increment adjusts the baud rate up or down by 1/64 or 1.56%.

D.6.5 SCI Control Register 1

SCCR1 — SCI Control Register 1 **$YFFC0A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED	LOOPS	WOMS	ILT	PT	PE	M	WAKE	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET:															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SCCR1 contains SCI configuration parameters, including transmitter and receiver enable bits, interrupt enable bits, and operating mode enable bits. SCCR0 can be read or written at any time. The SCI can modify the RWU bit under certain circumstances. Changing the value of SCCR1 bits during a transfer operation disrupts operation.

Bit 15 — Not Implemented

LOOPS — Loop Mode

0 = Normal SCI operation, no looping, feedback path disabled.
1 = Test SCI operation, looping, feedback path enabled.

WOMS — Wired-OR Mode for SCI Pins

0 = If configured as an output, TXD is a normal CMOS output.
1 = If configured as an output, TXD is an open-drain output.

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

ILT — Idle-Line Detect Type
0 = Short idle-line detect (start count on first one).
1 = Long idle-line detect (start count on first one after stop bit(s)).

PT — Parity Type
0 = Even parity
1 = Odd parity

PE — Parity Enable
0 = SCI parity disabled.
1 = SCI parity enabled.

M — Mode Select
0 = 10-bit SCI frame (1 start bit, 8 data bits, 1 stop bit)
1 = 11-bit SCI frame (1 start bit, 9 data bits, 1 stop bit)

WAKE — Wake-Up by Address Mark
0 = SCI receiver awakened by idle-line detection.
1 = SCI receiver awakened by address mark (last data bit set).

TIE — Transmit Interrupt Enable
0 = SCI TDRE interrupts disabled.
1 = SCI TDRE interrupts enabled.

TCIE — Transmit Complete Interrupt Enable
0 = SCI TC interrupts disabled.
1 = SCI TC interrupts enabled.

RIE — Receiver Interrupt Enable
0 = SCI RDRF and OR interrupts disabled.
1 = SCI RDRF and OR interrupts enabled.

ILIE — Idle-Line Interrupt Enable
0 = SCI IDLE interrupts disabled.
1 = SCI IDLE interrupts enabled.

TE — Transmitter Enable
0 = SCI transmitter disabled (TXD pin can be used as I/O).
1 = SCI transmitter enabled (TXD pin dedicated to SCI transmitter).

RE — Receiver Enable
0 = SCI receiver disabled.
1 = SCI receiver enabled.

RWU — Receiver Wake-Up
0 = Normal receiver operation (received data recognized).
1 = Wake-up mode enabled (received data ignored until receiver is awakened).

SBK — Send Break
0 = Normal operation
1 = Break frame(s) transmitted after completion of the current frame.

Freescale Semiconductor, Inc.

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

D.6.6 SCI Status Register

SCSR — SCI Status Register **$YFFC0C**

15 … 9	8	7	6	5	4	3	2	1	0
NOT USED	TDRE	TC	RDRF	RAF	IDLE	OR	NF	FE	PF
RESET:	1	1	0	0	0	0	0	0	0

SCSR contains flags that show SCI operating conditions. These flags are cleared either by SCI hardware or by a read/write sequence. The sequence consists of reading SCSR, then reading or writing SCDR.

If an internal SCI signal for setting a status bit comes after reading the asserted status bits, but before writing or reading SCDR, the newly set status bit is not cleared. SCSR must be read again with the bit set and SCDR must be read or written before the status bit is cleared.

A long-word read can consecutively access both SCSR and SCDR. This action clears receive status flag bits that were set at the time of the read, but does not clear TDRE or TC flags. Reading either byte of SCSR causes all 16 bits to be accessed, and any status bit already set in either byte is cleared on a subsequent read or write of SCDR.

Bits [15:9] — Not implemented

TDRE — Transmit Data Register Empty
- 0 = Transmit data register still contains data to be sent to the transmit serial shifter.
- 1 = A new character can now be written to the transmit data register.

TC — Transmit Complete
- 0 = SCI transmitter is busy.
- 1 = SCI transmitter is idle.

RDRF — Receive Data Register Full
- 0 = Receive data register is empty or contains previously read data.
- 1 = Receive data register contains new data.

RAF — Receiver Active
- 0 = SCI receiver is idle.
- 1 = SCI receiver is busy.

IDLE — Idle-Line Detected
- 0 = SCI receiver did not detect an idle-line condition.
- 1 = SCI receiver detected an idle-line condition.

OR — Overrun Error
- 0 = Receive data register is empty and can accept data from the receive serial shifter.
- 1 = Receive data register is full and cannot accept data from the receive serial shifter. Any data in the shifter is lost and RDRF remains set.

Freescale Semiconductor, Inc.

NF — Noise Error
0 = No noise detected in the received data.
1 = Noise detected in the received data.

FE — Framing Error
0 = No framing error detected in the received data.
1 = Framing error or break detected in the received data.

PF — Parity Error
0 = No parity error detected in the received data.
1 = Parity error detected in the received data.

D.6.7 SCI Data Register

SCDR — SCI Data Register **$YFFC0E**

15 … 9	8	7	6	5	4	3	2	1	0
NOT USED	R8/T8	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
RESET:	U	U	U	U	U	U	U	U	U

SCDR consists of two data registers located at the same address. The receive data register (RDR) is a read-only register that contains data received by the SCI serial interface. Data comes into the receive serial shifter and is transferred to RDR. The transmit data register (TDR) is a write-only register that contains data to be transmitted. Data is first written to TDR, then transferred to the transmit serial shifter, where additional format bits are added before transmission. R[7:0]/T[7:0] contain either the first eight data bits received when SCDR is read, or the first eight data bits to be transmitted when SCDR is written. R8/T8 are used when the SCI is configured for nine-bit operation. When the SCI is configured for 8-bit operation, R8/T8 has no meaning or effect.

D.6.8 Port QS Data Register

PORTQS — Port QS Data Register **$YFFC14**

15 … 8	7	6	5	4	3	2	1	0
NOT USED	PQS7	PQS6	PQS5	PQS4	PQS3	PQS2	PQS1	PQS0
RESET:	0	0	0	0	0	0	0	0

PORTQS latches I/O data. Writes drive pins defined as outputs. Reads return data present on the pins. To avoid driving undefined data, first write a byte to PORTQS, then configure DDRQS.

Freescale Semiconductor, Inc.

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

D.6.9 Port QS Pin Assignment Register/Data Direction Register

PQSPAR — PORT QS Pin Assignment Register **$YFFC16**
DDRQS — PORT QS Data Direction Register **$YFFC17**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED	PQSPA6	PQSPA5	PQSPA4	PQSPA3	NOT USED	PQSPA1	PQSPA0	DDQS7	DDQS6	DDQS5	DDQS4	DDQS3	DDQS2	DDQS1	DDQS0
RESET:															
	0	0	0	0		0	0	0	0	0	0	0	0	0	0

Clearing a bit in PQSPAR assigns the corresponding pin to general-purpose I/O. Setting a bit assigns the pin to the QSPI. PQSPAR does not affect operation of the SCI. **Table D-33** displays PQSPAR pin assignments.

Table D-33 PQSPAR Pin Assignments

PQSPAR Field	PQSPAR Bit	Pin Function
PQSPA0	0 1	PQS0 MISO
PQSPA1	0 1	PQS1 MOSI
—	— —	PQS2[1] SCK
PQSPA3	0 1	PQS3 PCS0/$\overline{SS}$
PQSPA4	0 1	PQS4 PCS1
PQSPA5	0 1	PQS5 PCS2
PQSPA6	0 1	PQS6 PCS3
—	— —	PQS7[2] TXD

NOTES:
1. PQS2 is a digital I/O pin unless the SPI is enabled (SPE set in SPCR1), in which case it becomes the QSPI serial clock SCK.
2. PQS7 is a digital I/O pin unless the SCI transmitter is enabled (TE set in SCCR1), in which case it becomes the SCI serial output TXD.

DDRQS determines whether pins configured for general-purpose I/O are inputs or outputs. Clearing a bit makes the corresponding pin an input; setting a bit makes the pin an output. DDRQS affects both QSPI function and I/O function.**Table D-34** shows the effect of DDRQS on QSM pin function.

Freescale Semiconductor, Inc.

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

Table D-34 Effect of DDRQS on QSM Pin Function

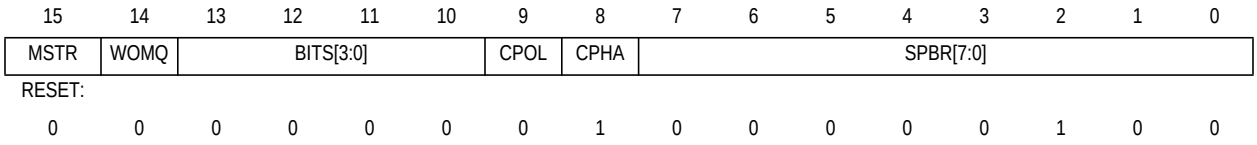
QSM Pin	Mode	DDRQS Bit	Bit State	Pin Function
MISO	Master	DDQS0	0	Serial data input to QSPI
			1	Disables data input
	Slave		0	Disables data output
			1	Serial data output from QSPI
MOSI	Master	DDQS1	0	Disables data output
			1	Serial data output from QSPI
	Slave		0	Serial data input to QSPI
			1	Disables data input
SCK[1]	Master	DDQS2	—	Clock output from QSPI
	Slave		—	Clock input to QSPI
PCS0/$\overline{SS}$	Master	DDQS3	0	Assertion causes mode fault
			1	Chip-select output
	Slave		0	QSPI slave select input
			1	Disables slave select Input
PCS[1:3]	Master	DDQS[4:6]	0	Disables chip-select output
			1	Chip-select outputs enabled
	Slave		0	No effect
			1	No effect
TXD[2]	—	DDQS7	X	Serial data output from SCI
RXD	—	None	NA	Serial data input to SCI

NOTES:
1. PQS2 is a digital I/O pin unless the SPI is enabled (SPE set in SPCR1), in which case it becomes the QSPI serial clock SCK.
2. PQS7 is a digital I/O pin unless the SCI transmitter is enabled (TE set in SCCR1), in which case it becomes the SCI serial data output TXD.

DDQS7 determines the direction of PQS7 only when the SCI transmitter is disabled. When the SCI transmitter is enabled, PQS7 is the TXD output.

D.6.10 QSPI Control Register 0

SPCR0 — QSPI Control Register 0 **$YFFC18**



SPCR0 contains parameters for configuring the QSPI and enabling various modes of operation. SPCR0 must be initialized before QSPI operation begins. Writing a new value to SPCR0 while the QSPI is enabled disrupts operation.

MSTR — Master/Slave Mode Select
0 = QSPI is a slave device.
1 = QSPI is the system master.

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

WOMQ — Wired-OR Mode for QSPI Pins

0 = Pins designated for output by DDRQS operate in normal mode.
1 = Pins designated for output by DDRQS operate in open-drain mode.

BITS[3:0] — Bits Per Transfer

In master mode, when BITSE is set in a command RAM byte, BITS[3:0] determines the number of data bits transferred. When BITSE is cleared, eight bits are transferred. Reserved values default to eight bits. In slave mode, the command RAM is not used and the setting of BITSE has no effect on QSPI transfers. Instead, the BITS[3:0] field determines the number of bits the QSPI will receive during each transfer before storing the received data.

Table D-35 shows the number of bits per transfer.

Table D-35 Bits Per Transfer

BITS[3:0]	Bits Per Transfer
0000	16
0001	Reserved
0010	Reserved
0011	Reserved
0100	Reserved
0101	Reserved
0110	Reserved
0111	Reserved
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

CPOL — Clock Polarity

0 = The inactive state of SCK is logic zero.
1 = The inactive state of SCK is logic one.

CPOL is used to determine the inactive state of the serial clock (SCK). It is used with CPHA to produce a desired clock/data relationship between master and slave devices.

CPHA — Clock Phase

0 = Data is captured on the leading edge of SCK and changed on the trailing edge of SCK.
1 = Data is changed on the leading edge of SCK and captured on the trailing edge of SCK

CPHA determines which edge of SCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce a desired clock/data relationship between master and slave devices.

Freescale Semiconductor, Inc.

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

SPBR[7:0] — Serial Clock Baud Rate

The QSPI uses a modulus counter to derive the SCK baud rate from the MCU system clock. Baud rate is selected by writing a value from two to 255 into SPBR[7:0]. The following equation determines the SCK baud rate:

$$\text{SCK Baud Rate} = \frac{f_{sys}}{2 \times \text{SPBR[7:0]}}$$

or

$$\text{SPBR[7:0]} = \frac{f_{sys}}{2 \times \text{SCK Baud Rate Desired}}$$

Giving SPBR[7:0] a value of zero or one disables the baud rate generator. SCK is disabled and assumes its inactive state value. No serial transfers occur. At reset, the SCK baud rate is initialized to one-eighth of the system clock frequency. SPBR has 254 active values. **Table D-36** lists several possible baud values and the corresponding SCK frequency based on a 16.78-MHz system clock.

Table D-36 Examples of SCK Frequencies

f_{sys}	Required Division Ratio	Value of SPBR	Actual SCK Frequency
16.78 MHz	4	2	4.19 MHz
	8	4	2.10 MHz
	16	8	1.05 MHz
	34	17	493 kHz
	168	84	100 kHz
	510	255	33 kHz

D.6.11 QSPI Control Register 1

SPCR1 — QSPI Control Register 1 **$YFFC1A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPE	DSCKL[6:0]							DTL[7:0]							
RESET:															
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

SPCR1 enables the QSPI and specifies transfer delays. SPCR1 must be written last during initialization because it contains SPE. Writing a new value to SPCR1 while the QSPI is enabled disrupts operation.

SPE — QSPI Enable

0 = QSPI is disabled. QSPI pins can be used for general-purpose I/O.
1 = QSPI is enabled. Pins allocated by PQSPAR are controlled by the QSPI.

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

DSCKL[6:0] — Delay before SCK

When the DSCK bit is set in a command RAM byte, this field determines the length of the delay from PCS valid to SCK transition. PCS can be any of the four peripheral chip-select pins. The following equation determines the actual delay before SCK:

$$\text{PCS to SCK Delay} = \frac{\text{DSCKL[6:0]}}{f_{sys}}$$

where DSCKL[6:0] is in the range of one to 127.

When DSCK is zero in a command RAM byte, then DSCKL[6:0] is not used. Instead, the PCS valid to SCK transition is one-half the SCK period.

DTL[7:0] — Length of Delay after Transfer

When the DT bit is set in a command RAM byte, this field determines the length of the delay after a serial transfer. The following equation is used to calculate the delay:

$$\text{Delay after Transfer} = \frac{32 \times \text{DTL[7:0]}}{f_{sys}}$$

where DTL is in the range of one to 255.

A zero value for DTL[7:0] causes a delay-after-transfer value of 8192 ÷ f_{sys}.

If DT is zero in a command RAM byte, a standard delay is inserted:

$$\text{Standard Delay after Transfer} = \frac{17}{f_{sys}}$$

Delay after transfer can be used to provide a peripheral deselect interval. A delay can also be inserted between consecutive transfers to allow serial A/D converters to complete conversion. This is controlled by the DT bit in a command RAM byte.

D.6.12 QSPI Control Register 2

SPCR2 — QSPI Control Register 2 **$YFFC1C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPIFIE	WREN	WRTO	0	ENDQP[3:0]				0	0	0	0	NEWQP[3:0]			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

SPCR2 contains QSPI queue pointers, wraparound mode control bits, and an interrupt enable bit. SPCR2 is buffered. New SPCR2 values become effective only after completion of the current serial transfer. Rewriting NEWQP in SPCR2 causes execution to restart at the designated location. Reads of SPCR2 return the value of the register, not the buffer.

SPIFIE — SPI Finished Interrupt Enable
0 = QSPI interrupts disabled.
1 = QSPI interrupts enabled.

WREN — Wrap Enable
0 = Wraparound mode disabled.
1 = Wraparound mode enabled.

WRTO — Wrap To
0 = Wrap to pointer address $0.
1 = Wrap to address in NEWQP.

Bit 12 — Not Implemented

ENDQP[3:0] — Ending Queue Pointer
This field contains the last QSPI queue address.

Bits [7:4] — Not Implemented

NEWQP[3:0] — New Queue Pointer Value
This field contains the first QSPI queue address.

D.6.13 QSPI Control Register 3

SPCR3 — QSPI Control Register **$YFFC1E**
SPSR — QSPI Status Register **$YFFC1F**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED					LOOPQ	HMIE	HALT	SPIF	MODF	HALTA	NOT USED	CPTQP[3:0]			
RESET:					0	0	0	0	0	0		0	0	0	0

SPCR3 contains the loop mode enable bit, halt and mode fault interrupt enable, and the halt control bit. SPCR3 must be initialized before QSPI operation begins. Writing a new value to SPCR3 while the QSPI is enabled disrupts operation. SPSR contains information concerning the current serial transmission.

Bits [15:11] — Not Implemented

LOOPQ — QSPI Loop Mode
0 = Feedback path disabled.
1 = Feedback path enabled.
LOOPQ controls feedback on the data serializer for testing.

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

HMIE — HALTA and MODF Interrupt Enable
0 = HALTA and MODF interrupts disabled.
1 = HALTA and MODF interrupts enabled.
HMIE enables interrupt requests generated by the HALTA status flag or the MODF status flag in SPSR.

HALT — Halt QSPI
0 = QSPI operates normally.
1 = QSPI is halted for subsequent restart.
When HALT is set, the QSPI stops on a queue boundary. It remains in a defined state from which it can later be restarted.

SPIF — QSPI Finished Flag
0 = QSPI is not finished.
1 = QSPI is finished.
SPIF is set after execution of the command at the address in ENDQP[3:0].

MODF — Mode Fault Flag
0 = Normal operation.
1 = Another SPI node requested to become the network SPI master while the QSPI was enabled in master mode.
The QSPI asserts MODF when the QSPI is in master mode (MSTR = 1) and the $\overline{SS}$ input pin is negated by an external driver.

HALTA — Halt Acknowledge Flag
0 = QSPI is not halted.
1 = QSPI is halted.
HALTA is set when the QSPI halts in response to setting the SPCR3 HALT bit.

Bit 4 — Not Implemented

CPTQP[3:0] — Completed Queue Pointer
CPTQP[3:0] points to the last command executed. It is updated when the current command is complete. When the first command in a queue is executing, CPTQP[3:0] contains either the reset value $0 or a pointer to the last command completed in the previous queue.

D.6.14 Receive Data RAM

RR[0:F] — Receive Data RAM **$YFFD00 – $YFFD1F**

Data received by the QSPI is stored in this segment. The CPU16 reads this segment to retrieve data from the QSPI. Data stored in receive RAM is right-justified. Unused bits in a receive queue entry are set to zero by the QSPI upon completion of the individual queue entry. Receive RAM data can be accessed using byte, word, or long-word addressing.

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

D.6.15 Transmit Data RAM

TR[0:F] — Transmit Data RAM **$YFFD20 – $YFFD3F**

Data that is to be transmitted by the QSPI is stored in this segment. The CPU16 normally writes one word of data into this segment for each queue command to be executed. Information to be transmitted must be written to the transmit data RAM in a right-justified format. The QSPI cannot modify information in the transmit data RAM. The QSPI copies the information to its data serializer for transmission. Information remains in the transmit RAM until overwritten.

D.6.16 Command RAM

CR[0:F] — Command RAM **$YFFD40 – $YFFD4F**

7	6	5	4	3	2	1	0
CONT	BITSE	DT	DSCK	PCS3	PCS2	PCS1	PCS0[1]
—	—	—	—	—	—	—	—
CONT	BITSE	DT	DSCK	PCS3	PCS2	PCS1	PCS0[1]
COMMAND CONTROL				PERIPHERAL CHIP SELECT			

NOTES:
1. The PCS0 bit represents the dual-function PCS0/$\overline{SS}$.

Command RAM is used by the QSPI when in master mode. The CPU16 writes one byte of control information to this segment for each QSPI command to be executed. The QSPI cannot modify information in command RAM.

Command RAM consists of 16 bytes. Each byte is divided into two fields. The peripheral chip-select field enables peripherals for transfer. The command control field provides transfer options.

A maximum of 16 commands can be in the queue. Queue execution proceeds from the address in NEWQP through the address in ENDQP (both of these fields are in SPCR2).

CONT — Continue
- 0 = Control of chip selects returned to PORTQS after transfer is complete.
- 1 = Peripheral chip selects remain asserted after transfer is complete. This allows for transfers greater than 16 bits to peripherals without negation of their chip-selects.

BITSE — Bits per Transfer Enable
- 0 = Eight bits
- 1 = Number of bits set in BITS field of SPCR0.

Freescale Semiconductor, Inc.

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

DT — Delay after Transfer

0 = Delay after transfer is 17 ÷ f_{sys}.
1 = SPCR1 DTL[7:0] specifies delay after transfer.

DSCK — PCS to SCK Delay

0 = PCS valid to SCK delay is one-half SCK.
1 = SPCR1 DSCKL[6:0] specifies delay from PCS valid to SCK.

PCS[3:0] — Peripheral Chip Select

Use peripheral chip-select bits to select one or more external devices for serial data transfers. More than one peripheral chip select may be activated at a time, and more than one peripheral chip can be connected to each PCS pin, provided proper fanout is observed. PCS0 shares a pin with the slave select ($\overline{SS}$) signal, which initiates slave mode serial transfers. If $\overline{SS}$ is taken low when the QSPI is in master mode, a mode fault occurs.

Freescale Semiconductor, Inc.

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

D.7 Multichannel Communication Interface Module

The MCCI is used only in the MC68HC16Z4 and the MC68CK16Z4. **Table D-37** shows the MCCI address map.

Table D-37 MCCI Address Map

Address[1]	15 8	7 0
$YFFC00	MCCI Module Configuration Register (MMCR)	
$YFFC02	MCCI Test Register (MTEST)	
$YFFC04	SCI Interrupt Level Register (ILSCI)	MCCI Interrupt Vector Register (MIVR)
$YFFC06	SPI Interrupt Level Register (ILSPI)	Not Used
$YFFC08	Not Used	MCCI Pin Assignment Register (MPAR)
$YFFC0A	Not Used	MCCI Data Direction Register (MDDR)
$YFFC0C	Not Used	MCCI Port Data Register (PORTMC)
$YFFC0E	Not Used	MCCI Port Pin State Register (PORTMCP)
$YFFC10 – $YFFC16	Not Used	
$YFFC18	SCIA Control Register 0 (SCCR0A)	
$YFFC1A	SCIA Control Register 1 (SCCR1A)	
$YFFC1C	SCIA Status Register (SCSRA)	
$YFFC1E	SCIA Data Register (SCDRA)	
$YFFC20 – $YFFC26	Not Used	
$YFFC28	SCIB Control Register 0 (SCCR0B)	
$YFFC2A	SCIB Control Register 1 (SCCR1B)	
$YFFC2C	SCIB Status Register (SCSRB)	
$YFFC2E	SCIB Data Register (SCDRB)	
$YFFC30 – $YFFC36	Not Used	
$YFFC38	SPI Control Register (SPCR)	
$YFFC3A	Not Used	
$YFFC3C	SPI Status Register (SPSR)	
$YFFC3E	SPI Data Register (SPDR)	

NOTES:
1. Y = M111, where M is the logic state of the module mapping (MM) bit in the SIMCR.

D.7.1 MCCI Module Configuration Register

MMCR — MCCI Module Configuration Register **$YFFC00**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP	NOT USED							SUPV	NOT USED			IARB[3:0]			
RESET:															
0								1				0	0	0	0

MMCR bits enable stop mode, establish the privilege level required to access certain MCCI registers, and determine the arbitration priority of MCCI interrupt requests.

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

STOP — Low-Power Stop Mode Enable
0 = MCCI clock operates normally.
1 = MCCI clock is stopped.
When STOP is set, the MCCI enters low-power stop mode. The system clock input to the module is disabled. While STOP is set, only MMCR reads and writes are guaranteed to be valid. Only writes to other MCCI registers are guaranteed valid. The SCI receiver and transmitter must be disabled before STOP is set. To stop the SPI, set the HALT bit in SPCR3, wait until the HALTA flag is set, then set STOP.

Bits [14:8] — Not Implemented

SUPV — Supervisor/Unrestricted
This bit has no effect because the CPU16 in the MCU operates only in supervisor mode.

Bits [6:4] — Not Implemented

IARB[3:0] — Interrupt Arbitration ID
The IARB field is used to arbitrate between simultaneous interrupt requests of the same priority. Each module that can generate interrupt requests must be assigned a unique, non-zero IARB field value.

D.7.2 MCCI Test Register

MTEST — MCCI Test Register **$YFFC02**

Used for factory test only.

D.7.3 SCI Interrupt Level Register/MCCI Interrupt Vector Register

ILSCI — SCI Interrupt Level Register **$YFFC04**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED		ILSCIB[2:0]			ILSCIA[2:0]			MIVR							
RESET:															
		0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bits [15:14] — Not Implemented

ILSCIA[2:0], ILSCIB[2:0] — Interrupt Level for SCIA, SCIB
The values of ILSCIA[2:0] and ILSCIB[2:0] in ILSCI determine the interrupt request levels of SCIA and SCIB interrupts, respectively. Program this field to a value from $0 (interrupts disabled) through $7 (highest priority).

Freescale Semiconductor, Inc.

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

D.7.4 MCCI Interrupt Vector Register

MIVR — MCCI Interrupt Vector Register **$YFFC05**

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

The MIVR determines which three vectors in the exception vector table are to be used for MCCI interrupts. The SPI and both SCI interfaces have separate interrupt vectors adjacent to one another. When initializing the MCCI, program INTV[7:2] so that INTV[7:0] correspond to three of the user-defined vectors ($40–$FF). INTV[1:0] are determined by the serial interface causing the interrupt, and are set by the MCCI.

At reset, MIVR is initialized to $0F, which corresponds to the uninitialized interrupt vector in the exception table.

INTV[7:2] — Interrupt Vector

INTV[7:2] are the six high-order bits of the three MCCI interrupt vectors for the MCCI, as programmed by the user.

INTV[1:0] — Interrupt Vector Source

INTV[1:0] are the two low-order bits of the three interrupt vectors for the MCCI. They are automatically set by the MCCI to indicate the source of the interrupt. Refer to **Table D-38**.

Table D-38 Interrupt Vector Sources

INTV[1:0]	Source of Interrupt
00	SCIA
01	SCIB
10	SPI

Writes to INTV0 and INTV1 have no meaning or effect. Reads of INTV0 and INTV1 return a value of one.

D.7.5 SPI Interrupt Level Register

ILSPI — SPI Interrupt Level Register **$YFFC06**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED		ILSPI[2:0]			NOT USED			NOT USED							
RESET:		0	0	0											

The ILSPI determines the priority level of interrupts requested by the SPI.

Bits [15:14] — Not Implemented

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

ILSPI[2:0] — Interrupt Level for SPI

ILSPI[2:0] determine the interrupt request levels of SPI interrupts. Program this field to a value from $0 (interrupts disabled) through $7 (highest priority). If the interrupt-request level programmed in this field matches the interrupt-request level programmed for one of the SCI interfaces and both request an interrupt simultaneously, the SPI is given priority.

Bits [10:8] — Not Implemented

D.7.6 MCCI Pin Assignment Register

MPAR — MCCI Pin Assignment Register **$YFFC08**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED												MPA3	NOT USED	MPA1	MPA0
RESET:															
0	0	0	0	0	0	0	0					0		0	0

The MPAR determines which of the SPI pins, with the exception of the SCK pin, are actually used by the SPI submodule, and which pins are available for general-purpose I/O. The state of SCK is determined by the SPI enable bit in SPCR1. Clearing a bit in MPAR assigns the corresponding pin to general-purpose I/O; setting a bit assigns the pin to the SPI. Refer to **Table D-39**.

Table D-39 MPAR Pin Assignments

MPAR Field	MPAR Bit	Pin Function
MPA0	0 1	PMC0 MISO
MPA1	0 1	PMC1 MOSI
—[1]	—	PMC2 SCK
MPA3	0 1	PMC3 $\overline{SS}$
—[1]	—	PMC4 RXDB
—[1]	—	PMC5 TXDB
—[1]	—	PMC6 RXDA
—[1]	—	PMC7 TXDA

NOTES:
1. MPA[7:4], MPA2 are not implemented.

Bits [15:8], [7:4], 2 — Not Implemented

SPI pins designated by the MPAR as general-purpose I/O are controlled only by MDDR and PORTMC. The SPI has no effect on these pins. The MPAR does not affect the operation of the SCI submodule.

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

D.7.7 MCCI Data Direction Register

MDDR — MCCI Data Direction Register **$YFFC0A**

15 … 8	7	6	5	4	3	2	1	0
NOT USED	DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0
RESET:	0	0	0	0	0	0	0	0

MDDR determines whether pins configured for general-purpose I/O are inputs or outputs. MDDR affects both SPI function and I/O function. During reset, all MCCI pins are configured as inputs. **Table D-40** shows the effect of MDDR on MCCI pin function.

Table D-40 Effect of MDDR on MCCI Pin Function

MCCI Pin	Mode	MDDR Bit	Bit State	Pin Function
MISO	Master	DDR0	0	Serial data input to SPI
			1	Disables data input
	Slave		0	Disables data output
			1	Serial data output from SPI
MOSI	Master	DDR1	0	Disables data output
			1	Serial data output from SPI
	Slave		0	Serial data input to SPI
			1	Disables data input
SCK[1]	Master	DDR2	—	Clock output from SPI
	Slave		—	Clock input to SPI
$\overline{SS}$	Master	DDR3	0	Assertion causes mode fault
			1	General-purpose I/O
	Slave		0	SPI slave-select input
			1	Disables slave-select input
RXDB[2]	—	DDR4	0	General-purpose I/O
			1	Serial data input to SCIB
TXDB[3]	—	DDR5	0	General-purpose I/O
			1	Serial data output from SCIB
RXDA	—	DDR6	0	General-purpose I/O
			1	Serial data input to SCIA
TXDA[3]	—	DDR7	0	General-purpose I/O
			1	Serial data output from SCIA

NOTES:
1. SCK is automatically assigned to the SPI whenever the SPI is enabled (when the SPE bit in the SPCR1 is set).
2. PMC4 and PMC6 function as general-purpose I/O pins when the corresponding RE bit in the SCI control register (SCCR0A or SCCR0B) is cleared.
3. PMC5 and PMC7 function as general-purpose I/O pins when the corresponding TE bit in the SCI control register (SCCR0A or SCCR0B) is cleared.

Freescale Semiconductor, Inc.

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

D.7.8 MCCI Port Data Registers

PORTMC — MCCI Port Data Register **$YFFC0C**
PORTMCP — MCCI Port Pin State Register **$YFFC0E**

15 … 9	8	7	6	5	4	3	2	1	0
NOT USED	PMC7	PMC6	PMC5	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0
RESET:	U	U	U	U	U	U	U	U	U

Two registers are associated with port MCCI, the MCCI general-purpose I/O port. Pins used for general-purpose I/O must be configured for that function. When using port MCCI as an output port, after configuring the pins as I/O, write the first byte to be output before writing to the MDDR. Afterwards, write to the MDDR to assign each I/O pin as either input or output. This outputs the value contained in register PORTMC for all pins defined as outputs. To output different data, write another byte to PORTMC.

Writes to PORTMC are stored in the internal data latch. If any bit of PORTMC is configured as discrete output, the value latched for that bit is driven onto the pin. Reads of PORTMC return the value of the pin only if the pin is configured as a discrete input. Otherwise, the value read is the value of the latch.

Reads of PORTMCP always return the state of the pins regardless of whether the pins are configured for input or output. Writes to PORTMCP have no effect.

D.7.9 SCI Control Register 0

SCCR0A — SCIA Control Register 0 **$YFFC18**
SCCR0B — SCIB Control Register 0 **$YFFC28**

15 … 13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED	SCBR[12:0]												
RESET:	0	0	0	0	0	0	0	0	0	0	1	0	0

SCCR0 contains the SCI baud rate selection field. Baud rate must be set before the SCI is enabled. The CPU16 can read and write SCCR0 at any time. Changing the value of SCCR0 bits during a transfer operation can disrupt the transfer.

Bits [15:13] — Not Implemented

SCBR[12:0] — SCI Baud Rate

SCI baud rate is programmed by writing a 13-bit value to this field. Writing a value of zero to SCBR disables the baud rate generator. Baud clock rate is calculated as follows:

Freescale Semiconductor, Inc.

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

$$\text{SCI Baud Rate} = \frac{f_{sys}}{32 \times \text{SCBR[12:0]}}$$

or

$$\text{SCBR[12:0]} = \frac{f_{sys}}{32 \times \text{SCI Baud Rate Desired}}$$

where SCBR[12:0] is in the range of one to 8191. Writing a value of zero to SCBR disables the baud rate generator. There are 8191 different baud rates available. The baud value depends on the value for SCBR and the system clock, as used in the equation above. **Table D-41** shows possible baud rates for a 16.78-MHz system clock. The maximum baud rate with this system clock speed is 524 kbaud.

Table D-41 Examples of SCI Baud Rates

Nominal Baud Rate	Actual Baud Rate	Percent Error	Value of SCBR
500,00.00	524,288.00	4.86	1
38,400.00	37,449.14	–2.48	14
32,768.00	32,768.00	0.00	16
19,200.00	19,418.07	1.14	27
9,600.00	9,532.51	–0.70	55
4,800.00	4,809.98	0.21	109
2,400.00	2,404.99	0.21	218
1,200.00	1,199.74	–0.02	437
600.00	599.87	–0.02	874
300.00	299.94	–0.02	1,748
110.00	110.01	0.01	4,766
64.00	64.00	0.01	8,191

D.7.10 SCI Control Register 1

SCCR1A — SCIA Control Register 1 **$YFFC1A**
SCCR1B — SCIB Control Register 1 **$YFFC2A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED	LOOPS	WOMS	ILT	PT	PE	M	WAKE	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET:															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SCCR1 contains SCI configuration parameters, including transmitter and receiver enable bits, interrupt enable bits, and operating mode enable bits. SCCR0 can be read or written at any time. The SCI can modify the RWU bit under certain circumstances. Changing the value of SCCR1 bits during a transfer operation can disrupt the transfer.

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

Bit 15 — Not Implemented

LOOPS — Loop Mode
0 = Normal SCI operation, no looping, feedback path disabled.
1 = Test SCI operation, looping, feedback path enabled.
The LOOPS bit in SCCR1 controls a feedback path on the data serial shifter. When LOOPS is set, SCI transmitter output is fed back into the receive serial shifter. The TXD pin is asserted (idle line). Both transmitter and receiver must be enabled prior to entering loop mode.

WOMS — Wired-OR Mode for SCI Pins
0 = If configured as an output, TXD is a normal CMOS output.
1 = If configured as an output, TXD is an open-drain output.

ILT — Idle-Line Detect Type
0 = Short idle-line detect (start count on first one).
1 = Long idle-line detect (start count on first one after stop bit(s)).

PT — Parity Type
0 = Even parity
1 = Odd parity

PE — Parity Enable
0 = SCI parity disabled.
1 = SCI parity enabled.

M — Mode Select
0 = 10-bit SCI frame (1 start bit, 8 data bits, 1 stop bit)
1 = 11-bit SCI frame (1 start bit, 9 data bits, 1 stop bit)

WAKE — Wake-Up by Address Mark
0 = SCI receiver awakened by idle-line detection.
1 = SCI receiver awakened by address mark (last data bit set).

TIE — Transmit Interrupt Enable
0 = SCI TDRE interrupts disabled.
1 = SCI TDRE interrupts enabled.

TCIE — Transmit Complete Interrupt Enable
0 = SCI TC interrupts disabled.
1 = SCI TC interrupts enabled.

RIE — Receiver Interrupt Enable
0 = SCI RDRF and OR interrupts disabled.
1 = SCI RDRF and OR interrupts enabled.

ILIE — Idle-Line Interrupt Enable
0 = SCI IDLE interrupts disabled.
1 = SCI IDLE interrupts enabled.

Freescale Semiconductor, Inc.

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

TE — Transmitter Enable
0 = SCI transmitter disabled (TXD pin can be used as I/O).
1 = SCI transmitter enabled (TXD pin dedicated to SCI transmitter).

RE — Receiver Enable
0 = SCI receiver disabled.
1 = SCI receiver enabled.

RWU — Receiver Wake-Up
0 = Normal receiver operation (received data recognized).
1 = Wake-up mode enabled (received data ignored until receiver is awakened).

SBK — Send Break
0 = Normal operation
1 = Break frame(s) transmitted after completion of the current frame.

D.7.11 SCI Status Register

SCSRA — SCIA Status Register **$YFFC1C**
SCSRB — SCIB Status Register **$YFFC2C**

15 … 9	8	7	6	5	4	3	2	1	0
NOT USED	TDRE	TC	RDRF	RAF	IDLE	OR	NF	FE	PF
RESET:	1	1	0	0	0	0	0	0	0

SCSR contains flags that show SCI operating conditions. These flags are cleared either by SCI hardware or by a read/write sequence. The sequence consists of reading SCSR, then reading or writing SCDR.

If an internal SCI signal for setting a status bit comes after reading the asserted status bits, but before writing or reading SCDR, the newly set status bit is not cleared. SCSR must be read again with the bit set and SCDR must be read or written before the status bit is cleared.

A long-word read can consecutively access both SCSR and SCDR. This action clears receive status flag bits that were set at the time of the read, but does not clear TDRE or TC flags. Reading either byte of SCSR causes all 16 bits to be accessed, and any status bit already set in either byte is cleared on a subsequent read or write of SCDR.

Bits [15:9] — Not Implemented

TDRE — Transmit Data Register Empty
0 = Transmit data register still contains data to be sent to the transmit serial shifter.
1 = A new character can now be written to the transmit data register.

TC — Transmit Complete
0 = SCI transmitter is busy.
1 = SCI transmitter is idle.

Freescale Semiconductor, Inc.

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

RDRF — Receive Data Register Full
0 = Receive data register is empty or contains previously read data.
1 = Receive data register contains new data.

RAF — Receiver Active
0 = SCI receiver is idle.
1 = SCI receiver is busy.

IDLE — Idle-Line Detected
0 = SCI receiver did not detect an idle-line condition.
1 = SCI receiver detected an idle-line condition.

OR — Overrun Error
0 = Receive data register is empty and can accept data from the receive serial shifter.
1 = Receive data register is full and cannot accept data from the receive serial shifter. Any data in the shifter is lost and RDRF remains set.

NF — Noise Error
0 = No noise detected in the received data.
1 = Noise detected in the received data.

FE — Framing Error
0 = No framing error detected in the received data.
1 = Framing error or break detected in the received data.

PF — Parity Error
0 = No parity error detected in the received data.
1 = Parity error detected in the received data.

D.7.12 SCI Data Register

SCDRA — SCIA Data Register **$YFFC1E**
SCDRB — SCIB Data Register **$YFFC2E**

15 … 9	8	7	6	5	4	3	2	1	0
NOT USED	R8/T8	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
RESET:	U	U	U	U	U	U	U	U	U

SCDR consists of two data registers located at the same address. The receive data register (RDR) is a read-only register that contains data received by the SCI serial interface. Data comes into the receive serial shifter and is transferred to RDR. The transmit data register (TDR) is a write-only register that contains data to be transmitted. Data is first written to TDR, then transferred to the transmit serial shifter, where additional format bits are added before transmission. R[7:0]/T[7:0] contain either the first eight data bits received when SCDR is read, or the first eight data bits to be transmitted when SCDR is written. R8/T8 are used when the SCI is configured for nine-bit operation. When the SCI is configured for 8-bit operation, R8/T8 have no meaning or effect.

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

D.7.13 SPI Control Register

SPCR — SPI Control Register **$YFFC38**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPIE	SPE	WOMP	MSTR	CPOL	CPHA	LSBF	SIZE	SPBR[7:0]							
RESET:															
0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0

The SPCR contains parameters for configuring the SPI. The register can be read or written at any time.

SPIE — SPI Interrupt Enable
- 0 = SPI interrupts disabled.
- 1 = SPI interrupts enabled.

SPE — SPI Enable
- 0 = SPI is disabled.
- 1 = SPI is enabled.

WOMP — Wired-OR Mode for SPI Pins
- 0 = Outputs have normal CMOS drivers.
- 1 = Pins designated for output by MDDR have open-drain drivers, regardless of whether the pins are used as SPI outputs or for general-purpose I/O, and regardless of whether the SPI is enabled.

MSTR — Master/Slave Mode Select
- 0 = SPI is a slave device.
- 1 = SPI is system master.

CPOL — Clock Polarity
- 0 = The inactive state value of SCK is logic level zero.
- 1 = The inactive state value of SCK is logic level one.

CPOL is used to determine the inactive state of the serial clock (SCK). It is used with CPHA to produce a desired clock/data relationship between master and slave devices.

CPHA — Clock Phase
- 0 = Data captured on the leading edge of SCK and changed on the trailing edge of SCK.
- 1 = Data is changed on the leading edge of SCK and captured on the trailing edge of SCK.

CPHA determines which edge of SCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce a desired clock/data relationship between master and slave devices.

LSBF — Least Significant Bit First
- 0 = Serial data transfer starts with LSB.
- 1 = Serial data transfer starts with MSB.

Freescale Semiconductor, Inc.

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

SIZE — Transfer Data Size
0 = 8-bit data transfer.
1 = 16-bit data transfer.

SPBR[7:0] — Serial Clock Baud Rate
The SPI uses a modulus counter to derive the SCK baud rate from the MCU system clock. Baud rate is selected by writing a value from 2 to 255 into SPBR[7:0].

The following expressions apply to SCK baud rate:

$$\text{SCK Baud Rate} = \frac{f_{sys}}{2 \times \text{SPBR[7:0]}}$$

or

$$\text{SPBR[7:0]} = \frac{f_{sys}}{2 \times \text{SCK Baud Rate Desired}}$$

Giving SPBR[7:0] a value of zero or one disables SCK (disable state determined by CPOL). At reset, the SCK baud rate is initialized to one-eighth of the system clock frequency.

D.7.14 SPI Status Register

SPSR — SPI Status Register **$YFFC3C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPIF	WCOL	0	MODF	0	0	0	0	0	0	0	0	0	0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPSR contains information concerning the current serial transmission. Only the SPI can set bits in SPSR. The CPU16 reads SPSR to obtain SPI status information and writes it to clear status flags.

SPIF — SPI Finished Flag
0 = SPI is not finished.
1 = SPI is finished.

WCOL — Write Collision
0 = No attempt to write to the SPDR happened during the serial transfer.
1 = Write collision occurred.
Clearing WCOL is accomplished by reading the SPSR while WCOL is set and then either reading the SPDR prior to SPIF being set, or reading or writing the SPDR after SPIF is set.

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

MODF — Mode Fault Flag

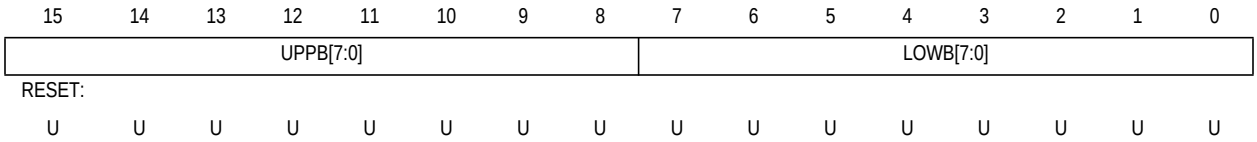
0 = Normal operation.

1 = Another SPI node requested to become the network SPI master while the SPI was enabled in master mode ($\overline{SS}$ input taken low).

The SPI asserts MODF when the SPI is in master mode (MSTR = 1) and the $\overline{SS}$ input pin is negated by an external driver.

D.7.15 SPI Data Register

SPDR — SPI Data Register **$YFFC3E**



UPPB — Upper Byte

In 16-bit transfer mode, the upper byte contains the most significant eight bits of the transmitted or received data. Bit 15 of the SPDR is the MSB of the 16-bit data.

LOWB — Lower Byte

In 8-bit transfer mode, the lower byte contains the transmitted or received data. MSB in 8-bit transfer mode is bit 7 of the SPDR. In 16-bit transfer mode, the lower byte holds the least significant eight bits of the data.

Freescale Semiconductor, Inc.

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

D.8 General-Purpose Timer

Table D-42 GPT Address Map

Address[1]	15 8	7 0
$YFF900	GPT Module Configuration Register (GPTMCR)	
$YFF902	GPT Module Test Register (GPTMTR)	
$YFF904	GPT Interrupt Configuration Register (ICR)	
$YFFE06	Port GP Data Direction Register (DDRGP)	Port GP Data Register (PORTGP)
$YFF908	Output Compare 1 Action Mask Register (OC1M)	Output Compare 1 Action Data Register (OC1D)
$YFF90A	Timer Counter Register (TCNT)	
$YFF90C	Pulse Accumulator Control Register (PACTL)	Pulse Accumulator Counter Register (PACNT)
$YFF90E	Timer Input Capture Register 1 (TIC1)	
$YFF910	Timer Input Capture Register 2 (TIC2)	
$YFF912	Timer Input Capture Register 3 (TIC3)	
$YFF914	Timer Output Compare Register 1 (TOC1)	
$YFF916	Timer Output Compare Register 2 (TOC2)	
$YFF918	Timer Output Compare Register 3 (TOC3)	
$YFF91A	Timer Output Compare Register 4 (TOC4)	
$YFF91C	Timer Input Capture 4/Output Compare Register 5 (TI4/O5)	
$YFF91E	Timer Control Register 1 (TCTL1)	Timer Control Register 2 (TCTL2)
$YFF920	Timer Mask Register 1 (TMSK1)	Timer Mask Register 2 (TMSK2)
$YFF922	Timer Flag Register 1 (TFLG1)	Timer Flag Register 2 (TFLG2)
$YFF924	Compare Force Register (CFORC)	PWM Control Register C (PWMC)
$YFF926	PWM Control Register A (PWMA)	PWM Control Register B (PWMB)
$YFF928	PWM Count Register (PWMCNT)	
$YFF92A	PWM Buffer Register A (PWMBUFA)	PWM Buffer Register B (PWMBUFB)
$YFF92C	GPT Prescaler Register (PRESCL)	
$YFF92E – $YFF93F	Reserved	

NOTES:
1. Y = M111, where M is the logic state of the MM bit in the SIMCR.

D.8.1 GPT Module Configuration Register

GPTMCR — GPT Module Configuration Register **$YFF900**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP	FRZ1	FRZ0	STOPP	INCP	0	0	0	SUPV	0	0	0	IARB			
RESET:															
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

The GPTMCR contains parameters for configuring the GPT.

Freescale Semiconductor, Inc.

STOP — Stop Clocks
0 = GPT clock operates normally.
1 = GPT clock is stopped.

FRZ1 — Not Implemented

FRZ0 — FREEZE Assertion Response
0 = Ignore IMB FREEZE signal.
1 = FREEZE the current state of the GPT.

STOPP — Stop Prescaler
0 = Normal operation.
1 = Stop prescaler and pulse accumulator from incrementing. Ignore changes to input pins.

INCP — Increment Prescaler
0 = Has no effect.
1 = If STOPP is asserted, increment prescaler once and clock input synchronizers once.

SUPV — Supervisor/Unrestricted Data Space
This bit has no effect because the CPU16 always operates in supervisor mode.

IARB[3:0] — Interrupt Arbitration ID
The IARB field is used to arbitrate between simultaneous interrupt requests of the same priority. Each module that can generate interrupt requests must be assigned a unique, non-zero IARB field value.

D.8.2 GPT Test Register

GPTMTR — GPT Module Test Register **$YFF902**

Used for factory test only.

D.8.3 GPT Interrupt Configuration Register

ICR — GPT Interrupt Configuration Register **$YFF904**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPA[3:0]				0	IPL[2:0]			IVBA[3:0]				0	0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICR fields determine internal and external interrupt priority, and provide the upper nibble of the interrupt vector number supplied to the CPU when an interrupt is acknowledged.

IPA[3:0] — Interrupt Priority Adjust
This field specifies which GPT interrupt source is given highest internal priority. Refer to **Table D-43**.

Freescale Semiconductor, Inc.

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

Table D-43 GPT Interrupt Sources

Name	Source Number	Source	Vector Number
—	0000	Adjusted Channel	IVBA : 0000
IC1	0001	Input Capture 1	IVBA : 0001
IC2	0010	Input Capture 2	IVBA : 0010
IC3	0011	Input Capture 3	IVBA : 0011
OC1	0100	Output Compare 1	IVBA : 0100
OC2	0101	Output Compare 2	IVBA : 0101
OC3	0110	Output Compare 3	IVBA : 0110
OC4	0111	Output Compare 4	IVBA : 0111
IC4/OC5	1000	Input Capture 4/Output Compare 5	IVBA : 1000
TO	1001	Timer Overflow	IVBA : 1001
PAOV	1010	Pulse Accumulator Overflow	IVBA : 1010
PAI	1011	Pulse Accumulator Input	IVBA : 1011

IPL[2:0] — Interrupt Priority Level
This field specifies the priority level of interrupts generated by the GPT.

IVBA[3:0] — Interrupt Vector Base Address
Most significant nibble of interrupt vector numbers generated by the GPT. Refer to **Table D-43**.

D.8.4 Port GP Data Direction Register/Data Register

DDRGP/PORTGP — Port GP Data Direction Register/Data Register **$YFF906**

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

When GPT pins are used as an 8-bit port, DDRGP determines whether pins are input or output and PORTGP holds the 8-bit data.

DDGP[7:0] — Port GP Data Direction Register
0 = Input only
1 = Output

D.8.5 OC1 Action Mask Register/Data Register

OC1M/OC1D — OC1 Action Mask Register/OC1 Action Data Register **$YFF908**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC1M[5:1]					0	0	0	OC1D[5:1]					0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All OC outputs can be controlled by the action of OC1. OC1M contains a mask that determines which pins are affected. OC1D determines what the outputs are.

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

OC1M[5:1] — OC1 Mask Field
OC1M[5:1] correspond to OC[5:1].
0 = Corresponding output compare pin is not affected by OC1 compare.
1 = Corresponding output compare pin is affected by OC1 compare.

OC1D[5:1] — OC1 Data Field
OC1D[5:1] correspond to OC[5:1].
0 = If OC1 mask bit is set, clear the corresponding output compare pin on OC1 match.
1 = If OC1 mask bit is set, the set corresponding output compare pin on OC1 match.

D.8.6 Timer Counter Register

TCNT — Timer Counter Register **$YFF90A**

TCNT is the 16-bit free-running counter associated with the input capture, output compare, and pulse accumulator functions of the GPT module.

D.8.7 Pulse Accumulator Control Register/Counter

PACTL/PACNT — Pulse Accumulator Control Register/Counter **$YFF90C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAIS	PAEN	PAMOD	PEDGE	PCLKS	I4/O5	PACLK[1:0]		PULSE ACCUMULATOR COUNTER							
RESET:															
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PACTL enables the pulse accumulator and selects either event counting or gated mode. In event counting mode, PACNT is incremented each time an event occurs. In gated mode, it is incremented by an internal clock.

PAIS — PAI Pin State (Read Only)

PAEN — Pulse Accumulator Enable
0 = Pulse accumulator disabled.
1 = Pulse accumulator enabled.

PAMOD — Pulse Accumulator Mode
0 = External event counting.
1 = Gated time accumulation.

PEDGE — Pulse Accumulator Edge Control
The effects of PAMOD and PEDGE are shown in **Table D-44**.

Freescale Semiconductor, Inc.

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

Table D-44 PAMOD and PEDGE Effects

PAMOD	PEDGE	Effect
0	0	PAI falling edge increments counter
0	1	PAI rising edge increments counter
1	0	Zero on PAI inhibits counting
1	1	One on PAI inhibits counting

PCLKS — PCLK Pin State (Read Only)

I4/O5 — Input Capture 4/Output Compare 5
0 = Output compare 5 enabled
1 = Input capture 4 enabled

PACLK[1:0] — Pulse Accumulator Clock Select (Gated Mode)
Table D-45 shows the PACLK[1:0] bit field effects.

Table D-45 PACLK[1:0] Effects

PACLK[1:0]	Pulse Accumulator Clock Selected
00	System clock divided by 512
01	Same clock used to increment TCNT
10	TOF flag from TCNT
11	External clock, PCLK

PACNT — Pulse Accumulator Counter
Eight-bit read/write counter used for external event counting or gated time accumulation.

D.8.8 Input Capture Registers 1–3

TIC[1:3] — Input Capture Registers 1–3 **$YFF90E – $YFF912**

The input capture registers are 16-bit read-only registers used to latch the value of TCNT when a specified transition is detected on the corresponding input capture pin. They are reset to $FFFF.

D.8.9 Output Compare Registers 1–4

TOC[1:4] — Output Compare Registers 1–4 **$YFF914 – $YFF91A**

The output compare registers are 16-bit read/write registers which can be used as output waveform controls or as elapsed time indicators. For output compare functions, they are written to a desired match value and compared against TCNT to control specified pin actions. They are reset to $FFFF.

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

D.8.10 Input Capture 4/Output Compare 5 Register

TI4/O5 — Input Capture 4/Output Compare 5 Register **$YFF91C**

This register serves either as input capture register 4 or output compare register 5, depending on the state of I4/O5 in PACTL. It is reset to $FFFF.

D.8.11 Timer Control Registers 1 and 2

TCTL1/TCTL2 — Timer Control Registers 1–2 **$YFF91E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM5	OL5	OM4	OL4	OM3	OL3	OM2	OL2	EDG4B	EDG4A	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCTL1 determines output compare mode and output logic level. TCTL2 determines the type of input capture to be performed.

OM/OL[5:2] — Output Compare Mode Bits and Output Compare Level Bits

Each pair of bits specifies an action to be taken when output comparison is successful. Refer to **Table D-46**.

Table D-46 OM/OL[5:2] Effects

OM/OL[5:2]	Action Taken
00	Timer disconnected from output logic
01	Toggle OCx output line
10	Clear OCx output line to zero
11	Set OCx output line to one

EDGE[4:1] — Input Capture Edge Control

Each pair of bits configures input sensing logic for the corresponding input capture. Refer to **Table D-47**.

Table D-47 EDGE[4:1] Effects

EDGE[4:1]	Configuration
00	Capture disabled
01	Capture on rising edge only
10	Capture on falling edge only
11	Capture on any (rising or falling) edge

D.8.12 Timer Interrupt Mask Registers 1 and 2

TMSK1/TMSK2 — Timer Interrupt Mask Registers 1–2 **$YFF920**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I4/O5I	OCI[4:1]				ICI[3:1]			TOI	0	PAOVI	PAII	CPROUT	CPR[2:0]		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Freescale Semiconductor, Inc.

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

TMSK1 enables OC and IC interrupts. TMSK2 controls pulse accumulator interrupts and TCNT functions.

I4/O5I — Input Capture 4/Output Compare 5 Interrupt Enable
0 = IC4/OC5 interrupt disabled.
1 = IC4/OC5 interrupt requested when I4/O5F flag in TFLG1 is set.

OCI[4:1] — Output Compare Interrupt Enable
OCI[4:1] correspond to OC[4:1].
0 = OC interrupt disabled.
1 = OC interrupt requested when OC flag set.

ICI[3:1] — Input Capture Interrupt Enable
ICI[3:1] correspond to IC[3:1].
0 = IC interrupt disabled.
1 = IC interrupt requested when IC flag set.

TOI — Timer Overflow Interrupt Enable
0 = Timer overflow interrupt disabled.
1 = Interrupt requested when TOF flag is set.

PAOVI — Pulse Accumulator Overflow Interrupt Enable
0 = Pulse accumulator overflow interrupt disabled.
1 = Interrupt requested when PAOVF flag is set.

PAII — Pulse Accumulator Input Interrupt Enable
0 = Pulse accumulator interrupt disabled.
1 = Interrupt requested when PAIF flag is set.

CPROUT — Capture/Compare Unit Clock Output Enable
0 = Normal operation for OC1 pin.
1 = TCNT clock driven out OC1 pin.

CPR[2:0] — Timer Prescaler/PCLK Select Field
This field selects one of seven prescaler taps or PCLK to be TCNT input. Refer to **Table D-48**.

Table D-48 CPR[2:0]/Prescaler Select Field

CPR[2:0]	System Clock Divide-By Factor
000	4
001	8
010	16
011	32
100	64
101	128
110	256
111	PCLK

Freescale Semiconductor, Inc.

D.8.13 Timer Interrupt Flag Registers 1 and 2

TFLG1/TFLG2 — Timer Interrupt Flag Registers 1–2 **$YFF922**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I4/O5F	OCF[4:1]				ICF[3:1]			TOF	0	PAOVF	PAIF	0	0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These registers show condition flags that correspond to GPT events. If the corresponding interrupt enable bit in TMSK1/TMSK2 is set, an interrupt occurs.

I4/O5F — Input Capture 4/Output Compare 5 Flag

When I4/O5 in PACTL is zero, this flag is set each time TCNT matches the TOC5 value in TI4/O5. When I4/O5 in PACTL is one, the flag is set each time a selected edge is detected at the I4/O5 pin.

OCF[4:1] — Output Compare Flags

An output compare flag is set each time TCNT matches the corresponding TOC register. OCF[4:1] correspond to OC[4:1].

ICF[3:1] — Input Capture Flags

A flag is set each time a selected edge is detected at the corresponding input capture pin. ICF[3:1] correspond to IC[3:1].

TOF — Timer Overflow Flag

This flag is set each time TCNT advances from a value of $FFFF to $0000.

PAOVF — Pulse Accumulator Overflow Flag

This flag is set each time the pulse accumulator counter advances from a value of $FF to $00.

PAIF — Pulse Accumulator Flag

In event counting mode, this flag is set when an active edge is detected on the PAI pin. In gated time accumulation mode, it is set at the end of the timed period.

D.8.14 Compare Force Register/PWM Control Register C

CFORC — Compare Force Register/PWM Control Register **$YFF924**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FOC					0	FPWMA	FPWMB	PPROUT	PPR			SFA	SFB	F1A	F1B
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Setting a bit in CFORC causes a specific output on OC or PWM pins. PWMC sets PWM operating conditions.

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

FOC[5:1] — Force Output Compare

FOC[5:1] correspond to OC[5:1].

0 = Has no effect.

1 = Causes pin action programmed for corresponding OC pin, but the OC flag is not set. FOC[5:1] correspond to OC[5:1].

FPWMA/B — Force PWM Value

0 = PWM pin A/B is used for PWM functions; normal operation.

1 = PWM pin A/B is used for discrete output. The value of the F1A/B bit will be driven out on the PWMA/B pin. This is true for PWMA regardless of the state of the PPROUT bit.

PPROUT — PWM Clock Output Enable

0 = Normal PWM operation on PWMA.

1 = Clock selected by PPR[2:0] is driven out PWMA pin.

PPR[2:0] — PWM Prescaler/PCLK Select

This field selects one of seven prescaler taps or PCLK to be PWMCNT input. Refer to **Table D-49**.

Table D-49 PPR[2:0] Field

PPR[2:0]	System Clock Divide-By Factor
000	2
001	4
010	8
011	16
100	32
101	64
110	128
111	PCLK

SFA — PWMA Slow/Fast Select

0 = PWMA period is 256 PWMCNT increments long.

1 = PWMA period is 32768 PWMCNT increments long.

SFB — PWMB Slow/Fast Select

0 = PWMB period is 256 PWMCNT increments long.

1 = PWMB period is 32768 PWMCNT increments long.

Table D-50 shows a range of PWM output frequencies using 16.78 MHz, 20.97 MHz, and 25.17 MHz system clocks.

Freescale Semiconductor, Inc.

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

Table D-50 PWM Frequency Ranges

PPR [2:0]	Prescaler Tap			SFA/B = 0			SFA/B = 1		
	16.78 MHz	20.97 MHz	25.17 MHz	16.78 MHz	20.97 MHz	25.17 MHz	16.78 MHz	20.97 MHz	25.17 MHz
000	Div 2 = 8.39 MHz	Div 2 = 10.5 MHz	Div 2 = 12.6 MHz	32.8 kHz	41 kHz	49.2 kHz	256 Hz	320 Hz	384 Hz
001	Div 4 = 4.19 MHz	Div 4 = 5.25 MHz	Div 4 = 6.29 MHz	16.4 kHz	20.5 kHz	24.6 kHz	128 Hz	160 Hz	192 Hz
010	Div 8 = 2.10 MHz	Div 8 = 2.62 MHz	Div 8 = 3.15 MHz	8.19 kHz	10.2 kHz	12.3 kHz	64.0 Hz	80.0 Hz	96 Hz
011	Div 16 = 1.05 MHz	Div 16 = 1.31 MHz	Div 16 = 1.57 MHz	4.09 kHz	5.15 kHz	6.13 kHz	32.0 Hz	40.0 Hz	48 Hz
100	Div 32 = 524 kHz	Div 32 = 655 kHz	Div 32 = 787 kHz	2.05 kHz	2.56 kHz	3.07 kHz	16.0 Hz	20.0 Hz	24 Hz
101	Div 64 = 262 kHz	Div 64 = 328 kHz	Div 64 = 393 kHz	1.02 kHz	1.28 kHz	1.54 kHz	8.0 Hz	10.0 Hz	12 Hz
110	Div 128 = 131 kHz	Div 128 = 164 kHz	Div 128 = 197 kHz	512 Hz	641 Hz	770 Hz	4.0 Hz	5.0 Hz	6 Hz
111	PCLK	PCLK	PCLK	PCLK/256	PCLK/256	PCLK/256	PCLK/32768	PCLK/32768	PCLK/32768

F1A/B — Force Logic Level One on PWMA/B
0 = Force logic level zero output on PWMA/B pin.
1 = Force logic level one output on PWMA/B pin.

D.8.15 PWM Registers A/B

PWMA — PWM Register A **$YFF926**
PWMB — PWM Register B **$YFF927**

The value in these registers determines pulse width of the corresponding PWM output. A value of $00 corresponds to continuously low output; a value of $80 to 50% duty cycle. Maximum value ($FF) selects an output that is high for 255/256 of the period. Writes to these registers are buffered by PWMBUFA and PWMBUFB.

D.8.16 PWM Count Register

PWMCNT — PWM Count Register **$YFF928**

PWMCNT is the 16-bit free-running counter used for GPT PWM functions.

D.8.17 PWM Buffer Registers A/B

PWMBUFA — PWM Buffer Register A **$YFF92A**
PWMBUFB — PWM Buffer Register B **$YFF92B**

To prevent glitches when PWM duty cycle is changed, the contents of PWMA and PWMB are transferred to these read-only registers at the end of each duty cycle. Reset state is $0000.

Freescale Semiconductor, Inc.

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

D.8.18 GPT Prescaler

PRESCL — GPT Prescaler **$YFF92C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED															
POWER ON RESET ONLY:															
							0	0	0	0	0	0	0	0	0

The 9-bit prescaler value can be read from bits [8:0] at this address. Bits [15:9] always read as zeros. Reset state is $0000.

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

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

APPENDIX E
INITIALIZATION AND PROGRAMMING EXAMPLES

This section contains basic initialization programs and several programming exercises using different M68HC16 Z-series modules. The purpose of these exercises is to provide the designer and programmer with a means to shorten design time. All of the programs were written to run on the M68HC16Z1EVB evaluation board. Refer to the *M68HC16Z1EVB Evaluation Board User's Manual* (M68HC16Z1EVB/D) for further information.

NOTE

These programs will also work on the Modular Evaluation Board (MEVB) using a microcontroller personality board for the appropriate Z-series derivative. See **APPENDIX C DEVELOPMENT SUPPORT** for more information on the MEVB.

E.1 Initialization Programs

The following initialization routines accompany the programming examples used in this manual. For information on assembler commands and directives, refer to the *M68HC16Z1EVB Evaluation Board User's Manual* (M68HC16Z1EVB/D).

- *EQUATES.ASM* — This program lists of all the MC68HC16 registers equated to their memory address. This allows programmers to use the register name in their programs, and then the assembler selects the correct memory address for that register.
- *ORG00000.ASM* — This program consists of five lines of code that set the reset vector information into the proper locations.
- *ORG00008.ASM* — This program initializes the interrupt/exception vectors ($0008 – $01FE). If an interrupt occurs requiring the use of any of these vectors, program flow continues at the label "BDM". The programmer must add code at the label location to put the program running on the EVB16 into background debug mode or some other appropriate routine.
- *INITSYS.ASM* — This program consists of ten to twelve lines of code that initialize the system clock, set the extension registers, and chip-selects. It also turns off the COP (software watchdog) that is set to "on" after reset.
- *INITRAM.ASM* — This program initializes the 1-Kbyte internal SRAM at $10000, and sets the stack inside it.
- *INITSCI.ASM* — This program initializes the SCI to transmit and receive at 9600 baud.

NOTE

These programs and others can be obtained from the Freescale web site at http://www.freescale.com

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

Freescale Semiconductor, Inc.

E.1.1 EQUATES.ASM

```
*DESCRIPTION :THIS IS A TABLE OF EQUATES FOR ALL M68HC16 Z-SERIES
*             REGISTERS.
*************************************************************************
***** SIM MODULE REGISTERS *****
SIMMCR  EQU $FA00   ;SIM MODULE CONFIGURATION REGISTER
SIMTR   EQU $FA02   ;SYSTEM INTEGRATION TEST REGISTER
SYNCR   EQU $FA04   ;CLOCK SYNTHESIZER CONTROL REGISTER
RSR     EQU $FA07   ;RESET STATUS REGISTER
SIMTRE  EQU $FA08   ;SYSTEM INTEGRATION TEST REGISTER (E CLOCK)
PORTE0  EQU $FA11   ;PORTE DATA REGISTER (SAME DATA AS PORTE1)
PORTE1  EQU $FA13   ;PORTE DATA REGISTER (SAME DATA AS PORTE0)
DDRE    EQU $FA15   ;PORTE DATA DIRECTION REGISTER
PEPAR   EQU $FA17   ;PORTE PIN ASSIGNMENT REGISTER
PORTF0  EQU $FA19   ;PORT F DATA REGISTER (SAME DATA AS PORTF1)
PORTF1  EQU $FA1B   ;PORT F DATA REGISTER (SAME DATA AS PORTF0)
DDRF    EQU $FA1D   ;PORT F DATA DIRECTION REGISTER
PFPAR   EQU $FA1F   ;PORT F PIN ASSIGNMENT REGISTER
SYPCR   EQU $FA21   ;SYSTEM PROTECTION CONTROL REGISTER
PICR    EQU $FA22   ;PERIODIC INTERRUPT CONTROL REGISTER
PITR    EQU $FA24   ;PERIODIC INTERRUPT TIMING REGISTER
SWSR    EQU $FA27   ;SOFTWARE SERVICE REGISTER
TSTMSRA EQU $FA30   ;MASTER SHIFT REGISTER A
TSTMSRB EQU $FA32   ;MASTER SHIFT REGISTER B
TSTSC   EQU $FA34   ;TEST MODULE SHIFT COUNT
TSTRC   EQU $FA36   ;TEST MODULE REPETITION COUNT
CREG    EQU $FA38   ;TEST SUBMODULE CONTROL REGISTER
DREG    EQU $FA3A   ;DISTRIBUTED REGISTER
CSPDR   EQU $FA41   ;PORT C DATA REGISTER
CSPAR0  EQU $FA44   ;CHIP-SELECT PIN ASSIGNMENT REGISTER 0
CSPAR1  EQU $FA46   ;CHIP-SELECT PIN ASSIGNMENT REGISTER 1
CSBARBT EQU $FA48   ;CHIP-SELECT BOOT BASE ADDRESS REGISTER
CSORBT  EQU $FA4A   ;CHIP-SELECT BOOT OPTION REGISTER
CSBAR0  EQU $FA4C   ;CHIP-SELECT 0 BASE ADDRESS REGISTER
CSOR0   EQU $FA4E   ;CHIP SELECT 0 OPTION REGISTER
CSBAR1  EQU $FA50   ;CHIP-SELECT 1 BASE ADDRESS REGISTER
CSOR1   EQU $FA52   ;CHIP-SELECT 1 OPTION REGISTER
CSBAR2  EQU $FA54   ;CHIP-SELECT 2 BASE ADDRESS REGISTER
CSOR2   EQU $FA56   ;CHIP-SELECT 2 OPTION REGISTER
CSBAR3  EQU $FA58   ;CHIP-SELECT 3 BASE ADDRESS REGISTER
CSOR3   EQU $FA5A   ;CHIP-SELECT 3 OPTION REGISTER
CSBAR4  EQU $FA5C   ;CHIP-SELECT 4 BASE ADDRESS REGISTER
CSOR4   EQU $FA5E   ;CHIP-SELECT 4 OPTION REGISTER
CSBAR5  EQU $FA60   ;CHIP-SELECT 5 BASE ADDRESS REGISTER
CSOR5   EQU $FA62   ;CHIP-SELECT 5 OPTION REGISTER
CSBAR6  EQU $FA64   ;CHIP-SELECT 6 BASE ADDRESS REGISTER
CSOR6   EQU $FA66   ;CHIP-SELECT 6 OPTION REGISTER
CSBAR7  EQU $FA68   ;CHIP-SELECT 7 BASE ADDRESS REGISTER
CSOR7   EQU $FA6A   ;CHIP-SELECT 7 OPTION REGISTER
CSBAR8  EQU $FA6C   ;CHIP-SELECT 8 BASE ADDRESS REGISTER
CSOR8   EQU $FA6E   ;CHIP-SELECT 8 OPTION REGISTER
CSBAR9  EQU $FA70   ;CHIP-SELECT 9 BASE ADDRESS REGISTER
CSOR9   EQU $FA72   ;CHIP-SELECT 9 OPTION REGISTER
CSBAR10 EQU $FA74   ;CHIP-SELECT 10 BASE ADDRESS REGISTER
CSOR10  EQU $FA76   ;CHIP-SELECT 10 OPTION REGISTER
```

Freescale Semiconductor, Inc.

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

```
***** SRAM MODULE REGISTERS *****
RAMMCR  EQU $FB00   ;RAM MODULE CONFIGURATION REGISTER
RAMTST  EQU $FB02   ;RAM TEST REGISTER
RAMBAH  EQU $FB04   ;RAM BASE ADDRESS HIGH REGISTER
RAMBAL  EQU $FB06   ;RAM BASE ADDRESS LOW REGISTER
***** MRM MODULE REGISTERS *****
MRMCR   EQU $F820   ;MASKED ROM MODULE CONFIGURATION REGISTER
ROMBAH  EQU $F824   ;ROM ARRAY BASE ADDRESS REGISTER HIGH
ROMBAL  EQU $F826   ;ROM ARRAY BASE ADDRESS REGISTER LOW
SIGHI   EQU $F828   ;SIGNATURE REGISTER HIGH
SIGLO   EQU $F82A   ;SIGNATURE REGISTER LOW
ROMBS0  EQU $F830   ;ROM BOOTSTRAP WORD 0
ROMBS1  EQU $F832   ;ROM BOOTSTRAP WORD 1
ROMBS2  EQU $F834   ;ROM BOOTSTRAP WORD 2
ROMBS3  EQU $F836   ;ROM BOOTSTRAP WORD 3
***** QSM MODULE REGISTERS *****
QMCR    EQU $FC00   ;QSM MODULE CONFIGURATION REGISTER
QTEST   EQU $FC02   ;QSM TEST REGISTER
QILR    EQU $FC04   ;QSM INTERRUPT LEVELS REGISTER
QIVR    EQU $FC05   ;QSM INTERRUPT VECTOR REGISTER
SCCR0   EQU $FC08   ;SCI CONTROL REGISTER 0
SCCR1   EQU $FC0A   ;SCI CONTROL REGISTER 1
SCSR    EQU $FC0C   ;SCI STATUS REGISTER
SCDR    EQU $FC0E   ;SCI DATA REGISTER (FULL WORD, NOT LAST 8 BITS)
QPDR    EQU $FC15   ;QSM PORT DATA REGISTER
QPAR    EQU $FC16   ;QSM PIN ASSIGNMENT REGISTER
QDDR    EQU $FC17   ;QSM DATA DIRECTION REGISTER
SPCR0   EQU $FC18   ;QSPI CONTROL REGISTER 0
SPCR1   EQU $FC1A   ;QSPI CONTROL REGISTER 1
SPCR2   EQU $FC1C   ;QSPI CONTROL REGISTER 2
SPCR3   EQU $FC1E   ;QSPI CONTROL REGISTER 3
SPSR    EQU $FC1F   ;QSPI STATUS REGISTER
RR0     EQU $FD00   ;SPI REC.RAM 0
RR1     EQU $FD02   ;SPI REC.RAM 1
RR2     EQU $FD04   ;SPI REC.RAM 2
RR3     EQU $FD06   ;SPI REC.RAM 3
RR4     EQU $FD08   ;SPI REC.RAM 4
RR5     EQU $FD0A   ;SPI REC.RAM 5
RR6     EQU $FD0C   ;SPI REC.RAM 6
RR7     EQU $FD0E   ;SPI REC.RAM 7
RR8     EQU $FD00   ;SPI REC.RAM 8
RR9     EQU $FD02   ;SPI REC.RAM 9
RRA     EQU $FD04   ;SPI REC.RAM A
RRB     EQU $FD06   ;SPI REC.RAM B
RRC     EQU $FD08   ;SPI REC.RAM C
RRD     EQU $FD0A   ;SPI REC.RAM D
RRE     EQU $FD0C   ;SPI REC.RAM E
RRF     EQU $FD0E   ;SPI REC.RAM F
TR0     EQU $FD20   ;SPI TXD.RAM 0
TR1     EQU $FD22   ;SPI TXD.RAM 1
TR2     EQU $FD24   ;SPI TXD.RAM 2
TR3     EQU $FD26   ;SPI TXD.RAM 3
TR4     EQU $FD28   ;SPI TXD.RAM 4
TR5     EQU $FD2A   ;SPI TXD.RAM 5
TR6     EQU $FD2C   ;SPI TXD.RAM 6
```

Freescale Semiconductor, Inc.

```
TR7     EQU $FD2E   ;SPI TXD.RAM 7
TR8     EQU $FD30   ;SPI TXD.RAM 8
TR9     EQU $FD32   ;SPI TXD.RAM 9
TRA     EQU $FD34   ;SPI TXD.RAM A
TRB     EQU $FD36   ;SPI TXD.RAM B
TRC     EQU $FD38   ;SPI TXD.RAM C
TRD     EQU $FD3A   ;SPI TXD.RAM D
TRE     EQU $FD3C   ;SPI TXD.RAM E
TRF     EQU $FD3E   ;SPI TXD.RAM F
CR0     EQU $FD40   ;SPI CMD.RAM 0
CR1     EQU $FD41   ;SPI CMD.RAM 1
CR2     EQU $FD42   ;SPI CMD.RAM 2
CR3     EQU $FD43   ;SPI CMD.RAM 3
CR4     EQU $FD44   ;SPI CMD.RAM 4
CR5     EQU $FD45   ;SPI CMD.RAM 5
CR6     EQU $FD46   ;SPI CMD.RAM 6
CR7     EQU $FD47   ;SPI CMD.RAM 7
CR8     EQU $FD48   ;SPI CMD.RAM 8
CR9     EQU $FD49   ;SPI CMD.RAM 9
CRA     EQU $FD4A   ;SPI CMD.RAM A
CRB     EQU $FD4B   ;SPI CMD.RAM B
CRC     EQU $FD4C   ;SPI CMD.RAM C
CRD     EQU $FD4D   ;SPI CMD.RAM D
CRE     EQU $FD4E   ;SPI CMD.RAM E
CRF     EQU $FD4F   ;SPI CMD.RAM F
***** MCCI MODULE REGISTERS *****
MMCR    EQU $FC00   ;MCCI MODULE CONFIGURATION REGISTER
MTEST   EQU $FC02   ;MCCI TEST REGISTER
ILSCI   EQU $FC04   ;SCI INTERRUPT LEVEL REGISTER
MIVR    EQU $FC05   ;MCCI INTERRUPT VECTOR REGISTER
ILSPI   EQU $FC06   ;SPI INTERRUPT LEVEL REGISTER
MPAR    EQU $FC09   ;MCCI PIN ASSIGNMENT REGISTER
MDDR    EQU $FC0B   ;MCCI DATA DIRECTION REGISTER
PORTMC  EQU $FC0D   ;MCCI PORT DATA REGISTER
PORTMCP EQU $FC0F   ;MCCI PORT PIN STATE REGISTER
SCCR0A  EQU $FC18   ;SCIA CONTROL REGISTER 0
SCCR1A  EQU $FC1A   ;SCIA CONTROL REGISTER 1
SCSRA   EQU $FC1C   ;SCIA STATUS REGISTER
SCDRA   EQU $FC1E   ;SCIA DATA REGISTER
SCCR0B  EQU $FC28   ;SCIB CONTROL REGISTER 0
SCCR1B  EQU $FC2A   ;SCIB CONTROL REGISTER 1
SCSRB   EQU $FC2C   ;SCIB STATUS REGISTER
SCDRB   EQU $FC2E   ;SCIB DATA REGISTER
SPCR    EQU $FC38   ;SPI CONTROL REGISTER
SPSR    EQU $FC3C   ;SPI STATUS REGISTER
SPDR    EQU $FC3E   ;SPI DATA REGISTER
***** GPT MODULE REGISTERS *****
GPTMCR  EQU $F900   ;GPT MODULE CONFIGURATION REGISTER
GPTMTR  EQU $F902   ;GPT MODULE TEST REGISTER (RESERVED)
ICR     EQU $F904   ;GPT INTERRUPT CONFIGURATION REGISTER
PDDR    EQU $F906   ;PARALLEL DATA DIRECTION REGISTER
GPTPDR  EQU $F907   ;PARALLEL DATA REGISTER
OC1M    EQU $F908   ;OC1 ACTION MASK REGISTER
OC1D    EQU $F909   ;OC1 ACTION DATA REGISTER
TCNT    EQU $F90A   ;TIMER COUNTER REGISTER
```

Freescale Semiconductor, Inc.

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

```
PACTL   EQU $F90C   ;PULSE ACCUMULATOR CONTROL REGISTER
PACNT   EQU $F90D   ;PULSE ACCUMULATOR COUNTER
TIC1    EQU $F90E   ;INPUT CAPTURE REGISTER 1
TIC2    EQU $F910   ;INPUT CAPTURE REGISTER 2
TIC3    EQU $F912   ;INPUT CAPTURE REGISTER 3
TOC1    EQU $F914   ;OUTPUT COMPARE REGISTER 1
TOC2    EQU $F916   ;OUTPUT COMPARE REGISTER 2
TOC3    EQU $F918   ;OUTPUT COMPARE REGISTER 3
TOC4    EQU $F91A   ;OUTPUT COMPARE REGISTER 4
TI4O5   EQU $F91C   ;INPUT CAPTURE 4 OR OUTPUT COMPARE 5
TCTL1   EQU $F91E   ;TIMER CONTROL REGISTER 1
TCTL2   EQU $F91F   ;TIMER CONTROL REGISTER 2
TMSK1   EQU $F920   ;TIMER INTERRUPT MASK REGISTER 1
TMSK2   EQU $F921   ;TIMER INTERRUPT MASK REGISTER 2
TFLG1   EQU $F922   ;TIMER INTERRUPT FLAG REGISTER 1
TFLG2   EQU $F923   ;TIMER INTERRUPT FLAG REGISTER 2
CFORC   EQU $F924   ;COMPARE FORCE REGISTER
PWMC    EQU $F924   ;PWM CONTROL REGISTER
PWMA    EQU $F926   ;PWM REGISTER A
PWMB    EQU $F927   ;PWM REGISTER B
PWMCNT  EQU $F928   ;PWM COUNTER REGISTER
PWMBUFA EQU $F92A   ;PWM BUFFER REGISTER A
PWMBUFB EQU $F92B   ;PWM BUFFER REGISTER B
PRESCL  EQU $F92C   ;GPT PRESCALER
***** ADC MODULE REGISTERS *****
ADCMCR  EQU $F700   ;ADC MODULE CONFIGURATION REGISTER
ADTEST  EQU $F702   ;ADC TEST REGISTER
ADCPDR  EQU $F706   ;ADC PORT DATA REGISTER
ADCTL0  EQU $F70A   ;A/D CONTROL REGISTER 0
ADCTL1  EQU $F70C   ;A/D CONTROL REGISTER 1
ADSTAT  EQU $F70E   ;ADC STATUS REGISTER
RJURR0  EQU $F710   ;RIGHT JUSTIFIED UNSIGNED RESULT REGISTER 0
RJURR1  EQU $F712   ;RIGHT JUSTIFIED UNSIGNED RESULT REGISTER 1
RJURR2  EQU $F714   ;RIGHT JUSTIFIED UNSIGNED RESULT REGISTER 2
RJURR3  EQU $F716   ;RIGHT JUSTIFIED UNSIGNED RESULT REGISTER 3
RJURR4  EQU $F718   ;RIGHT JUSTIFIED UNSIGNED RESULT REGISTER 4
RJURR5  EQU $F71A   ;RIGHT JUSTIFIED UNSIGNED RESULT REGISTER 5
RJURR6  EQU $F71C   ;RIGHT JUSTIFIED UNSIGNED RESULT REGISTER 6
RJURR7  EQU $F71E   ;RIGHT JUSTIFIED UNSIGNED RESULT REGISTER 7
LJSRR0  EQU $F720   ;LEFT JUSTIFIED SIGNED RESULT REGISTER 0
LJSRR1  EQU $F722   ;LEFT JUSTIFIED SIGNED RESULT REGISTER 1
LJSRR2  EQU $F724   ;LEFT JUSTIFIED SIGNED RESULT REGISTER 2
LJSRR3  EQU $F726   ;LEFT JUSTIFIED SIGNED RESULT REGISTER 3
LJSRR4  EQU $F728   ;LEFT JUSTIFIED SIGNED RESULT REGISTER 4
LJSRR5  EQU $F72A   ;LEFT JUSTIFIED SIGNED RESULT REGISTER 5
LJSRR6  EQU $F72C   ;LEFT JUSTIFIED SIGNED RESULT REGISTER 6
LJSRR7  EQU $F72E   ;LEFT JUSTIFIED SIGNED RESULT REGISTER 7
LJURR0  EQU $F730   ;LEFT JUSTIFIED UNSIGNED RESULT REGISTER 0
LJURR1  EQU $F732   ;LEFT JUSTIFIED UNSIGNED RESULT REGISTER 1
LJURR2  EQU $F734   ;LEFT JUSTIFIED UNSIGNED RESULT REGISTER 2
LJURR3  EQU $F736   ;LEFT JUSTIFIED UNSIGNED RESULT REGISTER 3
LJURR4  EQU $F738   ;LEFT JUSTIFIED UNSIGNED RESULT REGISTER 4
LJURR5  EQU $F73A   ;LEFT JUSTIFIED UNSIGNED RESULT REGISTER 5
LJURR6  EQU $F73C   ;LEFT JUSTIFIED UNSIGNED RESULT REGISTER 6
LJURR7  EQU $F73E   ;LEFT JUSTIFIED UNSIGNED RESULT REGISTER 7
```

Freescale Semiconductor, Inc.

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

E.1.2 ORG00000.ASM

```
*       Title : ORG00000
*       Description : This file is included to set up the reset
*                     vector ($00000 - $00006).
*
************************************************************************

                ORG      $0000          ;put the following reset vector
                                        ;information
                                        ;at address $00000 of the memory map
               DC.W      $0010          ;zk=0, sk=1, pk=0
               DC.W      $0200          ;pc=200 -- initial program counter
               DC.W      $03FE          ;sp=03fe -- initial stack pointer
               DC.W      $0000          ;iz=0 -- direct page pointer
```

E.1.3 ORG00008.ASM

```
*   Title : ORG00008
*   Description : This file initializes the interrupt/
*                 exception vectors ($0008 - $01fe).
*                 If an interrupt occurs requiring the use of
*                 any of these vectors, program flow will
*                 continue at the label “bdm” which must be
*                 added by the programmer to his/her code to
*                 put the program into background debug mode
*                 or some other appropriate routine.
*
************************************************************************

        ORG $0008               ;put the following code in memory
                                ;starting at address $0008 of the map
                                ;(after the reset vector).
                                ;there is a total of 252 of these
                                ;”DC.W BDM” lines

                                ;Vector Number (in base 10)
                                ;and Vector Description

        DC.W  BDM     ;4    Breakpoint (BKPT)
        DC.W  BDM     ;5    Bus Error (BERR)
        DC.W  BDM     ;6    Software Interrupt (SWI)
        DC.W  BDM     ;7    Illegal Instruction
        DC.W  BDM     ;8    Divide by Zero
        DC.W  BDM     ;9    (Unassigned Reserved)
        DC.W  BDM     ;10   (Unassigned Reserved)
        DC.W  BDM     ;11   (Unassigned Reserved)
        DC.W  BDM     ;12   (Unassigned Reserved)
        DC.W  BDM     ;13   (Unassigned Reserved)
        DC.W  BDM     ;14   (Unassigned Reserved)
        DC.W  BDM     ;15   Uninitialized Interrupt
        DC.W  BDM     ;16   (Unassigned Reserved)
        DC.W  BDM     ;17   Level 1 Interrupt Autovector
        DC.W  BDM     ;18   Level 2 Interrupt Autovector
        DC.W  BDM     ;19   Level 3 Interrupt Autovector
        DC.W  BDM     ;20   Level 4 Interrupt Autovector
```

Freescale Semiconductor, Inc.

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

```
DC.W  BDM     ;21   Level 5 Interrupt Autovector
DC.W  BDM     ;22   Level 6 Interrupt Autovector
DC.W  BDM     ;23   Level 7 Interrupt Autovector
DC.W  BDM     ;24   Spurious Interrupt
DC.W  BDM     ;25   (Unassigned Reserved)
DC.W  BDM     ;26   (Unassigned Reserved)
DC.W  BDM     ;27   (Unassigned Reserved)
DC.W  BDM     ;28   (Unassigned Reserved)
DC.W  BDM     ;29   (Unassigned Reserved)
DC.W  BDM     ;30   (Unassigned Reserved)
DC.W  BDM     ;31   (Unassigned Reserved)
DC.W  BDM     ;32   (Unassigned Reserved)
DC.W  BDM     ;33   (Unassigned Reserved)
DC.W  BDM     ;34   (Unassigned Reserved)
DC.W  BDM     ;35   (Unassigned Reserved)
DC.W  BDM     ;36   (Unassigned Reserved)
DC.W  BDM     ;37   (Unassigned Reserved)
DC.W  BDM     ;38   (Unassigned Reserved)
DC.W  BDM     ;39   (Unassigned Reserved)
DC.W  BDM     ;40   (Unassigned Reserved)
DC.W  BDM     ;41   (Unassigned Reserved)
DC.W  BDM     ;42   (Unassigned Reserved)
DC.W  BDM     ;43   (Unassigned Reserved)
DC.W  BDM     ;44   (Unassigned Reserved)
DC.W  BDM     ;45   (Unassigned Reserved)
DC.W  BDM     ;46   (Unassigned Reserved)
DC.W  BDM     ;47   (Unassigned Reserved)
DC.W  BDM     ;48   (Unassigned Reserved)
DC.W  BDM     ;49   (Unassigned Reserved)
DC.W  BDM     ;50   (Unassigned Reserved)
DC.W  BDM     ;51   (Unassigned Reserved)
DC.W  BDM     ;52   (Unassigned Reserved)
DC.W  BDM     ;53   (Unassigned Reserved)
DC.W  BDM     ;54   (Unassigned Reserved)
DC.W  BDM     ;55   (Unassigned Reserved)
DC.W  BDM     ;56   User Defined Interrupt Vector 1
DC.W  BDM     ;57   User Defined Interrupt Vector 2
DC.W  BDM     ;58   User Defined Interrupt Vector 3
DC.W  BDM     ;59   User Defined Interrupt Vector 4
DC.W  BDM     ;60   User Defined Interrupt Vector 5
DC.W  BDM     ;61   User Defined Interrupt Vector 6
DC.W  BDM     ;62   User Defined Interrupt Vector 7
DC.W  BDM     ;63   User Defined Interrupt Vector 8
DC.W  BDM     ;64   User Defined Interrupt Vector 9
DC.W  BDM     ;65   User Defined Interrupt Vector 10
DC.W  BDM     ;66   User Defined Interrupt Vector 11
DC.W  BDM     ;67   User Defined Interrupt Vector 12
DC.W  BDM     ;68   User Defined Interrupt Vector 13
DC.W  BDM     ;69   User Defined Interrupt Vector 14
DC.W  BDM     ;70   User Defined Interrupt Vector 15
DC.W  BDM     ;71   User Defined Interrupt Vector 16
DC.W  BDM     ;72   User Defined Interrupt Vector 17
DC.W  BDM     ;73   User Defined Interrupt Vector 18
DC.W  BDM     ;74   User Defined Interrupt Vector 19
DC.W  BDM     ;75   User Defined Interrupt Vector 20
```

Freescale Semiconductor, Inc.

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

```
DC.W  BDM     ;76   User Defined Interrupt Vector 21
DC.W  BDM     ;77   User Defined Interrupt Vector 22
DC.W  BDM     ;78   User Defined Interrupt Vector 23
DC.W  BDM     ;79   User Defined Interrupt Vector 24
DC.W  BDM     ;80   User Defined Interrupt Vector 25
DC.W  BDM     ;81   User Defined Interrupt Vector 26
DC.W  BDM     ;82   User Defined Interrupt Vector 27
DC.W  BDM     ;83   User Defined Interrupt Vector 28
DC.W  BDM     ;84   User Defined Interrupt Vector 29
DC.W  BDM     ;85   User Defined Interrupt Vector 30
DC.W  BDM     ;86   User Defined Interrupt Vector 31
DC.W  BDM     ;87   User Defined Interrupt Vector 32
DC.W  BDM     ;88   User Defined Interrupt Vector 33
DC.W  BDM     ;89   User Defined Interrupt Vector 34
DC.W  BDM     ;90   User Defined Interrupt Vector 35
DC.W  BDM     ;91   User Defined Interrupt Vector 36
DC.W  BDM     ;92   User Defined Interrupt Vector 37
DC.W  BDM     ;93   User Defined Interrupt Vector 38
DC.W  BDM     ;94   User Defined Interrupt Vector 39
DC.W  BDM     ;95   User Defined Interrupt Vector 40
DC.W  BDM     ;96   User Defined Interrupt Vector 41
DC.W  BDM     ;97   User Defined Interrupt Vector 42
DC.W  BDM     ;98   User Defined Interrupt Vector 43
DC.W  BDM     ;99   User Defined Interrupt Vector 44
DC.W  BDM     ;100  User Defined Interrupt Vector 45
DC.W  BDM     ;101  User Defined Interrupt Vector 46
DC.W  BDM     ;102  User Defined Interrupt Vector 47
DC.W  BDM     ;103  User Defined Interrupt Vector 48
DC.W  BDM     ;104  User Defined Interrupt Vector 49
DC.W  BDM     ;105  User Defined Interrupt Vector 50
DC.W  BDM     ;106  User Defined Interrupt Vector 51
DC.W  BDM     ;107  User Defined Interrupt Vector 52
DC.W  BDM     ;108  User Defined Interrupt Vector 53
DC.W  BDM     ;109  User Defined Interrupt Vector 54
DC.W  BDM     ;100  User Defined Interrupt Vector 55
DC.W  BDM     ;111  User Defined Interrupt Vector 56
DC.W  BDM     ;112  User Defined Interrupt Vector 57
DC.W  BDM     ;113  User Defined Interrupt Vector 58
DC.W  BDM     ;114  User Defined Interrupt Vector 59
DC.W  BDM     ;115  User Defined Interrupt Vector 60
DC.W  BDM     ;116  User Defined Interrupt Vector 61
DC.W  BDM     ;117  User Defined Interrupt Vector 62
DC.W  BDM     ;118  User Defined Interrupt Vector 63
DC.W  BDM     ;119  User Defined Interrupt Vector 64
DC.W  BDM     ;120  User Defined Interrupt Vector 65
DC.W  BDM     ;121  User Defined Interrupt Vector 66
DC.W  BDM     ;122  User Defined Interrupt Vector 67
DC.W  BDM     ;123  User Defined Interrupt Vector 68
DC.W  BDM     ;124  User Defined Interrupt Vector 69
DC.W  BDM     ;125  User Defined Interrupt Vector 70
DC.W  BDM     ;126  User Defined Interrupt Vector 71
DC.W  BDM     ;127  User Defined Interrupt Vector 72
DC.W  BDM     ;128  User Defined Interrupt Vector 73
DC.W  BDM     ;129  User Defined Interrupt Vector 74
DC.W  BDM     ;130  User Defined Interrupt Vector 75
```

Freescale Semiconductor, Inc.

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

```
DC.W  BDM     ;131  User Defined Interrupt Vector 76
DC.W  BDM     ;132  User Defined Interrupt Vector 77
DC.W  BDM     ;133  User Defined Interrupt Vector 78
DC.W  BDM     ;134  User Defined Interrupt Vector 79
DC.W  BDM     ;135  User Defined Interrupt Vector 80
DC.W  BDM     ;136  User Defined Interrupt Vector 81
DC.W  BDM     ;137  User Defined Interrupt Vector 82
DC.W  BDM     ;138  User Defined Interrupt Vector 83
DC.W  BDM     ;139  User Defined Interrupt Vector 84
DC.W  BDM     ;140  User Defined Interrupt Vector 85
DC.W  BDM     ;141  User Defined Interrupt Vector 86
DC.W  BDM     ;142  User Defined Interrupt Vector 87
DC.W  BDM     ;143  User Defined Interrupt Vector 88
DC.W  BDM     ;144  User Defined Interrupt Vector 89
DC.W  BDM     ;145  User Defined Interrupt Vector 90
DC.W  BDM     ;146  User Defined Interrupt Vector 91
DC.W  BDM     ;147  User Defined Interrupt Vector 92
DC.W  BDM     ;148  User Defined Interrupt Vector 93
DC.W  BDM     ;149  User Defined Interrupt Vector 94
DC.W  BDM     ;150  User Defined Interrupt Vector 95
DC.W  BDM     ;151  User Defined Interrupt Vector 96
DC.W  BDM     ;152  User Defined Interrupt Vector 97
DC.W  BDM     ;153  User Defined Interrupt Vector 98
DC.W  BDM     ;154  User Defined Interrupt Vector 99
DC.W  BDM     ;155  User Defined Interrupt Vector 100
DC.W  BDM     ;156  User Defined Interrupt Vector 101
DC.W  BDM     ;157  User Defined Interrupt Vector 102
DC.W  BDM     ;158  User Defined Interrupt Vector 103
DC.W  BDM     ;159  User Defined Interrupt Vector 104
DC.W  BDM     ;160  User Defined Interrupt Vector 105
DC.W  BDM     ;161  User Defined Interrupt Vector 106
DC.W  BDM     ;162  User Defined Interrupt Vector 107
DC.W  BDM     ;163  User Defined Interrupt Vector 108
DC.W  BDM     ;164  User Defined Interrupt Vector 109
DC.W  BDM     ;165  User Defined Interrupt Vector 110
DC.W  BDM     ;166  User Defined Interrupt Vector 111
DC.W  BDM     ;167  User Defined Interrupt Vector 112
DC.W  BDM     ;168  User Defined Interrupt Vector 113
DC.W  BDM     ;169  User Defined Interrupt Vector 114
DC.W  BDM     ;170  User Defined Interrupt Vector 115
DC.W  BDM     ;171  User Defined Interrupt Vector 116
DC.W  BDM     ;172  User Defined Interrupt Vector 117
DC.W  BDM     ;173  User Defined Interrupt Vector 118
DC.W  BDM     ;174  User Defined Interrupt Vector 119
DC.W  BDM     ;175  User Defined Interrupt Vector 120
DC.W  BDM     ;176  User Defined Interrupt Vector 121
DC.W  BDM     ;177  User Defined Interrupt Vector 122
DC.W  BDM     ;178  User Defined Interrupt Vector 123
DC.W  BDM     ;179  User Defined Interrupt Vector 124
DC.W  BDM     ;180  User Defined Interrupt Vector 125
DC.W  BDM     ;181  User Defined Interrupt Vector 126
DC.W  BDM     ;182  User Defined Interrupt Vector 127
DC.W  BDM     ;183  User Defined Interrupt Vector 128
DC.W  BDM     ;184  User Defined Interrupt Vector 129
DC.W  BDM     ;185  User Defined Interrupt Vector 130
```

Freescale Semiconductor, Inc.

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

```
DC.W  BDM     ;186  User Defined Interrupt Vector 131
DC.W  BDM     ;187  User Defined Interrupt Vector 132
DC.W  BDM     ;188  User Defined Interrupt Vector 133
DC.W  BDM     ;189  User Defined Interrupt Vector 134
DC.W  BDM     ;190  User Defined Interrupt Vector 135
DC.W  BDM     ;191  User Defined Interrupt Vector 136
DC.W  BDM     ;192  User Defined Interrupt Vector 137
DC.W  BDM     ;193  User Defined Interrupt Vector 138
DC.W  BDM     ;194  User Defined Interrupt Vector 139
DC.W  BDM     ;195  User Defined Interrupt Vector 140
DC.W  BDM     ;196  User Defined Interrupt Vector 141
DC.W  BDM     ;197  User Defined Interrupt Vector 142
DC.W  BDM     ;198  User Defined Interrupt Vector 143
DC.W  BDM     ;199  User Defined Interrupt Vector 144
DC.W  BDM     ;200  User Defined Interrupt Vector 145
DC.W  BDM     ;201  User Defined Interrupt Vector 146
DC.W  BDM     ;202  User Defined Interrupt Vector 147
DC.W  BDM     ;203  User Defined Interrupt Vector 148
DC.W  BDM     ;204  User Defined Interrupt Vector 149
DC.W  BDM     ;205  User Defined Interrupt Vector 150
DC.W  BDM     ;206  User Defined Interrupt Vector 151
DC.W  BDM     ;207  User Defined Interrupt Vector 152
DC.W  BDM     ;208  User Defined Interrupt Vector 153
DC.W  BDM     ;209  User Defined Interrupt Vector 154
DC.W  BDM     ;210  User Defined Interrupt Vector 155
DC.W  BDM     ;211  User Defined Interrupt Vector 156
DC.W  BDM     ;212  User Defined Interrupt Vector 157
DC.W  BDM     ;213  User Defined Interrupt Vector 158
DC.W  BDM     ;214  User Defined Interrupt Vector 159
DC.W  BDM     ;215  User Defined Interrupt Vector 160
DC.W  BDM     ;216  User Defined Interrupt Vector 161
DC.W  BDM     ;217  User Defined Interrupt Vector 162
DC.W  BDM     ;218  User Defined Interrupt Vector 163
DC.W  BDM     ;219  User Defined Interrupt Vector 164
DC.W  BDM     ;220  User Defined Interrupt Vector 165
DC.W  BDM     ;221  User Defined Interrupt Vector 166
DC.W  BDM     ;222  User Defined Interrupt Vector 167
DC.W  BDM     ;223  User Defined Interrupt Vector 168
DC.W  BDM     ;224  User Defined Interrupt Vector 169
DC.W  BDM     ;225  User Defined Interrupt Vector 170
DC.W  BDM     ;226  User Defined Interrupt Vector 171
DC.W  BDM     ;227  User Defined Interrupt Vector 172
DC.W  BDM     ;228  User Defined Interrupt Vector 173
DC.W  BDM     ;229  User Defined Interrupt Vector 174
DC.W  BDM     ;230  User Defined Interrupt Vector 175
DC.W  BDM     ;231  User Defined Interrupt Vector 176
DC.W  BDM     ;232  User Defined Interrupt Vector 177
DC.W  BDM     ;233  User Defined Interrupt Vector 178
DC.W  BDM     ;234  User Defined Interrupt Vector 179
DC.W  BDM     ;235  User Defined Interrupt Vector 180
DC.W  BDM     ;236  User Defined Interrupt Vector 181
DC.W  BDM     ;237  User Defined Interrupt Vector 182
DC.W  BDM     ;238  User Defined Interrupt Vector 183
DC.W  BDM     ;239  User Defined Interrupt Vector 184
DC.W  BDM     ;240  User Defined Interrupt Vector 185
```

Freescale Semiconductor, Inc.

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

```
        DC.W  BDM     ;241  User Defined Interrupt Vector 186
        DC.W  BDM     ;242  User Defined Interrupt Vector 187
        DC.W  BDM     ;243  User Defined Interrupt Vector 188
        DC.W  BDM     ;244  User Defined Interrupt Vector 189
        DC.W  BDM     ;245  User Defined Interrupt Vector 190
        DC.W  BDM     ;246  User Defined Interrupt Vector 191
        DC.W  BDM     ;247  User Defined Interrupt Vector 192
        DC.W  BDM     ;248  User Defined Interrupt Vector 193
        DC.W  BDM     ;249  User Defined Interrupt Vector 194
        DC.W  BDM     ;250  User Defined Interrupt Vector 195
        DC.W  BDM     ;251  User Defined Interrupt Vector 196
        DC.W  BDM     ;252  User Defined Interrupt Vector 197
        DC.W  BDM     ;253  User Defined Interrupt Vector 198
        DC.W  BDM     ;254  User Defined Interrupt Vector 199
        DC.W  BDM     ;255  User Defined Interrupt Vector 200
```

E.1.4 INITSYS.ASM

```
*       Title : INITSYS
*       Description : Initialize & configure system including
*                     the Software Watchdog and System Clock.
*************************************************************************

INITSYS:                        ;give initial values for extension registers
                                ;and initialize system clock and COP
        LDAB    #$0F
        TBEK                    ; point EK to bank F for register access
        LDAB    #$00
        TBXK                    ; point XK to bank 0
        TBYK                    ; point YK to bank 0
        TBZK                    ; point ZK to bank 0

        LDD     #$0003          ; at reset, the CSBOOT block size is 512k.
        STD     CSBARBT         ; this line sets the block size to 64k
        LDD     #$3830; async, both byte, R/W, AS, Zero WS, S/U SP, IPL all,
                                ;AVEC off
        STD     CSORBT          ;
        LDAA    #$7F            ; w=0, x=1, y=111111
        STAA    SYNCR           ; set system clock to 16.78 Mhz

        CLR     SYPCR           ; turn COP (software watchdog) off,
                                ; since COP is on after reset
```

E.1.5 INITRAM.ASM

```
*       Title : INITRAM
*       Description : Initialize the HC16's 1K internal SRAM
*                     (put SRAM in memory map at $10000, bank 1)
*                     and set the stack inside it.
*************************************************************************

INITRAM:                        ;initialize internal SRAM and stack
        LDD     #$0001
        STD     RAMBAH          ; store high ram array, bank 1
        LDD     #$0000
```


Freescale Semiconductor, Inc.

```
        STD     RAMBAL          ; store low ram array
        CLR     RAMMCR          ; enable ram

        LDAB    #$01
        TBSK                    ; set SK to bank 1 for system stack
        LDS     #$03FE          ; put SP at top of 1k internal SRAM
```

E.1.6 INITSCI.ASM

```
*       Title : INITSCI
*       Description :  Initialize the SCI to transmit and receive
*                      at 9600 baud.
*
*************************************************************************


INITSCI:                        ;initialize the SCI
        LDD     #$0037
        STD     SCCR0           ;set the SCI baud rate to 9600 baud

        LDD     #$000C
        STD     SCCR1           ;enable the SCI receiver and transmitter
```

E.2 Programming Examples

The following programming examples use different M68HC16 Z-series modules. All programs were written to run on the M68HC16Z1EVB evaluation board. Refer to the *M68HC16Z1EVB Evaluation Board User's Manual* (M68HC16Z1EVB/D) for further information.

NOTE

These programs will also work on the modular evaluation board (MEVB) using a microcontroller personality board for the appropriate Z-series derivative. See **APPENDIX C DEVELOPMENT SUPPORT** for more information on the MEVB.

Several of these programs send status messages using the SCI in the QSM on the MC68HC16Z1, MC68CK16Z1, MC68CM16Z1, MC68HC16Z2, and MC68HC16Z3. These programs can be made to function with SCIA in the MCCI on the MC68HC16Z4 and MC68CKZ4 as follows:

- Replace SCCR0 with SCCR0A.
- Replace SCCR1 with SCCR1A.
- Replace SCSR with SCSRA.
- Replace SCDR with SCDRA.

Freescale Semiconductor, Inc.

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

E.2.1 SIM Programming Examples

The following programming examples involve using the system integration module (SIM). The programs include:

- Using ports E and F.
- Setting up U1 and U3 RAM slots with two 32K X 8 RAM chips using chip selects.
- Demonstrating the ability of the M68HC16 to change clock frequencies on the fly.
- Demonstrating the software watchdog, the periodic interrupt, and an autovector.

Refer to **SECTION 5 SYSTEM INTEGRATION MODULE** for more information on the SIM.

E.2.1.1 Example 1 - Using Ports E and F

```
*    Description : This program demonstrates a simple I/O usage of
*                  Ports E and F with a loop that will load Port E
*                  with a number, pass that number over to
*                  Port F through a hardwire of the M68HC16Z1EVB, and then
                   read
*                  that number from the Port F data register.
*                  Port E will be incremented each loop.
*                  The hardwire of the M68HC16Z1EVB is from DSACK0 to MODCLK,
*                  from DSACK1 to IRQ1, and from AVEC to IRQ2.
*                  The numbers start at #$00 and go to #$07.
*
*************************************************************************

        INCLUDE  'EQUATES.ASM'     ;table of EQUates for common register
                                   ;addresses
        INCLUDE  'ORG00000.ASM'    ;initialize reset vector
        INCLUDE  'ORG00008.ASM'    ;initialize exception vectors

            ORG  $00200            ;start program right after exception
                                   ;vectors

*****  Initialize  *****

        INCLUDE  'INITSYS.ASM'     ;initially set  EK=F, XK=0, YK=0, ZK=0
                                   ;set sys clock at 16.78MHz, disable COP
        INCLUDE  'INITRAM.ASM'     ;turn on internal SRAM at $10000
                                   ;set stack in bank 1 (SK=1, SP=03FE)

        LDAB #$00
        STAB PEPAR          ;define the Port E pins as I/O pins
        STAB PFPAR          ;define the Port F pins as I/O pins

        LDAB #$00
        STAB PORTE0         ;clear the Port E data register
        STAB PORTF0         ;clear the Port F data register

        LDAB #$FF
        STAB DDRE           ;initialize Port E pins as outputs
                                   ;note that Port E pin 3 does not exist!
        LDAB #$FF
```


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

```
        STAB DDRF           ;Port F pins 0-7 as outputs (force 3-7 to 0)

        LDAB #$01
        LDAA #$00

*****  Main Loop  *****

START:  STAB PORTF0         ;store counter into Port E data register
        STAA PORTF0         ;load A register with Port F data register
        BRA  START          ;go back and start the counting again at zero

***** Exceptions/Interrupts  *****

BDM:    BGND                ;exception vectors point here
                            ;and will put user into background mode
```

E.2.1.2 Example 2 - Using Chip-Selects

```
*       Description : Demo of how to set up the U1 and U3 RAM slots with
*                     two 32Kx8 RAM chips using the Chip Selects.  The new
*                     memory will start at address $30000 and will be both
*                     byte and word readable/writable.
*                     This program assumes that the RAM to be installed
*                     (such as MCM60L256AP85 or MCM6206P85) have
*                     access times of 85 ns and require no wait states.
*                     The DSACK field of the Chip Select Option Registers
*                     may need to be adjusted for chips that have faster or
*                     slower access times.
*************************************************************************

        INCLUDE         'EQUATES.ASM'   ;table of EQUates for common register
addresses
        INCLUDE         'ORG00000.ASM'  ;initialize reset vector
        INCLUDE         'ORG00008.ASM'  ;initialize exception vector table

            ORG     $0200               ;start program right after vector table

*****  Initialization  *****

INIT:                           ;Initialization stuff.....

        INCLUDE         'INITSYS.ASM'   ;initially set EK=F, XK=0, YK=0, ZK=0
                                        ;set sys clock at 16.78 MHz, disable
COP
        INCLUDE         'INITRAM.ASM'   ;turn on 1k internal SRAM at $10000
                                        ;set stack in bank 1 (SK=1, SP=03FE)
        INCLUDE         'INITSCI.ASM'   ;set SCI baud rate at 9600 baud
                                        ;enable SCI transmitter and receiver

CSINIT:                         ;Initialize the Chip Selects.....
        LDD     #$0303
        STD     CSBAR0          ;set U1 RAM base addr to $30000: bank 3, 64k
        STD     CSBAR1          ;set U3 RAM base addr to $30000: bank 3, 64k
        LDD     #$5030
        STD     CSOR0           ;set Chip Select 0, upper byte, write only
```


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

Freescale Semiconductor, Inc.

```
        LDD     #$3030
        STD     CSOR1           ;set Chip Select 1, lower byte, write only
        LDD     #$0303
        STD     CSBAR2          ;set Chip Select 2 to fire at base addr $30000
        LDD     #$7830
        STD     CSOR2           ;set Chip Select 2, both bytes, read and write
        LDD     #$3FFF
        STD     CSPAR0          ;set Chip Selects 0,1,2 to 16-bit ports

*****  The Main Program  *****

MAIN:                           ;Move data from another place in memory
                                ;into the U1 and U3 RAM slots.....
        LDAB    #$00
        TBXK                    ;set XK to bank 0 for access to the STRING
        LDX     #STRING         ;load the starting address of STRING into IX

        LDAB    #$03            ;set ZK to bank 3
        TBZK                    ;for access to U1 & U3 during write in XLOOP
        LDZ     #0000           ;clear IZ so ZK:IZ = $30000

XLOOP:  LDD     0,X             ;load two bytes from $10000 into accum. D
        STD     0,Z             ;store accum. D into U1 and U3 RAM:
                                ;the chip select logic takes care of us!
        AIX     #2              ;increment X index register to next word
        AIZ     #2              ;increment Z index register to next word
        CMPA    #$00
        BEQ     PRINT           ;end xloop if the end of the string $00 is
        CMPB    #$00            ;detected in either accumulator A or B
        BNE     XLOOP

PRINT:                          ;This loop reads its string from the U1 and U3
                                ;slots and prints it at the dummy terminal....
        LDAB    #$03
        TBXK                    ;set XK:IX index to point to bank 3
        LDX     #$0000          ;point to the beginning of the ASCII string
        JSR     SEND_STRING     ;go output the ASCII string
        BRA     PRINT           ;loop back and print again

*****  Subroutines  *****

SEND_STRING:                    ;subroutine to send out the entire ASCII
                                ;string
        LDAB    0,X             ;get next char in string as pointed to by IX
        BEQ     STRING_DONE     ;if B=00, then message is done
        JSR     SEND_CH         ;go send out a character
        AIX     #1              ;increment IX to point to the next ASCII char
        BRA     SEND_STRING     ;loop back
STRING_DONE:
        RTS                     ;go back to whence we came

SEND_CH:                        ;subroutine to send out one character at a
time
        LDAA    SCSR            ;read the SCI status reg to check TDRE bit
        ANDA    #01             ;check only the tdre flag bit
```


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

```
        BEQ     SEND_CH         ;if TDR is not empty, go back to check it
again
        CLRA
        STD     SCDR            ;transmit one ASCII character to the screen
TC_LOOP:
        LDAB    SCSR+1
        ANDB    #$80            ;test the TC bit (transfer complete)
        BEQ     TC_LOOP         ;continue to wait until TC is set

        RTS                     ;finish sending out one character

*****  Exceptions/Interrupts  *****

BDM:    BGND                    ;exception vectors point here
                                ;and put the user in background mode

*****  The string  *****

        ORG     $0310

STRING  DC 'I LIKE MY NEW MEMORY!!!',0A,0D,00
```

E.2.1.3 Example 3 - Changing Clock Frequencies

```
*       Description : This program demonstrates the ability of the
*                     M68HC16 to change clock frequencies on the fly.
*                     In this particular case, we alternate between
*                     a frequency of 16.78MHz and 4.194MHz.  Note
*                     that because we are writing to the screen,
*                     we also need to correct the BAUD rate (1200)
*                     each time we change the frequency.  Make sure
*                     that your terminal has been set up to receive
*                     at 1200 baud.  An oscilloscope may be connected
*                     to the CLKOUT pin on the EVB to observe the
*                     frequency change.
*************************************************************************

        INCLUDE         'EQUATES.ASM'   ;table of EQUates for common registers
        INCLUDE         'ORG00000.ASM'  ;initialize reset vectors
        INCLUDE         'ORG00008.ASM'  ;initialize interrupt vectors

            ORG     $0200               ;start program after the exception
                                        ;vector table

*****  Initialize  *****

INIT:

        INCLUDE         'INITSYS.ASM'   ;initially set EK=F, XK=0, YK=0, ZK=0
                                        ;set sys clock at 16.78 MHz, disable
COP
        INCLUDE         'INITRAM.ASM'   ;turn on internal 1K SRAM at $10000
                                        ;set stack in bank 1 (SK=1, SP=03FE)

        LDD     #$000C
```


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

```
        STD     SCCR1           ;enable SCI receiver and transmitter

        LDAB    #$01
        TBZK                    ;point ZK at bank 1 (the SRAM)
        LDZ     #$0000          ;for indexing the variables CNT and DLY

CNT     EQU     $0000           ;loop counter
DLY     EQU     $0002           ;delay counter

*****  Main Program  *****

MAIN:   LDAB    #$7F              ;set clock speed to 16.777MHz
        STAB    SYNCR             ;w=0, x=1, y=111111
NOT_L:  BRCLR   SYNCR+1,#8,NOT_L  ;wait until synthesizer lock bit is set
        LDD     #$01B5
        STD     SCCR0           ;set baud rate to 1200
        JSR     DELAY           ;delay for modulus counter of SCI to flush
        LDX     #STRING         ;load address of string into IX
        JSR     SEND_STRING     ;subroutine to send string to dummy terminal
        LDAB    #$05            ;set up loop counter
        STAB    CNT,Z
LOOP1:  LDX     #SEC_STR        ;load address of string into IX
        JSR     SEND_STRING     ;subroutine to send string to dummy terminal
        DEC     CNT,Z           ;decrement loop counter
        BNE     LOOP1           ;loop 5 times

        LDAB    #$4F              ;change clock frequency to 4.194MHz
        STAB    SYNCR             ;w=0, x=1, y=001111
LOOP2:  BRCLR   SYNCR+1,#8,LOOP2  ;wait until synthesizer lock bit is set
        LDD     #$006D
        STD     SCCR0           ;set BAUD rate back to 1200
        JSR     DELAY           ;delay for modulus counter of SCI to flush
        LDX     #STRING2        ;load address of string into IX
        JSR     SEND_STRING     ;subroutine to send string to dummy terminal
        LDAB    #$05            ;set up # of loops for loop counter to 5
        STAB    CNT,Z
LOOP3:  LDX     #SEC_STR        ;load address of string into IX
        JSR     SEND_STRING     ;subroutine to send string to dummy terminal
        DEC     CNT,Z           ;decrement loop counter
        BNE     LOOP3           ;loop 5 times

        LBRA    MAIN            ;branch back to main

*****  Subroutines  *****

DELAY:  LDD     #$FFFF          ;delay loop
        STD     DLY,Z
LOOP4:  DEC     DLY,Z
        BNE     LOOP4
        RTS

SEND_STRING:                    ;subroutine to send out the entire ASCII
                                ;string
        LDAB    0,X             ;get next byte in string as pointed to by IX
        BEQ     STRING_DONE     ;if B=00, then message is done
```

Freescale Semiconductor, Inc.

```
        JSR     SEND_CH         ;go send out the byte
        AIX     #$01            ;increment IX to point to the next byte
        BRA     SEND_STRING     ;loop back and do next byte in string
STRING_DONE:
        JSR     DELAY           ;wait for a moment
        RTS                     ;go back to whence we came

SEND_CH:                        ;subroutine to send out one byte to SCI
        LDAA    SCSR            ;read SCI status reg to check/clear TDRE bit
        ANDA    #$01            ;check only the TDRE flag bit
        BEQ     SEND_CH         ;if TDR is not empty, go back to check it
                                ;again
        LDAA    #$00            ;clear A to send a full word to SCDR ($FFC0E)
        STD     SCDR            ;transmit one ASCII character to the screen
LOOP5:  LDAB    SCSR+1
        ANDB    #$80            ;test the TC bit (transfer complete)
        BEQ     LOOP5           ;continue to wait until TC is set
        RTS                     ;return to send_string subroutine

*****  The STRINGs  *****

STRING  DC      'The System Clock is now running at 16.777 MHz...',0a,0d,00
SEC_STR DC      'check this out!',0a,0d,00,00
STRING2 DC      'The System Clock is now running at 4.194 MHz...',0a,0d,00


*****  Interrupts/Exceptions  *****

BDM:    BGND                    ;exception vectors point here
                                ;and put the user into background debug mode
```

E.2.1.4 Example 4 - Software Watchdog, Periodic Interrupt, and Autovector Demo

```
*       Description : This program demonstrates the software watchdog,
*                     the periodic interrupt, and an autovector.
*                     The periodic interrupt runs a clock which is updated
*                     on the dummy terminal.  Every eight seconds the COP
*                     will force a reset unless IRQ6 is grounded.  When IRQ6
*                     is pulled low, an autovectored interrupt routine will
*                     “feed” the watchdog with $55 and $AA, and the
*                     clock will run without being reset on the dummy terminal.
*
*************************************************************************

        INCLUDE         'EQUATES.ASM'   ;table of EQUates for common registers
        INCLUDE         'ORG00000.ASM'  ;initialize reset vectors
        INCLUDE         'ORG00008.ASM'  ;initialize interrupt vectors


        ORG     $002C           ;put address of autovector routine
        DC.W      AUTOV         ;at the level 6 vector (IRQ6)

        ORG     $0070           ;put address of periodic interrupt routine
        DC.W      VECRT         ;at 1st user defined interrupt vector
```


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

```
        ORG     $0200           ;start program after interrupt table


*****  Initialize  *****

INIT:                           ;initialization stuff

TEMP    EQU     $0006           ;variable space used in hex to asc routine
SCCNT   EQU     $0004           ;stores the current number of seconds
MNCNT   EQU     $0002           ;stores the current number of minutes
HRCNT   EQU     $0000           ;stores the current number of hours

INITSYS:                        ;The following section is normally part of
                                ;INCLUDE 'INITSYS.ASM', but we want to
                                ;leave the COP on.
        LDAB    #$0F
        TBEK                    ;point EK to bank F for register access
        LDAB    #$00
        TBXK                    ;point XK to bank 0
        TBYK                    ;point YK to bank 0
        TBZK                    ;point ZK to bank 0
        LDD     #$00CF          ;initialize the SIM MCR
        STD    SIMMCR           ;this is redundant; it happens at reset
        LDAA    #$7F
        STAA    SYNCR           ;set system clock to 16.78 Mhz
        LDAB    #$C0
        STAB    SYPCR           ;enable the watchdog (COP)
                                ;and set time-out period to 8 seconds

        INCLUDE         'INITRAM.ASM'   ;turn on internal SRAM at $10000
                                        ;set stack in bank 1 (SK=1, SP=03FE)
        INCLUDE         'INITSCI.ASM'   ;set SCI baud rate at 9600 baud
                                        ;enable SCI transmitter and receiver

        LDD     #$0638
        STD     PICR            ;set the periodic interrupt at request level 6
                                ;& assign vector #56 (address $00070) to it
        LDD     #$0110
        STD     PITR            ;initialize PITR to interrupt every 1 sec
        LDD     #$FFF9          ;initialize Chip Sel Base Reg for Autovector
        STD     CSBAR3          ;on an IACK cycle: A24-A11=$FFF8, blK_sz = 8K
        LDD     #$7801          ;initialize Chip Sel Option Reg for
Autovector:
        STD     CSOR3           ;asynchronous, any Interrupt Priority Level
        LDAB    #$FF            ;set port F pins to be IRQ pins
        STAB    PFPAR           ;this is redundant: it happens at reset
        LDAB    #$01
        TBZK                    ;set zk nibble to bank 1 for variable access
        LDZ     #$0000          ;index those variables from $10000
        LDD     #$0000
        STD     SCCNT,Z         ;clear sccnt register
        STD     MNCNT,Z         ;clear mncnt register
        STD     HRCNT,Z         ;clear hrcnt register
        STD     TEMP,Z          ;clear “save” variable space
        LDAB    RSR             ;find out who caused the last reset
```


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

Freescale Semiconductor, Inc.

```
        CMPB    #$20
        BNE     NO_DOG          ;choose which string start address to load
        LDX     #DOG_STRING     ;COP caused the reset
        BRA     PRINT
NO_DOG: LDX     #NO_DOG_STR     ;COP did not cause the reset

PRINT:  JSR     SEND_STRING     ;print the string to the screen

        ANDP    #$FF1F          ;set interrupt priority mask level to 0


*****  The Main Program  *****

MAIN:   NOP
        BRA     MAIN            ;keep on looping until watchdog causes a reset
                                ;or we get some other interrupt
*****  Subroutines  *****

SEND_STRING:                    ;subroutine to send out the entire ASCII
string
        LDAB    0,X             ;get next byte in string as pointed to by IX
        BEQ     STRING_DONE     ;if B=00, then we're done
        JSR     SEND_CH         ;go send out the byte
        AIX     #$01            ;increment IX to point to the next byte
        BRA     SEND_STRING     ;loop back and do next byte in string
STRING_DONE:
        RTS                     ;go back to whence we came

SEND_CH:                        ;subroutine to send out one byte to SCI
        LDAA    SCSR            ;read SCI status reg to check/clear TDRE bit
        ANDA    #$01            ;check only the TDRE flag bit
        BEQ     SEND_CH         ;if TDR is not empty, go back to check it
again
        LDAA    #$00            ;clear A to send a full word to SCDR ($FFC0E)
        STD     SCDR            ;transmit one ASCII character to the screen
TC_LOOP:
        LDAB    SCSR+1
        ANDB    #$80            ;test the TC bit (transfer complete)
        BEQ     TC_LOOP         ;continue to wait until TC is set
        RTS                     ;finish sending out byte

*****  The STRINGs  *****

DOG_STRING   DC 'The Software Watchdog just caused a reset!',0a,0d,00
NO_DOG_STR   DC 'The last reset was not caused by the COP.',0a,0d,00
AUTOV_STRING DC 'Feeding the dog...',0a,0d,00

*****  Periodic Interrupt Vector Routine  *****

VECRT:                          ;When the processor is interrupted by the
                                ;periodic timer, it will run this routine.
                                ;This routine simply increments a clock
                                ;every time it is interrupted, and prints
                                ;the clock out on the dummy terminal.
SECONDS:                        ;advance the counter for seconds
```

Freescale Semiconductor, Inc.

```
        LDAA    SCCNT,Z
        ADDA    #$01            ;increment # of seconds
        DAA                     ;decimal adjust A
        CMPA    #$60            ;compare # of seconds to 60
        BEQ     MINUTES         ;if # of sec=60, then branch to minute routine
        STD     SCCNT,Z         ;if # of sec<60, store new # of sec
        JSR     DISPLAY         ;send new time to dummy terminal for display
        RTI                     ;return to main loop & wait for next interrupt

MINUTES:                        ;advance counter for minutes
        CLR     SCCNT,Z         ;set # of seconds to 0
        LDAA    MNCNT,Z
        ADDA    #$01            ;increment # of minutes
        DAA                     ;decimal adjust A
        CMPA    #$60            ;compare # of minutes to 60
        BEQ     HOURS           ;if # of min=60, then branch to hours routine
        STD     MNCNT,Z         ;if # of min<60, then store new # of min
        JSR     DISPLAY         ;send new time to dummy terminal for display
        RTI                     ;return to main loop & wait for next interrupt
HOURS:                          ;advance counter for hours
        CLR     MNCNT,Z         ;set # of minutes to 0
        LDAA    HRCNT,Z
        ADDA    #$01            ;increment # of hours
        DAA                     ;decimal adjust A
        STD     HRCNT,Z         ;store new # of hours
        CMPA    #$24            ;compare # of hours to 24
        BNE     RETURN          ;if # of hours < 24, then display new time
        CLR     HRCNT,Z         ;if # of hours=24 then clear # of hours

RETURN: JSR     DISPLAY         ;send new time to dummy terminal for display
        RTI                     ;return to main loop & wait for next interrupt

*****  Send Time to Dummy Terminal Routine  *****

DISPLAY:                        ;this routine takes what is stored in sccnt,
                                ;mncnt and hrcnt, converts them to ASCII
                                ;characters, and then sends them to a dummy
                                ;terminal.
SEND_HR:                        ;output the hours
        LDAB    HRCNT,Z
        JSR     HEXTOASC        ;convert hex number into ASCII and print

SEND_COL:                       ;output a colon
        LDAB    #$3A            ;ASCII number for a colon
        JSR     SEND_CH         ;send character to terminal

SEND_MIN:                       ;output the minutes
        LDAB     MNCNT,Z
        JSR     HEXTOASC        ;convert hex number to ASCII and print

SN_COLON:                       ;output another colon
        LDAB    #$3A            ;ASCII number for a colon
        JSR     SEND_CH         ;send character to terminal

SEND_SEC:                       ;output the seconds
```

Freescale Semiconductor, Inc.

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

```
        LDAB    SCCNT,Z
        JSR     HEXTOASC        ;convert hex number to ASCII and print

LINE_FD:                        ;output a linefeed
        LDAB    #$0A            ;load ASCII number for line feed
        JSR     SEND_CH         ;send character to terminal

CARRIAGE:                       ;output a carriage return
        LDAB    #$0D            ;ASCII number for carriage return
        JSR     SEND_CH         ;send character to terminal

        RTS                     ;done with display routine

*****  Hexadecimal to ASCII conversion  *****

HEXTOASC:                       ;the following code takes a number or
                                ;character stored in register D, assumes
                                ;it's in its hexadecimal form and converts
                                ;it to an ASCII equivalent. It also sends
                                ;that character to the screen.
        STD     TEMP,Z          ;store the hex number temporarily into “TEMP”
        JSR     PRTMSB
        LDD     TEMP,Z          ;reload value of hex number into D register
        ANDB    #$0F            ;get rid of upper 4 bits in hex number
        BRA     PRTLSB
PRTMSB: LSRB                    ;shift high 4 bits down to low 4 bits position
        LSRB
        LSRB
        LSRB
PRTLSB:                         ;the actual conversion process:
        ADDB    #$30            ;add $30 to the hex number
        CMPB    #$39            ;check for digithood
        BLS     NOTAF           ;go print now if it's a digit 0-9
        ADDB    #$07            ;it's a letter A-F, so add $07 before printing
NOTAF:  JSR     SEND_CH         ;send the character to the SCI
        RTS                     ;done with hex to ascii conversion

*****  Autovector Routine  *****

AUTOV:                  ;when IRQ6 is low, this autovector routine starts
        LDAB #$55       ;These four lines reset the watchdog and keep it
        STAB SWSR       ;from causing a system reset by writing to the SWSR
        LDAB #$AA       ;By writing a #$55 and a #$AA to the SWSR before
                        ;the end
        STAB SWSR       ;of every time-out period, the watchdog will be
reset.
        LDX  #AUTOV_STRING
        JSR  SEND_STRING
        RTI                     ;return to the main loop

*****  Other exception/interrupts  *****

BDM:    BGND                    ;all other exception vectors point here
                                ;and put the user into background mode
```


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

E.2.2 CPU16 Programming Example

The following programming example involves using the CPU16 indexed and extended addressing modes.

Refer to **SECTION 4 CENTRAL PROCESSOR UNIT** for more information on the CPU16.

E.2.2.1 Example 5 - Indexed and Extended Addressing

```
*       Description : This program demonstrates indexed and extended
*                     addressing.
*************************************************************************
****
        INCLUDE         'EQUATES.ASM'       ;table of EQUates for common
register addresses
        INCLUDE         'ORG00000.ASM'      ;initialize reset vector
        INCLUDE         'ORG00008.ASM'      ;initialize interrupt vectors

           ORG     $0200           ;start program after interrupt vectors

*****  Initialization Routines  *****

        INCLUDE         'INITSYS.ASM'       ;initially set EK=F, XK=0, YK=0,
ZK=0
                                            ;set sys clock at 16.78MHz, disable
COP
        INCLUDE         'INITRAM.ASM'       ;initialize and turn on SRAM
                                            ;set stack (SK=#$1, SP=#$03FE)

*****  Start of main program  *****

           LDAB    #$00
           TBZK                    ;point ZK to bank 0
           LDZ     #$FFFE          ;set IZ=#$FFFE
OFFSET     EQU     $02             ;SET OFFSET = $02
           LDAB    #$01
           TBEK                    ;point EK to bank 1
GO:        LDAA    #$00
           STAA    $0000           ;write 00 to $10000 (extended)
           LDAA    #$FF
           STAA    OFFSET,Z        ;write ff to $10000 (indexed)
           BRA     GO

*****  Exceptions/Interrupts  *****

BDM:       BGND                    ;exception vectors point here
                                   ;and put the user in background mode
```

Freescale Semiconductor, Inc.

E.2.3 QSM/SCI Programming Example

The following programming example involves using a port of the serial communication interface (SCI), one of the serial interfaces of the queued serial module (QSM), to display a message on a dummy terminal.

Refer to **SECTION 9 QUEUED SERIAL MODULE** for more information on the QSM or the SCI.

E.2.3.1 Example 6 - Using an SCI Port

```
*    Description : This program uses the SCI port to display
*                  a shameless message on a dummy terminal.  It includes
*                  a subroutine to print a single character to the SCI
*                  and a subroutine that uses the single character
*                  subroutine to print an entire string.
*
*************************************************************************

        INCLUDE  'EQUATES.ASM'     ;table of EQUates for common register
addresses
        INCLUDE  'ORG00000.ASM'    ;initialize reset vector
        INCLUDE  'ORG00008.ASM'    ;initialize interrupt vectors

         ORG  $0200                ;start program after exception vector table

*****  Initialize  *****

INIT:

        INCLUDE  'INITSYS.ASM'     ;initially set EK=F, XK=0, YK=0, ZK=0
                                   ;set sys clock at 16.78 MHz, disable COP
        INCLUDE  'INITRAM.ASM'     ;turn on internal SRAM at $10000
                                   ;set stack (SK=1, SP=03FE)
        INCLUDE  'INITSCI.ASM'     ;set the SCI baud rate to 9600 baud
                                   ;enable the SCI receiver and transmitter

        LDAB #$00
        TBXK                       ;set XK to bank 0 for STRING access
        LDAB #$01
        TBZK                       ;set ZK to bank 1 for delay counter access
        LDZ  #$0000                ;clear IZ for later use with delay counter

*****  Main Program  *****

MAIN    LDX  #STRING               ;point to the beginning of ASCII string
        JSR  SEND_STRING           ;go output the ASCII string
        BRA  MAIN                  ;branch back to main

*****  Subroutines  *****

SEND_STRING:               ;subroutine to send out the entire ASCII string
        LDAB 0,X           ;get next byte in string as pointed to by IX
        BEQ  STRING_DONE   ;if B=00, then goto delay between messages
        JSR  SEND_CH       ;go send out the byte
```


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

```
        AIX  #$01           ;increment IX to point to the next byte
        BRA  SEND_STRING    ;loop back and do next byte in string

STRING_DONE:                ;subroutine to implement delay between messages
        LDE  #$FFFF         ;load accumulator E with the delay time
        STE  0,Z            ;set up the counter
LOOP:   DECW 0,Z            ;decrement the counter
        BNE  LOOP           ;count down to zero
        RTS                 ;finish delay loop go back to main

SEND_CH:                    ;subroutine to send out one byte to SCI
        LDAA SCSR           ;read SCI status reg to check/clear TDRE bit
        ANDA #$01           ;check only the TDRE flag bit
        BEQ  SEND_CH        ;if TDR is not empty, go back to check it again
        LDAA #$00           ;clear A to send a full word to SCDR ($FFC0E)
        STD  SCDR           ;transmit one ASCII character to the screen
TC_LOOP:
        LDAB SCSR+1
        ANDB #$80           ;test the TC bit (transfer complete)
        BEQ  TC_LOOP        ;continue to wait until TC is set

        RTS                 ;finish sending out byte

STRING    DC     'I AM A HAPPY EVB16 RUNNING YOUR CODE!!!',0A,0D,00

*****  Interrupts/Exceptions  *****

BDM: BGND                   ;exception vectors point here
                            ;and put the user in background debug mode

*****  Reserve data and stack space  *****

        ORG  $10000         ;start of 1K internal SRAM for data & stack

COUNTER DS     2           ;space for delay counter
```

E.2.4 GPT Programming Example

The following programming example involves demonstrating basic general-purpose timer module (GPT) functions.

Refer to **SECTION 11 GENERAL-PURPOSE TIMER** for more information on the GPT.

E.2.4.1 Example 7 - Basic GPT Functions

```
*       Description : This program demonstrates some basic GPT functions.
*
*                     The 1st demo requires that the pins OC2, IC1, IC2,
*                     and IC3 be tied together so that OC2 may drive
*                     IC1, IC2, & IC3.
*
*                   * In the second demo, the PAI pin should be connected
*                     to the PWMA pin.  A bell on the dummy terminal
*                     will ring when the Pulse Accumulator Counter
```


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

Freescale Semiconductor, Inc.

```
*                    has overflowed ten times.
*
*************************************************************************

        INCLUDE      'EQUATES.ASM'   ;table of EQUates for common register
                                     ;addresses
        INCLUDE      'ORG00000.ASM'  ;initialize reset vector
        INCLUDE      'ORG00008.ASM'  ;initialize interrupt vectors

*       We are choosing User Defined Interrupt Vector 9 (interrupt vector 64
*       at address $0080) to be the base vector number (VBA) for the GPT
*       because the least significant nibble in the address must be a $0.
*
*        The VBA should be reflected in the GPT Interrupt Configuration
*        Register (ICR) at $YFF904.

        ORG     $0080             ;Address for interrupt vector 64

        DC.W      PAOV_ROUTINE    ;Adjusted Priority Channel -- PAC
        DC.W      IC1_ROUTINE     ;Input Capture 1
        DC.W      IC2_ROUTINE     ;Input Capture 2
        DC.W      IC3_ROUTINE     ;Input Capture 3
        DC.W      BDM             ;Output Compare 1
        DC.W      OC2_ROUTINE     ;Output Compare 2
        DC.W      BDM             ;Output Compare 3
        DC.W      BDM             ;Output Compare 4
        DC.W      BDM             ;Input Capture 4 / Output Compare 5
        DC.W      BDM             ;Timer Overflow
        DC.W      BDM             ;Pulse Accumulator Overflow -- elevated
        DC.W      BDM             ;Pulse Accumulator Input

        ORG     $0200             ;start program after interrupt vectors

*****  Initialization Routines  *****

        INCLUDE        'INITSYS.ASM'   ;initially set EK=F, XK=0, YK=0, ZK=0
                                       ;set sys clock at 16.78 MHz, disable
COP
        INCLUDE        'INITRAM.ASM'   ;turn on 1k internal SRAM at $10000
                                       ;set stack in bank 1 (SK=1, SP=03FE)
        INCLUDE        'INITSCI.ASM'   ;set SCI baud rate at 9600
                                       ;enable SCI transmitter and receiver

*       Set up the interrupts

        LDD     #$008E          ;Give the GPT an IARB of $E
        STD     GPTMCR          ;so we can generate interrupts
        LDD     #$A640          ;elevate interrupt priority of PAOV,
        STD     ICR             ;set GPT IRQ level to 6,
                                ;& assign vector 64 (User vector 9) of the
                                ;interrupt/exception vector table as the
                                ;GPT's Interrupt Vector Base Address
        LDAB    #$17            ;set OC2, IC1, IC2, IC3 to generate interrupts
        STAB    TMSK1
        LDAB    #$25            ;set PAC Overflows to generate interrupts
```

Freescale Semiconductor, Inc.

```
        STAB    TMSK2           ;& set the TCNT's prescale to sysclock/128

*       Set up Input Capture and Output Compare

        LDAB    #$27            ;Input Captures
        STAB    TCTL2           ;TIC1=either, TIC2=rise, TIC3=fall, TIC4=off
        LDAB    #$01            ;Output Compares
        STAB    TCTL1           ;TOC2=toggle, TOC3=off, TOC4=off, TOC5=off
        LDD     #$1000          ;set OC2 to toggle every time that
        STD     TOC2            ;TCNT is #$1000

*       Set up the Pulse Width Modulators A and B

        LDD     #$0064          ;set PWM prescaler to div by 128
        STD     PWMC            ;set PWMA fast (512 Hz)
                                ;and PWMB slow (4 Hz)
        LDAB    #$80            ;set 50% duty cycle
        STAB    PWMA            ;in PWMA
        STAB    PWMB            ;in PWMB

*       Set up the Pulse Accumulator

        LDD     #$5000          ;set PAC to sense rising edges in
        STD     PACTL           ;event counting mode

*       Other Initializations

        LDAB    #$00
        TBXK                    ;set XK to bank 0 for STRING access
PAOV_CNT EQU    0               ;counter variable for PAOV_ROUTINE
        LDAB    #$01
        TBZK
        LDZ     #$0000          ;PAOV_CNT will be indexed off ZK:IZ
        LDAB    #$0A
        STAB    PAOV_CNT,Z      ;load a 10 into the variable
        ANDP    #$FF1F          ;set interrupt priority mask level to 0

*****  Start of main program  *****

GO:     NOP
        BRA     GO              ;Let's loop until we're interrupted
*****  Subroutines  *****

SEND_STRING:
            EVEN                    ;subroutine to send out the entire ASCII
string
        LDAB    0,X             ;get next byte in string as pointed to by IX
        BEQ     STRING_DONE     ;if B=00, then the string is done
        JSR     SEND_CH         ;go send out the byte
        AIX     #$01            ;increment IX to point to the next byte
        BRA     SEND_STRING     ;loop back and do next byte in string
STRING_DONE:
        RTS                     ;go back to whence we came
SEND_CH:                        ;subroutine to send out one byte to SCI
        LDAA    SCSR            ;read SCI status reg to check/clear TDRE bit
```

Freescale Semiconductor, Inc.

```
        ANDA    #$01            ;check only the TDRE flag bit
        BEQ     SEND_CH         ;if TDR is not empty, go back to check it
again
        LDAA    #$00            ;clear A to send a full word to SCDR ($FFC0E)
        STD     SCDR            ;transmit one ASCII character to the screen
TC_LOOP:
        LDAB    SCSR+1
        ANDB    #$80            ;test the TC bit (transfer complete)
        BEQ     TC_LOOP         ;continue to wait until TC is set
        RTS                     ;finish sending out byte

*****  The STRINGS  *****

STRING_IC1      DC.W      'Input capture 1 caught a transition',0a,0d,00
STRING_IC2      DC.W      'Input capture 2 caught a rising edge',0a,0d,00
STRING_IC3      DC.W      'Input capture 3 caught a falling edge',0a,0d,00
STRING_OC2      DC.W      0a,'Output compare 2 just toggled',0a,0d,00
STRING_PAOV     DC.W      0a,'Pulse Accum. has overflowed 10
times!',07,0a,0d,00

*****  Exceptions/Interrupts  *****

*       Note that every one of the GPT interrupt service routines clears
*       its flag bit at the end of the routine before the RTI instruction.

  EVEN

IC1_ROUTINE:                    ;execute when IC1 senses a transition
        LDX     #STRING_IC1
        JSR     SEND_STRING     ;print the message
        BCLR    TFLG1,#$01      ;clear the IC1 flag bit
        RTI

IC2_ROUTINE:                    ;execute when IC2 senses a rising edge
        LDX     #STRING_IC2
        JSR     SEND_STRING     ;print the message
        BCLR    TFLG1,#$02      ;clear the IC2 flag bit
        RTI

IC3_ROUTINE:                    ;execute when IC3 senses a falling edge
        LDX     #STRING_IC3
        JSR     SEND_STRING     ;print the message
        BCLR    TFLG1,#$04      ;clear the IC3 flag bit
        RTI

OC2_ROUTINE:                    ;execute when OC2 does a toggle
        LDX     #STRING_OC2
        JSR     SEND_STRING     ;print the message
        BCLR    TFLG1,#$10      ;clear the OC2 flag bit
        RTI

PAOV_ROUTINE:                   ;execute on Pulse Accumulator Counter overflow
                                ;if PAI pin tied PWMA, bell approx every 5 sec
                                ;if PAI pin tied PWMB, bell approx every 10
min
```

Freescale Semiconductor, Inc.

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

```
        DEC     PAOV_CNT,Z
        BNE     PAOV_DONE       ;skip print routine if not finished
                                ;counting down from ten
        LDX     #STRING_PAOV
        JSR     SEND_STRING     ;print the message
        LDAB    #$0A
        STAB    PAOV_CNT,Z      ;reload the counter so we can do it again
PAOV_DONE:
        BCLR    TFLG2,#$20      ;clear the PAOV flag bit
        RTI                     ;all done!

BDM:    BGND                    ;all other exception vectors point here
                                ;and put the user in background mode
```

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

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

INDEX

–A–

ABIU 8-3
AC timing
  16.78 MHz A-21
  20.97 MHz A-23
  25.17 MHz A-25
  low voltage, 16.78 MHz A-19
Accumulator
  M overflow flag (MV) 4-4, D-3
  offset addressing mode 4-10
ADC 8-1
  AC characteristics A-65
    low voltage A-63
  address map D-29
  analog subsystem 8-4
  block diagram 8-2
  bus interface unit (ABIU) 8-3
  clock 8-6
  conversion
    accuracy diagram
      10-bit A-71
      8-bit A-69
      low voltage
        10-bit A-70
        8-bit A-68
    control logic 8-7
      modes 8-8
      multiple-channel conversions 8-11
      parameters 8-8
      single-channel conversions 8-10
    timing 8-12
  DC electrical characteristics
    5 V A-64
    low voltage A-63
  digital control subsystem 8-6
  external connections 8-1
  features 3-2
  maximum ratings A-62
  operating characteristics A-67
    low voltage A-66
  overview 8-1
  prescaler 8-6
  programmer's model 8-3
  registers
    control registers (ADCTL) 8-6, D-31, D-32
    left justified
      signed (LJSRR) D-36
      unsigned (LJURR) D-37
    module configuration register (ADCMCR) 8-3, D-30
    port ADA data register (PORTADA) D-30
    result registers 8-13
    right justified, unsigned (RJURR) D-36
    status register (ADCSTAT) 8-6, D-36
    test register (ADCTEST) D-30
  special operating modes 8-3
ADCMCR 8-1, 8-3, D-30
ADCSTAT D-36
ADCTEST D-30
ADCTL D-31, D-32
ADCTST 8-1
ADDD 4-9
ADDE 4-9
ADDR
  bus signals 5-31
  definition 2-6
  signal 5-35
  starting address D-18
Address
  bus (ADDR) 5-31
  extension 4-6
    fields 4-5
    register 4-5
  map 3-18
  -mark wakeup 9-30, 10-22
  space
    encoding 5-32
    maps 3-19
  strobe ($\overline{AS}$) 5-31
Addressing modes 4-8
  accumulator offset 4-10
  extended 4-10
  immediate 4-9
  indexed 4-10
  inherent 4-10
  post-modified index 4-10
  relative 4-10
  replacing direct mode 4-11
AIS 4-9
AIX/Y/Z 4-9
Analog
  input
    circuitry 8-15
    considerations 8-19
    pins 8-2, 8-21
      electrical model 8-21
  power pins 8-14
  reference pins 8-3, 8-14
  subsystem 8-4
  supply
    filtering and grounding 8-16

Freescale Semiconductor, Inc.

pins 8-3
-to-digital converter (ADC). *See* ADC 8-1
Arbitration 9-3
AS 4-41, 5-31, 5-40, 5-43, 5-45, 5-47, 5-54
ASPC 7-2, 7-3, D-26
Asserted (definition) 2-6
Asynchronous exceptions 4-39
Autocorrelation 4-45
Autovector enable (AVEC). *See* AVEC 5-24
Auxiliary timer clock input (PCLK) 11-8
AVEC 5-24, 5-33, 5-43, 5-54, 5-60, 5-65, 5-67, 5-68, D-21

–B–

Background
debug mode 4-40, 4-42, 5-41
commands 4-43
connector pinout 4-45
enabling 4-42
entering 4-42
recommended connection 4-45
serial
I/O block diagram 4-44
interface 4-44
sources 4-42
timing
16.78 MHz A-37
20.97 MHz A-38
25.17 MHz A-38
freeze assertion A-39
low voltage, 16.78 MHz A-37
serial communication A-39
Basic operand size 5-35
Baud
clock 9-26, 10-18
rate generator 9-2
BCD 4-6
BERR 5-33, 5-37, 5-41, 5-43, 5-44, 5-45, 5-54, 5-60
BG 5-46, 5-49, 5-54, 5-65
BGACK 5-46, 5-49, 5-54, 5-65
Binary
coded decimal (BCD) 4-6
-weighted capacitors 8-5
BITS D-47
encoding field 9-18
Bits per transfer
enable (BITSE) D-52
field (BITS) D-47
BITSE 9-20, D-52
Bit-time 9-25, 10-17
BKPT 4-41, 5-41, 5-49, 5-52, 5-53, 5-57
Block size (BLKSZ) 5-65, D-18
encoding 5-65, D-18
BME 5-25, D-13
BMT 5-24, D-13
BOOT D-26
Boot ROM control (BOOT) 7-3, D-26
Bootstrap words (ROMBS) 7-1
BR 5-46, 5-49, 5-54, 5-64, 5-65
Break frame 9-25, 10-17
Breakpoint
acknowledge cycle 5-41
exceptions 4-40
hardware breakpoints 5-41
mode selection 5-52
operation 5-42
Breakpoints 4-41
Buffer amplifier 8-5
Built-in emulation memory C-1
Bus
arbitration
for a single device 5-46
timing — active A-33
timing — idle A-34
cycle
regular 5-37
terminations for asynchronous cycles 5-44
error
exception processing 5-44
signal (BERR). *See* BERR. 5-24
timing of 5-44
exception control cycles 5-43
grant (BG). *See* BG 5-46
grant acknowledge (BGACK). *See* BGACK 5-46
monitor 5-24
external enable (BME) D-13
timeout period 5-25
timing (BMT) 5-24, D-13
request (BR). *See* BR 5-46
state analyzer 4-40
BYTE (upper/lower byte option) 5-66, D-19

–C–

C 4-4, D-3
Capture/compare unit 11-1
block diagram 11-11
clock output enable (CPROUT) bit D-73
Carry flag (C) 4-4, D-3
Case outlines
132-pin package B-4
144-pin package B-7
CCF D-36
CCR 4-4, D-3
CCTR D-36
CD/CA D-33
C_{DAC} 8-22
Central processing unit (CPU16). *See* CPU16 4-1
C_F 8-22
CFORC 11-8, 11-13, 11-14, D-74
Channel selection for A/D conversion D-33
Charge sharing 8-23
Chip-select
base address registers (CSBAR) 5-64, 5-65
reset values 5-69
operation 5-67
option registers (CSOR) 5-64, 5-66, D-18
reset values 5-69
pin assignment registers (CSPAR) 5-63, D-17
field encoding 5-64

Freescale Semiconductor, Inc.

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

reset operation 5-69
signals for interrupt acknowledge 5-68
timing diagram A-36
Clear (definition) 2-6
CL_I 4-37
Clipping errors 8-16
CLKOUT 5-36, 5-48
output timing diagram A-28
CLKRST (clock reset) 5-48
CL_O 4-37
Clock
ADC 8-6
control multipliers
16.78 MHz 5-9
20.97 MHz 5-11
25.17 MHz 5-13
frequency (calculation) D-7
mode
pin (MODCLK) 5-52
selection 5-52
modes
fast reference option (4.194 MHz) 5-4, 5-5
slow reference option (32.768 kHz) 5-4, 5-5
output (CLKOUT) 5-36
phase (CPHA) 10-8, D-47
= 0 transfer format 10-9
= 1 transfer format 10-10
polarity (CPOL) 10-8, D-47
synthesizer
operation 5-6
timing
16.78 MHz A-7
20.97 MHz A-8
25.17 MHz A-9
low voltage A-6
Closed-loop control routines 4-45
CL_P 4-37
CL_T 4-37
Coherency 11-10, 11-12
Combined program and data space map
MC68HC16Z1/CKZ1/CMZ1 3-20
MC68HC16Z2/Z3 3-21
MC68HC16Z4/CKZ4 3-22
Command RAM 9-8
Comparator 8-6
Completed queue pointer (CPTQP) D-51
Condition code register (CCR) 4-4, 11-6
CONT D-52
Contention 5-60
Continue (CONT) D-52
Continuous transfer mode 9-6
Conventions 2-6
Conversion
complete flags (CCF) D-36
control logic 8-7
modes 8-8
multiple-channel conversions 8-11
parameters 8-8
single-channel conversions 8-10
counter (CCTR) D-36
timing 8-12
CPHA 9-17, 10-8, 10-9, 10-10, D-47
CPOL 9-17, 10-8, D-47
CPR D-73
CPROUT 11-10, D-73
CPTQP 9-8, D-51
CPU space 5-68
address encoding 5-41
cycles 5-40, 5-68
encoding for interrupt acknowledge 5-68
CPU16 4-1
accumulators 4-3
address extension register 4-5
addressing modes 4-8
accumulator offset 4-10
extended 4-10
immediate 4-9
indexed 4-10
inherent 4-10
post-modified index 4-10
relative 4-10
breakpoints 4-41
compatibility with M68HC11 4-1
condition code register (CCR) 4-4
data types 4-6
extension fields 4-6
features 3-1
general information 4-1
index registers 4-3
instruction 4-11
comparison to M68HC11 4-31
execution model 4-34
set
abbreviations and symbols 4-30
summary 4-12
timing 4-36
levels of interrupt priority 5-58
memory
management 4-5
organization 4-7
program counter (PC) 4-3
reference manual 4-1
register model 4-2, D-2
registers
condition code register (CCR) D-3
mnemonics 2-2
multiply and accumulate (MAC) registers 4-5
stack pointer (SP) 4-3
CR D-52
CREG D-22
Cross-correlation 4-45
CSBAR D-17
CSBARBT D-17
$\overline{\text{CSBOOT}}$ 5-57, 5-63, 5-65, 5-66, 5-69, 5-70, 7-3
reset values 5-70
CSOR D-18
CSORBT D-18
CSPAR0/1 D-15

Freescale Semiconductor, Inc.

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

–D–

DAC capacitor array (C_{DAC}) 8-22
DATA 5-31
Data
  and size acknowledge (DSACK). *See* DSACK 5-24
  bus
    mode selection 5-50
    signals (DATA) 5-31
  frame 9-25, 10-17
  multiplexer 5-35
  strobe (DS). *See* DS 5-31
DATA (definition) 2-6
DC characteristics
  16.78 MHz A-12
  20.97 MHz A-14
  25.17 MHz A-16
  low voltage, 16.78 MHz A-10
DDRE 5-70, D-9
DDRF 5-70, D-11
DDRGP 11-8, 11-14, D-69
DDRM D-58
DDRQS 9-4, 9-16, 9-20, D-45
Delay
  after transfer (DT) 9-18, D-53
  before SCK (DSCKL) D-49
Designated CPU space 5-32
Design-Net database B-8
Development
  support for CPU16 4-40
  tools and support C-1
Digital
  control subsystem 8-6
  signal processing (DSP) 4-45
Divider/counter 5-6
Double
  -buffered 9-27, 9-28, 10-19, 10-20
  bus fault 5-45
DREG D-22
Driver types 3-12
DS 4-41, 5-31, 5-37, 5-40, 5-45, 5-47, 5-51
DSACK 5-24, 5-32, 5-37, 5-41, 5-43, 5-54, 5-60, 5-65, 5-66, 5-67, 5-68
  external/internal generation 5-40
  option fields 5-40
  signal effects 5-34
  source specification in asynchronous mode 5-66, D-19
DSCK D-53
DSCKL D-49
DSCLK 4-44, 5-53
DSI 5-53
DSO 5-53
DSP 4-45
DT D-53
DTL D-49
Dynamic bus sizing 5-33

–E–

EBI 5-60
ECLK 5-21
  bus timing
    16.78 MHz A-41
    20.97 MHz A-42
    25.17 MHz A-43
    low voltage A-40
  output timing diagram A-28
  timing diagram A-44
EDGE D-72
Edge-detection logic 11-12
EDGExA/B 11-12
EDIV 5-21, D-8
EK 4-5
Electrical characteristics A-1
EMUL D-26
Emulation mode control (EMUL) D-26
Ending queue pointer (ENDQP) D-50
ENDQP 9-8, D-50
EQUATES.ASM E-2
Error
  conditions 9-28, 10-21
  detection circuitry 9-2
EV 4-4
Event counting mode 11-15
Exception
  asynchronous 4-39
  definition 4-37
  multiple 4-40
  processing 5-48
    sequence 4-39
  stack frame 4-38
    format 4-38
  synchronous 4-39
  types 4-39
  vector 4-37, 5-48
    table 4-38
Execution
  process 4-36
  unit 4-35
EXOFF D-6
EXT D-8
EXTAL 5-5, 5-56
Extended addressing modes 4-10
Extension
  bit overflow flag (EV) 4-4
  field (SK) 4-3
  fields 4-6
External
  bus
    arbitration 5-46
    clock
      division bit (EDIV) 5-21, D-8
      operation during LPSTOP 5-21
      signal (ECLK) 5-21
    interface (EBI) 5-29
      control signals 5-31
  circuit settling time 8-23

Freescale Semiconductor, Inc.

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

clock
input signal (PCLK) 11-1
input timing diagram A-28
off (EXOFF) bit D-6
leakage 8-23
multiplexing of analog signal sources 8-20
reset (EXT) D-8
EXTRST (external reset) 5-55

–F–

F1A/B D-76
F1x bits 11-8
Factory test 5-71
Fast
mode 11-17
reference 5-4
reference circuit 5-5
termination cycles 5-36, 5-39
read cycle timing diagram A-31
write cycle timing diagram A-32
FC 5-32
FE 9-28, 10-21, D-44, D-63
Ferrite beads 8-14
f_{limp} 5-55
FOC 11-14, D-75
Force
compare bits (FOC) 11-14
logic level one on
PWMA/B (F1A/B) bit D-76
output compare (FOC) bit D-75
FPWMx 11-8
Frame 9-25, 10-17
size 9-29, 10-21
Framing error (FE) flag 9-28, 10-21, D-44, D-63
Free-running counter (TCNT) 11-1
FREEZE 4-43
assertion response (FRZ)
ADC 8-4, D-30
GPT 11-3, D-68
QSM 9-3, D-39
SIM 5-3
bus monitor (FRZBM) 5-3, D-6
software enable (FRZSW) 5-3, D-6
f_{ref} 5-5, 5-7
Frequency control bits
counter (Y) D-8
prescaler (X) D-8
VCO (W) D-8
FRZ 8-4, 11-3, D-30, D-39, D-68
FRZBM 5-3, D-6
FRZSW 5-3, D-6
f_{sys} 5-5, D-7
F-term encoding 5-40
Function code (FC) signals 5-32, 5-40
f_{VCO} 5-7

–G–

Gain 8-19
Gated time accumulation mode 11-15
General-purpose timer (GPT). *See* GPT 11-1
GPT
address map D-67
block diagram 11-2
capture/compare unit 11-10
block diagram 11-11
features 3-2
general
information 11-1
-purpose I/O 11-8
interrupt sources 11-6, D-69
interrupts 11-5
pins 11-7
polled and interrupt-driven operation 11-4
prescaler 11-8
pulse
accumulator 11-14
block diagram 11-15
-width modulation
unit (PWM) 11-16
block diagram 11-17
counter 11-18
reference manual 11-1
registers 11-2
capture/compare registers
action
data register (OC1D) 11-14
mask register (OC1M) 11-14
timer
compare force register (CFORC) 11-13, 11-14, D-74
interrupt flag register 2 (TFLG2) 11-10
input
capture 4/output compare 5 registers (TI4/O5) D-72
capture registers (TIC) D-71
interrupt
configuration register (ICR) D-68
control registers
timer interrupt mask registers (TMSK) 11-10, 11-12
module
configuration register (GPTMCR) D-67
test register (GPTMTR) D-68
OC1 action
data register (OC1D) D-69
mask register (OC1M) D-69
output compare registers (TOC) D-71
parallel I/O registers
port GP data
direction register (DDRGP) 11-8, 11-14, D-69
register (PORTGP) 11-8, D-69
prescaler register (PRESCL) D-77
pulse
accumulator registers

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

Freescale Semiconductor, Inc.

control register (PACTL) 11-8, 11-14, 11-16, D-70
counter register (PACNT) 11-14, D-70
-width modulation registers
counter register (PWMCNT) 11-18
PWM
buffer registers (PWMBUFA/PWMBUFB) D-76
control register C (PWMC) D-74
count register (PWMCNT) D-76
registers A/B (PWMA/PWMB) D-76
status registers
timer interrupt
flag register 1 (TFLG1) 11-12
timer
control registers (TCTL) D-72
counter register (TCNT) D-70
interrupt
flag registers (TFLG) D-74
mask registers (TMSK) D-72
single-step mode 11-4
special operation modes 11-3
status flags 11-5
test mode 11-4
timer counter register (TCNT) 11-2
GPTMCR D-67
GPTMTR D-68
Grounding 8-17

–H–

H 4-4, D-3
Half carry flag (H) 4-4, D-3
HALT 5-25, 5-33, 5-37, 5-43, 5-45
Halt
acknowledge flag (HALTA) D-51
monitor
enable (HME) 5-25, D-13
reset (HLT) D-8
operation 5-45
negating/reasserting 5-45
QSPI (HALT) D-51
HALT QSPI D-51
HALTA D-51
HALTA/MODF interrupt enable (HMIE) bit D-51
Handshaking 5-36
Hardware breakpoints 5-41
HCMOS 1-2
High-density complementary metal-oxide semiconductor (HCMOS) 1-2
HLT D-8
HME 5-25, D-13
HMIE D-51
Hysteresis 5-59, 11-8

–I–

I4/O5 11-14, D-71, D-73
I4/O5F D-74
IARB
GPT 11-6, D-68
MCCI 10-3, D-55
QSM 9-3, D-39
SIM 5-3, 5-59, D-7
IC4 11-14
ICD16/ICD32 C-2
ICF D-74
ICI D-73
ICR D-68
I_{DD} 5-54
IDLE 9-29, 10-21, D-43, D-63
Idle
frame 9-25, 10-17
-line
detect type (ILT) D-42, D-61
detected (IDLE) 9-29, 10-21, D-43, D-63
detection process 9-29, 10-21
interrupt enable (ILIE) 9-29, 10-22, D-42, D-61
type (ILT) 10-21
type (ILT) bit 9-29
I_{IN} 8-19
ILIE 9-29, 10-22, D-42, D-61
ILQSPI D-40
ILSCI 10-2, D-40, D-55
ILSCIA/B D-55
ILSPI 10-2, D-56, D-57
ILT 9-29, 10-21, D-42, D-61
IMB 11-1
IMM16 4-9
IMM8 4-9
Immediate addressing modes 4-9
Impedance 8-21
In-circuit debugger (ICD16/ICD32) C-2
INCP D-68
Increment prescaler (INCP) D-68
Indexed addressing modes 4-10
Inductors 8-14
Inherent addressing modes 4-10
Initialization programs E-1
INITRAM.ASM E-11
INITSCI.ASM E-12
INITSYS.ASM E-11
Input
capture
(IC) pins 11-7
/output compare (IC4/OC5) pin 11-7
4/output compare 5 11-14
(I4/O5) bit D-71
flag (I4/O5F) D-74
interrupt enable bit (I4/O5) D-73
conditioning signals 11-12
edge control bit field (EDGE) D-72
flags (ICF) D-74
functions 11-10
interrupt
enable (ICI) bit D-73
logic 11-12
timing example 11-13
leakage errors 8-23

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

Freescale Semiconductor, Inc.

Instruction
execution model 4-35
fetches 4-7
pipeline 4-35
set for CPU16 4-11
timing 4-36
Intermodule bus (IMB) 3-2, 11-1
Internal
bus
error ($\overline{\text{BERR}}$) 5-24, 5-25
monitor 5-24
register map 3-16
VCO frequency 5-8
Interrupt
acknowledge
and arbitration 5-59
bus cycles 5-61
arbitration 5-3, 9-3, 11-6
IARB field
GPT D-68
MCCI D-55
QSM D-39
SIM 5-59, D-7
exception processing 5-58
level (IL)
for QSPI (ILQSPI) D-40
for SCI (ILSCI) D-40
priority
adjust (IPA) D-68
and recognition 5-58
level field (IPL) 5-67, D-21
mask (IP) field 4-4, 5-58, 9-3, 11-6, D-3
processing summary 5-60
vector D-56
number 9-3, 11-6
field (INTV) D-40
Interrupts
GPT 11-5
MCCI 10-3
QSM 9-3
SIM 5-58
Inter-transfer delay 9-6
INTV D-40, D-56
I_{OUT} 8-19
IP 4-4, 9-3, D-3
IPA D-68
IPIPE0 4-43
IPIPE1 4-43
IPIPE1/0 5-53
IPL D-21
$\overline{\text{IRQ}}$ 5-58, 5-60
I_{SB} 6-3
IX 4-3
IY 4-3
IZ 4-3

–J–

Junction leakage 8-23

–L–

Leakage error 8-23
Length of delay after transfer (DTL) D-49
Level-sensitivity 5-58
LJSRR D-36
LJURR D-37
LOC D-8
LOCK 7-3, D-26
Lock registers (LOCK) D-26
Logic
analyzer pod connectors C-2
levels (definition) 2-6
Loop mode (LOOPS) D-41, D-61
LOOPQ D-50
LOOPS D-41, D-61
Loss of clock reset (LOC) D-8
Low-power
broadcast cycle 5-42
CPU space cycle 5-42
interrupt mask level 5-42
operation — SIM 5-29
stop mode enable (STOP)
ADC 8-3, D-30
GPT 11-3, D-68
MCCI 10-2, D-55
MRM 7-3, D-25
QSM 9-2, D-39
SRAM 6-2, D-23
LPSTOP 5-21, 5-29
LSB 2-6
LSBF 10-11
LSW 2-6

–M–

M 9-25, 10-18, D-42, D-61
M68HC11 instructions compared to CPU16 instructions 4-31
M68HC16Z1EVB evaluation board E-1
M68MEVB1632 modular evaluation board (MEVB) C-2
M68MMDS1632 modular development system (MMDS) C-1
MAC 4-5, 4-9, 4-45
Masked ROM module (MRM). *See* MRM 7-1
Master/slave mode select (MSTR) D-46
Maximum ratings (electricals) A-1
MCCI
address map D-54
address map information 10-2
block diagram 10-1
features 3-2
general 10-1
general-purpose I/O 10-4
initialization 10-23
interrupts 10-3
pin function D-58
reference manual 10-1
registers

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

Freescale Semiconductor, Inc.

data direction register (MDDR) 10-4
global registers
data direction register (DDRM) D-58
data direction register (MDDR) 10-2
interrupt vector register (MIVR) 10-2, D-56
module configuration register (MMCR) 10-2, D-54
pin control registers
pin assignment register (MPAR) 10-2, D-57
port
data register (PORTMC) 10-2, D-59
pin state register (PORTMCP) 10-2, D-59
SCI
interrupt level register (ILSCI) 10-2, D-55
SPI
interrupt level register (ILSPI) 10-2, D-56
test register (MTEST) 10-2, D-55
pin assignment register (MPAR) 10-4
SCI
control register 0 (SCCR0A/B) 10-13, D-59
control register 1 (SCCR1A/B) 10-16, D-60
data register (SCDRA/B) 10-16, D-63
status register (SCSRA/B) 10-16, D-62
SPI
control register (SPCR) 10-6, D-64
data register (SPDR) 10-6, D-66
status register (SPSR) 10-6, D-65
types 10-2
SCI 10-13
interrupt level (ILSCIA/B) D-55
SPI 10-4
MCU
132-pin assignment package
MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 3-7, B-2
MC68HC16Z4/CKZ4 3-9, B-3
144-pin assignment package
MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 3-8, B-5
MC68HC16Z4/CKZ4 3-10, B-6
address maps
MC68HC16Z1/CKZ1/CMZ1 3-17
MC68HC16Z2/Z3 3-18
MC68HC16Z4/CKZ4 3-18
basic system 5-30
block diagram
MC68HC16Z1/CKZ1/CMZ1 3-4
MC68HC16Z2/Z3 3-5
MC68HC16Z4/CKZ4 3-6
components 1-1
overview 1-1
personality board (MPB) C-2
pin characteristics 3-11
power connections 3-13
signal
characteristics 3-13
function 3-15
MDDR 10-2, 10-4
Mechanical data and ordering information B-1
Memory maps
combined program and data
MC68HC16Z1/CKZ1/CMZ1 3-20
MC68HC16Z2/Z3 3-21
MC68HC16Z4/CKZ4 3-22
internal register map 3-16
separate program and data
MC68HC16Z1/CKZ1/CMZ1 3-23
MC68HC16Z2/Z3 3-24
MC68HC16Z4/CKZ4 3-25
Microcontroller Development Tools Directory (MCUDE-VTLDIR/D Rev. 3) C-1
Microsequencer 4-35
Misaligned operand 5-35
MISO 9-16, 9-20
MIVR 10-2, D-56
MM 6-1, 7-1, D-7
MMCR 10-2, D-54
MMDS C-1
Mnemonics
range (definition) 2-6
specific (definition) 2-6
MODCLK 5-5, 5-6, 5-56
MODE 5-66, D-19
Mode
fault flag (MODF) 9-9, 10-12, D-51, D-66
select (M) D-42, D-61
MODF 9-9, D-51, D-66
Modular platform board C-2
Module
mapping (MM) bit 5-2, 6-1, 7-1, 11-2, D-1, D-7
pin functions 5-53
Modulus counter 9-26, 10-18
Monotonicity 8-1
MOSI 9-16, 9-20
MPAR 10-2, 10-4, D-57
MPB C-2
MRM 7-1
address map D-25
array address mapping 7-1
features 3-2
normal access 7-2
registers
module configuration register (MRMCR) 7-1, D-25
ROM
array base address registers (ROMBAH/BAL) 7-1, D-27
bootstrap words (ROMBS) 7-1, D-28
signature registers (RSIGHI/LO) 7-1, D-27
reset 7-3
ROM signature 7-3
MRMCR 7-1, D-25
MSB 2-6
MSTR 10-7, 10-8, D-46
MSTRST (master reset) 5-48, 5-55, 5-56, 5-57
MSW 2-6
MTEST 10-2, D-55
MULT D-32

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

Freescale Semiconductor, Inc.

Multichannel
communication interface module (MCCI). *See* MCCI 10-1
conversion (MULT) D-32
Multimaster operation 9-9
Multiple
-channel conversion D-35
exceptions 4-40
Multiplexer 8-4, 11-9
channels 8-4
outputs 11-10
Multiply and accumulate (MAC) 4-45
MV 4-4, D-3

–N–

N 4-4
Negated (definition) 2-6
Negative
flag (N) 4-4
integers 4-6
stress 8-18
New queue pointer value (NEWQP) D-50
NEWQP 9-8, 9-21, D-50
NF 9-28, 10-21, D-44, D-63
Nine-stage divider chain 11-9
Noise 8-14
error (NF) flag 9-28, 10-21, D-44, D-63
Non-maskable interrupt 5-58
NRZ 9-2, 10-2, 10-13

–O–

OC1D 11-14, D-70
OC1M 11-14, D-70
OC5 11-14
OCF D-74
OCI D-73
OCxF 11-13
OCxl 11-13
OM/OL D-72
OP (1 through 3) 5-34
Opcode tracking 4-40
breakpoints 4-42
combining with other capabilities 4-41
deterministic 4-40
Operand
alignment 5-35
byte order 5-34
misaligned 5-35
transfer cases 5-35
Operators 2-1
OR D-43, D-63
Ordering information B-8
ORG00000.ASM E-6
ORG00008.ASM E-6
Output
capture pins 11-7
compare
1 (single comparison operation) 11-14
flags (OCF) D-74
functions 11-13, 11-14
interrupt enable (OCI) bit D-73
mode bits/output compare level bits (OM/OL) D-72
status flag (OCxF) bit 11-13
Overflow flag (V) 4-4, D-3
Overrun error (OR) D-43, D-63
Overview information 3-1

–P–

PACLK D-71
PACNT 11-14, 11-16, D-70, D-71
PACTL 11-8, 11-14, 11-16, D-70
PAEN D-70
PAI 11-1, 11-15
PAI pin state (PAIS) D-70
PAIF 11-15, D-74
PAII D-73
PAIS D-70
PAMOD D-70
PAOVF 11-15, D-74
PAOVI D-73
Parallel I/O ports 5-70
Parasitic devices 8-18
Parentheses (definition) 2-6
Parity
checking 9-26, 10-19
enable (PE) D-42, D-61
error (PF) flag 9-28, 10-21, D-44, D-63
type (PT) 9-26, 10-19, D-42, D-61
PC 4-3
PCLK 11-1, 11-8
pin state (PCLKS) D-71
PCLKS D-71
PCS D-53
to SCK delay (DSCK) D-53
PCS0/$\overline{SS}$ 9-20
PE D-42, D-61
PEDGE 11-16, D-70
PEPAR 5-70, D-10
Periodic
interrupt
modulus counter 5-28
priority 5-29
request level (PIRQL) D-13
timer 5-27
components 5-27
modulus (PITM field) 5-28, D-14
PIT period calculation 5-28, D-14
vector (PIV) D-13
timer prescaler control (PTP) 5-28, D-14
Peripheral chip-selects (PCS) 9-21, D-53
PF 9-28, 10-21
PFPAR 5-70, D-11
Phase-locked loop (PLL) 1-1
PICR 5-60, D-13

Freescale Semiconductor, Inc.

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

Pin
characteristics 3-11
considerations 8-14
electrical state 5-53
function 5-53
reset states 5-54
Pipeline multiplexing 4-41
PIRQL D-13
PITM 5-28, D-14
PITR 5-28, D-14
PIV D-13
PK 4-4, 4-5, D-3
PLL 1-1, 5-6
Pointer 9-6
Polled operation 11-4
Port
C data register (PORTC) 5-67
E
data direction register (DDRE) 5-70
data register (PORTE) 5-71
pin assignment register (PEPAR) 5-70
F
data direction register (DDRF) 5-70
data register (PORTF) 5-71
pin assignment register (PFPAR) 5-70
parallel I/O in SIM 5-70
replacement unit (PRU) C-2
size 5-65
PORTADA 8-1, D-30
PORTC D-15
PORTE 5-71, D-9
PORTF 5-71
PORTF0/1 D-10
PORTGP 11-8, D-69
PORTMC 10-2, D-59
PORTMCP 10-2, D-59
PORTQS 9-4, D-44
Positive stress 8-18
Post-modified index addressing mode 4-10
POW D-8
Power
connections 3-13
-up reset (POW) D-8
PPR D-75
PPROUT D-75
PQSPAR 9-4, 9-16, 9-20, D-45
Prescaler 11-1, 11-8
block diagram 11-9
rate selection field (PRS) D-31
PRESCL D-77
Program
counter address extension field (PK) 4-4, D-3
flow changes 4-36
Programming examples E-12
CPU16 E-23
GPT E-25
QSM/SCI E-24
SIM E-13
PROUT 11-10
PRS D-31
PRU C-2
PSHM 4-9
PT 9-26, 10-19, D-42, D-61
PTP D-14
PULM 4-9
Pulse
accumulator 11-1
block diagram 11-15
clock
select (PACLK) D-71
select mux 11-10
counter (PACNT) D-71
edge control (PEDGE) D-70
enable (PAEN) D-70
flag (PAIF) 11-15, D-74
input
(PAI) pin 11-7, 11-8, 11-15
interrupt enable (PAII) bit D-73
mode (PAMOD) D-70
overflow
flag (PAOVF) 11-15, D-74
interrupt enable (PAOVI) bit D-73
-width modulation 11-1
pins (PWMA/PWMB) 11-8
unit (PWM) 11-16
block diagram 11-17
buffer register (PWMBUFA/B) 11-19
counter 11-18
duty cycle ratios 11-17
frequency ranges 11-18, D-76
function 11-18
PWM 11-16
clock output enable (PPROUT) D-75
prescaler/PCLK select (PPR) field D-75
slow/fast select
(SFA) bit D-75
(SFB) bit D-75
PWMA/B 11-8, D-76
PWMBUFA/B 11-19, D-76
PWMC 11-8, D-74
PWMCNT 11-18, D-76

–Q–

QILR 9-2, D-39
QIVR 9-2, D-39
QSM
address map 9-2, D-38
block diagram 9-1
features 3-2
general 9-1
interrupts 9-3
pin function 9-4, D-46
QSPI 9-5
operating modes 9-9
operation 9-8
pins 9-8
RAM 9-7
registers 9-6
reference manual 9-1

Freescale Semiconductor, Inc.

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

registers
command RAM (CR) D-52
global registers 9-2
interrupt
level register (QILR) 9-2, D-39
vector register (QIVR) 9-2, D-39
test register (QTEST) 9-2
module configuration register (QSMCR) D-38
pin control registers 9-4
port QS
data
direction register (DDRQS) D-45
register (PORTQS) D-44
data direction register (DDRQS) 9-4
data register (PORTQS) 9-4
pin assignment register (PQSPAR) D-45
QSPI
control register 0 (SPCR0) D-46
control register 1 (SPCR1) D-48
control register 2 (SPCR2) D-49
control register 3 (SPCR3) D-50
status register (SPSR) D-50
receive data RAM (RR) D-51
SCI
control register 0 (SCCR0) D-40
control register 1 (SCCR1) D-41
data register (SCDR) D-44
status register (SCSR) D-43
test register (QTEST) D-39
transmit data RAM (TR) D-52
types 9-2
SCI 9-21
operation 9-25
pins 9-25
registers 9-24
QSMCR D-38
QSPI 9-1, 9-5
block diagram 9-5
command RAM 9-8
enable (SPE) D-48
finished flag (SPIF) D-51
initialization operation 9-10
loop mode (LOOPQ) D-50
master operation flow 9-11
operating modes 9-9
master mode 9-9, 9-16
wraparound mode 9-19
slave mode 9-9, 9-20
wraparound mode 9-21
operation 9-8
peripheral chip-selects 9-21
pins 9-8
RAM 9-7
receive RAM 9-7
transmit RAM 9-7
registers 9-6
control registers 9-6
status register 9-7
timing A-46
— master, CPHA = 0/CPHA = 1 A-47
— slave, CPHA = 0/CPHA = 1 A-48
low voltage A-45
QTEST 9-2, D-39
Queue pointers
completed queue pointer (CPTQP) 9-8
end queue pointer (ENDQP) 9-8
new queue pointer (NEWQP) 9-8
Queued serial
module (QSM). *See* QSM 9-1
peripheral interface (QSPI) *See* QSPI. 9-1, 9-5

–R–

R/$\overline{W}$ 5-32
field 5-66, D-19
RAF D-43, D-63
RAM
array space (RASP) D-23
base address lock (RLCK) bit D-23
RAMBAH/BAL 6-1, D-24
RAMMCR 6-1, D-23
RAMTST 6-1, D-24
RASP 6-2, D-23
encoding 6-2, D-23
RC
DAC array 8-5
low pass filter 8-16
RDR 9-24
RDRF 9-28, 10-21, D-43, D-63
RE 9-28, 10-4, 10-13, 10-20, D-42, D-62
Read
/write signal (R/$\overline{W}$) 5-32
cycle 5-37
timing diagram A-29
Receive
data
(RXD) pin — QSM 9-25
(RXDA/B) pins (MCCI) 10-17
register full (RDRF) D-43, D-63
RAM 9-7
time sample clock (RT) 9-26, 9-28, 10-18, 10-21
Receiver
active (RAF) D-43, D-63
data register (RDRF) flag 9-28, 10-21
enable (RE) 9-28, 10-4, 10-13, 10-20, D-42, D-62
interrupt enable (RIE) D-42, D-61
wakeup (RWU) 9-30, 10-22, D-42, D-62
Register bit and field mnemonics 2-3
Relative addressing modes 4-10
$\overline{RES}$10 8-7, D-31
$\overline{RESET}$ 5-48, 5-50, 5-54, 5-55
Reset
and mode select timing A-36
exception processing 5-48
module pin function out of reset 5-52
operation in SIM 5-48
control logic 5-48
mode selection 5-49
power-on 5-55

Freescale Semiconductor, Inc.

processing summary 5-57
states of pins assigned to other MCU modules 5-54
status register (RSR) 5-24, 5-57
timing 5-55
Resistor-divider chain 8-5
Resolution 8-7
Result registers 8-13
Return-from-interrupt instruction (RTI) 4-40
R_F 8-22
RF energy 8-14
RIE D-42, D-61
RJURR D-36
RLCK 6-2, D-23
RMAC 4-9
ROM array space (ASPC) D-26
ROMBAH/BAL 7-1, D-27
ROMBS 7-1
ROMBS0-3 D-28
RR D-51
RS-232C terminal C-2
RSIGHI/LO 7-1, 7-3, D-27
RSR 5-24, D-8
RT 9-28, 10-21
RTI 4-40
RWU 9-30, 10-22, D-42, D-62
RXD (QSM) 9-25
RXDA/B (MCCI) 10-16, 10-17

–S–

S 4-4, D-3
S8CM D-32
Sample
capacitor 8-5
time 8-7
time selection (STS) field D-31
SAR 8-13
Saturate mode (SM) bit 4-4, D-3
SBK 9-27, 10-20, D-42, D-62
SCAN D-32
Scan mode selection (SCAN) D-32
SCBR D-40, D-59
SCCR 9-24
SCCR0 D-40
SCCR0A/B 10-13, D-59
SCCR1 D-41
SCCR1A/B 10-16, D-60
SCDR 9-24, D-44
SCDRA/B 10-16, D-63
SCF D-36
SCI 9-1, 9-2, 9-16, 9-21, 10-1
baud
clock 9-26, 10-18
rate 10-18, D-40, D-59
idle-line detection 9-29, 10-21
internal loop 9-30, 10-22
interrupt level (ILSCIA/B) D-55
operation 9-25, 10-17
parity checking 9-26, 10-19
pins (MCCI) 10-16
pins (QSM) 9-25
receiver
block diagram
MCCI 10-15
QSM 9-23
operation 9-28, 10-20
wakeup 9-29, 10-22
registers 9-24
control register 0 — MCCI (SCCR0A/B) 10-13, D-59
control register 1 — MCCI (SCCR1A/B) 10-16, D-60
control registers — QSM (SCCR) 9-24
data register
MCCI (SCDRA/B) 10-16, D-63
QSM (SCDR) 9-24
status register
MCCI (SCSRA/B) 10-16, D-62
QSM (SCSR) 9-24
serial formats 10-18
transmitter
block diagram
MCCI 10-14
QSM 9-22
operation 9-27
SCIA/B 10-2
SCK 9-16, 9-20
actual delay before SCK (equation) 9-17
baud rate (equation) 9-17
SCSR 9-24, D-43
SCSRA/B 10-16, D-62
Select eight-conversion sequence mode (S8CM) D-32
Send break (SBK) 9-27, 10-20, D-42, D-62
Separate program and data space map
MC68HC16Z1/CKZ1/CMZ1 3-23
MC68HC16Z2/Z3 3-24
MC68HC16Z4/CKZ4 3-25
Sequence complete flag (SCF) D-36
Serial
clock
baud rate (SPBR) D-48
frequency range 4-44
communication interface (SCI) 9-1, 9-21, 10-1, 10-13
data word 4-44
formats 9-25, 10-18
interface clock signal (DSCLK) 4-44
mode (M) bit 9-25, 10-18
peripheral interface (SPI) 10-1, 10-4
shifter 9-24, 9-27, 10-19
Set (definition) 2-6
Settling time 8-22
SFA 11-18, D-75
SFB 11-18, D-75
SHEN 5-47, D-6
Show cycle
enable (SHEN) 5-3, 5-47, D-6
operation 5-47
timing diagram A-35
Signal characteristics 3-13
Signature registers (RSIGHI/LO) 7-1

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

Freescale Semiconductor, Inc.

Signed fractions 4-6
SIM 5-1
address map D-4
block diagram 5-2
bus operation 5-36
chip-selects 5-61
external bus interface (EBI) 5-29
features 3-1
functional blocks 5-1
halt monitor 5-25
interrupt arbitration 5-3
interrupts 5-58
parallel I/O ports 5-70
periodic interrupt timer 5-27
reference manual 5-67
register access 5-3
registers
chip-select
base address
register boot (CSBARBT) D-17
registers (CSBAR) 5-64, 5-65, D-17
option
register boot (CSORBT) D-18
registers (CSOR) 5-64, 5-66, D-18
pin assignment registers (CSPAR) 5-64, D-15
clock synthesizer control register (SYNCR) D-7
master shift registers A/B (TSTMSRA/TSTMSRB) D-22
module configuration register (SIMCR) 5-2, D-6
periodic interrupt
control register (PICR) D-13
timer register (PITR) 5-28, D-14
port C data register (PORTC) 5-67, D-15
port E
data direction register (DDRE) 5-70, D-9
data register (PORTE) 5-71, D-9
pin assignment register (PEPAR) 5-70, D-10
port F
data direction register (DDRF) 5-70, D-11
data register (PORTF) 5-71
data registers (PORTF) D-10
pin assignment register (PFPAR) 5-70, D-11
reset status register (RSR) D-8
software service register (SWSR) D-15
system
protection control register (SYPCR) D-12
test register
(SIMTR) D-7
E (SIMTRE) D-9
test module
control register (CREG) D-22
distributed register (DREG) D-22
repetition count register (TSTRC) D-22
shift count register (TSTSC) D-22
reset 5-48
state of pins 5-54
software watchdog 5-25
block diagram (with PIT) 5-25
spurious interrupt monitor 5-25
system
clock
block diagram 5-4
protection 5-24
system clock 5-4
synthesizer operation 5-6
SIMCR 5-2, 8-3, D-6
SIMTR D-7
SIMTRE D-9
Single
-channel conversions D-34
-step mode 11-4
SIZ 5-54
SIZE (MCCI) 10-11
Size signals (SIZ) 5-32, 5-35, 5-47
SK 4-3, 4-5
Slave select signal ($\overline{SS}$). *See* $\overline{SS}$ 9-20
SLOCK D-8
Slow reference circuit 5-5
SM 4-4, D-3
Software watchdog 5-25
block diagram 5-27
clock rate 5-26
enable (SWE) 5-25, D-12
prescale (SWP) 5-26, D-12
ratio of SWP and SWT bits 5-26
reset (SW) D-8
timeout period calculation 5-26, D-12
timing field (SWT) 5-26, D-12
Source voltage level (V_{SRC}) 8-22
SP 4-3
SPACE (address space select) 5-67, D-21
SPBR D-48
SPCR 10-6, D-64
SPCR0 D-46
SPCR1 D-48
SPCR2 D-49
SPCR3 D-50
SPDR 10-6, D-66
SPE 9-6, D-48
SPI 10-1
block diagram 10-5
clock phase/polarity controls 10-8
finished flag (SPIF) D-65
interrupt level (ILSPI) D-57
mode fault 10-12
operating modes
master mode 10-7
slave mode 10-8
pins 10-6
registers
control register (SPCR) 10-6, D-64
data register (SPDR) 10-6, D-66
status register (SPSR) 10-6, D-65
serial clock baud rate 10-11
timing A-50
— master, CPHA = 0/CPHA = 1 A-51
— slave, CPHA = 0/CPHA = 1 A-52

Freescale Semiconductor, Inc.

low voltage A-49
transfer
data flow 10-5
size and direction 10-11
write collision 10-12
SPI finished interrupt enable (SPIFIE) D-50
SPIF D-51, D-65
SPIFIE D-50
SPSR 10-6, D-50, D-65
SRAM
address map D-23
array address mapping 6-2
features 3-1
normal access 6-2
registers
array base address register
high (RAMBAH) 6-1
low (RAMBAL) 6-1
array base address registers
high/low (RAMBAH/BAL) D-24
module configuration register (RAMMCR) 6-1, D-23
test register (RAMTST) 6-1, D-24
reset 6-3
standby and low-power stop operation 6-2
SS 9-20, 10-8, 10-9, 10-10, 10-12
Standard non-return to zero (NRZ) 9-2, 10-2, 10-13
Star-point ground system 8-17
Start bit (beginning of data frame) 9-25, 10-17
State machine 9-28, 10-20
STEXT 5-21, D-8
STOP 6-2, 7-3, 8-3, 9-2, 10-2, 11-3, D-23, D-25, D-30, D-39, D-55, D-68
enable (S) 4-4, D-3
Stop
mode
external clock (STEXT) 5-21, D-8
SIM clock (STSIM) 5-21, D-8
prescaler (STOPP) D-68
SCI end of data frame bit 9-25, 10-17
STOPP 11-4, D-68
STRB (address strobe/data strobe) bit 5-40, 5-66, D-19
Stress conditions 8-18
STS D-31
STSIM 5-21, D-8
Successive approximation register (SAR) 8-13
Supervisor
/unrestricted data space (SUPV)
ADC D-30
GPT D-68
MCCI 10-3, D-55
QSM D-39
SIM 5-3, D-6
SUPV 10-3, D-6, D-30, D-39, D-55, D-68
SW D-8
SWE 5-25, D-12
SWP 5-26, D-12
SWSR D-15
SWT 5-26, D-12
Symbols 2-1
Synchronous exceptions 4-39
SYNCR 5-5, D-7
Synthesizer lock flag (SLOCK) D-8
SYPCR D-12
SYS D-8
System
clock 5-4
output (CLKOUT) 5-36
sources 5-5
frequencies
16.78 MHz 5-15
20.97 MHz 5-17
25.17 MHz 5-19
integration module. *See* SIM 5-1
reset (SYS) D-8
test register
(SIMTR) D-7
E (SIMTRE) D-9

–T–

TC 9-27, 10-19, D-43, D-62
TCIE 9-28, 10-20, D-42, D-61
TCNT 11-1, 11-10, D-70
TCTL D-72
TDR 9-24
TDRE 9-27, 10-19, D-43, D-62
TE 10-4, 10-13, D-42, D-62
Test submodule reset (TST) D-8
TFLG D-74
TFLG1 11-12
TFLG2 11-10
Thermal characteristics A-5
Three-state control (TSC) 5-56
TI4/O5 D-72
TIC D-71
TIE 9-28, 10-20, D-42, D-61
Timer
counter (TCNT) 11-10
overflow
flag (TOF) 11-10, D-74
interrupt enable (TOI) bit D-73
prescaler/PCLK select (CPR) field D-73
TMSK D-72
TMSK1 11-12
TMSK2 11-10
TOC D-71
TOF D-74
TOI 11-10, D-73
TR D-52
Transfer length options 9-17
Transistion-sensitivity 5-58
Transmit
complete
(TC) flag
MCCI 10-19, D-62
QSM 9-27, D-43
interrupt enable (TCIE)
MCCI 10-20, D-61
QSM 9-28, D-42

Freescale Semiconductor, Inc.

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

data
(TXD) pin — QSM 9-25
(TXDA/B) pins — MCCI 10-17
register empty (TDRE) flag
MCCI 10-19, D-62
QSM 9-27, D-43
enable (TE)
MCCI 10-4, 10-13, 10-19, D-62
QSM 9-27, D-42
interrupt enable (TIE)
MCCI 10-20, D-61
QSM 9-28, D-42
RAM 9-7
TSC 5-54, 5-56
TST D-8
TSTMSRA/B D-22
TSTRC D-22
TSTSC D-22
Two/three wire transfers 10-4
Two's complement 4-6
TXD (QSM) 9-25
TXDA/B (MCCI) 10-16, 10-17
Typical ratings
2.7V to 3.6V, 16.78 MHz A-2
20.97 MHz A-3
25.17 MHz A-4
5V, 16.78 MHz A-3

–U–

UART 10-2, 10-13
Universal asynchronous receiver/transmitter (UART) 10-2, 10-13

–V–

V 4-4, D-3
V_{CF} 8-22
VCO 5-6
V_{DD} 5-55, 6-1
ramp time 5-56
V_{DDA} 8-1, 8-3, 8-14
V_{DDSYN} 5-6, 5-55
Vector sources D-56
V_I 8-22
V_{IH} 8-2
V_{IL} 8-2
Voltage
controlled oscillator (VCO) 5-6
frequency (f_{VCO}) 5-6
frequency ramp time 5-56
limiting diodes 8-19
V_{PP} C-2
V_{RH} 5-53, 8-1, 8-3, 8-14, 8-23
V_{RL} 5-53, 8-1, 8-3, 8-14, 8-23
V_{SRC} 8-22
V_{SSA} 8-1, 8-3, 8-14
V_{STBY} 6-2

–W–

W D-8
WAIT 7-3, D-26
Wait states field (WAIT) D-26
WAKE 9-30, 10-22, D-42, D-61
Wakeup
address mark (WAKE) 9-30, 10-22, D-42, D-61
functions 9-2
WCOL D-65
Wired-OR
mode
for QSPI pins (WOMQ) D-47
for SCI pins (WOMS)
MCCI 10-19, D-61
QSM 9-27, D-41
open-drain outputs 10-11
WOMC 10-16
WOMP 10-11
WOMQ D-47
WOMS 9-27, 10-19, D-41, D-61
Word composition 4-7
Wrap
enable (WREN) D-50
to (WRTO) D-50
Wraparound mode 9-6
master 9-19
slave 9-21
WREN D-50
Write
collision (WCOL) 10-12, D-65
cycle 5-38
flowchart 5-39
timing diagram A-30
WRTO D-50

–X–

X D-8
bit in SYNCR 5-6
XK 4-3, 4-5
XTAL 5-5
XTRST (external reset) 5-48

–Y–

Y D-8
YK 4-3, 4-5

–Z–

Z 4-4
Zero flag (Z) 4-4
ZK 4-3, 4-5

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View MC68HC16Z1MEH16 on WIN SOURCE
- NXP / Nexperia Information

Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management