THE DATASHEET OF

MC9S12T64MPKE16

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

DOCUMENT NUMBER
9S12T64AF16V1/D

MC9S12T64

Specification

February 28th, 2003 – Revision 1.1.1

Freescale Semiconductor, Inc.

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

Revision History

Rev.	Contents	Date	Who
1.0	- Initial Release.	Jan-28-2003	
1.1	- Added document number information. - Modified a diagram of FBDM data transfer in SPI mode in FBDM section (page 543).	Feb-13-2003	
1.1.1	- Corrected order number in ordering information (page 24).	Feb-28-2003	

Freescale Semiconductor, Inc.

List of Sections

List of Sections 3
Table of Contents 5
List of Figures 11
List of Tables 15
General Description 19
Central Processing Unit (CPU) 25
Pinout and Signal Description 59
System Configuration 79
Registers 85
Operating Modes 105
Module Mapping Control (MMC) 121
Multiplexed External Bus Interface (MEBI) 141
Resets and Interrupts 169
Voltage Regulator (VREG) 181
Low-Voltage Detector (LVD) 185
Flash EEPROM 64K 193
CALRAM 2K 235
Port Integration Module (PIM) 249
Clocks and Reset Generator (CRG) 271
Pulse Width Modulator (PWM8B8C) 327
Enhanced Capture Timer (ECT) 371
Serial Communications Interface (SCI) 419
Serial Peripheral Interface (SPI) 457
Analog to Digital Converter (ATD) 487
Fast Background Debug Module (FBDM) 517
Breakpoint (BKP) 547
Electrical Characteristics 561
. 591
Glossary 595
Literature Updates 605

Freescale Semiconductor, Inc.

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

List of Sections

Freescale Semiconductor, Inc.

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

Table of Contents

List of Sections

Table of Contents

List of Figures

List of Tables

General Description
Contents 19
Introduction 19
Features 20
MC9S12T64 Block Diagram 23
Ordering Information 24

Central Processing Unit (CPU)
Contents 25
Introduction 25
Programming Model 26
Data Format Summary 34
Addressing Modes 35
Instruction Set Overview 36

Pinout and Signal Description
Contents 59
MC9S12T64 Pin Assignments in 80-pin LQFP 59
Power Supply Pins 62
Signal Descriptions 65
Port Signals 74

System Configuration
Contents 79
Introduction 79
Modules Variabilities 79
MCU Variabilities 80
System Clock Description 81

Registers
Contents 85
Register Block 85
General Purpose Registers 103

Operating Modes
Contents 105
Introduction 105
Operating Modes 105
Background Debug Mode 116

Freescale Semiconductor, Inc.

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

Table of Contents

	Secured Mode of Operation	117
Module Mapping Control (MMC)	Contents	121
	Overview	121
	Features	121
	Block Diagram	122
	Register Map	123
	Register Descriptions	125
	Functional Description	133
	Memory Maps	138
Multiplexed External Bus Interface (MEBI)	Contents	141
	Overview	141
	Modes of Operation	141
	External Pin Descriptions	142
	Register Map	145
	Register Descriptions	146
	Functional Description	161
	Low-Power Options	167
Resets and Interrupts	Contents	169
	Introduction	169
	Register Map	170
	Exception Priority	171
	Maskable interrupts	171
	Latching of Interrupts	172
	Register Descriptions	175
	Resets	177
	Effects of Reset	178
Voltage Regulator (VREG)	Contents	181
	Overview	181
	Features	181
	Modes of Operation	181
	Block Diagram	183
	Functional Description	184
	Reset Initialization	184
Low-Voltage Detector (LVD)	Contents	185
	Glossary	185
	Overview	185
	Features	186
	Modes of Operation	186
	Block Diagram	187
	Register Map	188
	Register Descriptions	189
	Functional Description	191
	Interrupts	192

Freescale Semiconductor, Inc.

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

Flash EEPROM 64K

Contents 193
Overview 193
Glossary 194
Features 195
Modes of Operation 195
Block Diagram 197
External Pin Descriptions 198
Module Memory Map 198
Register Descriptions 206
Functional Description 220
Low Power Options 229
Background Debug Mode 230
Flash Security 231
Reset Initialization 231
Interrupts 232

CALRAM 2K

Contents 235
Glossary 235
Overview 236
Features 236
Modes of Operations 237
Block Diagram 238
External Pin Descriptions 239
Module Memory Map 239
Register Descriptions 241
Functional Description 242
Reset Initialization 248

Port Integration Module (PIM)

Contents 249
Overview 249
Block Diagram 251
External Pin Descriptions 252
Register Map 253
Register Descriptions 255
Functional Description 267
Low Power Options 269
Reset Initialization 269

Clocks and Reset Generator (CRG)

Contents 271
Overview 271
Features 272
Modes of Operation 273
Block Diagram 275
External Pin Descriptions 276
Register Map 279
Register Descriptions 280
Functional Description 294
Operation Modes 305

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

Freescale Semiconductor, Inc.

Table of Contents

	Low Power Options	305
	Reset Description	318
	Interrupts	324
Pulse Width Modulator (PWM8B8C)	Contents	327
	Overview	327
	Features	328
	Modes of Operation	328
	Block Diagram	330
	External Pin Descriptions	331
	Register Map	332
	Register Descriptions	334
	Functional Description	353
	Low Power Options	368
	Reset Initialization	368
	Interrupts	369
Enhanced Capture Timer (ECT)	Contents	371
	Overview	371
	Features	372
	Modes of Operation	372
	Abbreviations	372
	Block Diagram	373
	External Pin Descriptions	374
	Register Map	375
	Register Descriptions	378
	Functional Description	408
	Interrupts	416
Serial Communications Interface (SCI)	Contents	419
	Overview	419
	Features	419
	Modes of Operation	420
	Block Diagram	421
	External Pin Descriptions	422
	Register Map	423
	Register Descriptions	424
	Functional Description	434
	Interrupts	454
Serial Peripheral Interface (SPI)	Contents	457
	Overview	457
	Features	458
	Modes of Operation	458
	Block Diagram	459
	External Pin Descriptions	460
	Register Map	461
	Register Descriptions	462

Freescale Semiconductor, Inc.

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

Functional Description 470
Low Power Mode Options 483
Reset Initialization 484
Interrupts 484

Analog to Digital Converter (ATD)

Contents 487
Overview 487
Features 487
Modes of Operation 488
Block Diagram 490
External Pin Descriptions 491
Register Map 492
Register Descriptions 494
Functional Description 511
Low Power Modes 514
Reset Initialization 515
Interrupts 515

Fast Background Debug Module (FBDM)

Contents 517
Overview 517
Features 517
Modes of Operation 519
Block Diagram 520
External Pin Descriptions 521
Register Map 523
Register Descriptions 524
Functional Description 528
Low-Power Options 546
Interrupts 546

Breakpoint (BKP)

Contents 547
Overview 547
Features 548
Modes of Operation 549
Block Diagram 550
External Pin Descriptions 552
Register Map 552
Register Descriptions 553
Breakpoint Priority 560
Reset Initialization 560
Interrupts 560

Electrical Characteristics

Contents 561
General 561
ATD Characteristics 569
Flash EEPROM Characteristics 573
Voltage Regulator Characteristics 577
Reset, Oscillator and PLL Characteristics 578

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

Freescale Semiconductor, Inc.

Table of Contents

SPI Timing 582
External Bus Timing 587

Glossary

Literature Updates

Literature Distribution Centers 605
Customer Focus Center 606
Mfax 606
Motorola SPS World Marketing World Wide Web Server 606
Microcontroller Division's Web Site 606

Freescale Semiconductor, Inc.

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

List of Figures

Figure 1 MC9S12T64 Block Diagram . . . 23
Figure 2 Programming Model . . . 26
Figure 3 Accumulator A . . . 27
Figure 4 Accumulator B . . . 27
Figure 5 Index Register X . . . 28
Figure 6 Index Register Y . . . 28
Figure 7 Stack Pointer (SP) . . . 29
Figure 8 Program Counter (PC) . . . 29
Figure 9 Condition Code Register (CCR) . . . 30
Figure 10 Programming Model . . . 33
Figure 11 Pin Assignments in 80-pin LQFP for MC9S12T64 . . . 60
Figure 12 80-pin LQFP Mechanical Dimensions . . . 61
Figure 13 Clock Connections . . . 82
Figure 14 Module Mapping Control Block Diagram . . . 122
Figure 15 Module Mapping Control Register Summary . . . 123
Figure 16 MC9S12T64 Memory map after reset . . . 139
Figure 17 MEBI Register Map . . . 145
Figure 18 Queue Status Signal Timing . . . 164
Figure 19 Resets and Interrupts Register Map . . . 170
Figure 20 VREG Block Diagram . . . 183
Figure 21 LVD Block Diagram . . . 187
Figure 22 Flash 64K Block Diagram . . . 197
Figure 23 Flash Data Memory Map in Normal Modes . . . 199
Figure 24 Flash Data Memory Map in Special Modes . . . 201
Figure 25 Flash Control Register Map . . . 204
Figure 26 FADDR Address Mapping to Flash Relative Address . . . 218
Figure 27 PRDIV8 and FDIV bits Determination Procedure . . . 222
Figure 28 Example Program Algorithm . . . 225
Figure 29 Example Program Algorithm in Flash Super User Mode . . . 227
Figure 30 Flash Interrupt Implementation . . . 232
Figure 31 CALRAM 2K Byte Block Diagram . . . 238
Figure 32 CALRAM Data Memory Map . . . 239
Figure 33 Copy Data from Flash EEPROM to CALRAM . . . 244
Figure 34 Map CALRAM over Flash EEPROM . . . 245
Figure 35 An Example Memory Map When Erase/Program Flash EEPROM . . . 246
Figure 36 Remove CALRAM from memory mapping . . . 247

Freescale Semiconductor, Inc.

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

List of Figures

Figure 37 PIM Block Diagram . . . 251
Figure 38 PIM9T64 Register Map . . . 253
Figure 39 Illustration of I/O pin functionality . . . 268
Figure 40 Block diagram of CRG . . . 275
Figure 41 PLL Loop Filter Connections . . . 276
Figure 42 Colpitts Crystal Connections (XCLKS=1) . . . 277
Figure 43 Pierce Oscillator Connections (XCLKS=0) . . . 277
Figure 44 External Clock Connections (XCLKS=0) . . . 278
Figure 45 CRG Register Map . . . 279
Figure 46 PLL Functional Diagram . . . 295
Figure 47 Clock Generator . . . 299
Figure 48 Core Clock and Bus Clock relationship . . . 300
Figure 49 Check Window definition . . . 301
Figure 50 Sequence for Clock Quality Check . . . 302
Figure 51 Clock Chain for COP . . . 303
Figure 52 Clock Chain for RTI . . . 304
Figure 53 Wait Mode Entry/Exit Sequence . . . 307
Figure 54 Stop Mode Entry/Exit Sequence . . . 312
Figure 55 RESET Timing . . . 320
Figure 56 RESET pin tied to VDD (by a pull-up resistor) . . . 322
Figure 57 RESET pin held low externally . . . 322
Figure 58 PWM8B8C Block Diagram . . . 330
Figure 59 PWM Register Map . . . 332
Figure 60 PWM Clock Select Block Diagram . . . 354
Figure 61 PWM Timer Channel Block Diagram . . . 357
Figure 62 PWM Left Aligned Output Waveform . . . 361
Figure 63 PWM Left Aligned Output Example Waveform . . . 362
Figure 64 PWM Center Aligned Output Waveform . . . 363
Figure 65 PWM Center Aligned Output Example Waveform . . . 364
Figure 66 PWM 16-Bit Mode . . . 365
Figure 67 Timer Block Diagram . . . 373
Figure 68 Enhanced Capture Timer Register Map . . . 375
Figure 69 Timer Block Diagram in Latch Mode . . . 409
Figure 70 Timer Block Diagram in Queue Mode . . . 410
Figure 71 8-Bit Pulse Accumulators Block Diagram . . . 411
Figure 72 16-Bit Pulse Accumulators Block Diagram . . . 412
Figure 73 Interrupt Flag Setting . . . 413
Figure 74 SCI Block Diagram . . . 422
Figure 75 SCI Register Quick Reference . . . 423
Figure 76 Detailed SCI Block Diagram . . . 435
Figure 77 SCI Data Formats . . . 435
Figure 78 Transmitter Block Diagram . . . 438
Figure 79 SCI Receiver Block Diagram . . . 442

Freescale Semiconductor, Inc.

Figure 80 Receiver Data Sampling 444
Figure 81 Start Bit Search Example 1 446
Figure 82 Start Bit Search Example 2 446
Figure 83 Start Bit Search Example 3 447
Figure 84 Start Bit Search Example 4 447
Figure 85 Start Bit Search Example 5 448
Figure 86 Start Bit Search Example 6 448
Figure 87 Slow Data 449
Figure 88 Fast Data 450
Figure 89 Single-Wire Operation (LOOPS = 1, RSRC = 1) 453
Figure 90 Loop Operation (LOOPS = 1, RSRC = 0) 453
Figure 91 SPI Block Diagram. 459
Figure 92 SPI Register Summary 461
Figure 93 Master/Slave Transfer Block Diagram 474
Figure 94 SPI Clock Format 0 (CPHA = 0) 477
Figure 95 SPI Clock Format 1 (CPHA = 1) 479
Figure 96 Bus Clock Divisor Equation 480
Figure 97 ATD Module Block Diagram 490
Figure 98 ATD Register Map 492
Figure 99 Block Diagram of BDM in Single Wire Mode 520
Figure 100 Block Diagram of BDM in SPI Mode 520
Figure 101 BDM Register Map Summary 523
Figure 102 BDM Command Structure - Single Wire Mode 537
Figure 103 BDM Command Structure - SPI Mode. 538
Figure 104 BDM Host-to-Target Serial Bit Timing 541
Figure 105 BDM Target-to-Host Serial Bit Timing (Logic 1) 542
Figure 106 BDM Target-to-Host Serial Bit Timing (Logic 0) 543
Figure 107 8-Bit Data Transfer in SPI Mode 543
Figure 108 Breakpoint Block Diagram 551
Figure 109 Breakpoint Register Map 552
Figure 110 ATD Accuracy Definitions 572
Figure 111 PLL Loop Filter Connections 581
Figure 112 SPI Master Timing (CPHA = 0). 583
Figure 113 SPI Master Timing (CPHA =1) 583
Figure 114 SPI Slave Timing (CPHA = 0). 585
Figure 115 SPI Slave Timing (CPHA =1) 585
Figure 116 General External Bus Timing. 588

Freescale Semiconductor, Inc.

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

List of Figures

Freescale Semiconductor, Inc.

List of Tables

Table 1 MC9S12T64 Device Ordering Information 24
Table 2 MC9S12T64 Development Tools Ordering Information 24
Table 3 Addressing Mode Summary 35
Table 4 Instruction Set Summary 36
Table 5 Register and Memory Notation 51
Table 6 Source Form Notation 53
Table 7 Operation Notation 54
Table 8 Address Mode Notation 54
Table 9 Machine Code Notation 55
Table 10 Access Detail Notation 55
Table 11 Condition Code State Notation 58
Table 12 MC9S12T64 Power and Ground Connection Summary 64
Table 13 MC9S12T64 Signal Description Summary 70
Table 14 MC9S12T64 Port A, B, E, K, T, S, P Description Summary 77
Table 15 Port A, B, E, K, AD Pull-Up, Pull-Down and Reduced Drive Summary 78
Table 16 MMC Module Variable I/O Signals 79
Table 17 Assigned Part ID Numbers 80
Table 18 Module Availability in WAIT and RUN Modes 83
Table 19 MC9S12T64 Register Map 86
Table 20 Mode Selection 106
Table 21 MODC, MODB, MODA Write Capability 115
Table 22 : Security Bits 118
Table 23 EXSTR Stretch Bit Definition 128
Table 24 State of ROMON bit after reset 129
Table 25 Program space page index in special modes 133
Table 26 Mapping Precedence 134
Table 27 Access Type in Expanded Modes 135
Table 28 64K Byte Physical Flash/ROM Allocated 135
Table 29 External System Pins Associated With MEBI 142
Table 30 Access Type vs. Bus Control Pins 162
Table 31 IPIPE[1:0] Decoding when E Clock is High 165
Table 32 IPIPE[1:0] Decoding when E Clock is Low 165
Table 33 Interrupt Vector Table 172
Table 34 LVDF Flag Indication 190
Table 35 Flash Memory Mapping in Normal Modes 198
Table 36 Flash Memory Mapping in Special Modes 200
Table 37 Flash Protection/Security Field 202
Table 38 Memory Map Summary In Normal Modes 203
Table 39 Example FCLKDIV settings 207

Freescale Semiconductor, Inc.

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

List of Tables

Table 40 Security States 208
Table 41 Register Bank Selects 210
Table 42 Loading of the Protection Register from Flash 211
Table 43 Higher Address Range Protection 212
Table 44 Lower Address Range Protection 213
Table 45 Valid User Mode Commands 216
Table 46 Flash Interrupt Sources 232
Table 47 Example CALRAM Mapping 242
Table 48 Port Reset State and Priority Summary 252
Table 49 Pin Configuration Summary 255
Table 50 Clock Selection Based on XCLKS at reset 278
Table 51 RTI Frequency Divide Rates 289
Table 52 COP Watchdog Rates 292
Table 53 MCU configuration during Wait Mode 306
Table 54 Outcome of Clock Loss in Wait Mode 309
Table 55 Outcome of Clock Loss in Pseudo-Stop Mode 314
Table 56 Reset Summary 318
Table 57 Reset Vector Selection 319
Table 58 Relation between PORLVDRF and LVDF 323
Table 59 CRG Interrupt Vectors 324
Table 60 Clock B Prescaler Selects 341
Table 61 Clock A Prescaler Selects 341
Table 62 PWM Timer Counter Conditions 360
Table 63 16-bit Concatenation Mode Summary 366
Table 64 PWM Boundary Cases 367
Table 65 Compare Result Output Action 385
Table 66 Edge Detector Circuit Configuration 386
Table 67 Prescaler Selection 388
Table 68 Pin Action 392
Table 69 Clock Selection 392
Table 70 Modulus Counter Prescaler Select 397
Table 71 Delay Counter Select 400
Table 72 ECT Interrupts 416
Table 73 Loop Functions 426
Table 74 Example of 8-bit Data Formats 436
Table 75 Example of 9-Bit Data Formats 436
Table 76 Baud Rates (Example: Bus Clock = 16.0 MHz) 437
Table 77 Start Bit Verification 444
Table 78 Data Bit Recovery 445
Table 79 Stop Bit Recovery 445
Table 80 SCI Interrupt Sources 454
Table 81 SS Input / Output Selection 463
Table 82 Bidirectional Pin Configurations 465
Table 83 SPI Baud Rate Selection (16 MHz Bus Clock) 466
Table 84 Normal Mode and Bidirectional Mode 481
Table 85 SPI Interrupt Signals 485
Table 86 External Trigger Configurations 495

Freescale Semiconductor, Inc.

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

Table 87 Conversion Sequence Length Coding. 496
Table 88 ATD Behavior in Freeze Mode (Breakpoint) 498
Table 89 Sample Time Select 499
Table 90 Clock Prescaler Values 501
Table 91 Result Data Formats Available. 503
Table 92 Left Justified, Signed and Unsigned ATD Output Codes. 503
Table 93 Analog Input Channel Select Coding 504
Table 94 Special Channel Select Coding 507
Table 95 External Trigger Control Bits 513
Table 96 ATD module Interrupt Vectors 515
Table 97 Target Clock Selection Summary 526
Table 98 Hardware Commands 532
Table 99 Firmware Commands 534
Table 100 SPI Mode Timing 544
Table 101 Tag Pin Function 545
Table 102 Breakpoint Mask Bits for First Address 554
Table 103 Breakpoint Mask Bits for Second Address (Dual Address Mode) 555
Table 104 Breakpoint Mask Bits for Data Breakpoints (Full Breakpoint Mode) 556
Table 105 Absolute Maximum Ratings 564
Table 106 Operating Conditions 565
Table 107 Thermal Package Characteristics 567
Table 108 5V I/O Characteristics 567
Table 109 Supply Current Characteristics 568
Table 110 ATD Operating Characteristics 569
Table 111 ATD Electrical Characteristics 570
Table 112 ATD Conversion Performance 571
Table 113 NVM Timing Characteristics 575
Table 114 NVM Reliability Characteristics 576
Table 115 Voltage Regulator Recommended Load Capacitance 577
Table 116 Startup Characteristics 578
Table 117 Oscillator Characteristics 580
Table 118 PLL Characteristics 582
Table 119 SPI Master Mode Timing Characteristics 584
Table 120 SPI Slave Mode Timing Characteristics 586
Table 121 Expanded Bus Timing Characteristics – 16MHz 589

Freescale Semiconductor, Inc.

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

List of Tables

Freescale Semiconductor, Inc.

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

General Description

Contents

Introduction 19
Features 20
MC9S12T64 Block Diagram 23
Ordering Information 24

Introduction

The MC9S12T64 microcontroller unit (MCU) is a 16-bit device composed of standard on-chip peripherals including a 16-bit central processing unit (HCS12 CPU), 64K bytes of Flash EEPROM, 2K bytes of RAM, 2K bytes of CALRAM (Calibration RAM), two asynchronous serial communications interfaces (SCI), one serial peripheral interface (SPI), an 8 channel IC/OC enhanced capture timer, an 8-channel 10-bit analog-to-digital converter (ADC), an 8-channel pulse-width modulator (PWM), 25 discrete digital I/O channels (Port A, Port B, Port E and Port K). System resource mapping, clock generation, interrupt control and bus interfacing are managed by the System Integration Module (SIM). The MC9S12T64 has full 16-bit data paths throughout. However, the external bus can operate in an 8-bit narrow mode so single 8-bit wide memory can be interfaced for lower cost systems. The inclusion of a PLL circuit allows power consumption and performance to be adjusted to suit operational requirements.

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

Freescale Semiconductor, Inc.

General Description

Features

- HCS12 Core
  - 16-bit HCS12 CPU
    i. Upward compatible with M68HC11 instruction set
    ii. Interrupt stacking and programmer's model identical to M68HC11
    iii. 20-bit ALU
    iv. Instruction queue
    v. Enhanced indexed addressing
  - MEBI (Multiplexed External Bus Interface) [1]
  - MMC (Module Mapping Control)
  - INT (Interrupt control)
  - BKP (Breakpoints)
  - FBDM (Fast Background Debug Mode)
    i. Synchronous Serial Peripheral Interface (SPI mode) to allow fast read and write of internal memory contents.
    ii. 4M bit per second in SPI mode at 16MHz bus
    iii. Single Wire Interface
- CRG (low current oscillator, PLL, reset, clocks, COP watchdog, real time interrupt, clock monitor)
- LVD (Low Voltage Detector)
  - Low Voltage Detector to pull reset when the V_{DDR} Supply Voltage falls to LVD trip voltage.
- 64K byte Flash EEPROM [2]
  - Two 32K byte Flash EEPROM blocks independently programmable and erasable

1. Internal Flash EEPROM must be disabled to connect external memory devices with the external bus.

2. Whole 64K bytes of Flash EEPROM can not be used at a time, since 1K byte register block and 2K byte RAM array are always overlapped with Flash EEPROM.

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

Freescale Semiconductor, Inc.

  - Programmable during calibration
- 2K byte RAM
  - Single cycle misaligned 16-bit access
- 2K byte CALRAM (Calibration RAM)
  - 2K byte calibration block over Flash EEPROM
  - Access cycle compatible to Flash EEPROM
- 8 channel Analog-to-Digital Converters
  - 10-bit resolution
  - External conversion trigger
- Enhanced Capture Timer
  - 16-bit main counter with 7-bit prescaler
  - 8 programmable input capture or output compare channels
  - Two 8-bit or one 16-bit pulse accumulators
- 8 PWM channels
  - Programmable period and duty cycle
  - 8-bit 8-channel or 16-bit 4-channel
  - Separate control for each pulse width and duty cycle
  - Center-aligned or left-aligned outputs
  - Programmable clock select logic with a wide range of frequencies
  - Fast emergency shutdown input
  - Usable as interrupt inputs
- Serial interfaces
  - Two asynchronous Serial Communications Interfaces (SCI)
  - One synchronous Serial Peripheral Interface (SPI)
- Operating Condition
  - 32 MHz CPU equivalent to 16MHz bus operation

Freescale Semiconductor, Inc.

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

  - 2.25 to 2.75V Digital Supply Voltage generated using an internal voltage regulator
  - 4.75V to 5.25V Analog and I/O Supply Voltage
  - 80-Pin LQFP
- Technology: 0.25 micron CMOS

Freescale Semiconductor, Inc.

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

MC9S12T64 Block Diagram

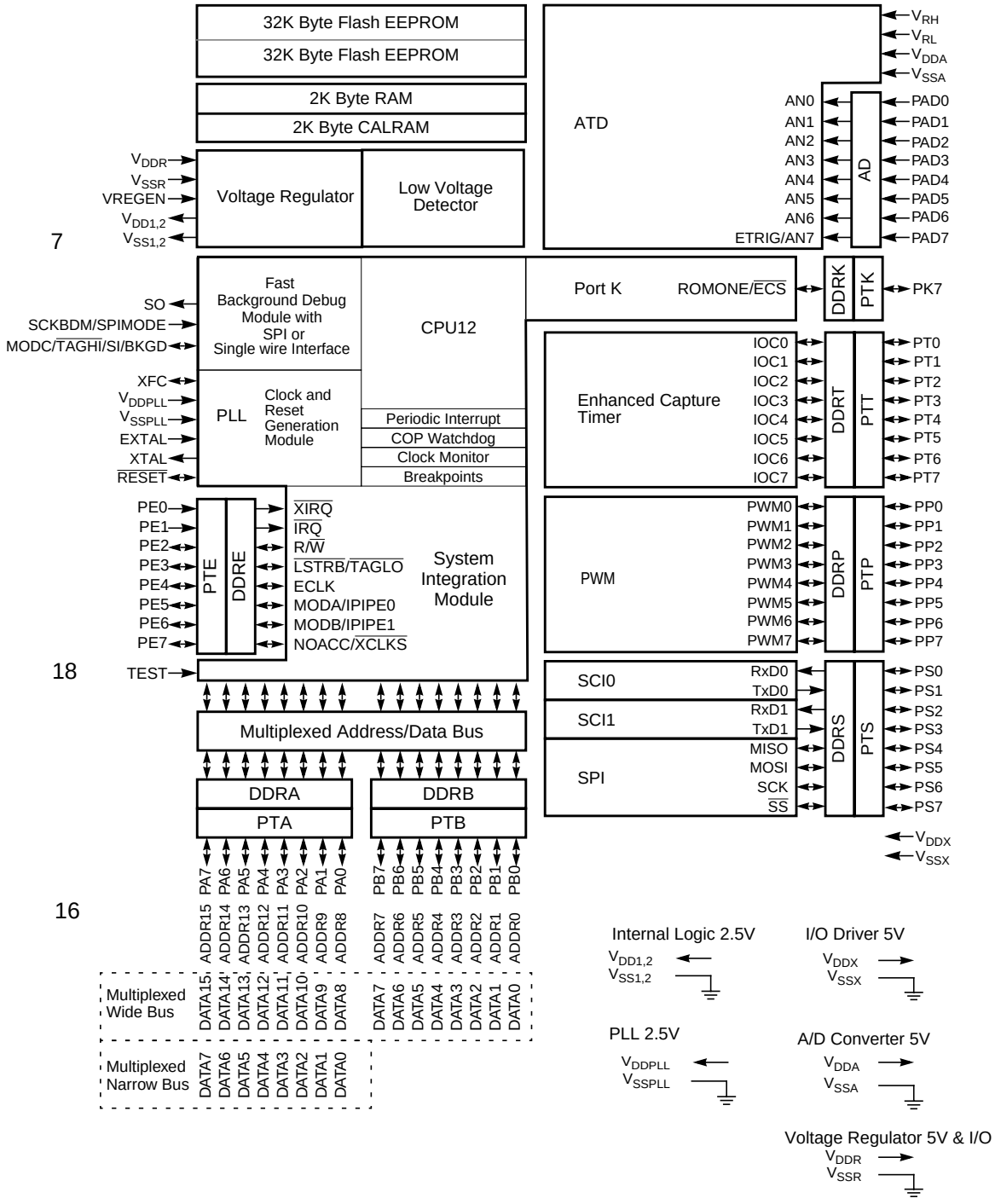


Figure 1 MC9S12T64 Block Diagram

Freescale Semiconductor, Inc.

Ordering Information

Table 1 MC9S12T64 Device Ordering Information

Package	Temperature		Voltage	Frequency	Order Number
	Range (˚C)	Designator			
80-Pin LQFP	–40 to +85	C	5V	Bus: 16MHz (CPU: 32MHz)	MC9S12T64CPK16
	–40 to +105	V			MC9S12T64VPK16
	–40 to +125	M			MC9S12T64MPK16

Table 2 MC9S12T64 Development Tools Ordering Information

Description	Details	Order Number
		(1)

1. Contact local sales for this information.

Freescale Semiconductor, Inc.

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

Central Processing Unit (CPU)

Contents

Introduction 25
Programming Model 26
Data Format Summary 34
Addressing Modes 35
Instruction Set Overview 36

Introduction

The HCS12 CPU is a high-speed, 16-bit processing unit. It has full 16-bit data paths and wider internal registers (up to 20 bits) for high-speed extended math instructions. The instruction set is a proper superset of the M68HC11instruction set. The HCS12 CPU allows instructions with odd byte counts, including many single-byte instructions. This provides efficient use of ROM space. An instruction pipe buffers program information so the CPU always has immediate access to at least three bytes of machine code at the start of every instruction. The HCS12 CPU also offers an extensive set of indexed addressing capabilities.

Freescale Semiconductor, Inc.

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

Programming Model

The Core CPU12 programming model, shown in Figure 2, is the same as that of the 68HC12 and 68HC11. The register set and data types used in the model are covered in the subsections that follow.

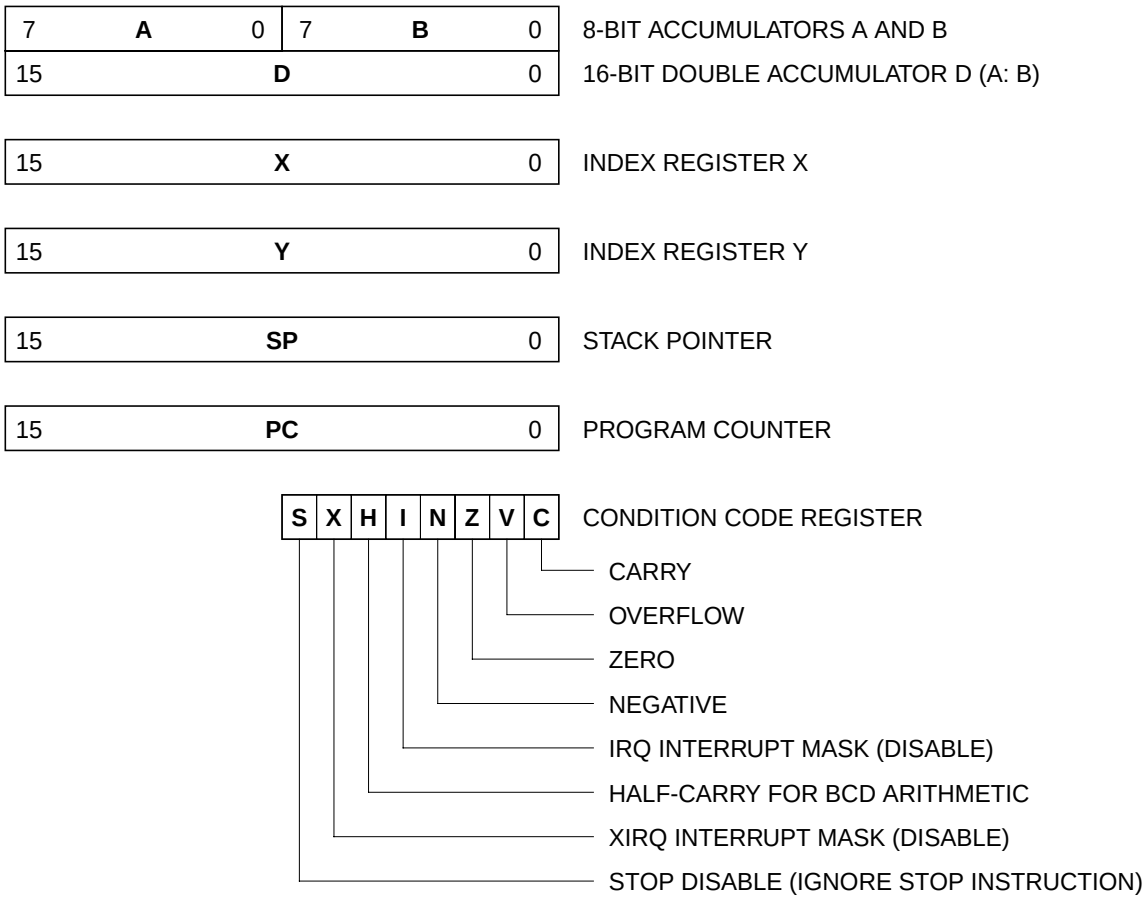


Figure 2 Programming Model

Accumulators

General-purpose 8-bit accumulators A and B hold operands and results of operations. Some instructions use the combined 8-bit accumulators, A:B, as a 16-bit double accumulator, D, with the most significant byte in A.

Freescale Semiconductor, Inc.

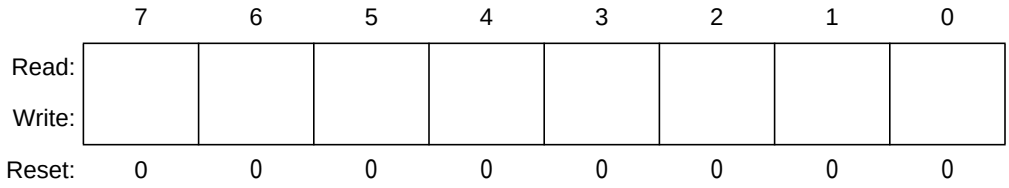


Figure 3 Accumulator A

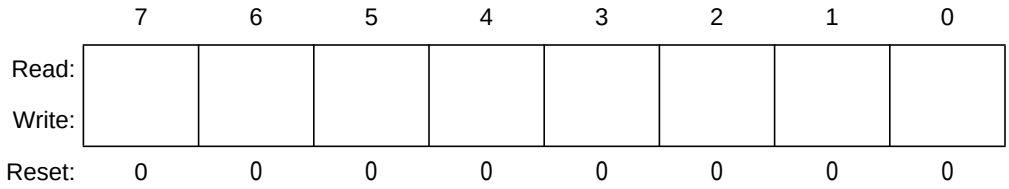


Figure 4 Accumulator B

Most operations can use accumulator A or B interchangeably. However, there are a few exceptions. Add, subtract, and compare instructions involving both A and B (ABA, SBA, and CBA) only operate in one direction, so it is important to verify that the correct operand is in the correct accumulator. The decimal adjust accumulator A (DAA) instruction is used after binary-coded decimal (BCD) arithmetic operations. There is no equivalent instruction to adjust accumulator B.

Index Registers (X and Y)

16-bit index registers X and Y are used for indexed addressing. In indexed addressing, the contents of an index register are added to a 5-bit, 9-bit, or 16-bit constant or to the contents of an accumulator to form the effective address of the instruction operand. Having two index registers is especially useful for moves and in cases where operands from two separate tables are used in a calculation.

Freescale Semiconductor, Inc.

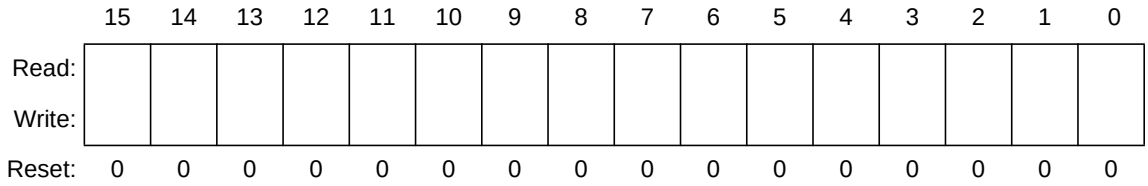


Figure 5 Index Register X

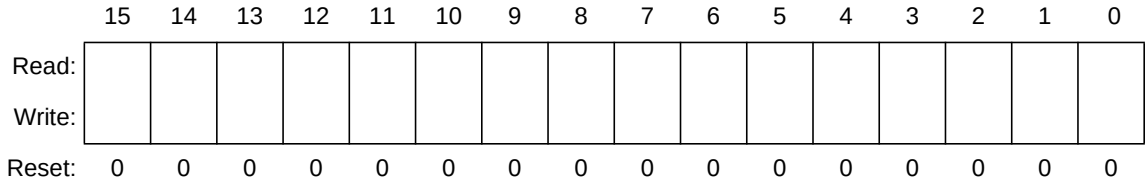


Figure 6 Index Register Y

Stack Pointer (SP)

The stack stores system context during subroutine calls and interrupts, and can also be used for temporary data storage. It can be located anywhere in the standard 64K byte address space and can grow to any size up to the total amount of memory available in the system.

SP holds the 16-bit address of the last stack location used. Normally, SP is initialized by one of the first instructions in an application program. The stack grows downward from the address pointed to by SP. Each time a byte is pushed onto the stack, the stack pointer is automatically decremented, and each time a byte is pulled from the stack, the stack pointer is automatically incremented.

When a subroutine is called, the address of the instruction following the calling instruction is automatically calculated and pushed onto the stack. Normally, a return from subroutine (RTS) is executed at the end of a subroutine. The return instruction loads the program counter with the previously stacked return address and execution continues at that address.

Freescale Semiconductor, Inc.

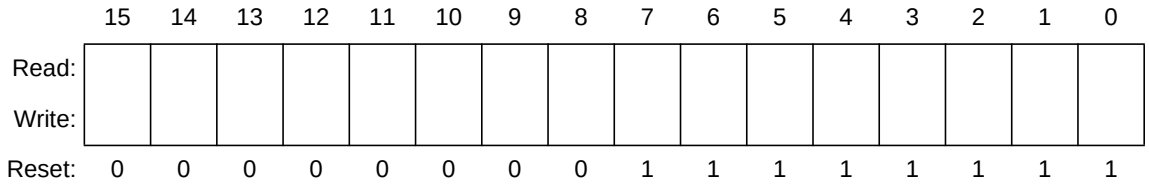


Figure 7 Stack Pointer (SP)

When an interrupt occurs, the CPU:

- Completes execution of the current instruction
- Calculates the address of the next instruction and pushes it onto the stack
- Pushes the contents of all the CPU registers onto the stack
- Loads the program counter with the address pointed to by the interrupt vector, and begins execution at that address

The stacked CPU registers are referred to as an interrupt stack frame. The Core stack frame is the same as that of the CPU.

Program Counter (PC)

PC is a 16-bit register that holds the address of the next instruction to be executed. The address in PC is automatically incremented each time an instruction is executed.

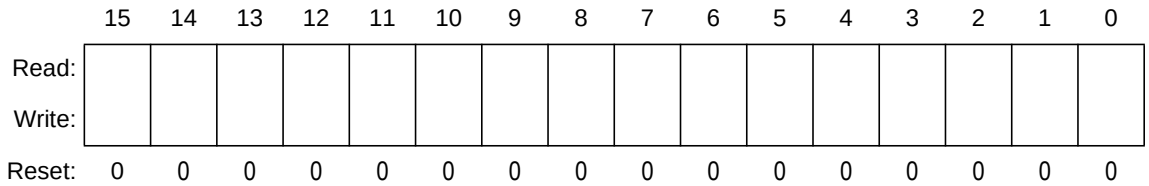


Figure 8 Program Counter (PC)

Condition Code Register (CCR)

CCR has five status bits, two interrupt mask bits, and a STOP instruction mask bit. It is named for the five conditions indicated by the status bits.

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

Freescale Semiconductor, Inc.

The status bits reflect the results of CPU operations. The five status bits are half-carry (H), negative (N), zero (Z), overflow (V), and carry/borrow (C). The half-carry bit is used only for BCD arithmetic operations. The N, Z, V, and C status bits allow for branching based on the results of a CPU operation.

Most instructions automatically update condition codes, so it is rarely necessary to execute extra instructions to load and test a variable. The condition codes affected by each instruction are shown in the HCS12 CORE User Guide.

The following paragraphs describe common uses of the condition codes. There are other, more specialized uses. For instance, the C status bit is used to enable weighted fuzzy logic rule evaluation. Specialized usages are described in the relevant portions of this guide and in the HCS12 CORE User Guide.

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	S	X	H	I	N	Z	V	C
Reset:	1	1	0	1	0	0	0	0

Figure 9 Condition Code Register (CCR)

S — STOP Mask Bit

Clearing the S bit enables the STOP instruction. Execution of a STOP instruction causes the on-chip oscillator to stop. This may be undesirable in some applications. When the S bit is set, the CPU treats the STOP instruction as a no-operation (NOP) instruction and continues on to the next instruction. Reset sets the S bit.

1 = STOP instruction disabled
0 = STOP instruction enabled

X — $\overline{\text{XIRQ}}$ Mask Bit

Clearing the X bit enables interrupt requests on the $\overline{\text{XIRQ}}$ pin. The $\overline{\text{XIRQ}}$ input is an updated version of the nonmaskable interrupt ($\overline{\text{NMI}}$) input found on earlier generations of Motorola microcontroller units

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

Freescale Semiconductor, Inc.

(MCUs). Nonmaskable interrupts are typically used to deal with major system failures such as loss of power. However, enabling nonmaskable interrupts before a system is fully powered and initialized can lead to spurious interrupts. The X bit provides a mechanism for masking nonmaskable interrupts until the system is stable.

Reset sets the X bit. As long as the X bit remains set, interrupt service requests made via the $\overline{\text{XIRQ}}$ pin are not recognized. Software must clear the X bit to enable interrupt service requests from the $\overline{\text{XIRQ}}$ pin. Once software clears the X bit, enabling $\overline{\text{XIRQ}}$ interrupt requests, only a reset can set it again. The X bit does not affect I bit maskable interrupt requests.

When the X bit is clear and an $\overline{\text{XIRQ}}$ interrupt request occurs, the CPU stacks the cleared X bit. It then automatically sets the X and I bits in the CCR to disable $\overline{\text{XIRQ}}$ and maskable interrupt requests during the $\overline{\text{XIRQ}}$ interrupt service routine.

An RTI instruction at the end of the interrupt service routine restores the cleared X bit to the CCR, re-enabling $\overline{\text{XIRQ}}$ interrupt requests.

1 = $\overline{\text{XIRQ}}$ interrupt requests disabled
0 = $\overline{\text{XIRQ}}$ interrupt requests enabled

H — Half-Carry Bit

The H bit indicates a carry from bit 3 of the result during an addition operation. The DAA instruction uses the value of the H bit to adjust the result in accumulator A to BCD format. ABA, ADD, and ADC are the only instructions that update the H bit.

1 = Carry from bit 3 after ABA, ADD, or ADC instruction
0 = No carry from bit 3 after ABA, ADD, or ADC instruction

I — Interrupt Mask Bit

Clearing the I bit enables maskable interrupt sources. Reset sets the I bit. To enable maskable interrupt requests, software must clear the I bit. Maskable interrupt requests that occur while the I bit is set remain pending until the I bit is cleared.

Freescale Semiconductor, Inc.

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

When the I bit is clear and a maskable interrupt request occurs, the CPU stacks the cleared I bit. It then automatically sets the I bit in the CCR to prevent other maskable interrupt requests during the interrupt service routine.

An RTI instruction at the end of the interrupt service routine restores the cleared I bit to the CCR, reenabling maskable interrupt requests. The I bit can be cleared within the service routine, but implementing a nested interrupt scheme requires great care, and seldom improves system performance.

1 = Maskable interrupt requests disabled
0 = Maskable interrupt requests enabled

N — Negative Bit

The N bit is set when the MSB of the result is set. N is most commonly used in two's complement arithmetic, where the MSB of a negative number is one and the MSB of a positive number is zero, but it has other uses. For instance, if the MSB of a register or memory location is used as a status bit, the user can test the bit by loading an accumulator.

1 = MSB of result set
0 = MSB of result clear

Z — Zero Bit

The Z bit is set when all the bits of the result are zeros. Compare instructions perform an internal implied subtraction, and the condition codes, including Z, reflect the results of that subtraction. The INX, DEX, INY, and DEY instructions affect the Z bit and no other condition bits. These operations can only determine = and ≠.

1 = Result all zeros
0 = Result not all zeros

V — Overflow Bit

The V bit is set when a two's complement overflow occurs as a result of an operation.

1 = Overflow
0 = No overflow

C — Carry Bit

Freescale Semiconductor, Inc.

The C bit is set when a carry occurs during addition or a borrow occurs during subtraction. The C bit also acts as an error flag for multiply and divide operations. Shift and rotate instructions operate through the C bit to facilitate multiple-word shifts.

1 = Carry or borrow
0 = No carry or borrow

HCS12 CPU registers are an integral part of the CPU and are not addressed as if they were memory locations.

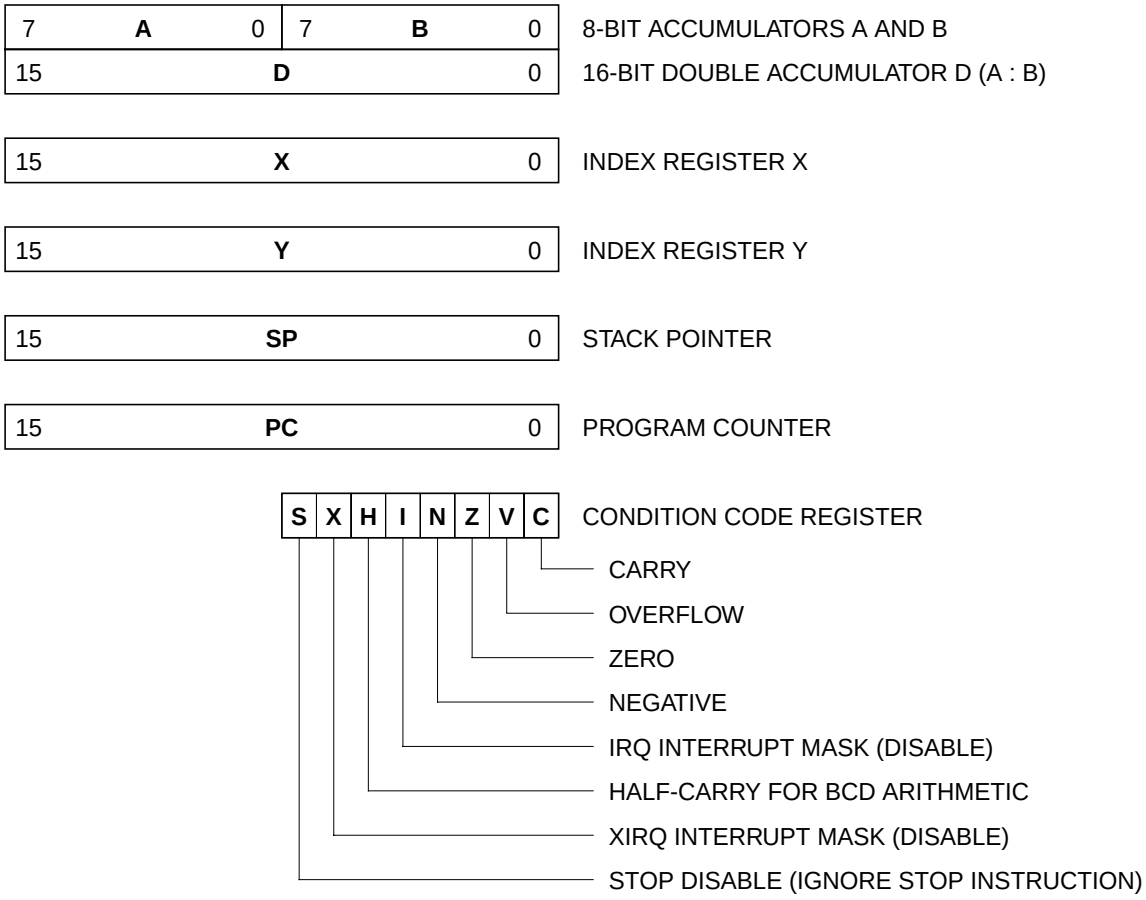


Figure 10 Programming Model

Freescale Semiconductor, Inc.

Data Format Summary

Following is a discussion of the data types used and their organization in memory for the Core.

Data Types

The CPU uses the following types of data:

- Bits
- 5-bit signed integers
- 8-bit signed and unsigned integers
- 8-bit, 2-digit binary coded decimal numbers
- 9-bit signed integers
- 16-bit signed and unsigned integers
- 16-bit effective addresses
- 32-bit signed and unsigned integers

NOTE: *Negative integers are represented in two's complement form.*

Five-bit and 9-bit signed integers are used only as offsets for indexed addressing modes. Sixteen-bit effective addresses are formed during addressing mode computations. Thirty-two-bit integer dividends are used by extended division instructions. Extended multiply and extended multiply-and-accumulate instructions produce 32-bit products.

Memory Organization

The standard HCS12 Core address space is 64K bytes. However, the CPU has special instructions to support paged memory expansion which increases the standard area by means of predefined windows within the available address space. See the Module Mapping Control (MMC) section for more information.

Eight-bit values can be stored at any odd or even byte address in available memory. Sixteen-bit values occupy two consecutive memory locations; the high byte is in the lowest address, but does not have to be aligned to an even boundary. Thirty-two-bit values occupy four

Freescale Semiconductor, Inc.

consecutive memory locations; the high byte is in the lowest address, but does not have to be aligned to an even boundary.

All I/O and all on-chip peripherals are memory-mapped. No special instruction syntax is required to access these addresses. On-chip register and memory mapping are determined at the SoC level and are configured during integration of the Core into the system.

Addressing Modes

A summary of the addressing modes used by the Core is given in Table 3 below. The operation of each of these modes is shown in detail in the HCS12 CORE user guide.

Table 3 Addressing Mode Summary

Addressing Mode	Source Form	Abbreviation	Description
Inherent	INST (no externally supplied operands)	INH	Operands (if any) are in CPU registers.
Immediate	INST #opr8i or INST #opr16i	IMM	Operand is included in instruction stream; 8-bit or 16-bit size implied by context.
Direct	INST opr8a	DIR	Operand is the lower 8-bits of an address in the range $0000–$00FF.
Extended	INST opr16a	EXT	Operand is a 16-bit address.
Relative	INST rel8 or INST rel16	REL	Effective address is the value in PC plus an 8-bit or 16-bit relative offset value.
Indexed (5-bit offset)	INST oprx5,xysp	IDX	Effective address is the value in X, Y, SP, or PC plus a 5-bit signed constant offset.
Indexed (predecrement)	INST oprx3,–xys	IDX	Effective address is the value in X, Y, or SP autodecremented by 1 to 8.
Indexed (preincrement)	INST oprx3,+xys	IDX	Effective address is the value in X, Y, or SP autoincremented by 1 to 8.
Indexed (postdecrement)	INST oprx3,xys–	IDX	Effective address is the value in X, Y, or SP. The value is postdecremented by 1 to 8.
Indexed (postincrement)	INST oprx3,xys+	IDX	Effective address is the value in X, Y, or SP. The value is postincremented by 1 to 8.
Indexed (accumulator offset)	INST abd,xysp	IDX	Effective address is the value in X, Y, SP, or PC plus the value in A, B, or D.
Indexed (9-bit offset)	INST oprx9,xysp	IDX1	Effective address is the value in X, Y, SP, or PC plus a 9-bit signed constant offset.

Freescale Semiconductor, Inc.

Table 3 Addressing Mode Summary (Continued)

Addressing Mode	Source Form	Abbreviation	Description
Indexed (16-bit offset)	INST oprx16,xysp	IDX2	Effective address is the value in X, Y, SP, or PC plus a 16-bit constant offset.
Indexed-indirect (16-bit offset)	INST [oprx16,xysp]	[IDX2]	The value in X, Y, SP, or PC plus a 16-bit constant offset points to the effective address.
Indexed-indirect (D accumulator offset)	INST [D,xysp]	[D,IDX]	The value in X, Y, SP, or PC plus the value in D points to the effective address.

Instruction Set Overview

All memory and I/O are mapped in a common 64K byte address space, allowing the same set of instructions to access memory, I/O, and control registers. Load, store, transfer, exchange, and move instructions facilitate movement of data to and from memory and peripherals.

There are instructions for signed and unsigned addition, division and multiplication with 8-bit, 16-bit, and some larger operands.

Special arithmetic and logic instructions aid stacking operations, indexing, BCD calculation, and condition code register manipulation. There are also dedicated instructions for multiply and accumulate operations, table interpolation, and specialized mathematical calculations for fuzzy logic operations.

A summary of the CPU instruction set is given in Figure 4 below. A detailed overview of the entire instruction set is covered in the HCS12 Core User Guide.

Table 4 Instruction Set Summary

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
ABA	Add B to A; (A)+(B)⇒A	INH	18 06	00	– – Δ – Δ Δ Δ Δ
ABXSame as LEAX B,X	Add B to X; (X)+(B)⇒X	IDX	1A E5	Pf	– – – – – – – –
ABYSame as LEAY B,Y	Add B to Y; (Y)+(B)⇒Y	IDX	19 ED	Pf	– – – – – – – –

Freescale Semiconductor, Inc.

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
ADCA *#opr8i* ADCA *opr8a* ADCA *opr16a* ADCA *oprx0_xysppc* ADCA *oprx9,xysppc* ADCA *oprx16,xysppc* ADCA [D,*xysppc*] ADCA [*oprx16,xysppc*]	Add with carry to A; (A)+(M)+C⇒A or (A)+imm+C⇒A	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	89 ii 99 dd B9 hh ll A9 xb A9 xb ff A9 xb ee ff A9 xb A9 xb ee ff	P rPf rPO rPf rPO frPP fIfrPf fIPrPf	– – Δ – Δ Δ Δ Δ
ADCB *#opr8i* ADCB *opr8a* ADCB *opr16a* ADCB *oprx0_xysppc* ADCB *oprx9,xysppc* ADCB *oprx16,xysppc* ADCB [D,*xysppc*] ADCB [*oprx16,xysppc*]	Add with carry to B; (B)+(M)+C⇒B or (B)+imm+C⇒B	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	C9 ii D9 dd F9 hh ll E9 xb E9 xb ff E9 xb ee ff E9 xb E9 xb ee ff	P rPf rPO rPf rPO frPP fIfrPf fIPrPf	– – Δ – Δ Δ Δ Δ

ADDA *#opr8i*
ADDA *opr8a*
ADDA *opr16a*
ADDA *oprx0_xysppc*
ADDA *oprx9,xysppc*
ADDA *oprx16,xysppc*
ADDA [D,*xysppc*]
ADDA [*oprx16,xysppc*]
	Add to A; (A)+(M)⇒A
	or (A)+imm⇒A
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			8B ii
			9B dd
			BB hh ll
			AB xb
			AB xb ff
			AB xb ee ff
			AB xb
			AB xb ee ff
				P
				rPf
				rPO
				rPf
				rPO
				frPP
				fIfrPf
				fIPrPf
					– – Δ – Δ Δ Δ Δ

ADDB *#opr8i*
ADDB *opr8a*
ADDB *opr16a*
ADDB *oprx0_xysppc*
ADDB *oprx9,xysppc*
ADDB *oprx16,xysppc*
ADDB [D,*xysppc*]
ADDB [*oprx16,xysppc*]
	Add to B; (B)+(M)⇒B
	or (B)+imm⇒B
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			CB ii
			DB dd
			FB hh ll
			EB xb
			EB xb ff
			EB xb ee ff
			EB xb
			EB xb ee ff
				P
				rPf
				rPO
				rPf
				rPO
				frPP
				fIfrPf
				fIPrPf
					– – Δ – Δ Δ Δ Δ

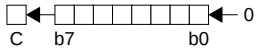
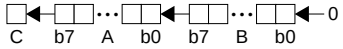
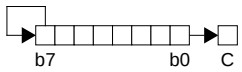
ADDD *#opr16i*
ADDD *opr8a*
ADDD *opr16a*
ADDD *oprx0_xysppc*
ADDD *oprx9,xysppc*
ADDD *oprx16,xysppc*
ADDD [D,*xysppc*]
ADDD [*oprx16,xysppc*]
	Add to D; (A:B)+(M:M+1)⇒A:B
	or (A:B)+imm⇒A:B
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			C3 jj kk
			D3 dd
			F3 hh ll
			E3 xb
			E3 xb ff
			E3 xb ee ff
			E3 xb
			E3 xb ee ff
				PO
				RPf
				RPO
				RPf
				RPO
				fRPP
				fIfRPf
				fIPRPf
					– – – – Δ Δ Δ Δ

ANDA *#opr8i* ANDA *opr8a* ANDA *opr16a* ANDA *oprx0_xysppc* ANDA *oprx9,xysppc* ANDA *oprx16,xysppc* ANDA [D,*xysppc*] ANDA [*oprx16,xysppc*]	AND with A; (A)•(M)⇒A or (A)•imm⇒A	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	84 ii 94 dd B4 hh ll A4 xb A4 xb ff A4 xb ee ff A4 xb A4 xb ee ff	P rPf rPO rPf rPO frPP fIfrPf fIPrPf	– – – – Δ Δ 0 –
ANDB *#opr8i* ANDB *opr8a* ANDB *opr16a* ANDB *oprx0_xysppc* ANDB *oprx9,xysppc* ANDB *oprx16,xysppc* ANDB [D,*xysppc*] ANDB [*oprx16,xysppc*]	AND with B; (B)•(M)⇒B or (B)•imm⇒B	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	C4 ii D4 dd F4 hh ll E4 xb E4 xb ff E4 xb ee ff E4 xb E4 xb ee ff	P rPf rPO rPf rPO frPP fIfrPf fIPrPf	– – – – Δ Δ 0 –
ANDCC *#opr8i*	AND with CCR; (CCR)•imm⇒CCR	IMM	10 ii	P	⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓

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

Freescale Semiconductor, Inc.

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
ASL *opr16a*Same as LSL ASL *oprx0_xysp* ASL *oprx9,xysppc* ASL *oprx16,xysppc* ASL [D,*xysppc*] ASL [*oprx16,xysppc*] ASLASame as LSLA ASLBSame as LSLB	Arithmetic shift left M  Arithmetic shift left A Arithmetic shift left B	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	78 hh ll 68 xb 68 xb ff 68 xb ee ff 68 xb 68 xb ee ff 48 58	rPw0 rPw rPw0 frPwP fIfrPw fIPrPw 0 0	– – – – Δ Δ Δ Δ
ASLDSame as LSLD	Arithmetic shift left D  	INH	59	0	– – – – Δ Δ Δ Δ
ASR *opr16a* ASR *oprx0_xysppc* ASR *oprx9,xysppc* ASR *oprx16,xysppc* ASR [D,*xysppc*] ASR [*oprx16,xysppc*] ASRA ASRB	Arithmetic shift right M  Arithmetic shift right A Arithmetic shift right B	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	77 hh ll 67 xb 67 xb ff 67 xb ee ff 67 xb 67 xb ee ff 47 57	rPw0 rPw rPw0 frPwP fIfrPw fIPrPw 0 0	– – – – Δ Δ Δ Δ
BCC *rel8*Same as BHS	Branch if C clear; if C=0, then (PC)+2+rel⇒PC	REL	24 rr	PPP (branch) P (no branch)	– – – – – – – –
BCLR *opr8a*, *msk8* BCLR *opr16a*, *msk8* BCLR *oprx0_xysppc*, *msk8* BCLR *oprx9,xysppc*, *msk8* BCLR *oprx16,xysppc*, *msk8*	Clear bit(s) in M; (M)•mask byte⇒M	DIR EXT IDX IDX1 IDX2	4D dd mm 1D hh ll mm 0D xb mm 0D xb ff mm 0D xb ee ff mm	rPw0 rPwP rPw0 rPwP frPwP0	– – – – Δ Δ 0 –
BCS *rel8*Same as BLO	Branch if C set; if C=1, then (PC)+2+rel⇒PC	REL	25 rr	PPP (branch) P (no branch)	– – – – – – – –
BEQ *rel8*	Branch if equal; if Z=1, then (PC)+2+rel⇒PC	REL	27 rr	PPP (branch) P (no branch)	– – – – – – – –
BGE *rel8*	Branch if ≥ 0, signed; if N⊕V=0, then (PC)+2+rel⇒PC	REL	2C rr	PPP (branch) P (no branch)	– – – – – – – –
BGND	Enter background debug mode	INH	00	VfPPP	– – – – – – – –
BGT *rel8*	Branch if > 0, signed; if Z | (N⊕V)=0, then (PC)+2+rel⇒PC	REL	2E rr	PPP (branch) P (no branch)	– – – – – – – –
BHI *rel8*	Branch if higher, unsigned; if C | Z=0, then (PC)+2+rel⇒PC	REL	22 rr	PPP (branch) P (no branch)	– – – – – – – –
BHS *rel8*Same as BCC	Branch if higher or same, unsigned; if C=0, then (PC)+2+rel⇒PC	REL	24 rr	PPP (branch) P (no branch)	– – – – – – – –
BITA #*opr8i* BITA *opr8a* BITA *opr16a* BITA *oprx0_xysppc* BITA *oprx9,xysppc* BITA *oprx16,xysppc* BITA [D,*xysppc*] BITA [*oprx16,xysppc*]	Bit test A; (A)•(M) or (A)•imm	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	85 ii 95 dd B5 hh ll A5 xb A5 xb ff A5 xb ee ff A5 xb A5 xb ee ff	P rPf rP0 rPf rP0 frPP fIfrPf fIPrPf	– – – – Δ Δ 0 –
BITB #*opr8i* BITB *opr8a* BITB *opr16a* BITB *oprx0_xysppc* BITB *oprx9,xysppc* BITB *oprx16,xysppc* BITB [D,*xysppc*] BITB [*oprx16,xysppc*]	Bit test B; (B)•(M) or (B)•imm	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	C5 ii D5 dd F5 hh ll E5 xb E5 xb ff E5 xb ee ff E5 xb E5 xb ee ff	P rPf rP0 rPf rP0 frPP fIfrPf fIPrPf	– – – – Δ Δ 0 –

Freescale Semiconductor, Inc.

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
BLE *rel8*	Branch if ≤ 0,signed; if Z | (N⊕V)=1, then (PC)+2+rel⇒PC	REL	2F rr	PPP (branch) P (no branch)	– – – – – – – –
BLO *rel8*Same as BCS	Branch if lower, unsigned; if C=1, then (PC)+2+rel⇒PC	REL	25 rr	PPP (branch) P (no branch)	– – – – – – – –
BLS *rel8*	Branch if lower or same, unsigned; if C | Z=1, then (PC)+2+rel⇒PC	REL	23 rr	PPP (branch) P (no branch)	– – – – – – – –
BLT *rel8*	Branch if < 0, signed; if N⊕V=1, then (PC)+2+rel⇒PC	REL	2D rr	PPP (branch) P (no branch)	– – – – – – – –
BMI *rel8*	Branch if minus; if N=1, then (PC)+2+rel⇒PC	REL	2B rr	PPP (branch) P (no branch)	– – – – – – – –
BNE *rel8*	Branch if not equal to 0; if Z=0, then (PC)+2+rel⇒PC	REL	26 rr	PPP (branch) P (no branch)	– – – – – – – –
BPL *rel8*	Branch if plus; if N=0, then (PC)+2+rel⇒PC	REL	2A rr	PPP (branch) P (no branch)	– – – – – – – –
BRA *rel8*	Branch always	REL	20 rr	PPP	– – – – – – – –
BRCLR *opr8a*, *msk8*, *rel8* BRCLR *opr16a*, *msk8*, *rel8* BRCLR *oprx0_xysppc*, *msk8*, *rel8* BRCLR *oprx9*,*xysppc*, *msk8*, *rel8* BRCLR *oprx16*,*xysppc*, *msk8*, *rel8*	Branch if bit(s) clear; if (M)•(mask byte)=0, then (PC)+2+rel⇒PC	DIR EXT IDX IDX1 IDX2	4F dd mm rr 1F hh ll mm rr 0F xb mm rr 0F xb ff mm rr 0F xb ee ff mm rr	rPPP rfPPP rPPP rfPPP PrfPPP	– – – – – – – –
BRN *rel8*	Branch never	REL	21 rr	P	– – – – – – – –
BRSET *opr8*, *msk8*, *rel8* BRSET *opr16a*, *msk8*, *rel8* BRSET *oprx0_xysppc*, *msk8*, *rel8* BRSET *oprx9*,*xysppc*, *msk8*, *rel8* BRSET *oprx16*,*xysppc*, *msk8*, *rel8*	Branch if bit(s) set; if ($\overline{M}$)•(mask byte)=0, then (PC)+2+rel⇒PC	DIR EXT IDX IDX1 IDX2	4E dd mm rr 1E hh ll mm rr 0E xb mm rr 0E xb ff mm rr 0E xb ee ff mm rr	rPPP rfPPP rPPP rfPPP PrfPPP	– – – – – – – –
BSET *opr8*, *msk8* BSET *opr16a*, *msk8* BSET *oprx0_xysppc*, *msk8* BSET *oprx9*,*xysppc*, *msk8* BSET *oprx16*,*xysppc*, *msk8*	Set bit(s) in M (M) | mask byte⇒M	DIR EXT IDX IDX1 IDX2	4C dd mm 1C hh ll mm 0C xb mm 0C xb ff mm 0C xb ee ff mm	rPwO rPwP rPwO rPwP frPwPO	– – – – Δ Δ 0 –
BSR *rel8*	Branch to subroutine; (SP)–2⇒SP RTN_H:RTN_L⇒M_{SP}:M_{SP+1} (PC)+2+rel⇒PC	REL	07 rr	SPPP	– – – – – – – –
BVC *rel8*	Branch if V clear; if V=0, then (PC)+2+rel⇒PC	REL	28 rr	PPP (branch) P (no branch)	– – – – – – – –
BVS *rel8*	Branch if V set; if V=1, then (PC)+2+rel⇒PC	REL	29 rr	PPP (branch) P (no branch)	– – – – – – – –
CALL *opr16a*, *page* CALL *oprx0_xysppc*, *page* CALL *oprx9*,*xysppc*, *page* CALL *oprx16*,*xysppc*, *page* CALL [D,*xysppc*] CALL [*oprx16*, *xysppc*]	Call subroutine in expanded memory (SP)–2⇒SP RTN_H:RTN_L⇒M_{SP}:M_{SP+1} (SP)–1⇒SP; (PPG)⇒M_{SP} pg⇒PPAGE register subroutine address⇒PC	EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	4A hh ll pg 4B xb pg 4B xb ff pg 4B xb ee ff pg 4B xb 4B xb ee ff	gnSsPPP gnSsPPP gnSsPPP fgnSsPPP fIignSsPPP fIignSsPPP	– – – – – – – –
CBA	Compare A to B; (A)–(B)	INH	18 17	OO	– – – – Δ Δ Δ Δ
CLCSame as ANDCC #$FE	Clear C bit	IMM	10 FE	P	– – – – – – – 0
CLISame as ANDCC #$EF	Clear I bit	IMM	10 EF	P	– – – 0 – – – –

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

Freescale Semiconductor, Inc.

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
CLR *oprx16a* CLR *oprx0_xysppc* CLR *oprx9,xysppc* CLR *oprx16,xysppc* CLR [D,*xysppc*] CLR [*oprx16,xysppc*] CLRA CLRB	Clear M; $00⇒M Clear A; $00⇒A Clear B; $00⇒B	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	79 hh ll 69 xb 69 xb ff 69 xb ee ff 69 xb 69 xb ee ff 87 C7	PwO Pw PwO PwP PIfw PIPw O O	– – – – 0 1 0 0
CLVSame as ANDCC #$FD	Clear V	IMM	10 FD	P	– – – – – – 0 –

CMPA #*opr8i*
CMPA *opr8a*
CMPA *opr16a*
CMPA *oprx0_xysppc*
CMPA *oprx9,xysppc*
CMPA *oprx16,xysppc*
CMPA [D,*xysppc*]
CMPA [*oprx16,xysppc*]
	Compare A
	(A)–(M) or (A)–imm
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			81 ii
			91 dd
			B1 hh ll
			A1 xb
			A1 xb ff
			A1 xb ee ff
			A1 xb
			A1 xb ee ff
				P
				rPf
				rPO
				rPf
				rPO
				frPP
				fIfrPf
				fIPrPf
					– – – – Δ Δ Δ Δ

CMPB #*opr8i*
CMPB *opr8a*
CMPB *opr16a*
CMPB *oprx0_xysppc*
CMPB *oprx9,xysppc*
CMPB *oprx16,xysppc*
CMPB [D,*xysppc*]
CMPB [*oprx16,xysppc*]
	Compare B
	(B)–(M) or (B)–imm
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			C1 ii
			D1 dd
			F1 hh ll
			E1 xb
			E1 xb ff
			E1 xb ee ff
			E1 xb
			E1 xb ee ff
				P
				rPf
				rPO
				rPf
				rPO
				frPP
				fIfrPf
				fIPrPf
					– – – – Δ Δ Δ Δ

COM *oprx16a* COM *oprx0_xysppc* COM *oprx9,xysppc* COM *oprx16,xysppc* COM [D,*xysppc*] COM [*oprx16,xysppc*] COMA COMB	Complement M; ($\overline{M}$)=$FF–(M)⇒M Complement A; ($\overline{A}$)=$FF–(A)⇒A Complement B; ($\overline{B}$)=$FF–(B)⇒B	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	71 hh ll 61 xb 61 xb ff 61 xb ee ff 61 xb 61 xb ee ff 41 51	rPwO rPw rPwO frPwP fIfrPw fIPrPw O O	– – – – Δ Δ 0 1
CPD #*opr16i* CPD *opr8a* CPD *opr16a* CPD *oprx0_xysppc* CPD *oprx9,xysppc* CPD *oprx16,xysppc* CPD [D,*xysppc*] CPD [*oprx16,xysppc*]	Compare D (A:B)–(M:M+1) or (A:B)–imm	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	8C jj kk 9C dd BC hh ll AC xb AC xb ff AC xb ee ff AC xb AC xb ee ff	PO RPf RPO RPf RPO fRPP fIfRPf fIPRPf	– – – – Δ Δ Δ Δ
CPS #*opr16i* CPS *opr8a* CPS *opr16a* CPS *oprx0_xysppc* CPS *oprx9,xysppc* CPS *oprx16,xysppc* CPS [D,*xysppc*] CPS [*oprx16,xysppc*]	Compare SP (SP)–(M:M+1) or (SP)–imm	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	8F jj kk 9F dd BF hh ll AF xb AF xb ff AF xb ee ff AF xb AF xb ee ff	PO RPf RPO RPf RPO fRPP fIfRPf fIPRPf	– – – – Δ Δ Δ Δ
CPX #*opr16i* CPX *opr8a* CPX *opr16a* CPX *oprx0_xysppc* CPX *oprx9,xysppc* CPX *oprx16,xysppc* CPX [D,*xysppc*] CPX [*oprx16,xysppc*]	Compare X (X)–(M:M+1) or (X)–imm	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	8E jj kk 9E dd BE hh ll AE xb AE xb ff AE xb ee ff AE xb AE xb ee ff	PO RPf RPO RPf RPO fRPP fIfRPf fIPRPf	– – – – Δ Δ Δ Δ

Freescale Semiconductor, Inc.

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
CPY #*opr16i* CPY *opr8a* CPY *opr16a* CPY *oprx0_xysppc* CPY *oprx9*,*xysppc* CPY *oprx16*,*xysppc* CPY [D,*xysppc*] CPY [*oprx16*,*xysppc*]	Compare Y (Y)–(M:M+1) or (Y)–imm	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	8D jj kk 9D dd BD hh ll AD xb AD xb ff AD xb ee ff AD xb AD xb ee ff	PO RPf RPO RPf RPO fRPP fIfRPf fIPRPf	– – – – Δ Δ Δ Δ
DAA	Decimal adjust A for BCD	INH	18 07	OfO	– – – – Δ Δ ? Δ
DBEQ *abdxysp*, *rel9*	Decrement and branch if equal to 0 (counter)–1⇒counter if (counter)=0, then branch	REL (9-bit)	04 lb rr	PPP (branch) PPO (no branch)	– – – – – – – –
DBNE *abdxysp*, *rel9*	Decrement and branch if not equal to 0; (counter)–1⇒counter; if (counter)≠0, then branch	REL (9-bit)	04 lb rr	PPP (branch) PPO (no branch)	– – – – – – – –
DEC *opr16a* DEC *oprx0_xysppc* DEC *oprx9*,*xysppc* DEC *oprx16*,*xysppc* DEC [D,*xysppc*] DEC [*oprx16*,*xysppc*] DECA DECB	Decrement M; (M)–1⇒M Decrement A; (A)–1⇒A Decrement B; (B)–1⇒B	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	73 hh ll 63 xb 63 xb ff 63 xb ee ff 63 xb 63 xb ee ff 43 53	rPwO rPw rPwO frPwP fIfrPw fIPrPw O O	– – – – Δ Δ Δ –
DESSame as LEAS –1,SP	Decrement SP; (SP)–1⇒SP	IDX	1B 9F	Pf	– – – – – – – –
DEX	Decrement X; (X)–1⇒X	INH	09	O	– – – – – Δ – –
DEY	Decrement Y; (Y)–1⇒Y	INH	03	O	– – – – – Δ – –
EDIV	Extended divide, unsigned; 32 by 16 to 16-bit; (Y:D)÷(X)⇒Y; remainder⇒D	INH	11	fffffffffO	– – – – Δ Δ Δ Δ
EDIVS	Extended divide,signed; 32 by 16 to 16-bit; (Y:D)÷(X)⇒Y remainder⇒D	INH	18 14	OfffffffffO	– – – – Δ Δ Δ Δ
EMACS *opr16a*	Extended multiply and accumulate, signed; (M_X:M_{X+1})×(M_Y:M_{Y+1})+(M~M+3)⇒M~M+3; 16 by 16 to 32-bit	Special	18 12 hh ll	ORROfffRRfWWP	– – – – Δ Δ Δ Δ
EMAXD *oprx0_xysppc* EMAXD *oprx9*,*xysppc* EMAXD *oprx16*,*xysppc* EMAXD [D,*xysppc*] EMAXD [*oprx16*,*xysppc*]	Extended maximum in D; put larger of 2 unsigned 16-bit values in D MAX[(D), (M:M+1)]⇒D N, Z, V, C bits reflect result of internal compare [(D)–(M:M+1)]	IDX IDX1 IDX2 [D,IDX] [IDX2]	18 1A xb 18 1A xb ff 18 1A xb ee ff 18 1A xb 18 1A xb ee ff	ORPf ORPO OfRPP OfIfRPf OfIPRPf	– – – – Δ Δ Δ Δ
EMAXM *oprx0_xysppc* EMAXM *oprx9*,*xysppc* EMAXM *oprx16*,*xysppc* EMAXM [D,*xysppc*] EMAXM [*oprx16*,*xysppc*]	Extended maximum in M; put larger of 2 unsigned 16-bit values in M MAX[(D), (M:M+1)]⇒M:M+1 N, Z, V, C bits reflect result of internal compare [(D)–(M:M+1)]	IDX IDX1 IDX2 [D,IDX] [IDX2]	18 1E xb 18 1E xb ff 18 1E xb ee ff 18 1E xb 18 1E xb ee ff	ORPW ORPWO OfRPWP OfIfRPW OfIPRPW	– – – – Δ Δ Δ Δ
EMIND *oprx0_xysppc* EMIND *oprx9*,*xysppc* EMIND *oprx16*,*xysppc* EMIND [D,*xysppc*] EMIND [*oprx16*,*xysppc*]	Extended minimum in D; put smaller of 2 unsigned 16-bit values in D MIN[(D), (M:M+1)]⇒D N, Z, V, C bits reflect result of internal compare [(D)–(M:M+1)]	IDX IDX1 IDX2 [D,IDX] [IDX2]	18 1B xb 18 1B xb ff 18 1B xb ee ff 18 1B xb 18 1B xb ee ff	ORPf ORPO OfRPP OfIfRPf OfIPRPf	– – – – Δ Δ Δ Δ

Freescale Semiconductor, Inc.

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

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
EMINM *oprx0_xysppc* EMINM *oprx9,xysppc* EMINM *oprx16,xysppc* EMINM [D,*xysppc*] EMINM [*oprx16,xysppc*]	Extended minimum in M; put smaller of 2 unsigned 16-bit values in M MIN[(D), (M:M+1)]⇒M:M+1 N, Z, V, C bits reflect result of internal compare [(D)–(M:M+1)]	IDX IDX1 IDX2 [D,IDX] [IDX2]	18 1F xb 18 1F xb ff 18 1F xb ee ff 18 1F xb 18 1F xb ee ff	ORPW ORPWO OfRPWP OfIfRPW OfIPRPW	– – – – Δ Δ Δ Δ
EMUL	Extended multiply, unsigned (D)×(Y)⇒Y:D; 16 by 16 to 32-bit	INH	13	ffO	– – – – Δ Δ – Δ
EMULS	Extended multiply, signed (D)×(Y)⇒Y:D; 16 by 16 to 32-bit	INH	18 13	OfO OffO (if followed by page 2 instruction)	– – – – Δ Δ – Δ
EORA #*opr8i* EORA *opr8a* EORA *opr16a* EORA *oprx0_xysppc* EORA *oprx9,xysppc* EORA *oprx16,xysppc* EORA [D,*xysppc*] EORA [*oprx16,xysppc*]	Exclusive OR A (A)⊕(M)⇒A or (A)⊕imm⇒A	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	88 ii 98 dd B8 hh ll A8 xb A8 xb ff A8 xb ee ff A8 xb A8 xb ee ff	P rPf rPO rPf rPO frPP fIfrPf fIPrPf	– – – – Δ Δ 0 –
EORB #*opr8i* EORB *opr8a* EORB *opr16a* EORB *oprx0_xysppc* EORB *oprx9,xysppc* EORB *oprx16,xysppc* EORB [D,*xysppc*] EORB [*oprx16,xysppc*]	Exclusive OR B (B)⊕(M)⇒B or (B)⊕imm⇒B	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	C8 ii D8 dd F8 hh ll E8 xb E8 xb ff E8 xb ee ff E8 xb E8 xb ee ff	P rPf rPO rPf rPO frPP fIfrPf fIPrPf	– – – – Δ Δ 0 –
ETBL *oprx0_xysppc*	Extended table lookup and interpolate, 16-bit; (M:M+1)+ [(B)×((M+2:M+3)–(M:M+1))]⇒D	IDX	18 3F xb	ORRffffffP	– – – – Δ Δ – Δ
Before executing ETBL, initialize B with fractional part of lookup value; initialize index register to point to first table entry (M:M+1). No extensions or indirect addressing allowed.					
EXG *abcdxysp*,*abcdxysp*	Exchange register contents (r1)⇔(r2) r1 and r2 same size $00:(r1)⇒r2r1=8-bit; r2=16-bit ($r1_L$)⇔(r2)r1=16-bit; r2=8-bit	INH	B7 eb	P	– – – – – – – –
FDIV	Fractional divide; (D)÷(X)⇒X remainder⇒D; 16 by 16-bit	INH	18 11	OfffffffffffO	– – – – – Δ Δ Δ
IBEQ *abdxysp*, *rel9*	Increment and branch if equal to 0 (counter)+1⇒counter If (counter)=0, then branch	REL (9-bit)	04 lb rr	PPP (branch) PPO (no branch)	– – – – – – – –
IBNE *abdxysp*, *rel9*	Increment and branch if not equal to 0 (counter)+1⇒counter If (counter)≠0, then branch	REL (9-bit)	04 lb rr	PPP (branch) PPO (no branch)	– – – – – – – –
IDIV	Integer divide, unsigned; (D)÷(X)⇒X Remainder⇒D; 16 by 16-bit	INH	18 10	OfffffffffffO	– – – – – Δ 0 Δ
IDIVS	Integer divide, signed; (D)÷(X)⇒X Remainder⇒D; 16 by 16-bit	INH	18 15	OfffffffffffO	– – – – Δ Δ Δ Δ
INC *opr16a* INC *oprx0_xysppc* INC *oprx9,xysppc* INC *oprx16,xysppc* INC [D,*xysppc*] INC [*oprx16,xysppc*] INCA INCB	Increment M; (M)+1⇒M Increment A; (A)+1⇒A Increment B; (B)+1⇒B	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	72 hh ll 62 xb 62 xb ff 62 xb ee ff 62 xb 62 xb ee ff 42 52	rPwO rPw rPwO frPwP fIfrPw fIPrPw O O	– – – – Δ Δ Δ –

Freescale Semiconductor, Inc.

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
INSSame as LEAS 1,SP	Increment SP; (SP)+1⇒SP	IDX	1B 81	Pf	– – – – – – – –
INX	Increment X; (X)+1⇒X	INH	08	O	– – – – – Δ – –
INY	Increment Y; (Y)+1⇒Y	INH	02	O	– – – – – Δ – –
JMP *opr16a* JMP *oprx0_xysppc* JMP *oprx9,xysppc* JMP *oprx16,xysppc* JMP [D,*xysppc*] JMP [*oprx16,xysppc*]	Jump Subroutine address⇒PC	EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	06 hh ll 05 xb 05 xb ff 05 xb ee ff 05 xb 05 xb ee ff	PPP PPP PPP fPPP fIfPPP fIfPPP	– – – – – – – –
JSR *opr8a* JSR *opr16a* JSR *oprx0_xysppc* JSR *oprx9,xysppc* JSR *oprx16,xysppc* JSR [D,*xysppc*] JSR [*oprx16,xysppc*]	Jump to subroutine (SP)–2⇒SP RTN_H:RTN_L⇒M_{SP}:M_{SP+1} Subroutine address⇒PC	DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	17 dd 16 hh ll 15 xb 15 xb ff 15 xb ee ff 15 xb 15 xb ee ff	SPPP SPPP PPPS PPPS fPPPS fIfPPPS fIfPPPS	– – – – – – – –
LBCC *rel16*Same as LBHS	Long branch if C clear; if C=0, then (PC)+4+rel⇒PC	REL	18 24 qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBCS *rel16*Same as LBLO	Long branch if C set; if C=1, then (PC)+4+rel⇒PC	REL	18 25 qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBEQ *rel16*	Long branch if equal; if Z=1, then (PC)+4+rel⇒PC	REL	18 27 qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBGE *rel16*	Long branch if ≥ 0, signed If N⊕V=0, then (PC)+4+rel⇒PC	REL	18 2C qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBGT *rel16*	Long branch if > 0, signed If Z | (N⊕V)=0, then (PC)+4+rel⇒PC	REL	18 2E qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBHI *rel16*	Long branch if higher, unsigned If C | Z=0, then (PC)+4+rel⇒PC	REL	18 22 qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBHS *rel16*Same as LBCC	Long branch if higher or same, unsigned; If C=0, (PC)+4+rel⇒PC	REL	18 24 qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBLE *rel16*	Long branch if ≤ 0, signed; if Z | (N⊕V)=1, then (PC)+4+rel⇒PC	REL	18 2F qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBLO *rel16*Same as LBCS	Long branch if lower, unsigned; if C=1, then (PC)+4+rel⇒PC	REL	18 25 qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBLS *rel16*	Long branch if lower or same, unsigned; If C | Z=1, then (PC)+4+rel⇒PC	REL	18 23 qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBLT *rel16*	Long branch if < 0, signed If N⊕V=1, then (PC)+4+rel⇒PC	REL	18 2D qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBMI *rel16*	Long branch if minus If N=1, then (PC)+4+rel⇒PC	REL	18 2B qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBNE *rel16*	Long branch if not equal to 0 If Z=0, then (PC)+4+rel⇒PC	REL	18 26 qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBPL *rel16*	Long branch if plus If N=0, then (PC)+4+rel⇒PC	REL	18 2A qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBRA *rel16*	Long branch always	REL	18 20 qq rr	OPPP	– – – – – – – –
LBRN *rel16*	Long branch never	REL	18 21 qq rr	OPO	– – – – – – – –
LBVC *rel16*	Long branch if V clear If V=0,then (PC)+4+rel⇒PC	REL	18 28 qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –
LBVS *rel16*	Long branch if V set If V=1,then (PC)+4+rel⇒PC	REL	18 29 qq rr	OPPP (branch) OPO (no branch)	– – – – – – – –

Freescale Semiconductor, Inc.

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

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C

LDAA *#opr8i*
LDAA *opr8a*
LDAA *opr16a*
LDAA *oprx0_xysppc*
LDAA *oprx9,xysppc*
LDAA *oprx16,xysppc*
LDAA [D,*xysppc*]
LDAA [*oprx16,xysppc*]
	Load A
	(M)⇒A
	or imm⇒A
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			86 ii
			96 dd
			B6 hh ll
			A6 xb
			A6 xb ff
			A6 xb ee ff
			A6 xb
			A6 xb ee ff
				P
				rPf
				rPO
				rPf
				rPO
				frPP
				fIfrPf
				fIPrPf
					– – – – Δ Δ 0 –

LDAB *#opr8i*
LDAB *opr8a*
LDAB *opr16a*
LDAB *oprx0_xysppc*
LDAB *oprx9,xysppc*
LDAB *oprx16,xysppc*
LDAB [D,*xysppc*]
LDAB [*oprx16,xysppc*]
	Load B
	(M)⇒B
	or imm⇒B
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			C6 ii
			D6 dd
			F6 hh ll
			E6 xb
			E6 xb ff
			E6 xb ee ff
			E6 xb
			E6 xb ee ff
				P
				rPf
				rPO
				rPf
				rPO
				frPP
				fIfrPf
				fIPrPf
					– – – – Δ Δ 0 –

LDD *#opr16i*
LDD *opr8a*
LDD *opr16a*
LDD *oprx0_xysppc*
LDD *oprx9,xysppc*
LDD *oprx16,xysppc*
LDD [D,*xysppc*]
LDD [*oprx16,xysppc*]
	Load D
	(M:M+1)⇒A:B
	or imm⇒A:B
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			CC jj kk
			DC dd
			FC hh ll
			EC xb
			EC xb ff
			EC xb ee ff
			EC xb
			EC xb ee ff
				PO
				RPf
				RPO
				RPf
				RPO
				fRPP
				fIfRPf
				fIPRPf
					– – – – Δ Δ 0 –

LDS *#opr16i*
LDS *opr8a*
LDS *opr16a*
LDS *oprx0_xysppc*
LDS *oprx9,xysppc*
LDS *oprx16,xysppc*
LDS [D,*xysppc*]
LDS [*oprx16,xysppc*]
	Load SP
	(M:M+1)⇒SP
	or imm⇒SP
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			CF jj kk
			DF dd
			FF hh ll
			EF xb
			EF xb ff
			EF xb ee ff
			EF xb
			EF xb ee ff
				PO
				RPf
				RPO
				RPf
				RPO
				fRPP
				fIfRPf
				fIPRPf
					– – – – Δ Δ 0 –

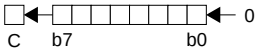
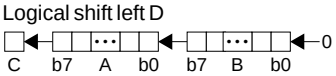
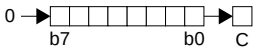
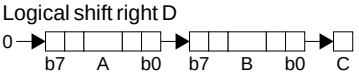
LDX *#opr16i*
LDX *opr8a*
LDX *opr16a*
LDX *oprx0_xysppc*
LDX *oprx9,xysppc*
LDX *oprx16,xysppc*
LDX [D,*xysppc*]
LDX [*oprx16,xysppc*]
	Load X
	(M:M+1)⇒X
	or imm⇒X
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			CE jj kk
			DE dd
			FE hh ll
			EE xb
			EE xb ff
			EE xb ee ff
			EE xb
			EE xb ee ff
				PO
				RPf
				RPO
				RPf
				RPO
				fRPP
				fIfRPf
				fIPRPf
					– – – – Δ Δ 0 –

LDY *#opr16i*
LDY *opr8a*
LDY *opr16a*
LDY *oprx0_xysppc*
LDY *oprx9,xysppc*
LDY *oprx16,xysppc*
LDY [D,*xysppc*]
LDY [*oprx16,xysppc*]
	Load Y
	(M:M+1)⇒Y
	or imm⇒Y
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			CD jj kk
			DD dd
			FD hh ll
			ED xb
			ED xb ff
			ED xb ee ff
			ED xb
			ED xb ee ff
				PO
				RPf
				RPO
				RPf
				RPO
				fRPP
				fIfRPf
				fIPRPf
					– – – – Δ Δ 0 –

LEAS *oprx0_xysppc* LEAS *oprx9,xysppc* LEAS *oprx16,xysppc*	Load effective address into SP EA⇒SP	IDX IDX1 IDX2	1B xb 1B xb ff 1B xb ee ff	Pf PO PP	– – – – – – – –
LEAX *oprx0_xysppc* LEAX *oprx9,xysppc* LEAX *oprx16,xysppc*	Load effective address into X EA⇒X	IDX IDX1 IDX2	1A xb 1A xb ff 1A xb ee ff	Pf PO PP	– – – – – – – –
LEAY *oprx0_xysppc* LEAY *oprx9,xysppc* LEAY *oprx16,xysppc*	Load effective address into Y EA⇒Y	IDX IDX1 IDX2	19 xb 19 xb ff 19 xb ee ff	Pf PO PP	– – – – – – – –

Freescale Semiconductor, Inc.

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
LSL *oprx16a*Same as ASL LSL *oprx0_xysppc* LSL *oprx9,xysppc* LSL *oprx16,xysppc* LSL [D,*xysppc*] LSL [*oprx16,xysppc*] LSLASame as ASLA LSLBSame as ASLB	Logical shift left M  Logical shift left A Logical shift left B	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	78 hh ll 68 xb 68 xb ff 68 xb ee ff 68 xb 68 xb ee ff 48 58	rOPw rPw rPOw frPPw fIfrPw fIPrPw O O	– – – – Δ Δ Δ Δ
LSLDSame as ASLD	Logical shift left D  	INH	59	O	– – – – Δ Δ Δ Δ
LSR *oprx16a* LSR *oprx0_xysppc* LSR *oprx9,xysppc* LSR *oprx16,xysppc* LSR [D,*xysppc*] LSR [*oprx16,xysppc*] LSRA LSRB	Logical shift right M  Logical shift right A Logical shift right B	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	74 hh ll 64 xb 64 xb ff 64 xb ee ff 64 xb 64 xb ee ff 44 54	rPwO rPw rPwO frPwP fIfrPw fIPrPw O O	– – – – 0 Δ Δ Δ
LSRD	Logical shift right D  	INH	49	O	– – – – 0 Δ Δ Δ
MAXA *oprx0_xysppc* MAXA *oprx9,xysppc* MAXA *oprx16,xysppc* MAXA [D,*xysppc*] MAXA [*oprx16,xysppc*]	Maximum in A; put larger of 2 unsigned 8-bit values in A MAX[(A), (M)]⇒A N, Z, V, C bits reflect result of internal compare [(A)–(M)]	IDX IDX1 IDX2 [D,IDX] [IDX2]	18 18 xb 18 18 xb ff 18 18 xb ee ff 18 18 xb 18 18 xb ee ff	OrPf OrPO OfrPP OfIfrPf OfIPrPf	– – – – Δ Δ Δ Δ
MAXM *oprx0_xysppc* MAXM *oprx9,xysppc* MAXM *oprx16,xysppc* MAXM [D,*xysppc*] MAXM [*oprx16,xysppc*]	Maximum in M; put larger of 2 unsigned 8-bit values in M MAX[(A), (M)]⇒M N, Z, V, C bits reflect result of internal compare [(A)–(M)]	IDX IDX1 IDX2 [D,IDX] [IDX2]	18 1C xb 18 1C xb ff 18 1C xb ee ff 18 1C xb 18 1C xb ee ff	OrPw OrPwO OfrPwP OfIfrPw OfIPrPw	– – – – Δ Δ Δ Δ
MEM	Determine grade of membership; μ (grade)⇒M_Y; (X)+4⇒X; (Y)+1⇒Y If (A)<P1 or (A)>P2, then μ=0; else μ=MIN[((A)–P1)×S1, (P2–(A))×S2, $FF] (A)=current crisp input value; X points at 4 data bytes (P1, P2, S1, S2) of a trapezoidal membership function; Y points at fuzzy input (RAM location)	Special	01	RRfOw	– – ? – ? ? ? ?
MINA *oprx0_xysppc* MINA *oprx9,xysppc* MINA *oprx16,xysppc* MINA [D,*xysppc*] MINA [*oprx16,xysppc*]	Minimum in A; put smaller of 2 unsigned 8-bit values in A MIN[(A), (M)]⇒A N, Z, V, C bits reflect result of internal compare [(A)–(M)]	IDX IDX1 IDX2 [D,IDX] [IDX2]	18 19 xb 18 19 xb ff 18 19 xb ee ff 18 19 xb 18 19 xb ee ff	OrPf OrPO OfrPP OfIfrPf OfIPrPf	– – – – Δ Δ Δ Δ
MINM *oprx0_xysppc* MINM *oprx9,xysppc* MINM *oprx16,xysppc* MINM [D,*xysppc*] MINM [*oprx16,xysppc*]	Minimum in N; put smaller of two unsigned 8-bit values in M MIN[(A), (M)]⇒M N, Z, V, C bits reflect result of internal compare [(A)–(M)]	IDX IDX1 IDX2 [D,IDX] [IDX2]	18 1D xb 18 1D xb ff 18 1D xb ee ff 18 1D xb 18 1D xb ee ff	OrPw OrPwO OfrPwP OfIfrPw OfIPrPw	– – – – Δ Δ Δ Δ

MOVB #*opr8*, *opr16a*
MOVB #*opr8i*, *oprx0_xysppc*
MOVB *opr16a*, *opr16a*
MOVB *opr16a*, *oprx0_xysppc*
MOVB *oprx0_xysppc*, *opr16a*
MOVB *oprx0_xysppc*, *oprx0_xysppc*
	Move byte
	Memory-to-memory 8-bit byte-move (M_1)⇒M_2
	First operand specifies byte to move
		IMM-EXT
		IMM-IDX
		EXT-EXT
		EXT-IDX
		IDX-EXT
		IDX-IDX
			18 0B ii hh ll
			18 08 xb ii
			18 0C hh ll hh ll
			18 09 xb hh ll
			18 0D xb hh ll
			18 0A xb xb
				OPwP
				OPwO
				OrPwPO
				OPrPw
				OrPwP
				OrPwO
					– – – – – – – –

Freescale Semiconductor, Inc.

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

Table 4 Instruction Set Summary (Continued)

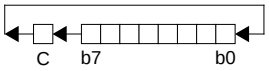
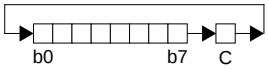
Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C

MOVW *#oprx16*, *opr16a*
MOVW *#opr16i*, *oprx0_xysppc*
MOVW *opr16a*, *opr16a*
MOVW *opr16a*, *oprx0_xysppc*
MOVW *oprx0_xysppc*, *opr16a*
MOVW *oprx0_xysppc*, *oprx0_xysppc*
	Move word
	Memory-to-memory 16-bit word-move
	$(M_1:M_1+1) \Rightarrow M_2:M_2+1$
	First operand specifies word to move
		IMM-EXT
		IMM-IDX
		EXT-EXT
		EXT-IDX
		IDX-EXT
		IDX-IDX
			18 03 jj kk hh ll
			18 00 xb jj kk
			18 04 hh ll hh ll
			18 01 xb hh ll
			18 05 xb hh ll
			18 02 xb xb
				OPWPO
				OPPW
				ORPWPO
				OPRPW
				ORPWP
				ORPWO
					– – – – – – – –

MUL	Multiply, unsigned (A)×(B)⇒A:B; 8 by 8-bit	INH	12	O	– – – – – – – Δ
NEG *opr16a* NEG *oprx0_xysppc* NEG *oprx9,xysppc* NEG *oprx16,xysppc* NEG [D,*xysppc*] NEG [*oprx16,xysppc*] NEGA NEGB	Negate M; 0–(M)⇒M or $(\overline{M})+1 \Rightarrow M$ Negate A; 0–(A)⇒A or $(\overline{A})+1 \Rightarrow A$ Negate B; 0–(B)⇒B or $(\overline{B})+1 \Rightarrow B$	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	70 hh ll 60 xb 60 xb ff 60 xb ee ff 60 xb 60 xb ee ff 40 50	rPwO rPw rPwO frPwP fIfrPw fIPrPw O O	– – – – Δ Δ Δ Δ
NOP	No operation	INH	A7	O	– – – – – – – –
ORAA *#opr8i* ORAA *opr8a* ORAA *opr16a* ORAA *oprx0_xysppc* ORAA *oprx9,xysppc* ORAA *oprx16,xysppc* ORAA [D,*xysppc*] ORAA [*oprx16,xysppc*]	OR accumulator A (A) | (M)⇒A or (A) | imm⇒A	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	8A ii 9A dd BA hh ll AA xb AA xb ff AA xb ee ff AA xb AA xb ee ff	P rPf rPO rPf rPO frPP fIfrPf fIPrPf	– – – – Δ Δ 0 –
ORAB *#opr8i* ORAB *opr8a* ORAB *opr16a* ORAB *oprx0_xysppc* ORAB *oprx9,xysppc* ORAB *oprx16,xysppc* ORAB [D,*xysppc*] ORAB [*oprx16,xysppc*]	OR accumulator B (B) | (M)⇒B or (B) | imm⇒B	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	CA ii DA dd FA hh ll EA xb EA xb ff EA xb ee ff EA xb EA xb ee ff	P rPf rPO rPf rPO frPP fIfrPf fIPrPf	– – – – Δ Δ 0 –
ORCC *#opr8i*	OR CCR; (CCR) | imm⇒CCR	IMM	14 ii	P	⇑ – ⇑ ⇑ ⇑ ⇑ ⇑ ⇑
PSHA	Push A; (SP)–1⇒SP; (A)⇒M_{SP}	INH	36	Os	– – – – – – – –
PSHB	Push B; (SP)–1⇒SP; (B)⇒M_{SP}	INH	37	Os	– – – – – – – –
PSHC	Push CCR; (SP)–1⇒SP; (CCR)⇒M_{SP}	INH	39	Os	– – – – – – – –
PSHD	Push D (SP)–2⇒SP; (A:B)⇒M_{SP}:M_{SP+1}	INH	3B	OS	– – – – – – – –
PSHX	Push X (SP)–2⇒SP; (X_H:X_L)⇒M_{SP}:M_{SP+1}	INH	34	OS	– – – – – – – –
PSHY	Push Y (SP)–2⇒SP; (Y_H:Y_L)⇒M_{SP}:M_{SP+1}	INH	35	OS	– – – – – – – –
PULA	Pull A (M_{SP})⇒A; (SP)+1⇒SP	INH	32	ufO	– – – – – – – –
PULB	Pull B (M_{SP})⇒B; (SP)+1⇒SP	INH	33	ufO	– – – – – – – –
PULC	Pull CCR (M_{SP})⇒CCR; (SP)+1⇒SP	INH	38	ufO	Δ ⇓ Δ Δ Δ Δ Δ
PULD	Pull D (M_{SP}:M_{SP+1})⇒A:B; (SP)+2⇒SP	INH	3A	UfO	– – – – – – – –

Freescale Semiconductor, Inc.

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
PULX	Pull X $(M_{SP}:M_{SP+1})\Rightarrow X_H:X_L$; (SP)+2⇒SP	INH	30	UfO	– – – – – – – –
PULY	Pull Y $(M_{SP}:M_{SP+1})\Rightarrow Y_H:Y_L$; (SP)+2⇒SP	INH	31	UfO	– – – – – – – –
REV	Rule evaluation, unweighted; find smallest rule input; store to rule outputs unless fuzzy output is larger	Special	18 3A	Orf(t^tx)O* ff+Orft^**	– – ? – ? ? Δ ?
*The t^tx loop is executed once for each element in the rule list. The ^ denotes a check for pending interrupt requests. **These are additional cycles caused by an interrupt: ff is the exit sequence and Orft^ is the re-entry sequence.					
REVW	Rule evaluation, weighted; rule weights optional; find smallest rule input; store to rule outputs unless fuzzy output is larger	Special	18 3B	ORf(t^Tx)O* or ORf(r^ffRf)O** ffff+ORft^*** ffff+ORfr^****	– – ? – ? ? Δ !
*With weighting not enabled, the t^Tx loop is executed once for each element in the rule list. The ^ denotes a check for pending interrupt requests. **With weighting enabled, the t^Tx loop is replaced by r^ffRf. ***Additional cycles caused by an interrupt when weighting is not enabled: ffff is the exit sequence and ORft^ is the re-entry sequence. **** Additional cycles caused by an interrupt when weighting is enabled: ffff is the exit sequence and ORfr^ is the re-entry sequence.					
ROL *opr16a* ROL *oprx0_xysppc* ROL *oprx9*,*xysppc* ROL *oprx16*,*xysppc* ROL [D,*xysppc*] ROL [*oprx16*,*xysppc*] ROLA ROLB	Rotate left M  Rotate left A Rotate left B	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	75 hh ll 65 xb 65 xb ff 65 xb ee ff 65 xb 65 xb ee ff 45 55	rPwO rPw rPwO frPwP fIfrPw fIPrPw O O	– – – – Δ Δ Δ Δ
ROR *opr16a* ROR *oprx0_xysppc* ROR *oprx9*,*xysppc* ROR *oprx16*,*xysppc* ROR [D,*xysppc*] ROR [*oprx16*,*xysppc*] RORA RORB	Rotate right M  Rotate right A Rotate right B	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	76 hh ll 66 xb 66 xb ff 66 xb ee ff 66 xb 66 xb ee ff 46 56	rPwO rPw rPwO frPwP fIfrPw fIPrPw O O	– – – – Δ Δ Δ Δ
RTC	Return from call; $(M_{SP})\Rightarrow$PPAGE (SP)+1⇒SP; $(M_{SP}:M_{SP+1})\Rightarrow PC_H:PC_L$ (SP)+2⇒SP	INH	0A	uUnfPPP	– – – – – – – –
RTI	Return from interrupt $(M_{SP})\Rightarrow$CCR; (SP)+1⇒SP $(M_{SP}:M_{SP+1})\Rightarrow$B:A;(SP)+2⇒SP $(M_{SP}:M_{SP+1})\Rightarrow X_H:X_L$;(SP)+4⇒SP $(M_{SP}:M_{SP+1})\Rightarrow PC_H:PC_L$;(SP)+2⇒SP $(M_{SP}:M_{SP+1})\Rightarrow Y_H:Y_L$;(SP)+4⇒SP	INH	0B	uUUUUPPP or uUUUUfVfPPP*	Δ ⇓ Δ Δ Δ Δ Δ Δ
*RTI takes 11 cycles if an interrupt is pending.					
RTS	Return from subroutine $(M_{SP}:M_{SP+1})\Rightarrow PC_H:PC_L$; (SP)+2⇒SP	INH	3D	UfPPP	– – – – – – – –
SBA	Subtract B from A; (A)–(B)⇒A	INH	18 16	OO	– – – – Δ Δ Δ Δ

Freescale Semiconductor, Inc.

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
SBCA *#opr8i* SBCA *opr8a* SBCA *opr16a* SBCA *oprx0_xysppc* SBCA *oprx9,xysppc* SBCA *oprx16,xysppc* SBCA [D,*xysppc*] SBCA [*oprx16,xysppc*]	Subtract with carry from A (A)–(M)–C⇒A or (A)–imm–C⇒A	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	82 ii 92 dd B2 hh ll A2 xb A2 xb ff A2 xb ee ff A2 xb A2 xb ee ff	P rPf rPO rPf rPO frPP fIfrPf fIPrPf	– – – – Δ Δ Δ Δ
SBCB *#opr8i* SBCB *opr8a* SBCB *opr16a* SBCB *oprx0_xysppc* SBCB *oprx9,xysppc* SBCB *oprx16,xysppc* SBCB [D,*xysppc*] SBCB [*oprx16,xysppc*]	Subtract with carry from B (B)–(M)–C⇒B or (B)–imm–C⇒B	IMM DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	C2 ii D2 dd F2 hh ll E2 xb E2 xb ff E2 xb ee ff E2 xb E2 xb ee ff	P rPf rPO rPf rPO frPP fIfrPf fIPrPf	– – – – Δ Δ Δ Δ
SECSame as ORCC #$01	Set C bit	IMM	14 01	P	– – – – – – – 1
SEISame as ORCC #$10	Set I bit	IMM	14 10	P	– – – 1 – – – –
SEVSame as ORCC #$02	Set V bit	IMM	14 02	P	– – – – – – 1 –
SEX *abc*,*dxysp*Same as TFR r1, r2	Sign extend; 8-bit r1 to 16-bit r2 $00:(r1)⇒r2 if bit 7 of r1 is 0 $FF:(r1)⇒r2 if bit 7 of r1 is 1	INH	B7 eb	P	– – – – – – – –
STAA *opr8a* STAA *opr16a* STAA *oprx0_xysppc* STAA *oprx9,xysppc* STAA *oprx16,xysppc* STAA [D,*xysppc*] STAA [*oprx16,xysppc*]	Store accumulator A (A)⇒M	DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	5A dd 7A hh ll 6A xb 6A xb ff 6A xb ee ff 6A xb 6A xb ee ff	Pw PwO Pw PwO PwP PIfw PIPw	– – – – Δ Δ 0 –
STAB *opr8a* STAB *opr16a* STAB *oprx0_xysppc* STAB *oprx9,xysppc* STAB *oprx16,xysppc* STAB [D,*xysppc*] STAB [*oprx16,xysppc*]	Store accumulator B (B)⇒M	DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	5B dd 7B hh ll 6B xb 6B xb ff 6B xb ee ff 6B xb 6B xb ee ff	Pw PwO Pw PwO PwP PIfw PIPw	– – – – Δ Δ 0 –
STD *opr8a* STD *opr16a* STD *oprx0_xysppc* STD *oprx9,xysppc* STD *oprx16,xysppc* STD [D,*xysppc*] STD [*oprx16,xysppc*]	Store D (A:B)⇒M:M+1	DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	5C dd 7C hh ll 6C xb 6C xb ff 6C xb ee ff 6C xb 6C xb ee ff	PW PWO PW PWO PWP PIfW PIPW	– – – – Δ Δ 0 –
STOP	Stop processing; (SP)–2⇒SP RTN_H:RTN_L⇒M_{SP}:M_{SP+1} (SP)–2⇒SP; (Y_H:Y_L)⇒M_{SP}:M_{SP+1} (SP)–2⇒SP; (X_H:X_L)⇒M_{SP}:M_{SP+1} (SP)–2⇒SP; (B:A)⇒M_{SP}:M_{SP+1} (SP)–1⇒SP; (CCR)⇒M_{SP} Stop all clocks	INH	18 3E	00SSSSsf (enter stop mode) fVfPPP (exit stop mode) ff (continue stop mode) 00 (if stop mode disabled by S=1)	– – – – – – – –

Freescale Semiconductor, Inc.

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
STS *opr8a* STS *opr16a* STS *oprx0_xysppc* STS *oprx9,xysppc* STS *oprx16,xysppc* STS [D,*xysppc*] STS [*oprx16,xysppc*]	Store SP $(SP_H:SP_L) \Rightarrow M:M+1$	DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	5F dd 7F hh ll 6F xb 6F xb ff 6F xb ee ff 6F xb 6F xb ee ff	PW PWO PW PWO PWP PIfW PIPW	– – – – Δ Δ 0 –
STX *opr8a* STX *opr16a* STX *oprx0_xysppc* STX *oprx9,xysppc* STX *oprx16,xysppc* STX [D,*xysppc*] STX [*oprx16,xysppc*]	Store X $(X_H:X_L) \Rightarrow M:M+1$	DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	5E dd 7E hh ll 6E xb 6E xb ff 6E xb ee ff 6E xb 6E xb ee ff	PW PWO PW PWO PWP PIfW PIPW	– – – – Δ Δ 0 –
STY *opr8a* STY *opr16a* STY *oprx0_xysppc* STY *oprx9,xysppc* STY *oprx16,xysppc* STY [D,*xysppc*] STY [*oprx16,xysppc*]	Store Y $(Y_H:Y_L) \Rightarrow M:M+1$	DIR EXT IDX IDX1 IDX2 [D,IDX] [IDX2]	5D dd 7D hh ll 6D xb 6D xb ff 6D xb ee ff 6D xb 6D xb ee ff	PW PWO PW PWO PWP PIfW PIPW	– – – – Δ Δ 0 –

SUBA #*opr8i*
SUBA *opr8a*
SUBA *opr16a*
SUBA *oprx0_xysppc*
SUBA *oprx9,xysppc*
SUBA *oprx16,xysppc*
SUBA [D,*xysppc*]
SUBA [*oprx16,xysppc*]
	Subtract from A
	(A)–(M)⇒A
	or (A)–imm⇒A
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			80 ii
			90 dd
			B0 hh ll
			A0 xb
			A0 xb ff
			A0 xb ee ff
			A0 xb
			A0 xb ee ff
				P
				rPf
				rPO
				rPf
				rPO
				frPP
				fIfrPf
				fIPrPf
					– – – – Δ Δ Δ Δ

SUBB #*opr8i*
SUBB *opr8a*
SUBB *opr16a*
SUBB *oprx0_xysppc*
SUBB *oprx9,xysppc*
SUBB *oprx16,xysppc*
SUBB [D,*xysppc*]
SUBB [*oprx16,xysppc*]
	Subtract from B
	(B)–(M)⇒B
	or (B)–imm⇒B
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			C0 ii
			D0 dd
			F0 hh ll
			E0 xb
			E0 xb ff
			E0 xb ee ff
			E0 xb
			E0 xb ee ff
				P
				rPf
				rPO
				rPf
				rPO
				frPP
				fIfrPf
				fIPrPf
					– – – – Δ Δ Δ Δ

SUBD #*opr16i*
SUBD *opr8a*
SUBD *opr16a*
SUBD *oprx0_xysppc*
SUBD *oprx9,xysppc*
SUBD *oprx16,xysppc*
SUBD [D,*xysppc*]
SUBD [*oprx16,xysppc*]
	Subtract from D
	(A:B)–(M:M+1)⇒A:B
	or (A:B)–imm⇒A:B
		IMM
		DIR
		EXT
		IDX
		IDX1
		IDX2
		[D,IDX]
		[IDX2]
			83 jj kk
			93 dd
			B3 hh ll
			A3 xb
			A3 xb ff
			A3 xb ee ff
			A3 xb
			A3 xb ee ff
				PO
				RPf
				RPO
				RPf
				RPO
				fRPP
				fIfRPf
				fIPRPf
					– – – – Δ Δ Δ Δ

SWI	Software interrupt; (SP)–2⇒SP $RTN_H:RTN_L \Rightarrow M_{SP}:M_{SP+1}$ (SP)–2⇒SP; $(Y_H:Y_L) \Rightarrow M_{SP}:M_{SP+1}$ (SP)–2⇒SP; $(X_H:X_L) \Rightarrow M_{SP}:M_{SP+1}$ (SP)–2⇒SP; (B:A)⇒$M_{SP}:M_{SP+1}$ (SP)–1⇒SP; (CCR)⇒M_{SP};1⇒I (SWI vector)⇒PC	INH	3F	VSPSSPSsP*	– – – 1 – – – –
*The CPU also uses VSPSSPSsP for hardware interrupts and unimplemented opcode traps.					
TAB	Transfer A to B; (A)⇒B	INH	18 0E	00	– – – – Δ Δ 0 –
TAP	Transfer A to CCR; (A)⇒CCR Assembled as TFR A, CCR	INH	B7 02	P	Δ ⇓ Δ Δ Δ Δ Δ
TBA	Transfer B to A; (B)⇒A	INH	18 0F	00	– – – – Δ Δ 0 –

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

Freescale Semiconductor, Inc.

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
TBEQ *abdxysp,rel9*	Test and branch if equal to 0 If (counter)=0, then (PC)+2+rel⇒PC	REL (9-bit)	04 lb rr	PPP (branch) PPO (no branch)	– – – – – – – –
TBL *oprx0_xysppc*	Table lookup and interpolate, 8-bit (M)+[(B)×((M+1)–(M))]⇒A	IDX	18 3D xb	ORfffP	– – – – Δ Δ – Δ
TBNE *abdxysp,rel9*	Test and branch if not equal to 0 If (counter)≠0, then (PC)+2+rel⇒PC	REL (9-bit)	04 lb rr	PPP (branch) PPO (no branch)	– – – – – – – –
TFR *abcdxysp,abcdxysp*	Transfer from register to register (r1)⇒r2r1 and r2 same size $00:(r1)⇒r2r1=8-bit; r2=16-bit ($r1_L$)⇒r2r1=16-bit; r2=8-bit	INH	B7 eb	P	– – – – – – – – or Δ ⇓ Δ Δ Δ Δ Δ
TPASame as TFR CCR,A	Transfer CCR to A; (CCR)⇒A	INH	B7 20	P	– – – – – – – –
TRAP *trapnum*	Trap unimplemented opcode; (SP)–2⇒SP RTN_H:RTN_L⇒M_{SP}:M_{SP+1} (SP)–2⇒SP; (Y_H:Y_L)⇒M_{SP}:M_{SP+1} (SP)–2⇒SP; (X_H:X_L)⇒M_{SP}:M_{SP+1} (SP)–2⇒SP; (B:A)⇒M_{SP}:M_{SP+1} (SP)–1⇒SP; (CCR)⇒M_{SP} 1⇒I; (trap vector)⇒PC	INH	18 tn tn = $30–$39 or tn = $40–$FF	OVSPSSPSsP	– – – 1 – – – –
TST *opr16a* TST *oprx0_xysppc* TST *oprx9,xysppc* TST *oprx16,xysppc* TST [D,*xysppc*] TST [*oprx16,xysppc*] TSTA TSTB	Test M; (M)–0 Test A; (A)–0 Test B; (B)–0	EXT IDX IDX1 IDX2 [D,IDX] [IDX2] INH INH	F7 hh ll E7 xb E7 xb ff E7 xb ee ff E7 xb E7 xb ee ff 97 D7	rPO rPf rPO frPP fIfrPf fIPrPf O O	– – – – Δ Δ 0 0
TSXSame as TFR SP,X	Transfer SP to X; (SP)⇒X	INH	B7 75	P	– – – – – – – –
TSYSame as TFR SP,Y	Transfer SP to Y; (SP)⇒Y	INH	B7 76	P	– – – – – – – –
TXSSame as TFR X,SP	Transfer X to SP; (X)⇒SP	INH	B7 57	P	– – – – – – – –
TYSSame as TFR Y,SP	Transfer Y to SP; (Y)⇒SP	INH	B7 67	P	– – – – – – – –
WAI	Wait for interrupt; (SP)–2⇒SP RTN_H:RTN_L⇒M_{SP}:M_{SP+1} (SP)–2⇒SP; (Y_H:Y_L)⇒M_{SP}:M_{SP+1} (SP)–2⇒SP; (X_H:X_L)⇒M_{SP}:M_{SP+1} (SP)–2⇒SP; (B:A)⇒M_{SP}:M_{SP+1} (SP)–1⇒SP; (CCR)⇒M_{SP}	INH	3E	OSSSSsf (before interrupt) fVfPPP (after interrupt)	– – – – – – – – or – – – 1 – – – – or – 1 – 1 – – – –
WAV	$\sum_{i=1}^{B} S_i F_i \Rightarrow Y{:}D$ $\sum_{i=1}^{B} F_i \Rightarrow X$ Calculate weighted average; sum of products (SOP) and sum of weights (SOW)*	Special	18 3C	Of(frr^ffff)O** SSS+UUUrr^***	– – ? – ? Δ ? ?
*Initialize B, X, and Y: B=number of elements; X points at first element in S_i list; Y points at first element in F_i list. All S_i and F_i elements are 8-bit values. **The frr^ffff sequence is the loop for one iteration of SOP and SOW accumulation. The ^ denotes a check for pending interrupt requests. ***Additional cycles caused by an interrupt: SSS is the exit sequence and UUUrr^ is the re-entry sequence. Intermediate values use six stack bytes.					
wavr*	Resume executing interrupted WAV	Special	3C	UUUrr^ffff(frr^ffff)O** SSS+UUUrr^***	– – ? – ? Δ ? ?

Freescale Semiconductor, Inc.

Table 4 Instruction Set Summary (Continued)

Source Form	Operation	Address Mode	Machine Coding (Hex)	Access Detail	S X H I N Z V C
*wavr is a pseudoinstruction that recovers intermediate results from the stack rather than initializing them to 0. **The `frr^ffff` sequence is the loop for one iteration of SOP and SOW recovery. The ^ denotes a check for pending interrupt requests. ***These are additional cycles caused by an interrupt: SSS is the exit sequence and `UUUrr^` is the re-entry sequence.					
XGDXSame as EXG D, X	Exchange D with X; (D)⇔(X)	INH	B7 C5	P	– – – – – – – –
XGDYSame as EXG D, Y	Exchange D with Y; (D)⇔(Y)	INH	B7 C6	P	– – – – – – – –

Register and Memory Notation

Table 5 Register and Memory Notation

A or *a*	Accumulator A
An	Bit n of accumulator A
B or *b*	Accumulator B
Bn	Bit n of accumulator B
D or *d*	Accumulator D
Dn	Bit n of accumulator D
X or *x*	Index register X
X_H	High byte of index register X
X_L	Low byte of index register X
Xn	Bit n of index register X
Y or *y*	Index register Y
Y_H	High byte of index register Y
Y_L	Low byte of index register Y
Yn	Bit n of index register Y
SP or *sp*	Stack pointer
SPn	Bit n of stack pointer
PC or *pc*	Program counter
PC_H	High byte of program counter
PC_L	Low byte of program counter
CCR or *c*	Condition code register
M	Address of 8-bit memory location
Mn	Bit n of byte at memory location M
Rn	Bit n of the result of an arithmetic or logical operation
In	Bit n of the intermediate result of an arithmetic or logical operation
RTN_H	High byte of return address
RTN_L	Low byte of return address
( )	Contents of

Freescale Semiconductor, Inc.

Source Form Notation

The **Source Form** column of the summary in Table 4 gives essential information about assembler source forms. For complete information about writing source files for a particular assembler, refer to the documentation provided by the assembler vendor.

Everything in the **Source Form** column, *except expressions in italic characters*, is literal information which must appear in the assembly source file exactly as shown. The initial 3- to 5-letter mnemonic is always a literal expression. All commas, pound signs (#), parentheses, square brackets ([or]), plus signs (+), minus signs (–), and the register designation (A, B, D), are literal characters.

The groups of italic characters shown in Table 6 represent variable information to be supplied by the programmer. These groups can include any alphanumeric character or the underscore character, but cannot include a space or comma. For example, the groups *xysppc* and *oprx0_xysppc* are both valid, but the two groups *oprx0 xysppc* are not valid because there is a space between them.

Freescale Semiconductor, Inc.

Table 6 Source Form Notation

abc	Register designator for A, B, or CCR
abcdxysp	Register designator for A, B, CCR, D, X, Y, or SP
abd	Register designator for A, B, or D
abdxysp	Register designator for A, B, D, X, Y, or SP
dxysp	Register designator for D, X, Y, or SP
msk8	8-bit mask value Some assemblers require the # symbol before the mask value.
opr8i	8-bit immediate value
opr16i	16-bit immediate value
opr8a	8-bit address value used with direct address mode
opr16a	16-bit address value
oprx0_xysp	Indexed addressing postbyte code: *oprx3*,–*xysp* — Predecrement X , Y, or SP by 1–8 *oprx3*,+*xysp* — Preincrement X , Y, or SP by 1–8 *oprx3*,*xysp*– — Postdecrement X, Y, or SP by 1–8 *oprx3*,*xysp*+ — Postincrement X, Y, or SP by 1–8 *oprx5*,*xysppc* — 5-bit constant offset from X, Y, SP, or PC *abd*,*xysppc* — Accumulator A, B, or D offset from X, Y, SP, or PC
oprx3	Any positive integer from 1 to 8 for pre/post increment/decrement
oprx5	Any integer from –16 to +15
oprx9	Any integer from –256 to +255
oprx16	Any integer from –32,768 to +65,535
page	8-bit value for PPAGE register Some assemblers require the # symbol before this value.
rel8	Label of branch destination within –256 to +255 locations
rel9	Label of branch destination within –512 to +511 locations
rel16	Any label within the 64-Kbyte memory space
trapnum	Any 8-bit integer from $30 to $39 or from $40 to $FF
xysp	Register designator for X or Y or SP
xysppc	Register designator for X or Y or SP or PC

Freescale Semiconductor, Inc.

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

Operation Notation

Table 7 Operation Notation

+	Add
–	Subtract
•	AND
|	OR
⊕	Exclusive OR
×	Multiply
÷	Divide
:	Concatenate
⇒	Transfer
⇔	Exchange

Address Mode Notation

Table 8 Address Mode Notation

INH	Inherent; no operands in instruction stream
IMM	Immediate; operand immediate value in instruction stream
DIR	Direct; operand is lower byte of address from $0000 to $00FF
EXT	Operand is a 16-bit address
REL	Two's complement relative offset; for branch instructions
IDX	Indexed (no extension bytes); includes: 5-bit constant offset from X, Y, SP or PC Pre/post increment/decrement by 1–8 Accumulator A, B, or D offset
IDX1	9-bit signed offset from X, Y, SP, or PC; 1 extension byte
IDX2	16-bit signed offset from X, Y, SP, or PC; 2 extension bytes
[IDX2]	Indexed-indirect; 16-bit offset from X, Y, SP, or PC
[D, IDX]	Indexed-indirect; accumulator D offset from X, Y, SP, or PC

Machine Code Notation

In the **Machine Code (Hex)** column of the summary in Table 4, digits 0–9 and upper case letters A–F represent hexadecimal values. Pairs of lower-case letters represent 8-bit values as shown in Table 9.

Freescale Semiconductor, Inc.

Table 9 Machine Code Notation

dd	8-bit direct address from $0000 to $00FF; high byte is $00
ee	High byte of a 16-bit constant offset for indexed addressing
eb	Exchange/transfer postbyte
ff	Low eight bits of a 9-bit signed constant offset in indexed addressing, or low byte of a 16-bit constant offset in indexed addressing
hh	High byte of a 16-bit extended address
ii	8-bit immediate data value
jj	High byte of a 16-bit immediate data value
kk	Low byte of a 16-bit immediate data value
lb	Loop primitive (DBNE) postbyte
ll	Low byte of a 16-bit extended address
mm	8-bit immediate mask value for bit manipulation instructions; bits that are set indicate bits to be affected
pg	Program page or bank number used in CALL instruction
qq	High byte of a 16-bit relative offset for long branches
tn	Trap number from $30 to $39 or from $40 to $FF
rr	Signed relative offset $80 (–128) to $7F (+127) relative to the byte following the relative offset byte, or low byte of a 16-bit relative offset for long branches
xb	Indexed addressing postbyte

Access Detail Notation

A single-letter code in the **Access Detail** column of Figure 4 represents a single CPU access cycle. An upper-case letter indicates a 16-bit access.

Table 10 Access Detail Notation

f	Free cycle. During an f cycle, the CPU does not use the bus. An f cycle is always one cycle of the system bus clock. An f cycle can be used by a queue controller or the background debug system to perform a single-cycle access without disturbing the CPU.
g	Read PPAGE register. A g cycle is used only in CALL instructions and is not visible on the external bus. Since PPAGE is an internal 8-bit register, a g cycle is never stretched.
I	Read indirect pointer. Indexed-indirect instructions use the 16-bit indirect pointer from memory to address the instruction operand. An I cycle is a 16-bit read that can be aligned or misaligned. An I cycle is extended to two bus cycles if the MCU is operating with an 8-bit external data bus and the corresponding data is stored in external memory. There can be additional stretching when the address space is assigned to a chip-select circuit programmed for slow memory. An I cycle is also stretched if it corresponds to a misaligned access to a memory that is not designed for single-cycle misaligned access.
i	Read indirect PPAGE value. An i cycle is used only in indexed-indirect CALL instructions. The 8-bit PPAGE value for the CALL destination is fetched from an indirect memory location. An i cycle is stretched only when controlled by a chip-select circuit that is programmed for slow memory.

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

Freescale Semiconductor, Inc.

Table 10 Access Detail Notation (Continued)

n	Write PPAGE register. An n cycle is used only in CALL and RTC instructions to write the destination value of the PPAGE register and is not visible on the external bus. Since the PPAGE register is an internal 8-bit register, an n cycle is never stretched.
O	Optional cycle. An O cycle adjusts instruction alignment in the instruction queue. An O cycle can be a free cycle (f) or a program word access cycle (P). When the first byte of an instruction with an odd number of bytes is misaligned, the O cycle becomes a P cycle to maintain queue order. If the first byte is aligned, the O cycle is an f cycle. The $18 prebyte for a page-two opcode is treated as a special one-byte instruction. If the prebyte is misaligned, the O cycle at the beginning of the instruction becomes a P cycle to maintain queue order. If the prebyte is aligned, the O cycle is an f cycle. If the instruction has an odd number of bytes, it has a second O cycle at the end. If the first O cycle is a P cycle (prebyte misaligned), the second O cycle is an f cycle. If the first O cycle is an f cycle (prebyte aligned), the second O cycle is a P cycle. An O cycle that becomes a P cycle can be extended to two bus cycles if the MCU is operating with an 8-bit external data bus and the program is stored in external memory. There can be additional stretching when the address space is assigned to a chip-select circuit programmed for slow memory. An O cycle that becomes an f cycle is never stretched.
P	Program word access. Program information is fetched as aligned 16-bit words. A P cycle is extended to two bus cycles if the MCU is operating with an 8-bit external data bus and the program is stored externally. There can be additional stretching when the address space is assigned to a chip-select circuit programmed for slow memory.
r	8-bit data read. An r cycle is stretched only when controlled by a chip-select circuit programmed for slow memory.
R	16-bit data read. An R cycle is extended to two bus cycles if the MCU is operating with an 8-bit external data bus and the corresponding data is stored in external memory. There can be additional stretching when the address space is assigned to a chip-select circuit programmed for slow memory. An R cycle is also stretched if it corresponds to a misaligned access to a memory that is not designed for single-cycle misaligned access.
s	Stack 8-bit data. An s cycle is stretched only when controlled by a chip-select circuit programmed for slow memory.
S	Stack 16-bit data. An S cycle is extended to two bus cycles if the MCU is operating with an 8-bit external data bus and the SP is pointing to external memory. There can be additional stretching if the address space is assigned to a chip-select circuit programmed for slow memory. An S cycle is also stretched if it corresponds to a misaligned access to a memory that is not designed for single-cycle misaligned access. The internal RAM is designed to allow single cycle misaligned word access.
w	8-bit data write. A w cycle is stretched only when controlled by a chip-select circuit programmed for slow memory.
W	16-bit data write. A W cycle is extended to two bus cycles if the MCU is operating with an 8-bit external data bus and the corresponding data is stored in external memory. There can be additional stretching when the address space is assigned to a chip-select circuit programmed for slow memory. A W cycle is also stretched if it corresponds to a misaligned access to a memory that is not designed for single-cycle misaligned access.
u	Unstack 8-bit data. A W cycle is stretched only when controlled by a chip-select circuit programmed for slow memory.

Freescale Semiconductor, Inc.

Table 10 Access Detail Notation (Continued)

U	Unstack 16-bit data. A U cycle is extended to two bus cycles if the MCU is operating with an 8-bit external data bus and the SP is pointing to external memory. There can be additional stretching when the address space is assigned to a chip-select circuit programmed for slow memory. A U cycle is also stretched if it corresponds to a misaligned access to a memory that is not designed for single-cycle misaligned access. The internal RAM is designed to allow single-cycle misaligned word access.
V	16-bit vector fetch. Vectors are always aligned 16-bit words. A V cycle is extended to two bus cycles if the MCU is operating with an 8-bit external data bus and the program is stored in external memory. There can be additional stretching when the address space is assigned to a chip-select circuit programmed for slow memory.
t	8-bit conditional read. A t cycle is either a data read cycle or a free cycle, depending on the data and flow of the REVW instruction. A t cycle is stretched only when controlled by a chip-select circuit programmed for slow memory.
T	16-bit conditional read. A T cycle is either a data read cycle or a free cycle, depending on the data and flow of the REV or REVW instruction. A T cycle is extended to two bus cycles if the MCU is operating with an 8-bit external data bus and the corresponding data is stored in external memory. There can be additional stretching when the address space is assigned to a chip-select circuit programmed for slow memory. A T cycle is also stretched if it corresponds to a misaligned access to a memory that is not designed for single-cycle misaligned access.
x	8-bit conditional write. An x cycle is either a data write cycle or a free cycle, depending on the data and flow of the REV or REVW instruction. An x cycle is stretched only when controlled by a chip-select circuit programmed for slow memory.
Special Notation for Branch Taken/Not Taken	
PPP/P	A short branch requires three cycles if taken, one cycle if not taken. Since the instruction consists of a single word containing both an opcode and an 8-bit offset, the not-taken case is simple — the queue advances, another program word fetch is made, and execution continues with the next instruction. The taken case requires that the queue be refilled so that execution can continue at a new address. First, the effective address of the destination is determined, then the CPU performs three program word fetches from that address.
OPPP/OPO	A long branch requires four cycles if taken, three cycles if not taken. An O cycle is required because all long branches are page two opcodes and thus include the $18 prebyte. The prebyte is treated as a one-byte instruction. If the prebyte is misaligned, the O cycle is a P cycle; if the prebyte is aligned, the O cycle is an f cycle. As a result, both the taken and not-taken cases use one O cycle for the prebyte. In the not-taken case, the queue must advance so that execution can continue with the next instruction, and another O cycle is required to maintain the queue. The taken case requires that the queue be refilled so that execution can continue at a new address. First, the effective address of the destination is determined, then the CPU performs three program word fetches from that address.

Freescale Semiconductor, Inc.

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

Condition Code State Notation

Table 11 Condition Code State Notation

–	Not changed by operation
0	Cleared by operation
1	Set by operation
Δ	Set or cleared by operation
⇓	May be cleared or remain set, but not set by operation
⇑	May be set or remain cleared, but not cleared by operation
?	May be changed by operation but final state not defined
!	Used for a special purpose

Freescale Semiconductor, Inc.

Pinout and Signal Description

Contents

MC9S12T64 Pin Assignments in 80-pin LQFP 59
Power Supply Pins 62
Signal Descriptions 65
Port Signals 74

MC9S12T64 Pin Assignments in 80-pin LQFP

The MC9S12T64 is available in an 80-pin low-profile quad flat package (LQFP). Most pins perform two or more functions, as described in the Signal Descriptions. Figure 11 shows pin assignments. In expanded narrow modes the lower byte data is multiplexed with higher byte data through pins PA7:0 (port A).

Freescale Semiconductor, Inc.

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

Pinout and Signal Description

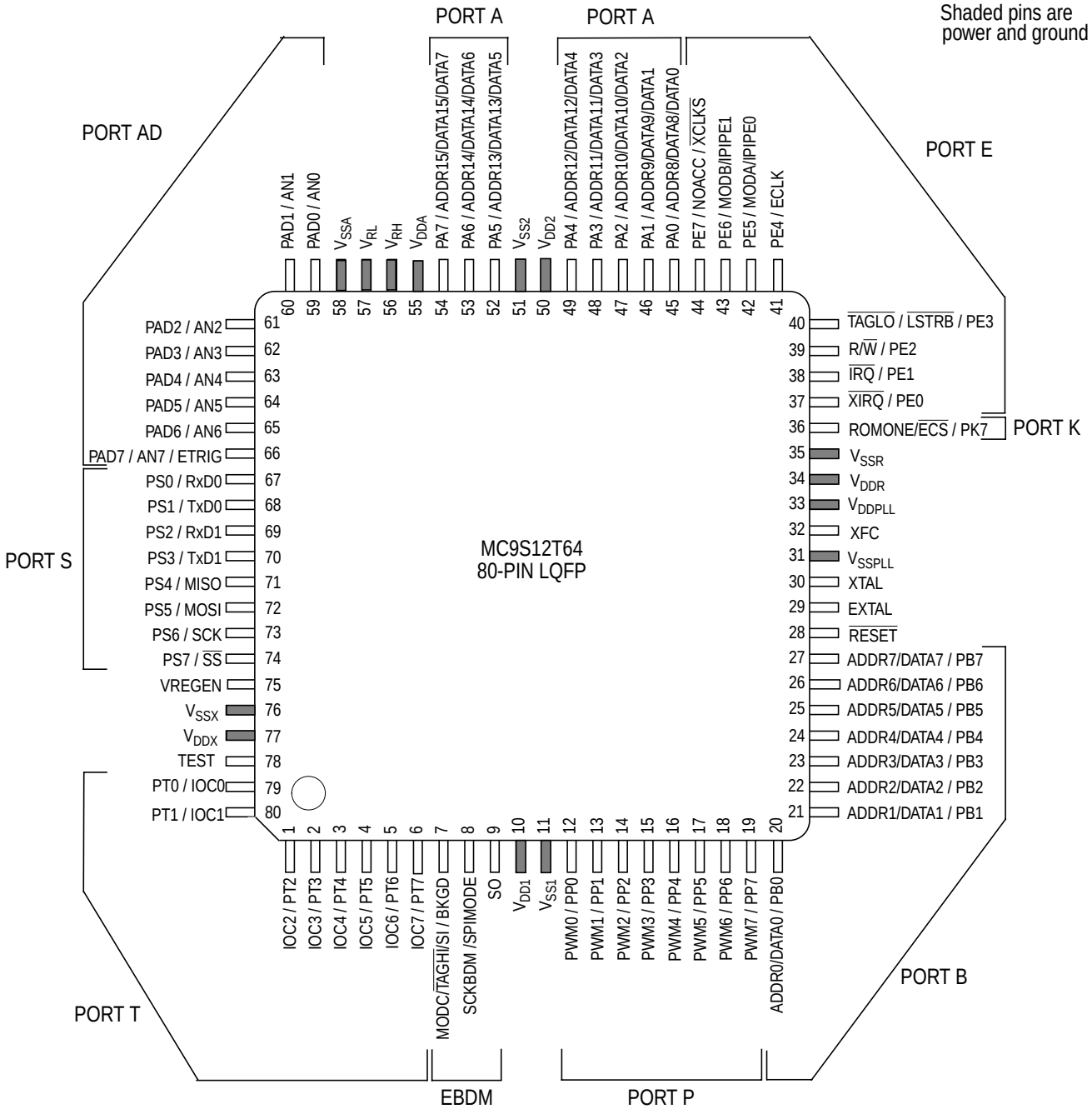
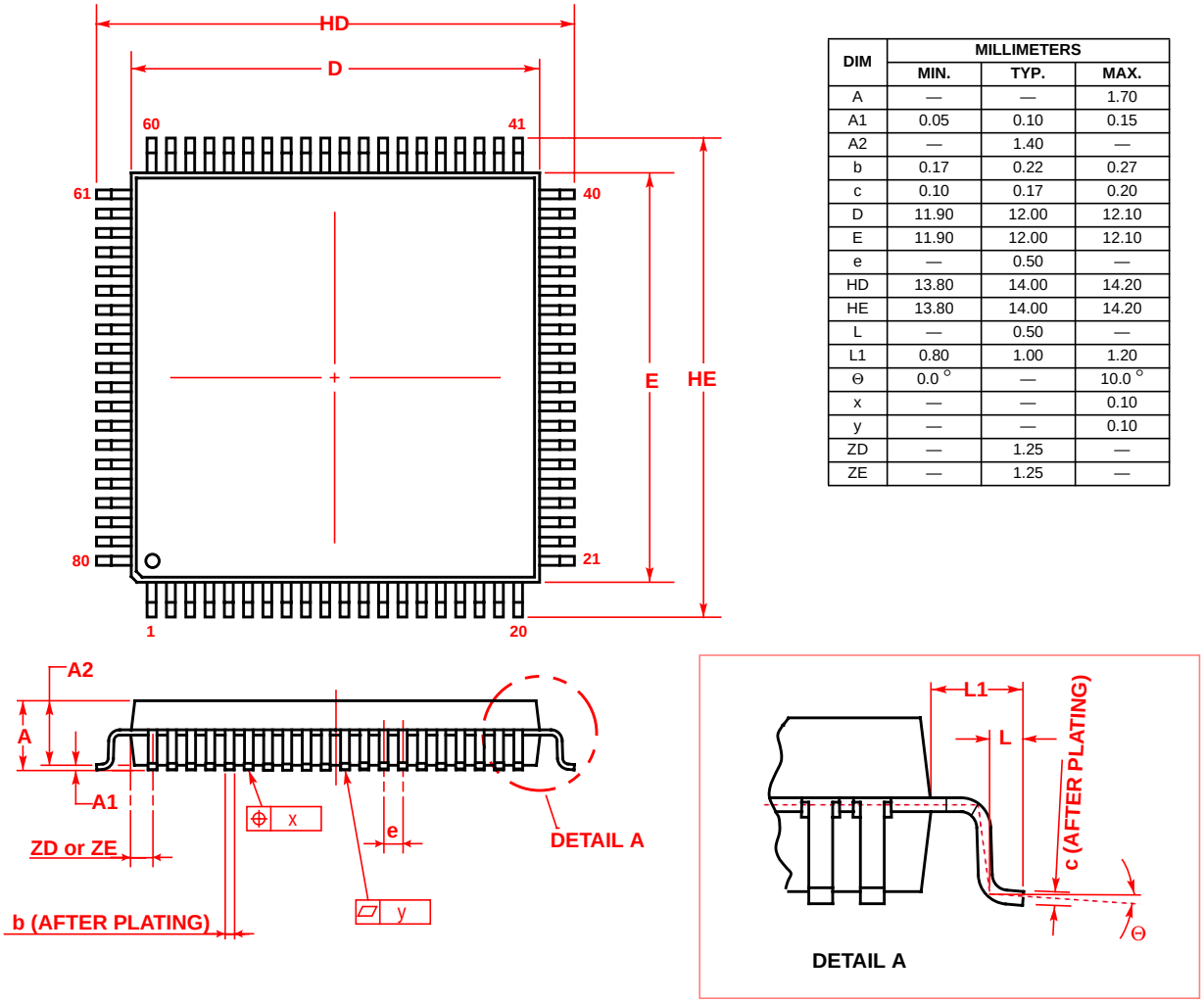


Figure 11 Pin Assignments in 80-pin LQFP for MC9S12T64

Freescale Semiconductor, Inc.

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



DIM	MILLIMETERS		
	MIN.	TYP.	MAX.
A	—	—	1.70
A1	0.05	0.10	0.15
A2	—	1.40	—
b	0.17	0.22	0.27
c	0.10	0.17	0.20
D	11.90	12.00	12.10
E	11.90	12.00	12.10
e	—	0.50	—
HD	13.80	14.00	14.20
HE	13.80	14.00	14.20
L	—	0.50	—
L1	0.80	1.00	1.20
Θ	0.0 °	—	10.0 °
x	—	—	0.10
y	—	—	0.10
ZD	—	1.25	—
ZE	—	1.25	—

Figure 12 80-pin LQFP Mechanical Dimensions

Freescale Semiconductor, Inc.

Power Supply Pins

MC9S12T64 power and ground pins are described below and summarized in Table 12.

NOTE: *All VSS pins must be connected together in the applications.*

V_{DDX}, V_{SSX} — Power & Ground Pins for I/O Drivers

External power and ground for I/O drivers. Because fast signal transitions place high, short-duration current demands on the power supply, use bypass capacitors with high-frequency characteristics and place them as close to the MCU as possible. Bypass requirements depend on how heavily the MCU pins are loaded.

V_{DDR}, V_{SSR} — Power & Ground Pins for I/O Drivers & for Internal Voltage Regulator

External power and ground for I/O drivers and input to the internal voltage regulator. Because fast signal transitions place high, short-duration current demands on the power supply, use bypass capacitors with high-frequency characteristics and place them as close to the MCU as possible. Bypass requirements depend on how heavily the MCU pins are loaded.

V_{DD1}, V_{DD2}, V_{SS1}, V_{SS2} — Core Power Pins

Power is supplied to the MCU through V_{DD} and V_{SS}. Because fast signal transitions place high, short-duration current demands on the power supply, use bypass capacitors with high-frequency characteristics and place them as close to the MCU as possible. This 2.5V supply is derived from the internal voltage regulator. No static load is allowed on these pins. The internal voltage regulator is turned off, if VREGEN is tied to ground.

NOTE: *No load allowed except for bypass capacitors.*

V_{DDA}, V_{SSA} — Power Supply Pins for ATD and VREG

V_{DDA} and V_{SSA} are the power supply and ground input pins for the voltage regulator and the analog to digital converter. It also provides the reference for the internal voltage regulator.This allows the supply voltage to the A/D and the reference voltage to be bypassed independently.

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

Freescale Semiconductor, Inc.

V_{RH}, V_{RL} — ATD Reference Voltage Input Pins

V_{RH} and V_{RL} are the reference voltage input pins for the analog-to-digital converter.

V_{DDPLL}, V_{SSPLL} — Power Supply Pins for PLL

Provides operating voltage and ground for the Oscillator and the Phased-Locked Loop. This allows the supply voltage to the Oscillator and PLL to be bypassed independently.This 2.5V voltage is generated by the internal voltage regulator.

NOTE: *No load allowed except for bypass capacitors.*

VREGEN — On Chip Voltage Regulator Enable

Enables the internal 5V to 2.5V voltage regulator. If this pin is tied low, VDD1,2 and VDDPLL must be supplied externally.

NOTE: *The voltage regulator must be enabled for proper LVD operation. If the voltage regulator is disabled, the LVD operation is unpredictable.*

Freescale Semiconductor, Inc.

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

Table 12 MC9S12T64 Power and Ground Connection Summary

Mnemonic	Pin Number 80 LQFP	Nominal Voltage	Description
V_{DD1}	10	2.5 V	Internal power and ground generated by internal regulator
V_{SS1}	11	0 V	
V_{DD2}	50	2.5 V	
V_{SS2}	51	0 V	
V_{DDR}	34	5.0 V	External power and ground, supply to pin drivers and internal voltage regulator.
V_{SSR}	35	0 V	
V_{DDX}	77	5.0 V	External power and ground, supply to pin drivers.
V_{SSX}	76	0 V	
V_{DDA}	55	5.0 V	Operating voltage and ground for the analog-to-digital converters and the reference for the internal voltage regulator, allows the supply voltage to the A/D to be bypassed independently.
V_{SSA}	58	0 V	
V_{RL}	57	0 V	Reference voltages for the analog-to-digital converter.
V_{RH}	56	5.0 V	
V_{DDPLL}	33	2.5 V	Provides operating voltage and ground for the Phased-Locked Loop. This allows the supply voltage to the PLL to be bypassed independently. Internal power and ground generated by internal regulator.
V_{SSPLL}	31	0 V	
VREGEN	75	5.0 V	Internal Voltage Regulator enable/disable

Freescale Semiconductor, Inc.

Signal Descriptions

EXTAL, XTAL — Oscillator Pins

EXTAL and XTAL are the crystal driver and external clock pins. On reset all the device clocks are derived from the EXTAL input frequency. XTAL is the crystal output.

Refer to the Clocks and Reset Generator (CRG) section for more information.

$\overline{\text{RESET}}$ — External Reset Pin

An active low bidirectional control signal, it acts as an input to initialize the MCU to a known start-up state, and an output when an internal MCU function causes a reset.

TEST — Test Pin

This input only pin is reserved for test.

NOTE: *The TEST pin must be tied to VSS in all applications.*

VREGEN — Voltage Regulator Enable Pin

This input only pin enables or disables the on-chip voltage regulator.

XFC — PLL Loop Filter Pin

PLL loop filter. Please ask your Motorola representative for the interactive application note to compute PLL loop filter elements. Any current leakage on this pin must be avoided. Refer to the Clocks and Reset Generator (CRG) section for more information.

BKGD / $\overline{\text{TAGHI}}$ / SI / MODC — Background Debug, Tag High, and Mode Pin

The BKGD/$\overline{\text{TAGHI}}$/SI/MODC pin is used as a pseudo-open-drain pin for the background debug communication. In MCU expanded modes of operation when instruction tagging is on, an input low on this pin during the falling edge of E-clock tags the high half of the instruction word being read into the instruction queue. It is used as a MCU operating mode select pin during reset. The state of this pin is latched to the MODC bit at the rising edge of $\overline{\text{RESET}}$.

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

Freescale Semiconductor, Inc.

This pin has an weak on-chip active pullup that is enabled at all times. It is assumed that there is an external pullup and that drivers connected to BKGD do not typically drive the high level.

NOTE: *The resistance of the internal pull-up may be too high depending on the speed and the load to ensure proper BDM communication. In this case an additional external pull-up resistor must be provided.*

SPIMODE/SCKBDM pin — This pin is used as serial clock when FBDM is in SPI mode. This pin has an weak on-chip active pull-down that is enabled at all times. FBDM will be in single wire mode as default. Pulling this pin high will put FBDM in SPI mode.

SO pin — This pin is used as a serial out data pin when FBDM is in SPI mode. During reset and immediately out of reset this pin will drive low.

PAD[7:0] / AN0[7:0] — Port AD Input Pins [7:0] — PAD7-PAD0 are general purpose input pins and analog inputs of the analog to digital converter.

PA[7:0] / ADDR[15:8] / DATA[15:8] — Port A I/O Pins — PA7-PA0 are general purpose input or output pins. In MCU expanded modes of operation, these pins are used for the multiplexed external address and data bus.

PB[7:0] / ADDR[7:0] / DATA[7:0] — Port B I/O Pins — PB7-PB0 are general purpose input or output pins. In MCU expanded modes of operation, these pins are used for the multiplexed external address and data bus.

PE7 / NOACC / $\overline{\text{XCLKS}}$ — Port E I/O Pin 7 — PE7 is a general purpose input or output pin. During MCU expanded modes of operation, the NOACC signal, when enabled, is used to indicate that the current bus cycle is an unused or “free” cycle. This signal will assert when the CPU is not using the bus.

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

Freescale Semiconductor, Inc.

The $\overline{\text{XCLKS}}$ input controls whether a crystal in combination with the internal Colpitts oscillator is used or whether Pierce oscillator/external clock circuitry is used. The state of this pin is latched at the rising edge of $\overline{\text{RESET}}$. If the input is a logic low, Pierce oscillator/external clock circuit is configured on the EXTAL and XTAL pins. If the input is a logic high, the Colpitts oscillator circuit is configured on EXTAL and XTAL. Since this pin is an input with a pull-up device, if the pin is left floating, the default configuration is the Colpitts oscillator circuit on EXTAL and XTAL.

PE6 / MODB / IPIPE1 — Port E I/O Pin 6

PE6 is a general purpose input or output pin. It is used as a MCU operating mode select pin during reset. The state of this pin is latched to the MODB bit at the rising edge of $\overline{\text{RESET}}$. This pin is shared with the instruction queue tracking signal IPIPE1. This pin is an input with a pull-down device which is only active when $\overline{\text{RESET}}$ is low.

PE5 / MODA / IPIPE0 — Port E I/O Pin 5

PE5 is a general purpose input or output pin. It is used as a MCU operating mode select pin during reset. The state of this pin is latched to the MODA bit at the rising edge of $\overline{\text{RESET}}$. This pin is shared with the instruction queue tracking signal IPIPE0. This pin is an input with a pull-down device which is only active when $\overline{\text{RESET}}$ is low.

PE4 / ECLK — Port E I/O Pin 4

PE4 is a general purpose input or output pin. It can be configured to drive the internal bus clock ECLK. ECLK can be used as a timing reference.

PE3 / $\overline{\text{LSTRB}}$ / $\overline{\text{TAGLO}}$ — Port E I/O Pin 3

PE3 is a general purpose input or output pin. In MCU expanded modes of operation, $\overline{\text{LSTRB}}$ can be used for the low-byte strobe function to indicate the type of bus access and when instruction tagging is on, $\overline{\text{TAGLO}}$ is used to tag the low half of the instruction word being read into the instruction queue.

PE2 / R/$\overline{\text{W}}$ — Port E I/O Pin 2

PE2 is a general purpose input or output pin. In MCU expanded modes of operations, this pin drives the read/write output signal for the external bus. It indicates the direction of data on the external bus.

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

Freescale Semiconductor, Inc.

PE1 / $\overline{\text{IRQ}}$ — Port E Input Pin 1

PE1 is a general purpose input pin and the maskable interrupt request input that provides a means of applying asynchronous interrupt requests. This will wake up the MCU from STOP or WAIT mode.

PE0 / $\overline{\text{XIRQ}}$ — Port E Input Pin 0

PE0 is a general purpose input pin and the non-maskable interrupt request input that provides a means of applying asynchronous interrupt requests. This will wake up the MCU from STOP or WAIT mode.

PK7 / $\overline{\text{ECS}}$ / ROMONE — Port K I/O Pin 7

PK7 is a general purpose input or output pin. During MCU expanded modes of operation, this pin is used as the emulation chip select output ($\overline{\text{ECS}}$). During MCU Normal Expanded and Emulation modes of operation, this pin is used to enable the Flash EEPROM memory in the memory map (ROMONE). At the rising edge of $\overline{\text{RESET}}$, the state of this pin determines the state of the ROMON bit (MISC register) according to Table 24 in page 129. See page 128 for details about the ROMON bit.

PP7 / PWM7 — Port P I/O Pin 7

PP7 is a general purpose input or output pin. It can be configured as Pulse Width Modulator (PWM) channel 7 output.

PP6 / PWM6 — Port P I/O Pin 6

PP6 is a general purpose input or output pin. It can be configured as Pulse Width Modulator (PWM) channel 6 output.

PP5 / PWM5 — Port P I/O Pin 5

PP5 is a general purpose input or output pin. It can be configured as Pulse Width Modulator (PWM) channel 5 output.

PP4 / PWM4 — Port P I/O Pin 4

PP4 is a general purpose input or output pin. It can be configured as Pulse Width Modulator (PWM) channel 4 output.

PP3 / PWM3 — Port P I/O Pin 3

PP3 is a general purpose input or output pin. It can be configured as Pulse Width Modulator (PWM) channel 3 output.

Freescale Semiconductor, Inc.

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

PP2 / PWM2 — Port P I/O Pin 2 PP2 is a general purpose input or output pin. It can be configured as Pulse Width Modulator (PWM) channel 2 output.

PP1 / PWM1 — Port P I/O Pin 1 PP1 is a general purpose input or output pin. It can be configured as Pulse Width Modulator (PWM) channel 1 output.

PP0 / PWM0 — Port P I/O Pin 0 PP0 is a general purpose input or output pin. It can be configured as Pulse Width Modulator (PWM) channel 0 output.

PS7 / $\overline{SS}$ — Port S I/O Pin 7 PS6 is a general purpose input or output pin. It can be configured as the slave select pin $\overline{SS}$ of the Serial Peripheral Interface.

PS6 / SCK — Port S I/O Pin 6 PS6 is a general purpose input or output pin. It can be configured as the serial clock pin SCK of the Serial Peripheral Interface.

PS5 / MOSI — Port S I/O Pin 5 PS5 is a general purpose input or output pin. It can be configured as master output (during master mode) or slave input pin (during slave mode) MOSI of the Serial Peripheral Interface.

PS4 / MISO — Port S I/O Pin 4 PS4 is a general purpose input or output pin. It can be configured as master input (during master mode) or slave output pin (during slave mode) MOSI of the Serial Peripheral Interface.

PS3 / TXD1 — Port S I/O Pin 3 PS3 is a general purpose input or output pin. It can be configured as the transmit pin TXD of Serial Communication Interface 1 (SCI1).

PS2 / RXD1 — Port S I/O Pin 2 PS2 is a general purpose input or output pin. It can be configured as the receive pin RXD of Serial Communication Interface 1 (SCI1).

PS1 / TXD0 — Port S I/O Pin 1 PS1 is a general purpose input or output pin. It can be configured as the transmit pin TXD of Serial Communication Interface 0 (SCI0).

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

Freescale Semiconductor, Inc.

PS0 / RXD0 — Port S I/O Pin 0 PS0 is a general purpose input or output pin. It can be configured as the receive pin RXD of Serial Communication Interface 0 (SCI0).

PT[7:0] / IOC[7:0] — Port T I/O Pins [7:0] PT7-PT0 are general purpose input or output pins. They can be configured as input capture or output compare pins IOC7-IOC0 of the Enhanced Capture Timer (ECT).

Table 13 MC9S12T64 Signal Description Summary

Pin Function	Pin Name	Powered by	Pin Number	Description
IOC2	PT2	VDDX	1	Capture Timer Channel
IOC3	PT3	VDDX	2	
IOC4	PT4	VDDX	3	
IOC5	PT5	VDDX	4	
IOC6	PT6	VDDX	5	
IOC7	PT7	VDDX	6	
MODC/$\overline{\text{TAGHI}}$/BKGD / SI	BKGD	VDDX	7	Pseudo_open_drain communication pin for the background debug mode. At the rising edge on $\overline{\text{RESET}}$, the state of this pin activates the BDM (when BKGD = 1). When instruction tagging is on, a 0 at the falling edge of E tags the high half of the instruction word being read into the instruction queue.
SPIMODE/SCKBDM	SPIMODE	VDDX	8	Serial clock pin when FBDM is in SPI mode.
SO	SO	VDDX	9	Serial out data pin when FBDM is in SPI mode.
	VDD1	VDD1	10	2.5V core supply
	VSS1	VSS1	11	
PWM0	PP0	VDDX	12	Pulse Width Modulator channel outputs
PWM1	PP1	VDDX	13	
PWM2	PP2	VDDX	14	
PWM3	PP3	VDDX	15	
PWM4	PP4	VDDX	16	
PWM5	PP5	VDDX	17	
PWM6	PP6	VDDX	18	
PWM7	PP7	VDDX	19	

Freescale Semiconductor, Inc.

Table 13 MC9S12T64 Signal Description Summary (Continued)

Pin Function	Pin Name	Powered by	Pin Number	Description
ADDR0 / DATA0	PB0	VDDR	20	External bus pins share function with general-purpose I/O port B. In single chip modes, the pins can be used for general-purpose I/O. In expanded modes, the pins are used for the external address and data buses.
ADDR1 / DATA1	PB1	VDDR	21	
ADDR2 / DATA2	PB2	VDDR	22	
ADDR3 / DATA3	PB3	VDDR	23	
ADDR4 / DATA4	PB4	VDDR	24	
ADDR5 / DATA5	PB5	VDDR	25	
ADDR6 / DATA6	PB6	VDDR	26	
ADDR7 / DATA7	PB7	VDDR	27	
$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	VDDR	28	An active low bidirectional control signal, $\overline{\text{RESET}}$ acts as an input to initialize the MCU to a known start-up state, and an output when COP or clock monitor or LVD causes a reset.
EXTAL	EXTAL	VDDPLL	29	Crystal driver and external clock input pins. On reset all the device clocks are derived from the EXTAL input frequency. XTAL is the crystal output.
XTAL	XTAL	VDDPLL	30	
	VSSPLL	–	31	2.5V PLL ground
	XFC	VDDPLL	32	External PLL Filter Capacitor
	VDDPLL	–	33	2.5V PLL supply
	VDDR	VDDR	34	5V Voltage Regulator and I/O Supply
	VSSR	VSSR	35	5V Voltage Regulator and I/O Ground
$\overline{\text{ECS}}$/ROMONE	PK7	VDDR	36	Emulation Chip select/ROMONE pin shares function with general-purpose I/O port.
$\overline{\text{XIRQ}}$	PE0	VDDR	37	The $\overline{\text{XIRQ}}$ input provides a means of requesting a nonmaskable interrupt after reset initialization. Because it is level sensitive, it can be connected to a multiple-source wired-OR network.
$\overline{\text{IRQ}}$	PE1	VDDR	38	Maskable interrupt request input provides a means of applying asynchronous interrupt requests to the MCU. Either falling edge-sensitive triggering or level-sensitive triggering is program selectable (IRQCR register).
R/$\overline{\text{W}}$	PE2	VDDR	39	Indicates direction of data on expansion bus. Shares function with general-purpose I/O. Read/write in expanded modes.
$\overline{\text{LSTRB}}$ / $\overline{\text{TAGLO}}$	PE3	VDDR	40	Low byte strobe (0 = low byte valid), in all modes this pin can be used as I/O. Pin function $\overline{\text{TAGLO}}$ used in instruction low byte tagging.
ECLK	PE4	VDDR	41	E Clock is the output connection for the external bus clock. ECLK is used as a timing reference and for address demultiplexing.

Freescale Semiconductor, Inc.

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

Table 13 MC9S12T64 Signal Description Summary (Continued)

Pin Function	Pin Name	Powered by	Pin Number	Description
MODA / IPIPE0	PE5	VDDR	42	State of mode select pins during reset determine the initial operating mode of the MCU. After reset, MODB and MODA can be configured as instruction queue tracking signals IPIPE1 and IPIPE0 or as general-purpose I/O pins.
MODB / IPIPE1	PE6	VDDR	43	
NOACC / $\overline{\text{XCLKS}}$	PE7	VDDR	44	No Access. Indicates free cycles in expanded mode. Selects also Colpitts oscillator or Pierce oscillator/external clock during reset. Can be used as general purpose I/O pin.
ADDR8 / DATA8 / DATA0	PA0	VDDR	45	External bus pins share function with general-purpose I/O ports A. In single chip modes, the pins can be used for general-purpose I/O. In expanded modes, the pins are used for the external buses.
ADDR9 / DATA9 / DATA1	PA1	VDDR	46	
ADDR10 / DATA10 / DATA2	PA2	VDDR	47	
ADDR11 / DATA11 / DATA3	PA3	VDDR	48	
ADDR12 / DATA12 / DATA4	PA4	VDDR	49	
	VDD2	VDD2	50	2.5V core supply
	VSS2	VSS2	51	
ADDR13 / DATA13 / DATA5	PA5	VDDR	52	External bus pins share function with general-purpose I/O ports A. In single chip modes, the pins can be used for general-purpose I/O. In expanded modes, the pins are used for the external buses.
ADDR14 / DATA14 / DATA6	PA6	VDDR	53	
ADDR15 / DATA15 / DATA7	PA7	VDDR	54	
	VDDA	–	55	5.0V supply analog to digital converter
	VRH	VDDA	56	analog to digital converter reference high
	VRL	VDDA	57	analog to digital converter reference low
	VSSA	–	58	ground analog to digital converter
AN0	PAD0	VDDA	59	A/D Converter channel 0
AN1	PAD1	VDDA	60	A/D Converter channel 1
AN2	PAD2	VDDA	61	A/D Converter channel 2
AN3	PAD3	VDDA	62	A/D Converter channel 3
AN4	PAD4	VDDA	63	A/D Converter channel 4
AN5	PAD5	VDDA	64	A/D Converter channel 5
AN6	PAD6	VDDA	65	A/D Converter channel 6

Freescale Semiconductor, Inc.

Table 13 MC9S12T64 Signal Description Summary (Continued)

Pin Function	Pin Name	Powered by	Pin Number	Description
AN7 / ETRIG	PAD7	VDDA	66	A/D Converter channel 7, or an external trigger for the ATD conversion
RxD0	PS0	VDDX	67	SCI0 receive pin
TxD0	PS1	VDDX	68	SCI0 transmit pin
RxD1	PS2	VDDX	69	SCI1 receive pin
TxD1	PS3	VDDX	70	SCI1 transmit pin
MISO	PS4	VDDX	71	Master in/slave out pin for serial peripheral interface
MOSI	PS5	VDDX	72	Master out/slave in pin for serial peripheral interface
SCK	PS6	VDDX	73	Serial clock for serial peripheral interface system
$\overline{SS}$	PS7	VDDX	74	Slave select output for SPI master mode, input for slave mode or master mode.
VREGEN	VREGEN	VDDX	75	Internal Voltage Regulator Enable
	VSSX	VSSX	76	5V I/O supply and Ground
	VDDX	VDDX	77	
TEST_VPP	TEST	VDDX	78	Configures the device for various test modes including SCAN testing. Also the programming voltage input for NVMs during factory test. This pin must be tied to ground in all applications.
IOC0	PT0	VDDX	79	Capture Timer Channel
IOC1	PT1	VDDX	80	

Freescale Semiconductor, Inc.

Port Signals

The MC9S12T64 incorporates seven ports which are used to control and access the various device subsystems. When not used for these purposes, port pins may be used for general-purpose I/O. In addition to the pins described below, each port consists of a data register which can be read and written at any time, and, with the exception of port AD and PE[1:0], a data direction register which controls the direction of each pin. After reset all general purpose I/O pins are configured as input.

Port A

Port A bits 7 through 0 are associated with address lines A15 through A8 respectively and data lines D15/D7 through D8/D0 respectively. When this port is not used for external addresses such as in single-chip mode, these pins can be used as general purpose I/O. Data Direction Register A (DDRA) determines the primary direction of each pin. DDRA also determines the source of data for a read of PORTA.

Register DDRA determines whether each port A pin is an input or output. DDRA is not in the address map during expanded and peripheral mode operation. Setting a bit in DDRA makes the corresponding bit in port A an output; clearing a bit in DDRA makes the corresponding bit in port A an input. The default reset state of DDRA is all zeroes.

This register is not in the on-chip map in expanded and peripheral modes.

Port B

Port B bits 7 through 0 are associated with address lines A7 through A0 respectively and data lines D7 through D0 respectively. When this port is not used for external addresses, such as in single-chip mode, these pins can be used as general purpose I/O. Data Direction Register B (DDRB) determines the primary direction of each pin. DDRB also determines the source of data for a read of PORTB.

Register DDRB determines whether each port B pin is an input or output. DDRB is not in the address map during expanded and peripheral mode operation. Setting a bit in DDRB makes the corresponding bit in port B

Freescale Semiconductor, Inc.

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

an output; clearing a bit in DDRB makes the corresponding bit in port B an input. The default reset state of DDRB is all zeroes.

This register is not in the on-chip map in expanded and peripheral modes.

Port E

Port E is associated with external bus control signals and interrupt inputs. These include mode select (PE7/NOACC/$\overline{\text{XCLKS}}$, PE6/MODB/IPIPE1, PE5/MODA/IPIPE0), E clock, size ($\overline{\text{LSTRB}}$/$\overline{\text{TAGLO}}$), read / write (R/$\overline{\text{W}}$), $\overline{\text{IRQ}}$, and $\overline{\text{XIRQ}}$. When the associated pin is not used for one of these specific functions, the pin can be used as general purpose I/O with the exception of $\overline{\text{IRQ}}$ and $\overline{\text{XIRQ}}$. The Port E Assignment Register (PEAR) selects the function of each pin and DDRE determines whether each pin is an input or output when it is configured to be general purpose I/O. DDRE also determines the source of data for a read of PORTE.

The $\overline{\text{XCLKS}}$ input selects between Colpitts oscillator or Pierce oscillator/an external clock configuration.

Some of these pins have software selectable pullups (NOACC, ECLK, $\overline{\text{LSTRB}}$, R/$\overline{\text{W}}$, $\overline{\text{IRQ}}$ and $\overline{\text{XIRQ}}$). A single control bit enables the pullups for all of these pins when they are configured as inputs.

This register is not in the on-chip map in peripheral mode or in expanded modes when the EME bit is set.

Port K

This port is associated with the internal memory emulation pin. When the port is not enabled to emulate the internal memory, the port pin is used as general-purpose I/O.

When input, this pin can be selected to be high impedance or pulled up, based upon the state of the PUPKE bit in the PUCR register.

Register DDRK determines whether port K pin is an input or output when configured as general-purpose I/O. DDRK is not in the address map during expanded and peripheral mode operation with EMK set. Setting a bit in DDRK makes the corresponding bit in port K an output; clearing

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

Freescale Semiconductor, Inc.

a bit in DDRK makes the corresponding bit in port K an input. The default reset state of DDRK is all zeroes.

This register is not in the map in peripheral mode or in expanded modes while the EMK bit in the MODE register is set.

NOTE: *The ports A, B, E, K, P, S, T can be configured in a very flexible way. For a full and detailed overview refer to Section Port Integration Module (PIM).*

AD PORT

This port is an analog input interface to the analog-to-digital subsystem. The digital function of the ports must explicitly be enabled on per pin basis using a control register in the ADC module. When the port data registers are read, they contain the digital levels appearing on the input pins at the time of the read. Input pins with signal potentials not meeting V_{IL} or V_{IH} specifications will have an indeterminate value.

Use of any AD port pin except ETRIG for digital input does not preclude the use of any other port pin for analog input. Note that the digital/analog multiplexing function is performed in the input pad. The port registers are connected to the input pads through tristate buffers. These buffers are only activated when the port is configured as digital pin so that analog signal potentials on the input pins will not cause the buffers to draw excess supply current.

Port S for SCI0, SCI1 and SPI

There are two identical SCI ports. Each SCI module uses two external pins. The RxD0, RxD1 and TxD0, TxD1pins share general purpose port SCI P[3:0]. TxD0, TxD1 are available for general-purpose I/O when it is not configured for transmitter operation. RxD0, RxD1 are available for general-purpose I/O when it is not configured for receiver operation.

The SPI module uses four external pins. $\overline{SS}$, SCK, MOSI, and MISO pins share general purpose port PS[7:4].

Port T for Timer Port

The timer module has eight external pins: PT[7:0]_IOC[7:0], for input capture and output compare functions. Two of the pins are also the pulse

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

Freescale Semiconductor, Inc.

accumulator inputs. All eight pins are available for general-purpose I/O when not configured for timer functions.

Port P for PWM

The PWM module has a total of 8 external pins on which the pulse width modulated waveforms are output. The 8 PWM outputs are multiplexed on the PP[7:0] pins.

.

Table 14 MC9S12T64 Port A, B, E, K, T, S, P Description Summary

Port Name	Data Direction Register	Description
PE[7:0]	PE[1:0] In PE[7:2] In/Out DDRE	Mode selection, bus control signals and interrupt service request signals; or general-purpose I/O.
PB[7:0]	In/Out DDRB	Port A and port B pins are used for address and data in expanded modes. The port data registers are not in the address map during expanded and peripheral mode operation. When in the map, port A and port B can be read or written any time. DDRA and DDRB are not in the address map in expanded or peripheral modes.
PA[7:0]	In/Out DDRA	
PK7	In/Out DDRK	Emulation chip select signal or general-purpose I/O.
PT[7:0]	In/Out DDRT	Enhanced Capture Timer signals; or general-purpose I/O.
PS[7:0]	In/Out DDRS	SPI and SCI signals; or general-purpose I/O.
PP[7:0]	In/Out DDRP	PWM signals; or general-purpose I/O.

Port Pull-Up Pull-Down and Reduced Drive

MCU ports can be configured for internal pull-up. To reduce power consumption and RFI, the pin output drivers can be configured to operate at a reduced drive level. Reduced drive causes a slight increase in transition time depending on loading and should be used only for ports which have a light loading. Table 15 summarizes the port pull-up default status and controls.

Freescale Semiconductor, Inc.

Table 15 Port A, B, E, K, AD Pull-Up, Pull-Down and Reduced Drive Summary

Port Name	Resistive Input Loads	Enable Bit			Reduced Drive Control Bit		
		Register (Address)	Bit Name	Reset State	Register (Address)	Bit Name	Reset State
Port A	Pull-up	PUCR ($000C)	PUPA	Disabled	RDRIV ($000D)	RDPA	Full drive
Port B	Pull-up	PUCR ($000C)	PUPB	Disabled	RDRIV ($000D)	RDPB	Full drive
Port E:							
PE7, PE[4:0]	Pull-up	PUCR ($000C)	PUPE	Enabled	RDRIV ($000D)	RDPE	Full drive
PE[6:5]	Pull-down (1)	—	—	Enabled	RDRIV ($000D)	RDPE	Full drive
Port K7	Pull-up	PUCR ($000C)	PUPKE	Enabled	RDRIV ($000D)	RDPK	Full drive
Port AD[7:0]	None	—					

1. Pull-down is only active when $\overline{\text{RESET}}$ is low.

Freescale Semiconductor, Inc.

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

System Configuration

Contents

Introduction 79
Modules Variabilities 79
MCU Variabilities. 80
System Clock Description 81

Introduction

This section describes the variabilities of the modules that are present at the MCU level and how they are connected.

Modules Variabilities

MMC

Table 16 MMC Module Variable I/O Signals

Signal Name	I/O	Configuration Description	Connected to
reg_sw0	I	Register Block Size set to 1K bytes	0
ram_sw2:0	I	RAM Memory Size set to 2K bytes	000
eep_sw1:0	I	CALRAM Size set to 2K bytes	01
rom_sw1:0	I	ROM Mapping Select set to 64K bytes	11
pag_sw1:0	I	64K bytes on-chip / 0K bytes off-chip space	11

The information is also readable by the MCU at address $1C, $1D (MEMSIZ0 and MEMSIZ1)

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

Freescale Semiconductor, Inc.

MCU Variabilities

Part ID Register Assignments

The PARTID register is located in the IPBI (IP-Bus interface) at address $__1A,$__1B. It contains a unique part ID for each revision of the chip. Table 17 contains the assigned part ID numbers.

Table 17 Assigned Part ID Numbers

Part Number	Mask Set Number	PARTID (1)
MC9S12T64	L42M	$422X

1. The coding is as follows:

Bit 15-12: Major Family identifier

Bit 11-8: Minor Family identifier

Bit 7-4: Major mask revision number including FAB transfers

Bit 3-0: Minor - non full - mask set revisions (Motorola use only for production control. This field may be changed. User program should not refer this field as an identification number.)

Part ID Register

This register is used to designate the part ID. Each revision of the chip will have a unique value in the contents of this register

Read anytime.

Write never.

.

PARTIDH — Part ID Register High

Address Offset: $001A (1)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
Write:								
Reset:	Refer to Table 17							

= Unimplemented

1. Register Address = Base Address (INITRG) + Address Offset

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

Freescale Semiconductor, Inc.

PARTIDL — Part ID Register High

Address Offset: $001B (1)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ID7	ID6	ID5	ID4	ID3 (2)	ID2 (2)	ID1 (2)	ID0 (2)
Write:								
Reset:	Refer to Table 17				—	—	—	—

= Unimplemented

1. Register Address = Base Address (INITRG) + Address Offset
2. Motorola use only for production control. This field may be changed. User program should not refer to this field as an identification number.

System Clock Description

The Clock and Reset Generator provides the internal clock signals for the core and all peripheral modules. Figure 13 shows the clock connections from the CRG to all modules. The gating condition placed on top of the individual clock gates indicates the dependencies of different modes (STOP, WAIT) and the setting of the respective configuration bits. For example, a WAIT(SYSWAI) gating condition states that when the SYSWAI bit is set, the correspondent gate will be disabled during WAIT mode.

Consult the CRG section in page 271 for details on clock generation and clock enabling conditions.

Table 18 summarizes the enabling conditions of specific modules according to the MCU mode of operation. All modules listed in Table 18 cease operation when the MCU enters STOP mode.

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

Freescale Semiconductor, Inc.

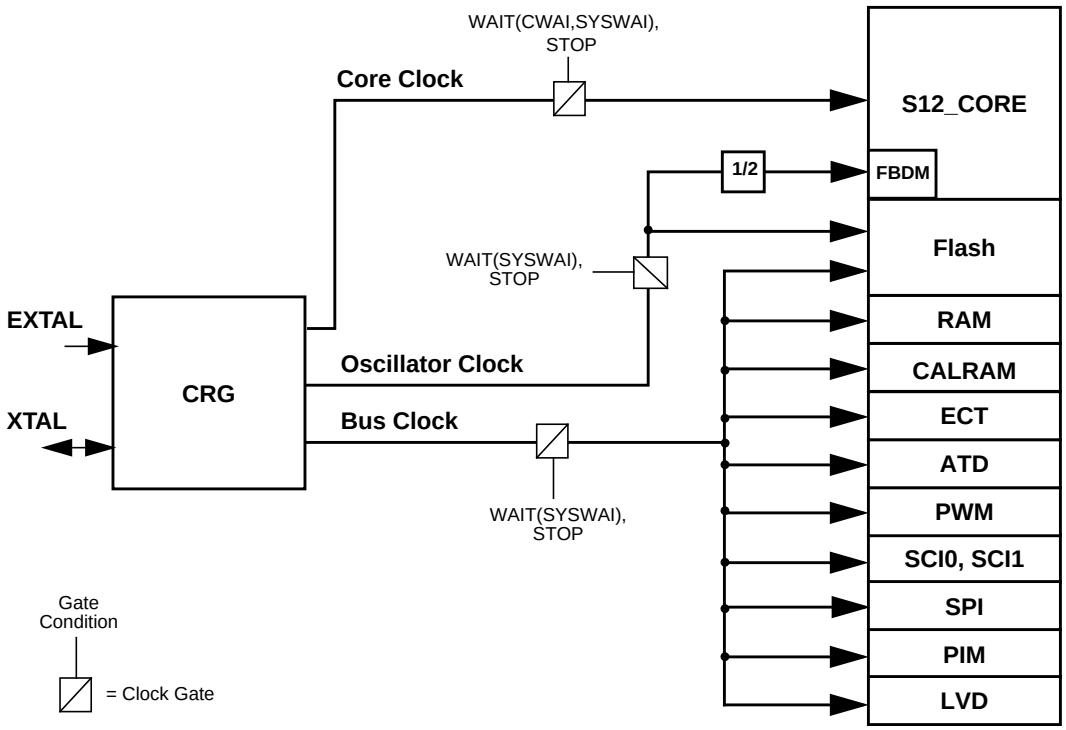


Figure 13 Clock Connections

Freescale Semiconductor, Inc.

Table 18 Module Availability in WAIT and RUN Modes

Module Name	Wait Mode[1] (Disable Bit)	Run Mode (Enable Bit)
HCS12 Core	Always Enabled	Always Enabled
Flash	Always Enabled	Always Enabled
CALRAM	Always Enabled	Always Enabled
SRAM	Always Enabled	Always Enabled
PIM	Always Enabled	Always Enabled
PWM	PSWAI (PWMCTL Register)	Always Enabled[2]
ECT	TSWAIT (TSCR1 Register)	Always Enabled
ATD	AWAI (ATDCTL2 Register)	Always Enabled[2]
SPI	SPISWAI (SPICR2 Register)	SPE (SPICR1 Register)
LVD	Disabled	LVDE (LVDCR Register)
SCI0, SCI1	SCISWAI (SCIxCR1 Register)	Always Enabled

1. Assuming all system clocks are running during WAIT mode (SYSWAI=0 in CLKSEL register - see page 284)

2. Low power options available. See the correspondent module section for details

Freescale Semiconductor, Inc.

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

System Configuration

Freescale Semiconductor, Inc.

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

Registers

Contents

Register Block 85
General Purpose Registers 103

Register Block

The register block can be mapped to any 2K byte boundary within the first 32K byte of the standard 64K byte address space by manipulating bits REG[14:11] in the INITRG register. INITRG establishes the upper five bits of the register block's 16-bit address. The register block occupies 1K byte. Default addressing (after reset) is indicated in the table below.

Non-user Registers Non-user registers are divided in three categories: Reserved, Unimplemented and Reserved for Factory Test. A detailed description follows below;

Reserved Registers Reserved registers return logic zero when read. Writes to these registers have no effect.

Unimplemented Registers Unimplemented registers return unpredictable values when read. Writes to these registers have no effect.

Reserved for Factory Test Registers Reserved for Factory Test registers return unpredictable values when read. Writes to these registers have no effect in Normal modes. Writing to these registers in Special modes might result in unpredictable MCU behavior.

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

Freescale Semiconductor, Inc.

Registers

Table 19 MC9S12T64 Register Map

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$0000	PORTA	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	MEBI
$0001	PORTB	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$0002	DDRA	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$0003	DDRB	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$0004	Reserved	Read: Write:	0	0	0	0	0	0	0	0	
$0005	Reserved	Read: Write:	0	0	0	0	0	0	0	0	
$0006	Reserved	Read: Write:	0	0	0	0	0	0	0	0	
$0007	Reserved	Read: Write:	0	0	0	0	0	0	0	0	
$0008	PORTE	Read: Write:	Bit 7	6	5	4	3	2	Bit 1	Bit 0	
$0009	DDRE	Read: Write:	Bit 7	6	5	4	3	Bit 2	0	0	
$000A	PEAR	Read: Write:	NOACCE	0	PIPOE	NECLK	LSTRE	RDWE	0	0	
$000B	MODE	Read: Write:	MODC	MODB	MODA	0	IVIS	0	EMK	EME	
$000C	PUCR	Read: Write:	PUPKE	0	0	PUPEE	0	0	PUPBE	PUPAE	
$000D	RDRIV	Read: Write:	RDPK	0	0	RDPE	0	0	RDPB	RDPA	
$000E	EBICTL	Read: Write:	0	0	0	0	0	0	0	ESTR	
$000F	Reserved	Read: Write:	0	0	0	0	0	0	0	0	

Freescale Semiconductor, Inc.

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$0010	INITRM	Read:	RAM15	RAM14	RAM13	RAM12	RAM11	0	0	RAMHAL	MMC
		Write:									
$0011	INITRG	Read:	0	REG14	REG13	REG12	REG11	0	0	0	
		Write:									
$0012	INITCRM	Read:	CRAM15	CRAM14	CRAM13	CRAM12	CRAM11	0	0	CRAMON	
		Write:									
$0013	MISC	Read:	0	0	0	0	EXSTR1	EXSTR0	ROMHM	ROMON	
		Write:									
$0014	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$0015	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$0016	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$0017	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$0018	Reserved	Read:	0	0	0	0	0	0	0	0	Peripheral
		Write:									
$0019	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$001A	PARTIDH	Read:	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	
		Write:									
$001B	PARTIDL	Read:	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	
		Write:									
$001C	MEMSIZ0	Read:	reg_sw0	0	eep_sw1	eep_sw0	0	ram_sw2	ram_sw1	ram_sw0	MMC
		Write:									
$001D	MEMSIZ1	Read:	rom_sw1	rom_sw0	0	0	0	0	pag_sw1	pag_sw0	
		Write:									
$001E	IRQCR	Read:	IRQE	IRQEN	0	0	0	0	0	0	MEBI
		Write:									
$001F	HPRIO	Read:	PSEL7	PSEL6	PSEL5	PSEL4	PSEL3	PSEL2	PSEL1	0	
		Write:									

Freescale Semiconductor, Inc.

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

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$0020	GPR0	Read: Write:	Bit 7	6	5	4	3	2	1	0	General Purpose Registers
$0021	GPR1	Read: Write:	Bit 7	6	5	4	3	2	1	0	
$0022	GPR2	Read: Write:	Bit 7	6	5	4	3	2	1	0	
$0023	GPR3	Read: Write:	Bit 7	6	5	4	3	2	1	0	
$0024	GPR4	Read: Write:	Bit 7	6	5	4	3	2	1	0	
$0025	GPR5	Read: Write:	Bit 7	6	5	4	3	2	1	0	
$0026	GPR6	Read: Write:	Bit 7	6	5	4	3	2	1	0	
$0027	GPR7	Read Write:	Bit 7	6	5	4	3	2	1	0	
$0028	BKPCT0	Read: Write:	BKEN	BKFULL	BKBDM	BKTAG	0	0	0	0	BKP
$0029	BKPCT1	Read: Write:	BK0MBH	BK0MBL	BK1MBH	BK1MBL	BK0RWE	BK0RW	BK1RWE	BK1RW	
$002A	BKP0X	Read: Write:	0	0	BK0V5	BK0V4	BK0V3	BK0V2	BK0V1	BK0V0	
$002B	BKP0H	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	
$002C	BKP0L	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$002D	BKP1X	Read: Write:	0	0	BK1V5	BK1V4	BK1V3	BK1V2	BK1V1	BK1V0	
$002E	BKP1H	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	
$002F	BKP1L	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$0030	PPAGE	Read: Write:	0	0	PIX5	PIX4	PIX3	PIX2	PIX1	PIX0	MMC
$0031	Reserved	Read: Write:	0	0	0	0	0	0	0	0	
$0032	PORTK	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	MEBI
$0033	DDRK	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	

Freescale Semiconductor, Inc.

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$0034	SYNR	Read:	0	0	SYN5	SYN4	SYN3	SYN2	SYN1	SYN0	CRG
		Write:									
$0035	REFDV	Read:	0	0	0	0	REFDV3	REFDV2	REFDV1	REFDV0	
		Write:									
$0036	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$0037	CRGFLG	Read:	RTIF	PORLVDRF	0	LOCKIF	LOCK	TRACK	SCMIF	SCM	
		Write:									
$0038	CRGINT	Read:	RTIE	0	0	LOCKIE	0	0	SCMIE	0	
		Write:									
$0039	CLKSEL	Read:	PLLSEL	PSTP	SYSWAI	ROAWAI	PLLWAI	CWAI	RTIWAI	COPWAI	
		Write:									
$003A	PLLCTL	Read:	CME	PLLON	AUTO	ACQ	0	PRE	PCE	SCME	
		Write:									
$003B	RTICTL	Read:	0	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0	
		Write:									
$003C	COPCTL	Read:	WCOP	RSBCK	0	0	0	CR2	CR1	CR0	
		Write:									
$003D	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$003E	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$003F	ARMCOP	Read:	0	0	0	0	0	0	0	0	
		Write:	Bit 7	6	5	4	3	2	1	Bit 0	

Freescale Semiconductor, Inc.

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

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$0040	TIOS	Read: Write:	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0	Enhanced Capture Timer
$0041	CFORC	Read: Write:	0 FOC7	0 FOC6	0 FOC5	0 FOC4	0 FOC3	0 FOC2	0 FOC1	0 FOC0	
$0042	OC7M	Read: Write:	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0	
$0043	OC7D	Read: Write:	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0	
$0044	TCNT (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	
$0045	TCNT (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$0046	TSCR1	Read: Write:	TEN	TSWAI	TSFRZ	TFFCA	0	0	0	0	
$0047	TTOV	Read: Write:	TOV7	TOV6	TOV5	TOV4	TOV3	TOV2	TOV1	TOV0	
$0048	TCTL1	Read: Write:	OM7	OL7	OM6	OL6	OM5	OL5	OM4	OL4	
$0049	TCTL2	Read: Write:	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0	
$004A	TCTL3	Read: Write:	EDG7B	EDG7A	EDG6B	EDG6A	EDG5B	EDG5A	EDG4B	EDG4A	
$004B	TCTL4	Read: Write:	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A	
$004C	TIE	Read: Write:	C7I	C6I	C5I	C4I	C3I	C2I	C1I	C0I	
$004D	TSCR2	Read: Write:	TOI	0	0	0	TCRE	PR2	PR1	PR0	
$004E	TFLG1	Read: Write:	C7F	C6F	C5F	C4F	C3F	C2F	C1F	C0F	
$004F	TFLG2	Read: Write:	TOF	0	0	0	0	0	0	0	

Freescale Semiconductor, Inc.

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$0050	TC0 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	Enhanced Capture Timer
$0051	TC0 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$0052	TC1 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	
$0053	TC1 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$0054	TC2 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	
$0055	TC2 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$0056	TC3 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	
$0057	TC3 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$0058	TC4 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	
$0059	TC4 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$005A	TC5 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	
$005B	TC5 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$005C	TC6 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	
$005D	TC6 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$005E	TC7 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	
$005F	TC7 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	

Freescale Semiconductor, Inc.

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$0060	PACTL	Read:	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI	Enhanced Capture Timer
		Write:									
$0061	PAFLG	Read:	0	0	0	0	0	0	PAOVF	PAIF	
		Write:									
$0062	PACN3 (hi)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$0063	PACN2 (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$0064	PACN1 (hi)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$0065	PACN0 (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$0066	MCCTL	Read:	MCZI	MODMC	RDMCL	0	0	MCEN	MCPR1	MCPR0	
		Write:				ICLAT	FLMC				
$0067	MCFLG	Read:	MCZF	0	0	0	POLF3	POLF2	POLF1	POLF0	
		Write:									
$0068	ICPAR	Read:	0	0	0	0	PA3EN	PA2EN	PA1EN	PA0EN	
		Write:									
$0069	DLYCT	Read:	0	0	0	0	0	0	DLY1	DLY0	
		Write:									
$006A	ICOVW	Read:	NOVW7	NOVW6	NOVW5	NOVW4	NOVW3	NOVW2	NOVW1	NOVW0	
		Write:									
$006B	ICSYS	Read:	SH37	SH26	SH15	SH04	TFMOD	PACMX	BUFEN	LATQ	
		Write:									
$006C	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$006D	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$006E	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$006F	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									

Freescale Semiconductor, Inc.

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$0070	PBCTL	Read:	0	PBEN	0	0	0	0	PBOVI	0	Enhanced Capture Timer
		Write:									
$0071	PBFLG	Read:	0	0	0	0	0	0	PBOVF	0	
		Write:									
$0072	PA3H	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$0073	PA2H	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$0074	PA1H	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$0075	PA0H	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$0076	MCCNT (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8	
		Write:									
$0077	MCCNT (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$0078	TC0H (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8	
		Write:									
$0079	TC0H (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$007A	TC1H (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8	
		Write:									
$007B	TC1H (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$007C	TC2H (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8	
		Write:									
$007D	TC2H (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$007E	TC3H (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8	
		Write:									
$007F	TC3H (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									

Freescale Semiconductor, Inc.

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

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$0080	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								ATD
		Write:									
$081	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$082	ATDCTL2	Read:	ADPU	AFFC	AWAI	ETRIGLE	ETRIGP	ETRIGE	ASCIE	ASCIF	
		Write:									
$0083	ATDCTL3	Read:	0	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0	
		Write:									
$0084	ATDCTL4	Read:	SRES8	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0	
		Write:									
$0085	ATDCTL5	Read:	DJM	DSGN	SCAN	MULT	0	CC	CB	CA	
		Write:									
$0086	ATDSTAT0	Read:	SCF	0	ETORF	FIFOR	0	CC2	CC1	CC0	
		Write:									
$0087	Unimplemented	Read:	Reads to this register return unpredictable values.								
		Write:									
$0088	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$0089	ATDTEST1	Read:	Reads to these bits return unpredictable values.							SC	
		Write:									
$008A	Unimplemented	Read:	Reads to this register return unpredictable values.								
		Write:									
$008B	ATDSTAT1	Read:	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0	
		Write:									
$008C	Unimplemented	Read:	Reads to this register return unpredictable values.								
		Write:									
$008D	ATDDIEN	Read:	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0	
		Write:									
$008E	Unimplemented	Read:	Reads to this register return unpredictable values.								
		Write:									
$008F	PORTAD	Read:	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0	
		Write:									

Freescale Semiconductor, Inc.

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$0090	ATDDR0H	Read:	Bit15	14	13	12	11	10	9	Bit8	ATD
		Write:									
$0091	ATDDR0L	Read:	Bit7	Bit6	0	0	0	0	0	0	
		Write:									
$0092	ATDDR1H	Read:	Bit15	14	13	12	11	10	9	Bit8	
		Write:									
$0093	ATDDR1L	Read:	Bit7	Bit6	0	0	0	0	0	0	
		Write:									
$0094	ATDDR2H	Read:	Bit15	14	13	12	11	10	9	Bit8	
		Write:									
$0095	ATDDR2L	Read:	Bit7	Bit6	0	0	0	0	0	0	
		Write:									
$0096	ATDDR3H	Read:	Bit15	14	13	12	11	10	9	Bit8	
		Write:									
$0097	ATDDR3L	Read:	Bit7	Bit6	0	0	0	0	0	0	
		Write:									
$0098	ATDDR4H	Read:	Bit15	14	13	12	11	10	9	Bit8	
		Write:									
$0099	ATDDR4L	Read:	Bit7	Bit6	0	0	0	0	0	0	
		Write:									
$009A	ATDDR5H	Read:	Bit15	14	13	12	11	10	9	Bit8	
		Write:									
$009B	ATDDR5L	Read:	Bit7	Bit6	0	0	0	0	0	0	
		Write:									
$009C	ATDDR6H	Read:	Bit15	14	13	12	11	10	9	Bit8	
		Write:									
$009D	ATDDR6L	Read:	Bit7	Bit6	0	0	0	0	0	0	
		Write:									
$009E	ATDDR7H	Read:	Bit15	14	13	12	11	10	9	Bit8	
		Write:									
$009F	ATDDR7L	Read:	Bit7	Bit6	0	0	0	0	0	0	
		Write:									

Freescale Semiconductor, Inc.

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

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$00A0	PWME	Read:	PWME7	PWME6	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0	PWM
		Write:									
$00A1	PWMPOL	Read:	PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0	
		Write:									
$00A2	PWMCLK	Read:	PCLK7	PCLKL6	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0	
		Write:									
$00A3	PWMPRCLK	Read:	0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0	
		Write:									
$00A4	PWMCAE	Read:	CAE7	CAE6	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0	
		Write:									
$00A5	PWMCTL	Read:	CON67	CON45	CON23	CON01	PSWAI	PFRZ	0	0	
		Write:									
$00A6	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$00A7	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$00A8	PWMSCLA	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$00A9	PWMSCLB	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
$00AA	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$00AB	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$00AC	PWMCNT0	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:	0	0	0	0	0	0	0	0	
$00AD	PWMCNT1	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:	0	0	0	0	0	0	0	0	
$00AE	PWMCNT2	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:	0	0	0	0	0	0	0	0	
$00AF	PWMCNT3	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:	0	0	0	0	0	0	0	0	

Freescale Semiconductor, Inc.

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$00B0	PWMCNT4	Read:	Bit 7	6	5	4	3	2	1	Bit 0	PWM
		Write:	0	0	0	0	0	0	0	0	
$00B1	PWMCNT5	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:	0	0	0	0	0	0	0	0	
$00B2	PWMCNT6	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:	0	0	0	0	0	0	0	0	
$00B3	PWMCNT7	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:	0	0	0	0	0	0	0	0	
$00B4	PWMPER0	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00B5	PWMPER1	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00B6	PWMPER2	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00B7	PWMPER3	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00B8	PWMPER4	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00B9	PWMPER5	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00BA	PWMPER6	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00BB	PWMPER7	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00BC	PWMDTY0	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00BD	PWMDTY1	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00BE	PWMDTY2	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00BF	PWMDTY3	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	

Freescale Semiconductor, Inc.

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

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$00C0	PWMDTY4	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	PWM
$00C1	PWMDTY5	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00C2	PWMDTY6	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00C3	PWMDTY7	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	
$00C4	PWMSDN	Read: Write:	PWMIF	PWMIE	0 PWMRSTRT	PWMLVL	0	PWM7IN	PWM7INL	PWM7ENA	
$00C5	Reserved	Read: Write:	0	0	0	0	0	0	0	0	
$00C6	Reserved	Read: Write:	0	0	0	0	0	0	0	0	
$00C7	Reserved	Read: Write:	0	0	0	0	0	0	0	0	
$00C8	SCI0BDH	Read: Write:	0	0	0	SBR12	SBR11	SBR10	SBR9	SBR8	SCI0
$00C9	SCI0BDL	Read: Write:	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	
$00CA	SCI0CR1	Read: Write:	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT	
$00CB	SCI0CR2	Read: Write:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	
$00CC	SCI0SR1	Read: Write:	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	
$00CD	SCI0SR2	Read: Write:	0	0	0	0	0	BRK13	TXDIR	RAF	
$00CE	SCI0DRH	Read: Write:	R8	T8	0	0	0	0	0	0	
$00CF	SCI0DRL	Read: Write:	R7 T7	R6 T6	R5 T5	R4 T4	R3 T3	R2 T2	R1 T1	R0 T0	

Freescale Semiconductor, Inc.

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$00D0	SCI1BDH	Read:	0	0	0	SBR12	SBR11	SBR10	SBR9	SBR8	SCI1
		Write:									
$00D1	SCI1BDL	Read:	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	
		Write:									
$00D2	SCI1CR1	Read:	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT	
		Write:									
$00D3	SCI1CR2	Read:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	
		Write:									
$00D4	SCI1SR1	Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	
		Write:									
$00D5	SCI1SR2	Read:	0	0	0	0	0	BRK13	TXDIR	RAF	
		Write:									
$00D6	SCI1DRH	Read:	R8	T8	0	0	0	0	0	0	
		Write:									
$00D7	SCI1DRL	Read:	R7	R6	R5	R4	R3	R2	R1	R0	
		Write:	T7	T6	T5	T4	T3	T2	T1	T0	
$00D8	SPICR1	Read:	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE	SPI
		Write:									
$00D9	SPICR2	Read:	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0	
		Write:									
$00DA	SPIBR	Read:	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0	
		Write:									
$00DB	SPISR	Read:	SPIF	0	SPTEF	MODF	0	0	0	0	
		Write:									
$00DC	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$00DD	SPIDR	Read:	Bit7	6	5	4	3	2	1	Bit0	
		Write:									
$00DE	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$00DF	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									

Freescale Semiconductor, Inc.

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

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$00E0	PTT	Read:	PTT7	PTT6	PTT5	PTT4	PTT3	PTT2	PTT1	PTT0	Port T
		Write:									
$00E1	PTIT	Read:	PTIT7	PTIT6	PTIT5	PTIT4	PTIT3	PTIT2	PTIT1	PTIT0	
		Write:									
$00E2	DDRT	Read:	DDRT7	DDRT7	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0	
		Write:									
$00E3	RDRT	Read:	RDRT7	RDRT6	RDRT5	RDRT4	RDRT3	RDRT2	RDRT1	RDRT0	
		Write:									
$00E4	PERT	Read:	PERT7	PERT6	PERT5	PERT4	PERT3	PERT2	PERT1	PERT0	
		Write:									
$00E5	PPST	Read:	PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0	
		Write:									
$00E6	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$00E7	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$00E8	PTS	Read:	PTS7	PTS6	PTS5	PTS4	PTS3	PTS2	PTS1	PTS0	Port S
		Write:									
$00E9	PTIS	Read:	PTIS7	PTIS6	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0	
		Write:									
$00EA	DDRS	Read:	DDRS7	DDRS7	DDRS5	DDRS4	DDRS3	DDRS2	DDRS1	DDRS0	
		Write:									
$00EB	RDRS	Read:	RDRS7	RDRS6	RDRS5	RDRS4	RDRS3	RDRS2	RDRS1	RDRS0	
		Write:									
$00EC	PERS	Read:	PERS7	PERS6	PERS5	PERS4	PERS3	PERS2	PERS1	PERS0	
		Write:									
$00ED	PPSS	Read:	PPSS7	PPSS6	PPSS5	PPSS4	PPSS3	PPSS2	PPSS1	PPSS0	
		Write:									
$00EE	WOMS	Read:	WOMS7	WOMS6	WOMS5	WOMS4	WOMS3	WOMS2	WOMS1	WOMS0	
		Write:									
$00EF	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									

Freescale Semiconductor, Inc.

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$00F0	PTP	Read:	PTP7	PTP6	PTP5	PTP4	PTP3	PTP2	PTP1	PTP0	Port P
		Write:									
$00F1	PTIP	Read:	PTIP7	PTIP6	PTIP5	PTIP4	PTIP3	PTIP2	PTIP1	PTIP0	
		Write:									
$00F2	DDRP	Read:	DDRP7	DDRP7	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0	
		Write:									
$00F3	RDRP	Read:	RDRP7	RDRP6	RDRP5	RDRP4	RDRP3	RDRP2	RDRP1	RDRP0	
		Write:									
$00F4	PERP	Read:	PERP7	PERP6	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0	
		Write:									
$00F5	PPSP	Read:	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSS0	
		Write:									
$00F6	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$00F7	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$00F8	LVDCR	Read:	LVDE	LVDRE	0	0	0	0	0	0	LVD
		Write:									
$00F9	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$00FA	LVDSR	Read:	LVDF	0	0	0	0	0	0	0	
		Write:									
$00FB	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$00FC	CALCFG	Read:	FSUM	0	0	0	0	0	0	0	CALRAM
		Write:									
$00FD	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$00FE	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$00FF	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									

Freescale Semiconductor, Inc.

Table 19 MC9S12T64 Register Map (Continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
$0100	FCLKDIV	Read:	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0	Flash Control
		Write:									
$0101	FSEC	Read:	KEYEN	NV6	NV5	NV4	NV3	NV2	SEC1	SEC0	
		Write:									
$0102	FTSTMOD	Read:	Reads to this register return unpredictable values in normal modes.								
		Write:									
$0103	FCNFG	Read:	CBEIE	CCIE	KEYACC	0	0	0	0	BKSEL0	
		Write:									
$0104	FPROT	Read:	FPOPEN	NV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0	
		Write:									
$0105	FSTAT	Read:	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	0	0	
		Write:									
$0106	FCMD	Read:	0	CMDB6	CMDB5	0	0	CMDB2	0	CMDB0	
		Write:									
$0107	Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								
		Write:									
$0108	FADDRHI	Read:	0	AB15	AB14	AB13	AB12	AB11	AB10	AB9	
		Write:									
$0109	FADDRLO	Read:	AB8	AB7	AB6	AB5	AB4	AB3	AB2	AB1	
		Write:									
$010A	FDATAHI	Read:	D15	D14	D13	D12	D11	D10	D9	D8	
		Write:									
$010B	FDATALO	Read:	D7	D6	D5	D4	D3	D2	D1	D0	
		Write:									
$010C	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$010D	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$010E	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$010F	Reserved	Read:	0	0	0	0	0	0	0	0	
		Write:									
$0110 - $03FF	Unimplemented	Read:	Reads to this register return unpredictable values.								
		Write:									

Freescale Semiconductor, Inc.

General Purpose Registers

General Purpose Registers (GPR[0-7]) are the register spaces reside in the first 256 bytes of the memory map (when register block is mapped at $0000) which can be used as the storage spaces by the user program.

GPR0-7 — General Purpose Registers

Address Offset: $0020 – $0027

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	GPR0–7							
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime
Write: Anytime

These registers will reset to 0 and can be written and read any time.

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

Operating Modes

Contents

Introduction 105
Operating Modes. 105
Background Debug Mode 116
Secured Mode of Operation 117

Introduction

Eight possible operating modes determine the operating configuration of the MC9S12T64. Each mode has an associated default memory map and external bus configuration.

Operating Modes

The operating mode out of reset is determined by the states of the MODC, MODB, and MODA pins during reset (refer to Table 20).

The MODC, MODB, and MODA bits in the MODE register show current operating mode and provide limited mode switching during operation. The states of the MODC, MODB, and MODA pins are latched into these bits on the rising edge of the reset signal.

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

Freescale Semiconductor, Inc.

Table 20 Mode Selection

Input BKGD & bit MODC	Input & bit MODB	Input & bit MODA	Mode Description
0	0	0	Special Single Chip, BDM allowed and ACTIVE. BDM is “allowed” in all other modes but a serial command is required to make BDM “active”.
0	0	1	Emulation Expanded Narrow, BDM allowed
0	1	0	Special Test (Expanded Wide) (1), BDM allowed
0	1	1	Emulation Expanded Wide, BDM allowed
1	0	0	Normal Single Chip, BDM allowed
1	0	1	Normal Expanded Narrow, BDM allowed
1	1	0	Peripheral (1); BDM allowed but bus operations would cause bus conflicts (must not be used)
1	1	1	Normal Expanded Wide, BDM allowed

1. This mode is intended for Motorola factory testing of the MCU.

There are two basic types of operating modes:

Normal modes — some registers and bits are protected against accidental changes.

Special modes — allow greater access to protected control registers and bits for special purposes such as testing.

A system development and debug feature, background debug mode (BDM), is available in all modes. In special single-chip mode, BDM is active immediately after reset.

The four 8-bit Ports (A, B, E and K) associated with the MEBI sub-block can serve as general purpose I/O pins or alternatively as the address, data and control signals for a multiplexed expansion bus. Address and data are multiplexed on Ports A and B. The control pin functions are dependent on the operating mode and the control registers PEAR and MODE. The initial state of bits in the PEAR and MODE registers are also established during reset to configure various aspects of the expansion

Freescale Semiconductor, Inc.

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

bus. After the system is running, application software can access the PEAR and MODE registers to modify the expansion bus configuration.

Some aspects of Port E are not mode dependent. Bit 1 of Port E is a general purpose input or the $\overline{IRQ}$ interrupt input. $\overline{IRQ}$ can be enabled by bits in the CPUs condition codes register but it is inhibited at reset so this pin is initially configured as a simple input with a pullup. Bit 0 of Port E is a general purpose input or the $\overline{XIRQ}$ interrupt input. $\overline{XIRQ}$ can be enabled by bits in the CPUs condition codes register but it is inhibited at reset so this pin is initially configured as a simple input with a pullup. The ESTR bit in the EBICTL register is set to one by reset in any user mode. This assures that the reset vector can be fetched even if it is located in an external slow memory device. The PE6/MODB/IPIPE1 and PE5/MODA/IPIPE0 pins act as high-impedance mode select inputs during reset.

The following paragraphs discuss the default bus setup and describe which aspects of the bus can be changed after reset on a per mode basis.

Normal Operating Modes

These modes provide three operating configurations. Background debug is available in all three modes, but must first be enabled for some operations by means of a BDM background command, then activated.

Normal Single-Chip Mode

There is no external expansion bus in this mode. All pins of Ports A, B and K are configured as general purpose I/O pins. Port E bits 1 and 0 are available as general purpose input only pins with internal pullups enabled. All other pins of Port E are bidirectional I/O pins that are initially configured as high-impedance inputs with internal pullups enabled.

The pins associated with Port E bits 7, 6, 5, 3, and 2 cannot be configured for their alternate functions IPIPE1, IPIPE0, $\overline{LSTRB}$, and R/$\overline{W}$ while the MCU is in single chip modes. The associated control bits PIPOE, LSTRE, and RDWE are reset to zero. Writing the opposite state into them in single chip mode does not change the operation of the associated Port E pins.

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

Freescale Semiconductor, Inc.

In normal single chip mode, the MODE register is writable one time. This allows a user program to change the bus mode to narrow or wide expanded mode and/or turn on visibility of internal accesses.

Port E, bit 4 can be configured for a free-running E clock output by clearing NECLK in the PEAR register. Typically, the only use for an E clock output while the MCU is in single chip modes would be to get a constant speed clock for use in the external application system.

Normal Expanded Wide Mode

In expanded wide modes, Ports A and B are configured as a 16-bit multiplexed address and data bus and Port E bit 4 is configured as the E clock output signal. These signals allow external memory and peripheral devices to be interfaced to the MCU.

Port E pins other than PE4/ECLK are configured as general purpose I/O pins (initially high-impedance inputs with internal pullup resistors enabled). Control bits PIPOE, NECLK, LSTRE, and RDWE in the PEAR register can be used to configure Port E pins to act as bus control outputs instead of general purpose I/O pins.

It is possible to enable the pipe status signals on Port E bits 6 and 5 by setting the PIPOE bit in PEAR, but it would be unusual to do so in this mode. Development systems where pipe status signals are monitored would typically use the special variation of this mode.

The Port E bit 2 pin can be reconfigured as the R/$\overline{W}$ bus control signal by writing “1” to the RDWE bit in PEAR. If the expanded system includes external devices that can be written, such as RAM, the RDWE bit would need to be set before any attempt to write to an external location. If there are no writable resources in the external system, PE2 can be left as a general purpose I/O pin.

The Port E bit 3 pin can be reconfigured as the $\overline{\text{LSTRB}}$ bus control signal by writing “1” to the LSTRE bit in PEAR. The default condition of this pin is a general purpose input because the $\overline{\text{LSTRB}}$ function is not needed in all expanded wide applications.

The Port E bit 4 pin is initially configured as ECLK output with stretch. The E clock output function depends upon the settings of the NECLK bit in the PEAR register, the IVIS bit in the MODE register and the ESTR bit

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

Freescale Semiconductor, Inc.

in the EBICTL register. The E clock is available for use in external select decode logic or as a constant speed clock for use in the external application system.

Normal Expanded Narrow Mode

This mode is used for lower cost production systems that use 8-bit wide external EPROMs or RAMs. Such systems take extra bus cycles to access 16-bit locations but this may be preferred over the extra cost of additional external memory devices.

Ports A and B are configured as a 16-bit address bus and Port A is multiplexed with data. Internal visibility is not available in this mode because the internal cycles would need to be split into two 8-bit cycles.

Since the PEAR register can only be written once in this mode, use care to set all bits to the desired states during the single allowed write.

The PE3/$\overline{\text{LSTRB}}$ pin is always a general purpose I/O pin in normal expanded narrow mode. Although it is possible to write the LSTRE bit in PEAR to “1” in this mode, the state of LSTRE is overridden and Port E bit 3 cannot be reconfigured as the $\overline{\text{LSTRB}}$ output.

It is possible to enable the pipe status signals on Port E bits 6 and 5 by setting the PIPOE bit in PEAR, but it would be unusual to do so in this mode. $\overline{\text{LSTRB}}$ would also be needed to fully understand system activity. Development systems where pipe status signals are monitored would typically use special expanded wide mode or occasionally special expanded narrow mode.

The PE4/ECLK pin is initially configured as ECLK output with stretch. The E clock output function depends upon the settings of the NECLK bit in the PEAR register, the IVIS bit in the MODE register and the ESTR bit in the EBICTL register. In normal expanded narrow mode, the E clock is available for use in external select decode logic or as a constant speed clock for use in the external application system.

The PE2/R/$\overline{\text{W}}$ pin is initially configured as a general purpose input with a pullup but this pin can be reconfigured as the R/$\overline{\text{W}}$ bus control signal by writing “1” to the RDWE bit in PEAR. If the expanded narrow system includes external devices that can be written such as RAM, the RDWE bit would need to be set before any attempt to write to an external

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

Freescale Semiconductor, Inc.

location. If there are no writable resources in the external system, PE2 can be left as a general purpose I/O pin.

Internal Visibility

Internal visibility is available when the MCU is operating in expanded wide modes, special test mode or emulation narrow mode. It is not available in single-chip, peripheral or normal expanded narrow modes. Internal visibility is enabled by setting the IVIS bit in the MODE register.

If an internal access is made while E, R/$\overline{W}$, and $\overline{LSTRB}$ are configured as bus control outputs and internal visibility is off (IVIS=0), E will remain low for the cycle, R/$\overline{W}$ will remain high, and address, data and the $\overline{LSTRB}$ pins will remain at their previous state.

When internal visibility is enabled (IVIS=1), certain internal cycles will be blocked from going external to prevent possible corruption of external devices. During cycles when the BDM is selected, R/$\overline{W}$ will remain high, data will maintain its previous state, and address and $\overline{LSTRB}$ pins will be updated with the internal value. During CPU no access cycles when the BDM is not driving, R/$\overline{W}$ will remain high, and address, data and the $\overline{LSTRB}$ pins will remain at their previous state.

Emulation Expanded Wide Mode

In expanded wide modes, Ports A and B are configured as a 16-bit multiplexed address and data bus and Port E provides bus control and status signals. These signals allow external memory and peripheral devices to be interfaced to the MCU. These signals can also be used by a logic analyzer to monitor the progress of application programs.

The bus control related pins in Port E (PE7/NOACC, PE6/MODB/IPIPE1, PE5/MODA/IPIPE0, PE4/ECLK, PE3/$\overline{LSTRB}$/$\overline{TAGLO}$, and PE2/R/$\overline{W}$) are all configured to serve their bus control output functions rather than general purpose I/O. Notice that writes to the bus control enable bits in the PEAR register in emulation mode are restricted.

The main difference between emulation modes and normal modes is that some of the bus control and system control signals cannot be written in emulation modes.

Freescale Semiconductor, Inc.

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

Emulation Expanded Narrow Mode

Expanded narrow modes are intended to allow connection of single 8-bit external memory devices for lower cost systems that do not need the performance of a full 16-bit external data bus. Accesses to internal resources that have been mapped external (i.e. PORTA, PORTB, DDRA, DDRB, PORTE, DDRE, PEAR, PUCR, RDRIV) will be accessed with a 16-bit data bus on Ports A and B. Accesses of 16-bit external words to addresses which are normally mapped external will be broken into two separate 8-bit accesses using Port A as an 8-bit data bus. Internal operations continue to use full 16-bit data paths. They are only visible externally as 16-bit information if IVIS=1.

Ports A and B are configured as multiplexed address and data output ports. During external accesses, address A15, data D15 and D7 are associated with PA7, address A0 is associated with PB0, and data D8 and D0 are associated with PA0. During internal visible accesses and accesses to internal resources that have been mapped external, address A15 and data D15 are associated with PA7, and address A0 and data D0 are associated with PB0.

The bus control related pins in Port E (PE7/NOACC, PE6/MODB/IPIPE1, PE5/MODA/IPIPE0, PE4/ECLK, PE3/$\overline{\text{LSTRB}}$/$\overline{\text{TAGLO}}$, and PE2/R/$\overline{\text{W}}$) are all configured to serve their bus control output functions rather than general purpose I/O. Notice that writes to the bus control enable bits in the PEAR register in emulation mode are restricted.

The main difference between emulation modes and normal modes is that some of the bus control and system control signals cannot be written in emulation modes.

Freescale Semiconductor, Inc.

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

Operating Modes

Special Operating Modes

There are two special operating modes that correspond to normal operating modes. These operating modes are commonly used in factory testing and system development.

Special Single-Chip Mode

When the MCU is reset in this mode, the background debug mode is enabled and “active”. The MCU does not fetch the reset vector and execute application code as it would in other modes. Instead, the active background debug mode is in control of CPU execution and BDM firmware is waiting for additional serial commands through the BKGD pin. When a serial command instructs the MCU to return to normal execution, the system will be configured as described below unless the reset states of internal control registers have been changed through background commands after the MCU was reset.

There is no external expansion bus after reset in this mode. Ports A and B are initially simple bidirectional I/O pins that are configured as high-impedance inputs with internal pullups disabled; however, writing to the mode select bits in the MODE register (which is allowed in special modes) can change this after reset. All of the Port E pins (except PE4/ECLK) are initially configured as general purpose high-impedance inputs with pullups enabled. PE4/ECLK is configured as the E clock output in this mode.

The pins associated with Port E bits 6, 5, 3, and 2 cannot be configured for their alternate functions IPIPE1, IPIPE0, $\overline{\text{LSTRB}}$, and R/$\overline{\text{W}}$ while the MCU is in single chip modes. The associated control bits PIPOE, LSTRE and RDWE are reset to zero. Writing the opposite value into these bits in this mode does not change the operation of the associated Port E pins.

Port E, bit 4 can be configured for a free-running E clock output by clearing NECLK in the PEAR register. Typically, the only use for an E clock output while the MCU is in single chip modes would be to get a constant speed clock for use in the external application system.

NOTE: *When the MCU starts in the special single chip mode, the INITCRM and PPAGE registers are overwritten by the secure BDM firmware. The CPU registers also overwritten by the firmware. These overwritten values are unknown and not guaranteed.*

Freescale Semiconductor, Inc.

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

Special Test Mode

This mode is intended for Motorola factory testing of the MCU. In expanded wide modes, Ports A and B are configured as a 16-bit multiplexed address and data bus and Port E provides bus control and status signals. In special test mode, the write protection of many control bits is lifted so that they can be thoroughly tested without needing to go through reset.

Test Operating Mode

There is a test operating mode in which an external master, such as an I.C. tester, can control the on-chip peripherals.

Peripheral Mode

This mode is intended for Motorola factory testing of the MCU. In this mode, the CPU is inactive and an external (tester) bus master drives address, data and bus control signals in through Ports A, B and E. In effect, the whole MCU acts as if it was a peripheral under control of an external CPU. This allows faster testing of on-chip memory and peripherals than previous testing methods. Since the mode control register is not accessible in peripheral mode, the only way to change to another mode is to reset the MCU into a different mode. Background debugging should not be used while the MCU is in special peripheral mode as internal bus conflicts between BDM and the external master can cause improper operation of both functions.

Freescale Semiconductor, Inc.

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

MODE Register (MODE)

Address Offset: $000B [1]

	Bit 7	6	5	4	3	2	1	Bit 0	
Read:	MODC	MODB	MODA	0	IVIS	0	EMK[2]	EME	
Write:									
Reset:	0	0	0	0	0	0	0	0	Special Single Chip
Reset:	0	0	1	0	1	0	1	1	Emulation Exp Nar
Reset:	0	1	0	0	1	0	0	0	Special Test
Reset:	0	1	1	0	1	0	1	1	Emulation Exp Wide
Reset:	1	0	0	0	0	0	0	0	Normal Single Chip
Reset:	1	0	1	0	0	0	0	0	Normal Exp Narrow
Reset:	1	1	0	0	0	0	0	0	Peripheral
Reset:	1	1	1	0	0	0	0	0	Normal Exp Wide

= Unimplemented

1. Register Address = Base Address (INITRG) + Address Offset

2. PK[6:0] pins are not bonded out.

Read: anytime (provided this register in the map)

Write: each bit has specific write conditions.

The MODE register is used to establish the operating mode and other miscellaneous functions (i.e. internal visibility and emulation of Port E and K).

In peripheral modes this register is not accessible but it is reset as shown to configure system features. Changes to bits in the MODE register are delayed one cycle after the write.

This register is not in the on-chip map in emulation and peripheral modes.

MODC, MODB, MODA — Mode Select bits

These bits indicate the current operating mode.

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

Freescale Semiconductor, Inc.

If MODA=1, then MODC, MODB, MODA are write never.

If MODC=MODA=0, then MODC, MODB, MODA are write anytime except that you cannot change to or from peripheral mode.

If MODC=1, MODB=0 and MODA=0, then MODC is write never, MODB, MODA are write once, except that you cannot change to peripheral, special test, special single chip, or emulation modes.

Table 21 MODC, MODB, MODA Write Capability(1)

MODC	MODB	MODA	Mode	MODx Write Capability
0	0	0	Special Single Chip	MODC, B, A write anytime but not to 110(2)
0	0	1	Emulation Exp Narrow	no write
0	1	0	Special Test	MODC, B, A write anytime but not to 110(2)
0	1	1	Emulation Exp Wide	no write
1	0	0	Normal Single Chip	MODC write never, MODB, A write once but not to 110
1	0	1	Normal Expanded Narrow (3)	no write
1	1	0	Special Peripheral	no write
1	1	1	Normal Expanded Wide (3)	no write

1. No writes to the MOD bits are allowed while operating in a SECURED mode.
2. If you are in a special single chip or special test mode and you write to this register, changing to normal single chip mode, then one allowed write to this register remains even if you write to a special mode. If you write to normal expanded or emulation mode, then no writes remain.
3. The external bus can not be used for memory expansion purpose with external memory devices, since this MCU does not have the PIX[5:0]/XADDR[19:14] pins to indicate page number.

IVIS — Internal Visibility (for both read and write accesses)

This bit determines whether internal accesses generate a bus cycle that is visible on the external bus. Refer to the page 110 about Internal Visibility.

Normal: write once

Emulation: write never

Special: write anytime

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

Freescale Semiconductor, Inc.

1 = Internal bus operations are visible on external bus.
0 = No visibility of internal bus operations on external bus.

NOTE: *The program page window ($8000–$BFFF) may not be correctly monitored with this internal visibility function, since there are no indications of PPAGE (page index register) to the outside from the MCU.*

EMK — Emulate Port K

Normal: write once

Emulation: write never

Special: write anytime

1 = If in any expanded mode or special peripheral mode, PORTK and DDRK are removed from the memory map.
0 = PORTK and DDRK are in the memory map so Port K can be used for general purpose I/O.

In single-chip modes, PORTK and DDRK are always in the map regardless of the state of this bit.

In peripheral modes, PORTK and DDRK are never in the map regardless of the state of this bit.

EME — Emulate Port E

Normal and Emulation: write never

Special: write anytime

1 = If in any expanded mode or special peripheral mode, PORTE and DDRE are removed from the memory map. Removing the registers from the map allows the user to emulate the function of these registers externally.
0 = PORTE and DDRE are in the memory map so Port E can be used for general purpose I/O.

In single-chip modes, PORTE and DDRE are always in the map regardless of the state of this bit.

Background Debug Mode

Background debug mode (BDM) is an auxiliary operating mode that is used for system development. BDM is implemented in on-chip hardware

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

Freescale Semiconductor, Inc.

and provides a full set of debug operations. Some BDM commands can be executed while the CPU is operating normally. Other BDM commands are firmware based, and require the BDM firmware to be enabled and active for execution.

In special single-chip mode, BDM is enabled and active immediately out of reset. BDM is available in all other operating modes, but must be enabled before it can be activated. BDM should not be used in special peripheral mode because of potential bus conflicts.

Once enabled, background debug mode can be made active by a serial command sent via the BKGD pin or execution of a CPU12 BGND instruction. While background debug mode is active, the CPU can interpret special debugging commands, and read and write CPU registers, peripheral registers, and locations in memory.

While BDM is active, the CPU executes code located in a small on-chip ROM mapped to addresses $FF20 to $FFFF, and BDM control registers are accessible at addresses $FF00 to $FF06. The BDM ROM replaces the regular system vectors while BDM is active. While BDM is active, the user memory from $FF00 to $FFFF is not in the map except through serial BDM commands.

Secured Mode of Operation

The device will make available a security feature preventing the unauthorized read and write of the memory contents. This feature allows:

- Protection of the contents of FLASH,
- Protection of the contents of CALRAM,
- Operation in single-chip mode,
- Operation from external memory with internal FLASH and CALRAM disabled.

The user must be reminded that part of the security must lie with the user’s code. An extreme example would be user’s code that dumps the

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

Freescale Semiconductor, Inc.

contents of the internal program. This code would defeat the purpose of security.

At the same time the user may also wish to put a “back door” in the user’s program. An example of this is the user downloads a “key” through the SCI which allows access to a programming routine that updates parameters stored in Flash EEPROM.

Securing the Microcontroller

Once the user has programmed the FLASH, the part can be secured by programming the security bits located in the FLASH module. These non-volatile bits will keep the part secured through resetting the part and through powering down the part.

The security byte resides in a portion of the Flash array.

Two bits are used for security. The state of the security bits and the resulting state of security are shown in **Table 22**. Note that there are three secured bit combinations and only one unsecured combination. The user can select any of the three combinations to secure the microcontroller.

Table 22 : Security Bits

sec1	sec0	secreq
0	0	1 (secured)
0	1	1 (secured)
1	0	0 (unsecured)
1	1	1 (secured)

NOTE: *When the MCU is in the secured state and in the special single chip mode, reading the CALRAM array returns $FFFF for any word accesses or $FF for any byte accesses.*

CAUTION: *Check the Flash section for more details on the security configuration.*

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

Freescale Semiconductor, Inc.

Operation of the Secured Microcontroller

Normal Single Chip Mode

This will be the most common usage of the secured part. Everything will appear the same as if the part was not secured with the exception of BDM operation. The BDM operation will be blocked.

Executing from External Memory

The user may wish to execute from external space with a secured microcontroller. This is accomplished by resetting directly into expanded mode. The internal FLASH and The CALRAM will be disabled. BDM operations will be blocked.

Unsecuring the Microcontroller

In order to unsecure the microcontroller, the internal FLASH must be erased. This can be done through an external program in expanded mode.

Once the user has erased the FLASH, the part can be reset into special single chip mode. This invokes a program that verifies the erasure of the internal FLASH. Once this program completes, the user can erase and program the FLASH security bits to the unsecured state. This is generally done through the BDM, but the user could also change to expanded mode (by writing the mode bits through the BDM) and jumping to an external program (again through BDM commands). Note that if the part goes through a reset before the security bits are reprogrammed to the unsecure state, the part will be secured again.

Freescale Semiconductor, Inc.

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

Operating Modes

Freescale Semiconductor, Inc.

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

Module Mapping Control (MMC)

Contents

Overview 121
Features 121
Block Diagram 122
Register Map 123
Register Descriptions 125
Functional Description 133
Memory Maps 138

Overview

The Module Mapping Control (MMC) sub-block of the Core performs all mapping and select operations for the on-chip and external memory blocks. The MMC also handles mapping functions for the system peripheral blocks and provides a global peripheral select to be decoded by the Motorola I.P. Bus when the Core is addressing a portion of the peripheral register map space. All bus-related data flow and multiplexing for the Core is handled within the MMC as well. Finally, the MMC also contains logic to determine the state of system security.

Features

- Registers for mapping of address space for on-chip RAM, CALRAM, and Flash EEPROM memory blocks and associated registers
- Memory mapping control and selection based upon address decode and system operating mode

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

Freescale Semiconductor, Inc.

- Core Address Bus control
- Core Data Bus control and multiplexing
- Core Security state decoding
- Emulation Chip Select signal generation ($\overline{ECS}$)
- Internal memory expansion
- Miscellaneous system control functions via the MISC register

Block Diagram

The block diagram of the MMC is shown in Figure 14 below.

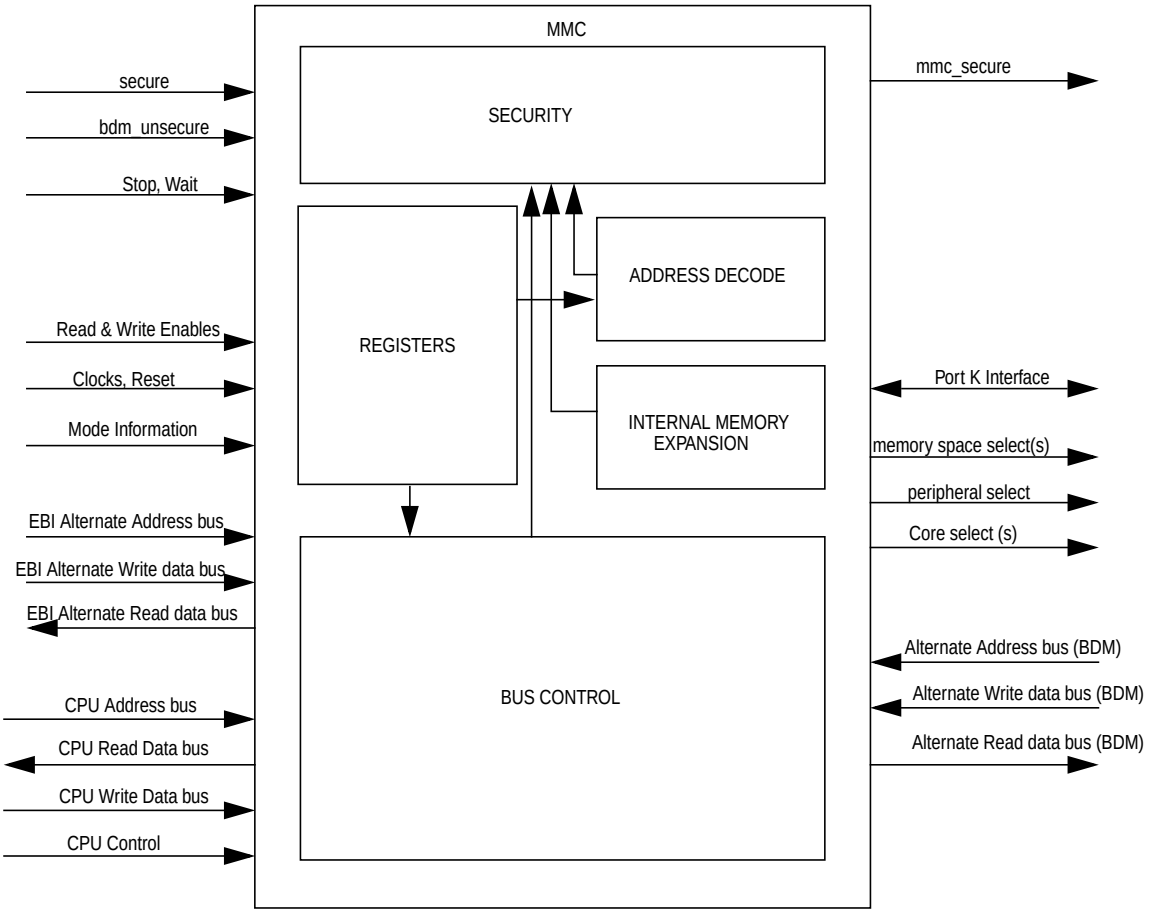


Figure 14 Module Mapping Control Block Diagram

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

Freescale Semiconductor, Inc.

Register Map

A summary of the registers associated with the MMC sub-block is shown in Figure 15 below. Detailed descriptions of the registers and bits are given in the subsections that follow.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0	Address Offset
INITRM	read	RAM15	RAM14	RAM13	RAM12	RAM11	0	0	RAMHAL	$0010
	write									
INITRG	read	0	REG14	REG13	REG12	REG11	0	0	0	$0011
	write									
INITCRM	read	CRAM15	CRAM14	CRAM13	CRAM12	CRAM11	0	0	CRAMON	$0012
	write									
MISC	read	0	0	0	0	EXSTR1	EXSTR0	ROMHM	ROMON	$0013
	write									
Reserved for Factory Test	read	Reads to this register return unpredictable values.								$0014
	write									
Reserved for Factory Test	read	Reads to this register return unpredictable values.								$0015
	write									
Reserved for Factory Test	read	Reads to this register return unpredictable values.								$0016
	write									
Reserved for Factory Test	read	Reads to this register return unpredictable values.								$0017
	write									
MEMSIZ0	read	reg_sw0	0	eep_sw1	eep_sw0	0	ram_sw2	ram_sw1	ram_sw0	$001C
	write									
MEMSIZ1	read	rom_sw1	rom_sw0	0	0	0	0	pag_sw1	pag_sw0	$001D
	write									
PPAGE	read	0	0	PIX5	PIX4	PIX3	PIX2	PIX1	PIX0	$0030
	write									
Reserved	read	0	0	0	0	0	0	0	0	$0031
	write									

= Unimplemented X = Indeterminate

Figure 15 Module Mapping Control Register Summary

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

Freescale Semiconductor, Inc.

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Freescale Semiconductor, Inc.

Register Descriptions

Initialization of Internal RAM Position Register (INITRM)

The MC9S12T64 has 2K bytes of fully static RAM that is used for storing instructions, variables, and temporary data during program execution. After reset, RAM addressing begins at location $0800 but can be assigned to any 2k boundary within the standard 64K byte address space. Mapping of internal RAM is controlled by five bits in the INITRM register and the value written in RAMHAL is always ignored.

Read: Anytime.

Write: Write once in Normal and Emulation modes. Write anytime in Special modes.

NOTE: *Writes to this register take one cycle to go into effect*.

Reset: $09 (RAM located from $0800 – $0FFF)

Address Offset: $0010

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	RAM15	RAM14	RAM13	RAM12	RAM11	0	0	RAMHAL
Write:								
Reset:	0	0	0	0	1	0	0	1

= Unimplemented

RAM[15:11] — Internal RAM map position

This register initializes the internal RAM position. The RAM15-RAM11 bits define the 2K page the RAM resides in.

RAMHAL - RAM High-align

This bit can be written and read, but is ignored in determining the RAM position.

1 = Don’t Care
0 = Don’t Care

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

Freescale Semiconductor, Inc.

Initialization of Register Block Position Register (INITRG)

This register initializes the internal register block position. Mapping of internal registers is controlled by five bits in the INITRG register. After reset the 1K byte register block resides at location $0000 but can be reassigned to any 2K byte boundary within the first 32K byte of the 64K byte address space.

Read: Anytime.

Write: Write once in Normal and Emulation modes. Write anytime in Special modes.

NOTE: *Writes to this register take one cycle to go into effect.*

Reset to $00 (Registers located from $0000 to $03FF)

Address Offset: $0011

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	REG14	REG13	REG12	REG11	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

REG[14:11] — Internal register map position

These four bits in combination with the leading zero supplied by bit 7 of INITRG determine the upper five bits of the base address for the system's internal registers (i.e. the minimum base address is $0000 and the maximum is $7FFF).

Freescale Semiconductor, Inc.

Initialization of CALRAM Position Register (INITCRM)

This register initializes the CALRAM position. Mapping of CALRAM is controlled by six bits in the INITCRM register. The MC9S12T64 has 2K bytes of CALRAM which is activated by the CRAMON bit in the INITCRM register. After reset CALRAM address space begins at location $1000 but can be mapped to any 2K byte boundary within the standard 64K byte address space.

Read: Anytime.

Write: Anytime.

NOTE: *Writes to this register take one cycle to go into effect.*

Reset: $11 (CALRAM located from $1000 – $17FF)

Address Offset: $0012

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CRAM15	CRAM14	CRAM13	CRAM12	CRAM11	0	0	CRAMON
Write:								
Reset:	0	0	0	1	0	0	0	1

= Unimplemented

CRAM[15:11] — Internal CALRAM map position

These bits specify the upper five bits of the 16-bit CALRAM address.

Read or write anytime.

CRAMON — internal CALRAM On (Enabled)

This bit enables the CALRAM in the memory map.

Read or write anytime.

1 = Place CALRAM in the memory map at the address selected by CRAM15–CRAM11.

0 = Removes the CALRAM from the map.

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

Freescale Semiconductor, Inc.

Miscellaneous System Control Register (MISC)

Additional mapping and external resource controls are available. To use external resources the part must be operated in one of the expanded modes.

Read: Anytime

Write: Refer to each bit for individual write conditions.

NOTE: *Writes to this register take one cycle to go into effect.*

Address Offset: $0013

	Bit 7	6	5	4	3	2	1	Bit 0	Mode
Read:	0	0	0	0	EXSTR1	EXSTR0	ROMHM	ROMON	
Write:									
Reset:	0	0	0	0	1	1	0	(1)	Normal Expanded or Emulation Mode
Reset:	0	0	0	0	1	1	0	1	Peripheral or Single Chip Mode
Reset:	0	0	0	0	1	1	0	0	Special Test Mode

= Unimplemented

1. Determined by state of PK7 pin during reset. See Table 24.

EXSTR1, EXSTR0 — External Access Stretch

Write: Once in Normal and Emulation modes and anytime in Special modes

This two bit field determines the amount of clock stretch on accesses to the external address space as shown in Table 23 below. In single chip and peripheral modes these bits have no meaning or effect.

Table 23 EXSTR Stretch Bit Definition

Stretch bit EXSTR1	Stretch bit EXSTR0	Number of E Clocks Stretched
0	0	0
0	1	1
1	0	2
1	1	3

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

Freescale Semiconductor, Inc.

ROMHM — Flash EEPROM only in second half of memory map

Write: Once in Normal and Emulation modes and anytime in Special modes

1 = Disables direct access to the 32K byte Flash EEPROM in location $0000 – $7FFF in the memory map. In special modes, the physical location of this 32K byte Flash can still be accessed through the Program Page window.

0 = The 32K byte Pages $3D(61) and $3E (62) of fixed Flash EEPROM in location $0000 – $7FFF can be accessed.

ROMON — Enable Flash EEPROM

Write: Once in Normal and Emulation modes and anytime in Special modes

This bit is used to enable the Flash EEPROM memory in the memory map.

In Normal Expanded or Emulation modes, the reset state of this bit is determined the state of the PK7 pin (Port K) during reset. See Table 24.

1 = Enables the Flash EEPROM in the memory map.

0 = Disables the Flash EEPROM from the memory map.

Table 24 State of ROMON bit after reset

		State of PK7 during reset	
		PK7=0	PK7=1
State of ROMON bit after reset	Normal Expanded mode	0	1
	Emulation mode	1	0

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

Freescale Semiconductor, Inc.

Memory Size Register Zero (MEMSIZ0)

The bits in this register provide read visibility to the system physical memory space allocations defined at system integration. Table 16 (page 79) in the *System Configuration* section shows the values assigned to these bits at system integration.

Read: Anytime

Write: Writes have no effect

Address Offset: $001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	reg_sw0	0	eep_sw1	eep_sw0	0	ram_sw2	ram_sw1	ram_sw0
Write:								
Reset:	0	0	0	1	0	0	0	0

= Unimplemented

reg_sw0 - Allocated System Register Space

eep_sw1:eep_sw0 - Allocated System CALRAM Memory Space

ram_sw2:ram_sw0 - Allocated System RAM Memory Space

Freescale Semiconductor, Inc.

Memory Size Register One (MEMSIZ1)

The bits in this register provide read visibility to the system physical memory space and on-chip/off-chip partitioning allocations defined at system integration. Table 16 (page 79) in the *System Configuration* section shows the values assigned to these bits at system integration.

Read: Anytime

Write: Writes have no effect

Address Offset: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	rom_sw1	rom_sw0	0	0	0	0	pag_sw1	pag_sw0
Write:								
Reset:	1	1	0	0	0	0	1	1

= Unimplemented

rom_sw1:rom_sw0 - Allocated System Flash EEPROM or ROM Physical Memory Space

pag_sw1:pag_sw0 - Allocated Off-Chip Flash EEPROM or ROM Memory Space

Freescale Semiconductor, Inc.

Program Page Index Register (PPAGE)

This register determines the 16KB active page viewed through the Program Page Window from $8000 – $BFFF. CALL and RTC instructions have a special single wire mechanism to read and write this register without using the address bus.

Read: anytime.

Write: Not writable in Normal and Emulation modes. Write anytime in Special modes.

Reset to $3C (Flash EEPROM Page $3C located from $8000 to $BFFF)

Address Offset: $0030

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	PIX5	PIX4	PIX3	PIX2	PIX1	PIX0
Write:								
Reset:	0	0	1	1	1	1	0	0

= Reserved

PIX5 – PIX0 — Program Page Index Bits 5-0

These six page index bits are used to select which of the 64 Flash EEPROM array pages is to be accessed in the Program Page Window. The 64KB address space of MC9S12T64 is divided in the four 16KB pages with their correspondent PPAGE values listed in Table 25.

CAUTION: *Proper functionality of the MCU is not guaranteed when PPAGE values other than the ones specified at Table 25,are used. Proper operation in Normal modes is not guaranteed if PPAGE values other than $3C are used.*

NOTE: *Normal writes to this register take one cycle to go into effect. Writes to this register using the special single wire mechanism of the CALL and*

Freescale Semiconductor, Inc.

RTC instructions will be complete before the end of the associated instruction.

Table 25 Program space page index in special modes

PIX5	PIX4	PIX3	PIX2	PIX1	PIX0	Program Space Selected
1	1	1	1	0	0	16K Flash EEPROM Page $3C
1	1	1	1	0	1	16K Flash EEPROM Page $3D
1	1	1	1	1	0	16K Flash EEPROM Page $3E
1	1	1	1	1	1	16K Flash EEPROM Page $3F

Functional Description

The MMC sub-block performs four basic functions of the Core operation: bus control, address decoding and select signal generation, memory expansion, and security decoding for the system. Each aspect is described in the following subsections.

Bus Control

The MMC controls the address bus and data buses that interface the Core with the rest of the system. This includes the multiplexing of the input data buses to the Core onto the main CPU read data bus and control of data flow from the CPU to the output address and data buses of the Core. In addition, the MMC handles all CPU read data bus swapping operations.

Address Decoding

As data flows on the Core address bus, the MMC decodes the address information, determines whether the internal Core register or firmware space, the peripheral space or a memory register or array space is being addressed and generates the correct select signal. This decoding operation also interprets the mode of operation of the system and the state of the mapping control registers in order to generate the proper select. The MMC also generates the Emulation Chip Select ($\overline{ECS}$) signal.

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

Freescale Semiconductor, Inc.

Select Priority

Although internal resources such as control registers and on-chip memory have default addresses, each can be relocated by changing the default values in control registers. Normally, I/O addresses, control registers, vector spaces, expansion windows, and on-chip memory are mapped so that their address ranges do not overlap. The MMC will make only one select signal active at any given time. This activation is based upon the priority outlined in Table 26 below. If two or more blocks share the same address space, only the select signal for the block with the highest priority will become active. An example of this is if the registers and the RAM are mapped to the same space, the registers will have priority over the RAM and the portion of RAM mapped in this shared space will not be accessible. The expansion windows have the lowest priority. This means that registers, vectors, and on-chip memory are always visible to a program regardless of the values in the page select registers.

Table 26 Mapping Precedence

Precedence	Resource
Highest	BDM space (Internal) when BDM is active this 256 byte block of registers and ROM appear at $FF00 – $FFFF
...	Register Space – 1K bytes fully blocked for registers
...	RAM (internal) – 2K bytes
...	CALRAM – 2K bytes
...	On-Chip Flash EEPROM – 64K bytes
Lowest	Remaining external

In expanded modes, the data registers and data directions registers for Ports A and B are removed from the on-chip memory map and become external accesses. If the EME bit in the MODE register is set, the data and data direction registers for Port E are also removed from the on-chip memory map and become external accesses.

In emulation modes, if the EMK bit in the MODE register is set, the data and data direction registers for Port K are removed from the on-chip memory map and become external accesses.

External Memory Accesses

All address space not used by internal resources is by default external memory space in expanded modes. In MC9S12T64, internal memory resources have to be disabled so that external memory can be accessed. The Flash EEPROM can be disabled by clearing the ROMON

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

Freescale Semiconductor, Inc.

in the MISC register during reset. In particular, one can also disable access to the first half of the FLASH EEPROM (locations $0000 - $7FFF) by setting the ROMHM bit. Although the CALRAM can also be disabled, ultimately it is the enabling condition FLASH EEPROM that determines if access will be external or internal. This is true because the FLASH EEPROM covers all the MCU address space and has a selection priority only superior to external accesses (See Table 26). Table 27 summarizes the conditions necessary for an access to be internal/external in expanded modes.

Table 27 Access Type in Expanded Modes

ROMON /ROMHM	MCU Address Range	Type of Access	
		Normal Mode	Special Mode
1/0	$0000–$FFFF	Internal	Internal
1/1	$0000–$7FFF	External only if internal resources cannot be accessed	External only if internal resources cannot be accessed
	$8000–$FFFF	Internal	Internal (1)
0/x	$0000–$FFFF	External only if internal resources cannot be accessed	External only if internal resources cannot be accessed

1. Accesses to all Flash pages, even to the ones disabled ($3D,$3E) by ROMHM bit, are allowed through the Program Page Window ($8000 <= MCU address <= $BFFF).

Emulation Chip Select ($\overline{ECS}$) Signal Functionality

When the EMK bit in the MODE register is set, Port K bit 7 is used as an active-low emulation chip select signal, $\overline{ECS}$. This signal is active when the system is in Emulation mode, the EMK bit is set and the Flash EEPROM or ROM space is being addressed. When the EMK bit is clear, this pin is used for general purpose I/O.The $\overline{ECS}$ signal functions based upon the assigned memory allocation. The operation of the $\overline{ECS}$ signal depends upon the state of the ROMHM bit in the MISC register. Table 28 below summarizes the functionality of these signals based upon the allocated memory configuration.

Table 28 64K Byte Physical Flash/ROM Allocated

Address Space	ROMHM	$\overline{ECS}$
$0000 - $7FFF	0	0
	1	1

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

Freescale Semiconductor, Inc.

Table 28 64K Byte Physical Flash/ROM Allocated (Continued)

Address Space	ROMHM	$\overline{ECS}$
$8000 - $BFFF	n/a	0
$C000 - $FFFF		

CALL and Return from Call Instructions

CALL and RTC are uninterruptable instructions that automate page switching in the program expansion window. CALL is similar to a JSR instruction, but the subroutine that is called can be located anywhere in the normal 64K byte address space or on any page of program expansion memory. CALL calculates and stacks a return address, stacks the current PPAGE value, and writes a new instruction-supplied value to PPAGE. The PPAGE value controls which of the 64 possible pages is visible through the 16K byte expansion window in the 64K byte memory map. Execution then begins at the address of the called subroutine.

NOTE: *The PPAGE register is not writable in Normal and Emulation modes.*

During the execution of a CALL instruction, the CPU:

- Writes the old PPAGE value into an internal temporary register and writes the new instruction-supplied PPAGE value into the PPAGE register.
- Calculates the address of the next instruction after the CALL instruction (the return address), and pushes this 16-bit value onto the stack.
- Pushes the old PPAGE value onto the stack.
- Calculates the effective address of the subroutine, refills the queue, and begins execution at the new address on the selected page of the expansion window.

This sequence is uninterruptable; there is no need to inhibit interrupts during CALL execution. A CALL can be performed from any address in memory to any other address.

The PPAGE value supplied by the instruction is part of the effective address. For all addressing mode variations except indexed-indirect

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

Freescale Semiconductor, Inc.

modes, the new page value is provided by an immediate operand in the instruction. In indexed-indirect variations of CALL, a pointer specifies memory locations where the new page value and the address of the called subroutine are stored. Using indirect addressing for both the new page value and the address within the page allows values calculated at run time rather than immediate values that must be known at the time of assembly.

The RTC instruction terminates subroutines invoked by a CALL instruction. RTC unstacks the PPAGE value and the return address and refills the queue. Execution resumes with the next instruction after the CALL.

During the execution of an RTC instruction, the CPU:

- Pulls the old PPAGE value from the stack
- Pulls the 16-bit return address from the stack and loads it into the PC
- Writes the old PPAGE value into the PPAGE register
- Refills the queue and resumes execution at the return address

This sequence is uninterruptable; an RTC can be executed from anywhere in memory, even from a different page of extended memory in the expansion window.

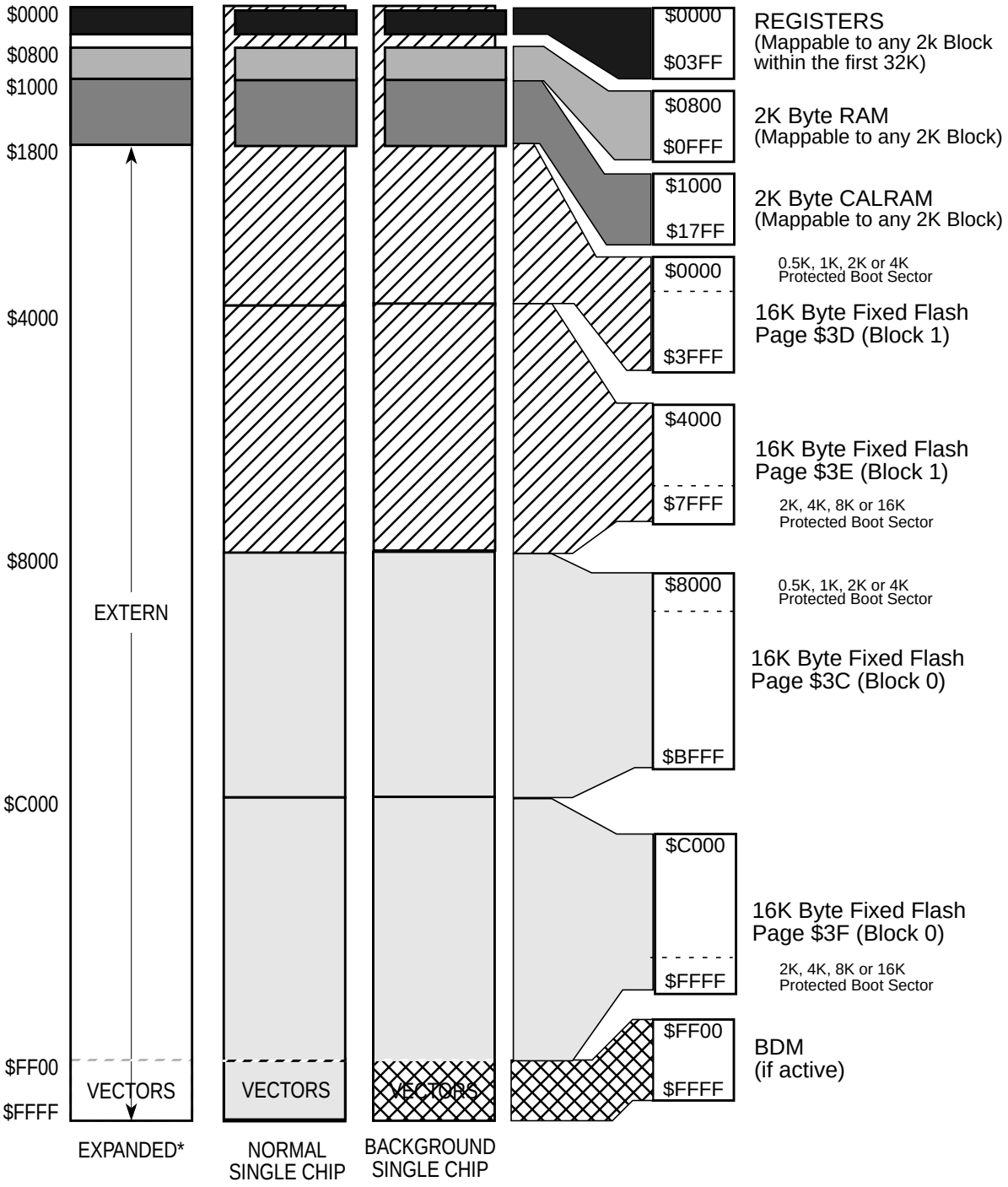
The CALL and RTC instructions behave like JSR and RTS, except they use more execution cycles. Therefore, routinely substituting CALL/RTC for JSR/RTS is not recommended. JSR and RTS can be used to access subroutines that are on the same page in expanded memory. However, a subroutine in expanded memory that can be called from other pages must be terminated with an RTC. And the RTC unstacks a PPAGE value. So any access to the subroutine, even from the same page, must use a CALL instruction so that the correct PPAGE value is in the stack.

Freescale Semiconductor, Inc.

Memory Maps

The following diagrams illustrate the memory map for each mode of operation immediately after reset.

Freescale Semiconductor, Inc.



* Assuming that a '0' was driven onto port K bit 7 during reset.

Figure 16 MC9S12T64 Memory map after reset

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

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

Multiplexed External Bus Interface (MEBI)

Contents

Overview 141
Modes of Operation 141
External Pin Descriptions 142
Register Map 145
Register Descriptions 146
Functional Description 161
Low-Power Options 167

Overview

The MEBI sub-block of the Core serves to provide access and/or visibility to internal Core data manipulation operations including timing reference information at the external boundary of the Core and/or system. Depending upon the system operating mode and the state of bits within the control registers of the MEBI, the internal 16-bit read and write data operations will be represented in 8-bit or 16-bit accesses externally. Using control information from other blocks within the system, the MEBI will determine the appropriate type of data access to be generated.

NOTE: *The internal Flash EEPROM is located in whole 64K byte memory space. Therefore the Flash EEPROM or parts of it must be disabled to connect external memory devices with the external bus.*

Modes of Operation

Refer to the Operating Modes section.

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

Freescale Semiconductor, Inc.

External Pin Descriptions

The MEBI sub-block of the Core interfaces directly with external system pins. Table 29 below outlines the pin names and functions and gives a brief description of their operation.

Table 29 External System Pins Associated With MEBI

Pin Name	Pin Functions	Description
PA7/A15/D15/D7 thru PA0/A8/D8/D0	PA7 - PA0	General purpose I/O pins, see PORTA and DDRA registers.
	A15 - A8	High-order address lines multiplexed during ECLK low. Outputs except in special peripheral mode where they are inputs from an external tester system.
	D15 - D8	High-order bidirectional data lines multiplexed during ECLK high in expanded wide modes, peripheral mode & visible internal accesses (IVIS=1) in emulation expanded narrow mode. Direction of data transfer is generally indicated by R/$\overline{W}$.
	D15/D7 thru D8/D0	Alternate high-order and low-order bytes of the bidirectional data lines multiplexed during ECLK high in expanded narrow modes and narrow accesses in wide modes. Direction of data transfer is generally indicated by R/W.
PB7/A7/D7 thru PB0/A0/D0	PB7 - PB0	General purpose I/O pins, see PORTB and DDRB registers.
	A7 - A0	Low-order address lines multiplexed during ECLK low. Outputs except in special peripheral mode where they are inputs from an external tester system.
	D7 - D0	Low-order bidirectional data lines multiplexed during ECLK high in expanded wide modes, peripheral mode & visible internal accesses (with IVIS=1) in emulation expanded narrow mode. Direction of data transfer is generally indicated by R/$\overline{W}$.
PE7/ NOACC	PE7	General purpose I/O pin, see PORTE and DDRE registers.
	NOACC	CPU No Access output. Indicates whether the current cycle is a free cycle. Only available in expanded modes.
PE6/IPIPE1/ MODB/CLKTO	MODB	At the rising edge of $\overline{RESET}$, the state of this pin is registered into the MODB bit to set the mode.
	PE6	General purpose I/O pin, see PORTE and DDRE registers.
	IPIPE1	Instruction pipe status bit 1, enabled by PIPOE bit in PEAR.

Freescale Semiconductor, Inc.

Table 29 External System Pins Associated With MEBI (Continued)

Pin Name	Pin Functions	Description
PE5/IPIPE0/ MODA	MODA	At the rising edge on $\overline{\text{RESET}}$, the state of this pin is registered into the MODA bit to set the mode.
	PE5	General purpose I/O pin, see PORTE and DDRE registers.
	IPIPE0	Instruction pipe status bit 0, enabled by PIPOE bit in PEAR.
PE4/ECLK	PE4	General purpose I/O pin, see PORTE and DDRE registers.
	ECLK	Bus timing reference clock, can operate as a free-running clock at the system clock rate or to produce one low-high clock per visible access, with the high period stretched for slow accesses. ECLK is controlled by the NECLK bit in PEAR, the IVIS bit in MODE and the ESTR bit in EBICTL.
PE3/$\overline{\text{LSTRB}}$/ $\overline{\text{TAGLO}}$	PE3	General purpose I/O pin, see PORTE and DDRE registers.
	$\overline{\text{LSTRB}}$	Low strobe bar, 0 indicates valid data on D7-D0.
	SZ8	In peripheral mode, this pin is an input indicating the size of the data transfer (0=16-bit; 1=8-bit).
	$\overline{\text{TAGLO}}$	In expanded wide mode or emulation narrow modes, when instruction tagging is on and low strobe is enabled, a 0 at the falling edge of E tags the low half of the instruction word being read into the instruction queue.
PE2/R/$\overline{\text{W}}$	PE2	General purpose I/O pin, see PORTE and DDRE registers.
	R/$\overline{\text{W}}$	Read/write, indicates the direction of internal data transfers. This is an output except in peripheral mode where it is an input.
PE1/$\overline{\text{IRQ}}$	PE1	General purpose input-only pin, can be read even if $\overline{\text{IRQ}}$ enabled.
	$\overline{\text{IRQ}}$	Maskable interrupt request, can be level sensitive or edge sensitive.
PE0/$\overline{\text{XIRQ}}$	PE0	General purpose input-only pin.
	$\overline{\text{XIRQ}}$	Non-maskable interrupt input.
PK7/$\overline{\text{ECS}}$	PK7	General purpose I/O pin, see PORTK and DDRK registers.
	$\overline{\text{ECS}}$	Emulation chip select

Freescale Semiconductor, Inc.

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

Table 29 External System Pins Associated With MEBI (Continued)

Pin Name	Pin Functions	Description
BKGD/SI/MODC /$\overline{\text{TAGHI}}$	MODC	At the rising edge on $\overline{\text{RESET}}$, the state of this pin is registered into the MODC bit to set the mode. (This pin always has an internal pullup.)
	BKGD	Pseudo-open-drain communication pin for the single-wire background debug mode. There is an internal pullup resistor on this pin.
	SI	The serial data from the host system to the FBDM uses this pin in SPI mode.
	TAGHI	When instruction tagging is on, a 0 at the falling edge of E tags the high half of the instruction word being read into the instruction queue.

Freescale Semiconductor, Inc.

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

Register Map

Register Name		Bit 7	6	5	4	3	2	1	Bit 0	Address Offset
PORTA	Read	Bit 7	6	5	4	3	2	1	Bit 0	$0000
	Write									
PORTB	Read	Bit 7	6	5	4	3	2	1	Bit 0	$0001
	Write									
DDRA	Read	Bit 7	6	5	4	3	2	1	Bit 0	$0002
	Write									
DDRB	Read	Bit 7	6	5	4	3	2	1	Bit 0	$0003
	Write									
Reserved	Read	0	0	0	0	0	0	0	0	$0004
	Write									
Reserved	Read	0	0	0	0	0	0	0	0	$0005
	Write									
Reserved	Read	0	0	0	0	0	0	0	0	$0006
	Write									
Reserved	Read	0	0	0	0	0	0	0	0	$0007
	Write									
PORTE	Read	Bit 7	6	5	4	3	2	1	Bit 0	$0008
	Write									
DDRE	Read	Bit 7	6	5	4	3	2	0	0	$0009
	Write									
PEAR	Read	NOACCE	0	PIPOE	NECLK	LSTRE	RDWE	0	0	$000A
	Write									
MODE (1)	Read	MODC	MODB	MODA	0	IVIS	0	EMK	EME	$000B
	Write									
PUCR	Read	PUPKE	0	0	PUPEE	0	0	PUPBE	PUPAE	$000C
	Write									
RDRIV	Read	RDPK	0	0	RDPE	0	0	RDPB	RDPA	$000D
	Write									
EBICTL	Read	0	0	0	0	0	0	0	ESTR	$000E
	Write									
Reserved	Read	0	0	0	0	0	0	0	0	$000F
	Write									
IRQCR (2)	Read	IRQE	IRQEN	0	0	0	0	0	0	$001E
	Write									
PORTK	Read	Bit 7	0	0	0	0	0	0	0	$0032
	Write									
DDRK	Read	Bit 7	0	0	0	0	0	0	0	$0033
	Write									

= Unimplemented X = Indeterminate

Figure 17 MEBI Register Map

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

Freescale Semiconductor, Inc.

1. Refer to the Operating Modes section for the MODE register.
2. Refer to the Resets and Interrupts section for the IRQCR register.

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Register Descriptions

Not all registers are visible in the MC9S12T64 memory map under certain conditions.

In special peripheral mode the first 16 registers associated with bus expansion are removed from the memory map.

In expanded modes, some or all of port A, port B, and port E are used for expansion buses and control signals. In order to allow emulation of the single-chip functions of these ports, some of these registers must be rebuilt in an external port replacement unit. In any expanded mode, port A, and port B, are used for address and data lines so registers for these ports, as well as the data direction registers for these ports, are removed from the on-chip memory map and become external accesses.

In any expanded mode, port E pins may be needed for bus control (e.g., ECLK, R/$\overline{W}$). To regain the single-chip functions of port E, the emulate port E (EME) control bit in the MODE register may be set. In this special case of expanded mode and EME set, PORTE and DDRE registers are removed from the on-chip memory map and become external accesses so port E may be rebuilt externally.

Freescale Semiconductor, Inc.

Port A Register (PORTA)

Address Offset: $0000

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BIT 7	6	5	4	3	2	1	BIT 0
Reset:	Unaffected by reset							
Single Chip:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Exp Wide, Emul Nar with IVIS & Periph:	ADDR15/ DATA15	ADDR14/ DATA14	ADDR13/ DATA13	ADDR12/ DATA12	ADDR11/ DATA11	ADDR10/ DATA10	ADDR9/ DATA9	ADDR8/ DATA8
Expanded narrow	ADDR15/ DATA15/ DATA7	ADDR14/ DATA14/ DATA6	ADDR13/ DATA13/ DATA5	ADDR12/ DATA12/ DATA4	ADDR11/ DATA11/ DATA3	ADDR10/ DATA10/ DATA2	ADDR9/ DATA9/ DATA1	ADDR8/ DATA8/ DATA0

Read and write: anytime (provided this register is in the map).

Port A bits 7 through 0 are associated with address lines A15 through A8 respectively and data lines D15/D7 through D8/D0 respectively. When this port is not used for external addresses such as in single-chip mode, these pins can be used as general purpose I/O. Data Direction Register A (DDRA) determines the primary direction of each pin. DDRA also determines the source of data for a read of PORTA.

This register is not in the on-chip map in expanded and peripheral modes.

CAUTION: *To ensure that you read the value present on the PORTA pins, always wait at least two cycles after writing to the DDRA register before reading from the PORTA register.*

Freescale Semiconductor, Inc.

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

Port A Data Direction Register (DDRA)

Address Offset: $0002

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BIT 7	6	5	4	3	2	1	BIT 0
Reset:	0	0	0	0	0	0	0	0

Read and write: anytime (provided this register is in the map).

This register controls the data direction for Port A. When Port A is operating as a general purpose I/O port, DDRA determines the primary direction for each Port A pin. A “1” causes the associated port pin to be an output and a “0” causes the associated pin to be a high-impedance input. The value in a DDR bit also affects the source of data for reads of the corresponding PORTA register. If the DDR bit is zero (input) the buffered pin input is read. If the DDR bit is one (output) the associated port data register bit state is read.

This register is not in the on-chip map in expanded and peripheral modes. It is reset to $00 so the DDR does not override the three-state control signals.

DDRA7–0 — Data Direction Port A

1 = Configure the corresponding I/O pin as an output

0 = Configure the corresponding I/O pin as an input

Freescale Semiconductor, Inc.

Port B Register (PORTB)

Address Offset: $0001

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BIT 7	6	5	4	3	2	1	BIT 0
Reset:	Unaffected by reset							
Single Chip:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Exp Wide, Emul Nar with IVIS & Periph:	ADDR7/ DATA7	ADDR6/ DATA6	ADDR5/ DATA5	ADDR4/ DATA4	ADDR3/ DATA3	ADDR2/ DATA2	ADDR1/ DATA1	ADDR0/ DATA0
Expanded narrow	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0

Read and write: anytime (provided this register is in the map).

Port B bits 7 through 0 are associated with address lines A7 through A0 respectively and data lines D7 through D0 respectively. When this port is not used for external addresses, such as in single-chip mode, these pins can be used as general purpose I/O. Data Direction Register B (DDRB) determines the primary direction of each pin. DDRB also determines the source of data for a read of PORTB.

This register is not in the on-chip map in expanded and peripheral modes.

CAUTION: *To ensure that you read the value present on the PORTB pins, always wait at least two cycles after writing to the DDRB register before reading from the PORTB register.*

Freescale Semiconductor, Inc.

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

Port B Data Direction Register (DDRB)

Address Offset: $0003

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BIT 7	6	5	4	3	2	1	BIT 0
Reset:	0	0	0	0	0	0	0	0

Read and write: anytime (provided this register is in the map).

This register controls the data direction for Port B. When Port B is operating as a general purpose I/O port, DDRB determines the primary direction for each Port B pin. A “1” causes the associated port pin to be an output and a “0” causes the associated pin to be a high-impedance input. The value in a DDR bit also affects the source of data for reads of the corresponding PORTB register. If the DDR bit is zero (input) the buffered pin input is read. If the DDR bit is one (output) the associated port data register bit state is read.

This register is not in the on-chip map in expanded and peripheral modes. It is reset to $00 so the DDR does not override the three-state control signals.

DDRB7–0 — Data Direction Port B

1 = Configure the corresponding I/O pin as an output

0 = Configure the corresponding I/O pin as an input

Freescale Semiconductor, Inc.

Port E Register (PORTE)

Address Offset: $0008

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BIT 7	6	5	4	3	BIT 2	BIT 1	BIT 0
Write:								
Reset:	Unaffected by reset							
Alt. Pin Function	$\overline{\text{XCLKS}}$ or NOACC	MODB or IPIPE1	MODA or IPIPE0	ECLK	LSTRB or TAGLO	R/$\overline{\text{W}}$	$\overline{\text{IRQ}}$	$\overline{\text{XIRQ}}$

= Unimplemented or reserved

Read and write: anytime (provided this register is in the map).

Port E is associated with external bus control signals and interrupt inputs. These include mode select ($\overline{\text{XCLKS}}$/NOACC, MODB/IPIPE1, MODA/IPIPE0), E clock, size ($\overline{\text{LSTRB}}$/$\overline{\text{TAGLO}}$), read / write (R/$\overline{\text{W}}$), $\overline{\text{IRQ}}$, and $\overline{\text{XIRQ}}$. When the associated pin is not used for one of these specific functions, the Port E pins 7–2 can be used as general purpose I/O and the Port E pins 1–0 can be used as general purpose input. The Port E Assignment Register (PEAR) selects the function of each pin and DDRE determines whether each pin is an input or output when it is configured to be general purpose I/O. DDRE also determines the source of data for a read of PORTE.

Some of these pins have software selectable pullups (PE7, ECLK, $\overline{\text{LSTRB}}$, R/$\overline{\text{W}}$, $\overline{\text{IRQ}}$ and $\overline{\text{XIRQ}}$). A single control bit enables the pullups for all of these pins when they are configured as inputs.

This register is not in the on-chip map in peripheral mode or in expanded modes when the EME bit is set.

CAUTION: *It is unwise to write PORTE and DRRE as a word access. If you are changing PORT E pins from being inputs to outputs, the data may have extra transitions during the write. It is best to initialize PORTE before enabling as outputs.*

CAUTION: *To ensure that you read the value present on the PORTE pins, always wait at least two cycles after writing to the DDRE register before reading from the PORTE register.*

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

Freescale Semiconductor, Inc.

Port E Data Direction Register (DDRE)

Address Offset: $0009

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BIT 7	6	5	4	3	BIT 2	BIT 1	BIT 0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or reserved

Read and write: anytime (provided this register is in the map).

Data Direction Register E is associated with Port E. For bits in Port E that are configured as general purpose I/O lines, DDRE determines the primary direction of each of these pins. A “1” causes the associated bit to be an output and a “0” causes the associated bit to be an input. Port E bit 1 (associated with $\overline{\text{IRQ}}$) and bit 0 (associated with $\overline{\text{XIRQ}}$) cannot be configured as outputs. Port E, bit 1, and bit 0 can be read regardless of whether the alternate interrupt function is enabled. The value in a DDR bit also affects the source of data for reads of the corresponding PORTE register. If the DDR bit is zero (input) the buffered pin input is read. If the DDR bit is one (output) the associated port data register bit state is read.

This register is not in the on-chip map in peripheral mode. It is also not in the map in expanded modes while the EME control bit is set.

DDRE7–2 — Data Direction Port E

1 = Configure the corresponding I/O pin as an output

0 = Configure the corresponding I/O pin as an input

CAUTION: *It is unwise to write PORTE and DRRE as a word access. If you are changing PORT E pins from being inputs to outputs, the data may have extra transitions during the write. It is best to initialize PORTE before enabling as outputs.*

Freescale Semiconductor, Inc.

Port E Assignment Register (PEAR)

Address Offset: $000A

	Bit 7	6	5	4	3	2	1	Bit 0	
Read:	NOACCE	0	PIPOE	NECLK	LSTRE	RDWE	0	0	
Write:									
Reset:	0	0	0	0	0	0	0	0	Special single chip
Reset:	0	0	1	0	1	1	0	0	Special Test
Reset:	0	0	0	0	0	0	0	0	Peripheral
Reset:	1	0	1	0	1	1	0	0	Emulation Exp Nar
Reset:	1	0	1	0	1	1	0	0	Emulation Exp Wide
Reset:	0	0	0	1	0	0	0	0	Normal Single Chip
Reset:	0	0	0	0	0	0	0	0	Normal Exp Nar
Reset:	0	0	0	0	0	0	0	0	Normal Exp Wide

= Unimplemented or reserved

Read: anytime (provided this register in the map)

Write: Each bit has specific write conditions.

Port E serves as general purpose I/O lines or as system and bus control signals. The PEAR register is used to choose between the general-purpose I/O functions and the alternate bus control functions. When an alternate control function is selected, the associated DDRE bits are overridden.

The reset condition of this register depends on the mode of operation because bus control signals are needed immediately after reset in some modes.

In normal single chip mode, no external bus control signals are needed so all of Port E is configured for general purpose I/O.

In normal expanded modes, only the E clock is configured for its alternate bus control function and the other bits of Port E are configured for general purpose I/O. As the reset vector is located in external memory, the E clock is required for this access. R/$\overline{W}$ is only

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

Freescale Semiconductor, Inc.

needed by the system when there are external writable resources. If the normal expanded system needs any other bus control signals, PEAR would need to be written before any access that needed the additional signals.

In special test and emulation modes, IPIPE1, IPIPE0, E, $\overline{\text{LSTRB}}$ and R/$\overline{\text{W}}$ are configured out of reset as bus control signals.

NOACCE — CPU No Access Output Enable

Normal: write once

Emulation: write never

Special: write anytime

1 = The associated pin (Port E bit 7) is output and indicates whether the cycle is a CPU free cycle.

0 = The associated pin (Port E bit 7) is general purpose I/O.

This bit has no effect in single chip or peripheral modes.

PIPOE — Pipe Status Signal Output Enable

Normal: write once

Emulation: write never

Special: write anytime.

1 = The associated pins (Port E bits 6:5) are outputs and indicate the state of the instruction queue

0 = The associated pins (Port E bits 6:5) are general purpose I/O.

This bit has no effect in single chip or peripheral modes.

NECLK — No External E Clock

Normal and Special: write anytime

Emulation: write never

1 = The associated pin (Port E bit 4) is a general purpose I/O pin.

0 = The associated pin (Port E bit 4) is the external E clock pin. External E clock is free-running if ESTR=0.

External E clock is available as an output in all modes.

Freescale Semiconductor, Inc.

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

LSTRE — Low Strobe ($\overline{\text{LSTRB}}$) Enable

Normal: write once

Emulation: write never

Special: write anytime.

1 = The associated pin (Port E bit 3) is configured as the $\overline{\text{LSTRB}}$ bus control output. If BDM tagging is enabled, $\overline{\text{TAGLO}}$ is multiplexed in on the rising edge of ECLK and $\overline{\text{LSTRB}}$ is driven out on the falling edge of ECLK.

0 = The associated pin (Port E bit 3) is a general purpose I/O pin.

This bit has no effect in single chip, peripheral or normal expanded narrow modes.

NOTE: *$\overline{\text{LSTRB}}$ is used during external writes. After reset in normal expanded mode, $\overline{\text{LSTRB}}$ is disabled to provide an extra I/O pin. If $\overline{\text{LSTRB}}$ is needed, it should be enabled before any external writes. External reads do not normally need $\overline{\text{LSTRB}}$ because all 16 data bits can be driven even if the MCU only needs 8 bits of data*

RDWE — Read / Write Enable

Normal: write once

Emulation: write never

Special: write anytime

1 = The associated pin (Port E bit 2) is configured as the R/$\overline{\text{W}}$ pin.

0 = The associated pin (Port E bit 2) is a general purpose I/O pin.

This bit has no effect in single chip or peripheral modes.

NOTE: *R/$\overline{\text{W}}$ is used for external writes. After reset in normal expanded mode, R/$\overline{\text{W}}$ is disabled to provide an extra I/O pin. If R/$\overline{\text{W}}$ is needed it should be enabled before any external writes.*

Freescale Semiconductor, Inc.

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

Pull-Up Control Register (PUCR)

Address Offset: $000C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PUPKE	0	0	PUPEE	0	0	PUPBE	PUPAE
Write:								
Reset:	1	0	0	1	0	0	0	0

= Unimplemented or reserved

Read and write: anytime (provided this register is in the map).

This register is used to select pullup resistors for the pins associated with the A, B, E, K ports. Pullups are assigned on a per-port basis and apply to any pin in the corresponding port that is currently configured as an input.

This register is not in the on-chip map in emulation and peripheral modes.

PUPKE — Pull-Up Port K Enable

1 = Enable pull-up devices for port K input pins.
0 = Port K pull-ups are disabled.

PUPEE — Pull-Up Port E Enable

1 = Enable pull-up devices for port E input pin bits 7, 4–0.
0 = Port E pull-ups on bit 7, 4–0 are disabled.

PUPBE — Pull-Up Port B Enable

1 = Enable pull-up devices for all port B input pins.
0 = Port B pull-ups are disabled.

PUPAE — Pull-Up Port A Enable

1 = Enable pull-up devices for all port A input pins.
0 = Port A pull-ups are disabled.

Freescale Semiconductor, Inc.

Reduced Drive of I/O Lines (RDRIV)

Address Offset: $000D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	RDPK	0	0	RDPE	0	0	RDPB	RDPA
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or reserved

Read and write: anytime (provided this register is in the map).

This register is used to select reduced drive for the pins associated with the A, B, E, K ports. This gives reduced power consumption and reduced RFI with a slight increase in transition time (depending on loading). This feature would be used on ports which have a light loading. The reduced drive function is independent of which function is being used on a particular port.

This register is not in the on-chip map in emulation and peripheral modes.

RDPK — Reduced Drive of Port K

1 = All port K output pins have reduced drive enabled.
0 = All port K output pins have full drive enabled.

RDPE — Reduced Drive of Port E

1 = All port E output pins have reduced drive enabled.
0 = All port E output pins have full drive enabled.

RDPB — Reduced Drive of Port B

1 = All port B output pins have reduced drive enabled.
0 = All port B output pins have full drive enabled.

RDPA — Reduced Drive of Port A

1 = All port A output pins have reduced drive enabled.
0 = All port A output pins have full drive enabled.

Freescale Semiconductor, Inc.

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

External Bus Interface Control (EBICTL)

Address Offset: $000E

	Bit 7	6	5	4	3	2	1	Bit 0	
Read:	0	0	0	0	0	0	0	ESTR	
Write:									
Reset:	0	0	0	0	0	0	0	0	Peripheral
Reset:	0	0	0	0	0	0	0	1	All other modes

= Unimplemented or reserved

Read: anytime (provided this register is in the map).

Write: refer to individual bit descriptions

The EBICTL register is used to control miscellaneous functions (i.e. stretching of external E clock).

This register is not in the on-chip map in peripheral mode.

ESTR — E Stretches

This control bit determines whether the E clock behaves as a simple free-running clock or as a bus control signal that is active only for external bus cycles.

Normal and Emulation: write once

Special: write anytime

1 = E stretches high during stretch cycles and low during non-visible internal accesses.

0 = E never stretches (always free running).

This bit has no effect in single chip modes.

Freescale Semiconductor, Inc.

Port K Data Register (PORTK)

Address Offset: $0032

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write	Bit 7	0	0	0	0	0	0	0
Reset:	Unaffected by reset							
Alt. pin function	$\overline{ECS}$ / ROMONE	0	0	0	0	0	0	0

= Reserved

Read and write anytime

This port is associated with the internal memory expansion emulation pins. When the port is not enabled to emulate the internal memory expansion, the port pins are used as general-purpose I/O. When Port K is operating as a general purpose I/O port, DDRK determines the primary direction for each Port K pin. A “1” causes the associated port pin to be an output and a “0” causes the associated pin to be a high-impedance input. The value in a DDR bit also affects the source of data for reads of the corresponding PORTK register. If the DDR bit is zero (input) the buffered pin input is read. If the DDR bit is one (output) the output of the port data register is read.This register is not in the map in peripheral or expanded modes while the EMK control bit in MODE register is set.

When inputs, these pins can be selected to be high impedance or pulled up, based upon the state of the PUPKE bit in the PUCR register.

Bit 7— Port K bit 7.

This bit is used as an emulation chip select signal for the emulation of the internal memory expansion, or as general purpose I/O, depending upon the state of the EMK bit in the MODE register. While this bit is used as a chip select, the external bit will return to its de-asserted state (vdd) for approximately 1/4 cycle just after the negative edge of

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

Freescale Semiconductor, Inc.

ECLK, unless the external access is stretched and ECLK is free-running (ESTR bit in EBICTL = 0). For the details see *Emulation Chip Select (ECS) Signal Functionality* in page 135.

Port K Data Direction Register (DDRK)

Address Offset: $0033

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDK7	0	0	0	0	0	0	0
Reset:	0	0	0	0	0	0	0	0

= Reserved

Read and write: anytime.
This register determines the primary direction for each port K pin configured as general-purpose I/O. This register is not in the map in peripheral or expanded modes while the EMK control bit in MODE register is set.

Bit 7 — The data direction select for Port K
1 = Associated pin is an output.
0 = Associated pin is a high-impedance input.

CAUTION: *It is unwise to write PORTK and DDRK as a word access. If you are changing Port K pins from inputs to outputs, the data may have extra transitions during the write. It is best to initialize PORTK before enabling as outputs.*

CAUTION: *To ensure that you read the correct value from the PORTK pins, always wait at least two cycles after writing to the DDRK register before reading from the PORTK register.*

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

Freescale Semiconductor, Inc.

Functional Description

There are four main sub-blocks within the MEBI: external bus control, external data bus interface, control and registers.

External Bus Control

The external bus control generates the miscellaneous control functions (pipe signals, ECLK, $\overline{\text{LSTRB}}$ and R/$\overline{\text{W}}$) that will be sent external on Port E, bits 6-2. It also generates the external addresses.

External Data Bus Interface

The external data bus interface block manages data transfers from/to the external pins to/from the internal read and write data buses. This block selectively couples 8-bit or 16-bit data to the internal data bus to implement a variety of data transfers including 8-bit, 16-bit, 16-bit swapped and 8-bit external to 16-bit internal accesses. Modes, addresses, chip selects, etc. affect the type of accesses performed during each bus cycle.

Control

The control block generates the register read/write control signals and miscellaneous port control signals.

Registers

The register block includes the fourteen 8-bit registers and five reserved register locations associated with the MEBI sub-block.

Detecting Access Type from External Signals

The external signals $\overline{\text{LSTRB}}$, R/$\overline{\text{W}}$, and A0 indicate the type of bus access that is taking place. Accesses to the internal RAM module are the only type of access that produce $\overline{\text{LSTRB}}$ = A0 = 1, because the internal RAM (except CALRAM) is specifically designed to allow misaligned 16-bit accesses in a single cycle. In these cases the data for the address that was accessed is on the low half of the data bus and the data for address + 1 is on the high half of the data bus.

Freescale Semiconductor, Inc.

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

Table 30 Access Type vs. Bus Control Pins

LSTRB	A0	R/$\overline{W}$	Type of Access
1	0	1	8-bit read of an even address
0	1	1	8-bit read of an odd address
1	0	0	8-bit write of an even address
0	1	0	8-bit write of an odd address
0	0	1	16-bit read of an even address
1	1	1	16-bit read of an odd address (low/high data swapped)
0	0	0	16-bit write to an even address
1	1	0	16-bit write to an odd address (low/high data swapped)

Stretched Bus Cycles

In order to allow fast internal bus cycles to coexist in a system with slower external memory resources, the HCS12 supports the concept of stretched bus cycles (module timing reference clocks for timers and baud rate generators are not affected by this stretching). Control bits in the MISC register specify the amount of stretch (0, 1, 2, or 3 periods of the internal bus-rate clock). While stretching, the CPU state machines are all held in their current state. At this point in the CPU bus cycle, write data would already be driven onto the data bus so the length of time write data is valid is extended in the case of a stretched bus cycle. Read data would not be captured by the MCU until the E clock falling edge. In the case of a stretched bus cycle, read data is not required until the specified setup time before the falling edge of the stretched E clock. The external address and R/W signals remain valid during the period of stretching (throughout the stretched E high time).

Internal Visibility

Internal visibility is available when the system is operating in expanded wide modes, special test mode, or emulation narrow mode. It is not available in single-chip, peripheral or normal expanded narrow modes. Internal visibility is enabled by setting the IVIS bit in the MODE register.

If an internal access is made while E, R/$\overline{W}$, and $\overline{LSTRB}$ are configured as bus control outputs and internal visibility is off (IVIS=0), E will remain low for the cycle, R/$\overline{W}$ will remain high, and address, data and the $\overline{LSTRB}$ pins will remain at their previous state.

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

Freescale Semiconductor, Inc.

When internal visibility is enabled (IVIS=1), certain internal cycles will be blocked from going external to prevent possible corruption of external devices. Specifically, during cycles when the BDM is selected, R/$\overline{W}$ will remain high, data will maintain its previous state, and address and $\overline{LSTRB}$ pins will be updated with the internal value. During CPU no access cycles when the BDM is not driving, R/$\overline{W}$ will remain high, and address, data and the $\overline{LSTRB}$ pins will remain at their previous state.

External Visibility Of Instruction Queue

The instruction queue buffers program information and increases instruction throughput. The queue consists of three 16-bit stages. Program information is always fetched in aligned 16-bit words. Normally, at least three bytes of program information are available to the CPU when instruction execution begins.

Program information is fetched and queued a few cycles before it is used by the CPU. In order to monitor cycle-by-cycle CPU activity, it is necessary to externally reconstruct what is happening in the instruction queue.

Two external pins, IPIPE[1:0], provide time-multiplexed information about data movement in the queue and instruction execution. To complete the picture for system debugging, it is also necessary to include program information and associated addresses in the reconstructed queue.

The instruction queue and cycle-by-cycle activity can be reconstructed in real time or from trace history captured by a logic analyzer. However, neither scheme can be used to stop the CPU at a specific instruction. By the time an operation is visible outside the system, the instruction has already begun execution. A separate instruction tagging mechanism is provided for this purpose. A tag follows the information in the queue as the queue is advanced. During debugging, the CPU enters active background debug mode when a tagged instruction reaches the head of the queue, rather than executing the tagged instruction. For more information about tagging, refer to Instruction Tagging in page 544.

Instruction Queue Status Signals

The IPIPE[1:0] signals carry time-multiplexed information about data movement and instruction execution during normal operation. The

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

Freescale Semiconductor, Inc.

signals are available on two multifunctional device pins. During reset, the pins are mode-select inputs MODA and MODB. After reset, information on the pins does not become valid until an instruction reaches stage two of the queue.

To reconstruct the queue, the information carried by the status signals must be captured externally. In general, data-movement and execution-start information are considered to be distinct two-bit values, with the low bit on IPIPE0 and the high bit on IPIPE1. Data-movement information is available when E clock is high or on falling edges of the E clock; execution-start information is available when E clock is low or on rising edges of the E clock, as shown in Figure 18. Data-movement information refers to data on the bus. Execution-start information is delayed one bus cycle to guarantee the indicated opcode is in stage three. Table 31 summarizes the information encoded on the IPIPE[1:0] pins.

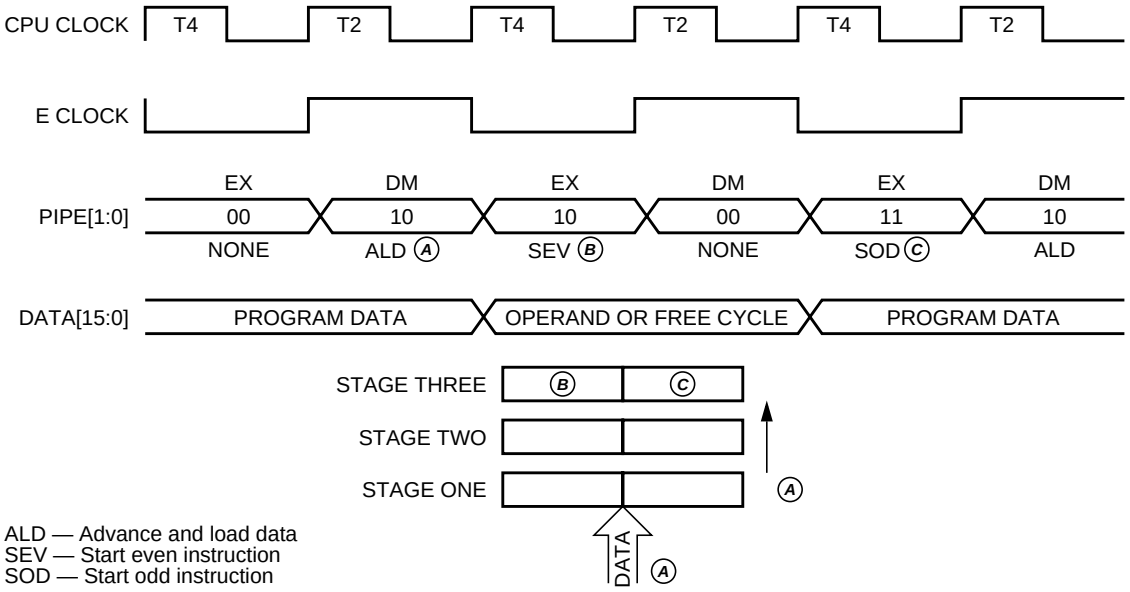


Figure 18 Queue Status Signal Timing

Data movement status is valid when the E clock is high and is represented by two states:

- No movement — There is no data shifting in the queue.

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

Freescale Semiconductor, Inc.

- Advance and load from data bus — The queue shifts up one stage with stage one being filled with the data on the read data bus.

Execution start status is valid when the E clock is low and is represented by four states:

- No start — Execution of the current instruction continues.
- Start interrupt — An interrupt sequence has begun.

NOTE: *The start-interrupt state is indicated when an interrupt request or tagged instruction alters program flow. SWI and TRAP instructions are part of normal program flow and are indicated as start even or start odd depending on their alignment. Since they are present in the queue, they can be tracked in an external queue rebuild. An external event that interrupts program flow is indeterministic. Program data is not present in the queue until after the vector jump.*

- Start even instruction — The current opcode is in the high byte of stage three of the queue.
- Start odd instruction — The current opcode is in the low byte of stage three of the queue.

Table 31 IPIPE[1:0] Decoding when E Clock is High

Data Movement (capture at E fall)	Mnemonic	Meaning
0:0	—	No movement
0:1	—	Reserved
1:0	ALD	Advance queue and load from bus
1:1	—	Reserved

Table 32 IPIPE[1:0] Decoding when E Clock is Low

Execution Start (capture at E rise)	Mnemonic	Meaning
0:0	—	No start
0:1	INT	Start interrupt sequence
1:0	SEV	Start even instruction
1:1	SOD	Start odd instruction

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

Freescale Semiconductor, Inc.

The execution-start status signals are delayed by one E clock cycle to allow a lagging program fetch and queue advance. Therefore the execution-start status always refers to the data in stage three of the queue.

The advance and load from bus signal can be used as a load-enable to capture the instruction word on the data bus. This signal is effectively the queue advance signal inside the CPU. Program data is registered into stage one on the rising edge of t4 when queue advance is asserted.

No Movement (0:0)

The 0:0 state at the falling edge of E indicates that there is no data movement in the instruction queue during the current cycle. The 0:0 state at the rising edge of E indicates continuation of an instruction or interrupt sequence during the previous cycle.

ALD — Advance and Load from Data Bus (1:0)

The three-stage instruction queue is advanced by one word and stage one is refilled with a word of program information from the data bus. The CPU requested the information two bus cycles earlier but, due to access delays, the information was not available until the E cycle immediately prior to the ALD.

INT — Start Interrupt (0:1)

This state indicates program flow has changed to an interrupt sequence. Normally this cycle is a read of the interrupt vector. However, in systems that have interrupt vectors in external memory and an 8-bit data bus, this cycle reads only the lower byte of the 16-bit interrupt vector.

SEV — Start Even Instruction (1:0)

This state indicates that the instruction is in the even (high) half of the word in stage three of the instruction queue. The queue treats the $18 prebyte of an instruction on page two of the opcode map as a special one-byte, one-cycle instruction. However, interrupts are not recognized at the boundary between the prebyte and the rest of the instruction.

SOD — Start Odd Instruction (1:1)

This state indicates that the instruction in the odd (low) half of the word in stage three of the instruction queue. The queue treats the $18 prebyte of an instruction on page two of the opcode map as a special one-byte, one-cycle instruction. However, interrupts are not recognized at the boundary between the prebyte and the rest of the instruction.

Freescale Semiconductor, Inc.

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

Low-Power Options

The MEBI does not contain any user-controlled options for reducing power consumption. The operation of the MEBI in low-power modes is discussed in the following subsections.

Run Mode

The MEBI does not contain any options for reducing power in run mode; however, the external addresses are conditioned with expanded mode to reduce power in single chip modes.

Wait Mode

The MEBI does not contain any options for reducing power in wait mode.

Stop Mode

The MEBI will cease to function during execution of a CPU STOP instruction.

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

Resets and Interrupts

Contents

Introduction 169
Register Map 170
Exception Priority 171
Maskable interrupts 171
Latching of Interrupts 172
Register Descriptions 175
Resets 177
Effects of Reset 178

Introduction

HCS12 exceptions include resets and interrupts. Each exception has an associated 16-bit vector, which points to the memory location where the routine that handles the exception is located. Vectors are stored in the upper 128 bytes of the standard 64K byte address map.

The six highest vector addresses are used for resets and non-maskable interrupt sources. The remainder of the vectors are used for maskable interrupts, and all must be initialized to point to the address of the appropriate service routine.

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

Freescale Semiconductor, Inc.

Register Map

Register name		Bit 7	6	5	4	3	2	1	Bit 0	Addr. Offset
IRQCR	Read:	IRQE	IRQEN	0	0	0	0	0	0	$001E
	Write:									
HPRIO	Read:	PSEL7	PSEL6	PSEL5	PSEL4	PSEL3	PSEL2	PSEL1	0	$001F
	Write:									

= Reserved or unimplemented

Figure 19 Resets and Interrupts Register Map

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Freescale Semiconductor, Inc.

Exception Priority

A hardware priority hierarchy determines which reset or interrupt is serviced first when simultaneous requests are made. Six sources are not maskable. The remaining sources are maskable, and any one of them can be given priority over other maskable interrupts.

The priorities of the non-maskable sources are:

1. LVD reset, POR or $\overline{\text{RESET}}$ pin
2. Clock monitor reset
3. COP watchdog reset
4. Unimplemented instruction trap
5. Software interrupt instruction (SWI)
6. $\overline{\text{XIRQ}}$ signal (if X bit in CCR = 0)

Maskable interrupts

Maskable interrupt sources include on-chip peripheral systems and external interrupt service requests. Interrupts from these sources are recognized when the global interrupt mask bit (I) in the CCR is cleared. The default state of the I bit out of reset is one, but it can be written at any time.

Interrupt sources are prioritized by default but any one maskable interrupt source may be assigned the highest priority by means of the HPRIO register. The relative priorities of the other sources remain the same.

An interrupt that is assigned highest priority is still subject to global masking by the I bit in the CCR, or by any associated local bits. Interrupt vectors are not affected by priority assignment. HPRIO can only be written while the I bit is set (interrupts inhibited). Table 33 lists interrupt sources and vectors in default order of priority. Before masking an interrupt by clearing the corresponding local enable bit, it is required to set the I-bit to avoid an SWI (Software Interrupt).

Freescale Semiconductor, Inc.

Latching of Interrupts

$\overline{\text{XIRQ}}$ is always level triggered and $\overline{\text{IRQ}}$ can be selected as a level triggered interrupt. These level triggered interrupt pins should only be released during the appropriate interrupt service routine. Generally the interrupt service routine will handshake with the interrupting logic to release the pin. In this way, the MCU will never start the interrupt service sequence only to determine that there is no longer an interrupt source. In event that this does occur the trap vector will be taken.

If $\overline{\text{IRQ}}$ is selected as an edge triggered interrupt, the hold time of the level after the active edge is independent of when the interrupt is serviced. As long as the minimum hold time is met, the interrupt will be latched inside the MCU. In this case the IRQ edge interrupt latch is cleared automatically when the interrupt is serviced.

All of the remaining interrupts are latched by the MCU with a flag bit. These interrupt flags should be cleared during an interrupt service routine or when interrupts are masked by the I bit. By doing this, the MCU will never get an unknown interrupt source and take the trap vector.

Table 33 Interrupt Vector Table

Vector Address	Interrupt Source	CCR Mask	Local Enable	HPRIO Value to Elevate
$FFFE, $FFFF	Reset (LVD, POR and $\overline{\text{RESET}}$ pin)	None	None	–
$FFFC, $FFFD	CRG Clock Monitor fail reset	None	PLLCTL (CME, SCME)	–
$FFFA, $FFFB	CRG COP failure reset	None	COPCTL (CR2-0)	–
$FFF8, $FFF9	Unimplemented instruction trap	None	None	–
$FFF6, $FFF7	SWI	None	None	–
$FFF4, $FFF5	XIRQ	X-Bit	None	–
$FFF2, $FFF3	IRQ	I-Bit	IRQCR (IRQEN)	$F2
$FFF0, $FFF1	CRG Real Time Interrupt	I-Bit	CRGINT (RTIE)	$F0
$FFEE, $FFEF	ECT channel 0	I-Bit	TIE (C0I)	$EE
$FFEC, $FFED	ECT channel 1	I-Bit	TIE (C1I)	$EC
$FFEA, $FFEB	ECT channel 2	I-Bit	TIE (C2I)	$EA
$FFE8, $FFE9	ECT channel 3	I-Bit	TIE (C3I)	$E8

Freescale Semiconductor, Inc.

Table 33 Interrupt Vector Table (Continued)

$FFE6, $FFE7	ECT channel 4	I-Bit	TIE (C4I)	$E6
$FFE4, $FFE5	ECT channel 5	I-Bit	TIE (C5I)	$E4
$FFE2, $FFE3	ECT channel 6	I-Bit	TIE (C6I)	$E2
$FFE0, $FFE1	ECT channel 7	I-Bit	TIE (C7I)	$E0
$FFDE, $FFDF	ECT overflow	I-Bit	TSCR2 (TOI)	$DE
$FFDC, $FFDD	ECT Pulse accumulator A overflow	I-Bit	PACTL (PAOVI)	$DC
$FFDA, $FFDB	ECT Pulse accumulator A input edge	I-Bit	PACTL (PAI)	$DA
$FFD8, $FFD9	SPI	I-Bit	SPICR1 (SPIE, SPTIE)	$D8
$FFD6, $FFD7	SCI 0	I-Bit	SCI0CR2 (TIE, TCIE, RIE, ILIE)	$D6
$FFD4, $FFD5	SCI 1	I-Bit	SCI1CR2 (TIE, TCIE, RIE, ILIE)	$D4
$FFD2, $FFD3	ATD	I-Bit	ATDCTL2 (ASCIE)	$D2
$FFD0, $FFD1	INTD0 reserved for future use	I-Bit	—	$D0
$FFCE, $FFCF	INTCE reserved for future use	I-Bit	—	$CE
$FFCC, $FFCD	INTCC reserved for future use	I-Bit	—	$CC
$FFCA, $FFCB	ECT Modulus down counter underflow	I-Bit	MCCTL(MCZI)	$CA
$FFC8, $FFC9	ECT Pulse accumulator B overflow	I-Bit	PBCTL(PBOVI)	$C8
$FFC6, $FFC7	CRG PLL lock	I-Bit	CRGINT (LOCKIE)	$C6
$FFC4, $FFC5	CRG Self Clock Mode	I-Bit	CRGINT (SCMIE)	$C4
$FFC2, $FFC3	INTC2 reserved for future use	I-Bit	—	$C2
$FFC0, $FFC1	INTC0 reserved for future use	I-Bit	—	$C0
$FFBE, $FFBF	INTBE reserved for future use	I-Bit	—	$BE
$FFBC, $FFBD	INTBC reserved for future use	I-Bit	—	$BC
$FFBA, $FFBB	INTBA reserved for future use	I-Bit	—	$BA
$FFB8, $FFB9	FLASH	I-Bit	FCTL(CCIE, CBEIE)	$B8
$FFB6, $FFB7	INTB6 reserved for future use	I-Bit	—	$B6
$FFB4, $FFB5	INTB4 reserved for future use	I-Bit	—	$B4
$FFB2, $FFB3	INTB2 reserved for future use	I-Bit	—	$B2
$FFB0, $FFB1	INTB0 reserved for future use	I-Bit	—	$B0
$FFAE, $FFAF	INTAE reserved for future use	I-Bit	—	$AE
$FFAC, $FFAD	INTAC reserved for future use	I-Bit	—	$AC
$FFAA, $FFAB	INTAA reserved for future use	I-Bit	—	$AA
$FFA8, $FFA9	INTA8 reserved for future use	I-Bit	—	$A8
$FFA6, $FFA7	INTA6 reserved for future use	I-Bit	—	$A6
$FFA4, $FFA5	INTA4 reserved for future use	I-Bit	—	$A4
$FFA2, $FFA3	INTA2 reserved for future use	I-Bit	—	$A2
$FFA0, $FFA1	INTA0 reserved for future use	I-Bit	—	$A0
$FF9E, $FF9F	INT9E reserved for future use	I-Bit	—	$9E
$FF9C, $FF9D	INT9C reserved for future use	I-Bit	—	$9C

Freescale Semiconductor, Inc.

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

Table 33 Interrupt Vector Table (Continued)

$FF9A, $FF9B	INT9A reserved for future use	I-Bit	—	$9A
$FF98, $FF99	INT98 reserved for future use	I-Bit	—	$98
$FF96, $FF97	INT96 reserved for future use	I-Bit	—	$96
$FF94, $FF95	INT94 reserved for future use	I-Bit	—	$94
$FF92, $FF93	INT92 reserved for future use	I-Bit	—	$92
$FF90, $FF91	INT90 reserved for future use	I-Bit	—	$90
$FF8E, $FF8F	INT8E reserved for future use	I-Bit	—	$8E
$FF8C, $FF8D	PWM Emergency Shutdown	I-Bit	PWMSDN(PWMIE)	$8C
$FF8A, $FF8B	INT8A reserved for future use	I-Bit	—	$8A
$FF88, $FF89	INT88 reserved for future use	I-Bit	—	$88
$FF86, $FF87	INT86 reserved for future use	I-Bit	—	$86
$FF84, $FF85	INT84 reserved for future use	I-Bit	—	$84
$FF82, $FF83	INT82 reserved for future use	I-Bit	—	$82
$FF80, $FF81	INT80 reserved for future use	I-Bit	—	$80
$FF10 - $FF7F	Reserved for future use (1)	—	—	—
$FF00 - $FF0F	Flash Protection/Security Field (Refer to Table 37 in page 202 for more information)			

1. This area can be used for both data and program space, however BDM firmware commands can not be used to debug the area.

Freescale Semiconductor, Inc.

Register Descriptions

Interrupt Control and Priority Register

IRQCR — IRQ Control Register

Address Offset: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IRQE	IRQEN	0	0	0	0	0	0
Write:								
Reset:	0	1	0	0	0	0	0	0

= Unimplemented or reserved

Read: refer to individual bit descriptions

Write: refer to individual bit descriptions

IRQE — IRQ Select Edge Sensitive Only

Special modes: read or write anytime

Normal & Emulation modes: read anytime, write once

1 = IRQ configured to respond only to falling edges. Falling edges on the IRQ pin will be detected anytime IRQE = 1 and will be cleared only upon a reset or the servicing of the IRQ interrupt.

0 = IRQ configured for low-level recognition.

IRQEN — External IRQ Enable

Normal, Emulation, and Special modes: read or write anytime

1 = External $\overline{\text{IRQ}}$ pin is connected to interrupt logic.

0 = External $\overline{\text{IRQ}}$ pin is disconnected from interrupt logic.

NOTE: *When IRQEN=0, the edge detect latch is disabled.*

Freescale Semiconductor, Inc.

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

HPRIO — Highest Priority I Interrupt

Address Offset: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PSEL7	PSEL6	PSEL5	PSEL4	PSEL3	PSEL2	PSEL1	0
Write:								
Reset:	1	1	1	1	0	0	1	0

= Unimplemented or reserved

READ: Anytime

WRITE: Only if I mask in CCR = 1

Determines which I maskable interrupt will be promoted to highest priority (of the I maskable interrupts). To promote an interrupt the user writes the least significant byte of the associated interrupt vector address to this register. If an unimplemented vector address or a non I-masked vector address (value higher than $F2) is written, then FFF2 will be the default highest priority interrupt.

Freescale Semiconductor, Inc.

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

Resets

There are four possible sources of reset. LVD (Low Voltage Detector) reset, Power-on reset (POR), and external reset on the $\overline{\text{RESET}}$ pin share the normal reset vector. The computer operating properly (COP) reset and the clock monitor reset each has a vector. Entry into reset is asynchronous and does not require a clock but the MCU cannot sequence out of reset without a system clock.

Power-On and LVD Resets Refer to the Clocks and Reset Generator (CRG) and the Low-Voltage Detector (LVD) sections.

External Reset Refer to the Clocks and Reset Generator (CRG) section.

COP Reset Refer to the Clocks and Reset Generator (CRG) section.

Clock Monitor Reset Refer to the Clocks and Reset Generator (CRG) section.

Freescale Semiconductor, Inc.

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

Effects of Reset

When a reset occurs, MCU registers and control bits are changed to known start-up states, as follows.

Operating Mode and Memory Map

Operating mode and default memory mapping are determined by the states of the BKGD, MODA, and MODB pins during reset. The MODA, MODB, and MODC bits in the MODE register reflect the status of the mode-select inputs at the rising edge of reset. Operating mode and default maps can subsequently be changed according to strictly defined rules.

Clock and Watchdog Control Logic

The COP watchdog system is enabled, with CR[2:0]=%011. The clock monitor is enabled. The RTIF flag is cleared and the real time interrupt is disabled. The RTR bits in the RTICTL are cleared, and must be initialized before the RTI system is used.

Interrupts

PSEL is initialized in the HPRIO register with the value $F2, causing the external $\overline{IRQ}$ pin to have the highest I-bit interrupt priority. The $\overline{IRQ}$ pin is configured for level-sensitive operation. However, the interrupt mask bits in the CPU12 CCR are set to mask X- and I-related interrupt requests.

Parallel I/O

If the MCU comes out of reset in a single-chip mode, all ports are configured as general-purpose high-impedance inputs.

If the MCU comes out of reset in an expanded mode, port A and port B are used for the address/data bus, and port E pins are normally used to control the external bus. Out of reset, port K, port E, port T, port S, port P and port AD are all configured as general-purpose inputs.

Central Processing Unit

After reset, the CPU fetches a vector from the appropriate address, then begins executing instructions. The stack pointer and other CPU registers are initialized immediately after reset. The CCR X and I interrupt mask

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

Freescale Semiconductor, Inc.

bits are set to mask any interrupt requests. The S bit is also set to inhibit the STOP instruction.

Memory

After reset, the internal register block is located from $0000 to $03FF, RAM is at $0800 to $0FFF and CALRAM is at $1000 to $17FF. In single chip mode 64K byte FLASH EEPROM module is located from $0000 to $FFFF.

Other Resources

The enhanced capture timer (ECT), pulse width modulation timer (PWM), serial communications interfaces (SCI0 and SCI1), serial peripheral interfaces (SPI), and analog-to-digital converters (ATD) are off after reset.

NOTE: *When the MCU starts in the special single chip mode, the INITCRM and PPAGE registers are overwritten by the secure BDM firmware. The CPU registers also overwritten by the firmware. These overwritten values are unknown and not guaranteed.*

Freescale Semiconductor, Inc.

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

Freescale Semiconductor, Inc.

Voltage Regulator (VREG)

Contents

Overview 181
Features 181
Modes of Operation 181
Block Diagram 183
Functional Description 184
Reset Initialization 184

Overview

The voltage regulator (VREG) converts the external VDDR (5V±5%) supply to VDD and VDDPLL (2.5V ± 10%) used to supply the internal core logic as well as PLL and clock system.

Features

- Dual linear voltage regulator with nmos output transistors
- Standby mode to minimize power consumption
- Voltage reference derived from VDDA/VSSA
- Power on reset generator

Modes of Operation

The voltage regulator has three operating modes: run, standby and disabled.

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

Freescale Semiconductor, Inc.

Normal Operation

Run Mode

In run mode, both regulating loops of the voltage regulator are active. This mode is selected whenever the CPU is neither in stop mode nor in pseudo stop mode, and VREGEN is externally connected to VDDA.

Special Operation

Standby Mode

Standby mode is selected when the CPU is in stop mode or in pseudo stop mode, and VREGEN is externally connected to VDDA. In standby mode, the gates of the power transistors are directly connected to the reference voltage V_{REF} ((VDDA - VSSA)/2), the loop amplifiers are switched off. In this case, the voltage regulator acts as a voltage clamp. In standby mode, the source resistance of the regulator is increased, but power consumption is significantly decreased.

Shutdown Mode

Shutdown mode can be selected by connecting VREGEN to VSSA. In this case, VDD and VDDPLL (2.5V 10%)must be supplied externally. In shutdown mode, VREG will also generate the power on reset signal.

Freescale Semiconductor, Inc.

Block Diagram

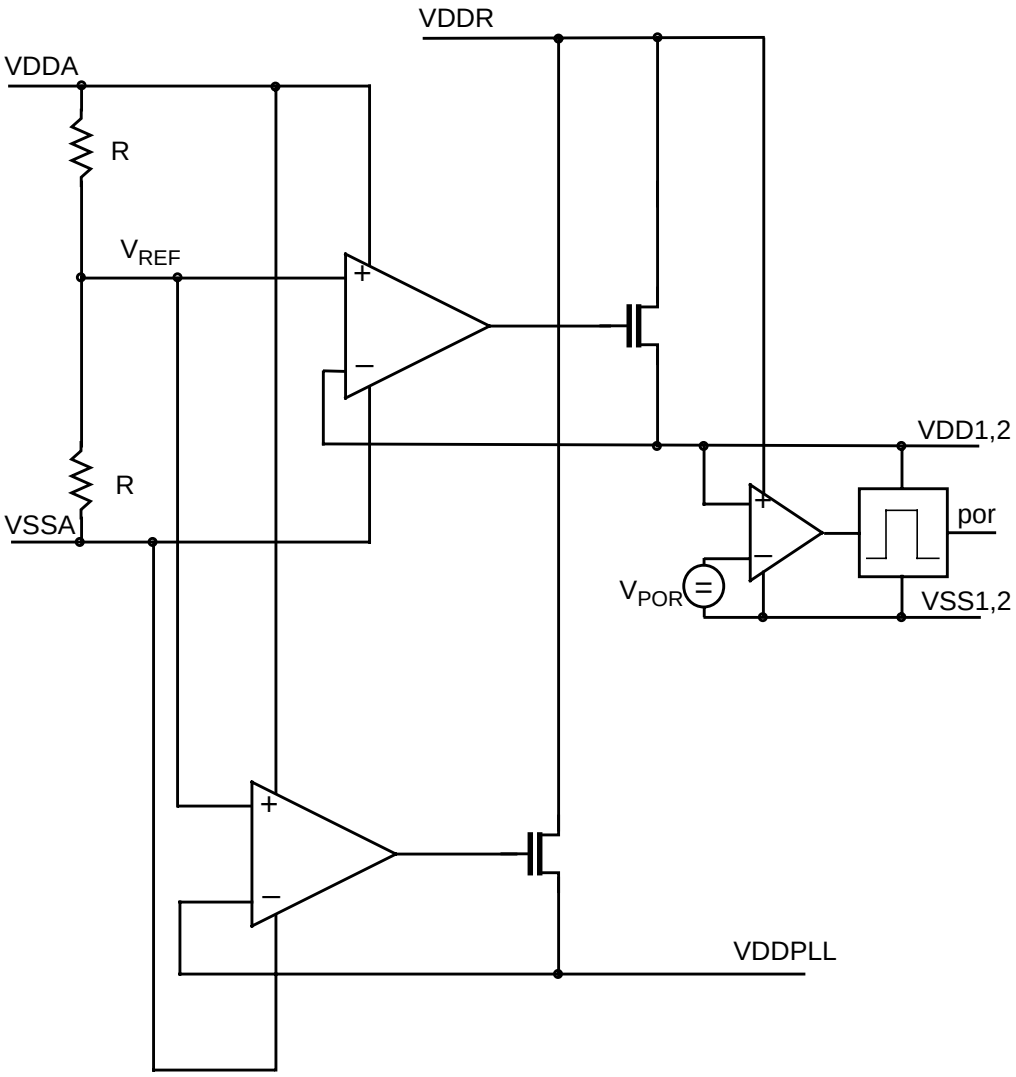


Figure 20 VREG Block Diagram

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

Functional Description

The voltage regulator module generates the supply voltage needed for the core logic as well as for the oscillator/pll section. The reference for the regulation loops are derived from a voltage divider connected between VDDA and VSSA. Both regulation loops, VDD and VDDPLL, consist of an operational amplifier driving an nmos power transistor in unit gain configuration. If there is no significant demand of output current (the CPU is in stop or pseudo stop mode) the voltage regulator is brought into standby mode, to decrease power consumption of the voltage regulator itself.

The voltage regulator can be enabled/disabled by the logic level on the VREGEN pin.

Please note VDDA, VDDR and VSSX are internally connected by anti parallel diodes.

Reset Initialization

On system power up, the voltage regulator is started in run mode if VREGEN is connected to VDDA. The LVD monitors VDDA to ensure that the MCU is not executing code while the power supply is out of specification limits to avoid erroneous operation.

Freescale Semiconductor, Inc.

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

Low-Voltage Detector (LVD)

Contents

Glossary 185
Overview 185
Features 186
Modes of Operation 186
Block Diagram 187
Register Map 188
Register Descriptions 189
Functional Description 191
Interrupts 192

Glossary

V_{LVRR} The release voltage for the LVD module. If a low voltage condition is detected, the MCU remains in reset until VDDR rises above V_{LVRR}.

V_{LVR} The trip voltage for the LVD module. Depending on its configuration, various actions can be taken by the LVD module if VDDR falls to V_{LVR} level and remains at or below that level.

Overview

The Low-Voltage Detector (LVD) module monitors the voltage on the VDDR pin and can force a reset when the VDDR voltage falls to V_{LVR} level and remains at or below that level.

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

Freescale Semiconductor, Inc.

Features

- Programmable LVD Reset.
- Programmable Power Consumption.
- Digital filtering of VDDR pin level detection

Modes of Operation

Run Mode Normal mode of operation.

The LVD module can generate a reset.

Wait Mode Not applicable

Stop Mode Not applicable

Freescale Semiconductor, Inc.

Block Diagram

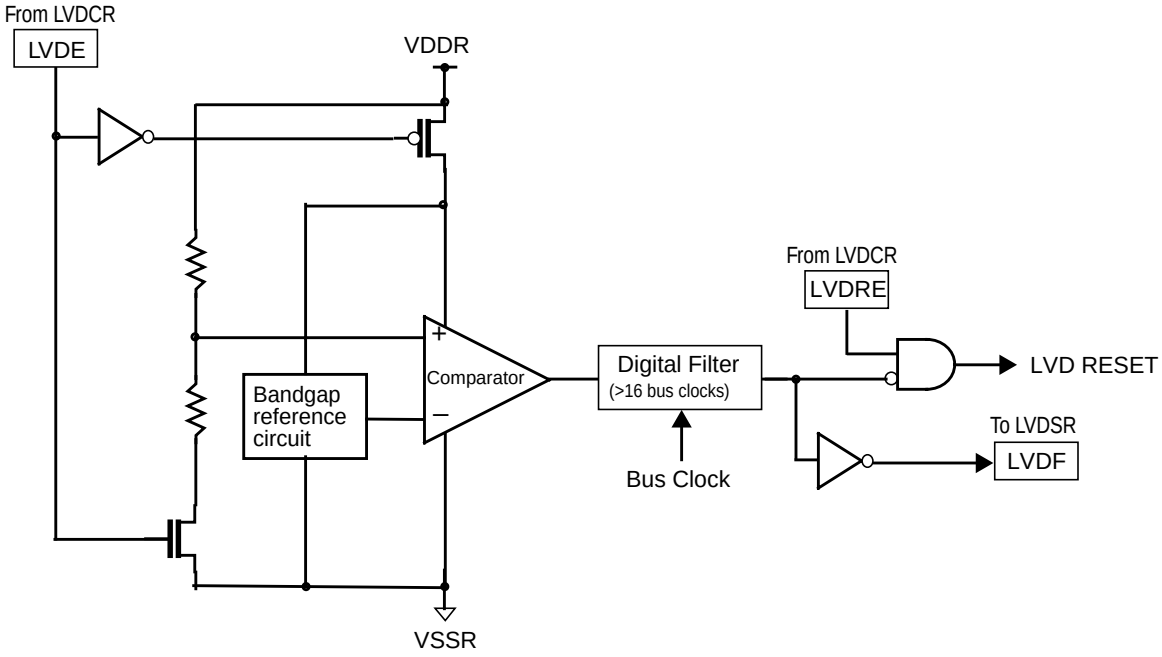


Figure 21 LVD Block Diagram

Freescale Semiconductor, Inc.

Register Map

The register map for the LVD appears below.

Register name		Bit 7	6	5	4	3	2	1	Bit 0	Addr. Offset
LVDCR	Read:	LVDE	LVDRE	0	0	0	0	0	0	$00F8
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$00F9
	Write:									
LVDSR	Read:	LVDF	0	0	0	0	0	0	0	$00FA
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$00FB
	Write:									

= Reserved or unimplemented

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Freescale Semiconductor, Inc.

Register Descriptions

LVD Control Register (LVDCR)

Address Offset: $00F8

	Bit 7	6	5	4	3	2	1	0
Read:	LVDE	LVDRE	0	0	0	0	0	0
Write:								
Power-On Reset:	1 (1)	1 (1)	0	0	0	0	0	0

= Unimplemented or reserved

1. LVDE and LVDRE bits are set when a power on reset (POR) occurs. Unaffected by non-POR resets.

Read anytime. Write anytime.

LVDE — LVD Enable bit

This LVDE bit controls whether the LVD is enabled.

1 = LVD enabled
0 = LVD disabled

LVDRE — LVD Reset Enable

The LVDRE bit controls the LVD reset if LVDE is set.

1 = LVD reset enabled
0 = LVD reset disabled

Freescale Semiconductor, Inc.

LVD Status Register (LVDSR)

Address Offset: $00FA

	Bit 7	6	5	4	3	2	1	0
Read:	LVDF	0	0	0	0	0	0	0
Write:								
Power-On Reset:	0 (1)	0	0	0	0	0	0	0

= Unimplemented or reserved

1. LVDF is cleared when a power on reset (POR) occurs. Unaffected by non-POR resets.

Read anytime. Write anytime.

LVDF — Low Voltage Detection Flag

The LVDF flag indicates the low voltage detect status when LVDE is set. The LVDF flag is cleared when a power on reset (POR) occurred. The LVDF flag is set when the VDDR voltage falls below the V_{LVR} voltage for 17 bus clock cycles (refer to Table 34). Unaffected by non-POR resets. This flag can only be cleared by writing a 1. Writing a 0 has no effect.

1 = Low voltage has been detected.

0 = Low voltage has not been detected.

Table 34 LVDF Flag Indication

VDDR		LVDF
At Level	**For Number of Bus Clock Cycles**	
VDDR > V_{LVRR}	Any	Keeps Previous Value
VDDR ≤ V_{LVR}	< 16 Bus Clock Cycles	Keeps Previous Value
	Between 16 and 17 Bus Clock Cycles	Keeps Previous Value Or Becomes “1”
	> 17 Bus Clock Cycles	Becomes “1”
V_{LVR} < VDDR < V_{LVRR}	Any	Keeps Previous Value

Freescale Semiconductor, Inc.

Functional Description

Figure 21 shows the structure of the LVD module. The LVD enable bit (LVDE) in the LVD control register (LVDCR) is set out of power-on reset, enabling the LVD to monitor the VDDR voltage. The LVD monitors the voltage on the VDDR pin by means of the bandgap reference circuit and the comparator.The LVDF flag in the LVD status register (LVDSR) is set whenever the VDDR voltage falls to V_{LVR} level and remains at or below that level for 17 or more consecutive bus clock cycles. Under such condition, the part is reset if the LVD reset enable bit (LVDRE) in LVDCR is logical 1. Once an LVD reset occurs, the MCU remains in reset until VDDR rises above the voltage V_{LVRR}.

An LVD reset also drives the $\overline{\text{RESET}}$ pin low to provide low-voltage protection to external peripheral devices.

Polled LVD Operation

In applications that can operate at VDDR levels below the V_{LVR} level, software can monitor VDDR by polling the LVDF bit. In the control register, the LVDE bit must be at logic 1 to enable the LVD module, and the LVDRE bit must be at logic 0 to disable LVD resets.

Forced Reset Operation

In applications that require VDDR to remain above the V_{LVR} level, enabling LVD resets allows the LVD module to reset the MCU when VDDR falls to the V_{LVR} level and remains at or below that level.In the control register, LVDE and LVDRE bits must be at logic 1 to enable the LVD module and to enable LVD resets.

False Reset Protections

The LVD module has two false reset protections.

1. The LVD module contains a hysteresis circuit to reduce the possibility of false resets due to power supply noise.
2. The VDDR pin level is digitally filtered to reduce false resets due to power supply noise. In order for the LVD module to reset the MCU, VDDR must fall to the V_{LVR} level and remains at or below that level for 17 or more consecutive bus clock cycles. VDDR must be above V_{LVRR} for only one bus clock cycle to bring the MCU out

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

Freescale Semiconductor, Inc.

of reset.

Interrupts

The LVD module does not generate interrupt requests.

Freescale Semiconductor, Inc.

Flash EEPROM 64K

Contents

Overview 193
Glossary 194
Features 195
Modes of Operation 195
Block Diagram 197
External Pin Descriptions 198
Module Memory Map 198
Register Descriptions 206
Functional Description 220
Low Power Options 229
Background Debug Mode 230
Flash Security 231
Reset Initialization 231
Interrupts 232

Overview

This section describes the Flash EEPROM module which is a 64k byte Flash (Non-Volatile) Memory. The Flash array is organized as 2 blocks of 32k bytes. Each block is organized as 512 rows of 64 bytes. The Flash block's erase sector size is 8 rows (512 bytes).

The Flash memory may be read as either bytes, aligned words or misaligned words. Read access time is one bus cycle for byte and aligned word, and two bus cycles for misaligned words.

Program and erase functions are controlled by a command driven interface. Both sector erase and mass erase of the entire 64k byte Flash block are supported. An erased bit reads '1' and a programmed bit reads

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

Freescale Semiconductor, Inc.

'0'. The high voltage required to program and erase is generated internally by on-chip charge pumps.

All Flash blocks can be programmed or erased at the same time, however it is not possible to read from a Flash block while it is being erased or programmed.

The Flash is ideal for program and data storage for single-supply applications allowing for field reprogramming without requiring external programming voltage sources.

WARNING: **A word must be erased before being programmed. Cumulative programming of bits within a word is not allowed.**

Glossary

Banked Register — A register operating on one Flash block which shares the same register address as the equivalent registers for the other Flash blocks. The active register bank is selected by a bank-select bit in the unbanked register space.

Common Register — A register which operates on all Flash blocks.

Command Sequence — A three-step MCU instruction sequence to program, erase or erase-verify a Flash block.

Erase Sector — 512 bytes of Flash (8 rows of 32 words)

Flash Block — 32K byte Flash macro organized as 16K by 16bit Words. Includes high voltage generation and parametric test features.

Flash Module — Includes Bus Interface, Command Control and two Flash blocks of 32K bytes.

Freescale Semiconductor, Inc.

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

Flash Page 16K bytes of Flash located in address range $0000–$3FFF, $4000–$7FFF, $8000–$FFFF or $C000–$FFFF.

Flash Super User Mode The flash super user mode allows the user to use erasing/programming sequences with FADDR/FDATA registers.

Unbanked Register A register which operates on all flash blocks.

Freescale Semiconductor, Inc.

Features

- 64K bytes of Flash memory comprising two 32K byte blocks
- Each block in the Flash module can be read, programmed or erased concurrently.
- Automated program and erase algorithm.
- Interrupts on Flash command completion and command buffer empty.
- Fast sector erase and word program operation.
- 2-stage command pipeline.
- Flexible protection scheme for protection against accidental program or erase.
- Single power supply program and erase.
- Security feature.

Modes of Operation

Secured Mode The Flash module provides the necessary security information to the rest of the chip. This information is stored within a byte in the Flash block 0 ($FF0F). This byte is read automatically after each reset and stored in a volatile register - FSEC. This information also protects the Flash

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

module from intrusive reads via the external bus interface or the background debug mode. The customer can disable the security by executing a mass erase command or by providing a 64 bit key.

Flash Super User Mode

When in Flash super user mode FADDR (FADDRHI and FADDRLO) and FDATA (FDATAHI and FDATALO) registers can be used to program Flash array area which is overlapped by the CALRAM. Refer to section *CALRAM 2K* in page 235 for details.

Freescale Semiconductor, Inc.

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

Block Diagram

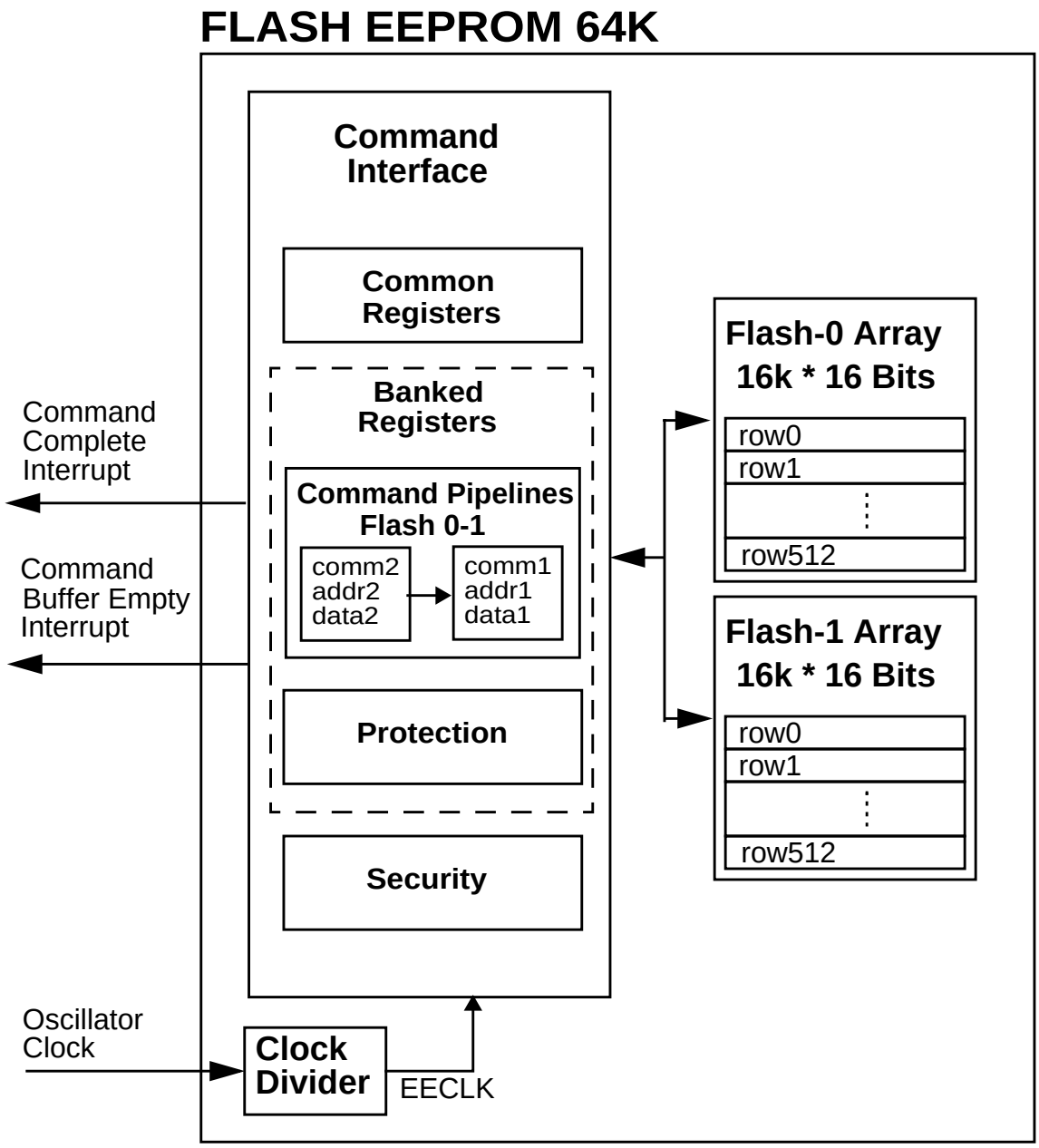


Figure 22 Flash 64K Block Diagram

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

External Pin Descriptions

This module contains no signals that connect off-chip.

Module Memory Map

Figure 23 shows the Flash memory map in normal modes. Figure 24 shows the Flash memory map in special modes. The HCS12 architecture places the Flash memory address between $0000 and $FFFF. Shown within the blocks are a protection/options field and user defined Flash protected sectors.

The FPOPEN bit in the FPROT register (see page 211) can globally protect the entirety of the memory block. However, two protected areas in each block, one starting from the Flash starting block address (called lower) towards higher addresses and the other one growing downward from the Flash block end address (called higher) can be activated. The high Flash block is mainly targeted to hold the boot loader code since it covers the vector space. All the other areas may be used to keep critical parameters.

The Flash module register space covers the addresses INITRG (Base Address) + $100 to INITRG + $10F.

Flash Data Memory Map In Normal Modes

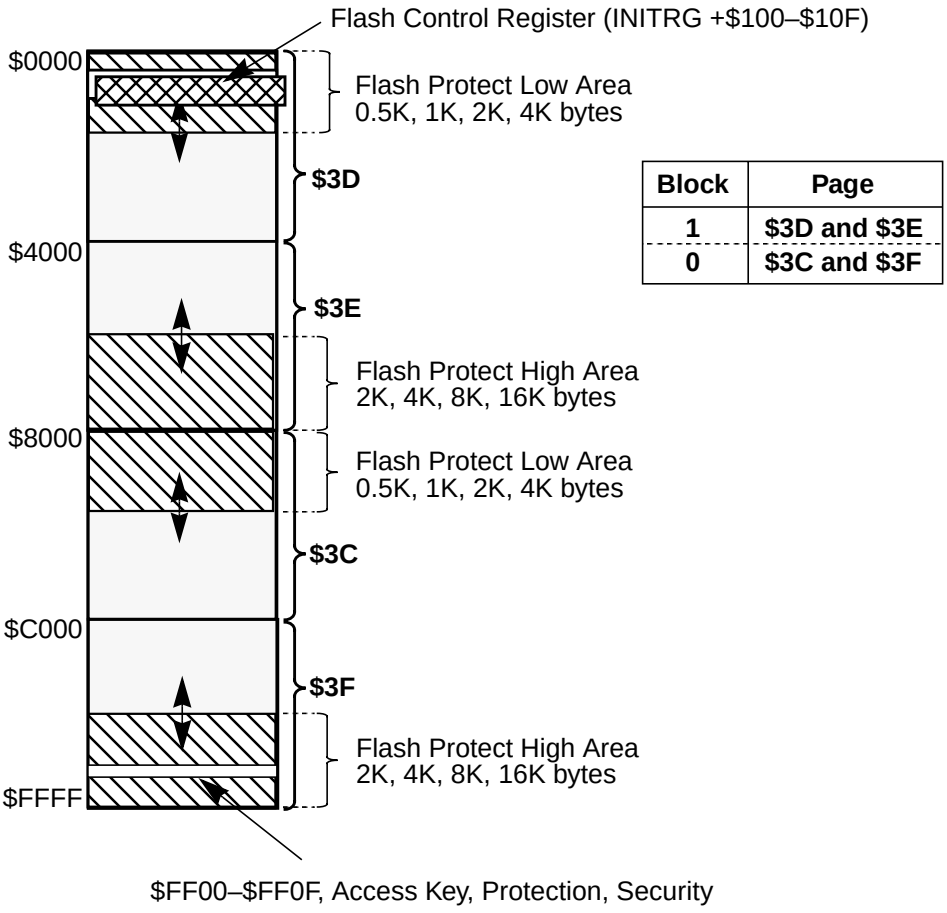
In normal modes, two blocks are mapped to fixed address since the PPAGE register always points to the page $3C and cannot be changed. See Table 35 and Figure 23.

Table 35 Flash Memory Mapping in Normal Modes

MCU Address Range	Page	Flash Block	Flash Relative Address
$0000–$3FFF	$3D	1	$0000–$3FFF
$4000–$7FFF	$3E	1	$4000–$7FFF
$8000–$BFFF	$3C (PPAGE)	0	$8000–$BFFF
$C000–$FFFF	$3F	0	$C000–$FFFF

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

Freescale Semiconductor, Inc.



Block	Page
1	$3D and $3E
0	$3C and $3F

Figure 23 Flash Data Memory Map in Normal Modes

Freescale Semiconductor, Inc.

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

Flash Data Memory Map In Special Modes

In special modes, the blocks will be mapped through an address window from $8000–$BFFF in 16K byte blocks. The additional address bits are located in the PPAGE register. See Table 36 and Figure 24.

Table 36 Flash Memory Mapping in Special Modes

MCU Address Range	Page	Flash Block	Flash Relative Address
$0000–$3FFF	$3D	1	$0000–$3FFF
$4000–$7FFF	$3E	1	$4000–$7FFF
$8000–$BFFF	PPAGE=$3D	1	$0000–$3FFF
“	PPAGE=$3E	1	$4000–$7FFF
“	PPAGE=$3C	0	$8000–$BFFF
“	PPAGE=$3F	0	$C000–$FFFF
$C000–$FFFF	$3F	0	$C000–$FFFF

Freescale Semiconductor, Inc.

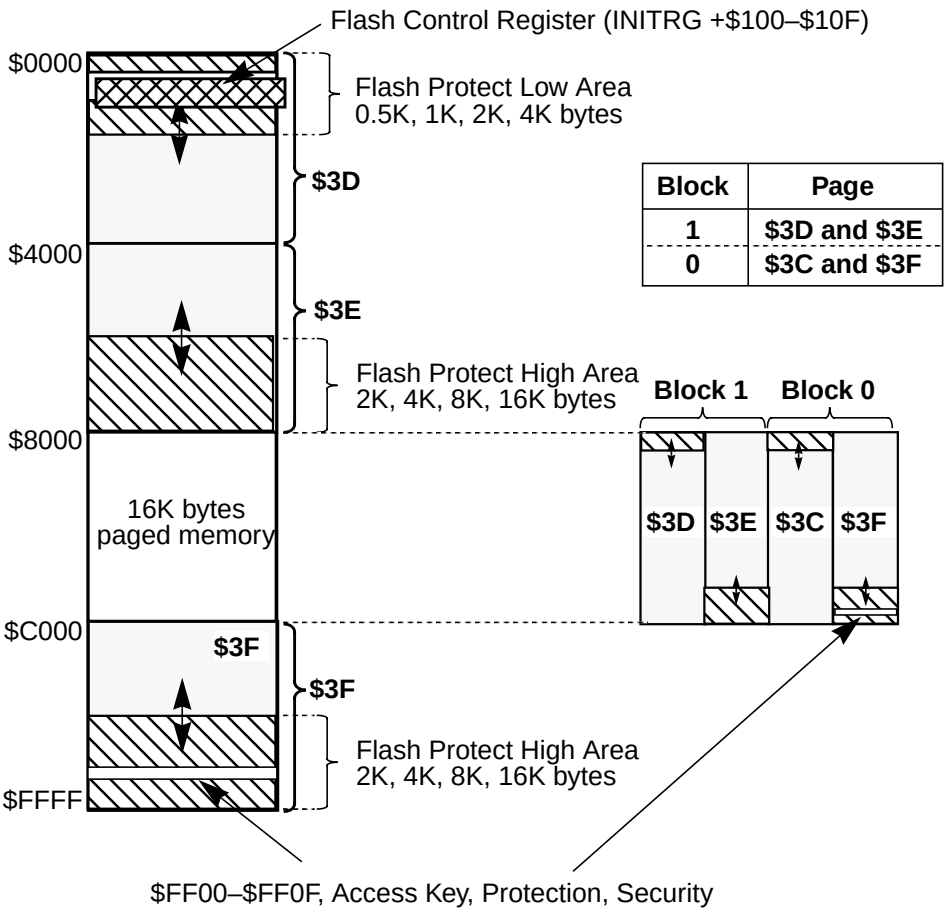


Figure 24 Flash Data Memory Map in Special Modes

Freescale Semiconductor, Inc.

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

Flash Protection Option Fields

Security information that allows the MCU to prevent intrusive access to the Flash module is stored in the Flash block's Flash Protection/Options field. A description of the 16 bytes used in this field is given in Table 37.

Table 37 Flash Protection/Security Field

Address	Size	Description
$FF00–$FF07	8	Backdoor comparison key
$FF08–$FF0B	4	Reserved
$FF0C	1	Protection byte for Flash block 1 (The byte is loaded into FPROT banked register during reset)
$FF0D	1	Protection byte for Flash block 0 (The byte is loaded into FPROT banked register during reset)
$FF0E	1	Reserved
$FF0F	1	Security Byte (The byte is loaded into FSEC register during reset)

The Flash module has hardware interlocks which protect data from accidental corruption by sectors as shown in Table 38. Flash block 0 has a protected sector located at the higher address end, just below $FFFF, and another protected sector located at the lower address end, starting at address $8000. Flash block 1 has a protected sector located at the higher address end, just below $7FFF, and another protected sector located at the lower address end, starting at address $0000.The high address protected sectors in each Flash block can be sized from 2K

Freescale Semiconductor, Inc.

bytes to 16K bytes. The low address protected sectors in each Flash block can be sized from 0.5K bytes to 4K bytes.

Table 38 Memory Map Summary In Normal Modes

MCU Address Range	Page	Protectable Low Range	Protectable High Range	Flash Block
$0000-$7FFF	$3D	$0000-$01FF $0000-$03FF $0000-$07FF $0000-$0FFF	N.A.	1
	$3E	N.A.	$7800-$7FFF $7000-$7FFF $6000-$7FFF $4000-$7FFF	
$8000-$FFFF	$3C	$8000-$81FF $8000-$83FF $8000-$87FF $8000-$8FFF	N.A.	0
	$3F	N.A.	$F800-$7FFF $F000-$7FFF $E000-$7FFF $C000-$7FFF	

NOTE: *The use of backdoor keys $0000 and $FFFF is not allowed.*

Freescale Semiconductor, Inc.

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

Register Memory Map

The Flash module also contains a set of 16 control and status registers located in address space INITRG + $100 to INITRG + $10F. In order to accommodate two Flash blocks with a minimum register address space, a set of registers (INITRG+$104 to INITRG+$10B) is duplicated in two banks. The active bank is selected by the BKSEL bit in the unbanked Flash Configuration Register (FCNFG). A summary of these registers is given in Figure 25.

Register name		Bit 7	6	5	4	3	2	1	Bit 0	Addr. Offset
FCLKDIV	Read:	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0	$0100
	Write:									
FSEC	Read:	KEYEN	NV6	NV5	NV4	NV3	NV2	SEC1	SEC0	$0101
	Write:									
FTSTMOD	Read:	Reads to this register return unpredictable values in normal modes.								$0102
	Write:									
FCNFG	Read:	CBEIE	CCIE	KEYACC	0	0	0	0	BKSEL	$0103
	Write:									
Unbanked Registers										
Banked Registers										
FPROT	Read:	FPOPEN	NV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0	$0104
	Write:									
FSTAT	Read:	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	0	0	$0105
	Write:									
FCMD	Read:	0	CMDB6	CMDB5	0	0	CMDB2	0	CMDB0	$0106
	Write:									
Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								$0107
	Write:									
FADDRHI	Read:	0	AB15	AB14	AB13	AB12	AB11	AB10	AB9	$0108
	Write:									
FADDRLO	Read:	AB8	AB7	AB6	AB5	AB4	AB3	AB2	AB1	$0109
	Write:									

= Unimplemented or reserved

Figure 25 Flash Control Register Map

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

Freescale Semiconductor, Inc.

Register name		Bit 7	6	5	4	3	2	1	Bit 0	Addr. Offset
FDATAHI	Read:	DHI7	DHI6	DHI5	DHI4	DHI3	DHI2	DHI1	DHI0	$010A
	Write:									
FDATALO	Read:	DLO7	DLO6	DLO5	DLO4	DLO3	DLO2	DLO1	DLO0	$010B
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$010C
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$010D
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$010E
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$010F
	Write:									

= Unimplemented or reserved

Figure 25 Flash Control Register Map (Continued)

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Freescale Semiconductor, Inc.

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

Register Descriptions

NOTE: *All bits of all registers in this module are completely synchronous to internal clocks during a register read.*

FCLKDIV — Flash Clock Divider Register

The FCLKDIV register is used to control timed events in program and erase algorithms. This register is unbanked.

Address Offset: $0100

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime
Write: Once in normal mode, anytime in special mode

NOTE: *Access to this register during Flash Super User mode (FSUM=1) will cause the ACCERR bit set.*

FDIVLD — Flash Clock Divider Loaded
1 = Register has been written to since the last reset.
0 = Register has not been written.

PRDIV8 — Enable Prescaler by 8
1 = Enables a prescaler by 8, to divide the Flash module input oscillator clock before feeding into the FCLKDIV divider.
0 = The input oscillator clock is directly fed into the FCLKDIV divider

FDIV[5:0] — Flash Clock Divider Bits

The combination of PRDIV8 and FDIV[5:0] effectively divides the Flash module input oscillator clock down to a frequency of 150-200 KHz. The maximum divide ratio is 512. Table 35 show some FCLKDIV settings. Please refer to Writing the FCLKDIV Register in page 220 for details about how to calculate these values.

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

Freescale Semiconductor, Inc.

Table 39 Example FCLKDIV settings

Oscillator Frequency	Bus Clock	PRDIV8	FDIV[5:0]	FCLK Frequency	FCLK Period
1.0MHz	1.0 MHz 2.0 MHz 4.0 MHz 8.0 MHz 16.0 MHz	0	000101	166.66 KHz	6us
2.0MHz	1.0 MHz	0	001011	166.66 KHz	6us
	2.0 MHz 4.0 MHz 8.0 MHz 16.0 MHz	0	001010	181.81 KHz	5.5us
4.0MHz	2.0 MHz	0	010101	181.81 KHz	5.5us
	4.0 MHz 8.0 MHz 16.0 MHz	0	010100	190.47 KHz	5.25us
16.0 MHz	8.0 MHz 16.0 MHz	1	001010	181.81 KHz	5.5us
32.0 MHz	16.0 MHz	1	010100	190.47 KHz	5.25us

Freescale Semiconductor, Inc.

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

FSEC — Flash Security Register

This FSEC register holds all bits associated with the device security. This register is unbanked.

Address Offset: $0101

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	KEYEN	NV6	NV5	NV4	NV3	NV2	SEC1	SEC0
Write:								
Reset:	F	F	F	F	F	F	F	F

= Reserved or unimplemented

Read: Anytime
Write: Never

NOTE: *Access to this register during Flash Super User mode (FSUM=1) will cause the ACCERR bit set.*

The FSEC register is loaded from the Flash Protection/Options field byte at $FF0F during the reset sequence, indicated by “F” in the reset row of the register description.

KEYEN — Enable backdoor key to security
1 = backdoor to Flash is enabled
0 = backdoor to Flash is disabled

NV[6:2] — Non Volatile Flag Bits

These 5 bits are available to the user as non-volatile flags.

SEC[1:0] — Memory Security Bits

The SEC[1:0] bits define the security state of the device as shown in Table 40. If the Flash is unsecured using the Backdoor Key Access, the SEC bits are forced to 10.

Table 40 Security States

SEC[1:0]	Description
00	secured
01	secured
10	unsecured
11	secured

Freescale Semiconductor, Inc.

The security function in the Flash module is described in page 231.

FTSTMOD — Flash Test Mode Register

The unbanked FTSTMOD register is used primarily to control the Flash Test modes.

Address Offset: $0102

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	N/A	N/A	N/A	WRALL	0	0	0	N/A
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

In normal modes, all bits in the FTSTMOD register read zero and are not writable. The WRALL bit is writable only in special modes. The purpose of this bit is to launch a command on all blocks in parallel. This can be useful for mass erase and blank check operations. All other bits in this register must be written to zero at all times.

WRALL —Write to all register banks.

If this bit is set, all banked registers sharing the same address will be written simultaneously.

1 = Write to all register banks.

0 = Write only to the bank selected via BKSEL.

FCNFG — Flash Configuration Register

The FCNFG register enables the Flash interrupts, gates the security backdoor writes and selects the register bank to be operated on. This register is not banked.

Address Offset: $0103

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CBEIE	CCIE	KEYACC	0	0	0	0	BKSEL
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime

Write: Anytime

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

Freescale Semiconductor, Inc.

NOTE: *Access to this register during Flash Super User mode (FSUM=1) will cause the ACCERR bit set.*

CBEIE — Command Buffer Empty Interrupt Enable

The CBEIE bit enables the interrupts in case of an empty command buffer in the Flash.

1 = An interrupt will be requested whenever the CBEIF flag is set
0 = Command Buffer Empty interrupts disabled

CCIE — Command Complete Interrupt Enable

The CCIE bit enables the interrupts in case of all commands being completed in the Flash.

1 = An interrupt will be requested whenever the CCIF flag is set
0 = Command Complete interrupts disabled

KEYACC — Enable Security Key Writing

1 = Writes to Flash array are interpreted as keys to open the backdoor. Reads of the Flash array return invalid data
0 = Flash writes are interpreted as the start of a program or erase sequence.

BKSEL — Register Bank Select

The BKSEL bit is used to select one among all available register banks. The register bank associated with Flash 0 is the default out of reset. The bank selection is according to Table 41.

Table 41 Register Bank Selects

BKSEL	Selected Register Bank
0	Flash block 0
1	Flash block 1

Freescale Semiconductor, Inc.

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

FPROT — Flash Protection Register

The FPROT register defines which Flash sectors are protected against program or erase. This register is banked.

Address Offset: $0104

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	FPOPEN	NV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
Reset:	F	F	F	F	F	F	F	F

= Reserved or unimplemented

The FPROT register is readable in normal and special modes. Bit NV6 is not writable. FPOPEN, FPHDIS and FPLDIS bits in the FPROT register can only be written to the protected state (i.e. 0). FPLS[1:0] can be written anytime until bit FPLDIS is cleared. FPHS[1:0] bits can be written anytime until bit FPHDIS is cleared. If the FPOPEN bit is cleared, then the state of the FPHDIS, FPHS[1:0], FPLDIS and FPLS[1:0] bits is irrelevant. The FPROT register is loaded from Flash array during reset according to the following table.

Table 42 Loading of the Protection Register from Flash

Flash Address	Protection byte for
$FF0D	Flash block 0
$FF0C	Flash block 1

NOTE: *Access to this register during Flash Super User mode (FSUM=1) will cause the ACCERR bit set.*

To change the Flash protection that will be loaded on reset, the upper sector of Flash must be unprotected, then the Flash Protect/Security byte located as described in Table 37 must be written to.

A protected Flash sector is disabled by the bits FPHDIS and FPLDIS while the size of the protected sector is defined by FPHS[1:0] and FPLS[1:0] in the FPROT register.

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

Freescale Semiconductor, Inc.

Trying to alter any of the protected areas will result in a protect violation error and bit PVIOL will be set in the Flash Status Register FSTAT. A mass erase of a whole Flash block is only possible when protection is fully disabled by setting FPLDIS and FPHDIS bits.

FPOPEN — Opens the Flash for program or erase.

1 = The Flash sectors not protected are enabled for program or erase.

0 = The whole Flash array is protected. In this case the FPHDIS, FPHS[1:0], FPLDIS and FPLS[1:0] bits within the protection register are don't care.

FPHDIS — Flash Protection Higher address range Disable

The FPHDIS bit determines whether there is a protected area in the higher space of the Flash address map.

1 = Protection disabled

0 = Protection enabled

FPHS[1:0] — Flash Protection Higher Address Size

The FPHS[1:0] bits determine the size of the protected sector. Refer to Table 43.

Table 43 Higher Address Range Protection

FPHS[1:0]	Protected Address Range		Protected Size (Bytes)
	Block 0	Block 1	
00	$F800–$FFFF	$7800–$7FFF	2K
01	$F000–$FFFF	$7000–$7FFF	4K
10	$E000–$FFFF	$6000–$7FFF	8K
11	$C000–$FFFF	$4000–$7FFF	16K

FPLDIS — Flash Protection Lower address range Disable

The FPLDIS bit determines whether there is a protected sector in the lower space of the Flash address map.

1 = Protection disabled

0 = Protection enabled

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

Freescale Semiconductor, Inc.

FPLS[1:0] — Flash Protection Lower Address Size
The FPLS[1:0] bits determine the size of the protected sector. Refer to Table 44.

Table 44 Lower Address Range Protection

FPLS[1:0]	Protected Address Range		Protected Size (Bytes)
	Block 0	Block 1	
00	$8000–$81FF	$0000–$01FF	512 Bytes
01	$8000–$83FF	$0000–$03FF	1K
10	$8000–$87FF	$0000–$07FF	2K
11	$8000–$8FFF	$0000–$0FFF	4K

NV6 — Non Volatile Flag Bit

This bit is available as non-volatile flag.

FSTAT — Flash Status Register

The FSTAT register defines the Flash state machine command status and Flash array access, protection and blank verify status. This register is banked.

Address Offset: $0105

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	0	0
Write:								
Reset:	1	1	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime
Write: Anytime to clear flags

CBEIF — Command Buffer Empty Interrupt Flag

The CBEIF flag indicates that the address, data and command buffers are empty so that a new command sequence can be started. The CBEIF flag is cleared by writing a “1” to CBEIF. Writing a “0” to the

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

Freescale Semiconductor, Inc.

CBEIF flag has no effect on CBEIF but sets ACCERR, which can be used to abort a command sequence. This bit, CBEIF, is used together with the enable bit CBEIE, to generate the interrupt request.

1 = Buffers are ready to accept a new command.

0 = Buffers are full.

CCIF — Command Complete Interrupt Flag

The CCIF flag indicates that there are no more commands pending. The CCIF flag is cleared when CBEIF is clear and sets automatically upon completion of all active and pending commands. The CCIF flag does not set when an active commands completes and a pending command is fetched from the command buffer. Writing to the CCIF flag has no effect. This bit, CCIF, is used together with the enable bit CCIE, to generate the interrupt request.

1 = All commands are completed

0 = Command in progress

PVIOL — Protection Violation

The PVIOL flag indicates an attempt was made to program or erase an address in a protected Flash memory area. The PVIOL flag is cleared by writing a “1” to PVIOL. Writing a “0” to the PVIOL flag has no effect on PVIOL. While PVIOL is set, it is not possible to launch another command. Refer to PVIOL flag set condition in page 229 for more information.

1 = A protection violation has occurred.

0 = No failure

ACCERR — Flash Access Error

The ACCERR flag indicates an illegal access to the Flash array. This can be either a violation of the command sequence, issuing an illegal command (illegal combination of the CMDBx bits in the FCMD register) or the execution of a CPU STOP instruction while a command is executing (CCIF=0). The ACCERR flag is cleared by writing a “1” to ACCERR. Writing a “0” to the ACCERR flag has no effect on ACCERR.While ACCERR is set, it is not possible to launch another command. Refer to ACCERR flag set condition in page 228

1 = Access error has occurred.

0 = No failure

Freescale Semiconductor, Inc.

BLANK — Array has been verified as erased

The BLANK flag indicates that an Erase Verify command has checked the Flash block and found it to be blank. The BLANK flag is cleared by hardware when CBEIF is cleared as part of a new valid command sequence. Writing to the BLANK flag has no effect on BLANK.

1 = Flash block verifies as erased.

0 = If an Erase Verify command has been requested, and the CCIF flag is set, then a zero in BLANK indicates the block is not erased

Freescale Semiconductor, Inc.

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

FCMD — Flash Command Register

The FCMD register defines the Flash commands. This register is banked.

Address Offset: $0106

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	CMDB6	CMDB5	0	0	CMDB2	0	CMDB0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime
Write: Only writable during a command sequence

CMDB6, 5, 2 and 0 — Valid Flash User mode commands are shown in Table 45. Any command other than those mentioned in Table 45 sets the ACCERR bit in the FSTAT register.

Table 45 Valid User Mode Commands

Command	Meaning	Remarks
$05	Erase Verify	Verify all memory bytes of the Flash block are erased. If the array is erased the BLANK bit will set in the FSTAT register upon command completion.
$20	Memory Program	Program a word (two bytes)
$40	Sector Erase	Erase 512 bytes of Flash
$41	Mass Erase	Erase all the Flash array of the Flash block. A mass erase of the full block is only possible when FPLDIS, FPHDIS and FPOPEN are set.
other	Illegal	Generate an access error

Freescale Semiconductor, Inc.

FADDR — 16-bit Address Register

FADDRHI and FADDRLO are the Flash address registers. These registers are banked.

Flash Address High (FADDRHI) Register

Address Offset: $0108

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	AB15	AB14	AB13	AB12	AB11	AB10	AB9
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Flash Address Low (FADDRLO) Register

Address Offset: $0109

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AB8	AB7	AB6	AB5	AB4	AB3	AB2	AB1
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Only in the flash super user mode. In user modes, the FADDR (FADDRHI, FADDRLO) register reads zeros.

Write: Only in the flash super user mode.

For sector erase, the MCU address bits AB[8:0] are don't care.

For mass erase, any address within the block is valid to start the command.

AB[15:1] — 64K byte Relative Address of Flash Array

These address bits are used in the program/erase sequence in the flash super user mode. Refer to Low Power Options in page 229 for the program/erase sequence.

The mapping of the address information in the FADDR register to the Flash relative address is shown in Figure 26.

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

Freescale Semiconductor, Inc.

NOTE: *The address bit AB0 is not stored, since no byte access from this register is possible. For sector erase, the MCU address bits AB[8:0] are don't care. For mass erase, any address within the block is valid to start the command.*

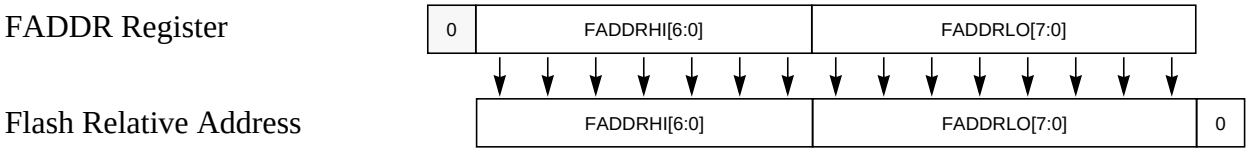


Figure 26 FADDR Address Mapping to Flash Relative Address

Freescale Semiconductor, Inc.

FDATA — Flash 16-bit Data Buffer and Register

FDATAHI and FDATALO are the Flash data registers. These registers are banked.

Flash Data High (FDATAHI) Register

Address Offset: $010A

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DHI7	DHI6	DHI5	DHI4	DHI3	DHI2	DHI1	DHI0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Flash Data Low (FDATALO) Register

Address Offset: $010B

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DLO7	DLO6	DLO5	DLO4	DLO3	DLO2	DLO1	DLO0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Only in the flash super user mode. In user modes, all FDATA bits read zero

Write: Only in the flash super user mode.

All FDATA(FDATAHI and FDATALO) bits read zero and are not writable when the Flash EEPROM module is not in the flash super user mode.

DHI[7:0] and DLO[7:0] — 16-bit Data Register

In flash super user mode, all FDATA bits are readable and writable when writing to an address within the Flash address range.

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

Freescale Semiconductor, Inc.

Functional Description

Program and Erase Operation

Write and read operations are both used for the program and erase algorithms described in this section. These algorithms are controlled by a state machine whose timebase FCLK is derived from the oscillator clock via a programmable divider. The command register as well as the associated address and data registers operate as a buffer and a register (2-stage FIFO), so that a new command along with the necessary data and address can be stored to the buffer while the previous command is still in progress. This pipelined operation allows a time optimization when programming more than one word on a specific row, as the high voltage generation can be kept ON in between two programming commands. The pipelined operation also allows a simplification of command launching. Buffer empty as well as command completion are signalled by flags in the Flash status register. Interrupts for the Flash will be generated if enabled.

The next four subsections describe:

- How to write the FCLKDIV register.
- The write sequences used to program, erase and erase-verify the Flash.
- Valid Flash commands.
- Errors resulting from illegal Flash operations.

Writing the FCLKDIV Register

Prior to issuing any program or erase command, it is first necessary to write the FCLKDIV register to divide the oscillator down to within the 150kHz to 200kHz range.The program and erase timings are also a function of the bus clock, such that the FCLKDIV determination must take this information into account. If we define:

- FCLK as the clock of the Flash timing control block
- Tbus as the period of the bus clock
- INT(x) as taking the integer part of x (e.g. INT(4.323)=4),

then FCLKDIV register bits PRDIV8 and FDIV[5:0] are to be set as described in Figure 27.

For example, if the oscillator clock frequency is 4Mz and the bus clock is 25MHz, FCLKDIV bits FDIV[5:0] should be set to 20 (010100) and bit

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

Freescale Semiconductor, Inc.

PRDIV8 set to 0. The resulting FCLK is then 190kHz. As a result, the Flash algorithm timings are increased over optimum target by:

$$(200 - 190)/200 \times 100 = 5\%$$

NOTE: Command execution time will increase proportionally with the period of FCLK.

WARNING: **Because of the impact of clock synchronization on the accuracy of the functional timings, programming or erasing the Flash cannot be performed if the bus clock runs at less than 1 MHz. Programming or erasing the Flash with an input clock < 150kHz should be avoided. Setting FCLKDIV to a value such that FCLK < 150kHz can destroy the Flash due to overstress. Setting FCLKDIV to a value such that (1/FCLK+1/(Bus Clock)) < 5μs can result in incomplete programming or erasure of the memory array cells.**

If the FCLKDIV register is written, the bit FDIVLD is set automatically. If this bit is zero, the register has not been written since the last reset. Program and erase commands will not be executed if this register has not been written to.

Freescale Semiconductor, Inc.

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

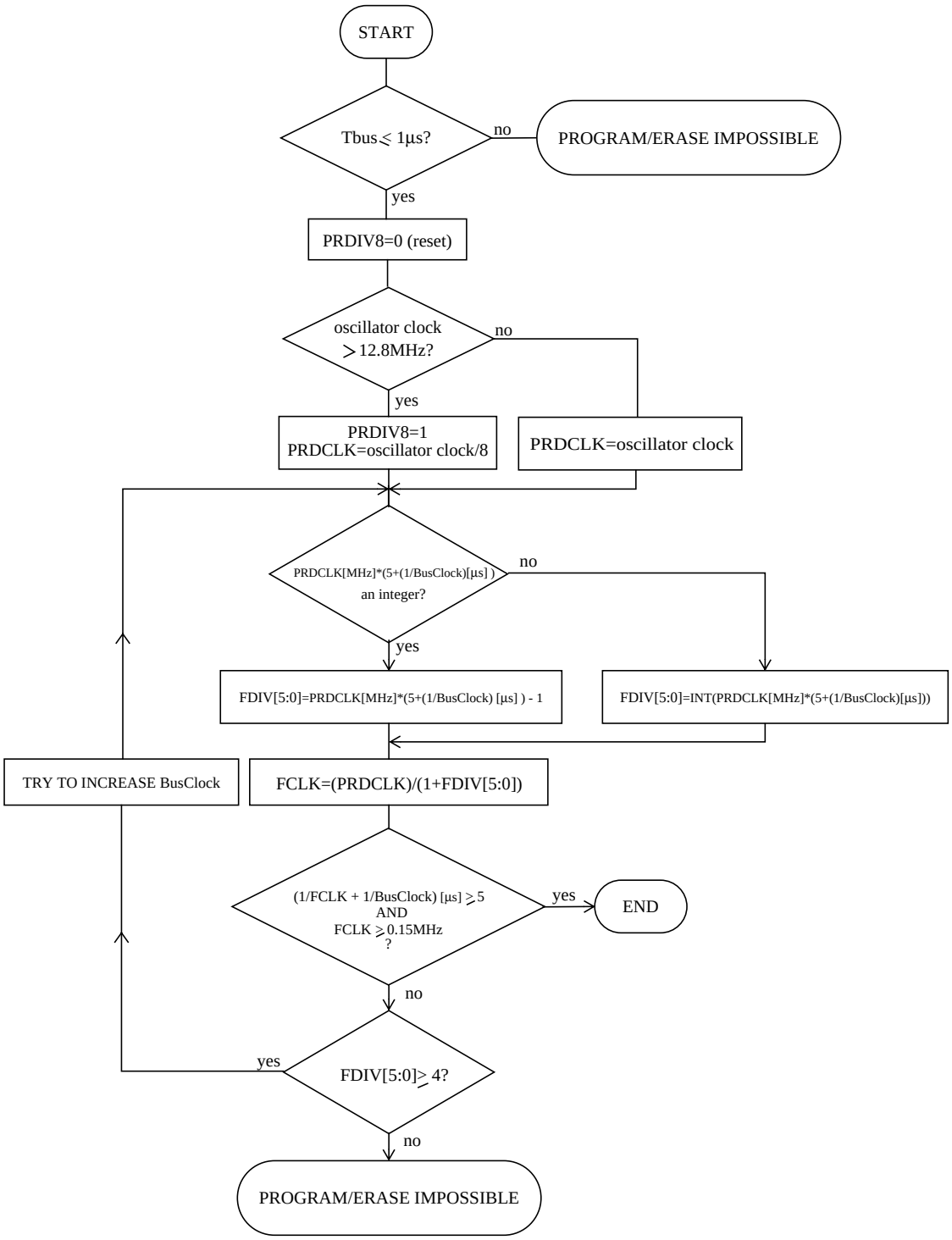


Figure 27 PRDIV8 and FDIV bits Determination Procedure

Freescale Semiconductor, Inc.

Program and Erase Sequence in Normal Mode

A Command State Machine is used to supervise the write sequencing for program and erase. The erase-verify command follows the same flow. Before starting a command sequence, it is necessary to verify that there is no pending access error or protection violation (the ACCERR and PVIOL flags should be cleared in the FSTAT register). It is then required to set the PPAGE register when in special modes. The procedure is as follows:

1. Verify that all ACCERR and PVIOL flags in the FSTAT register are cleared in all banks. This requires to check the FSTAT content for all combinations of the BKSEL bit in the FCNFG register.
2. Write to bit BKSEL in the FCNFG register to select the bank of registers corresponding to the 32K flash block to be programmed or erased (i.e. Flash 0 or 1). See Figure 23, 24 for further details.
3. In special modes, write to the core PPAGE register ($x030) to select one of the 16K pages to be programmed if programming in the $8000–$BFFF address range. There is no need to set PPAGE when programming in the $4000-$7FFF or $C000-$FFFF address ranges or when operating in normal modes.

After this possible initialization step the CBEIF flag should be tested to ensure that the address, data and command buffers are empty. If so, the program/erase command write sequence can be started. The following 3-step command write sequence must be strictly adhered to and no intermediate writes to the Flash module are permitted between the 3 steps. It is possible to read any Flash register during a command sequence. The command sequence is as follows:

1. Write the aligned data word (16-bits) to be programmed to the valid Flash address space between $0000 and $FFFF. The address and data will be stored in internal buffers. For program, all address bits are valid. For erase, the value of the data bytes is don’t care. For mass erase the address can be anywhere in the available address space of the 32K byte block to be erased. For sector erase the address bits 8:0 are ignored for the Flash.
2. Write the program or erase command to the command buffer. These commands are listed in Table 45.
3. Clear the CBEIF flag by writing a “1” to it to launch the command. When the CBEIF flag is cleared, the CCIF flag is cleared by hardware indicating that the command was successfully launched. The ACCERR and PVIOL flags should be tested to

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

Freescale Semiconductor, Inc.

ensure the command sequence was valid. The CBEIF flag will be set again indicating the address, data and command buffers are ready for a new command sequence to begin.

The completion of the command is indicated by the CCIF flag setting (Command Complete Interrupt Flag). The CCIF flag only sets when all active and pending commands have been completed.

NOTE: *The Command State Machine will flag errors in program or erase write sequences by means of the ACCERR (access error) and PVIOL (protection violation) flags in the FSTAT register. An erroneous command write sequence will abort and set the appropriate flag. If set, the user must clear the ACCERR or PVIOL flags before commencing another command write sequence. By writing a 0 to the CBEIF flag the command sequence can be aborted after the word write to the Flash address space or after writing a command to the FCMD register and before the command is launched. Writing a "0" to the CBEIF flag in this way will set the ACCERR flag.*

A summary of the program algorithm is shown in Figure 28. For the erase algorithm, the user writes either a mass or sector erase command to the FCMD register.

Freescale Semiconductor, Inc.

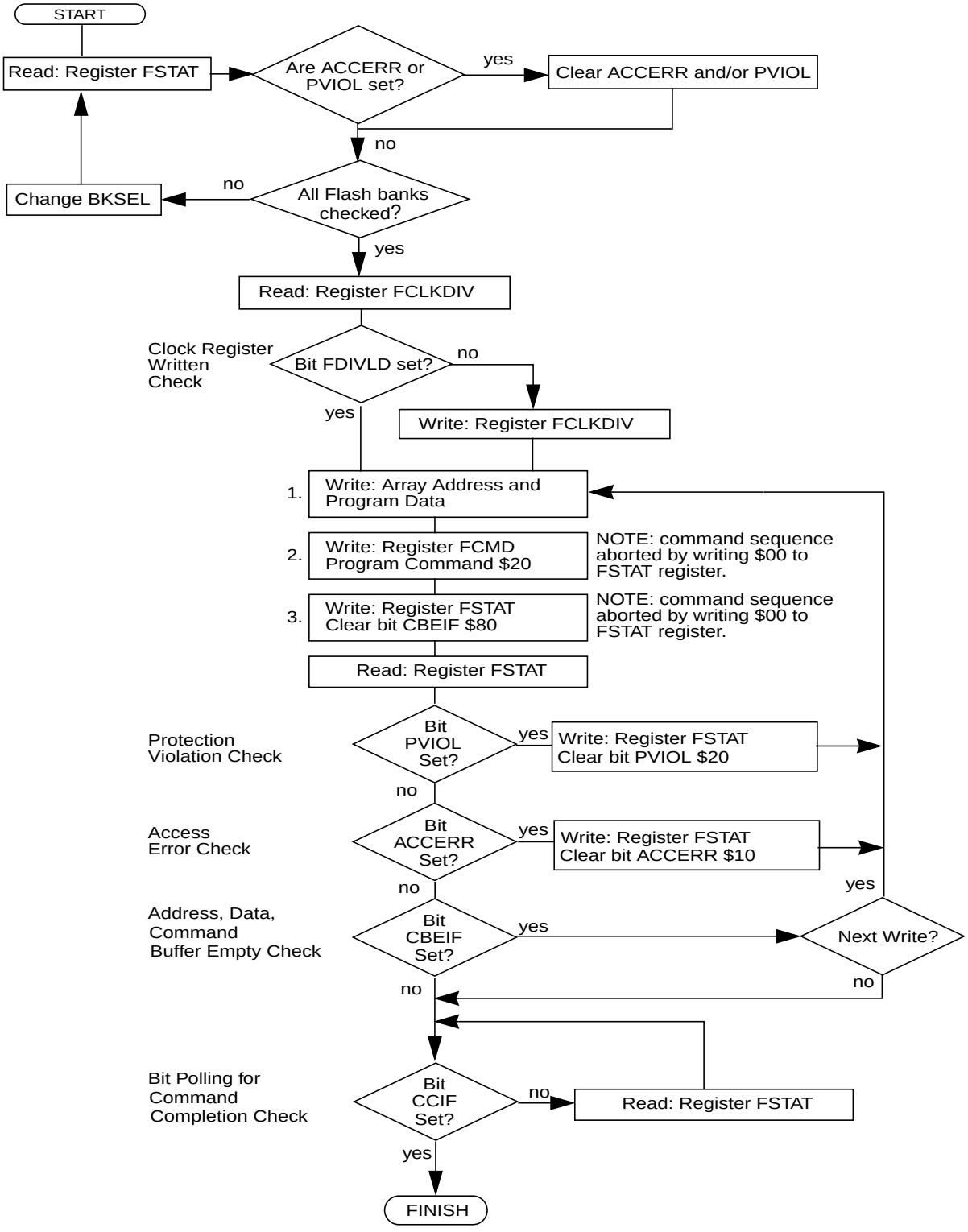


Figure 28 Example Program Algorithm

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

Freescale Semiconductor, Inc.

Program and Erase Sequence in Flash Super User Mode

The Flash EEPROM module requires that the USER program/erase operations be allowed via programming the address (FADDR) and data (FDATA) registers. The operation is controlled via the FSUM bit in the CALCFG register (refer to page 241). Super user operation is indicated when this bit is set.

When FSUM = 1, a register write to the address register will start the monitor state machine for the operation. During this sequence the address presented on the core data bus is monitored for protection violation and operation sequence is monitored for access errors. The access error is set if the strict sequence of, address register access followed by the data register access is not observed. Address alignment is also checked for access error indications. The operation sequence is shown below:

1. Write the address word to be programmed to the valid Flash address space, shown in Table 35, to the FADDR register. The address will be stored in internal buffers.
2. Write the data word to the FDATA register. The data will be stored in internal buffers.

NOTE: *For program, all address bits are valid. For erase, the value of the data bytes is don't care. For mass erase, the address can be anywhere in the available address space of the block to be erased. For sector erase the address bits[8:0] are ignored for the Flash.*

3. Write the program or erase command to the command buffer. These commands are described in *FCMD — Flash Command Register* and listed in Table 45 in page 216.
4. Clear the CBEIF flag by writing a "1" to it to launch the command. When the CBEIF flag is cleared, the CCIF flag is cleared by hardware indicating that the command was successfully launched. The CBEIF flag will be set again indicating the address, data and command buffers are ready for a new command sequence to begin.

A summary of the program algorithm is shown in Figure 29. For the erase algorithm, the user writes either a mass or sector erase command to the FCMD register.

Freescale Semiconductor, Inc.

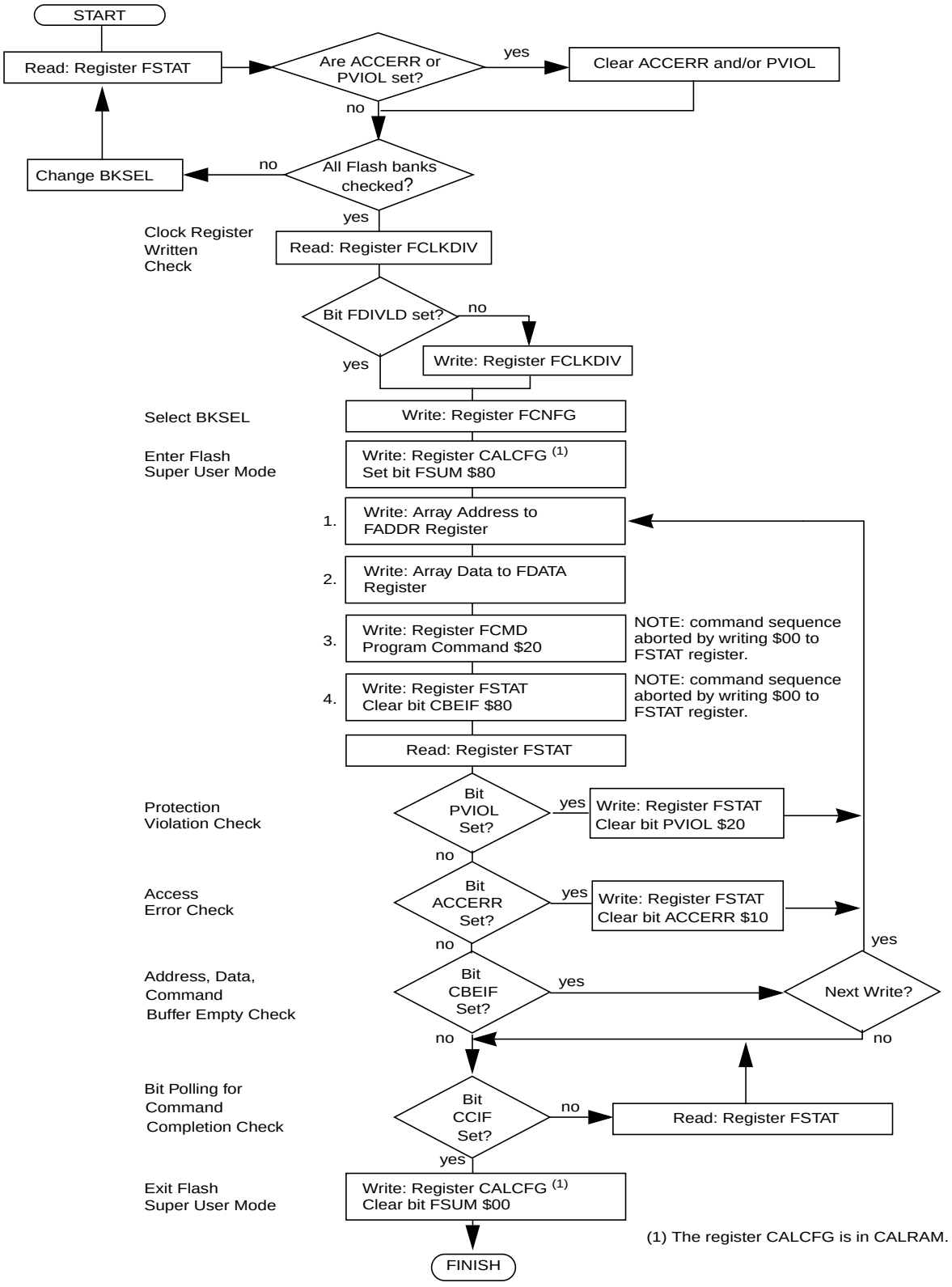


Figure 29 Example Program Algorithm in Flash Super User Mode

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

Freescale Semiconductor, Inc.

Valid Flash Commands

Figure 45 in page 216 summarizes the valid Flash User commands. Also shown are the effects of the commands on the Flash

WARNING: **It is not permitted to program a Flash word without first erasing the sector in which that word resides.**

Illegal Flash Operations

This subsection describes the conditions that set ACCERR and PVIOL flags.

ACCERR flag set condition

The ACCERR flag will be set during the command write sequence if any of the following illegal operations are performed, causing the command write sequence to immediately abort:

1. Writing to the Flash address space before initializing FCLKDIV.
2. Writing to the Flash address space in the range $8000–$BFFF when PPAGE register does not select a 16K bytes page in the Flash block selected by the BKSEL bit in special modes.
3. Writing to the Flash address space $0000–$7FFF or $8000–$FFFF with the BKSEL bit in the FCNFG register not selecting the Flash block in normal modes.
4. Writing a misaligned word or a byte to the valid Flash address space.
5. Writing to the Flash address space while CBEIF is not set.
6. Writing a second word to the Flash address space before executing a program or erase command on the previously written word.
7. Writing to any Flash register other than FCMD after writing a word to the Flash address space.
8. Writing a second command to the FCMD register before executing the previously written command.
9. Writing an invalid user command to the FCMD register in user mode.
10. Writing to any Flash register other than FSTAT (to clear CBEIF) after writing to the command register FCMD.
11. The part enters STOP mode and a program or erase command is in progress. The command is aborted and any pending command is killed.

Freescale Semiconductor, Inc.

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

12. When security is enabled, a command other than Mass-Erase or Erase-Verify originating from a non-secure memory or from the Background Debug Mode is written to FCMD.
13. A “0” is written to the CBEIF bit in the FSTAT register.
14. Violating the register write sequence (FADDR followed by FDATA) in Flash Super User Mode.

The ACCERR flag will not be set if any Flash register is read during the command sequence.

If the Flash array is read during execution of an algorithm (i.e. CCIF bit in the FSTAT register is low) the read will return non valid data and the ACCERR flag will not be set.

If an ACCERR flag is set in any of the banked FSTAT registers, the Command State Machine is locked. It is not possible to launch another command on any block until the ACCERR flag is cleared.

PVIOL flag set condition

The PVIOL flag will be set during the command write sequence after the word write to the Flash address space if any of the following illegal operations are performed, causing the command sequence to immediately abort:

1. Writing a Flash address to program in a protected area of the Flash.
2. Writing a Flash address to erase in a protected area of the Flash.
3. Writing the mass erase command to FCMD while any protection is enabled.

If a PVIOL flag is set in any of the banked FSTAT registers, the Command State Machine is locked. It is not possible to launch another command on any block until the PVIOL flag is cleared.

Low Power Options

When the array or the registers are not being accessed clocking to the register block is shut off to save power. The only exceptions to this are

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

Freescale Semiconductor, Inc.

the flag bits in the FSTAT registers which are updated by internal state machines.

Run Mode

No special current saving modes available.

Wait Mode

When the MCU enters WAIT mode and any command is active (CCIF = 0) the command will be completed. If enabled, interrupts can be used to wake the MCU out of Wait mode.

Stop Mode

No low power options exist for this module in stop mode. If a command is active (CCIF = 0) when the MCU enters the STOP mode, the command will be aborted and the high voltage circuitry to the Flash array will be switched off. CCIF and ACCERR flags will be set. If commands are active in more than one block when STOP occurs, then all the corresponding CCIF and ACCERR flags will be set. Upon exit from STOP the CBEIF flag is set and any pending command will not be executed. The ACCERR flag must be cleared before returning to normal operation.

WARNING: *As active commands are immediately aborted when the MCU enters STOP mode, it is strongly recommended that the user does not use the STOP command during program and erase execution.*

Background Debug Mode

In Background Debug Mode (BDM), the FPROT registers are writable. If the chip is unsecured then all Flash commands listed in Table 45 in 216 can be executed. In special single chip mode if the chip is secured then the only possible command to execute is Mass Erase.

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

Freescale Semiconductor, Inc.

Flash Security

The Flash module provides the necessary security information to the rest of the chip. After each reset, the Flash module determines the security state of the microcontroller as defined in section FSEC — Flash Security Register in page 208.

The contents of the Flash Protection/Options byte at $FF0F in the Flash Protection/Options Field must be changed directly by programming $FF0F when the device is unsecured and the higher address sector is unprotected. If the Flash Protection/Options byte is left in the secure state, any reset will cause the microcontroller to return to the secure operating mode

Reset Initialization

Out of reset the module holds core activity while the Protection and Security registers are loaded from Flash 0. Thereafter, the Flash module is immediately accessible, operating in read mode.

If a reset occurs while any command is in progress that command will be immediately aborted. The state of the word being programmed or the sector / block being erased is not guaranteed.

Freescale Semiconductor, Inc.

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

Interrupts

General

This module can generate an interrupt when all commands are completed or the address, data and command buffers are empty.

Table 46 Flash Interrupt Sources

Interrupt Source	Interrupt Flag	Local Enable	Global (CCR) Mask
Flash Address, Data and Command Buffers empty	CBEIF Flash Block 0 or 1	CBEIE	I Bit
All Commands are completed	CCIF Flash Block 0 or 1	CCIE	I Bit

Description of Interrupt Operation

Figure 30 shows the logic used for generating interrupt via the relevant block.

This system uses the CBEIF and CCIF flags in combination with the enable bits CBIE and CCIE in addition to the BKSEL bit, to discriminate for the interrupt generation. By taking account of the possible selected bank, the system is prevented from generating false interrupts when the command buffer is empty in an unselected bank.

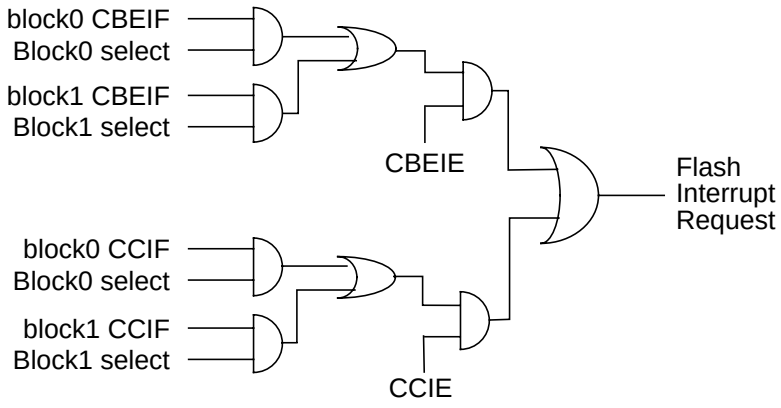


Figure 30 Flash Interrupt Implementation.

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

Freescale Semiconductor, Inc.

CAUTION: *When programming or erasing Flash Block 0 the interrupt vectors are not readable. Therefore, it is not recommended to program or erase the Flash Block 0 with interrupts enabled.*

Recovery from STOP or WAIT

The module can recover the part from WAIT, if the interrupts are enabled.

There is no capability to recover from STOP.

Freescale Semiconductor, Inc.

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

Freescale Semiconductor, Inc.

CALRAM 2K

Contents

Glossary 235
Overview 236
Features 236
Modes of Operations 237
Block Diagram 238
External Pin Descriptions 239
Module Memory Map 239
Register Descriptions 241
Functional Description 242
Reset Initialization 248

Glossary

CALRAM Module Includes Bus Interface, Calibration Control and a 2K byte RAM block.

Flash Super User Mode The flash super user mode allows the user to use erasing/programming sequences with FADDR/FDATA registers.

Freescale Semiconductor, Inc.

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

CALRAM 2K

Overview

The 2K byte CALRAM (Calibration RAM) module can overlap with the Flash EEPROM array, and CPU can access data in the CALRAM instead of in the Flash EEPROM array. During overlapping, the Flash EEPROM array can be erased or programmed by entering the flash super user mode.

The CALRAM may be read or written as either bytes, aligned words or misaligned words. For compatibility with Flash EEPROM, access time is one bus cycle for byte and aligned word accesses, and two bus cycles for misaligned word accesses.

When the MCU is in the secured state and in the special single chip mode, reading the CALRAM array returns $FFFF for any word accesses or $FF for any byte accesses instead of returning actual values stored in the array.

Features

- 2k bytes of CALRAM
  - 2K byte calibration block over Flash EEPROM
  - Access cycle compatible to Flash EEPROM
  - Flash EEPROM erase or program possible while the CALRAM overlaps with the Flash EEPROM
  - Re-mappable to any 2K byte boundary

Freescale Semiconductor, Inc.

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

Modes of Operations

Secured Mode The flash module provides the necessary security information to the rest of the chip. This information is stored within a byte in the flash block 0. This byte is read automatically after each reset and stored in a volatile register in the flash module. This information also protects the CALRAM module from intrusive reads via the external bus interface or the background debug mode.

Run Mode Normal mode of operation.

Wait Mode The CALRAM modules operates normally during WAIT mode if the bus clock is enable.

Stop Mode All clocks are stopped.

Freescale Semiconductor, Inc.

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

Block Diagram

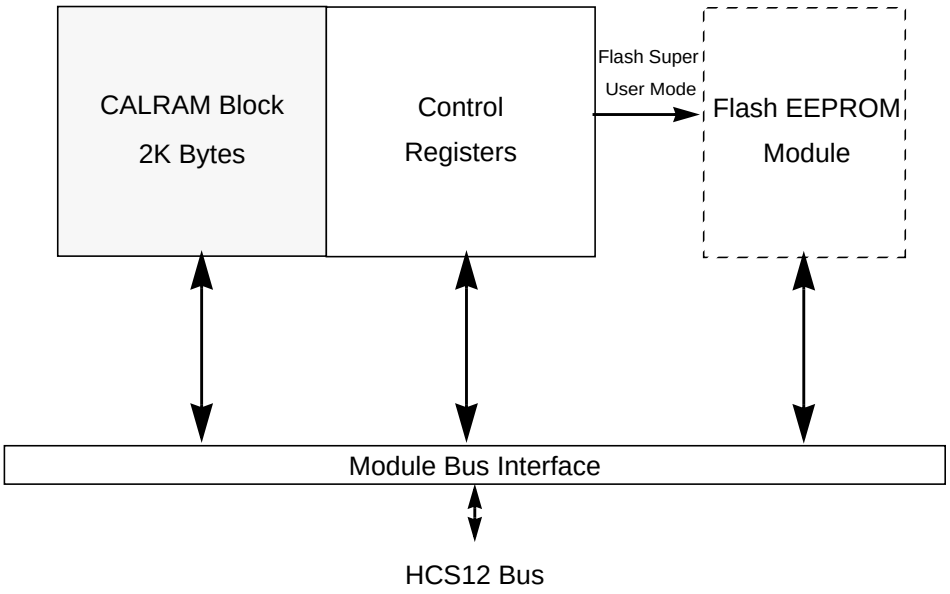


Figure 31 CALRAM 2K Byte Block Diagram

Freescale Semiconductor, Inc.

External Pin Descriptions

This module does not have external pins relevant for the user.

Module Memory Map

Overview

The memory data is accessible in the address range $1000 - $17FF after reset and can be re-mapped to any 2k byte boundary within the MCU address space. The CALRAM module contains a control register in the same address space INITRG +$0FC - INITRG +$0FF.

Data Memory Map

CALRAM array can be mapped to any 2K byte boundary within the 64K byte address space. The CALRAM array address can be changed anytime through the INITCRM register.

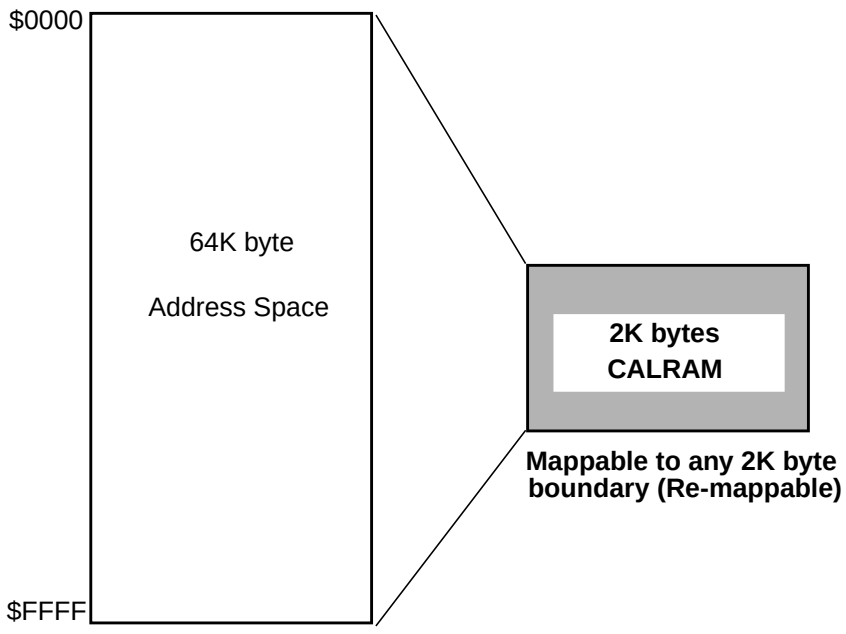


Figure 32 CALRAM Data Memory Map

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

Freescale Semiconductor, Inc.

Register Memory Map

Register name		Bit 7	6	5	4	3	2	1	Bit 0	Addr. Offset
CALCFG	Read:	FSUM	0	0	0	0	0	0	0	$00FC
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$00FD-$00FF
	Write:									

= Unimplemented or reserved

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Freescale Semiconductor, Inc.

Register Descriptions

CALCFG — Calibration Configuration Register

Address Offset: $00FC

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	FSUM	0	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime
Write: Anytime

The FSUM bit is used to establish the operating mode for the flash module. Setting this bit will put the flash module in the flash super user mode.

FSUM — Flash Super User Mode bit

When this bit is cleared, the Flash EEPROM module enters in normal mode. When this bit is set, the Flash EEPROM module enters in flash super user mode and FADDR (FADDRHI and FADDRLO) and FDATA (FDATAHI and FDATALO) registers can be used to program flash array area which is overlapped by the CALRAM. See Flash EEPROM section in page 193 for more information.

1 = Flash module is in Flash Super user mode
0 = Flash module is in Normal mode

Freescale Semiconductor, Inc.

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

Functional Description

The CALRAM is designed to replace data in the Flash EEPROM for calibration purposes without stopping the user's control program.

This function is controlled by the register INITCRM (see Module Mapping Control (MMC) in page 121) and the CALCFG register.

Calibration Block Specified by INITCRM

CALRAM can be mapped to any 2K byte boundary within the 64K byte memory space.

The CRAM[15:11] bits in the register INITCRM are used to map the CALRAM to a 2K byte boundary. These write-anytime bits have to be written in order to choose the calibration block area in the Flash EEPROM. Since data in the CALRAM array is undefined after reset, data may be written in the array before mapping to the calibration area to prevent reading unpredictable values. Table 47 shows example CALRAM mapping.

Table 47 Example CALRAM Mapping

CRAM[15:11]	Address
00000	$0000 – $07FF
00001	$0800 – $0FFF
00010	$1000 – $17FF
.....	.....
11101	$E800 – $EFFF
11110	$F000 – $F7FF
11111	$F800 – $FFFF

There is another bit called CRAMON in the register INITCRM. This write-anytime bit makes the CALRAM array visible in the address specified by the CRAM bits. If this bit is set the CALRAM is used, otherwise the Flash memory is used.

Flash Super User Mode Control

The FSUM bit in the register CALCFG determines if the Flash EEPROM module enters the flash super user mode. When FSUM is set, the Flash

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

Freescale Semiconductor, Inc.

EEPROM module is in the flash super user mode and the FADDR (FADDRHI and FADDRLO) and FDATA (FDATAHI and FDATALO) become visible in the memory map. These registers are used to erase or to program the Flash EEPROM array which is overlapped by the CALRAM. If the user needs to erase or to program a Flash EEPROM block while the CPU is running, it is necessary to make sure that CPU is running from another Flash EEPROM block or from another memory (e.g. RAM). During erasing or programming a Flash EEPROM block, the block cannot be read due to Flash EEPROM limitations.

Freescale Semiconductor, Inc.

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

Recommended Calibration Flow

Following flow is recommended to calibrate the 2K byte Flash EEPROM array area. It is assumed that this flow is executed after reset without initialization of the registers INITCRM and CALCFG.

1. Copy a 2K byte block of data from the Flash EEPROM into the 2K byte CALRAM array. Figure 33 shows an example. This is the first step for calibration, but this step may be skipped when initialization for calibration block is not needed.

The data to be calibrated in Flash EEPROM has to be located within the 2K byte area which is overlapped by the CALRAM.

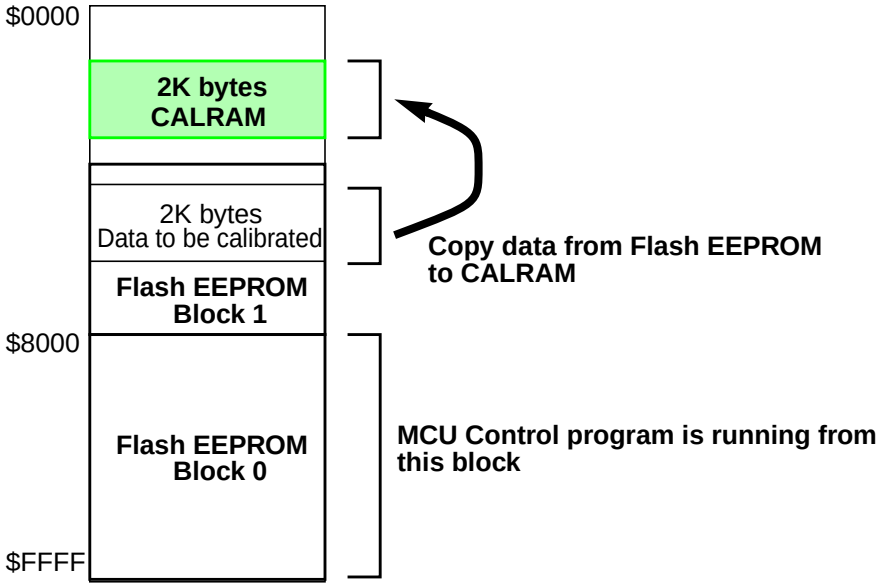


Figure 33 Copy Data from Flash EEPROM to CALRAM

Freescale Semiconductor, Inc.

2. Initialize the CRAM[15:11] bits and set the CRAMON bit in the register INITCRM to overlap the 2K byte CALRAM array with the Flash EEPROM. Figure 34 shows an example of how to map the CALRAM over the Flash EEPROM.

NOTE: *Writes to the INITCRM register take one cycle to go into effect.*

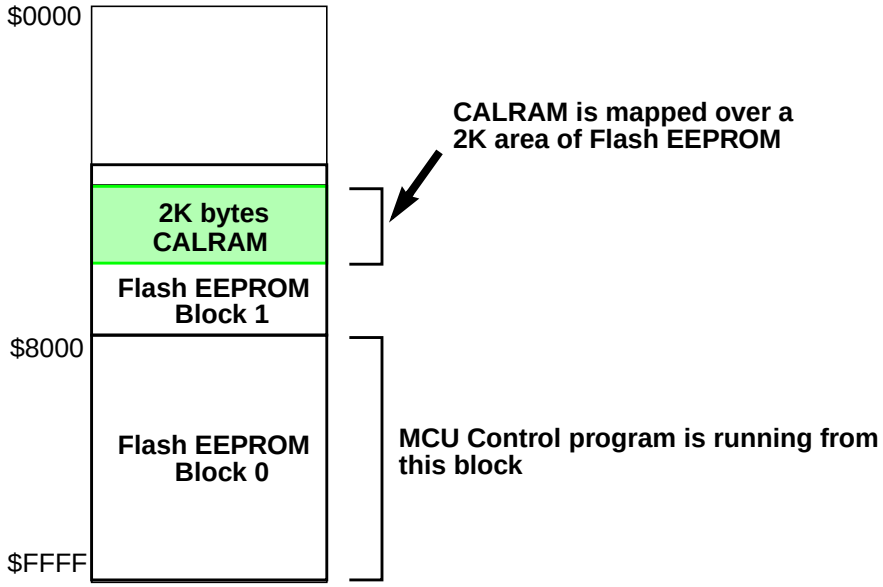


Figure 34 Map CALRAM over Flash EEPROM

3. Calibrate data in the CALRAM

The user can start to change data in the CALRAM array for calibration purposes.

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

Freescale Semiconductor, Inc.

4. Set FSUM in the register CALCFG.

After calibration is done, the FSUM bit is written to erase and to program the Flash EEPROM. Setting the FSUM bit enables the flash super user mode and the FADDR (FADDRHI and FADDRLO) and FDATA (FDATAHI and FDATALO) registers become visible in the memory map.

Erase/program sequences in the flash super user mode must be used, since writing calibration data area only causes the CALRAM array to be written. See the erase/program sequence in the flash super user mode in page 229.

Note that during erasing/programming a Flash EEPROM block, the data contained into the correspondent block cannot be read regardless if it is been overlapped by the CALRAM or not.

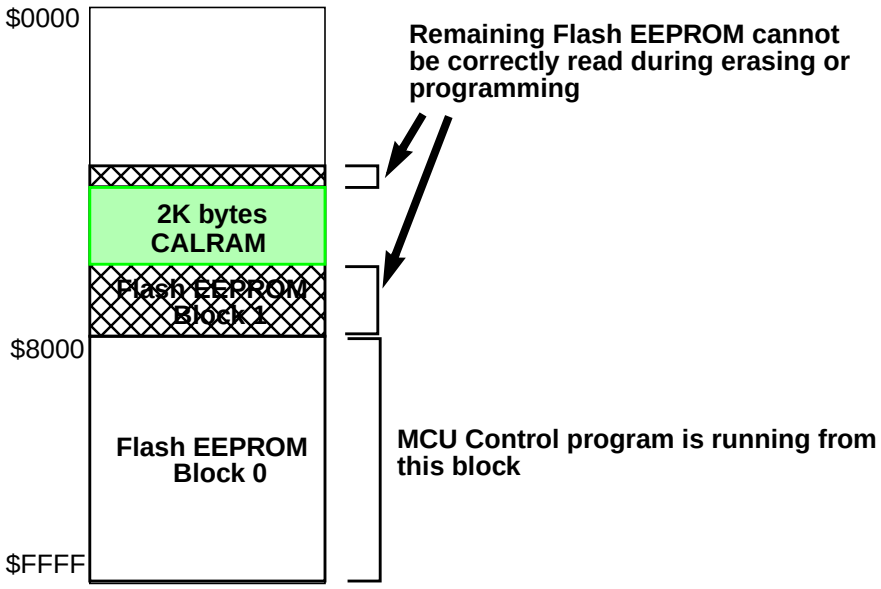


Figure 35 An Example Memory Map When Erase/Program Flash EEPROM

5. Erase data in the Flash EEPROM.

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

Freescale Semiconductor, Inc.

6. Program data contained in the CALRAM with the Flash EEPROM.

7. Clear FSUM in the register CALCFG and clear CRAMON in the register INITCRM.

After the FSUM bit is cleared, the Flash EEPROM module exits the flash super user mode and the registers FADDR and FDATA become invisible in the memory map.

When the CRAMON bit is cleared, the CALRAM array disappears from the 64K byte address space and the calibrated data can be read from Flash EEPROM directly. Figure 36 shows the memory map obtained after calibration.

NOTE: *Writes to the INITCRM register take one cycle to go into effect.*

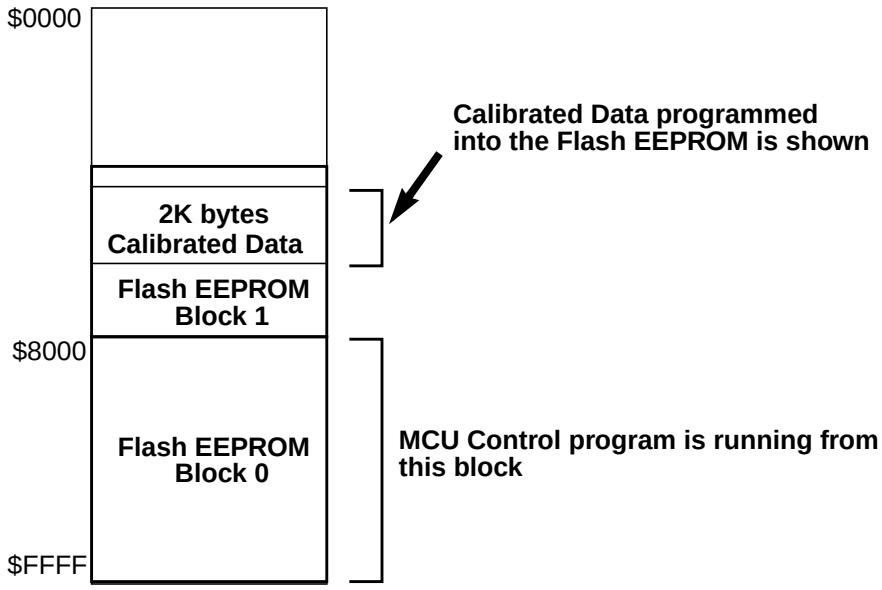


Figure 36 Remove CALRAM from memory mapping

8. If needed, continue calibration by repeating from the step 2.

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

Freescale Semiconductor, Inc.

NOTE: *During Flash program or erase, data cannot be read from a Flash EEPROM block that is under a erase or program operation. However, it is possible to read from a different block. For example: During erasing or programming the flash block 1, the flash block 1 should not be read while the flash block 0 could still be used for MCU control program storage.*

Reset Initialization

All registers get set/reset asynchronously.

Freescale Semiconductor, Inc.

Port Integration Module (PIM)

Contents

Overview 249
Block Diagram 251
External Pin Descriptions 252
Register Map 253
Register Descriptions 255
Functional Description 267
Low Power Options 269
Reset Initialization 269

Overview

The Port Integration Module establishes the interface between the peripheral modules and the I/O pins for all ports except A, B, E and K. Ports A, B, E and K are handled by the HC12 multiplexed bus interface and described in Multiplexed External Bus Interface (MEBI) section on page 141, due to their tight link with the external bus interface and special modes.

The 8-bit port associated with the ATD is included in the ATD module due to their sensitivity to electrical noise, requiring special care on routing and design.

This section covers port T connected to the timer module, the serial port S associated with 2 SCI and 1 SPI module and port P connected to the PWM.

Each I/O pin is associated with a set of registers which configure items like input/output selection, drive strength reduction, and pull resistors enable and selection.

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

Freescale Semiconductor, Inc.

Port Integration Module (PIM)

The port integration module is device dependant which is reflected in its naming.

A standard port has the following minimum features:

- Input/output selection
- 5V output drive with two selectable drive strength
- 5V digital and analog input
- Input with selectable pull-up or pull-down device

Optional features:

- Open drain for wired-or connections

Freescale Semiconductor, Inc.

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

Block Diagram

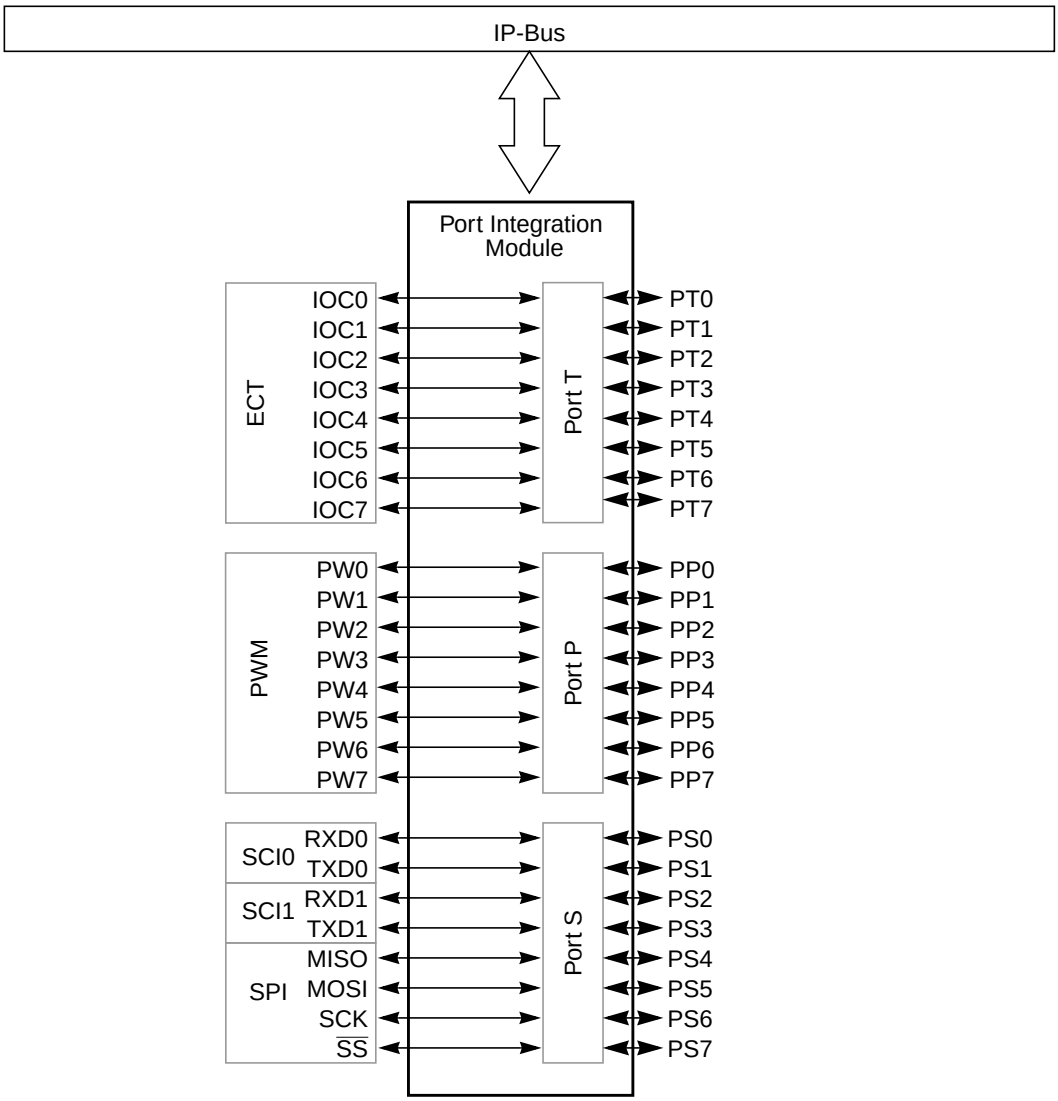


Figure 37 PIM Block Diagram

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

Freescale Semiconductor, Inc.

External Pin Descriptions

All ports start up as general purpose inputs on reset.

Table 48 Port Reset State and Priority Summary

Port	Reset States					Priority
	Data Direction	Pull Mode	Red. Drive	Wired-Or Mode	Inter-rupt	
T	input	hiz	disabled	n/a	n/a	ECT > GPIO
S	input	pull-up	disabled	disabled	n/a	SCI, SPI > GPIO
P	input	hiz	disabled	n/a	n/a	PWM > GPIO

Freescale Semiconductor, Inc.

Register Map

Register name		Bit 7	6	5	4	3	2	1	Bit 0	Addr. Offset
PTT	Read:	PTT7	PTT6	PTT5	PTT4	PTT3	PTT2	PTT1	PTT0	$00E0
	Write:									
PTIT	Read:	PTIT7	PTIT6	PTIT5	PTIT4	PTIT3	PTIT2	PTIT1	PTIT0	$00E1
	Write:									
DDRT	Read:	DDRT7	DDRT6	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0	$00E2
	Write:									
RDRT	Read:	RDRT7	RDRT6	RDRT5	RDRT4	RDRT3	RDRT2	RDRT1	RDRT0	$00E3
	Write:									
PERT	Read:	PERT7	PERT6	PERT5	PERT4	PERT3	PERT2	PERT1	PERT0	$00E4
	Write:									
PPST	Read:	PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0	$00E5
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$00E6
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$00E7
	Write:									
PTS	Read:	PTS7	PTS6	PTS5	PTS4	PTS3	PTS2	PTS1	PTS0	$00E8
	Write:									
PTIS	Read:	PTIS7	PTIS6	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0	$00E9
	Write:									
DDRS	Read:	DDRS7	DDRS6	DDRS5	DDRS4	DDRS3	DDRS2	DDRS1	DDRS0	$00EA
	Write:									
RDRS	Read:	RDRS7	RDRS6	RDRS5	RDRS4	RDRS3	RDRS2	RDRS1	RDRS0	$00EB
	Write:									
PERS	Read:	PERS7	PERS6	PERS5	PERS4	PERS3	PERS2	PERS1	PERS0	$00EC
	Write:									

= Unimplemented or reserved

Figure 38 PIM9T64 Register Map

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

Freescale Semiconductor, Inc.

Register name		Bit 7	6	5	4	3	2	1	Bit 0	Addr. Offset
PPSS	Read: Write:	PPSS7	PPSS6	PPSS5	PPSS4	PPSS3	PPSS2	PPSS1	PPSS0	$00ED
WOMS	Read: Write:	WOMS7	WOMS6	WOMS5	WOMS4	WOMS3	WOMS2	WOMS1	WOMS0	$00EE
Reserved	Read:	0	0	0	0	0	0	0	0	$00EF
	Write:									
PTP	Read: Write:	PTP7	PTP6	PTP5	PTP4	PTP3	PTP2	PTP1	PTP0	$00F0
PTIP	Read:	PTIP7	PTIP6	PTIP5	PTIP4	PTIP3	PTIP2	PTIP1	PTIP0	$00F1
	Write:									
DDRP	Read: Write:	DDRP7	DDRP6	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0	$00F2
RDRP	Read: Write:	RDRP7	RDRP6	RDRP5	RDRP4	RDRP3	RDRP2	RDRP1	RDRP0	$00F3
PERP	Read: Write:	PERP7	PERP6	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0	$00F4
PPSP	Read: Write:	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0	$00F5
Reserved	Read:	0	0	0	0	0	0	0	0	$00F6
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$00F7
	Write:									

= Unimplemented or reserved

Figure 38 PIM9T64 Register Map (Continued)

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Freescale Semiconductor, Inc.

Register Descriptions

The following table summarizes the effect on the various configuration bits, data direction (DDR), output level (I/O), reduced drive (RDR), pull enable (PE), and polarity select (PS) for the ports. The configuration bit PS is used for the purposing of selecting either a pull-up or pull-down device if PE is active.

Table 49 Pin Configuration Summary

DDR	I/O	RDR	PE	PS	Function	Pull Device
0	X	X	0	X	Input	Disabled
0	X	X	1	0	Input	Pull Up
0	X	X	1	1	Input	Pull Down
1	0	0	X	X	Output, full drive to 0	Disabled
1	1	0	X	X	Output, full drive to 1	Disabled
1	0	1	X	X	Output, reduced drive to 0	Disabled
1	1	1	X	X	Output, reduced drive to 1	Disabled

NOTE: *All bits of all registers in this module are completely synchronous to internal clocks during a register read.*

Freescale Semiconductor, Inc.

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

Port T I/O Register (PTT)

Address Offset: $00E0

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTT7	PTT6	PTT5	PTT4	PTT3	PTT2	PTT1	PTT0
ECT:	I/OC7	I/OC6	I/OC5	I/OC4	I/OC3	I/OC2	I/OC1	I/OC0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

If the data direction bits of the associated I/O pins are set to 1, a read returns the value of the port register, otherwise the value at the pins is read.

Port T Input Register (PTIT)

Address Offset: $00E1

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTIT7	PTIT6	PTIT5	PTIT4	PTIT3	PTIT2	PTIT1	PTIT0
Write:								
Reset:	-	-	-	-	-	-	-	-

= Reserved or unimplemented

Read: Anytime.
Write: Never; writes to this register have no effect.

This register always reads back the status of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

Freescale Semiconductor, Inc.

Port T Data Direction Register (DDRT)

Address Offset: $00E2

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRT7	DDRT6	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

This register configures each port T pin as either input or output. The ECT forces the I/O state to be an output for each timer port associated with an enabled output compare. In these cases the data direction bits will not change.
The DDRT bits revert to controlling the I/O direction of a pin when the associated timer output compare is disabled.
The timer input capture always monitors the state of the pin.

DDRT[7:0] — Data Direction Port T

1 = Associated pin is configured as output.
0 = Associated pin is configured as input.

Due to internal synchronization circuits, it can take up to 2 bus cycles until the correct value is read on PTT or PTIT registers, when changing the DDRT register.

Freescale Semiconductor, Inc.

Port T Reduced Drive Register (RDRT)

Address Offset: $00E3

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	RDRT7	RDRT6	RDRT5	RDRT4	RDRT3	RDRT2	RDRT1	RDRT0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

This register configures the drive strength of each port T output pin as either full or reduced. If the port is used as input this bit is ignored.

RDRT[7:0] — Reduced Drive Port T

1 = Associated pin drives at about 1/3 of the full drive strength.
0 = Full drive strength at output.

Port T Pull Device Enable Register (PERT)

Address Offset: $00E4

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PERT7	PERT6	PERT5	PERT4	PERT3	PERT2	PERT1	PERT0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

This register configures whether a pull-up or a pull-down device is activated, if the port is used as input. This bit has no effect if the port is used as output. Out of reset no pull device is enabled.

PERT[7:0] — Pull Device Enable Port T

1 = Either a pull-up or pull-down device is enabled.
0 = Pull-up or pull-down device is disabled.

Freescale Semiconductor, Inc.

Port T Polarity Select Register (PPST)

Address Offset: $00E5

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

This register selects whether a pull-down or a pull-up device is connected to the pin.

PPST[7:0] — Pull Select Port T

1 = A pull-down device is connected to the associated port T pin, if enabled by the associated bit in register PERT and if the port is used as input.

0 = A pull-up device is connected to the associated port T pin, if enabled by the associated bit in register PERT and if the port is used as input.

Port S I/O Register (PTS)

Address Offset: $00E8

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTS7	PTS6	PTS5	PTS4	PTS3	PTS2	PTS1	PTS0
SPI/SCI:	$\overline{SS}$	SCK	MOSI	MISO	TxD1	RxD1	TxD0	RxD0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

If the data direction bits of the associated I/O pins are set to 1, a read returns the value of the port register, otherwise the value at the pins is read.

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

Freescale Semiconductor, Inc.

The SPI pins (PS[7:4]) configuration is determined by several status bits in the SPI module. See Serial Peripheral Interface (SPI) section on page 457 for details.

The SCI pins associated with transmit pins 3 and 1 are configured as outputs if the transmitter is enabled.

The SCI pins associated with receive pins 2 and 0 are configured as inputs if the receiver is enabled. See Serial Communications Interface (SCI) section on page 419 for details.

Port S Input Register (PTIS)

Address Offset: $00E9

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTIS7	PTIS6	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0
Write:								
Reset:	-	-	-	-	-	-	-	-

= Reserved or unimplemented

Read: Anytime.

Write: Never; writes to this register have no effect.

This register always reads back the status of the associated pins. This also can be used to detect overload or short circuit conditions on output pins.

Port S Data Direction Register (DDRS)

Address Offset: $00EA

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	DDRS7	DDRS6	DDRS5	DDRS4	DDRS3	DDRS2	DDRS1	DDRS0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.

Write: Anytime.

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

Freescale Semiconductor, Inc.

This register configures each port S pin as either input or output.
If SPI is enabled, the SPI determines the pin direction. For details see Serial Peripheral Interface (SPI) section on page 457.
If the associated SCI transmit or receive channel is enabled this register has no effect on the pins. The pin is forced to be an output if a SCI transmit channel is enabled, it is forced to be an input if the SCI receive channel is enabled.
The DDRS bits revert to controlling the I/O direction of a pin when the associated channel is disabled.

DDRS[7:0] — Data Direction Port S
1 = Associated pin is configured as output.
0 = Associated pin is configured as input.

Due to internal synchronization circuits, it can take up to 2 bus cycles until the correct value is read on PTS or PTIS registers, when changing the DDRS register.

Port S Reduced Drive Register (RDRS)

Address Offset: $00EB

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	RDRS7	RDRS6	RDRS5	RDRS4	RDRS3	RDRS2	RDRS1	RDRS0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

This register configures the drive strength of each port S output pin as either full or reduced. If the port is used as input this bit is ignored.

RDRS[7:0] — Reduced Drive Port S
1 = Associated pin drives at about 1/3 of the full drive strength.
0 = Full drive strength at output.

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

Port S Pull Device Enable Register (PERS)

Address Offset: $00EC

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PERS7	PERS6	PERS5	PERS4	PERS3	PERS2	PERS1	PERS0
Reset:	1	1	1	1	1	1	1	1

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

This register configures whether a pull-up or a pull-down device is activated, if the port is used as input or as output in wired-or (open drain) mode. This bit has no effect if the port is used as push-pull output. Out of reset a pull-up device is enabled.

PERS[7:0] — Pull Device Enable Port S

1 = Either a pull-up or pull-down device is enabled.
0 = Pull-up or pull-down device is disabled.

Port S Polarity Select Register (PPSS)

Address Offset: $00ED

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PPSS7	PPSS6	PPSS5	PPSS4	PPSS3	PPSS2	PPSS1	PPSS0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

This register selects whether a pull-down or a pull-up device is connected to the pin.

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

Freescale Semiconductor, Inc.

PPSS[7:0] — Pull Select Port S

1 = A pull-down device is connected to the associated port S pin, if enabled by the associated bit in register PERS and if the port is used as input.

0 = A pull-up device is connected to the associated port S pin, if enabled by the associated bit in register PERS and if the port is used as input or as wired-or output.

Port S Wired-Or Mode Register (WOMS)

Address Offset: $00EE

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	WOMS7	WOMS6	WOMS5	WOMS4	WOMS3	WOMS2	WOMS1	WOMS0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

This register configures the output pins as wired-or. If enabled the output is driven active low only (open-drain). A logic level of ‘1’ is not driven. It applies also to the SPI and SCI outputs and allows a multipoint connection of several serial modules. This bit has no influence on pins used as inputs.

WOMS[7:0] — Wired-Or Mode Port S

1 = Output buffers operate as open-drain outputs.

0 = Output buffers operate as push-pull outputs.

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

Freescale Semiconductor, Inc.

Port P I/O Register (PTP)

Address Offset: $00F0

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTP7	PTP6	PTP5	PTP4	PTP3	PTP2	PTP1	PTP0
PWM:	PWM7	PWM6	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

If the data direction bits of the associated I/O pins are set to 1, a read returns the value of the port register, otherwise the value at the pins is read.
The PWM function takes precedence over the general purpose I/O function if the associated PWM channel is enabled. While channels 6–0 are output only if the respective channel is enabled, channel 7 can be PWM output or input if the shutdown feature is enabled. See Pulse Width Modulator (PWM8B8C) section on page 249 for details.

Port P Input Register (PTIP)

Address Offset: $00F1

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTIP7	PTIP6	PTIP5	PTIP4	PTIP3	PTIP2	PTIP1	PTIP0
Write:								
Reset:	-	-	-	-	-	-	-	-

= Reserved or unimplemented

Read: Anytime.
Write: Never; writes to this register have no effect.

This register always reads back the status of the associated pins. This can be also used to detect overload or short circuit conditions on output pins.

Freescale Semiconductor, Inc.

Port P Data Direction Register (DDRP)

Address Offset: $00F2

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRP7	DDRP6	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

This register configures each port P pin as either input or output.
If the associated PWM channel is enabled this register has no effect on the pins.
The PWM forces the I/O state to be an output for each port line associated with an enabled PWM7–0 channel. Channel 7 can force the pin to input if the shutdown feature is enabled.
The DDRP bits revert to controlling the I/O direction of a pin when the associated PWM channel is disabled.

DDRP[7:0] — Data Direction Port P

1 = Associated pin is configured as output.
0 = Associated pin is configured as input.

Due to internal synchronization circuits, it can take up to 2 bus cycles until the correct value is read on PTP or PTIP registers, when changing the DDRP register.

Freescale Semiconductor, Inc.

Port P Reduced Drive Register (RDRP)

Address Offset: $00F3

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	RDRP7	RDRP6	RDRP5	RDRP4	RDRP3	RDRP2	RDRP1	RDRP0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

This register configures the drive strength of each port P output pin as either full or reduced. If the port is used as input this bit is ignored.

RDRP[7:0] — Reduced Drive Port P

1 = Associated pin drives at about 1/3 of the full drive strength.
0 = Full drive strength at output.

Port P Pull Device Enable Register (PERP)

Address Offset: $00F4

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PERP7	PERP6	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

This register configures whether a pull-up or a pull-down device is activated, if the port is used as input. This bit has no effect if the port is used as output. Out of reset no pull device is enabled.

PERP[7:0] — Pull Device Enable Port P

1 = Either a pull-up or pull-down device is enabled.
0 = Pull-up or pull-down device is disabled.

Freescale Semiconductor, Inc.

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

Port P Polarity Select Register (PPSP)

Address Offset: $00F5

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: Anytime.
Write: Anytime.

This register selects a pull-up or a pull-down device if enabled.

PPSP[7:0] — Polarity Select Port P

1 = A pull-down device is connected to the associated port P pin, if enabled by the associated bit in register PERP and if the port is used as input.

0 = A pull-up device is connected to the associated port P pin, if enabled by the associated bit in register PERP and if the port is used as input.

Functional Description

General

Each pin can act as general purpose I/O. In addition the pin can act as an output or as an input of a peripheral module. A set of configuration registers is common to all ports. All registers can be written at any time, however certain configurations might not become active.

Example:

Selecting a pull-up resistor. This resistor does not become active while the port is used as a push-pull output.

I/O register

This register holds the value driven out to the pin if the port is used as a general purpose I/O. Writing to this register has only an effect on the pin

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

Freescale Semiconductor, Inc.

if the port is used as general purpose output. When reading this address, the value of the pins is returned if the data direction register bits are set to 0. If the data direction register bits are set to 1, the contents of the I/O register is returned. This is independent of any other configuration.

Input register

This is a read-only register and always returns the value of the pin.

Data direction register

This register defines whether the pin is used as an input or an output. If a peripheral module controls the pin the contents of the data direction register is ignored.

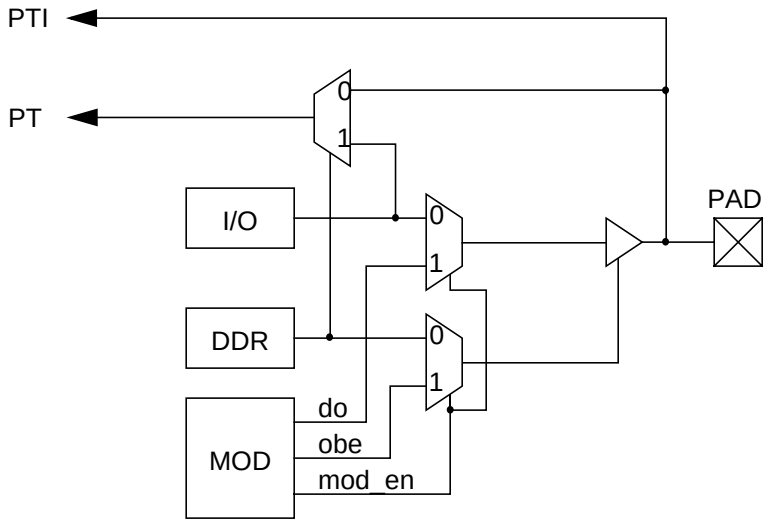


Figure 39 Illustration of I/O pin functionality

Reduced drive register

If the port is used as an output the register allows the configuration of the drive strength.

Pull device enable register

This register turns on a pull-up or pull-down device. It becomes only active if the pin is used as an input or as a wired-or output.

Polarity select register

This register selects either a pull-up or pull-down device if enabled. It becomes only active if the pin is used as an input. A pull-up device can be activated if the pin is used as a wired-or output.

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

Freescale Semiconductor, Inc.

Port T This port is associated with the Enhanced Capture Timer module. In all modes, port T pins PT[7:0] can be used for either general-purpose I/O, or with the channels of the Enhanced Capture Timer. During reset, port T pins are configured as high-impedance inputs.

Port S This port is associated with the serial SCI and SPI modules. In all modes, port S pins PS[7:0] can be used either for general-purpose I/O, or with the SCI and SPI subsystems. During reset, port S pins are configured as inputs with pull-up.

Port P This port is associated with the PWM module. In all modes, port P pins PP[7:0] can be used for either general purpose I/O or with the PWM subsystem. If the PWM is enabled the pins become PWM output channels with the exception of pin 7 which can be PWM input or output. During reset, port P pins are configured as high-impedance inputs.

Low Power Options

Run Mode No low power options exist for this module in run mode.

Wait Mode No low power options exist for this module in wait mode.

Stop Mode All clocks are stopped.

Reset Initialization

All registers including the data registers get set/reset asynchronously.

Freescale Semiconductor, Inc.

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

Freescale Semiconductor, Inc.

Clocks and Reset Generator (CRG)

Contents

Overview 271
Features 272
Modes of Operation 273
Block Diagram 275
External Pin Descriptions 276
Register Map 279
Register Descriptions 280
Functional Description 294
Operation Modes 305
Low Power Options 305
Reset Description 318
Interrupts 324

Overview

This specification describes the function of the Clocks and Reset Generator (CRG) and Oscillator (OSC) modules.

Freescale Semiconductor, Inc.

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

Features

The main features of this block are:

- Crystal (or ceramic resonator) oscillator (OSC)
  - User selectable Oscillator type: Colpitts (low power) or Pierce
  - Clock Monitor (CM)
  - Startup counter
- Phase Locked Loop (PLL) frequency multiplier
  - Reference divider
  - Automatic bandwidth control mode for low-jitter operation
  - Automatic frequency lock detector
  - CPU interrupt on entry or exit from locked condition
  - Self clock mode in absence of reference clock
- System Clock Generator (CGEN)
  - External clock mode
  - Clock switch for either Oscillator or PLL based system clocks
  - User selectable disabling of clocks during Wait Mode for reduced power consumption.
- Computer Operating Properly (COP) Watchdog Timer with time-out clear window.
- System Reset generation from the following possible sources
  - Power on reset
  - COP reset
  - Loss of clock reset
  - External pin reset
- Real-Time Interrupt (RTI)

Freescale Semiconductor, Inc.

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

Modes of Operation

This subsection lists and briefly describes all CRG operating modes supported by the CRG. This is a high level description only, detailed descriptions of operating modes are contained in later sections.

Run Mode

All functional parts of the CRG are running during Run Mode. If RTI or COP functionality is required the individual bits of the associated rate select registers (COPCTL, RTICTL) have to be set to a non zero value[1].

Wait Mode

Depending on the configuration of the individual bits in the CLKSEL register this mode allows to disable the system and core clocks.

Stop Mode

Depending on the setting of the PSTP bit Stop Mode can be differentiated between Full Stop Mode (PSTP=0) and Pseudo Stop Mode (PSTP=1).

- Full Stop Mode

  The oscillator is disabled and thus all system and core clocks are stopped. The COP and the RTI remain frozen.

- Pseudo Stop Mode

  The oscillator continues to run and most of the system and core clocks are stopped. If the respective enable bits are set the COP and RTI will continue to run, else they remain frozen.

Self Clock Mode

Self Clock Mode will be entered if the Clock Monitor Enable Bit (CME) and the Self Clock Mode Enable Bit (SCME) are both asserted and the clock monitor detects a loss of clock (external oscillator or crystal). As soon as Self Clock Mode is entered the CRG starts to perform a clock check. Self Clock Mode remains active until the clock check indicates the required quality of the incoming clock signal is met (frequency and amplitude). Self Clock Mode should be used for safety purposes only. It

1. COPCTL register is write once only

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

Freescale Semiconductor, Inc.

provides reduced functionality to the MCU in case a loss of clock is causing severe system conditions.

Freescale Semiconductor, Inc.

Block Diagram

The block diagram below shows a high level view of the CRG and OSC modules.

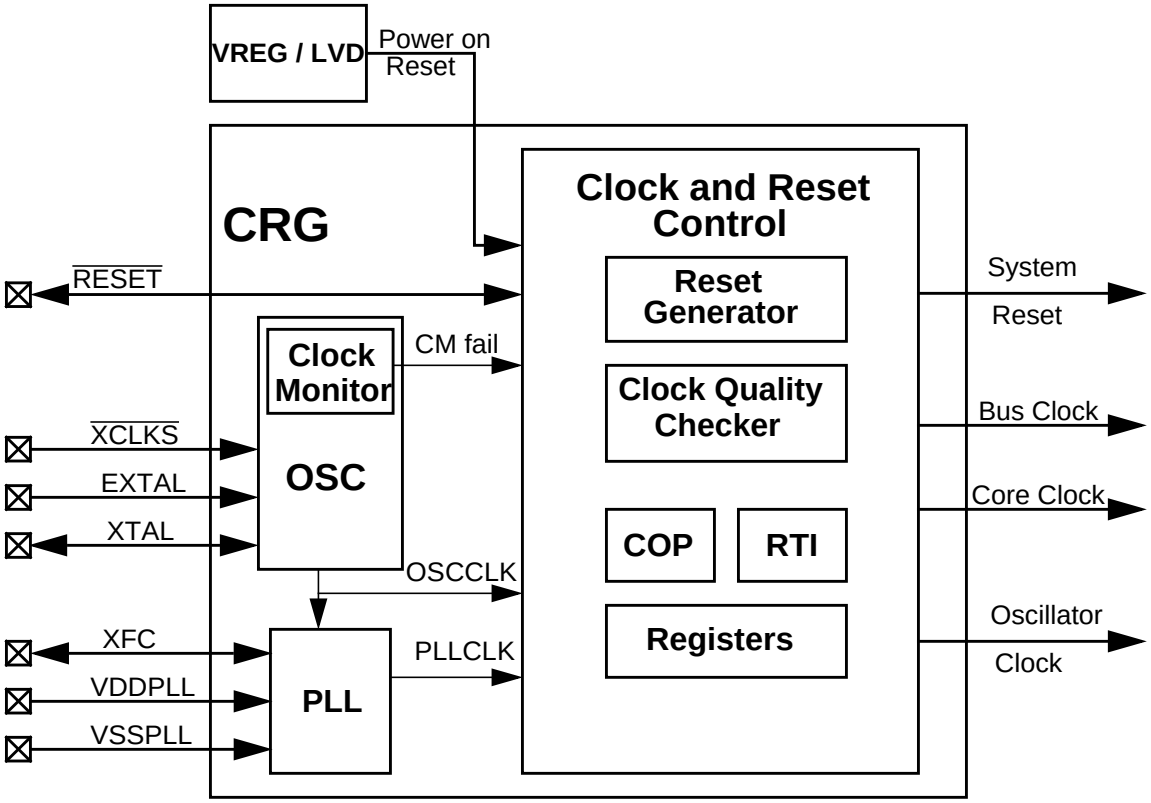


Figure 40 Block diagram of CRG

Freescale Semiconductor, Inc.

External Pin Descriptions

Overview

This section lists and describes the signals that connect off chip.

Detailed Signal Descriptions

VDDPLL, VSSPLL

Theses pins provides operating voltage (VDDPLL) and ground (VSSPLL) for the PLL circuitry. This allows the supply voltage to the PLL to be independently bypassed. Even if PLL usage is not required VDDPLL and VSSPLL must be connected to properly.

XFC

A passive external loop filter must be placed on the XFC pin. The filter is a second-order, low-pass filter to eliminate the VCO input ripple. The value of the external filter network and the reference frequency determines the speed of the corrections and the stability of the PLL. If PLL usage is not required the XFC pin must be tied to VDDPLL.

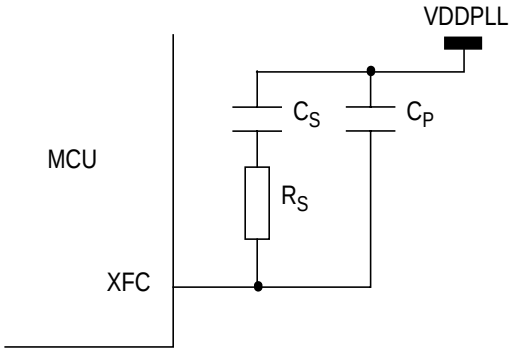


Figure 41 PLL Loop Filter Connections

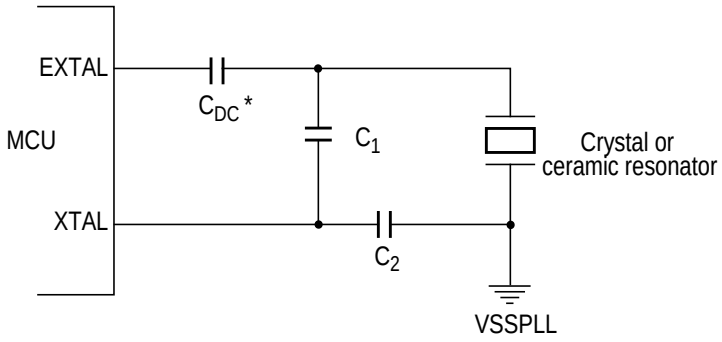
EXTAL, XTAL

These pins provide the interface for either a crystal or a CMOS compatible clock to control the internal clock generator circuitry. EXTAL is the external clock input or the input to the crystal oscillator amplifier. XTAL is the output of the crystal oscillator amplifier. All the MCU internal system clocks are derived from the EXTAL input frequency.

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

Freescale Semiconductor, Inc.

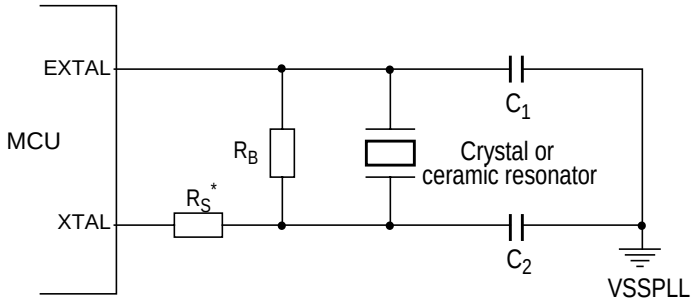
NOTE: *Crystal circuit is changed from standard!*



* Due to the nature of a translated ground Colpitts oscillator a DC voltage bias is applied to the crystal

. .Please contact the crystal manufacturer for crystal DC bias conditions and recommended capacitor value C_{DC}.

Figure 42 Colpitts Crystal Connections ($\overline{\text{XCLKS}}$=1)



* Rs can be zero (shorted) when use with higher frequency crystals. Refer to manufacturer's data.

Figure 43 Pierce Oscillator Connections ($\overline{\text{XCLKS}}$=0)

Freescale Semiconductor, Inc.

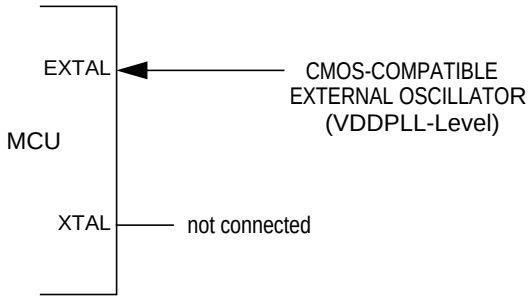


Figure 44 External Clock Connections ($\overline{\text{XCLKS}}$=0)

$\overline{\text{RESET}}$

$\overline{\text{RESET}}$ is an active low bidirectional reset pin. As an input it initializes the MCU asynchronously to a known start-up state. As an open-drain output it indicates that an system reset (internal to MCU) has been triggered.

$\overline{\text{XCLKS}}$

The $\overline{\text{XCLKS}}$ is an input signal which controls whether a crystal in combination with the internal Colpitts (low power) oscillator is used or whether Pierce oscillator/external clock circuitry is used. The $\overline{\text{XCLKS}}$ pin is sampled during reset with the rising edge of $\overline{\text{RESET}}$. Table 50 lists the state coding of the sampled $\overline{\text{XCLKS}}$ signal.

Table 50 Clock Selection Based on $\overline{\text{XCLKS}}$ at reset

$\overline{\text{XCLKS}}$	Description
1	Colpitts Oscillator selected
0	Pierce Oscillator / External clock selected

Freescale Semiconductor, Inc.

Register Map

The register map for the CRG appears below.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0	Address offset
SYNR	Read:	0	0	SYN5	SYN4	SYN3	SYN2	SYN1	SYN0	$0034
	Write:									
REFDV	Read:	0	0	0	0	REFDV3	REFDV2	REFDV1	REFDV0	$0035
	Write:									
Reserved for Factory Test	Read:	Reads to this register return unpredictable values								$0036
	Write:									
CRGFLG	Read:	RTIF	PORLVDRF	0	LOCKIF	LOCK	TRACK	SCMIF	SCM	$0037
	Write:									
CRGINT	Read:	RTIE	0	0	LOCKIE	0	0	SCMIE	0	$0038
	Write:									
CLKSEL	Read:	PLLSEL	PSTP	SYSWAI	ROAWAI	PLLWAI	CWAI	RTIWAI	COPWAI	$0039
	Write:									
PLLCTL	Read:	CME	PLLON	AUTO	ACQ	0	PRE	PCE	SCME	$003A
	Write:									
RTICTL	Read:	0	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0	$003B
	Write:									
COPCTL	Read:	WCOP	RSBCK	0	0	0	CR2	CR1	CR0	$003C
	Write:									
Reserved for Factory Test	Read:	Reads to this register return unpredictable values								$003D
	Write:									

= Reserved or unimplemented

Figure 45 CRG Register Map

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

Freescale Semiconductor, Inc.

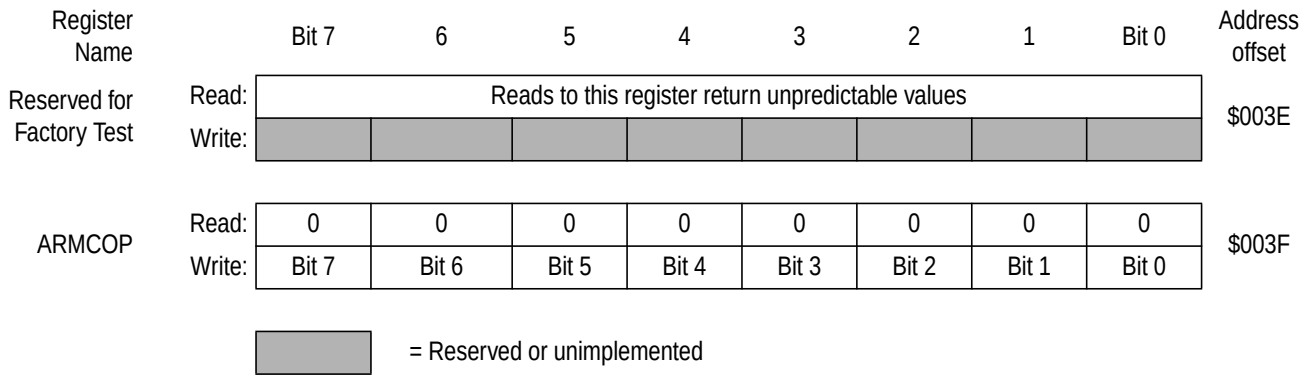


Figure 45 CRG Register Map (Continued)

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Register Descriptions

NOTE: *All bits of all registers in this module are completely synchronous to internal clocks during a register read.*

CRG Synthesizer Register (SYNR)

The SYNR register controls the multiplication factor of the PLL. If the PLL is on, the count in the loop divider (SYNR) register effectively multiplies up the PLLCLK from the reference frequency by 2 x (SYNR+1). PLLCLK will not be below the minimum VCO frequency (f_{SCM}).

$$PLLCLK = 2xOSCCLKx\frac{(SYNR+1)}{(REFDV+1)}$$

NOTE: *If PLL is selected (PLLSEL=1)PLLCLK, Bus Clock =PLLCLK/2. Bus Clock must not exceed the maximum operating system frequency*

.

Address Offset: $0034

	Bit 7	6	5	4	3	2	1	0
Read:	0	0	SYN5	SYN4	SYN3	SYN2	SYN1	SYN0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or reserved

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

Freescale Semiconductor, Inc.

Read: anytime.

Write: anytime except if PLLSEL = 1.

NOTE: *Write to this register initializes the lock detector bit and the track detector bit.*

CRG Reference Divider Register (REFDV)

The REFDV register provides a finer granularity for the PLL multiplier steps. The count in the reference divider divides OSCCLK frequency by REFDV+1.

Address Offset: $0035

	Bit 7	6	5	4	3	2	1	0
Read:	0	0	0	0	REFDV3	REFDV2	REFDV1	REFDV0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or reserved

Read: anytime.

Write: anytime except when PLLSEL = 1.

NOTE: *Write to this register initializes the lock detector bit and the track detector bit.*

CRG Flags Register (CRGFLG)

This register provides CRG status bits and flags.

Address Offset: $0037

	Bit 7	6	5	4	3	2	1	0
Read:	RTIF	PORLVDRF	0	LOCKIF	LOCK	TRACK	SCMIF	SCM
Write:								
Reset:	0	(1)	0	0	0	0	0	0

= Unimplemented or reserved

1. PORLVDRF set to 1 when a power on reset or low-voltage detection reset occurs. Unaffected by non-POR resets.

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

Freescale Semiconductor, Inc.

Read: anytime.

Write: refer to each bit for write conditions.

RTIF — Real Time Interrupt Flag

RTIF is set to 1 at the end of every RTI period. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (RTIE=1), RTIF causes an interrupt request.

1 = RTI time-out has occurred.
0 = RTI time-out has not yet occurred.

PORLVDRF — Power on Reset / Low-Voltage-Detector Reset Flag

PORLVDRF is set to 1 when a power on reset or low-voltage reset occurs. This flag can only be cleared by writing a 1. Writing a 0 has no effect.

1 = Power on reset or a low voltage reset has occurred
0 = Neither power on reset nor low voltage reset has occurred

LOCKIF — PLL Lock Interrupt Flag

LOCKIF is set to 1 when LOCK status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect.If enabled (LOCKIE=1), LOCKIF causes an interrupt request.

1 = LOCK bit has changed.
0 = No change in LOCK bit

LOCK — Lock Status Bit

LOCK reflects the current state of PLL lock condition. This bit is cleared in Self Clock Mode. Writes have no effect.

1 = PLL VCO is within the desired tolerance of the target frequency.
0 = PLL VCO is not within the desired tolerance of the target frequency.

TRACK — Track Status Bit

TRACK reflects the current state of PLL track condition. This bit is cleared in Self Clock Mode. Writes have no effect. See Acquisition and Tracking Modes in page 296 for more information.

1 = Tracking mode status.
0 = Acquisition mode status.

Freescale Semiconductor, Inc.

SCMIF — Self-clock mode Interrupt Flag

SCMIF is set to 1 when SCM status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (SCMIE=1), SCMIF causes an interrupt request.

1 = SCM condition has changed, either entered or exited self-clock mode.

0 = No change in SCM bit.

SCM — Self-clock mode Status Bit

SCM reflects the current clocking mode. Writes have no effect.

1 = MCU is operating in Self Clock Mode with OSCCLK in an unknown state. All clocks are derived from PLLCLK running at its minimum frequency f_{SCM}. See Table 118 in page 582 for the actual value of this parameter.

0 = MCU is operating normally with OSCCLK available.

CRG Interrupt Enable Register (CRGINT)

This register enables CRG interrupt requests.

Address Offset: $0038

	Bit 7	6	5	4	3	2	1	0
Read:	RTIE	0	0	LOCKIE	0	0	SCMIE	0
Write:								
Reset:	0	0	0	0	0	0	0	0

Read: anytime

Write: anytime.

RTIE — Real Time Interrupt Enable Bit

1 = Interrupt will be requested whenever RTIF is set.

0 = Interrupt requests from RTI are disabled.

LOCKIE — Lock Interrupt Enable Bit

1 = Interrupt will be requested whenever LOCKIF is set.

0 = LOCK interrupt requests are disabled.

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

Freescale Semiconductor, Inc.

SCMIE — Self-clock mode Interrupt Enable Bit

1 = Interrupt will be requested whenever SCMIF is set.

0 = SCM interrupt requests are disabled.

CRG Clock Select Register (CLKSEL)

This register controls CRG clock selection.

Address Offset: $0039

	Bit 7	6	5	4	3	2	1	0
Read: Write:	PLLSEL	PSTP	SYSWAI	ROAWAI	PLLWAI	CWAI	RTIWAI	COPWAI
Reset:	0	0	0	0	0	0	0	0

Read: anytime.

Write: refer to each bit for individual write conditions.

PLLSEL — PLL Select Bit

Write: anytime

Writing a one when LOCK=0 and AUTO=1, or TRACK=0 and AUTO=0 has no effect. This prevents the selection of an unstable PLLCLK as SYSCLK.

PLLSEL bit is cleared when the MCU enters Self Clock Mode, Stop Mode or Wait Mode with PLLWAI bit set.

1 = SYSCLK is derived from PLLCLK. (Bus Clock=PLLCLK/2)

0 = SYSCLK is derived from OSCCLK. (Bus Clock=OSCCLK/2)

PSTP — Pseudo Stop Bit

Write: anytime

This bit controls the functionality of the oscillator during Stop Mode.

1 = Oscillator continues to run in Stop Mode (Pseudo Stop). The oscillator amplitude is reduced.

0 = Oscillator is disabled in Stop Mode.

NOTE: *Pseudo-STOP allows for faster STOP recovery and reduces the mechanical stress and aging of the resonator in case of frequent STOP conditions at the expense of a slightly increased power consumption.*

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

Freescale Semiconductor, Inc.

Lower oscillator amplitude exhibits lower power consumption but could have adverse effects during any Electro-Magnetic Susceptibility (EMS) tests.

SYSWAI — System clocks stop in WAIT Mode Bit

Write: anytime

This bit controls the bus clock during Wait Mode.
1 = In Wait Mode, the system clocks stop.
0 = In Wait Mode, the system clocks continue to run.

NOTE: *RTI and COP are not affected by SYSWAI bit.*

ROAWAI — Reduced Oscillator Amplitude in WAIT Mode Bit

Write: anytime

This bit controls oscillator amplitude during Wait Mode.
1 = Reduced oscillator amplitude in Wait Mode.
0 = Normal oscillator amplitude in Wait Mode.

NOTE: *Lower oscillator amplitude exhibit lower power consumption but could have adverse effects during any Electro-Magnetic Susceptibility (EMS) tests.*

PLLWAI — PLL stops in WAIT Mode Bit

Write: anytime

If PLLWAI is set, the CRG will clear the PLLSEL bit before entering Wait Mode. The PLLON bit remains set during Wait Mode but the PLL is powered down. Upon exiting Wait Mode, the PLLSEL bit has to be set manually if PLL clock is required.

While the PLLWAI bit is set the AUTO bit is set to 1 in order to allow the PLL to automatically lock on the selected target frequency after exiting Wait Mode.
1 = PLL stops in Wait Mode.
0 = PLL keeps running in Wait Mode.

CWAI — Core stops in WAIT Mode Bit

Write: anytime

This bit controls the core clock in Wait Mode.

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

Freescale Semiconductor, Inc.

1 = Core clock stops in Wait Mode.
0 = Core clock keeps running in Wait Mode.

RTIWAI — RTI stops in WAIT Mode Bit

Write: anytime
1 = RTI stops and initializes the RTI dividers whenever the part goes into Wait Mode.
0 = RTI keeps running in Wait Mode.

COPWAI — COP stops in WAIT Mode Bit

Write: once
1 = COP stops and initializes the COP dividers whenever the part goes into Wait Mode.
0 = COP keeps running in Wait Mode.

CRG PLL Control Register (PLLCTL)

This register controls the PLL functionality.

Address Offset: $003A

	Bit 7	6	5	4	3	2	1	0
Read:	CME	PLLON	AUTO	ACQ	0	PRE	PCE	SCME
Write:								
Reset:	1	1	1	1	0	0	0	1

Read: anytime.

Write: refer to each bit for individual write conditions.

CME — Clock Monitor Enable Bit

Write: anytime except when SCM = 1

CME enables the clock monitor.
1 = Clock monitor is enabled. Slow or stopped clocks will cause a clock monitor reset sequence or Self Clock Mode.
0 = Clock monitor is disabled.

NOTE: *Operating with CME=0 will not detect any loss of clock. In case of poor clock quality this could cause unpredictable operation of the MCU!*

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

Freescale Semiconductor, Inc.

In Stop Mode (PSTP=0) the clock monitor is disabled independently of the CME bit setting and any loss of clock will not be detected.

PLLON — Phase Lock Loop On Bit

Write: anytime except when PLLSEL = 1.

PLLON turns on the PLL circuitry. In Self Clock Mode, the PLL is turned on, but the PLLON bit reads the last latched value.

1 = PLL is turned on. If AUTO bit is set, the PLL will lock automatically.

0 = PLL is turned off.

AUTO — Automatic Bandwidth Control Bit

Write: anytime except when PLLWAI=1, because PLLWAI sets the AUTO bit to 1

AUTO selects either the high bandwidth (acquisition) mode or the low bandwidth (tracking) mode depending on how close to the desired frequency the VCO is running.

1 = Automatic Mode Control is enabled and ACQ bit has no effect.

0 = Automatic Mode Control is disabled and the PLL is under software control, using ACQ bit.

Freescale Semiconductor, Inc.

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

ACQ — Acquisition Bit

Write: anytime. If AUTO =1 this bit has no effect.

1 = High bandwidth filter is selected

0 = Low bandwidth filter is selected

PRE — RTI Enable during Pseudo Stop Mode Bit

PRE enables the RTI during Pseudo Stop Mode.

1 = RTI continues running during Pseudo Stop Mode.

0 = RTI stops during Pseudo Stop Mode.

NOTE: *If the PRE bit is cleared the RTI dividers will go static while Pseudo Stop Mode is active. The RTI dividers will not initialize like in Wait Mode with RTIWAI bit set.*

PCE — COP Enable during Pseudo Stop Mode Bit

PCE enables the COP during Pseudo Stop Mode.

1 = COP continues running during Pseudo Stop Mode.

0 = COP stops during Pseudo Stop Mode.

NOTE: *If the PCE bit is cleared the COP dividers will go static while Pseudo Stop Mode is active. The COP dividers will not initialize like in Wait Mode with COPWAI bit set.*

SCME — Self-clock mode enable

SCME can not be cleared while operating in Self Clock Mode (SCM=1).

Write: once

1 = Detection of crystal clock failure forces the MCU in self-clock mode only.

0 = Detection of crystal clock failure causes clock monitor reset.

CRG RTI Control Register (RTICTL)

This register selects the timeout period for the Real Time Interrupt.

Address Offset: $003B

	Bit 7	6	5	4	3	2	1	0
Read:	0	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
Write:								

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

Freescale Semiconductor, Inc.

Address Offset: $003B

	Bit 7	6	5	4	3	2	1	0
Reset:	0	0	0	0	0	0	0	0

Read: anytime

Write: anytime.

NOTE: *A write to this register will initialize the RTI counter.*

RTR[6:4] — Real Time Interrupt Prescale Rate Select Bits

These bits select the prescale rate for the RTI. See Table 51.

RTR[3:0] — Real Time Interrupt Modulus Counter Select

These bits select the modulus counter target value to provide additional granularity. See Table 51.

Table 51 shows all possible divide values selectable by the RTICTL register. The source clock for the RTI is OSCCLK.

Table 51 RTI Frequency Divide Rates

RTR[3:0]	RTR[6:4] =							
	000 (OFF)	**001 (2^{10})**	**010 (2^{11})**	**011 (2^{12})**	**100 (2^{13})**	**101 (2^{14})**	**110 (2^{15})**	**111 (2^{16})**
0 (÷1)	OFF (1)(2)	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
1 (÷2)	OFF (2)	$2x2^{10}$	$2x2^{11}$	$2x2^{12}$	$2x2^{13}$	$2x2^{14}$	$2x2^{15}$	$2x2^{16}$
2 (÷3)	OFF (2)	$3x2^{10}$	$3x2^{11}$	$3x2^{12}$	$3x2^{13}$	$3x2^{14}$	$3x2^{15}$	$3x2^{16}$
3 (÷4)	OFF (2)	$4x2^{10}$	$4x2^{11}$	$4x2^{12}$	$4x2^{13}$	$4x2^{14}$	$4x2^{15}$	$4x2^{16}$
4 (÷5)	OFF (2)	$5x2^{10}$	$5x2^{11}$	$5x2^{12}$	$5x2^{13}$	$5x2^{14}$	$5x2^{15}$	$5x2^{16}$
5 (÷6)	OFF (2)	$6x2^{10}$	$6x2^{11}$	$6x2^{12}$	$6x2^{13}$	$6x2^{14}$	$6x2^{15}$	$6x2^{16}$
6 (÷7)	OFF (2)	$7x2^{10}$	$7x2^{11}$	$7x2^{12}$	$7x2^{13}$	$7x2^{14}$	$7x2^{15}$	$7x2^{16}$
7 (÷8)	OFF (2)	$8x2^{10}$	$8x2^{11}$	$8x2^{12}$	$8x2^{13}$	$8x2^{14}$	$8x2^{15}$	$8x2^{16}$
8 (÷9)	OFF (2)	$9x2^{10}$	$9x2^{11}$	$9x2^{12}$	$9x2^{13}$	$9x2^{14}$	$9x2^{15}$	$9x2^{16}$
9 (÷10)	OFF (2)	$10x2^{10}$	$10x2^{11}$	$10x2^{12}$	$10x2^{13}$	$10x2^{14}$	$10x2^{15}$	$10x2^{16}$
10 (÷11)	OFF (2)	$11x2^{10}$	$11x2^{11}$	$11x2^{12}$	$11x2^{13}$	$11x2^{14}$	$11x2^{15}$	$11x2^{16}$
11 (÷12)	OFF (2)	$12x2^{10}$	$12x2^{11}$	$12x2^{12}$	$12x2^{13}$	$12x2^{14}$	$12x2^{15}$	$12x2^{16}$
12 (÷13)	OFF (2)	$13x2^{10}$	$13x2^{11}$	$13x2^{12}$	$13x2^{13}$	$13x2^{14}$	$13x2^{15}$	$13x2^{16}$

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

Table 51 RTI Frequency Divide Rates (Continued)

RTR[3:0]	RTR[6:4] =							
	000 (OFF)	**001 (2^{10})**	**010 (2^{11})**	**011 (2^{12})**	**100 (2^{13})**	**101 (2^{14})**	**110 (2^{15})**	**111 (2^{16})**
13 (÷14)	OFF (2)	$14x2^{10}$	$14x2^{11}$	$14x2^{12}$	$14x2^{13}$	$14x2^{14}$	$14x2^{15}$	$14x2^{16}$
14 (÷15)	OFF (2)	$15x2^{10}$	$15x2^{11}$	$15x2^{12}$	$15x2^{13}$	$15x2^{14}$	$15x2^{15}$	$15x2^{16}$
15 (÷16)	OFF (2)	$16x2^{10}$	$16x2^{11}$	$16x2^{12}$	$16x2^{13}$	$16x2^{14}$	$16x2^{15}$	$16x2^{16}$

1. Denotes default value out of reset

2. These values should be used to disable the RTI to ensure future backwards compatibility.

Freescale Semiconductor, Inc.

CRG COP Control Register (COPCTL)

This register controls the COP (Computer Operating Properly) watchdog.

Address Offset: $003C

	Bit 7	6	5	4	3	2	1	0
Read:	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
Write:								
Reset:	0	1	0	0	0	0	1	1

Read: anytime.

Write: once

WCOP — Window COP mode Bit

When set, a write to the ARMCOP register must occur in the last 25% of the selected period. A write during the first 75% of the selected period will reset the part. As long as all writes occur during this window, $55 can be written as often as desired. Once $AA is written after the $55, the time-out logic restarts and the user must wait until the next window before writing to ARMCOP. Table 52 shows the exact duration of this window for the seven available COP rates.

1 = Window COP operation
0 = Normal COP operation

RSBCK — COP and RTI stop in Active BDM Bit

1 = Stops the COP and RTI counters whenever the part is in Active background debug mode.
0 = Allows the COP and RTI to keep running in Active background debug mode.

CR[2:0] — COP Watchdog Timer Rate select

These bits select the COP time-out rate (see Table 52). The COP time-out period is OSCCLK period divided by CR[2:0] value. Writing a nonzero value to CR[2:0] enables the COP counter and starts the time-out period. A COP counter time-out causes a system reset. This can be avoided by periodically (before time-out) re-initializing the COP counter via the ARMCOP register.

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

Freescale Semiconductor, Inc.

Table 52 COP Watchdog Rates(1)

CR2	CR1	CR0	Divide OSCCLK by
0	0	0	OFF
0	0	1	2^{14}
0	1	0	2^{16}
0	1	1	2^{18}
1	0	0	2^{20}
1	0	1	2^{22}
1	1	0	2^{23}
1	1	1	2^{24}

1. Times are referenced from the previous COP time-out reset (writing $55/$AA to the ARMCOP register)

CRG COP Timer Arm/Reset Register (ARMCOP)

This register is used to restart the COP time-out period.

Address Offset: $003F

	Bit 7	6	5	4	3	2	1	0
Read:	0	0	0	0	0	0	0	0
Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	0	0	0	0	0	0	0	0

Read: always reads $00.

Write: anytime

When the COP is disabled (CR[2:0] = “000”) writing to this register has no effect.

When the COP is enabled by setting CR[2:0] nonzero, the following applies:

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

Freescale Semiconductor, Inc.

Writing any value other than $55 or $AA causes a COP reset. To restart the COP time-out period you must write $55 followed by a write of $AA. Other instructions may be executed between these writes but the sequence ($55, $AA) must be completed prior to COP end of time-out period to avoid a COP reset. Sequences of $55 writes or sequences of $AA writes are allowed. When the WCOP bit is set, $55 and $AA writes must be done in the last 25% of the selected time-out period; writing any value in the first 75% of the selected period will cause a COP reset.

Freescale Semiconductor, Inc.

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

Functional Description

General

This section provides a complete functional description of the CRG. It gives detailed informations on the internal operation of the design.

Functional Blocks

Oscillator (OSC)

The oscillator block has two external pins, EXTAL and XTAL. The oscillator input pin, EXTAL, is intended to be connected to either a crystal or an external clock source. The selection of Colpitts oscillator or Pierce Oscillator/External clock depends on the $\overline{\text{XCLKS}}$ signal which is sampled during reset. The XTAL pin is an output signal that provides crystal circuit feedback and can be buffered to drive other devices with same voltage amplitude.

A buffered EXTAL signal, OSCCLK, becomes the internal reference clock. The oscillator is enabled based on the PSTP bit, and the STOP condition. The oscillator is disabled when the part is in STOP mode except when Pseudo-Stop mode is enabled.

To improve noise immunity, the oscillator is powered by the VDDPLL and VSSPLL power supply pins.

The Colpitts oscillator is equipped with a feedback system which does not waste current generating harmonics. Its configuration is “Colpitts oscillator with translated ground”. The transconductor used is driven by a current source under the control of a peak detector which will measure the amplitude of the AC signal appearing on EXTAL node in order to implement an Amplitude Limitation Control (ALC) loop. The ALC loop is in charge of reducing the quiescent current in the transconductor as a result of an increase in the oscillation amplitude.

The Pierce Oscillator can be used for higher frequencies than possible with the Colpitts oscillator.

Freescale Semiconductor, Inc.

Phase Locked Loop (PLL)

The PLL is used to run the MCU from a different time base than the incoming OSCCLK. For increased flexibility, OSCCLK can be divided in a range of 1 to 16 to generate the reference frequency. This offers a finer multiplication granularity. The PLL can multiply this reference clock by a multiple of 2, 4, 6, ...,126,128 based on the SYNR register.

$$PLLCLK = 2 \times OSCCLK \times \frac{[SYNR + 1]}{[REFDV + 1]}$$

NOTE: *Although it is possible to set the two dividers to command a very high clock frequency, do not exceed the specified bus frequency limit for the MCU. If PLLSEL=1, Bus Clock = PLLCLK/2.*

The PLL is a frequency generator that operates in either acquisition mode or tracking mode, depending on the difference between the output frequency and the target frequency. The PLL can change between acquisition and tracking modes either automatically or manually.

The VCO has a minimum operating frequency, which corresponds to the self clock mode frequency f_{SCM}.

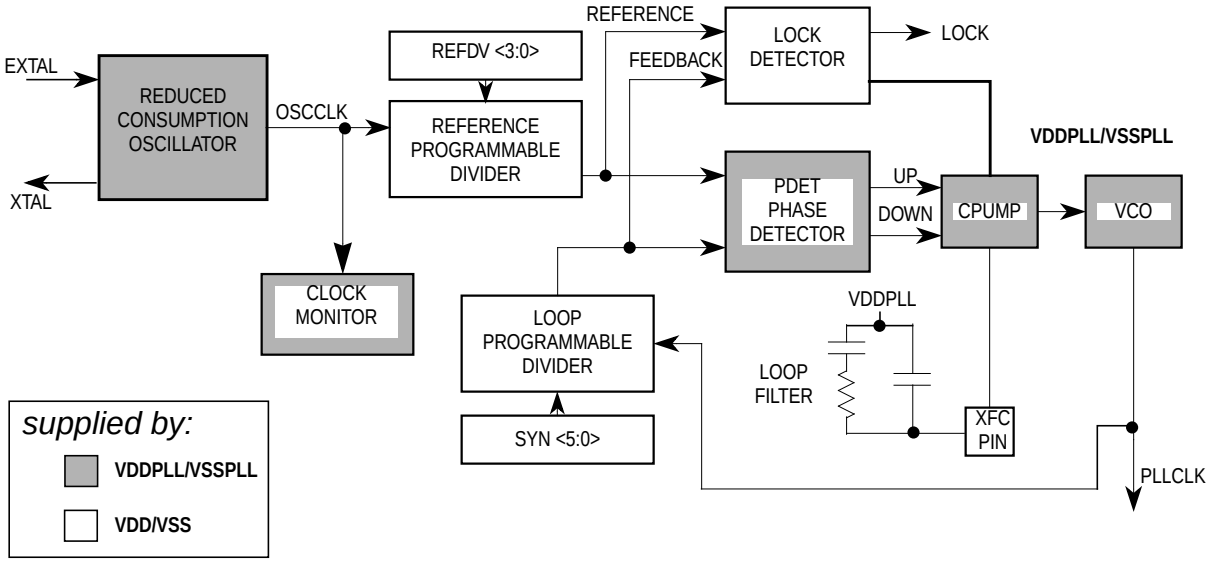


Figure 46 PLL Functional Diagram

PLL Operation

The oscillator output clock signal (OSCCLK) is fed through the reference programmable divider and is divided in a range of 1 to 16 (REFDV+1) to

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

output the REFERENCE clock. The VCO output clock, (PLLCLK) is fed back through the programmable loop divider and is divided in a range of 2 to 128 in increments of [2 x (SYNR +1)] to output the FEEDBACK clock. See Figure 46.

The phase detector then compares the FEEDBACK clock, with the REFERENCE clock. Correction pulses are generated based on the phase difference between the two signals. The loop filter then slightly alters the DC voltage on the external filter capacitor connected to XFC pin, based on the width and direction of the correction pulse. The filter can make fast or slow corrections depending on its mode, as described in the next subsection. The values of the external filter network and the reference frequency determine the speed of the corrections and the stability of the PLL.

Acquisition and Tracking Modes

The lock detector compares the frequencies of the FEEDBACK clock, and the REFERENCE clock. Therefore, the speed of the lock detector is directly proportional to the final reference frequency. The circuit determines the mode of the PLL and the lock condition based on this comparison[1].

The PLL filter can be manually or automatically configured into one of two possible operating modes:

- Acquisition mode

  In acquisition mode, the filter can make large frequency corrections to the VCO. This mode is used at PLL start-up or when the PLL has suffered a severe noise hit and the VCO frequency is far off the desired frequency. When in acquisition mode, the TRACK status bit is cleared in the CRGFLG register.

- Tracking mode

1. See Table 118 in page 582 for actual values of the parameters mentioned in this section.

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

Freescale Semiconductor, Inc.

In tracking mode, the filter makes only small corrections to the frequency of the VCO. PLL jitter is much lower in tracking mode, but the response to noise is also slower. The PLL enters tracking mode when the VCO frequency is nearly correct and the TRACK bit is set in the CRGFLG register.

The PLL can change the bandwidth or operational mode of the loop filter manually or automatically.

In automatic bandwidth control mode (AUTO = 1), the lock detector automatically switches between acquisition and tracking modes. Automatic bandwidth control mode also is used to determine when the PLL clock (PLLCLK) is safe to use as the source for the system and core clocks. If PLL LOCK interrupt requests are enabled, the software can wait for an interrupt request and then check the LOCK bit. If CPU interrupts are disabled, software can poll the LOCK bit continuously (during PLL start-up, usually) or at periodic intervals. In either case, only when the LOCK bit is set, is the PLLCLK clock safe to use as the source for the system and core clocks. If the PLL is selected as the source for the system and core clocks and the LOCK bit is clear, the PLL has suffered a severe noise hit and the software must take appropriate action, depending on the application.

The following conditions apply when the PLL is in automatic bandwidth control mode (AUTO=1):

- The TRACK bit is a read-only indicator of the mode of the filter.
- The LOCK bit is a read-only indicator of the locked state of the PLL.
- CPU interrupts can occur if enabled (LOCKIE = 1) when the lock condition changes, toggling the LOCK bit.

The PLL can also operate in manual mode (AUTO = 0). Manual mode is used by systems that do not require an indicator of the lock condition for proper operation. Such systems typically operate well below the maximum system frequency (f_{sys}) and require fast start-up. The following conditions apply when in manual mode:

Freescale Semiconductor, Inc.

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

- ACQ is a writable control bit that controls the mode of the filter. Before turning on the PLL in manual mode, the ACQ bit should be asserted to configure the filter in acquisition mode.
- After turning on the PLL by setting the PLLON bit software must wait a given time before entering tracking mode (ACQ = 0).
- After entering tracking mode software must wait a given time before selecting the PLLCLK as the source for system and core clocks (PLLSEL = 1).

Freescale Semiconductor, Inc.

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

System Clocks Generator

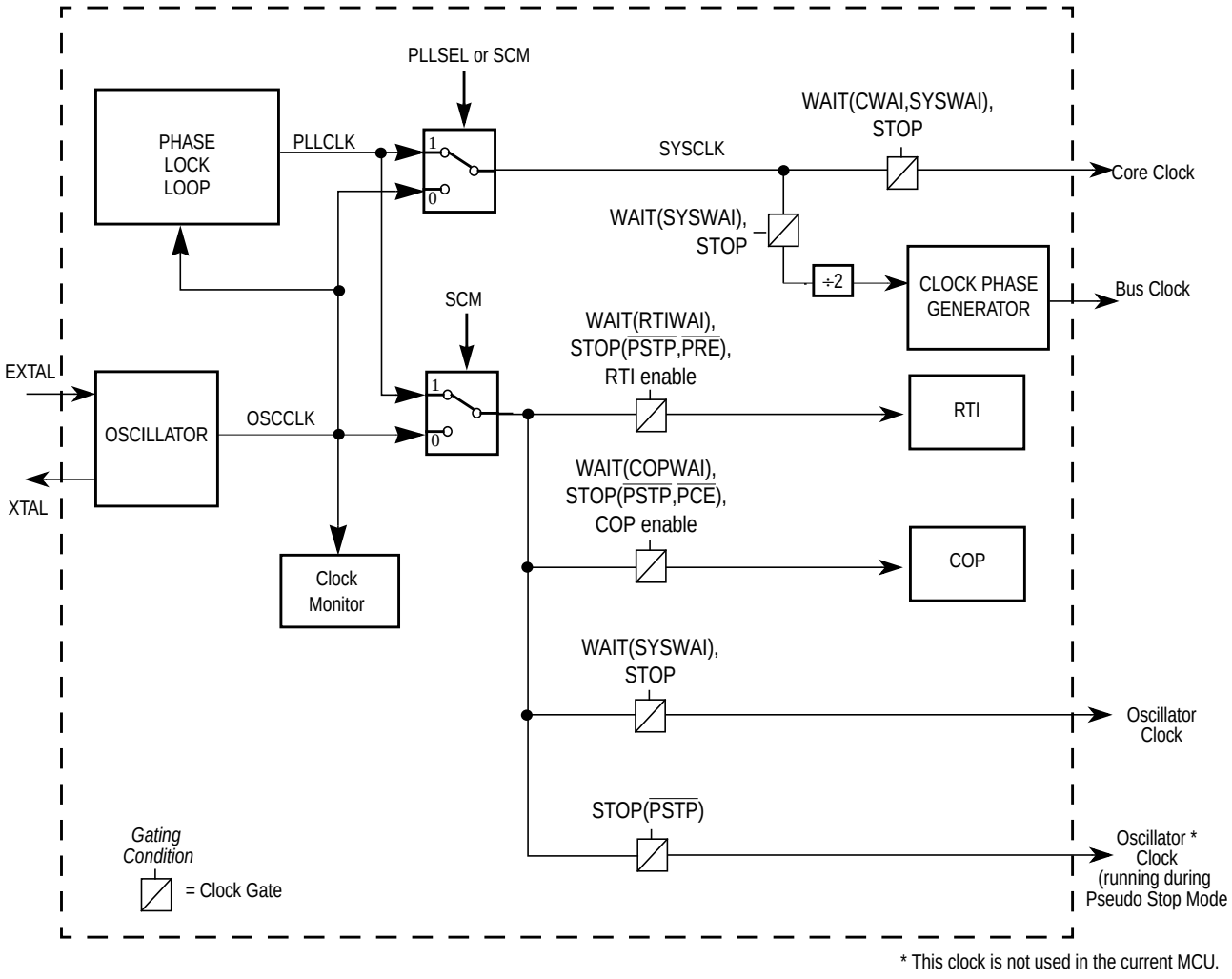


Figure 47 Clock Generator

The clock generator creates the clocks used in the MCU (see Figure 47). The gating condition placed on top of the individual clock gates indicates the dependencies of different modes (STOP, WAIT) and the setting of the respective configuration bits. For example, a WAIT(SYSWAI) gating condition states that when the SYSWAI bit is set, the correspondent gate will be disabled during WAIT mode.

Freescale Semiconductor, Inc.

The peripheral modules use the Bus Clock. Some peripheral modules also use the Oscillator Clock. The memory blocks use the Bus Clock. If the MCU enters Self Clock Mode (see Self Clock Mode in page 305) Oscillator clock source is switched to PLLCLK running at its minimum frequency f_{SCM}. The Bus Clock is used to generate the clock visible at the ECLK pin. The Core Clock signal is the clock for the HCS12 core.The Core Clock is twice the Bus Clock as shown in Figure 48. But note that a CPU cycle corresponds to one Bus Clock.

PLL clock mode is selected with PLLSEL bit in the CLKSEL register. When selected, the PLL output clock drives SYSCLK for the main system including the CPU and peripherals. The PLL cannot be turned off by clearing the PLLON bit, if the PLL clock is selected. When PLLSEL is changed, it takes a maximum of 4 OSCCLK plus 4 PLLCLK cycles to make the transition. During the transition, all clocks freeze and CPU activity ceases.

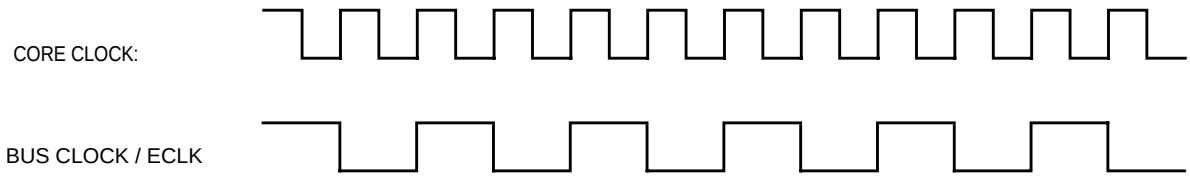


Figure 48 Core Clock and Bus Clock relationship

Clock Monitor (CM)

The clock monitor circuit is based on an internal resistor-capacitor (RC) time delay so that it can operate without any MCU clocks. If no OSCCLK edges are detected within this RC time delay, the clock monitor indicates failure which asserts self clock mode or generates a system reset depending on the state of SCME bit in the PLLCTL register (page 286). If the SCME bit is cleared, CRG generates a Clock Monitor Reset; otherwise, it enters Self Clock Mode. If the clock monitor is disabled or the presence of clocks is detected no failure is indicated.The clock monitor function is enabled/disabled by the CME control bit.

Freescale Semiconductor, Inc.

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

Clock Quality Checker

The clock monitor performs a coarse check on the incoming clock signal. The clock quality checker provides a more accurate check in addition to the clock monitor. The clock quality checker expects a valid OSCLK to have 4096 rising OSCCLK edges within a time window of 50 000 VCO clock cycles[1] (See Figure 49). This time window is called *check window*. If the requested number of 4096 rising OSCCLK edges occur within the check window, the quality of the OSCCLK is considered to be valid and the OSCCLK becomes the source for systems and core clocks. Note that if 4096 OSCCLK edges are counted within the check window, the check window is immediately terminated.

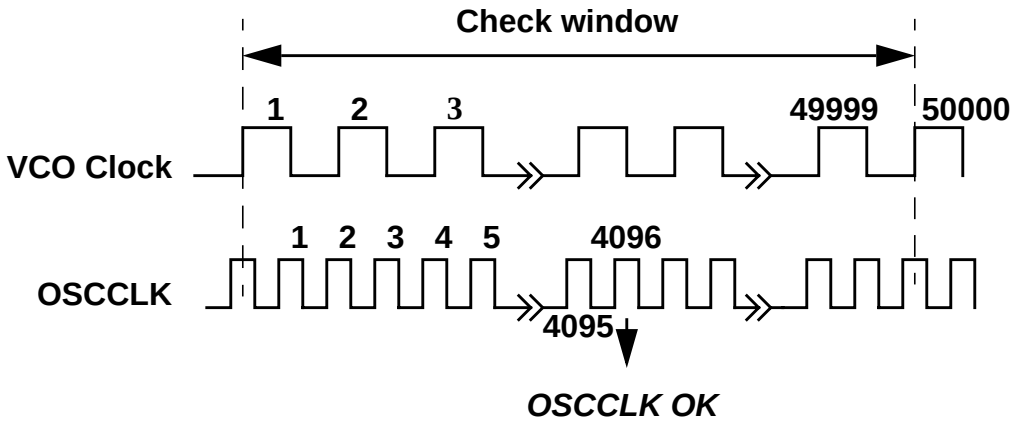


Figure 49 Check Window definition

A clock quality check is triggered by any of the following events:

- Power-on Reset (*POR*)
- Low Voltage Detection Reset (*LVDR*)
- Wake-up from Full Stop Mode (*Exit_Full_Stop*)
- Clock Monitor fail indication (*CM_Fail*)

1. VCO clock cycles are generated by the PLL when running at minimum frequency f_{SCM}.

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

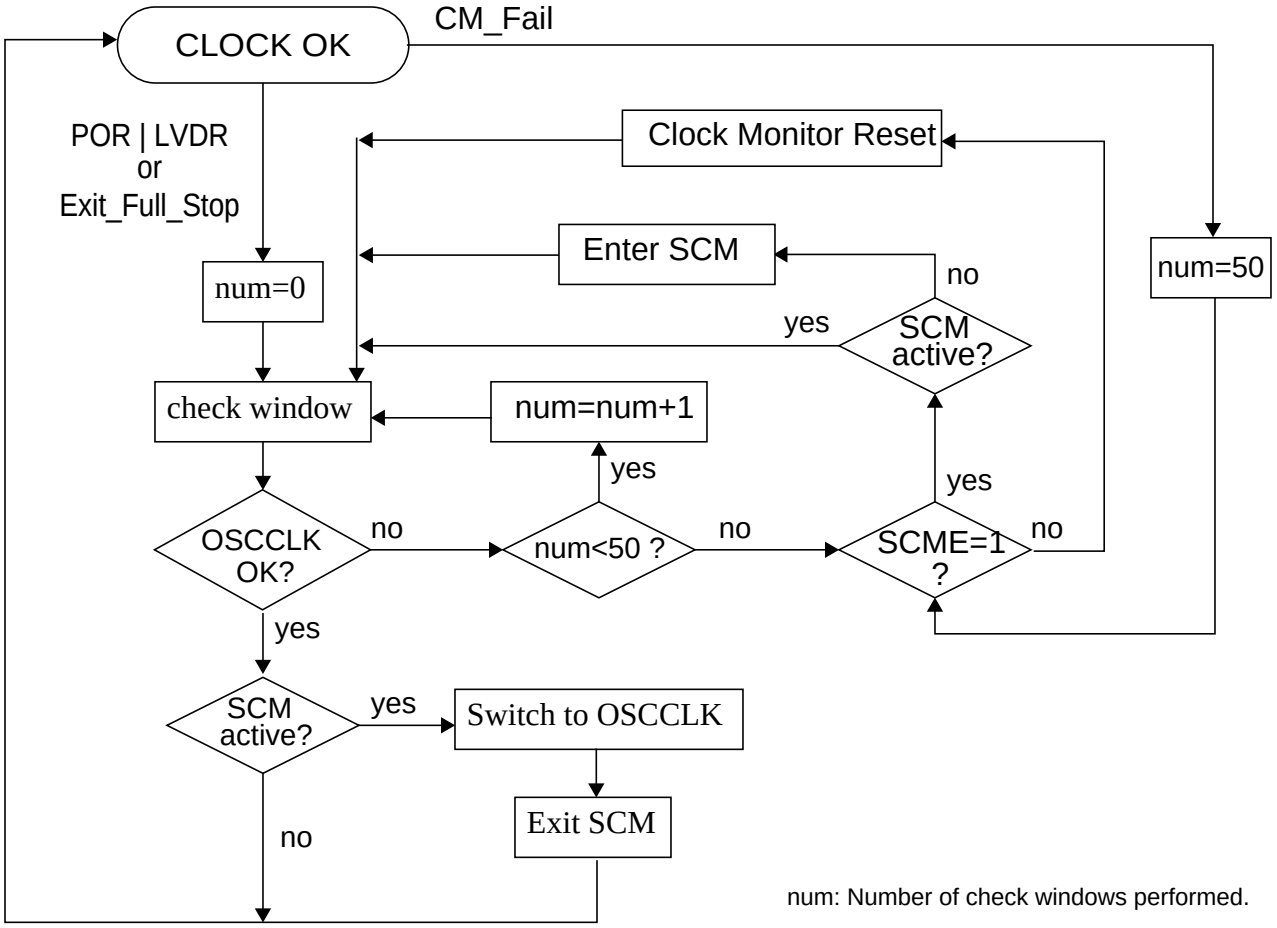


Figure 50 Sequence for Clock Quality Check

NOTE: *Remember that in parallel to additional actions caused by Self Clock Mode or Clock Monitor Reset[1] handling the clock quality checker* ***continues*** *to check the OSCCLK signal.*

NOTE: *The Clock Quality Checker enables the PLL and the voltage regulator (VREG) anytime a clock check has to be performed. An ongoing clock quality check could also cause a running PLL (f_{SCM}) and an active VREG during Pseudo-Stop Mode or Wait Mode*

1. A Clock Monitor Reset will always set the SCME bit to logical'1'

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

Freescale Semiconductor, Inc.

Computer Operating Properly Watchdog (COP)

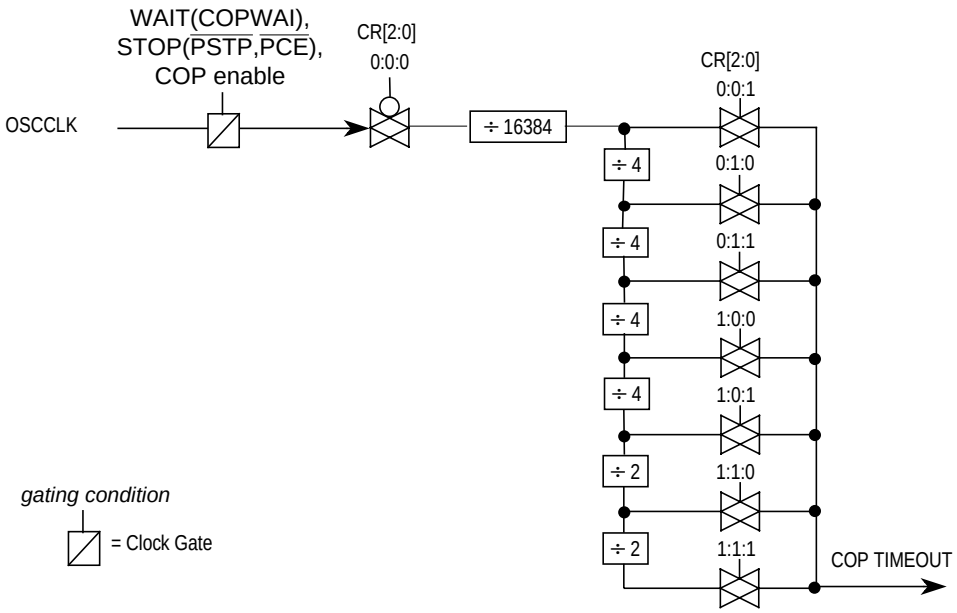


Figure 51 Clock Chain for COP

The COP (free running watchdog timer) enables the user to check that a program is running and sequencing properly. When the COP is being used, software is responsible for keeping the COP from timing out. If the COP times out it is an indication that the software is no longer being executed in the intended sequence; thus a system reset is initiated (see Computer Operating Properly Watchdog (COP) Reset in page 321). The COP runs with a gated OSCCLK (see Figure 51). Three control bits in the COPCTL register allow selection of seven COP time-out periods.

When COP is enabled, the program must write $55 and $AA (in this order) to the ARMCOP register during the selected time-out period. Once this is done, the COP time-out period is restarted. If the program fails to do this and the COP times out, the part will reset. Also, if any value other than $55 or $AA is written, the part is immediately reset.

Window COP operation is enabled by setting WCOP in the COPCTL register. In this mode, writes to the ARMCOP register to clear the COP timer must occur in the last 25% of the selected time-out period. A premature write will immediately reset the part.

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

If PCE bit is set, the COP will continue to run in Pseudo-Stop Mode.

Real Time Interrupt (RTI)

The RTI can be used to generate a hardware interrupt at a fixed periodic rate. If enabled (by setting RTIE=1), this interrupt will occur at the rate selected by the RTICTL register. The RTI runs with a gated OSCCLK (see Figure 52). At the end of the RTI time-out period the RTIF flag is set to one and a new RTI time-out period starts immediately.

A write to the RTICTL register restarts the RTI time-out period.

If the PRE bit is set, the RTI will continue to run in Pseudo-Stop Mode.

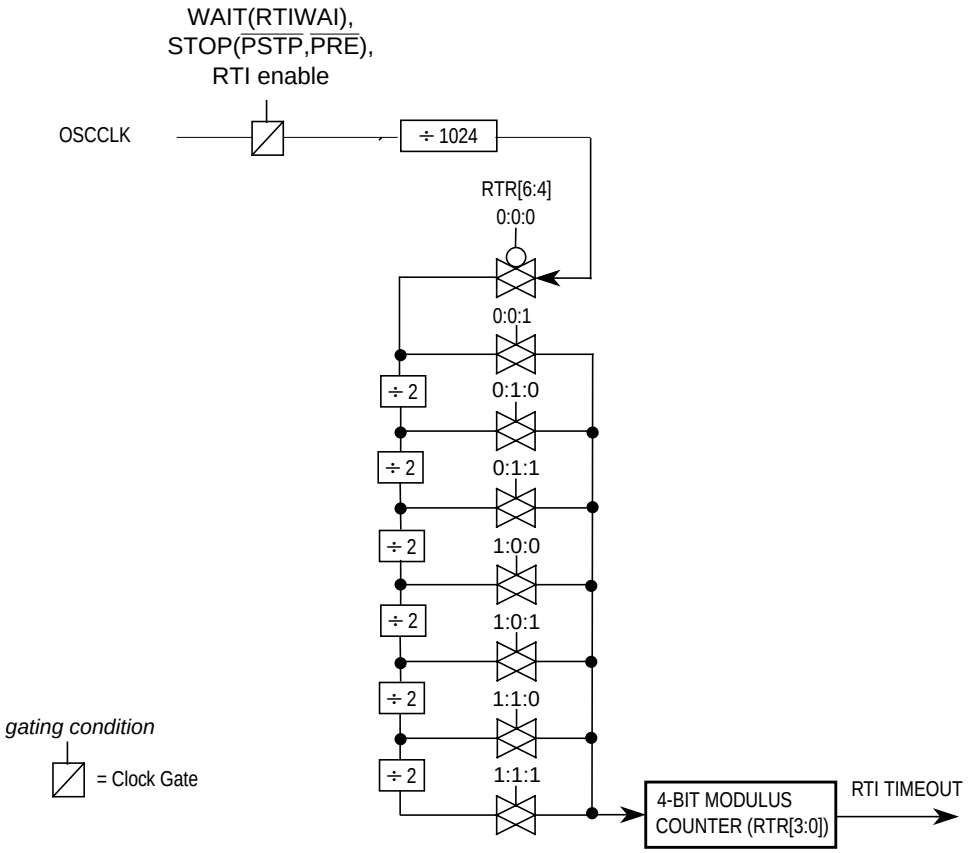


Figure 52 Clock Chain for RTI

Freescale Semiconductor, Inc.

Operation Modes

Normal Mode

The CRG module behaves as described within this specification in all normal modes.

Self Clock Mode

The VCO has a minimum operating frequency, f_{SCM}. If the external clock frequency is not available due to a failure or due to long crystal start-up time, the Bus Clock and the Core Clock are derived from the VCO running at minimum operating frequency; this mode of operation is called Self Clock Mode. This requires CME=1 and SCME=1. If the MCU was clocked by the PLL clock prior to entering Self Clock Mode, the PLLSEL bit will be cleared. If the external clock signal has stabilized again, the CRG will automatically select OSCCLK to be the system clock and return to normal mode. See Clock Quality Checker in page 301 for more information on entering and leaving Self Clock Mode.

NOTE: *In order to detect a potential clock loss the CME bit should be always enabled (CME=1)!*

If CME bit is disabled and the MCU is configured to run on PLL clock (PLLCLK), a loss of external clock (OSCCLK) will not be detected and will cause the system clock to drift towards the VCO's minimum frequency f_{SCM}. As soon as the external clock is available again the system clock ramps up to its PLL target frequency. If the MCU is running on external clock any loss of clock will cause the system to go static.

Low Power Options

This section summarizes the low power options available in the CRG.

Run Mode

The RTI can be stopped by setting the associated rate select bits to zero.

The COP can be stopped by setting the associated rate select bits to zero.

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

Freescale Semiconductor, Inc.

Wait Mode

The WAI instruction puts the MCU in a low power consumption stand-by mode depending on setting of the individual bits in the CLKSEL register. All individual Wait Mode configuration bits can be superposed. This provides enhanced granularity in reducing the level of power consumption during Wait Mode. Table 53 lists the individual configuration bits and the parts of the MCU that are affected in Wait Mode.

Table 53 MCU configuration during Wait Mode

	PLLWAI	CWAI	SYSWAI	RTIWAI	COPWAI	ROAWAI
PLL	stopped	-	-	-	-	-
Core	-	stopped	stopped	-	-	-
System	-	-	stopped	-	-	-
RTI	-	-	-	stopped	-	-
COP	-	-	-	-	stopped	-
Oscillator	-	-	-	-	-	reduced

After executing the WAI instruction the core requests the CRG to switch MCU into Wait Mode. The CRG then checks whether the PLLWAI, CWAI and SYSWAI bits are asserted (see Figure 53). Depending on the configuration the CRG switches the system and core clocks to OSCCLK by clearing the PLLSEL bit, disables the PLL, disables the core clocks and finally disables the remaining system clocks. As soon as all clocks are switched off Wait Mode is active.

Freescale Semiconductor, Inc.

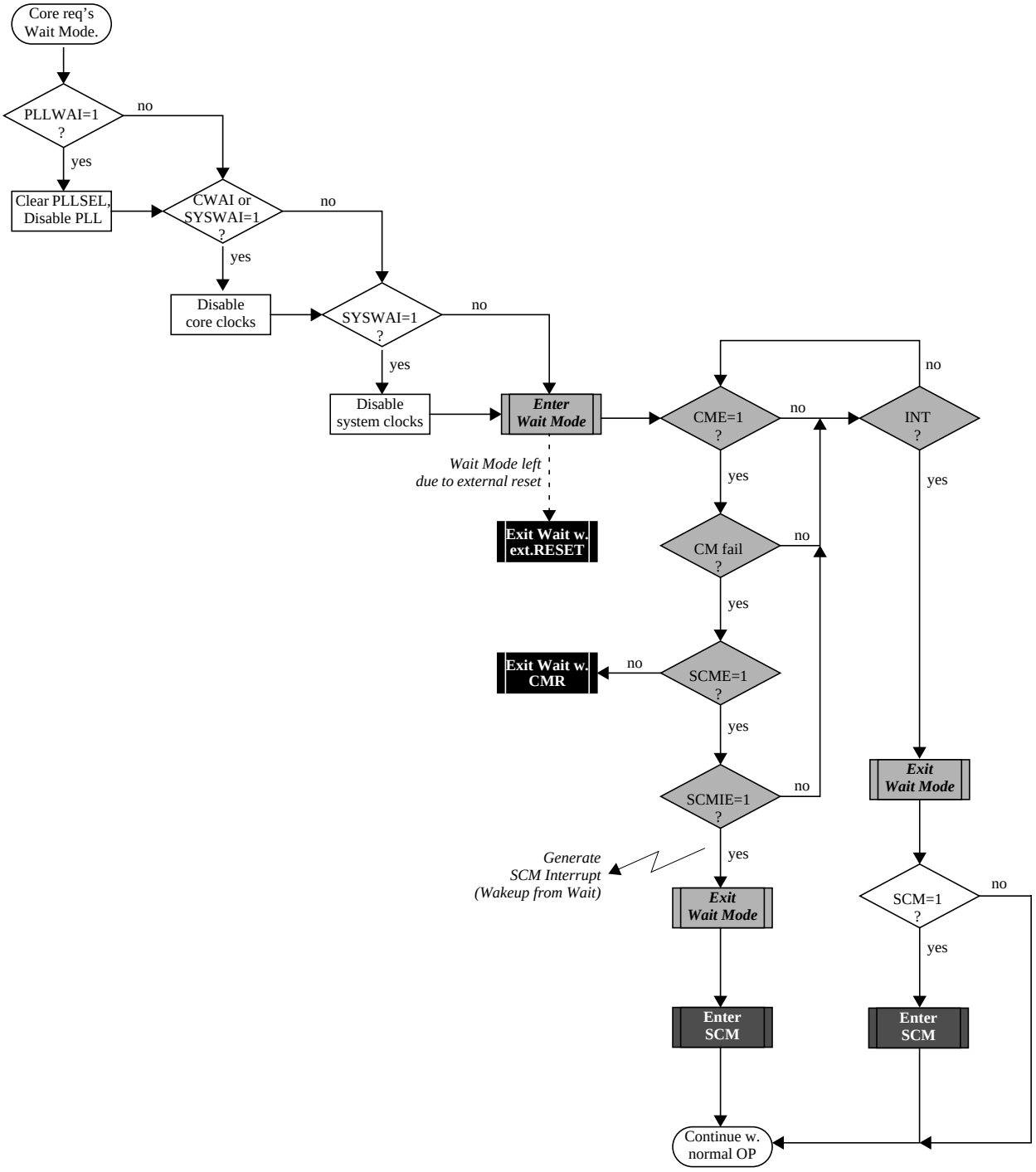


Figure 53 Wait Mode Entry/Exit Sequence

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

There are five different scenarios for the CRG to restart the MCU from Wait Mode:

- External Reset
- Clock Monitor Reset
- COP Reset
- Real Time Interrupt (RTI)
- Wake-up Interrupt[1]

If the MCU gets an external reset during Wait Mode active, the CRG asynchronously restores all configuration bits in the register space to its default settings and starts the reset generator. After completing the reset sequence processing begins by fetching the normal reset vector. Wait Mode is left and the MCU is in Run Mode again.

If the clock monitor is enabled (CME=1) the MCU is able to leave Wait-Mode when loss of oscillator/external clock is detected by a clock monitor fail. If the SCME bit is not asserted the CRG generates a clock monitor fail reset (CMR). The CRG's behavior for CMR is the same compared to external reset, but another reset vector is fetched after completion of the reset sequence. In case the SCME bit is asserted the CRG generates a SCM interrupt if enabled (SCMIE=1). After generating the interrupt the CRG enters Self-Clock Mode and starts the clock quality checker (see Clock Quality Checker in page 301). Then the MCU continues with normal operation. In case the SCM interrupt is blocked by SCMIE=0, the SCMIF flag will be asserted and clock quality checks will be performed but the MCU will not wake-up from Wait-Mode.

If any other interrupt source (e.g. RTI) triggers exit from Wait Mode, the MCU immediately continues with normal operation. If the PLL has been powered-down during Wait-Mode the PLLSEL bit is cleared and the MCU runs on OSCCLK after leaving Wait-Mode. The software must manually set the PLLSEL bit again, in order to switch system and core clocks to the PLLCLK.

1. Interrupts generated by other modules of the MCU (e.g. SCI, ATD, SPI, etc.)

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

Freescale Semiconductor, Inc.

If Wait Mode is entered from Self-Clock Mode the CRG will continue to check the clock quality until clock check is successful. The PLL and voltage regulator (VREG) will remain enabled.

Table 54 summarizes the outcome of a clock loss while in Wait Mode.

Table 54 Outcome of Clock Loss in Wait Mode

CME	SCME	SCMIE	CRG Actions
0	X	X	Clock failure --> No action, clock loss not detected.
1	0	X	Clock failure --> CRG performs Clock Monitor Reset immediately

Freescale Semiconductor, Inc.

Table 54 Outcome of Clock Loss in Wait Mode (Continued)

CME	SCME	SCMIE	CRG Actions
1	1	0	Clock failure --> Scenario 1: OSCCLK **recovers** prior to exiting Wait Mode. - MCU remains in Wait Mode, - VREG enabled, - PLL enabled, - SCM activated, - Start Clock Quality Check, - Set SCMIF interrupt flag. *Some time later OSCCLK recovers.* - CM no longer indicates a failure, - 4096 OSCCLK cycles later Clock Quality Check indicates clock o.k., - SCM deactivated, - PLL disabled depending on PLLWAI, - VREG remains enabled *(never gets disabled in Wait Mode)*. - MCU remains in Wait Mode. *Some time later either a wakeup interrupt occurs (no SCM interrupt)* - Exit Wait Mode using OSCCLK as system clock (SYSCLK), - Continue normal operation. *or an External Reset is applied.* - Exit Wait Mode using OSCCLK as system clock, - Start reset sequence. Scenario 2: OSCCLK **does not recover** prior to exiting Wait Mode. - MCU remains in Wait Mode, - VREG enabled, - PLL enabled, - SCM activated, - Start Clock Quality Check, - Set SCMIF interrupt flag, - Keep performing Clock Quality Checks (could continue infinitely) while in Wait Mode. *Some time later either a wakeup interrupt occurs (no SCM interrupt)* - Exit Wait Mode in SCM using PLL clock (f_{SCM}) as system clock, - Continue to perform additional Clock Quality Checks until OSCCLK is o.k. again. *or an External RESET is applied.* - Exit Wait Mode in SCM using PLL clock (f_{SCM}) as system clock, - Start reset sequence, - Continue to perform additional Clock Quality Checks until OSCCLK is o.k.again.

Freescale Semiconductor, Inc.

Table 54 Outcome of Clock Loss in Wait Mode (Continued)

CME	SCME	SCMIE	CRG Actions
1	1	1	Clock failure --> - VREG enabled, - PLL enabled, - SCM activated, - Start Clock Quality Check, - SCMIF set. *SCMIF generates Self Clock Mode wakeup interrupt.* - Exit Wait Mode in SCM using PLL clock (f_{SCM}) as system clock, - Continue to perform a additional Clock Quality Checks until OSCCLK is o.k. again.

CPU Stop Mode

All clocks are stopped in STOP mode, dependent of the setting of the PCE, PRE and PSTP bit. The oscillator is disabled in STOP mode unless the PSTP bit is set. All counters and dividers remain frozen but do not initialize. If the PRE or PCE bits are set, the RTI or COP continues to run in Pseudo-Stop Mode. In addition to disabling system and core clocks the CRG requests other functional units of the MCU (e.g. voltage-regulator) to enter their individual powersaving modes (if available). This is the main difference between Pseudo-Stop Mode and Wait Mode.

After executing the STOP instruction the core requests the CRG to switch the MCU into Stop Mode. If the PLLSEL bit is still set when entering Stop-Mode, the CRG will switch the system and core clocks to OSCCLK by clearing the PLLSEL bit. Then the CRG disables the PLL, disables the core clock and finally disables the remaining system clocks. As soon as all clocks are switched off Stop-Mode is active.

If Pseudo-Stop Mode (PSTP=1) is entered from Self-Clock Mode the CRG will continue to check the clock quality until clock check is successful. The PLL and the voltage regulator (VREG) will remain enabled. If Full-Stop Mode (PSTP=0) is entered from Self-Clock Mode an ongoing clock quality check will be stopped. A complete timeout window check will be started when Stop Mode is left again.

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

Freescale Semiconductor, Inc.

Wake-up from Stop-Mode also depends on the setting of the PSTP bit.

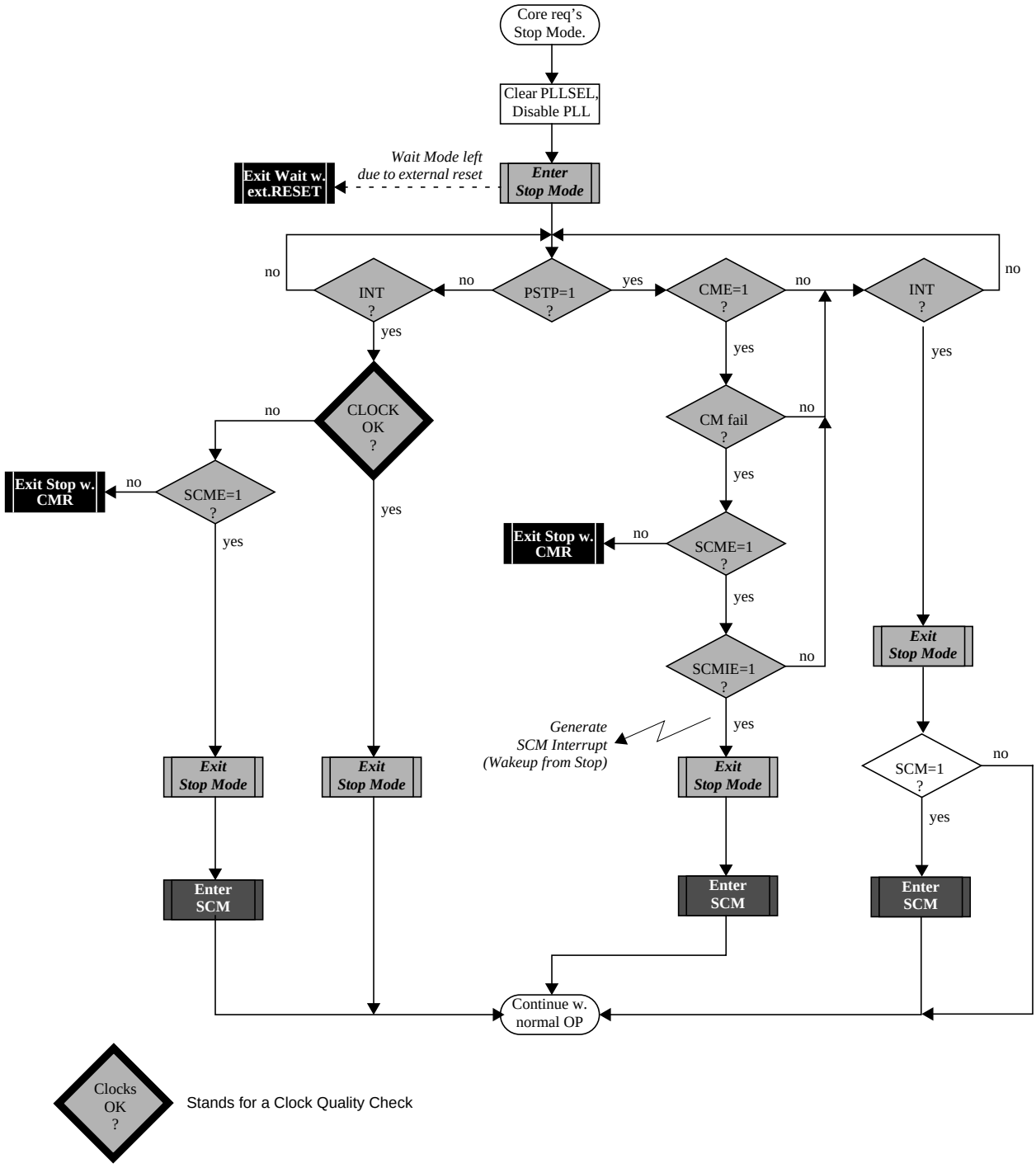


Figure 54 Stop Mode Entry/Exit Sequence

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

Freescale Semiconductor, Inc.

Wake-up from Pseudo-Stop (PSTP=1)

Wake-up from Pseudo-Stop is the same as wake-up from Wait-Mode. There are also five different scenarios for the CRG to restart the MCU from Pseudo-Stop Mode:

- External Reset
- Clock Monitor Fail
- Wake-up Interrupt

If the MCU gets an external reset during Pseudo-Stop Mode active, the CRG asynchronously restores all configuration bits in the register space to its default settings and starts the reset generator. After completing the reset sequence processing begins by fetching the normal reset vector. Pseudo-Stop Mode is left and the MCU is in Run Mode again.

If the clock monitor is enabled (CME=1) the MCU is able to leave Pseudo-Stop Mode when loss of oscillator/external clock is detected by a clock monitor fail. If the SCME bit is not asserted the CRG generates a clock monitor fail reset (CMR). The CRG's behavior for CMR is the same compared to external reset, but another reset vector is fetched after completion of the reset sequence. If the SCME bit is asserted the CRG generates a SCM interrupt if enabled (SCMIE=1). After generating the interrupt the CRG enters Self-Clock Mode and starts the clock quality checker (see Clock Quality Checker in page 301). Then the MCU continues with normal operation. If the SCM interrupt is blocked by SCMIE=0, the SCMIF flag will be asserted but the CRG will not wake-up from Pseudo-Stop Mode.

If any other interrupt source (e.g. RTI) triggers exit from Pseudo-Stop Mode, the MCU immediately continues with normal operation. Because the PLL has been powered-down during Stop-Mode the PLLSEL bit is cleared and the MCU runs on OSCCLK after leaving Stop-Mode. The software must set the PLLSEL bit again, in order to switch system and core clocks to the PLLCLK.

Table 55 summarizes the outcome of a clock loss while in Pseudo-Stop Mode.

Freescale Semiconductor, Inc.

Table 55 Outcome of Clock Loss in Pseudo-Stop Mode

CME	SCME	SCMIE	CRG Actions
0	X	X	Clock failure --> No action, clock loss not detected.
1	0	X	Clock failure --> CRG performs Clock Monitor Reset immediately

Freescale Semiconductor, Inc.

Table 55 Outcome of Clock Loss in Pseudo-Stop Mode (Continued)

CME	SCME	SCMIE	CRG Actions
1	1	0	Clock Monitor failure --> Scenario 1: OSCCLK **recovers** prior to exiting Pseudo-Stop Mode. - MCU remains in Pseudo-Stop Mode, - VREG enabled, - PLL enabled, - SCM activated, - Start Clock Quality Check, - Set SCMIF interrupt flag. *Some time later OSCCLK recovers.* - CM no longer indicates a failure, - 4096 OSCCLK cycles later Clock Quality Check indicates clock o.k., - SCM deactivated, - PLL disabled, - VREG disabled. - MCU remains in Pseudo-Stop Mode. S*ome time later either a wakeup interrupt occurs (no SCM interrupt)* - Exit Pseudo-Stop Mode using OSCCLK as system clock (SYSCLK), - Continue normal operation. *or an External Reset is applied.* - Exit Pseudo-Stop Mode using OSCCLK as system clock, - Start reset sequence. Scenario 2: OSCCLK **does not recover** prior to exiting Pseudo-Stop Mode. - MCU remains in Pseudo-Stop Mode, - VREG enabled, - PLL enabled, - SCM activated, - Start Clock Quality Check, - Set SCMIF interrupt flag, - Keep performing Clock Quality Checks (could continue infinitely) while in Pseudo-Stop Mode. S*ome time later either a wakeup interrupt occurs (no SCM interrupt)* - Exit Pseudo-Stop Mode in SCM using PLL clock (f_{SCM}) as system clock - Continue to perform additional Clock Quality Checks until OSCCLK is o.k. again. *or an External RESET is applied.* - Exit Pseudo-Stop Mode in SCM using PLL clock (f_{SCM}) as system clock - Start reset sequence, - Continue to perform additional Clock Quality Checks until OSCCLK is o.k.again.

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

Freescale Semiconductor, Inc.

Table 55 Outcome of Clock Loss in Pseudo-Stop Mode (Continued)

CME	SCME	SCMIE	CRG Actions
1	1	1	Clock failure --> - VREG enabled, - PLL enabled, - SCM activated, - Start Clock Quality Check, - SCMIF set. *SCMIF generates Self Clock Mode wakeup interrupt.* - Exit Pseudo-Stop Mode in SCM using PLL clock (f_{SCM}) as system clock, - Continue to perform a additional Clock Quality Checks until OSCCLK is o.k. again.

Wake-up from Full Stop (PSTP=0)

The MCU requires an external interrupt or an external reset in order to wake-up from Stop-Mode.

If the MCU gets an external reset during Full Stop Mode active, the CRG asynchronously restores all configuration bits in the register space to its default settings and will perform a maximum of 50 clock *check windows* (see Clock Quality Checker in page 301). After completing the clock quality check the CRG starts the reset generator. After completing the reset sequence processing begins by fetching the normal reset vector. Full Stop-Mode is left and the MCU is in Run Mode again.

If the MCU is woken-up by an interrupt, the CRG will also perform a maximum of 50 clock *check windows* (see Clock Quality Checker in page 301). If the clock quality check is successful, the CRG will release all system and core clocks and will continue with normal operation. If all clock checks within the Timeout-Window are failing, the CRG will switch to Self-Clock Mode or generate a clock monitor reset (CMR) depending on the setting of the SCME bit in the PLLCTL register (page 286). If the SCME bit is cleared, CRG generates a Clock Monitor Reset; otherwise, it enters Self Clock Mode.

Because the PLL has been powered-down during Stop-Mode the PLLSEL bit is cleared and the MCU runs on OSCCLK after leaving Stop-Mode. The software must manually set the PLLSEL bit again, in order to switch system and core clocks to the PLLCLK.

Freescale Semiconductor, Inc.

NOTE: *In Full Stop Mode the clock monitor is disabled and any loss of clock will not be detected.*

Freescale Semiconductor, Inc.

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

Reset Description

General

This section describes how to reset the CRG and how the CRG itself controls the reset of the MCU. It explains all special reset requirements. Since the reset generator for the MCU is part of the CRG this section also describes all automatic actions that occur during or as a result of individual reset conditions. The reset values of registers and signals are provided in Register Descriptions in page 280. All reset sources are listed in Table 56. Refer to MCU specification for related vector addresses and priorities.

Table 56 Reset Summary

Vector Address	Reset Source	Local Enable
$FFFE, $FFFF	Power-on Reset	None
	LVD Reset	LVDCR (LVDE=1 and LVDRE=1)
	External Reset	None
$FFFC, $FFFD	Clock Monitor Reset	PLLCTL (CME=1, SCME=0)
$FFFA, $FFFB	COP Watchdog Reset	COPCTL (CR[2:0] nonzero)

Description of Reset Operation

The reset sequence is initiated by any of the following events:

- Low level is detected at the $\overline{\text{RESET}}$ pin (External Reset).
- Power-on is detected. (Power-on Reset - POR)
- Low voltage condition is detected. (Low Voltage Detection Reset LVDR)
- COP watchdog times out. (COP reset - COPR)
- Clock monitor failure is detected and Self-Clock Mode was disabled (SCME=0). (Clock Monitor Reset - CMR)

Upon detection of any reset event, an internal circuit drives the $\overline{\text{RESET}}$ pin low for 128 SYSCLK cycles (see Figure 55). Since entry into reset is asynchronous it does not require a running SYSCLK. However, the internal reset circuit of the CRG cannot sequence out of current reset condition without a running SYSCLK. The number of 128 SYSCLK cycles might be increased by n=3 to 6 additional SYSCLK cycles

Freescale Semiconductor, Inc.

depending on the internal synchronization latency. After 128+n SYSCLK cycles the $\overline{\text{RESET}}$ pin is released. The reset generator circuit of the CRG waits for additional 64 SYSCLK cycles and then samples the RESET pin to determine the originating source. Table 57 shows which vector will be fetched.

Table 57 Reset Vector Selection

Sampled $\overline{\text{RESET}}$ pin (64 SYSCLK cycles after release)	Clock Monitor Reset pending	COP Reset pending	Vector fetch
1	0	0	POR / LVDR / External Reset
1	1	X	Clock Monitor Reset
1	0	1	COP Reset
0	X	X	POR / LVDR / External Reset with rise of $\overline{\text{RESET}}$ pin

NOTE: *External circuitry connected to the $\overline{\text{RESET}}$ pin should not include a large capacitance that would interfere with the ability of this signal to rise to a valid logic one within 64 SYSCLK cycles after the low drive is released.*

The internal reset of the MCU remains asserted while the reset generator completes the 192 SYSCLK long reset sequence. The reset generator circuitry always makes sure the internal reset is negated synchronously after completion of the 192 SYSCLK cycles. In case the $\overline{\text{RESET}}$ pin is externally driven low for more than these 192 SYSCLK cycles (External Reset), the internal reset remains asserted too.

Freescale Semiconductor, Inc.

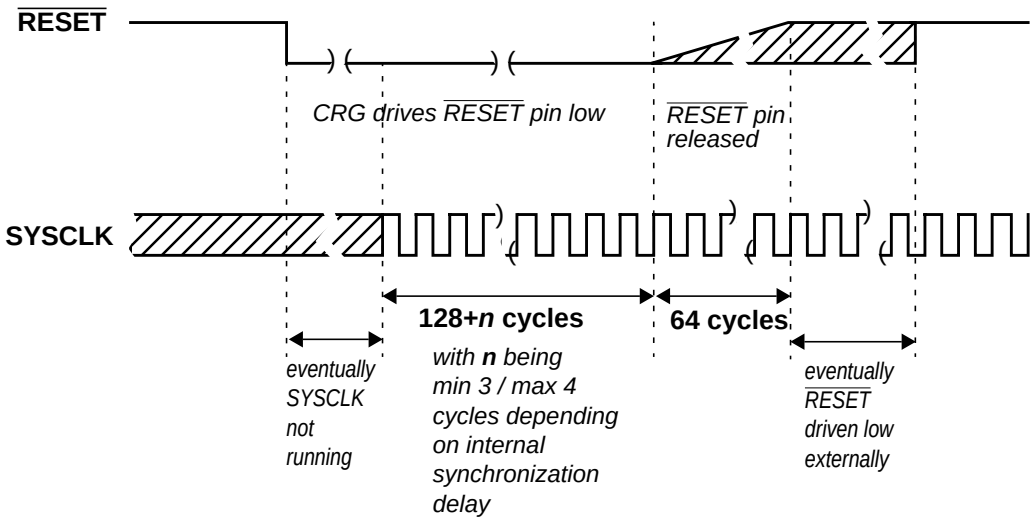


Figure 55 RESET Timing

Clock Monitor Reset

The CRG generates a Clock Monitor Reset in case all of the following conditions are true:

- Clock monitor is enabled (CME=1)
- Loss of clock is detected
- Self-Clock Mode is disabled (SCME=0).

The reset event asynchronously forces the configuration registers to their default settings (see Register Descriptions in page 280). In detail the CME and the SCME are reset to logical “1” (which doesn’t change the state of the CME bit, because it has already been set). As a consequence the CRG immediately enters Self Clock Mode and starts its internal reset sequence. The clock quality check starts in parallel. As soon as the clock quality check indicates a valid OSCCLK, CRG switches to OSCCLK and leaves Self Clock Mode. Since the clock quality checker is running in parallel to the reset generator, the CRG may leave Self Clock Mode while still completing the internal reset sequence, e.g. when a high frequency OSCCLK is provided. When the reset sequence is finished the CRG checks the internally latched state of the clock monitor fail circuit. If a clock monitor fail is indicated processing begins by fetching the Clock Monitor Reset vector.

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

Freescale Semiconductor, Inc.

Computer Operating Properly Watchdog (COP) Reset

When COP is enabled, the CRG expects sequential write of $55 and $AA (in this order) to the ARMCOP register during the selected time-out period. Once this is done, the COP time-out period restarts. If the program fails to do this the CRG will generate a reset. Also, if any value other than $55 or $AA is written, the CRG immediately generates a reset. In case window COP operation is enabled writes ($55 or $AA) to the ARMCOP register must occur in the last 25% of the selected time-out period. A premature write the CRG will immediately generate a reset.

As soon as the reset sequence is completed the reset generator checks the reset condition. If no clock monitor failure is indicated and the latched state of the COP timeout is true, processing begins by fetching the COP vector.

Low Voltage Detection Reset

The CRG generates a Low Voltage Detection Reset in case all of the following conditions are true:

- VDDR voltage falls to V_{LVR} level and remains at or below that level for 17 or more consecutive bus clock cycles.
- The LVD reset enable bit (LVDRE) in the LVDCR register is set.

As showed in Figure 50, after a LVD reset occurs, the CRG will perform a maximum of 50 check windows before entering Self Clock Mode and executing the reset sequence. If OSCCLK is considered valid before 50 check windows are complete, the clock quality check is successfully terminated and reset sequence is executed. More details about the LVD module can be found in the section in page 185.

Power-On Reset

The on-chip voltage regulator detects when VDD to the MCU has reached a certain level and asserts power on reset. As soon as a power on reset is triggered the CRG performs a quality check on the incoming clock signal. As soon as clock quality check indicates a valid Oscillator Clock signal the reset sequence starts using the Oscillator clock. If after 50 check windows the clock quality check indicated a non-valid Oscillator Clock the reset sequence starts using Self-Clock Mode. See Figure 50.

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

Freescale Semiconductor, Inc.

Figure 56 and Figure 57 show the power-up sequence for cases when the $\overline{\text{RESET}}$ pin is tied to VDD and when the $\overline{\text{RESET}}$ pin is held low. Note that the reset sequence only starts after a running SYSCLK is detected.

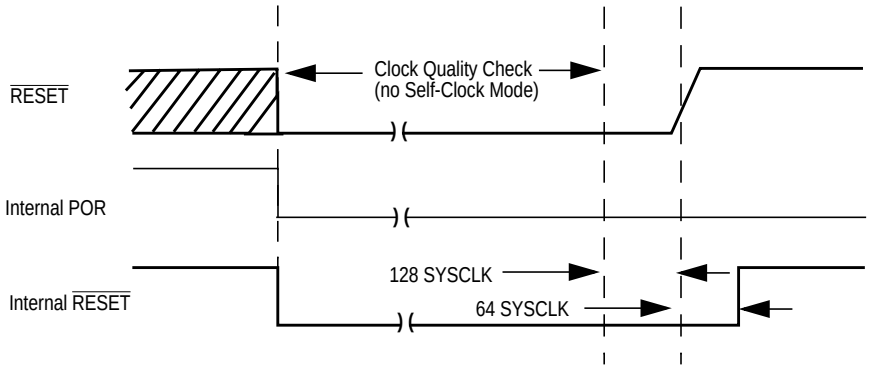


Figure 56 $\overline{\text{RESET}}$ pin tied to VDD (by a pull-up resistor)

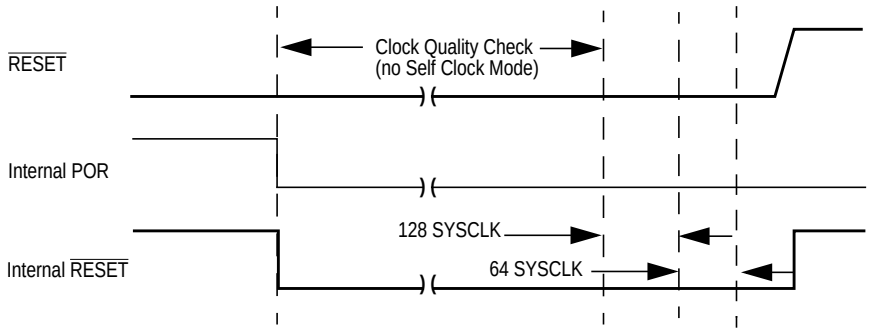


Figure 57 $\overline{\text{RESET}}$ pin held low externally

NOTE: *Proper function of the LVD module requires the voltage regulator to be enable during operation (VREGEN pin tied high).*

NOTE: *POR is only rearmed if the VDD voltage falls below the POR rearm level (V_{POR} [1]).*

There are four possible combinations of PORLVDRF (CRGFLG register) and LVDF (LVDSR register) flags that can be obtained during operation.

1. See Table 116 in page 578 for the actual values of these parameters.

Freescale Semiconductor, Inc.

These combinations are shown in Table 58 with the correspondent events that caused them to be set. PORLVDRF is set when a power-on or a low-voltage reset occurs, while LVDF is set when a low-voltage condition is detected.

Table 58 Relation between PORLVDRF and LVDF

PORLVDRF	LVDF	Event
0	0	Neither POR nor LVDR occurred. No low-voltage condition detected(1)
0	1	Neither POR nor LVDR occurred. Low-voltage condition detected but LVD reset not enabled (LVDRE=0)(1)
1	0	POR occurred
1	1	LVDR occurred

1. Considering that PORLVDRF and LVDF flags were cleared by the program after POR or LVDR.

Freescale Semiconductor, Inc.

Interrupts

General

This section describes all interrupts originated by the CRG.

The interrupts/reset vectors requested by the CRG are listed in Table 59.

Table 59 CRG Interrupt Vectors

Vector Address	Interrupt Source	CCR Mask	Local Enable
$FFF0, $FFF1	Real Time interrupt	I bit	CRGINT (RTIE)
$FFC6, $FFC7	LOCK interrupt	I bit	CRGINT (LOCKIE)
$FFC4, $FFC5	SCM interrupt	I bit	CRGINT (SCMIE)

Description of Interrupt Operation

Real Time Interrupt

The CRG generates a real time interrupt when the selected interrupt time period elapses. RTI interrupts are locally disabled by setting the RTIE bit to zero. The real time interrupt flag (RTIF) is set to1 when a timeout occurs, and is cleared to 0 by writing a 1 to the RTIF bit.

The RTI continues to run during Pseudo Stop Mode if the PRE bit is set to 1. This feature can be used for periodic wakeup from Pseudo Stop if the RTI interrupt is enabled.

PLL Lock Interrupt

The CRG generates a PLL Lock interrupt when the LOCK condition of the PLL has changed, either from a locked state to an unlocked state or vice versa. Lock interrupts are locally disabled by setting the LOCKIE bit to zero. The PLL Lock interrupt flag (LOCKIF) is set to1 when the LOCK condition has changed, and is cleared to 0 by writing a 1 to the LOCKIF bit.

Freescale Semiconductor, Inc.

Self Clock Mode Interrupt

The CRG generates a Self Clock Mode interrupt when the SCM condition of the system has changed, either entered or exited Self Clock Mode. SCM conditions can only change if the Self Clock Mode enable bit (SCME) is set to 1. SCM conditions are caused by a failing clock quality check after Power-on-Reset (POR) or recovery from Full Stop Mode (PSTP=0) or Clock Monitor failure. For details on the clock quality check refer to Clock Quality Checker in page 301. If the clock monitor is enabled (CME=1) a loss of external clock will also cause a SCM condition (SCME=1).

SCM interrupts are locally disabled by setting the SCMIE bit to zero. The SCM interrupt flag (SCMIF) is set to1 when the SCM condition has changed, and is cleared to 0 by writing a 1 to the SCMIF bit.

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

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

Pulse Width Modulator (PWM8B8C)

Contents

Overview 327
Features 328
Modes of Operation 328
Block Diagram 330
External Pin Descriptions 331
Register Map 332
Register Descriptions 334
Functional Description 353
Low Power Options 368
Reset Initialization 368
Interrupts 369

Overview

The PWM_8B8C definition is based on the MC68HC11KD2 and the HC12 PWM definitions. This PWM_8B8C contains the basic features from the HC11 with some of the enhancements incorporated on the HC12. In addition, the module is now expanded to eight channels with independent control of left and center aligned outputs on each channel. In all basic functionality, it will behave similarly to the HC11 PWM with the additional HC12 PWM features of center aligned output mode and four available clock sources.

The PWM_8B8C module contains eight channels. Each of these channels has a programmable period and duty cycle as well as a dedicated counter. A flexible clock select scheme allows a total of four different clock sources to be used with the counters. Each of the modulators can create independent continuous waveforms with

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

Freescale Semiconductor, Inc.

software-selectable duty rates from 0% to 100%. The PWM outputs can be programmed as left aligned outputs or center aligned outputs.

Features

- Eight independent PWM channels with programmable period and duty cycle.
- Dedicated counter for each PWM channel.
- Programmable PWM enable/disable for each channel.
- Software selection of PWM duty pulse polarity for each channel.
- Period and duty cycle are double buffered. Change takes effect when the end of the effective period is reached (PWM counter reaches zero) or when the channel is disabled.
- Programmable center or left aligned outputs on individual channels.
- Eight 8-bit channel or four 16-bit channel PWM resolution.
- Four clock sources (A, B, SA and SB) provide for a wide range of frequencies.
- Programmable Clock Select Logic.
- Emergency shutdown.

Modes of Operation

Normal Modes The PWM module behaves as described within this specification in all normal modes.

Special Modes The PWM module behaves as described within this specification in all special modes.

Freescale Semiconductor, Inc.

WARNING: ***While in special modes, do not access registers $0006, $0007, $000A and $000B. Writing to any of these registers can alter the PWM functionality.***

Freeze Mode

The PWM module enters freeze mode when background debug mode (BDM) is active.

Emulation Modes

In Freeze Mode, there is an option to disable the input clock to the prescaler by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode the input clock to the prescaler is disabled. This feature is useful during emulation as it allows the PWM function to be suspended. In this way, the counters of the PWM can be stopped while in freeze mode so that once normal program flow is continued, the counters are re-enabled to simulate real-time operations. Since the registers can still be accessed in this mode, to re-enable the prescaler clock, either disable the PFRZ bit or exit freeze mode.

Freescale Semiconductor, Inc.

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

Block Diagram

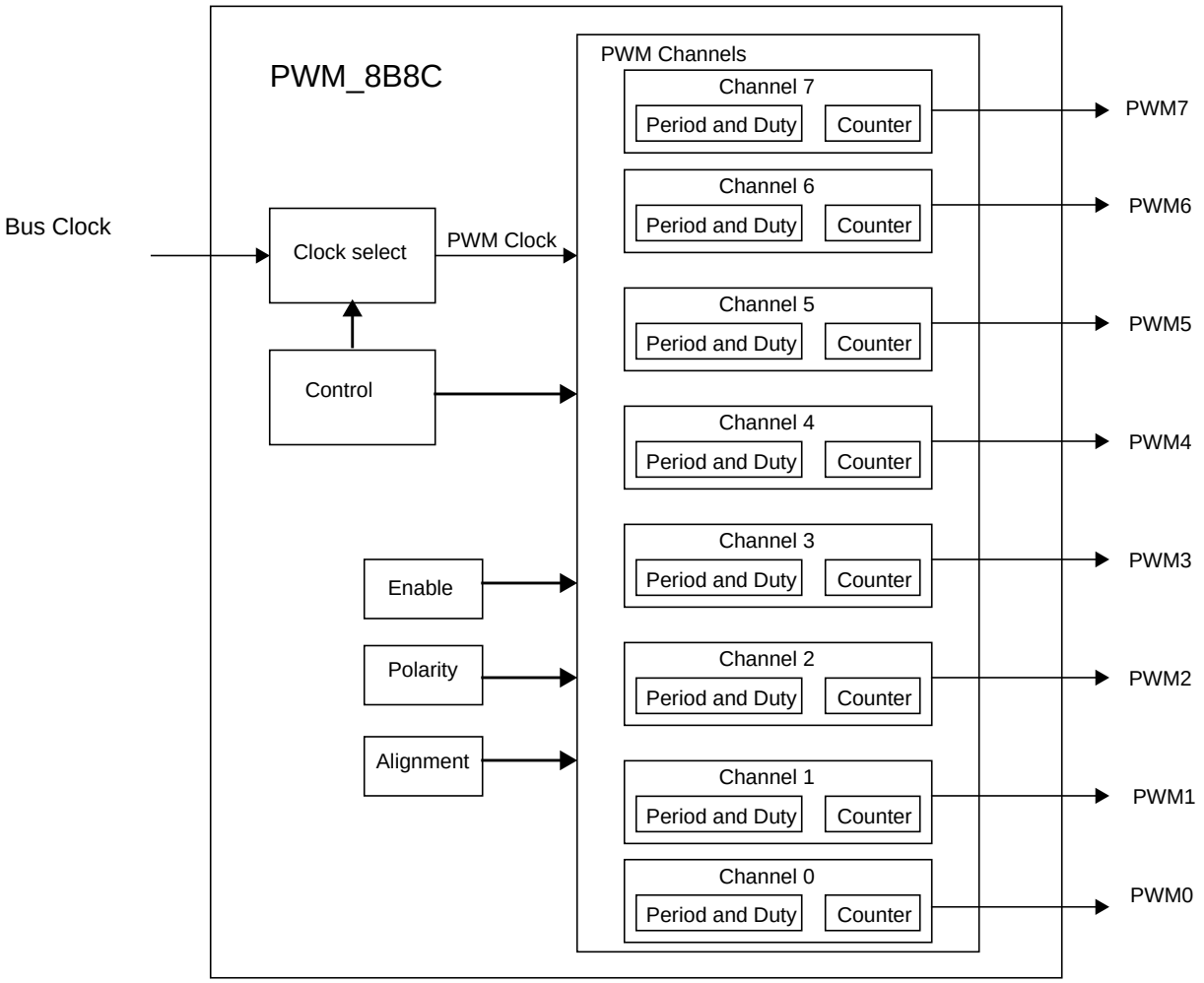


Figure 58 PWM8B8C Block Diagram

Freescale Semiconductor, Inc.

External Pin Descriptions

The PWM8B8C module has a total of 8 external pins.

PWM7 (PP7) PWM8B8C Channel 7 - This pin serves as waveform output of PWM channel 7 and as an input for the emergency shutdown feature.

PWM6 (PP6) PWM8B8C Channel 6 - This pin serves as waveform output of PWM channel 6.

PWM5 (PP5) PWM8B8C Channel 5 - This pin serves as waveform output of PWM channel 5.

PWM4 (PP4) PWM8B8C Channel 4 - This pin serves as waveform output of PWM channel 4.

PWM3 (PP3) PWM8B8C Channel 3 - This pin serves as waveform output of PWM channel 3.

PWM2 (PP2) PWM8B8C Channel 2 - This pin serves as waveform output of PWM channel 2.

PWM1 (PP1) PWM8B8C Channel 1 - This pin serves as waveform output of PWM channel 1.

PWM0 (PP0) PWM8B8C Channel 0 - This pin serves as waveform output of PWM channel 0.

Freescale Semiconductor, Inc.

Register Map

This section describes the content of the registers in the PWM. The figure below shows the registers associated with the PWM and their relative offset from the base address. The register detail description follows the order they appear in the register map.

Reserved bits within a register will always read as 0 and the write will be unimplemented. Unimplemented functions are indicated by shaded bits.

Register Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Blt 1	Bit 0	Address Offset
PWME	Read:	PWME7	PWME6	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0	$00A0
	Write:									
PWMPOL	Read:	PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0	$00A1
	Write:									
PWMCLK	Read:	PCLK7	PCLKL6	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0	$00A2
	Write:									
PWMPRCLK	Read:	0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0	$00A3
	Write:									
PWMCAE	Read:	CAE7	CAE6	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0	$00A4
	Write:									
PWMCTL	Read:	CON67	CON45	CON23	CON01	PSWAI	PFRZ	0	0	$00A5
	Write:									
Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								$00A6
	Write:									
Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								$00A7
	Write:									
PWMSCLA	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00A8
	Write:									
PWMSCLB	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00A9
	Write:									
Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								$00AA
	Write:									
Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								$00AB
	Write:									
PWMCNT0	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00AC
	Write:	0	0	0	0	0	0	0	0	
PWMCNT1	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00AD
	Write:	0	0	0	0	0	0	0	0	

= Unimplemented

Figure 59 PWM Register Map

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

Freescale Semiconductor, Inc.

Register Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	BIt 1	Bit 0	Address Offset
PWMCNT2	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00AE
	Write:	0	0	0	0	0	0	0	0	
PWMCNT3	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00AF
	Write:	0	0	0	0	0	0	0	0	
PWMCNT4	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00B0
	Write:	0	0	0	0	0	0	0	0	
PWMCNT5	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00B1
	Write:	0	0	0	0	0	0	0	0	
PWMCNT6	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00B2
	Write:	0	0	0	0	0	0	0	0	
PWMCNT7	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00B3
	Write:	0	0	0	0	0	0	0	0	
PWMPER0	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00B4
PWMPER1	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00B5
PWMPER2	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00B6
PWMPER3	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00B7
PWMPER4	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00B8
PWMPER5	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00B9
PWMPER6	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00BA
PWMPER7	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00BB
PWMDTY0	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00BC
PWMDTY1	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00BD
PWMDTY2	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00BE
PWMDTY3	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00BF
PWMDTY4	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00C0
PWMDTY5	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$00C1

= Unimplemented

Figure 59 PWM Register Map (Continued)

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

Freescale Semiconductor, Inc.

Register Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Blt 1	Bit 0	Address Offset
PWMDTY6	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00C2
	Write:									
PWMDTY7	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00C3
	Write:									
PWMSDN	Read:	PWMIF	PWMIE	0	PWMLVL	0	PWM7IN	PWM7INL	PWM7ENA	$00C4
	Write:			PWMRSTRT						
Reserved	Read:	0	0	0	0	0	0	0	0	$00C5
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$00C6
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$00C7
	Write:									

= Unimplemented

Figure 59 PWM Register Map (Continued)

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Register Descriptions

This section describes in detail all the registers and register bits in the PWM module.

NOTE: *All bits of all registers in this module are completely synchronous to internal clocks during a register read.*

Freescale Semiconductor, Inc.

PWM Enable Register (PWME)

Address Offset: $00A0

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PWME7	PWME6	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0
Reset:	0	0	0	0	0	0	0	0

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx=1), the associated PWM output is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle due to the synchronization of PWMEx and the clock source.

The first PWM cycle after enabling the channel can be irregular.

An exception to this is when channels are concatenated. Once concatenated mode is enabled (CONxx bits set in PWMCTL register) then enabling/disabling the corresponding 16-bit PWM channel is controlled by the low order PWMEx bit. In this case, the high order bytes PWMEx bits have no effect and their corresponding PWM output lines are disabled.

While in run mode, if all eight PWM channels are disabled PWMEx=0), the prescaler counter shuts off for power savings.

Read: anytime
Write: anytime

PWME7 — Pulse Width Channel 7 Enable

1 = Pulse Width channel 7 is enabled. The pulse modulated signal becomes available at PWM output bit7 when its clock source begins its next cycle.

0 = Pulse Width channel 7 is disabled.

PWME6 — Pulse Width Channel 6 Enable

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

Freescale Semiconductor, Inc.

1 = Pulse Width channel 6 is enabled. The pulse modulated signal becomes available at port PWM output bit6 when its clock source begins its next cycle. If CON67=1, then bit has no effect and PWM output bit6 is disabled.
0 = Pulse Width channel 6 is disabled.

PWME5 — Pulse Width Channel 5 Enable
1 = Pulse Width channel 5 is enabled. The pulse modulated signal becomes available at PWM output bit 5 when its clock source begins its next cycle.
0 = Pulse Width channel 5 is disabled.

PWME4 — Pulse Width Channel 4 Enable
1 = Pulse Width channel 4 is enabled. The pulse modulated signal becomes available at PWM output bit 4 when its clock source begins its next cycle. If CON45=1, then bit has no effect and PWM output bit4 is disabled.
0 = Pulse Width channel 4 is disabled.

PWME3 — Pulse Width Channel 3 Enable
1 = Pulse Width channel 3 is enabled. The pulse modulated signal becomes available at PWM output bit 3 when its clock source begins its next cycle.
0 = Pulse Width channel 3 is disabled.

PWME2 — Pulse Width Channel 2 Enable
1 = Pulse Width channel 2 is enabled. The pulse modulated signal becomes available at PWM output bit 2 when its clock source begins its next cycle. If CON23=1, then bit has no effect and PWM output bit2 is disabled.
0 = Pulse Width channel 2 is disabled.

PWME1 — Pulse Width Channel 1 Enable
1 = Pulse Width channel 1 is enabled. The pulse modulated signal becomes available at PWM output bit 1 when its clock source begins its next cycle.
0 = Pulse Width channel 1 is disabled.

PWME0 — Pulse Width Channel 0 Enable

Freescale Semiconductor, Inc.

1 = Pulse Width channel 0 is enabled. The pulse modulated signal becomes available at PWM, o/p bit 0 when its clock source begins its next cycle. If CON01=1, then bit has no effect and PWM output bit0 is disabled.
0 = Pulse Width channel 0 is disabled.

PWM Polarity Register (PWMPOL)

Address Offset: $00A1

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0
Reset:	0	0	0	0	0	0	0	0

The starting polarity of each PWM channel waveform is determined by the associated PPOLx bit in the PWMPOL register. If the polarity bit is one, the PWM channel output is high at the beginning of the cycle and then goes low when the duty count is reached. Conversely, if the polarity bit is zero, the output starts low and then goes high when the duty count is reached.

Read: anytime
Write: anytime

CAUTION: *PPOLx register bits can be written anytime. If the polarity is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition.*

PPOL7 — Pulse Width Channel 7 Polarity
1 = PWM channel 7 output is high at the beginning of the period, then goes low when the duty count is reached.
0 = PWM channel 7 output is low at the beginning of the period, then goes high when the duty count is reached.

PPOL6 — Pulse Width Channel 6 Polarity

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

Freescale Semiconductor, Inc.

1 = PWM channel 6 output is high at the beginning of the period, then goes low when the duty count is reached.
0 = PWM channel 6 output is low at the beginning of the period, then goes high when the duty count is reached.

PPOL5 — Pulse Width Channel 5 Polarity
1 = PWM channel 5 output is high at the beginning of the period, then goes low when the duty count is reached.
0 = PWM channel 5 output is low at the beginning of the period, then goes high when the duty count is reached.

PPOL4 — Pulse Width Channel 4 Polarity
1 = PWM channel 4 output is high at the beginning of the period, then goes low when the duty count is reached.
0 = PWM channel 4 output is low at the beginning of the period, then goes high when the duty count is reached.

PPOL3 — Pulse Width Channel 3 Polarity
1 = PWM channel 3 output is high at the beginning of the period, then goes low when the duty count is reached.
0 = PWM channel 3 output is low at the beginning of the period, then goes high when the duty count is reached.

PPOL2 — Pulse Width Channel 2 Polarity
1 = PWM channel 2 output is high at the beginning of the period, then goes low when the duty count is reached.
0 = PWM channel 2 output is low at the beginning of the period, then goes high when the duty count is reached.

PPOL1 — Pulse Width Channel 1 Polarity
1 = PWM channel 1 output is high at the beginning of the period, then goes low when the duty count is reached.
0 = PWM channel 1 output is low at the beginning of the period, then goes high when the duty count is reached.

PPOL0 — Pulse Width Channel 0 Polarity
1 = PWM channel 0 output is high at the beginning of the period, then goes low when the duty count is reached.
0 = PWM channel 0 output is low at the beginning of the period, then goes high when the duty count is reached.

Freescale Semiconductor, Inc.

PWM Clock Select Register (PWMCLK)

Address Offset: $00A2

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PCLK7	PCLKL6	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0
Reset:	0	0	0	0	0	0	0	0

Each PWM channel has a choice of two clocks to use as the clock source for that channel as described below.

Read: anytime
Write: anytime

CAUTION: *Register bits PCLK0 to PCLK7 can be written anytime. If a clock select is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition.*

PCLK7 — Pulse Width Channel 7 Clock Select
1 = Clock SB is the clock source for PWM channel 7.
0 = Clock B is the clock source for PWM channel 7.

PCLK6 — Pulse Width Channel 6 Clock Select
1 = Clock SB is the clock source for PWM channel 6.
0 = Clock B is the clock source for PWM channel 6.

PCLK5 — Pulse Width Channel 5 Clock Select
1 = Clock SA is the clock source for PWM channel 5.
0 = Clock A is the clock source for PWM channel 5.

PCLK4 — Pulse Width Channel 4 Clock Select
1 = Clock SA is the clock source for PWM channel 4.
0 = Clock A is the clock source for PWM channel 4.

PCLK3 — Pulse Width Channel 3 Clock Select
1 = Clock SB is the clock source for PWM channel 3.
0 = Clock B is the clock source for PWM channel 3.

Freescale Semiconductor, Inc.

PCLK2 — Pulse Width Channel 2 Clock Select

1 = Clock SB is the clock source for PWM channel 2.
0 = Clock B is the clock source for PWM channel 2.

PCLK1 — Pulse Width Channel 1 Clock Select

1 = Clock SA is the clock source for PWM channel 1.
0 = Clock A is the clock source for PWM channel 1.

PCLK0 — Pulse Width Channel 0 Clock Select

1 = Clock SA is the clock source for PWM channel 0.
0 = Clock A is the clock source for PWM channel 0.

PWM Prescale Clock Select Register (PWMPRCLK)

Address Offset: $00A3

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

This register selects the prescale clock source for clocks A and B independently.

Read: anytime
Write: anytime

CAUTION: *PCKB2–0 and PCKA2–0 register bits can be written anytime. If the clock pre-scale is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition.*

PCKB2–PCKB0 — Prescaler Select for Clock B

Clock B is one of two clock sources which can be used for channels 2, 3, 6, or 7. These three bits determine the rate of clock B, as shown in the following table.

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

Freescale Semiconductor, Inc.

Table 60 Clock B Prescaler Selects

PCKB2	PCKB1	PCKB0	Value of Clock B
0	0	0	Bus Clock
0	0	1	Bus Clock / 2
0	1	0	Bus Clock / 4
0	1	1	Bus Clock / 8
1	0	0	Bus Clock / 16
1	0	1	Bus Clock / 32
1	1	0	Bus Clock / 64
1	1	1	Bus Clock / 128

PCKA2–PCKA0 — Prescaler Select for Clock A

Clock A is one of two clock sources which can be used for channels 0, 1, 4, or 5. These three bits determine the rate of clock A, as shown in the following table.

Table 61 Clock A Prescaler Selects

PCKA2	PCKA1	PCKA0	Value of Clock A
0	0	0	Bus Clock
0	0	1	Bus Clock / 2
0	1	0	Bus Clock / 4
0	1	1	Bus Clock / 8
1	0	0	Bus Clock / 16
1	0	1	Bus Clock / 32
1	1	0	Bus Clock / 64
1	1	1	Bus Clock / 128

Freescale Semiconductor, Inc.

PWM Center Align Enable Register (PWMCAE)

Address Offset: $00A4

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	CAE7	CAE6	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0
Reset:	0	0	0	0	0	0	0	0

The PWMCAE register contains eight control bits for the selection of center aligned outputs or left aligned outputs for each PWM channel. If the CAEx bit is set to a one, the corresponding PWM output will be center aligned. If the CAEx bit is cleared, the corresponding PWM output will be left aligned. Reference Left Aligned Outputs and Center Aligned Outputs for a more detailed description of the PWM output modes.

Read: anytime
Write: anytime

CAUTION: *Write these bits only when the corresponding channel is disabled.*

CAE7 — Center Aligned Output Mode on channel 7
1 = Channel 7 operates in Center Aligned Output Mode.
0 = Channel 7 operates in Left Aligned Output Mode.

CAE6 — Center Aligned Output Mode on channel 6
1 = Channel 6 operates in Center Aligned Output Mode.
0 = Channel 6 operates in Left Aligned Output Mode.

CAE5 — Center Aligned Output Mode on channel 5
1 = Channel 5 operates in Center Aligned Output Mode.
0 = Channel 5 operates in Left Aligned Output Mode.

CAE4 — Center Aligned Output Mode on channel 4
1 = Channel 4 operates in Center Aligned Output Mode.
0 = Channel 4 operates in Left Aligned Output Mode.

CAE3 — Center Aligned Output Mode on channel 3
1 = Channel 3 operates in Center Aligned Output Mode.
0 = Channel 3 operates in Left Aligned Output Mode.

Freescale Semiconductor, Inc.

CAE2 — Center Aligned Output Mode on channel 2
1 = Channel 2 operates in Center Aligned Output Mode.
0 = Channel 2 operates in Left Aligned Output Mode.

CAE1 — Center Aligned Output Mode on channel 1
1 = Channel 1 operates in Center Aligned Output Mode.
0 = Channel 1 operates in Left Aligned Output Mode.

CAE0 — Center Aligned Output Mode on channel 0
1 = Channel 0 operates in Center Aligned Output Mode.
0 = Channel 0 operates in Left Aligned Output Mode.

PWM Control Register (PWMCTL)

Address Offset: $00A5

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CON67	CON45	CON23	CON01	PSWAI	PFRZ	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

The PWMCTL register provides for various control of the PWM module.
Read: anytime
Write: anytime

There are three control bits for concatenation, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. When channels 6 and 7 are concatenated, channel 6 registers become the high order bytes of the double byte channel as shown in Figure 66. Similarly, when channels 4 and 5 are concatenated, channel 4 registers become the high order bytes of the double byte channel. When channels 2 and 3 are concatenated, channel 2 registers become the high order bytes of the double byte channel. When channels 0 and 1 are concatenated, channel 0 registers become the high order bytes of the double byte channel.

Reference PWM 16-Bit Functions for a more detailed description of the concatenation PWM Function.

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

Freescale Semiconductor, Inc.

CAUTION: *Change these bits only when both corresponding channels are disabled.*

CON67 — Concatenate channels 6 and 7

1 = Channels 6 and 7 are concatenated to create one 16-bit PWM channel. Channel 6 becomes the high order byte and channel 7 becomes the low order byte. Channel 7 output pin is used as the output for this 16-bit PWM (bit 7 of port PWMP). Channel 7 clock select control-bit determines the clock source, channel 7 polarity bit determines the polarity, channel 7 enable bit enables the output and channel 7 center aligned enable bit determines the output mode.

0 = Channels 6 and 7 are separate 8-bit PWMs.

CON45 — Concatenate channels 4 and 5

1 = Channels 4 and 5 are concatenated to create one 16-bit PWM channel. Channel 4 becomes the high order byte and channel 5 becomes the low order byte. Channel 5 output pin is used as the output for this 16-bit PWM (bit 5 of port PWMP). Channel 5 clock select control-bit determines the clock source, channel 5 polarity bit determines the polarity, channel 5 enable bit enables the output and channel 5 center aligned enable bit determines the output mode.

0 = Channels 4 and 5 are separate 8-bit PWMs.

CON23 — Concatenate channels 2 and 3

1 = Channels 2 and 3 are concatenated to create one 16-bit PWM channel. Channel 2 becomes the high order byte and channel 3 becomes the low order byte. Channel 3 output pin is used as the output for this 16-bit PWM (bit 3 of port PWMP). Channel 3 clock select control-bit determines the clock source, channel 3 polarity bit determines the polarity, channel 3 enable bit enables the output and channel 3 center aligned enable bit determines the output mode.

0 = Channels 2 and 3 are separate 8-bit PWMs.

CON01 — Concatenate channels 0 and 1

Freescale Semiconductor, Inc.

1 = Channels 0 and 1 are concatenated to create one 16-bit PWM channel. Channel 0 becomes the high order byte and channel 1 becomes the low order byte. Channel 1 output pin is used as the output for this 16-bit PWM (bit 1 of port PWMP). Channel 1 clock select control-bit determines the clock source, channel 1 polarity bit determines the polarity, channel 1 enable bit enables the output and channel 1 center aligned enable bit determines the output mode.

0 = Channels 0 and 1 are separate 8-bit PWMs.

PSWAI — PWM Stops in Wait Mode

Enabling this bit allows for lower power consumption in Wait Mode by disabling the input clock to the prescaler.

1 = Stop the input clock to the prescaler whenever the MCU is in Wait Mode.

0 = Allow the clock to the prescaler to continue while in wait mode.

PFRZ — PWM Counters Stop in Freeze Mode

In Freeze Mode, there is an option to disable the input clock to the prescaler by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode the input clock to the prescaler is disabled. This feature is useful during emulation as it allows the PWM function to be suspended. In this way, the counters of the PWM can be stopped while in freeze mode so that once normal program flow is continued, the counters are re-enabled to simulate real-time operations. Since the registers can still be accessed in this mode, to re-enable the prescaler clock, either disable the PFRZ bit or exit freeze mode.

1 = Disable PWM input clock to the prescaler whenever the part is in freeze mode. This is useful for emulation.

0 = Allow PWM to continue while in freeze mode.

NOTE: *The PWM module enters freeze mode when background debug mode (BDM) is active. Refer to the Fast Background Debug Module (FBDM) section about the background debug mode.*

Freescale Semiconductor, Inc.

PWM Scale A Register (PWMSCLA)

Address Offset: $00A8

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	0	0	0	0	0	0	0	0

PWMSCLA is the programmable scale value used in scaling clock A to generate clock SA. Clock SA is generated by taking clock A, dividing it by the value in the PWMSCLA register and dividing that by two.

Clock SA = Clock A / (2 * PWMSCLA)

NOTE: When PWMSCLA = $00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLA).
Read: anytime
Write: anytime (causes the scale counter to load the PWMSCLA value)

PWM Scale B Register (PWMSCLB)

Address Offset: $00A9

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	0	0	0	0	0	0	0	0

PWMSCLB is the programmable scale value used in scaling clock B to generate clock SB. Clock SB is generated by taking clock B, dividing it by the value in the PWMSCLB register and dividing that by two.

Clock SB = Clock B / (2 * PWMSCLB)

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

Freescale Semiconductor, Inc.

NOTE: When PWMSCLB = $00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLB).
Read: anytime
Write: anytime (causes the scale counter to load the PWMSCLB value)

PWM Channel Counter Registers (PWMCNTx)

Where: x=0,1,2,3,4,5,6,7

Address Offset: $00AC, $00AD, $00AE,$00AF, $00B0, $00B1, $00B2, $00B3

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:	0	0	0	0	0	0	0	0
Reset:	0	0	0	0	0	0	0	0

Read: anytime
Write: anytime (any value written causes PWM counter to be reset to $00)

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source. The counter can be read at any time without affecting the count or the operation of the PWM channel. In left aligned output mode, the counter counts from 0 to the value in the period register – 1. In center aligned output mode, the counter counts from 0 up to the value in the period register and then back down to 0.

Any value written to the counter causes the counter to reset to $00, the counter direction to be set to up, the immediate load of both duty and period registers with values from the buffers, and the output to change according to the polarity bit. The counter is also cleared at the end of the effective period (see Sections Left Aligned Outputs and Center Aligned Outputs for more details). When the channel is disabled (PWMEx=0), the PWMCNTx register does not count. When a channel becomes enabled (PWMEx=1), the associated PWM counter starts at the count in

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

Freescale Semiconductor, Inc.

the PWMCNTx register. For more detailed information on the operation of the counters, reference PWM Timer Counters.

In concatenated mode, writes to the 16-bit counter by using a 16-bit access or writes to either the low or high order byte of the counter will reset the 16-bit counter. Reads of the 16-bit counter must be made by 16-bit access to maintain data coherency

CAUTION: *Writing to the counter while the channel is enabled can cause an irregular PWM cycle to occur.*

PWM Channel Period Registers (PWMPERx)

Where: x=0,1,2,3,4,5,6,7

Address Offset: $00B4, $00B5, $00B6,$00B7, $00B8, $00B9, $00BA, $00BB

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	1	1	1	1	1	1	1	1

Read: anytime
Write: anytime

There is a dedicated period register for each channel. The value in this register determines the period of the associated PWM channel.

The period registers for each channel are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period register will go directly to the latches as well as the buffer.

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

Freescale Semiconductor, Inc.

NOTE: *Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active period due to the double buffering scheme.*

Reference PWM Period and Duty for more information.

To calculate the output period, take the selected clock source period for the channel of interest (A, B, SA, or SB) and multiply it by the value in the period register for that channel:

- Left Aligned Output (CAEx=0)

  PWMx Period = Channel Clock Period * PWMPERx

- Center Aligned Output (CAEx=1)

  PWMx Period = Channel Clock Period * (2 * PWMPERx)

For Boundary Case programming values, please refer to PWM Boundary Cases.

Freescale Semiconductor, Inc.

PWM Channel Duty Registers (PWMDTYx)

Where: x=0,1,2,3,4,5,6,7

Address Offset: $00BC, $00BD, $00BE,$00BF, $00C0, $00C1, $00C2, $00C3

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	1	1	1	1	1	1	1	1

Read: anytime
Write: anytime

There is a dedicated duty register for each channel. The value in this register determines the duty of the associated PWM channel. The duty value is compared to the counter and if it is equal to the counter value a match occurs and the output changes state.

The duty registers for each channel are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM will always be either the old duty waveform or the new duty waveform, not some variation in between. If the channel is not enabled, then writes to the duty register will go directly to the latches as well as the buffer.

NOTE: *Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active duty due to the double buffering scheme.*

Reference PWM Period and Duty for more information.

NOTE: *Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time. If the polarity bit is one, the output starts high and then goes low when the duty count is reached, so the*

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

Freescale Semiconductor, Inc.

duty registers contain a count of the high time. If the polarity bit is zero, the output starts low and then goes high when the duty count is reached, so the duty registers contain a count of the low time.

To calculate the output duty cycle (high time as a% of period) for a particular channel:

- Polarity = 0 (PPOLx=0)

  Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%

- Polarity = 1 (PPOLx=1)

  Duty Cycle = [PWMDTYx / PWMPERx] * 100%

For Boundary Case programming values, please refer to PWM Boundary Cases.

PWM Shutdown and Interrupt Control Register (PWMSDN)

Address Offset: $00C4

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PWMIF	PWMIE	0	PWMLVL	0	PWM7IN	PWM7INL	PWM7ENA
Write:			PWMRSTRT					
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

The PWMSDN register provides for the shutdown functionality of the PWM module in the emergency cases. Interruptions are enabled and monitored through this register.

Read: anytime
Write: anytime

PWMIF — PWM Interrupt Flag

Any change from passive to asserted (active) state or from active to passive state will be flagged by setting the PWMIF flag = 1. The flag is cleared by writing a ‘1’ to it. Writing a ‘0’ has no effect.

Freescale Semiconductor, Inc.

1 = change on PWM7IN input
0 = No change on PWM7IN input.

PWMIE — PWM Interrupt Enable

If interrupt is enabled an interrupt to the CPU is asserted.
1 = PWM interrupt is enabled.
0 = PWM interrupt is disabled.

PWMRSTRT — PWM Restart.

The PWM can only be restarted if the PWM channel input 7 is de-asserted. After writing a ‘1’ to the PWMRSTRT bit (trigger event) the PWM channels start running after the corresponding counter passes next ‘counter == 0’ phase.
Also if the PWM7ENA bit is reset to 0, the PWM do not start before the counter passes $00.
The bit is always read as ‘0’.

PWMLVL — PWM shutdown output Level.

If active level as defined by the PWM7IN input, gets asserted all enabled PWM channels are immediately driven to the level defined by PWMLVL.
1 = PWM outputs are forced to 1.
0 = PWM outputs are forced to 0

PWM7IN — PWM channel 7 input status.

This reflects the current status of the PWM7 pin.

PWM7INL — PWM shutdown active input level for ch. 7.

If the emergency shutdown feature is enabled (PWM7ENA = 1), this bit determines the active level of the PWM7channel.
1 = Active level is high
0 = Active level is low

PWM7ENA — PWM emergency shutdown Enable

If this bit is ‘1’ the pin associated with channel 7 is forced to input and the emergency shutdown feature is enabled.All the other bits in this register are meaningful only if PWM7ENA = 1.
1 = PWM emergency feature is enabled.
0 = PWM emergency feature disabled.

Freescale Semiconductor, Inc.

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

Functional Description

PWM Clock Select

There are four available clocks called clock A, clock B, clock SA (Scaled A), and clock SB (Scaled B). These four clocks are based on the Bus Clock.

Clock A and B can be software selected to be 1, 1/2, 1/4, 1/8,..., 1/64, 1/128 times the Bus Clock. Clock SA uses clock A as an input and divides it further with a reloadable counter. Similarly, Clock SB uses clock B as an input and divides it further with a reloadable counter. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8,..., or 512 in increments of divide by 2. Similar rates are available for clock SB. Each PWM channel has the capability of selecting one of two clocks, either the pre-scaled clock (clock A or B) or the scaled clock (clock SA or SB).

The block diagram in Figure 60 shows the four different clocks and how the scaled clocks are created.

Freescale Semiconductor, Inc.

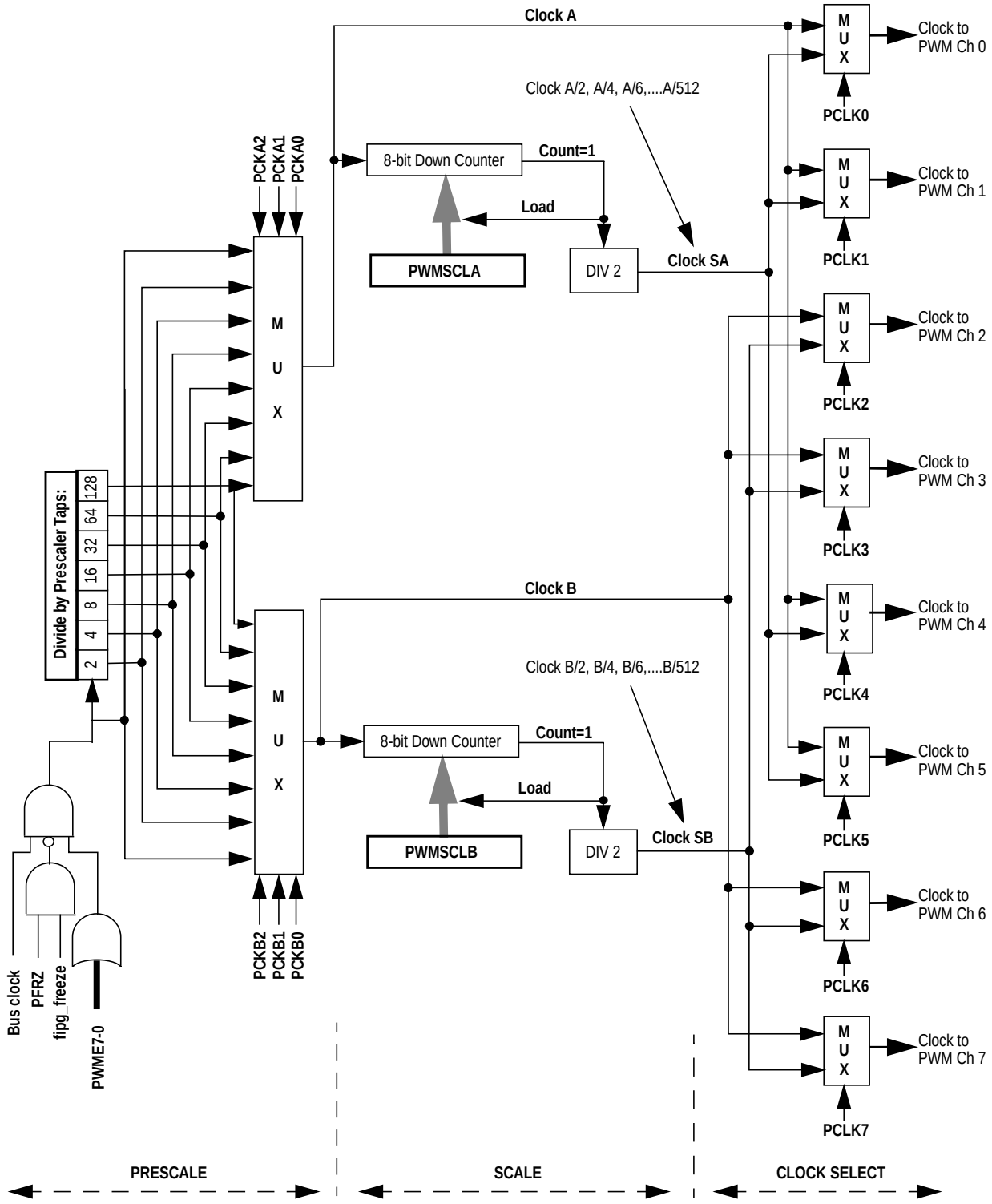


Figure 60 PWM Clock Select Block Diagram

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

Freescale Semiconductor, Inc.

Prescale

The input clock to the PWM prescaler is the Bus Clock. It can be disabled whenever the part is in freeze mode by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode the input clock to the prescaler is disabled. This is useful for emulation in order to freeze the PWM. The input clock can also be disabled when all eight PWM channels are disabled (PWMEx=0 in the PWME register; where x=0,1,...,7). This is useful for reducing power by disabling the prescale counter.

Clock A and clock B are scaled values of the input clock. The value is software selectable for both clock A and clock B and has options of 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, or 1/128 times the Bus Clock. The value selected for clock A is determined by the PCKA2, PCKA1, PCKA0 bits in the PWMPRCLK register. The value selected for clock B is determined by the PCKB2, PCKB1, PCKB0 bits also in the PWMPRCLK register.

Clock Scale

The scaled A clock uses clock A as an input and divides it further with a user programmable value and then divides this by 2. The scaled B clock uses clock B as an input and divides it further with a user programmable value and then divides this by 2. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8,..., or 512 in increments of divide by 2. Similar rates are available for clock SB.

Clock A is used as an input to an 8-bit down counter. This down counter loads a user programmable scale value from the scale register (PWMSCLA). When the down counter reaches one, two things happen; a pulse is output and the 8-bit counter is re-loaded. The output signal from this circuit is further divided by two. This gives a greater range with only a slight reduction in granularity. Clock SA equals Clock A divided by two times the value in the PWMSCLA register.

NOTE: *Clock SA = Clock A / (2 * PWMSCLA)*

When PWMSCLA = $00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Similarly, Clock B is used as an input to an 8-bit down counter followed by a divide by two producing clock SB. Thus, clock SB equals Clock B divided by two times the value in the PWMSCLB register.

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

Freescale Semiconductor, Inc.

NOTE: *Clock SB = Clock B / (2 * PWMSCLB)*

When PWMSCLB = $00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

As an example, consider the case in which the user writes $FF into the PWMSCLA register. Clock A for this case will be the Bus Clock divided by 4. A pulse will occur at a rate of once every 255x4 Bus Clock cycles. Passing this through the divide by two circuit produces a clock signal that is the Bus Clock divided by 2040 rate. Similarly, a value of $01 in the PWMSCLA register, when clock A is the Bus Clock divided by 4, will produce a clock that is the Bus Clock divided by 8 rate.

Writing to PWMSCLA or PWMSCLB causes the associated 8-bit down counter to be re-loaded. Otherwise, when changing rates the counter would have to count down to $01 before counting at the proper rate. Forcing the associated counter to re-load the scale register value every time PWMSCLA or PWMSCLB is written prevents this.

CAUTION: *Writing to the scale registers while channels are operating can cause irregularities in the PWM outputs.*

Clock Select

Each PWM channel has the capability of selecting one of two clocks. For channels 0, 1, 4, and 5 the clock choices are clock A or clock SA. For channels 2, 3, 6, and 7 the choices are clock B or clock SB. The clock selection is done with the PCLKx control bits in the PWMCLK register.

CAUTION: *Changing clock control bits while channels are operating can cause irregularities in the PWM outputs.*

PWM Channel Timers

The main part of the PWM module are the actual timers. Each of the timer channels has a counter, a period register and a duty register (each are 8-bit). The waveform output period is controlled by a match between the period register and the value in the counter. The duty is controlled by a match between the duty register and the counter value and causes the state of the output to change during the period. The starting polarity of the output is also selectable on a per channel basis. Shown below in Figure 61 is the block diagram for the PWM timer

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

Freescale Semiconductor, Inc.

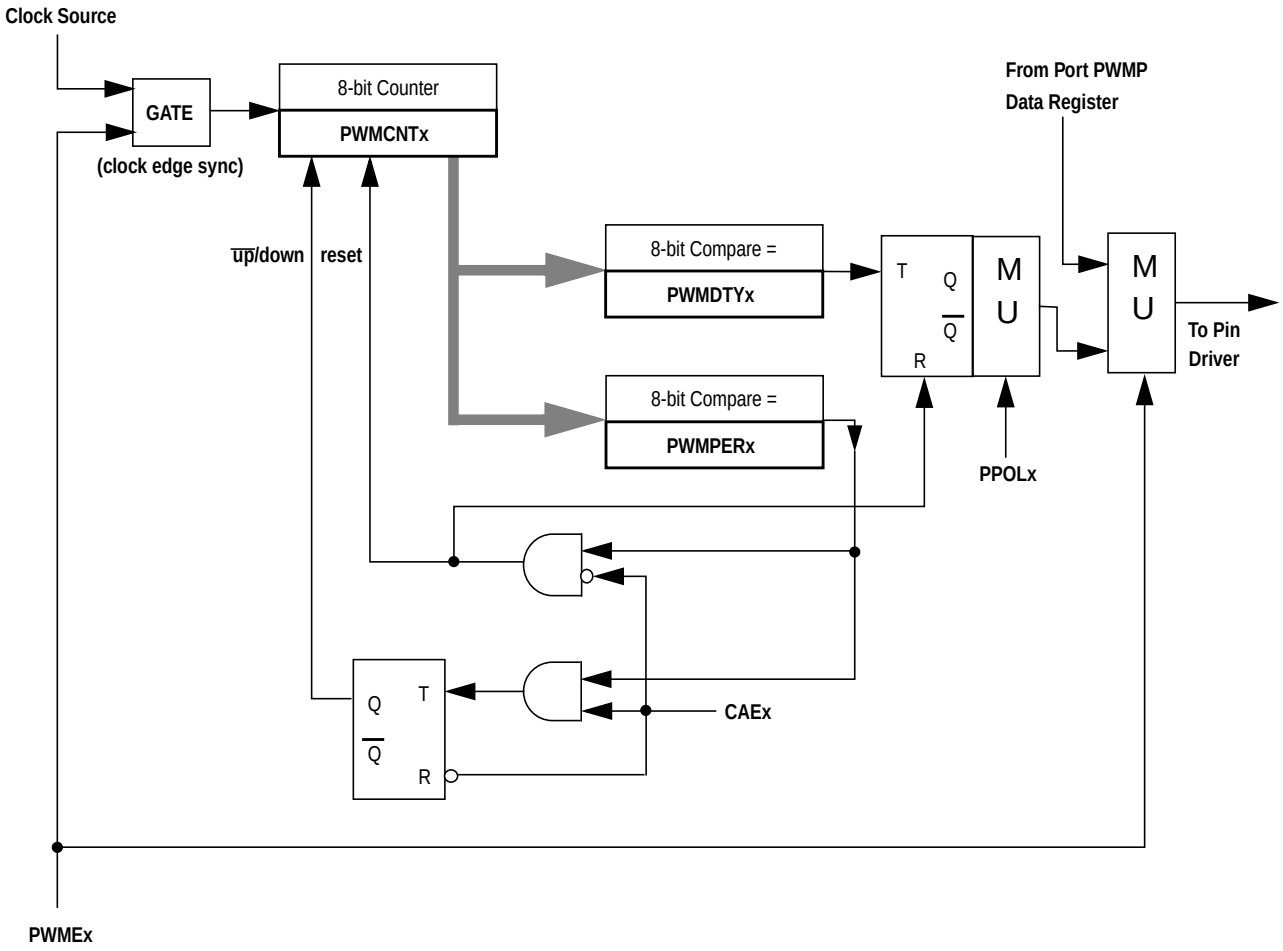


Figure 61 PWM Timer Channel Block Diagram

PWM Enable

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx=1), the associated PWM output signal is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle due to the synchronization of PWMEx and the clock source. An exception to this is when channels are concatenated. Refer to PWM 16-Bit Functions for more detail.

CAUTION: *The first PWM cycle after enabling the channel can be irregular.*

On the front end of the PWM timer, the clock is enabled to the PWM circuit by the PWMEx bit being high. There is an edge-synchronizing circuit to guarantee that the clock will only be enabled or disabled at an

Freescale Semiconductor, Inc.

edge. When the channel is disabled (PWMEx=0), the counter for the channel does not count.

PWM Polarity

Each channel has a polarity bit to allow starting a waveform cycle with a high or low signal. This is shown on the block diagram as a mux select of either the Q output or the $\overline{Q}$ output of the PWM output flip flop in Figure 61. When one of the bits in the PWMPOL register is set, the associated PWM channel output is high at the beginning of the waveform, then goes low when the duty count is reached. Conversely, if the polarity bit is zero, the output starts low and then goes high when the duty count is reached.

PWM Period and Duty

Dedicated period and duty registers exist for each channel and are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period and duty registers will go directly to the latches as well as the buffer.

A change in duty or period can be forced into effect ‘immediately’ by writing the new value to the duty and/or period registers and then writing to the counter. This forces the counter to reset and the new duty and/or period values to be latched. In addition, since the counter is readable it is possible to know where the count is with respect to the duty value and software can be used to make adjustments

CAUTION: *When forcing a new period or duty into effect immediately, an irregular PWM cycle can occur.*

NOTE: Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time.

Freescale Semiconductor, Inc.

PWM Timer Counters

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source (reference PWM Clock Select for the available clock sources and rates). The counter compares to two registers, a duty register and a period register as shown in Figure 61. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match between the PWM counter and the period register behaves differently depending on what output mode is selected as shown in Figure 61 and described in Left Aligned Outputs and Center Aligned Outputs.

Each channel counter can be read at anytime without affecting the count or the operation of the PWM channel.

Any value written to the counter causes the counter to reset to $00, the counter direction to be set to up, the immediate load of both duty and period registers with values from the buffers, and the output to change according to the polarity bit. When the channel is disabled (PWMEx=0), the counter stops. When a channel becomes enabled (PWMEx=1), the associated PWM counter continues from the count in the PWMCNTx register. This allows the waveform to continue where it left off when the channel is re-enabled. When the channel is disabled, writing ‘0’ to the period register will cause the counter to reset on the next selected clock.

NOTE: *If the user wants to start a new ‘clean’ PWM waveform without any ‘history’ from the old waveform, the user must write to channel counter (PWMCNTx) prior to enabling the PWM channel (PWMEx=1).*

Generally, writes to the counter are done prior to enabling a channel in order to start from a known state. However, writing a counter can also be done while the PWM channel is enabled (counting). The effect is similar to writing the counter when the channel is disabled except that the new period is started immediately with the output set according to the polarity bit.

CAUTION: *Writing to the counter while the channel is enabled can cause an irregular PWM cycle to occur.*

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

Freescale Semiconductor, Inc.

The counter is cleared at the end of the effective period (see Left Aligned Outputs and Center Aligned Outputs for more details).

Table 62 PWM Timer Counter Conditions

Counter Clears ($00)	Counter Counts	Counter Stops
When PWMCNTx register written to any value	When PWM channel is enabled (PWMEx=1). Counts from last value in PWMCNTx.	When PWM channel is disabled (PWMEx=0)
Effective period ends		

Left Aligned Outputs

The PWM timer provides the choice of two types of outputs, Left Aligned or Center Aligned outputs. They are selected with the CAEx bits in the PWMCAE register. If the CAEx bit is cleared (CAEx=0), the corresponding PWM output will be left aligned.

In left aligned output mode, the 8-bit counter is configured as an up counter only. It compares to two registers, a duty register and a period register as shown in the block diagram in Figure 61. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match between the PWM counter and the period register resets the counter and the output flip-flop as shown in Figure 61 as well as performing a load from the double buffer period and duty register to the associated registers as described in PWM Period and Duty. The counter counts from 0 to the value in the period register – 1.

NOTE: *Changing the PWM output mode from Left Aligned Output to Center Aligned Output (or vice versa) while channels are operating can cause irregularities in the PWM output. It is recommended to program the output mode before enabling the PWM channel.*

Freescale Semiconductor, Inc.

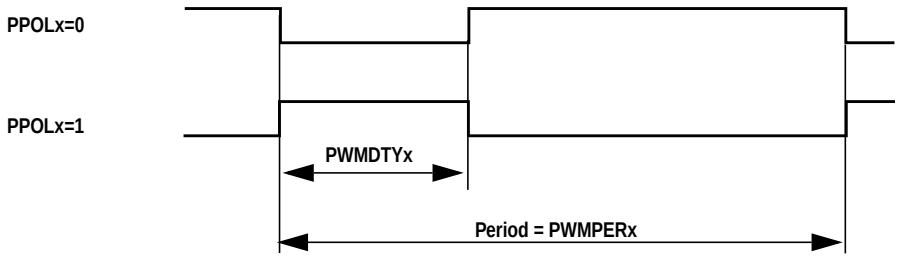


Figure 62 PWM Left Aligned Output Waveform

To calculate the output frequency in left aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB) and divide it by the value in the period register for that channel.

- PWMx Frequency = Clock (A, B, SA, or SB) / PWMPERx
- PWMx Duty Cycle (high time as a% of period):
  - Polarity = 0 (PPOLx=0)

    Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
  - Polarity = 1 (PPOLx=1)

    Duty Cycle = [PWMDTYx / PWMPERx] * 100%

As an example of a left aligned output, consider the following case:

Clock Source = Bus Clock, where Bus Clock=10MHz (100ns period)

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10MHz/4 = 2.5MHz

PWMx Period = 400ns

PWMx Duty Cycle = 3/4 *100% = 75%

The output waveform generated is shown in Figure 63.

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

Freescale Semiconductor, Inc.

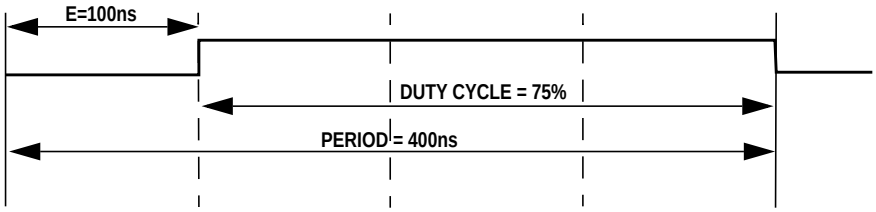


Figure 63 PWM Left Aligned Output Example Waveform

Center Aligned Outputs

For Center Aligned Output Mode selection, set the CAEx bit (CAEx=1) in the PWMCAE register and the corresponding PWM output will be center aligned.

The 8-bit counter operates as an up/down counter in this mode and is set to up whenever the counter is equal to $00. The counter compares to two registers, a duty register and a period register as shown in the block diagram in Figure 61. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match between the PWM counter and the period register changes the counter direction from an up-count to a down-count. When the PWM counter decrements and matches the duty register again, the output flip-flop changes state causing the PWM output to also change state. When the PWM counter decrements and reaches zero, the counter direction changes from a down-count back to an up-count and a load from the double buffer period and duty registers to the associated registers is performed as described in PWM Period and Duty. The counter counts from 0 up to the value in the period register and then back down to 0. Thus the effective period is PWMPERx*2.

NOTE: *Changing the PWM output mode from Left Aligned Output to Center Aligned Output (or vice versa) while channels are operating can cause irregularities in the PWM output. It is recommended to program the output mode before enabling the PWM channel.*

Freescale Semiconductor, Inc.

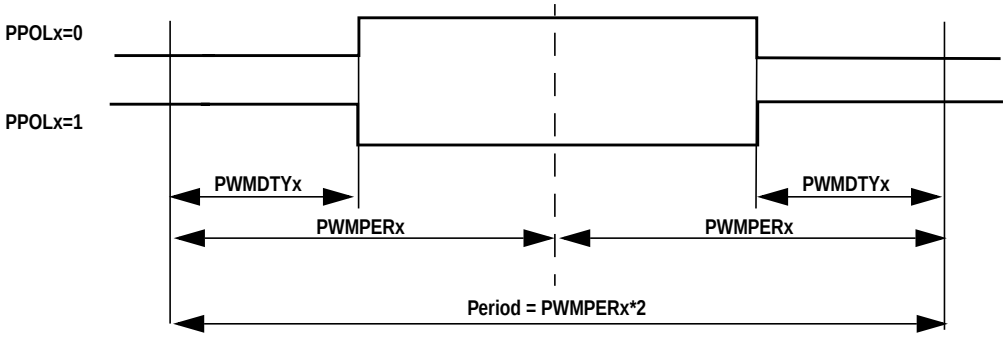


Figure 64 PWM Center Aligned Output Waveform

To calculate the output frequency in center aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB) and divide it by twice the value in the period register for that channel.

- PWMx Frequency = Clock (A, B, SA, or SB) / (2*PWMPERx)
- PWMx Duty Cycle (high time as a% of period):
  - Polarity = 0 (PPOLx=0)

    Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
  - Polarity = 1 (PPOLx=1)

    Duty Cycle = [PWMDTYx / PWMPERx] * 100%

As an example of a center aligned output, consider the following case:

Clock Source = Bus Clock, where Bus Clock =10MHz (100ns period)

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10MHz/8 = 1.25MHz

PWMx Period = 800ns

PWMx Duty Cycle = 3/4 *100% = 75%

The output waveform generated is shown in Figure 65

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

Freescale Semiconductor, Inc.

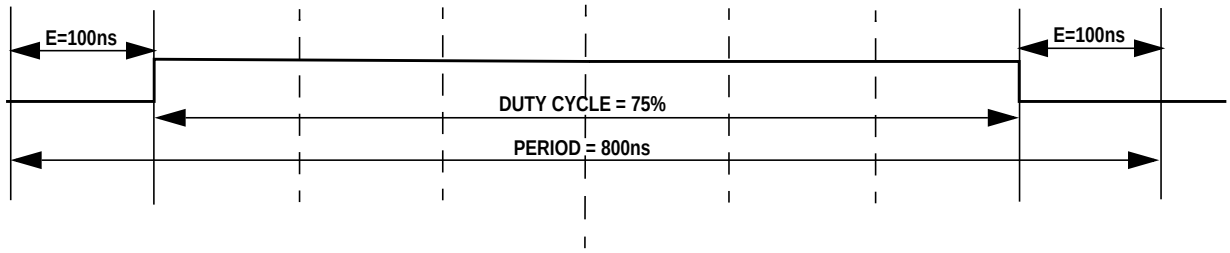


Figure 65 PWM Center Aligned Output Example Waveform

PWM 16-Bit Functions

The PWM timer also has the option of generating 8-channels of 8-bits or 4-channels of 16-bits for greater PWM resolution. This 16-bit channel option is achieved through the concatenation of two 8-bit channels.

The PWMCTL register contains four control bits, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. Channels 6 and 7 are concatenated with the CON67 bit, channels 4 and 5 are concatenated with the CON45 bit, channels 2 and 3 are concatenated with the CON23 bit, and channels 0 and 1 are concatenated with the CON01 bit.

CAUTION: *Change these bits only when both corresponding channels are disabled.*

When channels 6 and 7 are concatenated, channel 6 registers become the high order bytes of the double byte channel as shown in Figure 66. Similarly, when channels 4 and 5 are concatenated, channel 4 registers become the high order bytes of the double byte channel. When channels 2 and 3 are concatenated, channel 2 registers become the high order bytes of the double byte channel. When channels 0 and 1 are concatenated, channel 0 registers become the high order bytes of the double byte channel.

Freescale Semiconductor, Inc.

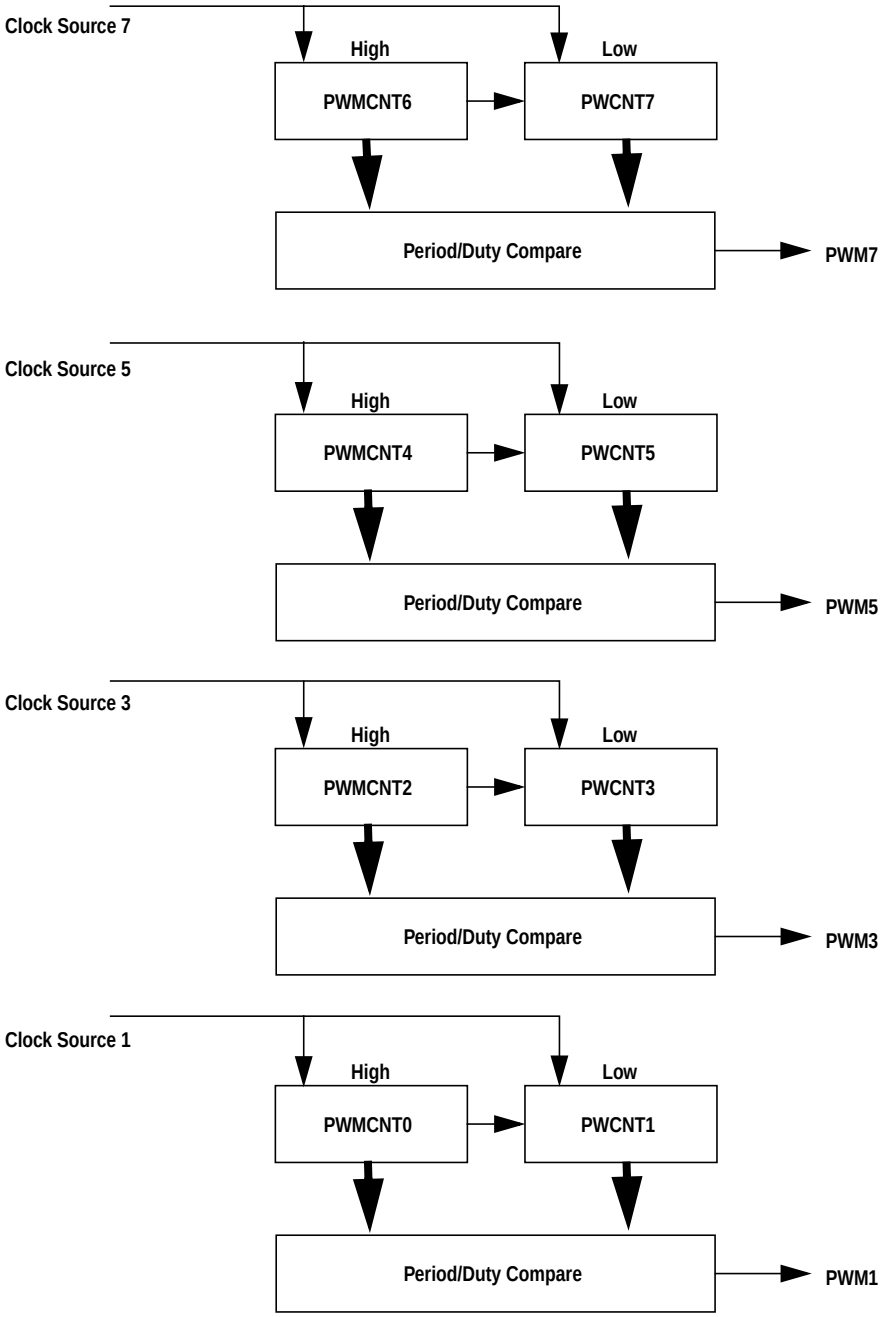


Figure 66 PWM 16-Bit Mode

When using the 16-bit concatenated mode, the clock source is determined by the low order 8-bit channel clock select control bits. That is channel 7 when channels 6 and 7 are concatenated, channel 5 when channels 4 and 5 are concatenated, channel 3 when channels 2 and 3

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

Freescale Semiconductor, Inc.

are concatenated, and channel 1 when channels 0 and 1 are concatenated. The resulting PWM is output to the pins of the corresponding low order 8-bit channel as also shown in Figure 66. The polarity of the resulting PWM output is controlled by the PPOLx bit of the corresponding low order 8-bit channel as well.

Once concatenated mode is enabled (CONxx bits set in PWMCTL register) then enabling/disabling the corresponding 16-bit PWM channel is controlled by the low order PWMEx bit. In this case, the high order bytes PWMEx bits have no effect and their corresponding PWM output is disabled.

In concatenated mode, writes to the 16-bit counter by using a 16-bit access or writes to either the low or high order byte of the counter will reset the 16-bit counter. Reads of the 16-bit counter must be made by 16-bit access to maintain data coherency.

Either left aligned or center aligned output mode can be used in concatenated mode and is controlled by the low order CAEx bit. The high order CAEx bit has no effect.

The table shown below is used to summarize which channels are used to set the various control bits when in 16-bit mode.

Table 63 16-bit Concatenation Mode Summary

CONxx	PWMEx	PPOLx	PCLKx	CAEx	PWMx OUTPUT
CON67	PWME7	PPOL7	PCLK7	CAE7	PWM7
CON45	PWME5	PPOL5	PCLK5	CAE5	PWM5
CON23	PWME3	PPOL3	PCLK3	CAE3	PWM3
CON01	PWME1	PPOL1	PCLK1	CAE1	PWM1

Freescale Semiconductor, Inc.

PWM Boundary Cases

The following table summarizes the boundary conditions for the PWM regardless of the output mode (Left Aligned or Center Aligned) and 8-bit (normal) or 16-bit (concatenation):

Table 64 PWM Boundary Cases

PWMDTYx	PWMPERx	PPOLx	PWMx Output
$00 (indicates no duty)	>$00	1	Always Low
$00 (indicates no duty)	>$00	0	Always High
XX	$00[1] (indicates no period)	1	Always High
XX	$00[1] (indicates no period)	0	Always Low
>= PWMPERx	XX	1	Always High
>= PWMPERx	XX	0	Always Low

1. Counter=$00 and does not count.

Freescale Semiconductor, Inc.

Low Power Options

This section summarizes the low power options available in the PWM module. Low power design practices are implemented where possible.

Run Mode

While in run mode, if all eight PWM channels are disabled (PWMEx=0 in the PWME register, where x=0,1,...,7), the prescaler counter shuts off for power savings (see Figure 60).

Wait Mode

The PWM will keep running in WAIT unless the PSWAI bit in the PWMCTL register is enabled. This allows for lower power consumption in WAIT mode by disabling the input clock to the prescaler. When in WAIT mode with this bit set, no activity in the PWM occurs and the PWM outputs go to a static state (high or low).

Stop Mode

In STOP mode, the PWM module is stopped since all the clocks from IP bus to the module are stopped. The PWM outputs go to a static state (high or low).

Reset Initialization

The reset state of each individual bit is listed within the Register Description section (see Register Descriptions) which details the registers and their bit-fields. All special functions or modes which are initialized during or just following reset are described within this section.

- The 8-bit up/down counter is configured as an up counter out of reset.
- All the channels are disabled and all the counters don’t count.

Freescale Semiconductor, Inc.

Interrupts

The PWM module has only one interrupt which is generated at the time of emergency shutdown, if the corresponding enable bit (PWMIE in the PWMSDN register) is set. The vector address for the PWM Emergency Shutdown interrupt is $FF8C,$FF8D.

Freescale Semiconductor, Inc.

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

Freescale Semiconductor, Inc.

Enhanced Capture Timer (ECT)

Contents

Overview 371
Features 372
Modes of Operation 372
Abbreviations 372
Block Diagram 373
External Pin Descriptions 374
Register Map 375
Register Descriptions 378
Functional Description 408
Interrupts 416

Overview

The HC12 Enhanced Capture Timer module has the features of the HC12 Standard Timer module enhanced by additional features in order to enlarge the field of applications, in particular for automotive ABS applications.

The basic timer consists of a 16-bit, software-programmable counter driven by a prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from microseconds to many seconds.

A full access for the counter registers or the input capture/output compare registers should take place in one clock cycle. Accessing high byte and low byte separately for all of these registers may not yield the same result as accessing them in one word.

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

Freescale Semiconductor, Inc.

Features

- 16-Bit Buffer Register for four Input Capture (IC) channels.
- Four 8-Bit Pulse Accumulators with 8-bit buffer registers associated with the four buffered IC channels. Configurable also as two 16-Bit Pulse Accumulators.
- 16-Bit Modulus Down-Counter with 4-bit Prescaler.
- Four user selectable Delay Counters for input noise immunity increase.
- Support for only 16-bit access on the IP bus.

Modes of Operation

STOP:	Timer and modulus counter are off since clocks are stopped.
FREEZE:	The ECT module enters freeze mode when background debug mode (BDM) is active. In freeze mode, timer and modulus counter keep on running, unless TSFRZ in TSCR1 (see page 382) is set to one.
WAIT:	Counters keep on running, unless TSWAI in TSCR1 is set to one.
NORMAL:	Timer and modulus counter keep on running, unless TEN in TSCR1 and MCEN in MCCTL (see page 396), respectively, are cleared.

Abbreviations

Following abbreviations are used in the document.

PACLK – 16-bit pulse accumulator A (PACA) clock

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

Freescale Semiconductor, Inc.

Block Diagram

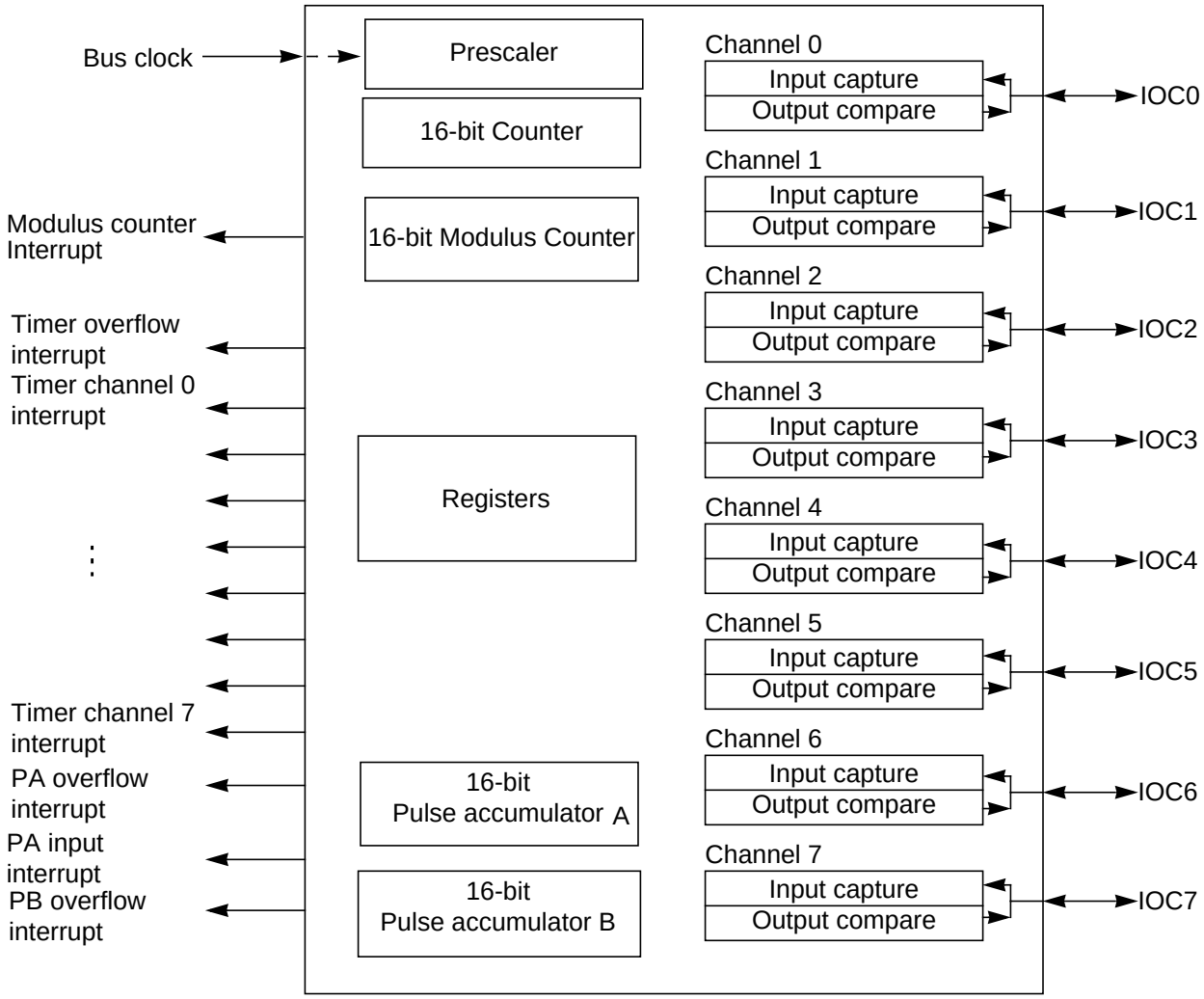


Figure 67 Timer Block Diagram

Freescale Semiconductor, Inc.

External Pin Descriptions

IOC7 (PT7) Input capture and Output compare channel 7 - This pin serves as input capture or output compare for channel 7.

IOC6 (PT6) Input capture and Output compare channel 6 - This pin serves as input capture or output compare for channel 6.

IOC5 (PT5) Input capture and Output compare channel 5 - This pin serves as input capture or output compare for channel 5.

IOC4 (PT4) Input capture and Output compare channel 4 - This pin serves as input capture or output compare for channel 4.

IOC3 (PT3) Input capture and Output compare channel 3 - This pin serves as input capture or output compare for channel 3.

IOC2 (PT2) Input capture and Output compare channel 2 - This pin serves as input capture or output compare for channel 2.

IOC1 (PT1) Input capture and Output compare channel 1 - This pin serves as input capture or output compare for channel 1.

IOC0 (PT0) Input capture and Output compare channel 0 - This pin serves as input capture or output compare for channel 0.

Freescale Semiconductor, Inc.

Register Map

Register name		Bit 7	6	5	4	3	2	1	Bit 0	Address Offset
TIOS	Read:	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0	$0040
	Write:									
CFORC	Read:	0	0	0	0	0	0	0	0	$0041 (1)
	Write:	FOC7	FOC6	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0	
OC7M	Read:	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0	$0042
	Write:									
OC7D	Read:	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0	$0043
	Write:									
TCNT (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8	$0044 (2)
	Write:									
TCNT (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$0045 (2)
	Write:									
TSCR1	Read:	TEN	TSWAI	TSFRZ	TFFCA	0	0	0	0	$0046
	Write:									
TTOV	Read:	TOV7	TOV6	TOV5	TOV4	TOV3	TOV2	TOV1	TOV0	$0047
	Write:									
TCTL1	Read:	OM7	OL7	OM6	OL6	OM5	OL5	OM4	OL4	$0048
	Write:									
TCTL2	Read:	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0	$0049
	Write:									
TCTL3	Read:	EDG7B	EDG7A	EDG6B	EDG6A	EDG5B	EDG5A	EDG4B	EDG4A	$004A
	Write:									
TCTL4	Read:	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A	$004B
	Write:									
TIE	Read:	C7I	C6I	C5I	C4I	C3I	C2I	C1I	C0I	$004C
	Write:									
TSCR2	Read:	TOI	0	0	0	TCRE	PR2	PR1	PR0	$004D
	Write:									
TFLG1	Read:	C7F	C6F	C5F	C4F	C3F	C2F	C1F	C0F	$004E
	Write:									
TFLG2	Read:	TOF	0	0	0	0	0	0	0	$004F
	Write:									
TC0 (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8	$0050 (3)
	Write:									
TC0 (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$0051 (3)
	Write:									

= Unimplemented or reserved

Figure 68 Enhanced Capture Timer Register Map

Freescale Semiconductor, Inc.

Register name		Bit 7	6	5	4	3	2	1	Bit 0	Address Offset
TC1 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	$0052 (3)
TC1 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$0053 (3)
TC2 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	$0054 (3)
TC2 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$0055 (3)
TC3 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	$0056 (3)
TC3 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$0057 (3)
TC4 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	$0058 (3)
TC4 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$0059 (3)
TC5 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	$005A (3)
TC5 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$005B (3)
TC6 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	$005C (3)
TC6 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$005D (3)
TC7 (hi)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8	$005E (3)
TC7 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$005F (3)
PACTL	Read: Write:	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI	$0060
PAFLG	Read: Write:	0	0	0	0	0	0	PAOVF	PAIF	$0061
PACN3 (hi)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$0062
PACN2 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$0063
PACN1 (hi)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$0064
PACN0 (lo)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0	$0065

= Unimplemented or reserved

Figure 68 Enhanced Capture Timer Register Map (Continued)

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

Freescale Semiconductor, Inc.

Register name		Bit 7	6	5	4	3	2	1	Bit 0	Address Offset
MCCTL	Read:	MCZI	MODMC	RDMCL	0	0	MCEN	MCPR1	MCPR0	$0066
	Write:				ICLAT	FLMC				
MCFLG	Read:	MCZF	0	0	0	POLF3	POLF2	POLF1	POLF0	$0067
	Write:									
ICPAR	Read:	0	0	0	0	PA3EN	PA2EN	PA1EN	PA0EN	$0068
	Write:									
DLYCT	Read:	0	0	0	0	0	0	DLY1	DLY0	$0069
	Write:									
ICOVW	Read:	NOVW7	NOVW6	NOVW5	NOVW4	NOVW3	NOVW2	NOVW1	NOVW0	$006A
	Write:									
ICSYS	Read:	SH37	SH26	SH15	SH04	TFMOD	PACMX	BUFEN	LATQ	$006B (4)
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$006C
	Write:									
Reserved for Factory Test	Read:	Reads to this register return unpredictable values								$006D (2)
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$006E
	Write:									
Reserved	Read:	0	0	0	0	0	0	0	0	$006F
	Write:									
PBCTL	Read:	0	PBEN	0	0	0	0	PBOVI	0	$0070
	Write:									
PBFLG	Read:	0	0	0	0	0	0	PBOVF	0	$0071
	Write:									
PA3H	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$0072 (5)
	Write:									
PA2H	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$0073 (5)
	Write:									
PA1H	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$0074 (5)
	Write:									
PA0H	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$0075 (5)
	Write:									
MCCNT (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8	$0076
	Write:									
MCCNT (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$0077
	Write:									
TC0H (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8	$0078 (5)
	Write:									
TC0H (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$0079 (5)
	Write:									

= Unimplemented or reserved

Figure 68 Enhanced Capture Timer Register Map (Continued)

Freescale Semiconductor, Inc.

Register name		Bit 7	6	5	4	3	2	1	Bit 0	Address Offset
TC1H (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8	$007A (5)
	Write:									
TC1H (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$007B (5)
	Write:									
TC2H (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8	$007C (5)
	Write:									
TC2H (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$007D (5)
	Write:									
TC3H (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8	$007E (5)
	Write:									
TC3H (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$007F (5)
	Write:									

= Unimplemented or reserved

Figure 68 Enhanced Capture Timer Register Map (Continued)

1. Always read $00.
2. Only writable in special modes.
3. Write to these registers have no meaning or effect during input capture.
4. May be written once in normal modes but writes are always permitted when special modes.
5. Write has no effect.

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

Freescale Semiconductor, Inc.

Timer Input Capture/Output Compare Select (TIOS)

Register offset: $0040

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0
Reset:	0	0	0	0	0	0	0	0

Read or write anytime.

IOS[7:0] — Input Capture or Output Compare Channel Configuration

1 = The corresponding channel acts as an output compare.

0 = The corresponding channel acts as an input capture

Timer Compare Force Register (CFORC)

Register offset: $0041

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	0
Write:	FOC7	FOC6	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0
Reset:	0	0	0	0	0	0	0	0

Read anytime but will always return $00 (1 state is transient). Write anytime.

FOC[7:0] — Force Output Compare Action for Channel 7–0

A write to this register with the corresponding data bit(s) set causes the action which is programmed for output compare ‘n’ to occur immediately. The action taken is the same as if a successful comparison had just taken place with the TCn register except the interrupt flag does not get set

NOTE: *A successful channel 7 output compare overrides any channel 6:0 compares. If forced output compare on any channel occurs at the same*

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

Freescale Semiconductor, Inc.

time as the successful output compare then forced output compare action will take precedence and interrupt flag won't get set.

Output Compare 7 Mask Register (OC7M)

Register offset: $0042

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0
Reset:	0	0	0	0	0	0	0	0

Read or write anytime.

Setting the OC7Mn (n ranges from 0 to 6) will set the corresponding port to be an output port when the corresponding IOSn (TIOS register; n ranges from 0 to 6) bit is set to be an output compare.

NOTE: *A successful channel 7 output compare overrides any channel 6:0 compares. For each OC7M bit that is set, the output compare action reflects the corresponding OC7D bit.*

Freescale Semiconductor, Inc.

Output Compare 7 Data Register (OC7D)

Register offset: $0043

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0
Reset:	0	0	0	0	0	0	0	0

Read or write anytime.

A channel 7 output compare can cause bits in the output compare 7 data register to transfer to the timer port data register depending on the output compare 7 mask register.

Timer Count Register (TCNT)

Register offset: $0044–$0045

	Bit 15	14	13	12	11	10	9	Bit 8
Read:	Bit 15	14	13	12	11	10	9	Bit 8
Write:								

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read anytime.

Write has no meaning or effect in the normal mode; only writable in special modes.

The 16-bit main timer is an up counter.

A full access for the counter register should take place in one clock cycle. A separate read/write for high byte and low byte will give a different result than accessing them as a word.

Freescale Semiconductor, Inc.

The period of the first count after a write to the TCNT registers may be a different size because the write is not synchronized with the prescaler clock.

Timer System Control Register 1 (TSCR1)

Register offset: $0046

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TEN	TSWAI	TSFRZ	TFFCA	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read or write anytime.

TEN — Timer Enable

1 = Allows the timer to function normally.

0 = Disables the main timer, including the counter. Can be used for reducing power consumption.

If for any reason the timer is not active, there is no ÷64 clock for the pulse accumulator since the ÷64 is generated by the timer prescaler.

TSWAI — Timer Module Stops While in Wait

1 = Disables the timer module when the MCU is in the wait mode. Timer interrupts cannot be used to get the MCU out of wait.

0 = Allows the timer module to continue running during wait.

TSWAI also affects pulse accumulators and modulus down counters.

Freescale Semiconductor, Inc.

TSFRZ — Timer and Modulus Counter Stop While in Freeze Mode

1 = Disables the timer and modulus counter whenever the MCU is in freeze mode. This is useful for emulation.

0 = Allows the timer and modulus counter to continue running while in freeze mode.

TSFRZ does not stop the pulse accumulator.

NOTE: *The ECT module enters freeze mode when background debug mode (BDM) is active. Refer to the Fast Background Debug Module (FBDM) section about the background debug mode.*

TFFCA — Timer Fast Flag Clear All

1 = For TFLG1, a read from an input capture or a write to the output compare channel causes the corresponding channel flag, CnF, to be cleared. For TFLG2, any access to the TCNT register clears the TOF flag. Any access to the PACN3 and PACN2 registers clears the PAOVF and PAIF flags in the PAFLG register. Any access to the PACN1 and PACN0 registers clears the PBOVF flag in the PBFLG register. This has the advantage of eliminating software overhead in a separate clear sequence. Extra care is required to avoid accidental flag clearing due to unintended accesses.

0 = Allows the timer flag clearing to function normally.

Timer Toggle On Overflow Register 1 (TTOV)

Register offset: $0047

	Bit 7	6	5	4	3	2	1	0
Read: Write:	TOV7	TOV6	TOV5	TOV4	TOV3	TOV2	TOV1	TOV0
Reset:	0	0	0	0	0	0	0	0

Read or write anytime.

TOVx — Toggle On Overflow Bits

Freescale Semiconductor, Inc.

TOVx toggles output compare pin on overflow. This feature only takes effect when in output compare mode. When set, it takes precedence over forced output compare but not channel 7 override events.

1 = Toggle output compare pin on overflow feature enabled

0 = Toggle output compare pin on overflow feature disabled

Freescale Semiconductor, Inc.

Timer Control Registers 1and 2 (TCTL1, TCTL2)

Register offset: $0048-$0049

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	OM7	OL7	OM6	OL6	OM5	OL5	OM4	OL4
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
Reset:	0	0	0	0	0	0	0	0

Read or write anytime.

OMn — Output Mode

OLn — Output Level

These eight pairs of control bits are encoded to specify the output action to be taken as a result of a successful OCn compare. When either OMn or OLn is one, the pin associated with OCn becomes an output tied to OCn.

NOTE: *To enable output action by OMn and OLn bits on timer port, the corresponding bit in OC7M should be cleared.*

Table 65 Compare Result Output Action

OMn	OLn	Action
0	0	Timer disconnected from output pin logic
0	1	Toggle OCn output line
1	0	Clear OCn output line to zero
1	1	Set OCn output line to one

To operate the 16-bit pulse accumulators A and B (PACA and PACB) independently of input capture or output compare 7 and 0 respectively the user must set the corresponding bits IOSn = 1, OMn = 0 and OLn = 0. OC7M7 or OC7M0 in the OC7M register must also be cleared.

Freescale Semiconductor, Inc.

Timer Control Registers 3 and 4 (TCTL3, TCTL4)

Register offset: $004A-$004B

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	EDG7B	EDG7A	EDG6B	EDG6A	EDG5B	EDG5A	EDG4B	EDG4A
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
Reset:	0	0	0	0	0	0	0	0

Read or write anytime.

EDGnB, EDGnA — Input Capture / Pulse Accumulator Edge Control

These eight pairs of control bits configure the input capture edge detector circuits.

The four pairs of control bits of TCTL4 also configure the 8 bit pulse accumulators PAC0–3.

For 16-bit pulse accumulator PACB, EDG0B & EDG0A, control bits of TCTL4 will decide the active edge.

Table 66 Edge Detector Circuit Configuration

EDGnB	EDGnA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge (rising or falling)

NOTE: *For the 16-bit pulse accumulator PACA, refer to the register PACTL for active edge control.*

Freescale Semiconductor, Inc.

Timer Interrupt Enable Register (TIE)

Register offset: $004C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	C7I	C6I	C5I	C4I	C3I	C2I	C1I	C0I
Reset:	0	0	0	0	0	0	0	0

Read or write anytime.

The bits in TIE correspond bit-for-bit with the bits in the TFLG1 status register. If cleared, the corresponding flag is disabled from causing a hardware interrupt. If set, the corresponding flag is enabled to cause a interrupt.

C7I–C0I — Input Capture/Output Compare “x” Interrupt Enable

Timer System Control Register 2 (TSCR2)

Register offset: $004D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOI	0	0	0	TCRE	PR2	PR1	PR0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read or write anytime.

TOI — Timer Overflow Interrupt Enable

1 = Hardware interrupt requested when TOF flag set

0 = Interrupt inhibited

TCRE — Timer Counter Reset Enable

This bit allows the timer counter to be reset by a successful output compare 7 event. This mode of operation is similar to an up-counting modulus counter.

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

Freescale Semiconductor, Inc.

1 = Counter reset by a successful output compare 7
0 = Counter reset inhibited and counter free runs

If TC7 = $0000 and TCRE = 1, TCNT will stay at $0000 continuously. If TC7 = $FFFF and TCRE = 1, TOF in TFLG2 (see page 389) will never be set when TCNT is reset from $FFFF to $0000.

PR2, PR1, PR0 — Timer Prescaler Select

These three bits specify the number of ÷2 stages that are to be inserted between the bus clock and the main timer counter.

Table 67 Prescaler Selection

PR2	PR1	PR0	Prescale Factor
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

The newly selected prescale factor will not take effect until the next synchronized edge where all prescale counter stages equal zero.

Main Timer Interrupt Flag 1 (TFLG1)

Register offset: $004E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	C7F	C6F	C5F	C4F	C3F	C2F	C1F	C0F
Reset:	0	0	0	0	0	0	0	0

TFLG1 indicates when interrupt conditions have occurred.

Read anytime.

Freescale Semiconductor, Inc.

Write used in the clearing mechanism. Writing one will cause the corresponding bits to be cleared. Writing a zero has no effect.

Use of the TFMOD bit in the ICSYS register (see page 401) in conjunction with the use of the ICOVW (see page 400) register allows a timer interrupt to be generated after capturing two values in the capture and holding registers instead of generating an interrupt for every capture.

When TFFCA bit in TSCR1 register (see page 382) is set, a read from an input capture or a write into an output compare channel will cause the corresponding channel flag CnF to be cleared.

C7F–C0F — Input Capture/Output Compare Channel ‘n’ Flag.

C0F can also be set by 16-bit Pulse Accumulator B (PACB). C3F–C0F can also be set by 8-bit pulse accumulators PAC3–PAC0.

Main Timer Interrupt Flag 2 (TFLG2)

Register offset: $004F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOF	0	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

TFLG2 indicates when interrupt conditions have occurred.

Read anytime.

Write used in the clearing mechanism. Writing one will cause the corresponding bits to be cleared. Writing a zero has no effect.

Any access to TCNT will clear TFLG2 register if the TFFCA bit in TSCR1 (see page 382) register is set.

TOF — Timer Overflow Flag

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

Set when 16-bit free-running timer overflows from $FFFF to $0000. This bit is cleared automatically by a write to the TFLG2 register with bit 7 set. (See also TCRE control bit explanation in TSCR2 register - page 387.)

Timer Input Capture/Output Compare Registers 0–7 (TC0–TC7)

Register offset: $0050–$005F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
	Bit 15	14	13	12	11	10	9	Bit 8
Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
Reset:	0	0	0	0	0	0	0	0

Read anytime.

Write anytime for output compare function. Writes to these registers have no meaning or effect during input capture. All timer input capture/output compare registers are reset to $0000.

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Freescale Semiconductor, Inc.

16-Bit Pulse Accumulator A Control Register (PACTL)

Register offset: $0060

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read or write any time.

16-Bit Pulse Accumulator A (PACA) is formed by cascading the 8-bit pulse accumulators PAC3 and PAC2.

When PAEN is set, the PACA is enabled. The PACA shares the input pin with IC7.

PAEN — Pulse Accumulator A System Enable

1 = Pulse Accumulator A system enabled. The two 8-bit pulse accumulators PAC3 and PAC2 are cascaded to form the PACA 16-bit pulse accumulator. When PACA is enabled, the PACN3 and PACN2 registers contents are respectively the high and low byte of the PACA.
PA3EN and PA2EN control bits in ICPAR (see page 399) have no effect.
Pulse Accumulator Input Edge Flag (PAIF) function is enabled.

0 = 16-Bit Pulse Accumulator A system disabled. 8-bit PAC3 and PAC2 can be enabled when their related enable bits in ICPAR are set.
Pulse Accumulator Input Edge Flag (PAIF) function is disabled.

PAEN is independent from TEN in TSCR1(see page 382). With timer disabled, the pulse accumulator can still function unless pulse accumulator is disabled.

Freescale Semiconductor, Inc.

PAMOD — Pulse Accumulator Mode

This bit is active only when the Pulse Accumulator A is enabled (PAEN = 1).

1 = gated time accumulation mode
0 = event counter mode

PEDGE — Pulse Accumulator Edge Control

This bit is active only when the Pulse Accumulator A is enabled (PAEN = 1).

For PAMOD bit = 0 (event counter mode).

1 = rising edges on PT7 pin cause the count to be incremented
0 = falling edges on PT7 pin cause the count to be incremented

For PAMOD bit = 1 (gated time accumulation mode).

1 = PT7 input pin low enables bus clock divided by 64 clock to Pulse Accumulator and the trailing rising edge on PT7 sets the PAIF flag
0 = PT7 input pin high enables bus clock divided by 64 clock to Pulse Accumulator and the trailing falling edge on PT7 sets the PAIF flag.

Table 68 Pin Action

PAMOD	PEDGE	Pin Action
0	0	Falling edge
0	1	Rising edge
1	0	Div. by 64 clock enabled with pin high level
1	1	Div. by 64 clock enabled with pin low level

If the timer is not active (TEN = 0 in TSCR1), there is no divide-by-64 since the ÷64 clock is generated by the timer prescaler.

CLK1, CLK0 — Clock Select Bits

Table 69 Clock Selection

CLK1	CLK0	Clock Source
0	0	Use timer prescaler clock as timer counter clock
0	1	Use PACLK as input to timer counter clock
1	0	Use PACLK/256 as timer counter clock frequency
1	1	Use PACLK/65536 as timer counter clock frequency

For the description of PACLK please refer to Figure 72 in page 412.

Freescale Semiconductor, Inc.

If the pulse accumulator is disabled (PAEN = 0), the prescaler clock from the timer is always used as an input clock to the timer counter. The change from one selected clock to the other happens immediately after these bits are written.

PAOVI — Pulse Accumulator A Overflow Interrupt enable
1 = interrupt requested if PAOVF is set
0 = interrupt inhibited

PAI — Pulse Accumulator Input Interrupt enable
1 = interrupt requested if PAIF is set
0 = interrupt inhibited

Pulse Accumulator A Flag Register (PAFLG)

Register offset: $0061

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	PAOVF	PAIF
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read or write anytime. When the TFFCA bit in the TSCR1 register (see page 382) is set, any access to the PACNT register will clear all the flags in the PAFLG register.

PAOVF — Pulse Accumulator A Overflow Flag

Set when the 16-bit pulse accumulator A overflows from $FFFF to $0000,or when 8-bit pulse accumulator 3 (PAC3) overflows from $FF to $00.

When PACMX = 1, PAOVF bit can also be set if 8-bit pulse accumulator 3 (PAC3) reaches $FF followed by an active edge on PT3.

This bit is cleared automatically by a write to the PAFLG register with bit 1 set.

Freescale Semiconductor, Inc.

PAIF — Pulse Accumulator Input edge Flag

Set when the selected edge is detected at the PT7 input pin. In event mode the event edge triggers PAIF and in gated time accumulation mode the trailing edge of the gate signal at the PT7 input pin triggers PAIF.

This bit is cleared by a write to the PAFLG register with bit 0 set. Any access to the PACN3, PACN2 registers will clear all the flags in this register when TFFCA bit in register TSCR1 is set.

Pulse Accumulators Count Registers (PACN3, PACN2)

Register offset: $0062–$0063

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BIt 7	6	5	4	3	2	1	Bit 0
Reset:	0	0	0	0	0	0	0	0

Read or write any time.

The two 8-bit pulse accumulators PAC3 and PAC2 are cascaded to form the PACA 16-bit pulse accumulator. When PACA in enabled (PAEN=1 in PACTL) the PACN3 and PACN2 registers contents are respectively the high and low byte of the PACA.

When PACN3 overflows from $FF to $00, the Interrupt flag PAOVF in PAFLG is set.

Full count register access should take place in one clock cycle. A separate read/write for high byte and low byte will give a different result than accessing them as a word.

NOTE: *The input capture edge circuits of 8-bit pulse accumulators are configured with control bits EDGnA and EDGnB in the TCTL4 register (see page 386).*

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

Freescale Semiconductor, Inc.

NOTE: *When clocking pulse and write to the registers occurs simultaneously, write takes priority and the register is not incremented.*

Pulse Accumulators Count Registers (PACN1, PACN0)

Register offset: $0064–$0065

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	0	0	0	0	0	0	0	0

Read or write any time.

The two 8-bit pulse accumulators PAC1 and PAC0 are cascaded to form the PACB 16-bit pulse accumulator. When PACB is enabled, (PBEN=1 in PBCTL) the PACN1 and PACN0 registers contents are respectively the high and low byte of the PACB.

When PACN1 overflows from $FF to $00, the Interrupt flag PBOVF in PBFLG (see page 405) is set.

Full count register access should take place in one clock cycle. A separate read/write for high byte and low byte will give a different result than accessing them as a word.

NOTE: *The input capture edge circuits of 8-bit pulse accumulators are configured with control bits EDGnA EDGnB in the TCTL4 register (see page 386).*

NOTE: *When clocking pulse and write to the registers occurs simultaneously, write takes priority and the register is not incremented.*

Freescale Semiconductor, Inc.

16–Bit Modulus Down-Counter Control Register (MCCTL)

Register offset: $0066

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MCZI	MODMC	RDMCL	0	0	MCEN	MCPR1	MCPR0
Write:				ICLAT	FLMC			
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read or write any time except for bit 4.

MCZI — Modulus Counter Underflow Interrupt Enable

1 = Modulus counter interrupt is enabled.

0 = Modulus counter interrupt is disabled.

MODMC — Modulus Mode Enable

1 = Modulus mode is enabled. When the counter reaches $0000, the counter is loaded with the latest value written to the modulus count register.

0 = The counter counts once from the value written to it and will stop at $0000.

NOTE: *For proper operation, the MCEN bit should be cleared before modifying the MODMC bit in order to reset the modulus counter to $FFFF.*

RDMCL — Read Modulus Down-Counter Load

1 = Reads of the modulus count register will return the contents of the load register.

0 = Reads of the modulus count register will return the present value of the count register.

ICLAT — Input Capture Force Latch Action

When input capture latch mode is enabled - LATQ and BUFEN bit in ICSYS (see page 401) are set - a write one to this bit immediately forces the contents of the input capture registers TC0 to TC3 and their

Freescale Semiconductor, Inc.

corresponding 8-bit pulse accumulators to be latched into the associated holding registers. The pulse accumulators will be automatically cleared when the latch action occurs.

Writing zero to this bit has no effect. Read of this bit will return always zero.

FLMC — Force Load Register into the Modulus Counter Count Register

This bit is active only when the modulus down-counter is enabled (MCEN=1).

A write one into this bit loads the load register into the modulus counter count register. This also resets the modulus counter prescaler.

Write zero to this bit has no effect.

When MODMC=0, counter starts counting and stops at $0000.

Read of this bit will return always zero.

MCEN — Modulus Down-Counter Enable

1 = Modulus counter is enabled.
0 = Modulus counter disabled.

When MCEN=0, the counter is preset to $FFFF. This will prevent an early interrupt flag when the modulus down-counter is enabled.

MCPR1, MCPR0 — Modulus Counter Prescaler select

These two bits specify the division rate of the modulus counter prescaler.

The newly selected prescaler division rate will not be effective until a load of the load register into the modulus counter count register occurs.

Table 70 Modulus Counter Prescaler Select

MCPR1	MCPR0	Prescaler division rate
0	0	1
0	1	4
1	0	8
1	1	16

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

Freescale Semiconductor, Inc.

16-Bit Modulus Down-Counter FLAG Register (MCFLG)

Register offset: $0067

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MCZF	0	0	0	POLF3	POLF2	POLF1	POLF0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: any time
Write: Only for clearing bit 7

MCZF — Modulus Counter Underflow Flag

The flag is set when the modulus down-counter reaches $0000.

A write one to this bit clears the flag. Write zero has no effect.

Any access to the MCCNT register will clear the MCZF flag in this register when TFFCA bit in register TSCR1 (see page 382) is set.

POLF3–POLF0 — First Input Capture Polarity Status

This are read only bits. Write to these bits has no effect.

Each status bit gives the polarity of the first edge which has caused an input capture to occur after capture latch has been read.

Each POLFx corresponds to a timer PORTx input.

1 = The first input capture has been caused by a rising edge.
0 = The first input capture has been caused by a falling edge.

Freescale Semiconductor, Inc.

Input Control Pulse Accumulators Register (ICPAR)

Register offset: $0068

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	PA3EN	PA2EN	PA1EN	PA0EN
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

The 8-bit pulse accumulators PAC3 and PAC2 can be enabled only if PAEN in PACTL is cleared. If PAEN is set, PA3EN and PA2EN have no effect.

The 8-bit pulse accumulators PAC1 and PAC0 can be enabled only if PBEN in PBCTL is cleared. If PBEN is set, PA1EN and PA0EN have no effect.

Read or write any time.

PAxEN — 8-Bit Pulse Accumulator ‘x’ Enable
1 = 8-Bit Pulse Accumulator is enabled.
0 = 8-Bit Pulse Accumulator is disabled.

NOTE: *The input capture edge circuits of 8-bit pulse accumulators are configured with control bits EDGnA and EDGnB in the TCTL4 register (see page 386).*

Delay Counter Control Register (DLYCT)

Register offset: $0069

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	DLY1	DLY0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Freescale Semiconductor, Inc.

Read or write any time.

If enabled, after detection of a valid edge on input capture pin, the delay counter counts the pre-selected number of bus clock cycles, then it will generate a pulse on its output. The pulse is generated only if the level of input signal, after the preset delay, is the opposite of the level before the transition.This will avoid reaction to narrow input pulses.

After counting, the counter will be cleared automatically.

Delay between two active edges of the input signal period should be longer than the selected counter delay.

DLYx — Delay Counter Select

Table 71 Delay Counter Select

DLY1	DLY0	Delay
0	0	Disabled (bypassed)
0	1	256 bus clock cycles
1	0	512 bus clock cycles
1	1	1024 bus clock cycles

Input Control Overwrite Register (ICOVW)

Register offset: $006A

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	NOVW7	NOVW6	NOVW5	NOVW4	NOVW3	NOVW2	NOVW1	NOVW0
Reset:	0	0	0	0	0	0	0	0

Read or write any time.

Freescale Semiconductor, Inc.

NOVWx — No Input Capture Overwrite

1 = The related capture register or holding register cannot be written by an event unless they are empty (see IC Channels). This will prevent the captured value to be overwritten until it is read or latched in the holding register.

0 = The contents of the related capture register or holding register can be overwritten when a new input capture or latch occurs.

NOTE: An IC register is empty when it has been read or latched into the holding register.

NOTE: A holding register is empty when it has been read.

Input Control System Control Register (ICSYS)

Register offset: $006B

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SH37	SH26	SH15	SH04	TFMOD	PACMX	BUFEN	LATQ
Reset:	0	0	0	0	0	0	0	0

Read: any time

Write: May be written once in normal modes. Writes are always permitted when special modes.

SHxy — Share Input action of Input Capture Channels x and y

1 = The channel input ‘x’ causes the same action on the channel ‘y’. The port pin ‘x’ and the corresponding edge detector is used to be active on the channel ‘y’.

0 = Normal operation

TFMOD — Timer Flag-setting Mode

Use of the TFMOD bit in the ICSYS register in conjunction with the use of the ICOVW register allows a timer interrupt to be generated after capturing two values in the capture and holding registers instead of generating an interrupt for every capture.

Freescale Semiconductor, Inc.

By setting TFMOD in queue mode, when NOVW bit is set and the corresponding capture and holding registers are emptied, an input capture event will first update the related input capture register with the main timer contents. At the next event the TCn data is transferred to the TCnH register, The TCn is updated and the CnF interrupt flag is set. See Figure 73 in page 413.

In all other input capture cases the interrupt flag is set by a valid external event on PTn.

1 = If in queue mode (BUFEN=1 and LATQ=0), the timer flags C3F–C0F in TFLG1 are set only when a latch on the corresponding holding register occurs.
If the queue mode is not engaged, the timer flags C3F–C0F are set the same way as for TFMOD=0.

0 = The timer flags C3F–C0F in TFLG1 are set when a valid input capture transition on the corresponding port pin occurs.

PACMX — 8-Bit Pulse Accumulators Maximum Count

1 = When the 8-bit pulse accumulator has reached the value $FF, it will not be incremented further. The value $FF indicates a count of 255 or more.

0 = Normal operation. When the 8-bit pulse accumulator has reached the value $FF, with the next active edge, it will be incremented to $00.

BUFEN — IC Buffer Enable

1 = Input Capture and pulse accumulator holding registers are enabled. The latching mode is defined by LATQ control bit. Write one into ICLAT bit in MCCTL (see page 396), when LATQ is set will produce latching of input capture and pulse accumulators registers into their holding registers.

0 = Input Capture and pulse accumulator holding registers are disabled.

LATQ — Input Control Latch or Queue Mode Enable

The BUFEN control bit should be set in order to enable the IC and pulse accumulators holding registers. Otherwise LATQ latching modes are disabled.

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

Freescale Semiconductor, Inc.

Write one into ICLAT bit in MCCTL, when LATQ and BUFEN are set will produce latching of input capture and pulse accumulators registers into their holding registers.

1 = Latch Mode is enabled. Latching function occurs when modulus down-counter reaches zero or a zero is written into the count register MCCNT (see Buffered IC Channels).
With a latching event the contents of IC registers and 8-bit pulse accumulators are transferred to their holding registers.
8-bit pulse accumulators are cleared.

0 = Queue Mode of Input Capture is enabled.
The main timer value is memorized in the IC register by a valid input pin transition.
With a new occurrence of a capture, the value of the IC register will be transferred to its holding register and the IC register memorizes the new timer value.

Freescale Semiconductor, Inc.

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

16-Bit Pulse Accumulator B Control Register (PBCTL)

Register offset: $0070

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PBEN	0	0	0	0	PBOVI	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read or write any time.

16-Bit Pulse Accumulator B (PACB) is formed by cascading the 8-bit pulse accumulators PAC1 and PAC0.

When PBEN is set, the PACB is enabled. The PACB shares the input pin with IC0.

PBEN — Pulse Accumulator B System Enable

1 = Pulse Accumulator B system enabled. The two 8-bit pulse accumulators PAC1 and PAC0 are cascaded to form the PACB 16-bit pulse accumulator. When PACB in enabled, the PACN1 and PACN0 registers contents are respectively the high and low byte of the PACB.
PA1EN and PA0EN control bits in ICPAR have no effect.

0 = 16-bit Pulse Accumulator system disabled. 8-bit PAC1 and PAC0 can be enabled when their related enable bits in ICPAR are set.

PBEN is independent from TEN in TSCR1 (see page 382). With timer disabled, the pulse accumulator can still function unless pulse accumulator is disabled.

PBOVI — Pulse Accumulator B Overflow Interrupt enable

1 = interrupt requested if PBOVF in PBFLG is set

0 = interrupt inhibited

Freescale Semiconductor, Inc.

NOTE: *The input capture edge circuits of the 16-bit pulse accumulator PACB are configured with control bits EDG0B and EDG0A in the TCTL4 register (see page 386).*

Pulse Accumulator B Flag Register (PBFLG)

Register offset: $0071

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	PBOVF	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read or write any time.

PBOVF — Pulse Accumulator B Overflow Flag

This bit is set when the 16-bit pulse accumulator B overflows from $FFFF to $0000, or when 8-bit pulse accumulator 1 (PAC1) overflows from $FF to $00.

This bit is cleared by a write to the PBFLG register with bit 1 set.

Any access to the PACN1 and PACN0 registers will clear the PBOVF flag in this register when TFFCA bit in register TSCR1 (see page 382) is set.

When PACMX=1 in ICSYS (see page 401), PBOVF bit can also be set if 8-bit pulse accumulator 1 (PAC1) reaches $FF and followed an active edge comes on PT1.

Freescale Semiconductor, Inc.

8-Bit Pulse Accumulators Holding Registers (PA3H–PA0H)

Register offset: $0072–$0075

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: any time
Write: has no effect.

These registers are used to latch the value of the corresponding pulse accumulator when the related bits in register ICPAR are enabled (see Pulse Accumulators).

Modulus Down-Counter Count Register (MCCNT)

Register offset: $0076–$0077

	Bit 15	14	13	12	11	10	9	8
Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	1	1	1	1	1	1	1	1

= Reserved or unimplemented

Read or write any time.

A full access for the counter register should take place in one clock cycle. A separate read/write for high byte and low byte will give different result than accessing them as a word.

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

Freescale Semiconductor, Inc.

If the RDMCL bit in MCCTL register (see page 396) is cleared, reads of the MCCNT register will return the present value of the count register. If the RDMCL bit is set, reads of the MCCNT will return the contents of the load register.

If a $0000 is written into MCCNT and modulus counter while LATQ and BUFEN in ICSYS register are set, the input capture and pulse accumulator registers will be latched.

With a $0000 write to the MCCNT, the modulus counter will stay at zero and does not set the MCZF flag in MCFLG register.

If modulus mode is enabled (MODMC=1), a write to this address will update the load register with the value written to it. The count register will not be updated with the new value until the next counter underflow.

The FLMC bit in MCCTL can be used to immediately update the count register with the new value if an immediate load is desired.

If modulus mode is not enabled (MODMC=0), a write to this address will clear the prescaler and will immediately update the counter register with the value written to it and down-counts once to $0000

Timer Input Capture Holding Registers 0–3 (TC0H–TC3H)

Register offset: $0078–$007F

	Bit 15	14	13	12	11	10	9	8
Read:	Bit 15	14	13	12	11	10	9	Bit 8
Write:								
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

Read: any time
Write: has no effect.

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

These registers are used to latch the value of the input capture registers TC0–TC3. The corresponding IOSx bits in TIOS should be cleared (see IC Channels).

Functional Description

The Enhanced Capture Timer has 8 Input Capture, Output Compare (IC/OC) channels same as on the HC12 standard timer (timer channels TC0 to TC7). When channels are selected as input capture by selecting the IOSx bit in TIOS register, they are called Input Capture (IC) channels. Figures 69 (page 409) and 70 (page 410) show the Timer Block Diagram for Latch and Queue modes, respectively.

Four IC channels are the same as on the standard timer with one capture register each which memorizes the timer value captured by an action on the associated input pin.

Four other IC channels, in addition to the capture register, have also one buffer each called holding register. This permits to memorize two different timer values without generation of any interrupt.

Four 8-bit pulse accumulators are associated with the four buffered IC channels. Each pulse accumulator has a holding register to memorize their value by an action on its external input. Each pair of pulse accumulators can be used as a 16-bit pulse accumulator. See Figures 71 (page 411) and 72 (page 412) for Block Diagrams of these accumulators.

The 16-bit modulus down-counter can control the transfer of the IC registers contents and the pulse accumulators to the respective holding registers for a given period, every time the count reaches zero.

The modulus down-counter can also be used as a stand-alone time base with periodic interrupt capability.

Freescale Semiconductor, Inc.

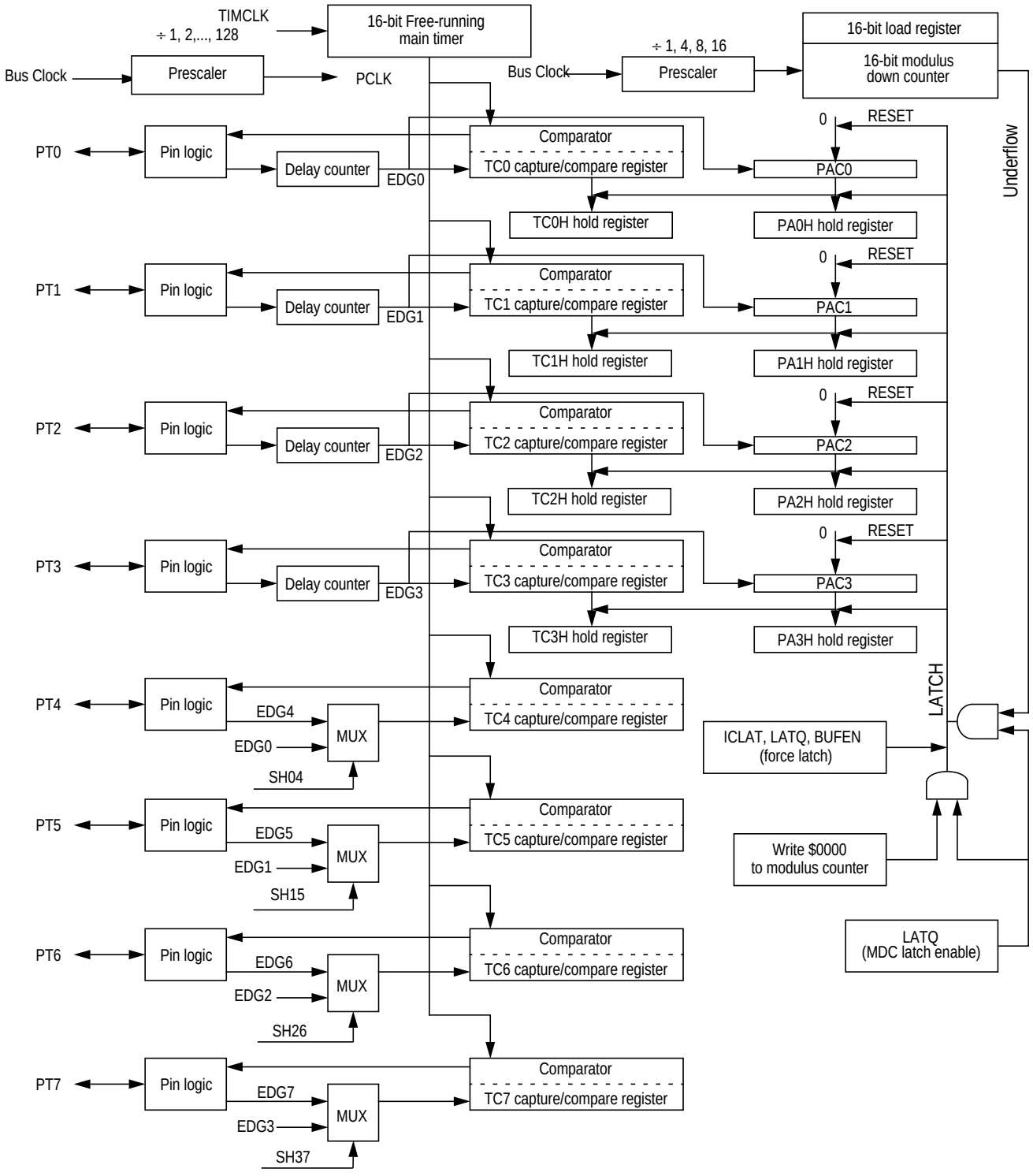


Figure 69 Timer Block Diagram in Latch Mode

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

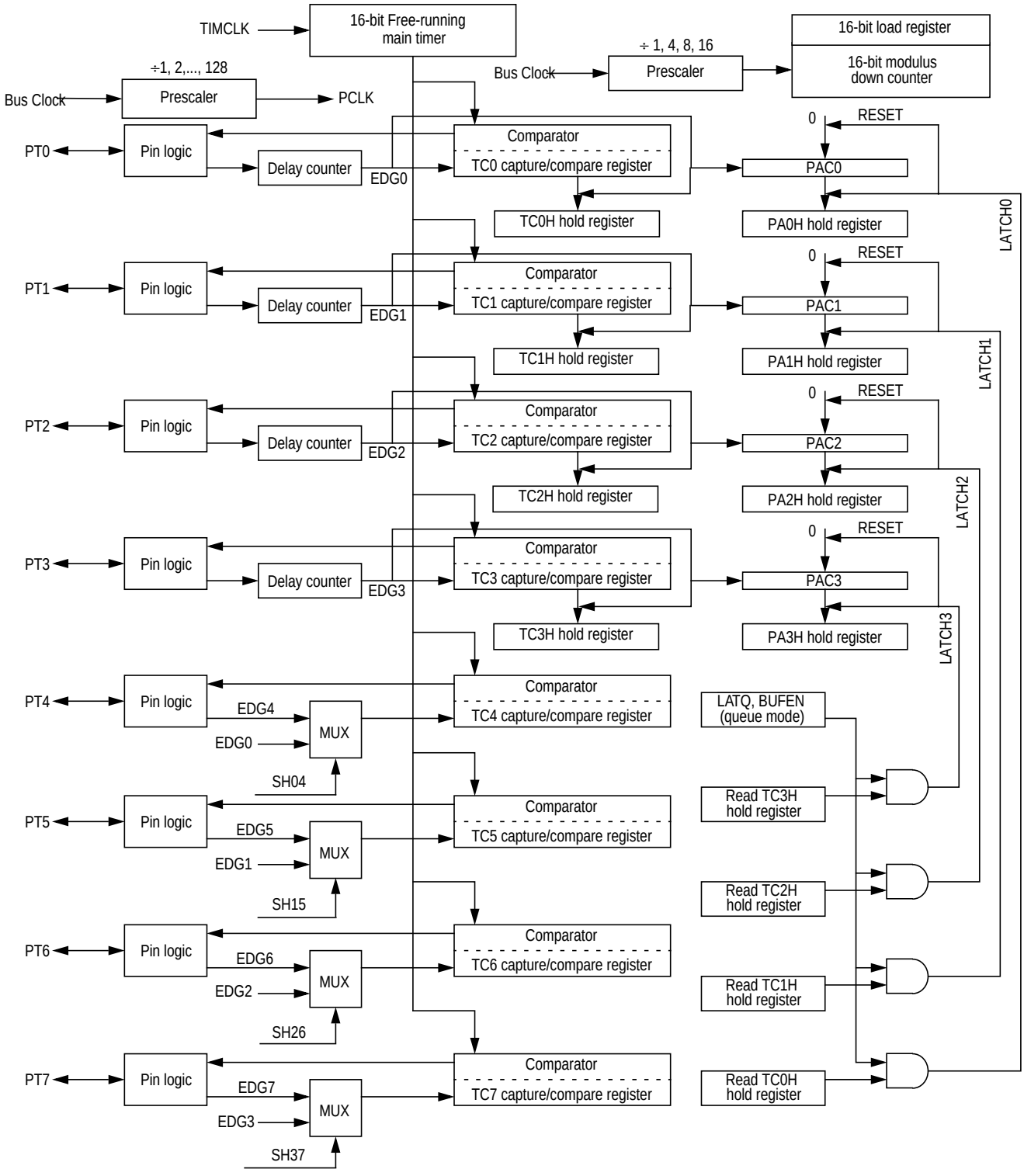


Figure 70 Timer Block Diagram in Queue Mode

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

Freescale Semiconductor, Inc.

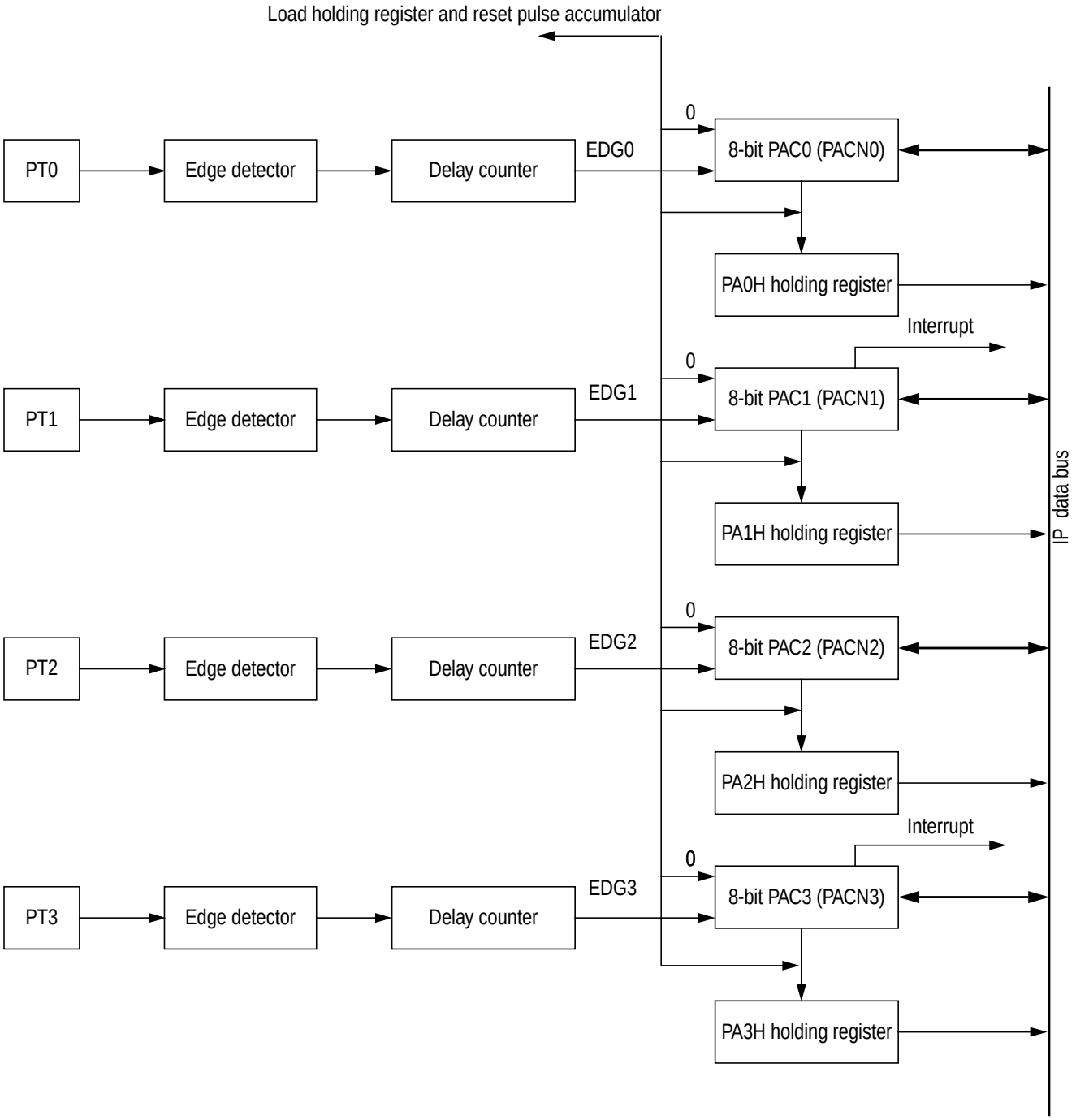


Figure 71 8-Bit Pulse Accumulators Block Diagram

Freescale Semiconductor, Inc.

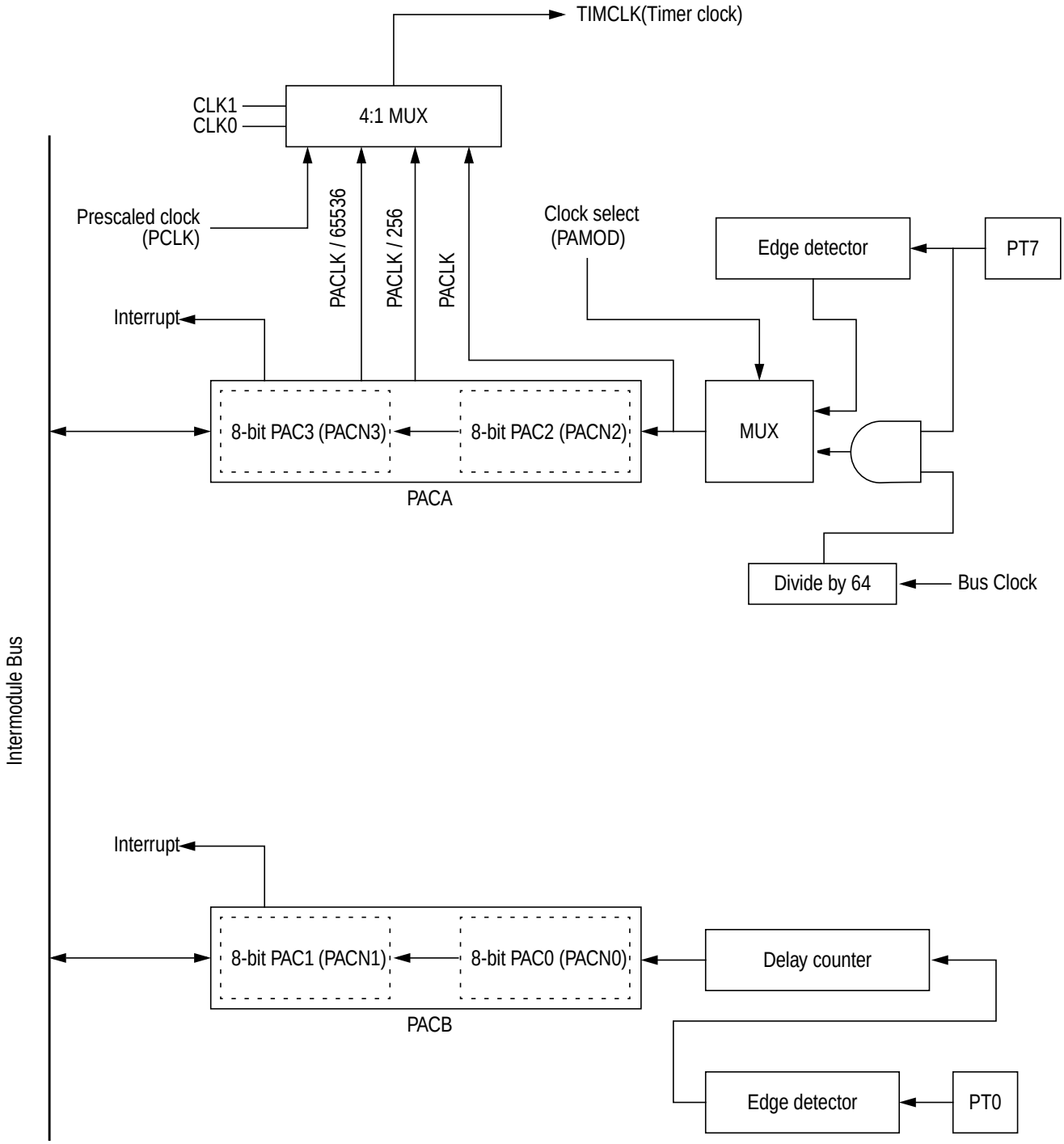


Figure 72 16-Bit Pulse Accumulators Block Diagram

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

Freescale Semiconductor, Inc.

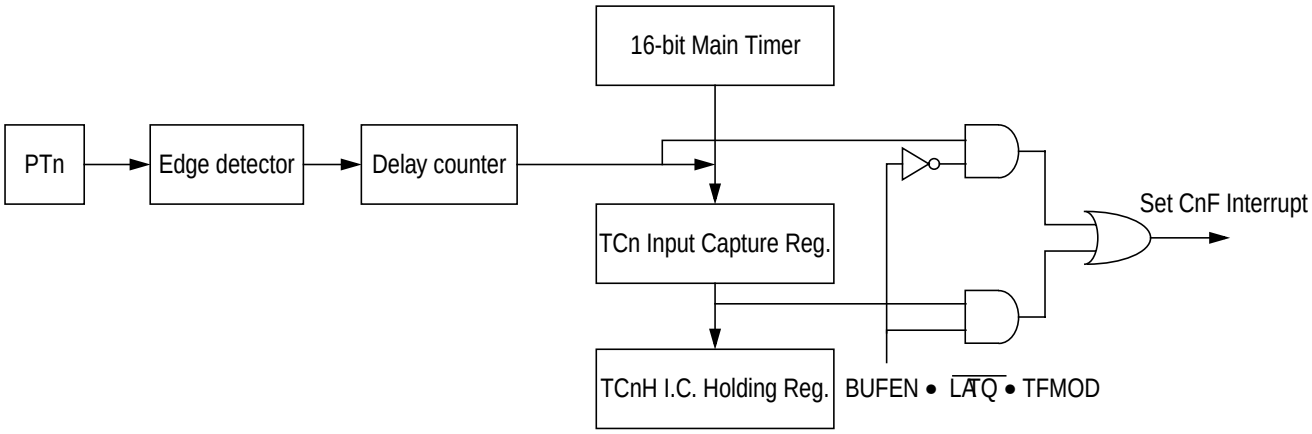


Figure 73 Interrupt Flag Setting

IC Channels

The IC channels are composed of four standard IC registers and four buffered IC channels.

An **IC register** is **empty** when it has been read or latched into the holding register.

A **holding register** is **empty** when it has been read.

Non-Buffered IC Channels

The main timer value is memorized in the IC register by a valid input pin transition.

If the corresponding NOVWx bit of the ICOVW register (see page 400) is cleared, with a new occurrence of a capture, the contents of IC register are overwritten by the new value.

If the corresponding NOVWx bit of the ICOVW register is set, the capture register cannot be written unless it is empty.

This will prevent the captured value to be overwritten until it is read.

Buffered IC Channels

There are two modes of operations for the buffered IC channels.

- IC Latch Mode:

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

Freescale Semiconductor, Inc.

When enabled (LATQ=1), the main timer value is memorized in the IC register by a valid input pin transition.

The value of the buffered IC register is latched to its holding register by the Modulus counter for a given period when the count reaches zero, by a write $0000 to the modulus counter or by a write to ICLAT in the MCCTL register. See Figure 69 in page 409.

If the corresponding NOVWx bit of the ICOVW register is cleared, with a new occurrence of a capture, the contents of IC register are overwritten by the new value. In case of latching, the contents of its holding register are overwritten.

If the corresponding NOVWx bit of the ICOVW register is set, the capture register or its holding register cannot be written by an event unless they are empty (see IC Channels). This will prevent the captured value to be overwritten until it is read or latched in the holding register.

- IC queue mode:

When enabled (LATQ=0), the main timer value is memorized in the IC register by a valid input pin transition. See Figure 70 in page 410.

If the corresponding NOVWx bit of the ICOVW register is cleared, with a new occurrence of a capture, the value of the IC register will be transferred to its holding register and the IC register memorizes the new timer value.

If the corresponding NOVWx bit of the ICOVW register is set, the capture register or its holding register cannot be written by an event unless they are empty (see IC Channels).

In queue mode, reads of holding register will latch the corresponding pulse accumulator value to its holding register.

Pulse Accumulators

There are four 8-bit pulse accumulators with four 8-bit holding registers associated with the four IC buffered channels. A pulse accumulator counts the number of active edges at the input of its channel.

The active edge can be programmed through the TCTL4 register (for the 8-bit pulse accumulators and the 16-bit pulse accumulator PACB) and

Freescale Semiconductor, Inc.

the PEDGE bit in the PACTL register (for the 16-bit pulse accumulator PACA).

The user can prevent 8-bit pulse accumulators counting further than $FF by PACMX control bit in ICSYS. In this case a value of $FF means that 255 counts or more have occurred.

Each pair of pulse accumulators can be used as a 16-bit pulse accumulator.

See Figures 71 (page 411) and 72 (page 412) for Block Diagrams of 8-bit and 16-bit accumulators.

There are two modes of operation for the pulse accumulators.

Pulse Accumulator latch mode

The value of the pulse accumulator is transferred to its holding register when the modulus down-counter reaches zero, a write $0000 to the modulus counter or when the force latch control bit ICLAT is written.

At the same time the pulse accumulator is cleared.

Pulse Accumulator queue mode

When queue mode is enabled, reads of an input capture holding register will transfer the contents of the associated pulse accumulator to its holding register.

At the same time the pulse accumulator is cleared.

Modulus Down-Counter

The modulus down-counter can be used as a time base to generate a periodic interrupt. It can also be used to latch the values of the IC registers and the pulse accumulators to their holding registers.

The action of latching can be programmed to be periodic or only once.

Freescale Semiconductor, Inc.

Interrupts

This section describes interrupts originated by the ECT block.The MCU must service the interrupt requests. Table 72 lists the interrupts generated by the ECT to communicate with the MCU.

Table 72 ECT Interrupts

Interrupt	Vector Address	Source	Description
ECTCH0	$FFEE, $FFEF	Timer channel 0	Active high timer channel interrupts 0
ECTCH1	$FFEC, $FFED	Timer channel 1	Active high timer channel interrupts 1
ECTCH2	$FFEA, $FFEB	Timer channel 2	Active high timer channel interrupts 2
ECTCH3	$FFE8, $FFE9	Timer channel 3	Active high timer channel interrupts 3
ECTCH4	$FFE6, $FFE7	Timer channel 4	Active high timer channel interrupts 4
ECTCH5	$FFE4, $FFE5	Timer channel 5	Active high timer channel interrupts 5
ECTCH6	$FFE2, $FFE3	Timer channel 6	Active high timer channel interrupts 6
ECTCH7	$FFE0, $FFE1	Timer channel 7	Active high timer channel interrupts 7
ECTMCUFI	$FFCA, $FFCB	Modulus down counter underflow	Active high modulus counter interrupt
ECTPABO	$FFC8, $FFC9	Pulse Accumulator B overflow	Active high pulse accumulator B interrupt
ECTPAAI	$FFDA, $FFDB	Pulse Accumulator A input edge	Active high pulse accumulator A input interrupt
ECTPAAO	$FFDC, $FFDD	Pulse Accumulator A overflow	Pulse accumulator overflow interrupt
ECTOVI	$FFDE, $FFDF	Timer overflow	Timer Overflow interrupt

Description of Interrupt Operation

The ECT only originates interrupt requests. The following is a description of how the ECT makes a request and how the MCU should acknowledge that request.

Channel [7:0] Interrupt (ECTCH(0-7))

This active high output will be asserted by the module to request a timer channel 7–0 interrupt to be serviced by the system controller.

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

Freescale Semiconductor, Inc.

Modulus Down Counter Underflow Interrupt (ECTMCUFI)	This active high output will be asserted by the module to request a modulus counter underflow interrupt to be serviced by the system controller.
Pulse Accumulator B Overflow Interrupt (ECTPABO)	This active high output will be asserted by the module to request a timer pulse accumulator B overflow interrupt to be serviced by the system controller.
Pulse Accumulator A Input Interrupt (ECTPAAI)	This active high output will be asserted by the module to request a timer pulse accumulator A input interrupt to be serviced by the system controller.
Pulse Accumulator A Overflow Interrupt (ECTPAAO)	This active high output will be asserted by the module to request a timer pulse accumulator A overflow interrupt to be serviced by the system controller.
Timer Overflow Interrupt (ECTOVI)	This active high output will be asserted by the module to request a timer overflow interrupt to be serviced by the system controller.

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

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

Serial Communications Interface (SCI)

Contents

Overview 419
Features 419
Modes of Operation 420
Block Diagram 421
External Pin Descriptions 422
Register Map 423
Register Descriptions 424
Functional Description 434
Interrupts 454

Overview

The SCI allows asynchronous serial communications with peripheral devices and other CPUs.

Features

The SCI includes these distinctive features:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- 13-bit baud rate selection
- Programmable 8-bit or 9-bit data format
- Separately enabled transmitter and receiver
- Programmable transmitter output parity

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

Freescale Semiconductor, Inc.

- Two receiver wake-up methods:
  - Idle line wake-up
  - Address mark wake-up
- Interrupt-driven operation with eight flags:
  - Transmitter empty
  - Transmission complete
  - Receiver full
  - Idle receiver input
  - Receiver overrun
  - Noise error
  - Framing error
  - Parity error
- Receiver framing error detection
- Hardware parity checking
- 10/13 bit software selectable break character length
- 1/16 bit-time noise detection

Modes of Operation

The SCI functions the same in normal, special, and emulation modes. It has two low power modes, wait and stop modes.

Run Mode

Normal mode of operation.

Wait Mode

SCI operation in wait mode depends on the state of the SCISWAI bit in the SCI control register 1 (SCIxCR1).

- If SCISWAI is clear, the SCI operates normally when the CPU is in wait mode.

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

Freescale Semiconductor, Inc.

- If SCISWAI is set, SCI clock generation ceases and the SCI module enters a power-conservation state when the CPU is in wait mode. Setting SCISWAI does not affect the state of the receiver enable bit, RE, or the transmitter enable bit, TE.

  If SCISWAI is set, any transmission or reception in progress stops at wait mode entry. The transmission or reception resumes when either an internal or external interrupt brings the CPU out of wait mode. Exiting wait mode by reset aborts any transmission or reception in progress and resets the SCI.

Stop Mode

The SCI is inactive during stop mode for reduced power consumption. The STOP instruction does not affect the SCI register states, but the bus clock will be disabled. The SCI operation resumes from where it left off after an external interrupt brings the CPU out of stop mode. Exiting stop mode by reset aborts any transmission or reception in progress and resets the SCI.

Block Diagram

Figure 74 is a block diagram of the SCI.

Freescale Semiconductor, Inc.

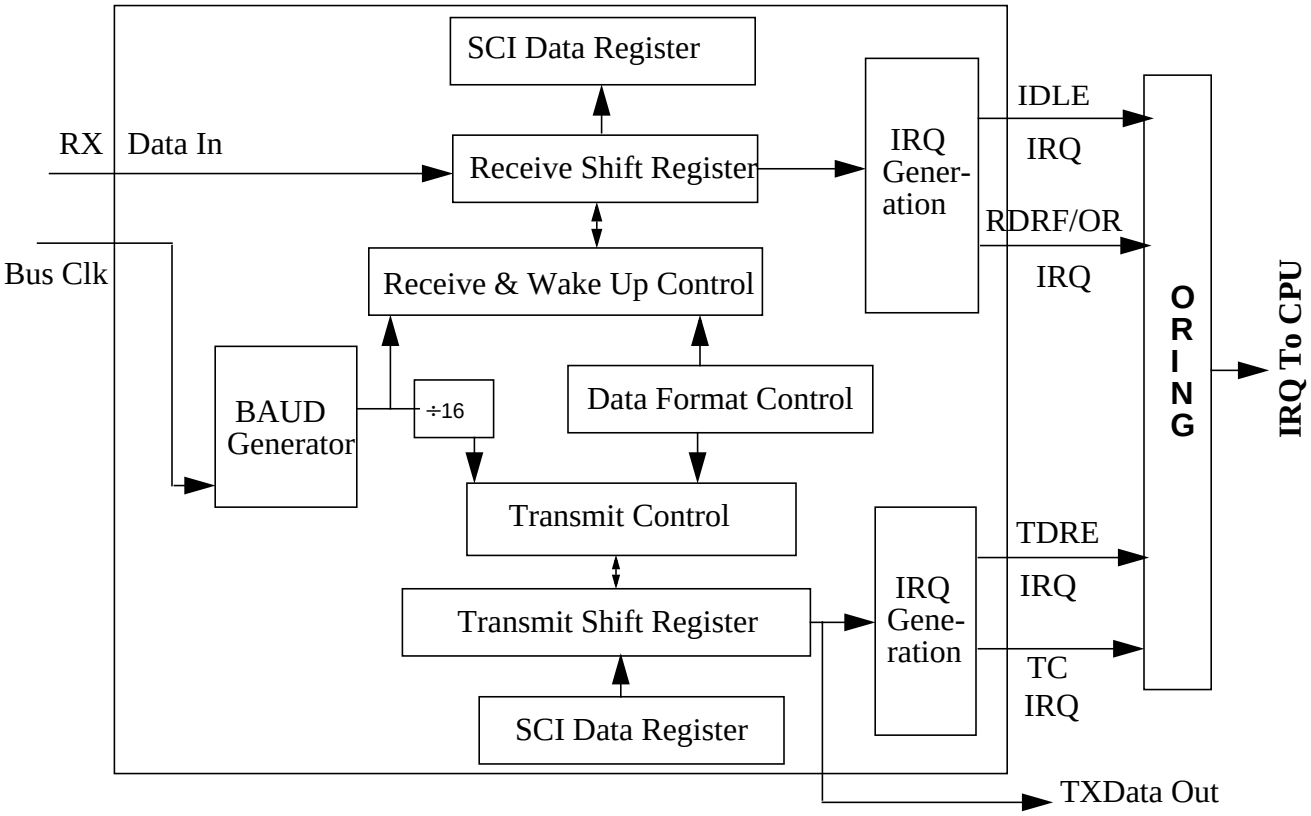


Figure 74 SCI Block Diagram

External Pin Descriptions

RXD (PS0, PS2) SCI receive pin. This pin serves as receive data input of SCI. When the SCI operates in single wire mode (LOOPS=1 in SCRxCR1 register), this channel is unused and corresponding MCU pins (PS0, PS2) can be used as general purpose I/O.

TXD (PS1, PS3) SCI transmit pin. This pin serves as transmit data output of SCI.

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

Freescale Semiconductor, Inc.

Register Map

Register name (1)		Bit 7	6	5	4	3	2	1	Bit 0	Address offset SCI0	Address offset SCI1
SCIxBDH	Read:	0	0	0	SBR12	SBR11	SBR10	SBR9	SBR8	$00C8	$00D0
	Write:										
SCIxBDL	Read:	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	$00C9	$00D1
	Write:										
SCIxCR1	Read:	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT	$00CA	$00D2
	Write:										
SCIxCR2	Read:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	$00CB	$00D3
	Write:										
SCIxSR1	Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	$00CC	$00D4
	Write:										
SCIxSR2	Read:	0	0	0	0	0	BRK13	TXDIR	RAF	$00CD	$00D5
	Write:										
SCIxDRH	Read:	R8	T8	0	0	0	0	0	0	$00CE	$00D6
	Write:										
SCIxDRL	Read:	R7	R6	R5	R4	R3	R2	R1	R0	$00CF	$00D7
	Write:	T7	T6	T5	T4	T3	T2	T1	T0		

= Reserved or unimplemented

Figure 75 SCI Register Quick Reference

1. Where: x = 0 for SCI0, or 1 for SCI1

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Freescale Semiconductor, Inc.

Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Writes to a reserved register location do not have any effect and reads of these locations return a zero. Details of register bit and field function follow the register diagrams, in bit order.

SCI Baud Rate Registers (SCIxBDH/L)

Where: x = 0 for SCI0, or 1 for SCI1

Address Offset: $00C8 (SCI0BDH), $00D0 (SCI1BDH)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	SBR12	SBR11	SBR10	SBR9	SBR8
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Address Offset: $00C9 (SCI0BDL), $00D1 (SCI1BDL)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
Write:								
Reset:	0	0	0	0	0	1	0	0

Read: anytime. If only SCIxBDH is written to, a read will not return the correct data until SCIxBDL is written to as well, following a write to SCIxBDH.

Write: anytime. Writing to SCIxBDH has no effect without writing to SCIxBDL, since writing to SCIxBDH puts the data in a temporary location until SCIxBDL is written to.

The SCI Baud Rate Register is used by the counter to determine the baud rate of the SCI. The formula for calculating the baud rate is:

SCI baud rate = Bus Clock / (16 x BR),

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

Freescale Semiconductor, Inc.

where BR is the content of the SCI baud rate registers, bits SBR12 through SBR0. The baud rate registers can contain a value from 1 to 8191.

SBR12–SBR0 — SCI Baud Rate Bits

The baud rate for the SCI is determined by these 13 bits.

NOTE: *The baud rate generator is disabled until TE or RE bits in SCIxCR2 are set for the first time after reset. The baud rate generator is disabled when BR = 0.*

SCI Control Register 1 (SCIxCR1)

Where: x = 0 for SCI0, or 1 for SCI1

Address Offset: $00CA (SCI0CR1), $00D2 (SCI1CR1)

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Read: anytime
Write: anytime

LOOPS — Loop Select Bit

LOOPS enables loop operation. In loop operation, the RxD pin is disconnected from the SCI and the transmitter output is internally connected to the receiver input. Both the transmitter and the receiver must be enabled to use the loop function.

1 = Loop operation enabled
0 = Normal operation enabled

The receiver input is determined by the RSRC bit.

SCISWAI — SCI Stop in Wait Mode Bit

SCISWAI disables the SCI in wait mode.

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

Freescale Semiconductor, Inc.

1 = SCI disabled in wait mode
0 = SCI enabled in wait mode

RSRC — Receiver Source Bit

When LOOPS = 1, the RSRC bit determines the source for the receiver shift register input.

1 = Receiver input connected externally to transmitter
0 = Receiver input internally connected to transmitter output

Table 73 Loop Functions

LOOPS	RSRC	Function
0	x	Normal operation
1	0	Loop mode with Rx input internally connected to Tx output
1	1	Single-wire mode with Rx input connected to **TXD**

M — Data Format Mode Bit

MODE determines whether data characters are eight or nine bits long.

1 = One start bit, nine data bits, one stop bit
0 = One start bit, eight data bits, one stop bit

WAKE — Wakeup Condition Bit

WAKE determines which condition wakes up the SCI: a logic 1 (address mark) in the most significant bit position of a received data character or an idle condition on the **RXD**.

1 = Address mark wakeup
0 = Idle line wakeup

ILT — Idle Line Type Bit

ILT determines when the receiver starts counting logic 1s as idle character bits. The counting begins either after the start bit or after the stop bit. If the count begins after the start bit, then a string of logic 1s preceding the stop bit may cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions.

1 = Idle character bit count begins after stop bit
0 = Idle character bit count begins after start bit

Freescale Semiconductor, Inc.

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

PE — Parity Enable Bit

PE enables the parity function. When enabled, the parity function inserts a parity bit in the most significant bit position.

1 = Parity function enabled
0 = Parity function disabled

PT — Parity Type Bit

PT determines whether the SCI generates and checks for even parity or odd parity. With even parity, an even number of 1s clears the parity bit and an odd number of 1s sets the parity bit. With odd parity, an odd number of 1s clears the parity bit and an even number of 1s sets the parity bit.

1 = Odd parity
0 = Even parity

SCI Control Register 2 (SCIxCR2)

Where: x = 0 for SCI0, or 1 for SCI1

Address Offset: $00CB (SCI0CR2), $00D3 (SCI1CR2)

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Read: anytime
Write: anytime

TIE — Transmitter Interrupt Enable Bit

TIE enables the transmit data register empty flag, TDRE, to generate interrupt requests.

1 = TDRE interrupt requests enabled
0 = TDRE interrupt requests disabled

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

Freescale Semiconductor, Inc.

TCIE — Transmission Complete Interrupt Enable Bit

TCIE enables the transmission complete flag, TC, to generate interrupt requests.

1 = TC interrupt requests enabled
0 = TC interrupt requests disabled

RIE — Receiver Full Interrupt Enable Bit

RIE enables the receive data register full flag, RDRF, or the overrun flag, OR, to generate interrupt requests.

1 = RDRF and OR interrupt requests enabled
0 = RDRF and OR interrupt requests disabled

ILIE — Idle Line Interrupt Enable Bit

ILIE enables the idle line flag, IDLE, to generate interrupt requests.

1 = IDLE interrupt requests enabled
0 = IDLE interrupt requests disabled

TE — Transmitter Enable Bit

TE enables the SCI transmitter and configures the TxD pin as being controlled by the SCI. The TE bit can be used to queue an idle preamble.

1 = Transmitter enabled
0 = Transmitter disabled

RE — Receiver Enable Bit

RE enables the SCI receiver.

1 = Receiver enabled
0 = Receiver disabled

RWU — Receiver Wakeup Bit

Standby state

1 = RWU enables the wakeup function and inhibits further receiver interrupt requests. Normally, hardware wakes the receiver by automatically clearing RWU.
0 = Normal operation.

Freescale Semiconductor, Inc.

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

SBK — Send Break Bit

Toggling SBK sends one break character (10 or 11 logic 0s, respectively 13 or 14 logics 0s if BRK13 is set). Toggling implies clearing the SBK bit before the break character has finished transmitting. As long as SBK is set, the transmitter continues to send complete break characters (10 or 11, respectively 13 or 14 bits).

1 = Transmit break characters
0 = No break characters

SCI Status Register 1 (SCIxSR1)

Where: x = 0 for SCI0, or 1 for SCI1

The SCIxSR1 and SCIxSR2 register provides inputs to the MCU for generation of SCI interrupts. Also, these registers can be polled by the MCU to check the status of these bits. The flag-clearing procedures require that the status register be read followed by a read or write to the SCI Data Register. It is permissible to execute other instructions between the two steps as long as it does not compromise the handling of I/O, but the order of operations is important for flag clearing.

Address Offset: $00CC (SCI0SR1), $00D4 (SCI1SR1)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
Write:								
Reset:	1	1	0	0	0	0	0	0

= Unimplemented or Reserved

Read: anytime
Write: has no meaning or effect

TDRE — Transmit Data Register Empty Flag

TDRE is set when the transmit shift register receives a byte from the SCI data register. When TDRE is 1, the transmit data register (SCIxDRH/L) is empty and can receive a new value to transmit. Clear TDRE by reading SCI status register 1 (SCIxSR1) with TDRE set and then writing to SCI data register low (SCIxDRL).

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

Freescale Semiconductor, Inc.

1 = Byte transferred to transmit shift register; transmit data register empty
0 = No byte transferred to transmit shift register

TC — Transmit Complete Flag

TC is set low when there is a transmission in progress or when a preamble or break character is loaded. TC is set high when the TDRE flag is set and no data, preamble, or break character is being transmitted. When TC is set, the **TXD out signal** becomes idle (logic 1). Clear TC by reading SCI status register 1 (SCIxSR1) with TC set and then writing to SCI data register low (SCIxDRL). TC is cleared automatically when data, preamble, or break is queued and ready to be sent. TC is cleared in the event of a simultaneous set and clear of the TC flag (transmission not complete).

1 = No transmission in progress
0 = Transmission in progress

RDRF — Receive Data Register Full Flag

RDRF is set when the data in the receive shift register transfers to the SCI data register. Clear RDRF by reading SCI status register 1 (SCIxSR1) with RDRF set and then reading SCI data register low (SCIxDRL).

1 = Received data available in SCI data register
0 = Data not available in SCI data register

IDLE — Idle Line Flag

IDLE is set when 10 consecutive logic 1s (if M=0) or 11 consecutive logic 1s (if M=1) appear on the receiver input. Once the IDLE flag is cleared, a valid frame must again set the RDRF flag before an idle condition can set the IDLE flag.Clear IDLE by reading SCI status register 1 (SCIxSR1) with IDLE set and then reading SCI data register low (SCIxDRL).

1 = Receiver input has become idle
0 = Receiver input is either active now or has never become active since the IDLE flag was last cleared

NOTE: *When the receiver wakeup bit (RWU) is set, an idle line condition does not set the IDLE flag.*

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

Freescale Semiconductor, Inc.

OR — Overrun Flag

OR is set when software fails to read the SCI data register before the receive shift register receives the next frame. The OR bit is set immediately after the stop bit has been completely received for the second frame. The data in the shift register is lost, but the data already in the SCI data registers is not affected. Clear OR by reading SCI status register 1 (SCIxSR1) with OR set and then reading SCI data register low (SCIxDRL).

1 = Overrun
0 = No overrun

NF — Noise Flag

NF is set when the SCI detects noise on the receiver input. NF bit is set during the same cycle as the RDRF flag but does not get set in the case of an overrun. Clear NF by reading SCI status register 1 (SCIxSR1) and then reading SCI data register low (SCIxDRL).

1 = Noise
0 = No noise

FE — Framing Error Flag

FE is set when a logic 0 is accepted as the stop bit. FE bit is set during the same cycle as the RDRF flag, but does not get set in the case of an overrun. FE inhibits further data reception until it is cleared. Clear FE by reading SCI status register 1 (SCIxSR1) with FE set and then reading the SCI data register low (SCIxDRL).

1 = Framing error
0 = No framing error

PF — Parity Error Flag

PF is set when the parity enable bit, PE, is set and the parity of the received data does not match its parity bit. Clear PF by reading SCI status register 1 (SCIxSR1), and then reading SCI data register low (SCIxDRL).

1 = Parity error
0 = No parity error

Freescale Semiconductor, Inc.

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

SCI Status Register 2 (SCIxSR2)

Where: x = 0 for SCI0, or 1 for SCI1

Address Offset: $00CD (SCI0SR2), $00D5 (SCI1SR2)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	BRK13	TXDIR	RAF
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Read: anytime

Write: anytime; writing accesses SCI status register 2; writing to any bits except TXDIR and BRK13 has no effect

BRK13 — Break Transmit character length

This bit determines whether the transmit break character is 10 or 11 bit respectively 13 or 14 bits long. The detection of a framing error is not affected by this bit.

1 = Break character is 13 or 14 bit long

0 = Break Character is 10 or 11 bit long

TXDIR — Transmitter pin data direction in Single-Wire mode.

This bit determines whether the TxD pin is going to be used as an input or output, in the Single-Wire mode of operation. This bit is only relevant in the Single-Wire mode of operation.

1 = TxD pin to be used as an output in Single-Wire mode

0 = TxD pin to be used as an input in Single-Wire mode

RAF — Receiver Active Flag

RAF is set when the receiver detects a logic 0 during the RT1 time period of the start bit search. RAF is cleared when the receiver detects an idle character.

1 = Reception in progress

0 = No reception in progress

Freescale Semiconductor, Inc.

SCI Data Registers (SCIxDRH/L)

Where: x = 0 for SCI0, or 1 for SCI1

Address Offset: $00CE (SCI0DRH), $00D6 (SCI1DRH)

	Bit7	6	5	4	3	2	1	Bit 0
Read:	R8	T8	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Address Offset: $OOCF (SCI0DRL), $00D7 (SCI1DRL)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R7	R6	R5	R4	R3	R2	R1	R0
Write:	T7	T6	T5	T4	T3	T2	T1	T0
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Read: anytime; reading accesses SCI receive data register
Write: anytime; writing accesses SCI transmit data register; writing to R8 has no effect

R8 — Received Bit 8

R8 is the ninth data bit received when the SCI is configured for 9-bit data format (M = 1).

T8 — Transmit Bit 8

T8 is the ninth data bit transmitted when the SCI is configured for 9-bit data format (M = 1).

R7–R0 — Received bits seven through zero for 9-bit or 8-bit data formats

T7–T0 — Transmit bits seven through zero for 9-bit or 8-bit formats

NOTE: *If the value of T8 is the same as in the previous transmission, T8 does not have to be rewritten. The same value is transmitted until T8 is rewritten.*

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

Freescale Semiconductor, Inc.

NOTE: *In 8-bit data format, only SCI data register low (SCIxDRL) needs to be accessed.*

NOTE: *When transmitting in 9-bit data format and using 8-bit write instructions, write first to SCI data register high (SCIxDRH), then SCIxDRL.*

Functional Description

This section provides a complete functional description of the SCI block, detailing the operation of the design from the end user perspective in a number of subsections.

Figure 76 shows the structure of the SCI module. The SCI allows full duplex, asynchronous, NRZ serial communication between the CPU and remote devices, including other CPUs. The SCI transmitter and receiver operate independently, although they use the same baud rate generator. The CPU monitors the status of the SCI, writes the data to be transmitted, and processes received data.

Freescale Semiconductor, Inc.

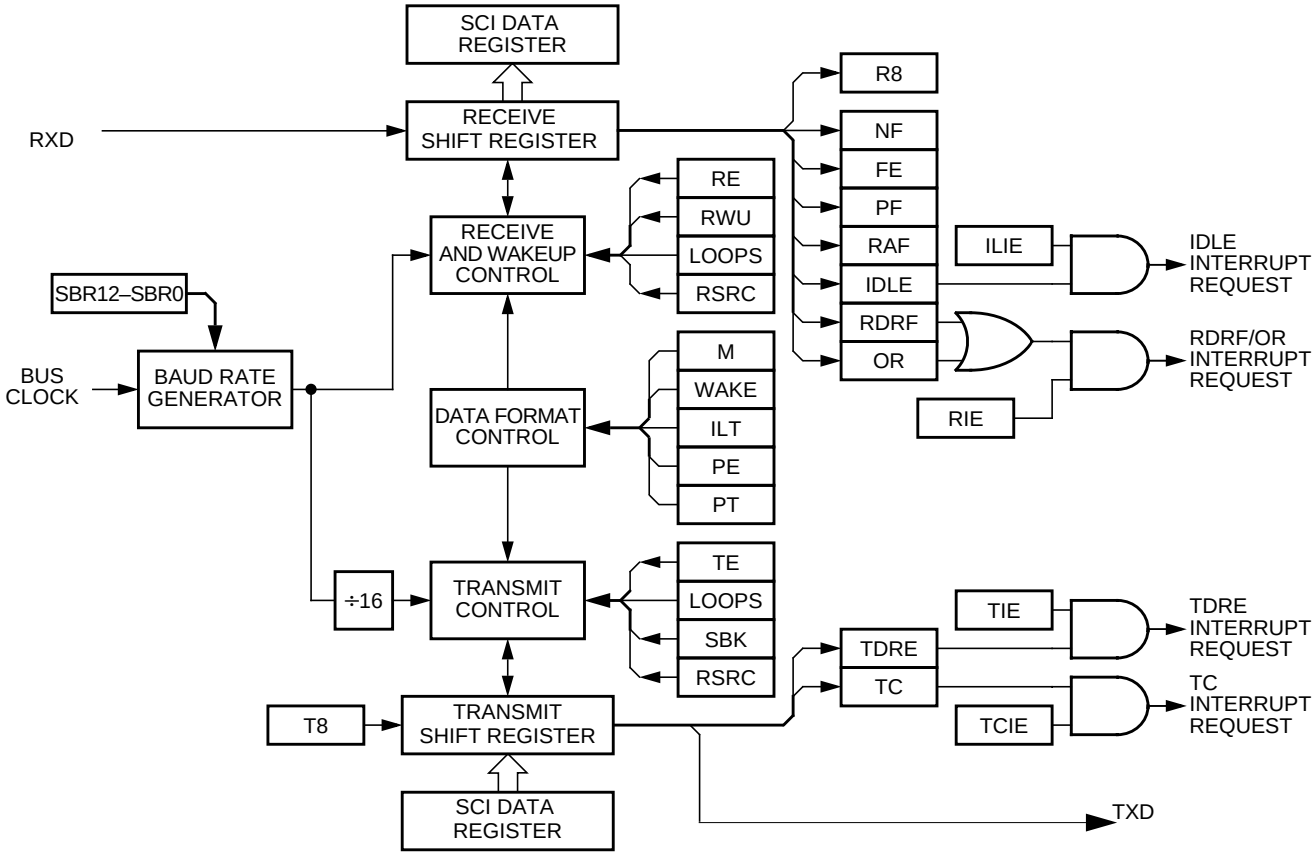


Figure 76 Detailed SCI Block Diagram

Data Format

The SCI uses the standard NRZ mark/space data format illustrated in Figure 77 below.

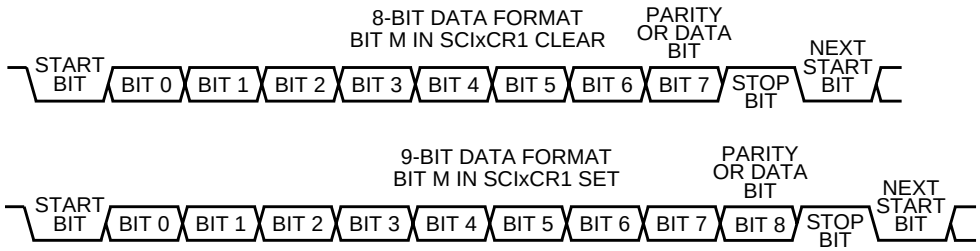


Figure 77 SCI Data Formats

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

Each data character is contained in a frame that includes a start bit, eight or nine data bits, and a stop bit. Clearing the M bit in SCI control register 1 configures the SCI for 8-bit data characters. A frame with eight data bits has a total of 10 bits. Setting the M bit configures the SCI for nine-bit data characters. A frame with nine data bits has a total of 11 bits.

Table 74 Example of 8-bit Data Formats

Start Bit	Data Bits	Address Bits	Parity Bits	Stop Bit
1	8	0	0	1
1	7	0	1	1
1	7	1(1)	0	1

1. The address bit identifies the frame as an address character. See section on Receiver Wakeup.

When the SCI is configured for 9-bit data characters, the ninth data bit is the T8 bit in SCI data register high (SCIxDRH). It remains unchanged after transmission and can be used repeatedly without rewriting it. A frame with nine data bits has a total of 11 bits.

Table 75 Example of 9-Bit Data Formats

Start Bit	Data Bits	Address Bits	Parity Bits	Stop Bit
1	9	0	0	1
1	8	0	1	1
1	8	1(1)	0	1

1. The address bit identifies the frame as an address character. See section on Receiver Wakeup.

Baud Rate Generation

A 13-bit modulus counter in the baud rate generator derives the baud rate for both the receiver and the transmitter. The value from 0 to 8191 written to the SBR12–SBR0 bits determines the bus clock divisor. The SBR bits are in the SCI baud rate registers (SCIxBDH and SCIxBDL). The baud rate clock is synchronized with the bus clock and drives the receiver. The baud rate clock divided by 16 drives the transmitter. The receiver has an acquisition rate of 16 samples per bit time.

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

Freescale Semiconductor, Inc.

Baud rate generation is subject to two sources of error:

- Integer division of the bus clock may not give the exact target frequency.
- Synchronization with the bus clock can cause phase shift.

Table 76 lists some examples of achieving target baud rates with a bus clock frequency of 16 MHz

SCI baud rate = Bus Clock / (16 *SBR[12:0])

Table 76 Baud Rates (Example: Bus Clock = 16.0 MHz)

Bits SBR[12-0]	Receiver Clock (Hz)	Transmitter Clock (Hz)	Target Baud Rate	Error (%)
13	1,230,769	76923	76800	0.16
26	615,385	38,462	38,400	0.16
52	307,692	19,231	19,200	0.16
104	153,846	9615	9600	0.16
208	76,923	4808	4800	0.16
417	38,369	2398	2400	-0.08
833	19,207	1200	1200	0.04
1664	9,598	600	600	-0.02

NOTE: *The maximum divider rate is 8191.*

Freescale Semiconductor, Inc.

Transmitter

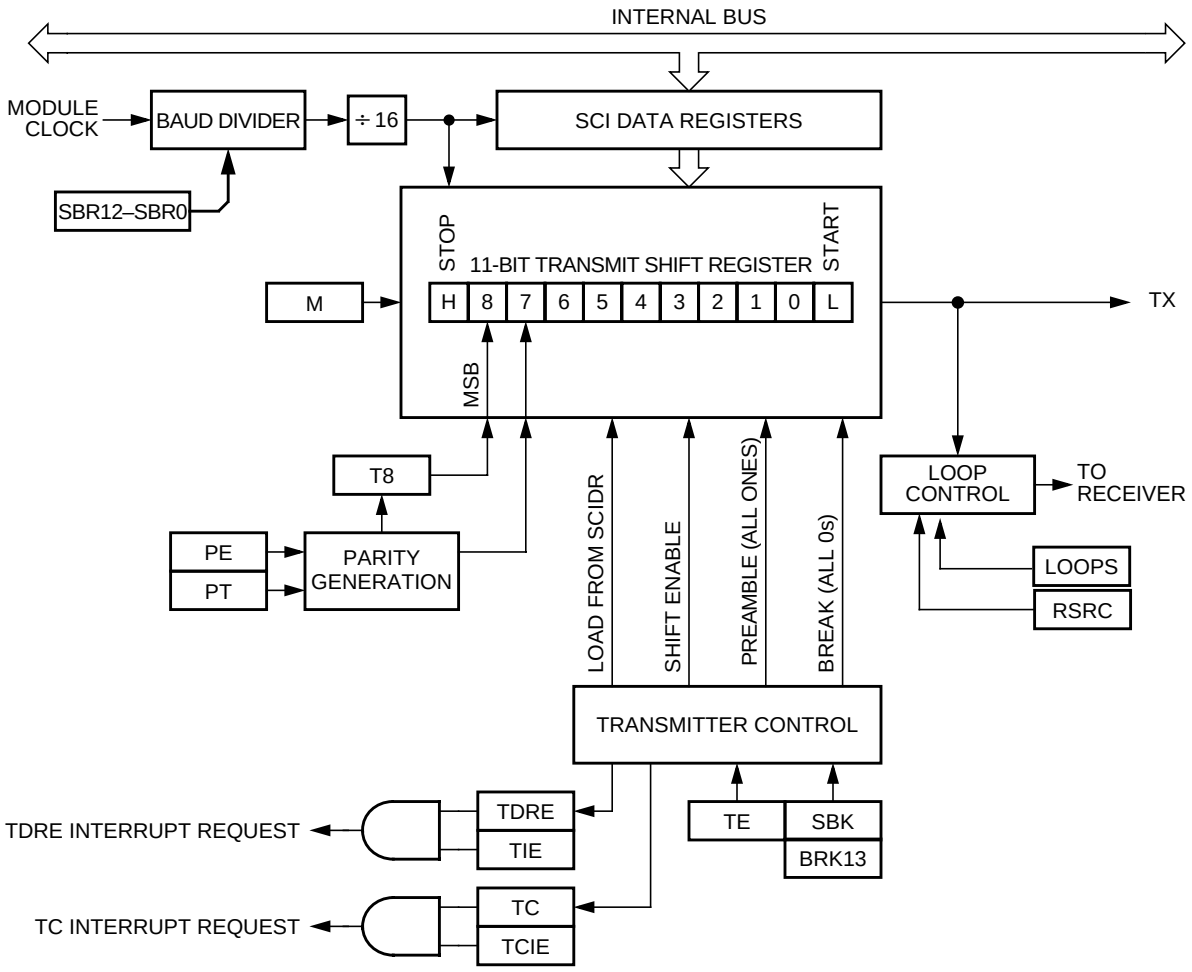


Figure 78 Transmitter Block Diagram

Transmitter Character Length

The SCI transmitter can accommodate either 8-bit or 9-bit data characters. The state of the M bit in SCI control register 1 (SCIxCR1) determines the length of data characters. When transmitting 9-bit data, bit T8 in SCI data register high (SCIxDRH) is the ninth bit (bit 8).

Character Transmission

To transmit data, the MCU writes the data bits to the SCI data registers (SCIxDRH/SCIxDRL), which in turn are transferred to the transmitter

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

Freescale Semiconductor, Inc.

shift register. The transmit shift register then shifts a frame out through the **TX output** signal, after it has prefaced it with a start bit and appended it with a stop bit. The SCI data registers (SCIxDRH and SCIxDRL) are the write-only buffers between the internal data bus and the transmit shift register.

The SCI also sets a flag, the transmit data register empty flag (TDRE), every time it transfers data from the buffer (SCIxDRH/L) to the transmitter shift register.The transmit driver routine may respond to this flag by writing another byte to the Transmitter buffer (SCIxDRH/SCIxDRL), while the shift register is still shifting out the first byte.

To initiate an SCI transmission:

9. Configure the SCI:
    a. Select a baud rate. Write this value to the SCI baud registers (SCIxBDH/L) to begin the baud rate generator. Remember that the baud rate generator is disabled when the baud rate is zero. Writing to the SCIxBDH has no effect without also writing to SCIxBDL.
    b. Write to SCIxCR1 to configure word length, parity, and other configuration bits (LOOPS, RSRC, M, WAKE, ILT, PE, PT).
    c. Enable the transmitter, interrupts, receive, and wake up as required, by writing to the SCIxCR2 register bits (TIE, TCIE, RIE, ILIE, TE, RE, RWU, SBK). A preamble or idle character will now be shifted out of the transmitter shift register.
10. Transmit Procedure for Each Byte:
    a. Poll the TDRE flag by reading the SCIxSR1 or responding to the TDRE interrupt. Keep in mind that the TDRE bit resets to one.
    b. If the TDRE flag is set, write the data to be transmitted to SCIxDRH/L, where the ninth bit is written to the T8 bit in SCIxDRH if the SCI is in 9-bit data format. A new transmission will not result until the TDRE flag has been cleared.
11. Repeat step 2 for each subsequent transmission.

Freescale Semiconductor, Inc.

NOTE: *The TDRE flag is set when the shift register is loaded with the next data to be transmitted from SCIxDRH/L, which happens, generally speaking, a little over half-way through the stop bit of the previous frame. Specifically, this transfer occurs 9/16ths of a bit time AFTER the start of the stop bit of the previous frame.*

Writing the TE bit from 0 to a 1 automatically loads the transmit shift register with a preamble of 10 logic 1s (if M = 0) or 11 logic 1s (if M = 1). After the preamble shifts out, control logic transfers the data from the SCI data register into the transmit shift register. A logic 0 start bit automatically goes into the least significant bit position of the transmit shift register. A logic 1 stop bit goes into the most significant bit position.

Hardware supports odd or even parity. When parity is enabled, the most significant bit (msb) of the data character is the parity bit.

The transmit data register empty flag, TDRE, in SCI status register 1 (SCIxSR1) becomes set when the SCI data register transfers a byte to the transmit shift register. The TDRE flag indicates that the SCI data register can accept new data from the internal data bus. If the transmit interrupt enable bit, TIE, in SCI control register 2 (SCIxCR2) is also set, the TDRE flag generates a transmitter interrupt request.

When the transmit shift register is not transmitting a frame, the **TX output** signal goes to the idle condition, logic 1. If at any time software clears the TE bit in SCI control register 2 (SCIxCR2), the transmitter enable signal goes low and the transmit signal goes idle.

If software clears TE while a transmission is in progress (TC = 0), the frame in the transmit shift register continues to shift out. To avoid accidentally cutting off the last frame in a message, always wait for TDRE to go high after the last frame before clearing TE.

To separate messages with preambles of minimum idle line time, use this sequence between messages:

1. Write the last byte of the first message to SCIxDRH/L.
2. Wait for the TDRE flag to go high, indicating the transfer of the last frame to the transmit shift register.
3. Queue a preamble by clearing and then setting the TE bit.

Freescale Semiconductor, Inc.

4. Write the first byte of the second message to SCIxDRH/L.

Break Characters

Writing a logic 1 to the send break bit, SBK, in SCI control register 2 (SCIxCR2) loads the transmit shift register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on the M bit in SCI control register 1 (SCIxCR1). As long as SBK is at logic 1, transmitter logic continuously loads break characters into the transmit shift register. After software clears the SBK bit, the shift register finishes transmitting the last break character and then transmits at least one logic 1. The automatic logic 1 at the end of a break character guarantees the recognition of the start bit of the next frame.

The SCI recognizes a break character when a start bit is followed by eight or nine logic 0 data bits and a logic 0 where the stop bit should be. Receiving a break character has these effects on SCI registers:

- Sets the framing error flag, FE
- Sets the receive data register full flag, RDRF
- Clears the SCI data registers (SCIxDRH/L)
- May set the overrun flag, OR, noise flag, NF, parity error flag, PE, or the receiver active flag, RAF (see SCI Status Register 1 (SCIxSR1) and SCI Status Register 2 (SCIxSR2))

Idle Characters

An idle character contains all logic 1s and has no start, stop, or parity bit. Idle character length depends on the M bit in SCI control register 1 (SCIxCR1). The preamble is a synchronizing idle character that begins the first transmission initiated after writing the TE bit from 0 to 1.

If the TE bit is cleared during a transmission, the **TX output** signal becomes idle after completion of the transmission in progress. Clearing and then setting the TE bit during a transmission queues an idle character to be sent after the frame currently being transmitted.

NOTE: *When queueing an idle character, return the TE bit to logic 1 before the stop bit of the current frame shifts out through the **TX output** signal. Setting TE after the stop bit appears on **TX output signal** causes data previously written to the SCI data register to be lost. Toggle the TE bit*

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

Freescale Semiconductor, Inc.

for a queued idle character while the TDRE flag is set and immediately before writing the next byte to the SCI data register.

NOTE: *If the TE bit is clear and the transmission is complete, the SCI is not the master of the TxD pin.*

Receiver

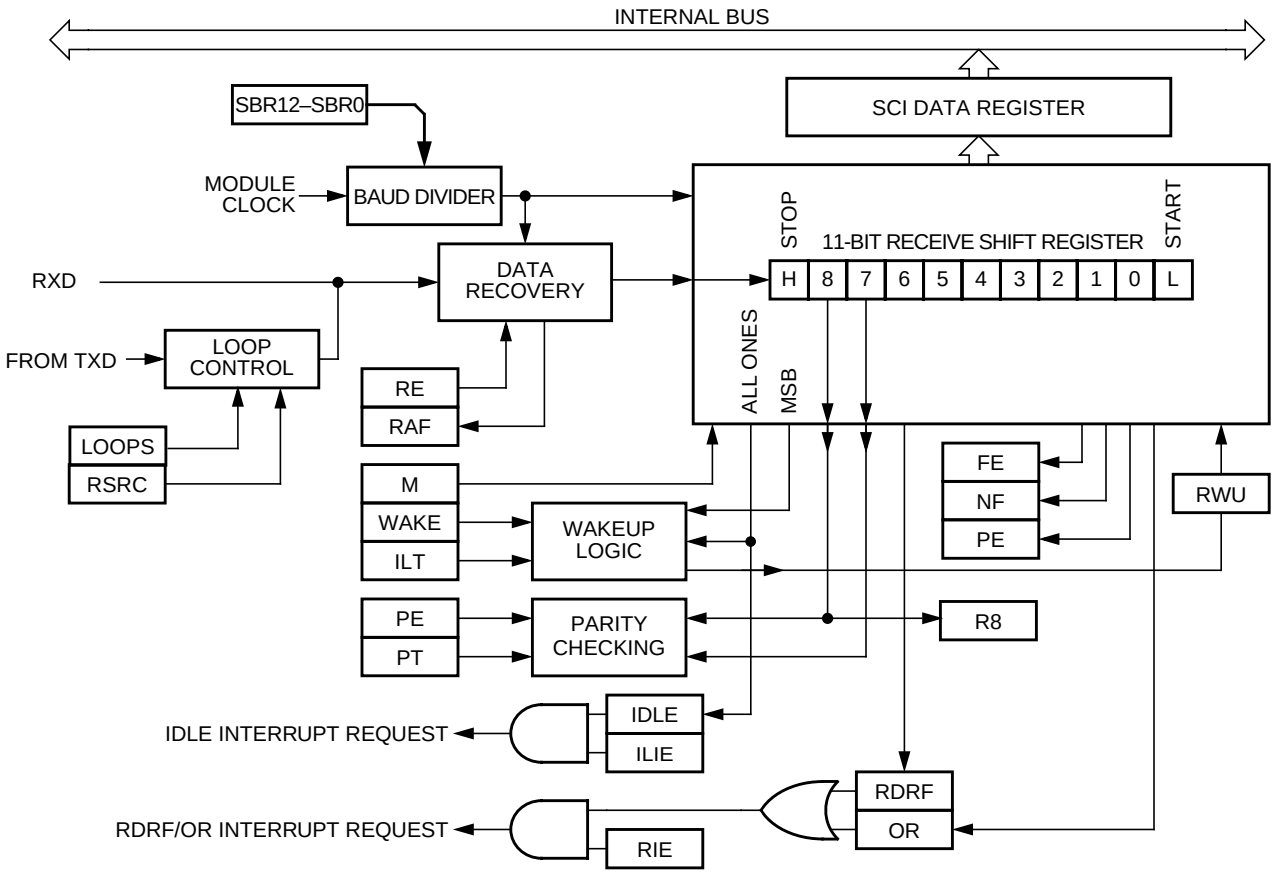


Figure 79 SCI Receiver Block Diagram

Receiver Character Length The SCI receiver can accommodate either 8-bit or 9-bit data characters. The state of the M bit in SCI control register 1 (SCIxCR1) determines the

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

Freescale Semiconductor, Inc.

length of data characters. When receiving 9-bit data, bit R8 in SCI data register high (SCIxDRH) is the ninth bit (bit 8).

Character Reception

During an SCI reception, the receive shift register shifts a frame in from the **RX input** signal. The SCI data register is the read-only buffer between the internal data bus and the receive shift register.

After a complete frame shifts into the receive shift register, the data portion of the frame transfers to the SCI data register. The receive data register full flag, RDRF, in SCI status register 1 (SCIxSR1) becomes set, indicating that the received byte can be read. If the receive interrupt enable bit, RIE, in SCI control register 2 (SCIxCR2) is also set, the RDRF flag generates an RDRF interrupt request.

Data Sampling

The receiver samples the **RX input** signal at the RT clock rate. The RT clock is an internal signal with a frequency 16 times the baud rate. To adjust for baud rate mismatch, the RT clock (see Figure 80) is re-synchronized:

- After every start bit
- After the receiver detects a data bit change from logic 1 to logic 0 (after the majority of data bit samples at RT8, RT9, and RT10 returns a valid logic 1 and the majority of the next RT8, RT9, and RT10 samples returns a valid logic 0)

To locate the start bit, data recovery logic does an asynchronous search for a logic 0 preceded by three logic 1s. When the falling edge of a possible start bit occurs, the RT clock begins to count to 16.

Freescale Semiconductor, Inc.

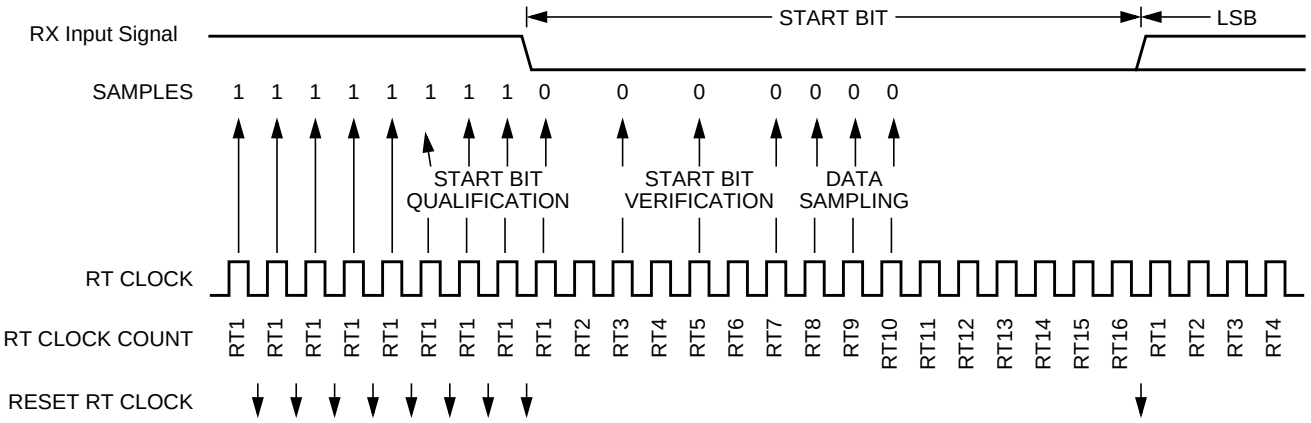


Figure 80 Receiver Data Sampling

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7. Table 77 summarizes the results of the start bit verification samples.

Table 77 Start Bit Verification

RT3, RT5, and RT7 Samples	Start Bit Verification	Noise Flag
000	Yes	0
001	Yes	1
010	Yes	1
011	No	0
100	Yes	1
101	No	0
110	No	0
111	No	0

If start bit verification is not successful, the RT clock is reset and a new search for a start bit begins.

To determine the value of a data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 78 summarizes the results of the data bit samples.

Freescale Semiconductor, Inc.

Table 78 Data Bit Recovery

RT8, RT9, and RT10 Samples	Data Bit Determination	Noise Flag
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

NOTE: *The RT8, RT9, and RT10 samples do not affect start bit verification. If any or all of the RT8, RT9, and RT10 start bit samples are logic 1s following a successful start bit verification, the noise flag (NF) is set and the receiver assumes that the bit is a start bit (logic 0).*

To verify a stop bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 79 summarizes the results of the stop bit samples.

Table 79 Stop Bit Recovery

RT8, RT9, and RT10 Samples	Framing Error Flag	Noise Flag
000	1	0
001	1	1
010	1	1
011	0	1
100	1	1
101	0	1
110	0	1
111	0	0

In Figure 81, the verification samples RT3 and RT5 determine that the first low detected was noise and not the beginning of a start bit. The RT clock is reset and the start bit search begins again. The noise flag is not set because the noise occurred before the start bit was found.

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

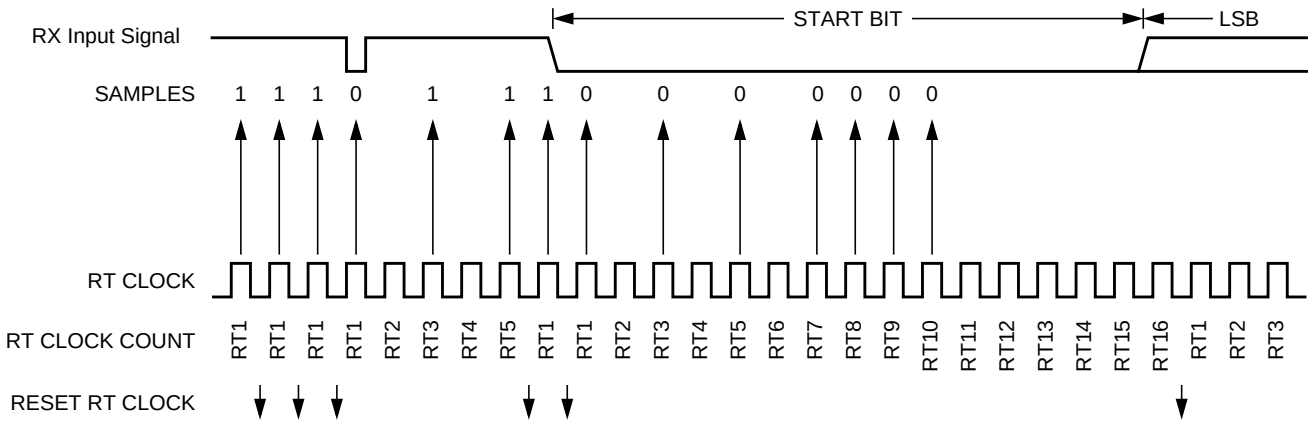


Figure 81 Start Bit Search Example 1

In Figure 82, verification sample at RT3 is high. The RT3 sample sets the noise flag. Although the perceived bit time is misaligned, the data samples RT8, RT9, and RT10 are within the bit time and data recovery is successful.

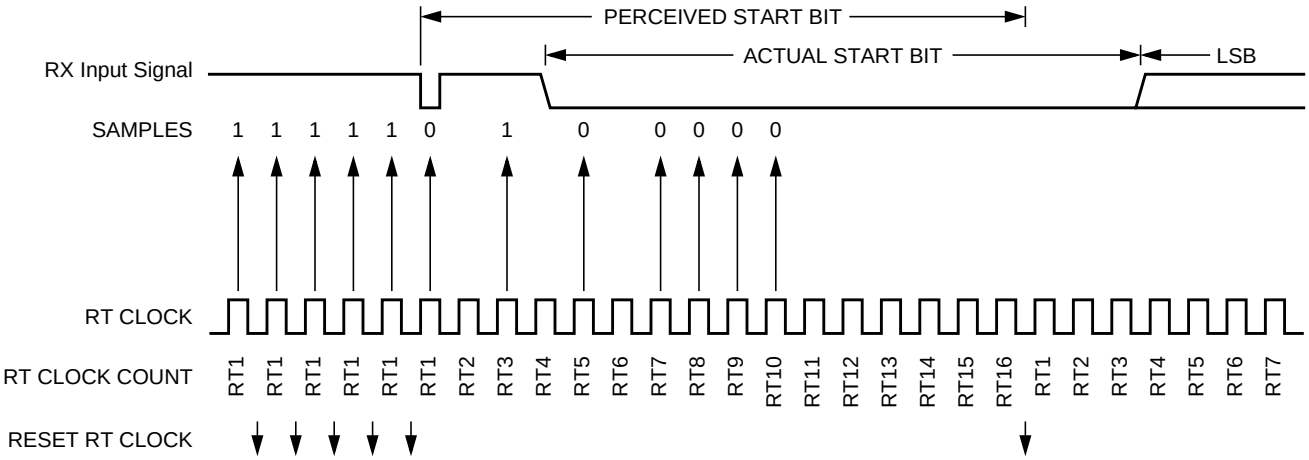


Figure 82 Start Bit Search Example 2

In Figure 83, a large burst of noise is perceived as the beginning of a start bit, although the test sample at RT5 is high. The RT5 sample sets the noise flag. Although this is a worst-case misalignment of perceived

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

Freescale Semiconductor, Inc.

bit time, the data samples RT8, RT9, and RT10 are within the bit time and data recovery is successful.

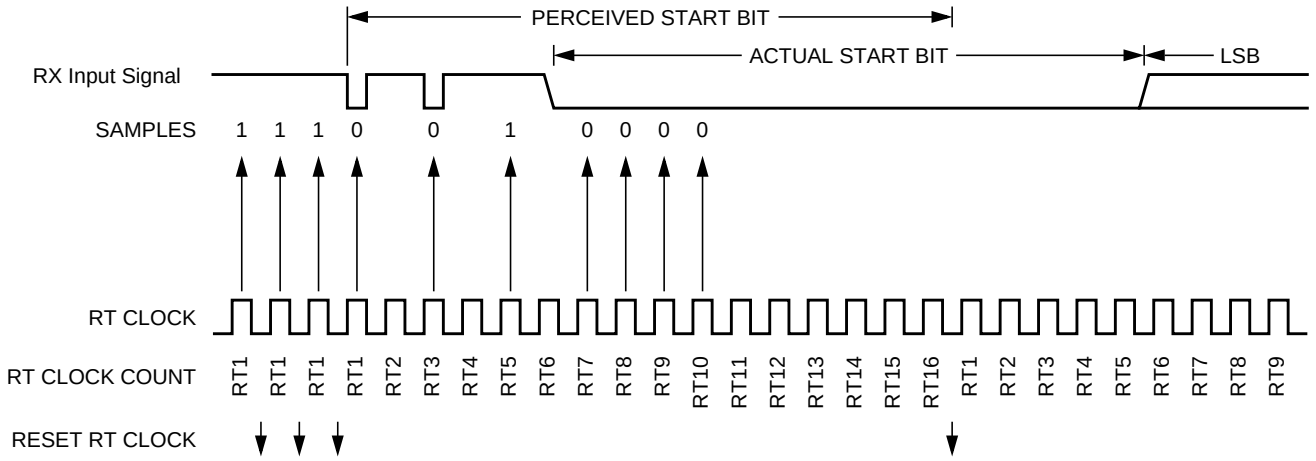


Figure 83 Start Bit Search Example 3

Figure 84 shows the effect of noise early in the start bit time. Although this noise does not affect proper synchronization with the start bit time, it does set the noise flag.

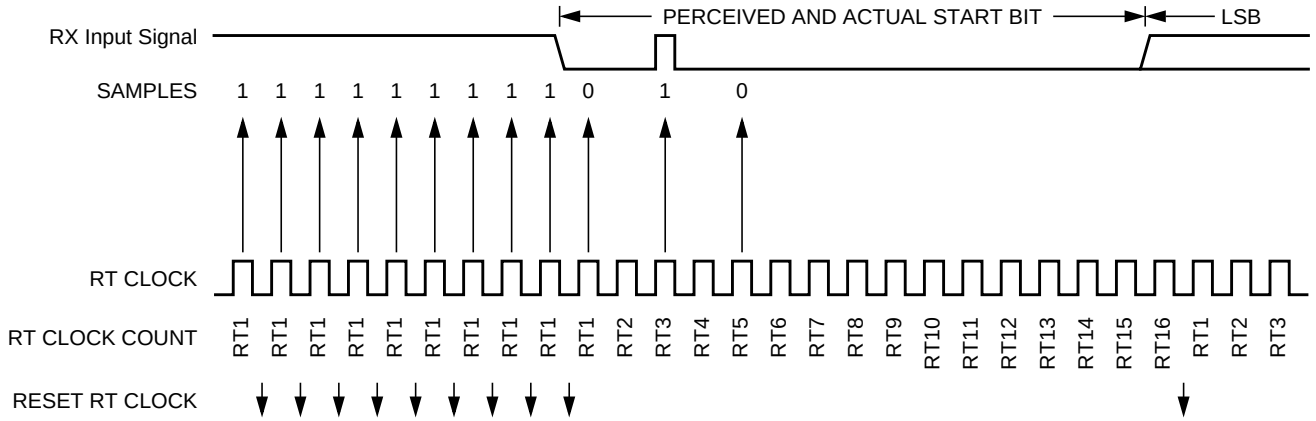


Figure 84 Start Bit Search Example 4

Figure 85 shows a burst of noise near the beginning of the start bit that resets the RT clock. The sample after the reset is low but is not preceded

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

by three high samples that would qualify as a falling edge. Depending on the timing of the start bit search and on the data, the frame may be missed entirely or it may set the framing error flag.

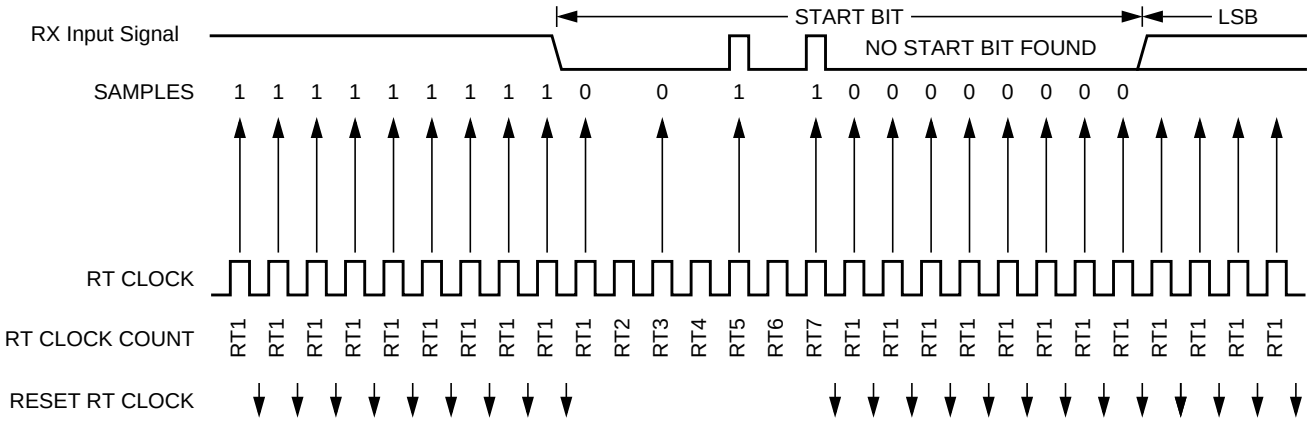


Figure 85 Start Bit Search Example 5

In Figure 86, a noise burst makes the majority of data samples RT8, RT9, and RT10 high. This sets the noise flag but does not reset the RT clock. In start bits only, the RT8, RT9, and RT10 data samples are ignored.

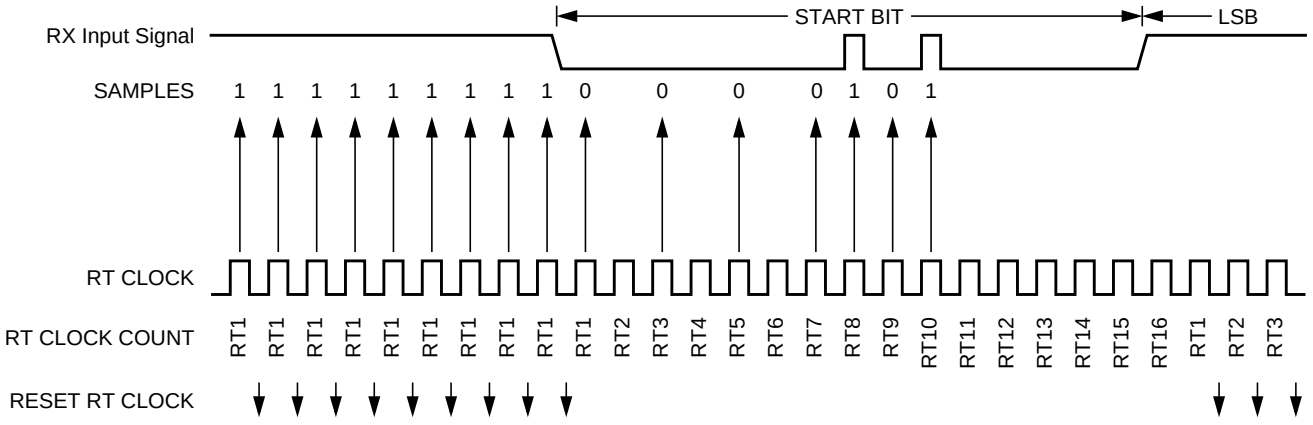


Figure 86 Start Bit Search Example 6

Freescale Semiconductor, Inc.

Framing Errors

If the data recovery logic does not detect a logic 1 where the stop bit should be in an incoming frame, it sets the framing error flag, FE, in SCI status register 1 (SCIxSR1). A break character also sets the FE flag because a break character has no stop bit. The FE flag is set at the same time that the RDRF flag is set.

Baud Rate Tolerance

A transmitting device may be operating at a baud rate below or above the receiver baud rate. Accumulated bit time misalignment can cause one of the three stop bit data samples (RT8, RT9, and RT10) to fall outside the actual stop bit. A noise error will occur if the RT8, RT9, and RT10 samples are not all the same logical values. A framing error will occur if the receiver clock is misaligned in such a way that the majority of the RT8, RT9, and RT10 stop bit samples are a logic zero.

As the receiver samples an incoming frame, it re-synchronizes the RT clock on any valid falling edge within the frame. Resynchronization within frames will correct a misalignment between transmitter bit times and receiver bit times.

Slow Data Tolerance

Figure 87 shows how much a slow received frame can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.

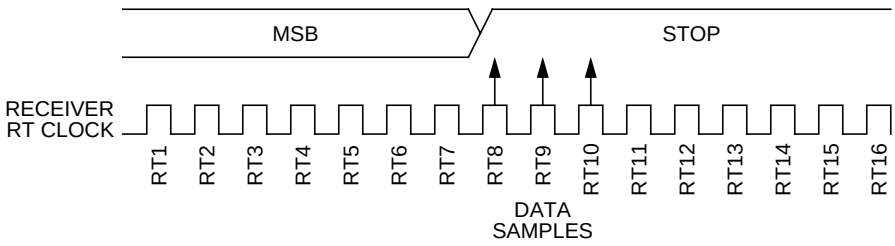


Figure 87 Slow Data

For an 8-bit data character, data sampling of the stop bit takes the receiver 9 bit times x 16 RT cycles +10 RT cycles =154 RT cycles.

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

Freescale Semiconductor, Inc.

With the misaligned character shown in Figure 87, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 9 bit times x 16 RT cycles + 3 RT cycles = 147 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit data character with no errors is:

((154 – 147) / 154) x 100 = 4.54%

For a 9-bit data character, data sampling of the stop bit takes the receiver 10 bit times x 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in Figure 87, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 10 bit times x 16 RT cycles + 3 RT cycles = 163 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is:

((170 – 163) / 170) x 100 = 4.12%

Fast Data Tolerance

Figure 88 shows how much a fast received frame can be misaligned. The fast stop bit ends at RT10 instead of RT16 but is still sampled at RT8, RT9, and RT10.

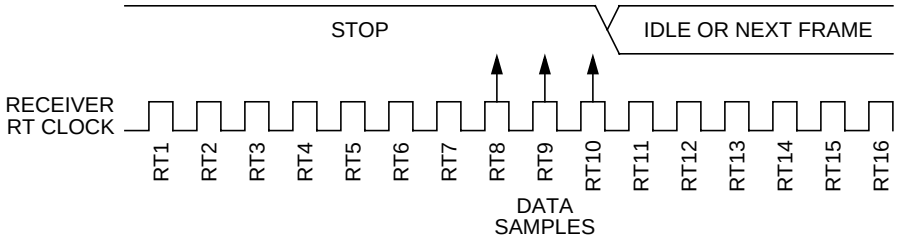


Figure 88 Fast Data

For an 8-bit data character, data sampling of the stop bit takes the receiver 9 bit times x 16 RT cycles + 10 RT cycles = 154 RT cycles.

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

Freescale Semiconductor, Inc.

With the misaligned character shown in Figure 88, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 10 bit times x 16 RT cycles = 160 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is:

((154 – 160) / 154) x 100 = 3.90%

For a 9-bit data character, data sampling of the stop bit takes the receiver 10 bit times x 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in Figure 88, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 11 bit times x 16 RT cycles = 176 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is:

((170 – 176) / 170) x 100 = 3.53%

Receiver Wakeup

To enable the SCI to ignore transmissions intended only for other receivers in multiple-receiver systems, the receiver can be put into a standby state. Setting the receiver wakeup bit, RWU, in SCI control register 2 (SCIxCR2) puts the receiver into a standby state during which receiver interrupts are disabled.The SCI will still load the receive data into the SCIxDRH/L registers, but it will not set the RDRF flag.

The transmitting device can address messages to selected receivers by including addressing information in the initial frame or frames of each message.

The WAKE bit in SCI control register 1 (SCIxCR1) determines how the SCI is brought out of the standby state to process an incoming message. The WAKE bit enables either idle line wakeup or address mark wakeup.

Idle input line wakeup (WAKE = 0)

In this wakeup method, an idle condition on the **RX input** signal clears the RWU bit and wakes up the SCI. The initial frame or frames of every message contain addressing information. All receivers evaluate the addressing information, and receivers for which the message is

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

Freescale Semiconductor, Inc.

addressed process the frames that follow. Any receiver for which a message is not addressed can set its RWU bit and return to the standby state. The RWU bit remains set and the receiver remains on standby until another idle character appears on the **RX input** signal.

Idle line wakeup requires that messages be separated by at least one idle character and that no message contains idle characters.

The idle character that wakes a receiver does not set the receiver idle bit, IDLE, or the receive data register full flag, RDRF.

The idle line type bit, ILT, determines whether the receiver begins counting logic 1s as idle character bits after the start bit or after the stop bit. ILT is in SCI control register 1 (SCIxCR1).

Address mark wakeup (WAKE = 1)

In this wakeup method, a logic 1 in the most significant bit (msb) position of a frame clears the RWU bit and wakes up the SCI. The logic 1 in the msb position marks a frame as an address frame that contains addressing information. All receivers evaluate the addressing information, and the receivers for which the message is addressed process the frames that follow. Any receiver for which a message is not addressed can set its RWU bit and return to the standby state. The RWU bit remains set and the receiver remains on standby until another address frame appears on the **RX input** signal.

The logic 1 msb of an address frame clears the receiver's RWU bit before the stop bit is received and sets the RDRF flag.

Address mark wakeup allows messages to contain idle characters but requires that the msb be reserved for use in address frames.

NOTE: *With the WAKE bit clear, setting the RWU bit after the **RX input** signal has been idle can cause the receiver to wake up immediately.*

Single-Wire Operation

Normally, the SCI uses two pins for transmitting and receiving. In single-wire operation, the RxD pin is disconnected from the SCI. The SCI uses the TxD pin for both receiving and transmitting.

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

Freescale Semiconductor, Inc.

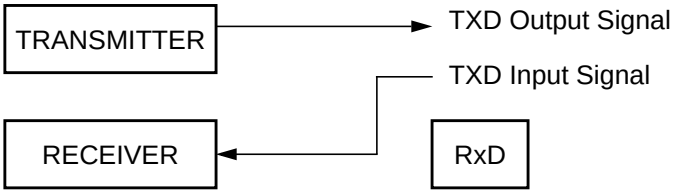


Figure 89 Single-Wire Operation (LOOPS = 1, RSRC = 1)

Enable single-wire operation by setting the LOOPS bit and the receiver source bit, RSRC, in SCI control register 1 (SCIxCR1). Setting the LOOPS bit disables the path from the **RX input** signal to the receiver. Setting the RSRC bit connects the receiver input to the output of the TxD pin driver. Both the transmitter and receiver must be enabled (TE=1 and RE=1).The TXDIR bit (SCIxSR2[1]) determines whether the TxD pin is going to be used as an input (TXDIR = 0) or an output (TXDIR = 1) in this mode of operation.

Loop Operation

In loop operation the transmitter output goes to the receiver input. The **RX input** signal is disconnected from the SCI.

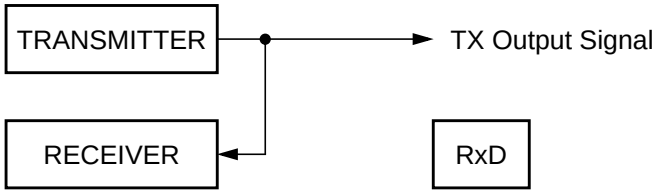


Figure 90 Loop Operation (LOOPS = 1, RSRC = 0)

Enable loop operation by setting the LOOPS bit and clearing the RSRC bit in SCI control register 1 (SCIxCR1). Setting the LOOPS bit disables the path from the **RX input** signal to the receiver. Clearing the RSRC bit connects the transmitter output to the receiver input. Both the transmitter and receiver must be enabled (TE = 1 and RE = 1).

Freescale Semiconductor, Inc.

Interrupts

Recovery from Wait Mode

The SCI interrupt request can be used to bring the CPU out of wait mode.

System Level Interrupt Sources

This section describes the interrupt originated by the SCI block.The MCU must service the interrupt requests. Table 80 lists the five interrupt sources of the SCI. The local enables for the five SCI interrupt sources, are described in Table 80.

Table 80 SCI Interrupt Sources

Interrupt	Vector Address		Source	Description
	SCI0	SCI1		
TDRE	$FFD6, $FFD7	$FFD4, $FFD5	SCIxSR1[7]	Active high level detect. Indicates that a byte was transferred from SCIxDRH/L to the transmit shift register.
TC			SCIxSR1[6]	Active high level detect. Indicates that a transmit is complete.
RDRF			SCIxSR1[5]	Active high level detects. The RDRF interrupt indicates that received data is available in the SCI data register.
OR			SCIxSR1[3]	Active high level detects. This interrupt indicates that an overrun condition has occurred.
IDLE			SCIxSR1[4]	Active high level detect. Indicates that receiver input has become idle.

The SCI only originates interrupt requests. The following is a description of how the SCI makes a request and how the MCU should acknowledge that request. The SCI only has a single interrupt line (**SCI Interrupt signal**, active high operation) and all the following interrupts, when generated, are ORed together and issued through that port.

TDRE Description

The TDRE interrupt is set high by the SCI when the transmit shift register receives a byte from the SCI data register. A TDRE interrupt indicates that the transmit shift register (SCIxDRH/L) is empty and that a new byte can be written to the SCIxDRH/L for transmission.Clear TDRE by

Freescale Semiconductor, Inc.

reading SCI status register 1 with TDRE set and then writing to SCI data register low (SCIxDRL).

TC Description

The TC interrupt is set by the SCI when a transmission has been completed. A TC interrupt indicates that there is no transmission in progress. TC is set high when the TDRE flag is set and no data, preamble, or break character is being transmitted. When TC is set, the TxD pin becomes idle (logic 1). Clear TC by reading SCI status register 1 (SCIxSR1) with TC set and then writing to SCI data register low (SCIxDRL). TC is cleared automatically when data, preamble, or break is queued and ready to be sent.

RDRF Description

The RDRF interrupt is set when the data in the receive shift register transfers to the SCI data register. A RDRF interrupt indicates that the received data has been transferred to the SCI data register and that the byte can now be read by the MCU. The RDRF interrupt is cleared by reading the SCI status register one (SCIxSR1) and then reading SCI data register low (SCIxDRL).

OR Description

The OR interrupt is set when software fails to read the SCI data register before the receive shift register receives the next frame. The newly acquired data in the shift register will be lost in this case, but the data already in the SCI data registers is not affected. The OR interrupt is cleared by reading the SCI status register one (SCIxSR1) and then reading SCI data register low (SCIxDRL).

IDLE Description

The IDLE interrupt is set when 10 consecutive logic 1s (if M=0) or 11 consecutive logic 1s (if M=1) appear on the receiver input. Once the IDLE is cleared, a valid frame must again set the RDRF flag before an idle condition can set the IDLE flag. Clear IDLE by reading SCI status register 1 (SCIxSR1) with IDLE set and then reading SCI data register low (SCIxDRL).

Freescale Semiconductor, Inc.

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

Freescale Semiconductor, Inc.

Serial Peripheral Interface (SPI)

Contents

Overview 457
Features 458
Modes of Operation 458
Block Diagram 459
External Pin Descriptions 460
Register Map 461
Register Descriptions 462
Functional Description 470
Low Power Mode Options 483
Reset Initialization 484
Interrupts 484

Overview

The Serial Peripheral Interface module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or the SPI operation can be interrupt driven.

Freescale Semiconductor, Inc.

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

Features

The Serial Peripheral Interface includes these distinctive features:

- Master mode and slave mode
- Bi-directional mode
- Slave select output
- Mode fault error flag with CPU interrupt capability
- Double-buffered operation
- Serial clock with programmable polarity and phase
- Control of SPI operation during wait mode

Modes of Operation

The SPI functions in three modes, run, wait, and stop.

- Run Mode

  This is the basic mode of operation.

- Wait Mode

  SPI operation in wait mode is a configurable low power mode. Depending on the state of internal bits, the SPI can operate normally when the CPU is in wait mode or the SPI clock generation can be turned off and the SPI module enters a power conservation state during wait mode. During wait mode, any master transmission in progress stops if the SPISWAI bit is set in the SPICR2 register. Reception and transmission of a byte as slave continues so that the slave is synchronized to the master.

- Stop Mode

  The SPI is inactive in stop mode for reduced power consumption. The STOP instruction does not affect or depend on SPI register states. Again, reception and transmission of a byte as slave continues to stay synchronized with the master.

Freescale Semiconductor, Inc.

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

This is a high level description only, detailed descriptions of operating modes are contained in section Low Power Mode Options.

Block Diagram

Figure 91 is a block diagram of the Serial Peripheral Interface.

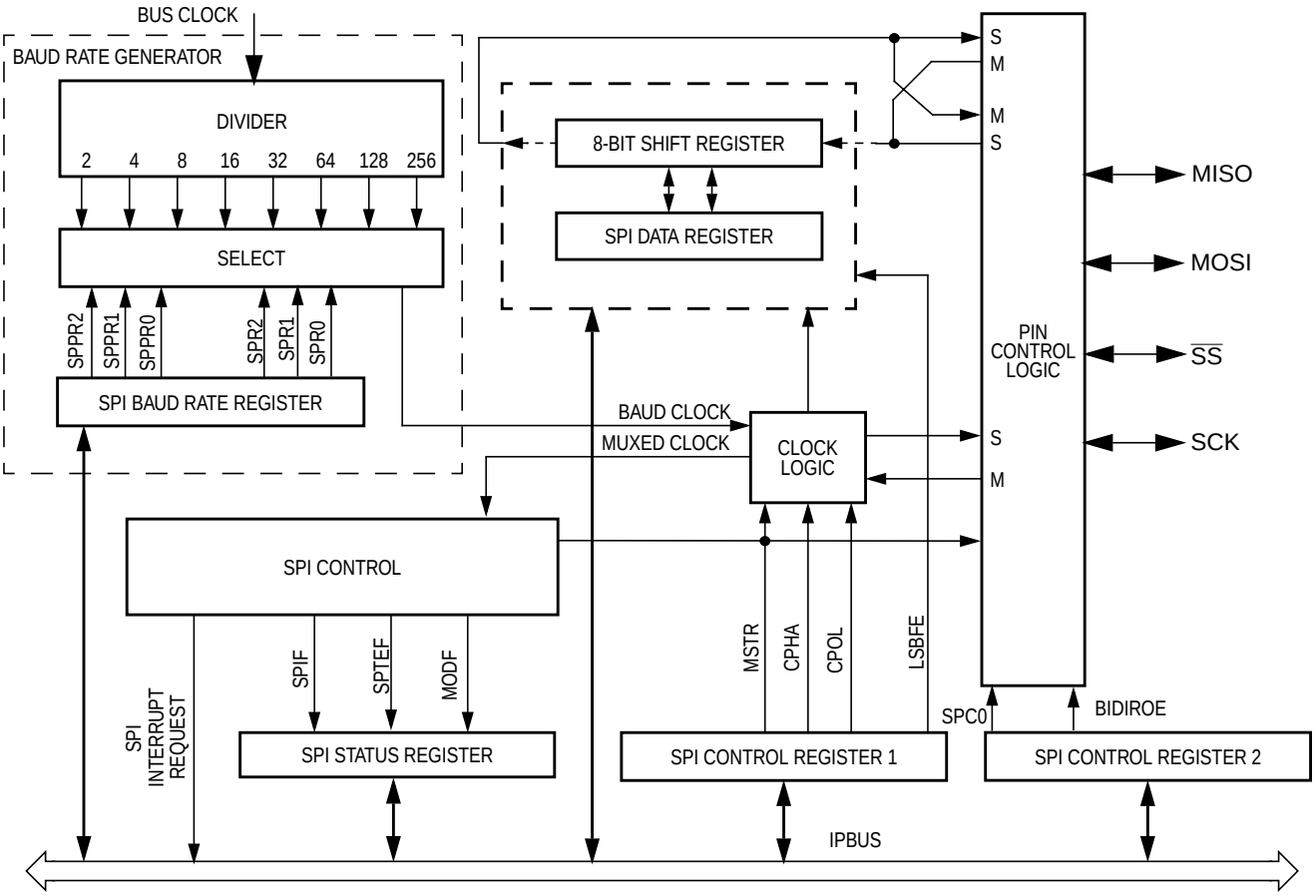


Figure 91 SPI Block Diagram

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

External Pin Descriptions

MISO (PS4) Master In / Slave Out. This pin is used to transmit data out of the SPI module when it is configured as a Slave, and receive data when it is configured as Master

MOSI (PS5) Master Out/ Slave In. This pin is used to transmit data out of the SPI module when it is configured as a Master, and receive data when it is configured as Slave.

SCK (PS6) Serial Clock. This pin is used to output the clock with respect to which the SPI module tranfers and receives data.

$\overline{SS}$ (PS7) Slave Select. This pin is used to output the select signal from the SPI module to another peripheral with which a data transfer is to take place.

Freescale Semiconductor, Inc.

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

Register Map

Register Name		Bit 7	6	5	4	3	2	1	Bit 0	Address Offset
SPICR1	Read:	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE	$00D8
	Write:									
SPICR2	Read:	0	0	0	MOD-FEN	BIDIROE	0	SPISWAI	SPC0	$00D9
	Write:									
SPIBR	Read:	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0	$00DA
	Write:									
SPISR	Read:	SPIF	0	SPTEF	MODF	0	0	0	0	$00DB
	Write:									
SPI Reserved	Read:	0	0	0	0	0	0	0	0	$00DC
	Write:									
SPIDR	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$00DD
	Write:									
SPI Reserved	Read:	0	0	0	0	0	0	0	0	$00DE
	Write:									
SPI Reserved	Read:	0	0	0	0	0	0	0	0	$00DF
	Write:									

= Unimplemented or Reserved

Figure 92 SPI Register Summary

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Freescale Semiconductor, Inc.

Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

SPI Control Register 1 (SPICR1)

Address Offset: $00D8

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
Reset:	0	0	0	0	0	1	0	0

Read: anytime
Write: anytime

SPIE — SPI Interrupt Enable Bit

This bit enables SPI interrupts each time the SPIF or MODF status flag is set.

1 = SPI interrupts enabled.
0 = SPI interrupts disabled.

SPE — SPI System Enable Bit

This bit enables the SPI system and dedicates the SPI port pins to SPI system functions.

1 = SPI port pins are dedicated to SPI functions.
0 = SPI disabled. (lower power consumption)

SPTIE — SPI Transmit Interrupt Enable

This bit enables SPI interrupt generated each time the SPTEF flag is set.

1 = SPTEF interrupt enabled.
0 = SPTEF interrupt disabled.

MSTR — SPI Master/Slave Mode Select Bit

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

Freescale Semiconductor, Inc.

1 = Master mode
0 = Slave mode

CPOL — SPI Clock Polarity Bit

This bit selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values.

1 = Active-low clocks selected; SCK idles high
0 = Active-high clocks selected; SCK idles low

CPHA — SPI Clock Phase Bit

This bit is used to shift the SCK serial clock.

1 = The first SCK edge is issued at the beginning of the 8-cycle transfer operation
0 = The first SCK edge is issued one-half cycle into the 8-cycle transfer operation

SSOE — Slave Select Output Enable

The $\overline{SS}$ output feature is enabled only in the master mode by asserting the SSOE as shown in Table 81.

Table 81 $\overline{SS}$ Input / Output Selection

MOD FEN	SSOE	Master Mode	Slave Mode
0	0	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
0	1	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
1	0	$\overline{SS}$ input with MODF feature	$\overline{SS}$ input
1	1	$\overline{SS}$ output	$\overline{SS}$ input

LSBFE — SPI LSB-First Enable

This bit does not affect the position of the msb and lsb in the data register. Reads and writes of the data register always have the msb in bit 7.

1 = Data is transferred least significant bit first.
0 = Data is transferred most significant bit first.

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

Freescale Semiconductor, Inc.

SPI Control Register 2 (SPICR2)

Address Offset: $00D9

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Read: anytime
Write: anytime; writes to unimplemented bits have no effect

MODFEN — Mode Fault Enable Bit

This bit when set allows the MODF flag to be set. If the MODF flag is set, clearing the MODFEN does not clear the MODF flag. If the SPI is enabled as master and the MODFEN bit is low, then the $\overline{SS}$ pin is not used by the SPI.

When the SPI is enabled as a slave, the $\overline{SS}$ is available only as an input regardless of the value of MODFEN.

1 = Enable setting the MODF error
0 = Disable the MODF error

BIDIROE — Output enable in the Bidirectional mode of operation

This bit along with the MSTR bit of SPCR1 is used to enable the output buffer when the SPI is configured in bidirectional mode.

1 = Output buffer enabled
0 = Output buffer disabled

SPISWAI — SPI Stop in Wait Mode Bit

This bit is used for power conservation while in wait mode.

1 = Stop SPI clock generation when in wait mode
0 = SPI clock operates normally in wait mode

SPC0 — Serial Pin Control Bit 0

With the MSTR control bit, this bit enables bidirectional pin configurations as shown in Table 82.

Freescale Semiconductor, Inc.

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

Table 82 Bidirectional Pin Configurations

Pin Mode		SPC0	MSTR	MISO (1)	MOSI (2)	SCK (3)	$\overline{SS}$ (4)
A	Normal	0	0	Slave Out	Slave In	SCK in	$\overline{SS}$ in
B			1	Master In	Master Out	SCK out	$\overline{SS}$ I/O
C	Bidirectional	1	0	Slave I/O	---	SCK in	$\overline{SS}$ In
D			1	---	Master I/O	SCK out	$\overline{SS}$ I/O

1. Slave output is enabled if BIDIROE bit = 1, $\overline{SS}$ = 0, and MSTR = 0 (C)
2. Master output is enabled if BIDIROE bit = 1 and MSTR = 1 (D)
3. SCK output is enabled if MSTR = 1 (B, D)
4. $\overline{SS}$ output is enabled if MODFEN bit = 1, SSOE = 1, and MSTR = 1 (B, D).

SPI Baud Rate Register (SPIBR)

Address Offset: $00DA

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Read: anytime
Write: anytime; writes to unimplemented bits have no effect

NOTE: *Writing to this register during data transfers may cause spurious results*

SPPR2–SPPR0 — SPI Baud Rate Preselection Bits

SPR2–SPR0 — SPI Baud Rate Selection Bits

These bits are used in the determination of the SPI baud rates as shown in the Table 83. For details see *SPI Baud Rate Generation* in page 479.

The bus clock divisor equation is as follows;

$$BusClockDivisor = (SPPR + 1) \bullet 2^{(SPR + 1)}$$

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

Freescale Semiconductor, Inc.

$$BaudRate = \frac{BusClock}{BusClockDivisor}$$

Table 83 SPI Baud Rate Selection (16 MHz Bus Clock)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Bus Clock Divisor	Baud Rate
0	0	0	0	0	0	2 (1)	8 MHz
0	0	0	0	0	1	4 (1)	4 MHz
0	0	0	0	1	0	8	2 MHz
0	0	0	0	1	1	16	1 MHz
0	0	0	1	0	0	32	500 KHz
0	0	0	1	0	1	64	250 KHz
0	0	0	1	1	0	128	125 KHz
0	0	0	1	1	1	256	62.5 KHz
0	0	1	0	0	0	4 (1)	4 MHz
0	0	1	0	0	1	8	2 MHz
0	0	1	0	1	0	16	1 MHz
0	0	1	0	1	1	32	500 KHz
0	0	1	1	0	0	64	250 KHz
0	0	1	1	0	1	128	125 KHz
0	0	1	1	1	0	256	62.5 KHz
0	0	1	1	1	1	512	31.25 KHz
0	1	0	0	0	0	6	2.67 MHz
0	1	0	0	0	1	12	1.33 MHz
0	1	0	0	1	0	24	666.67 KHz
0	1	0	0	1	1	48	333.33 KHz
0	1	0	1	0	0	96	166.67 KHz
0	1	0	1	0	1	192	83.33 KHz
0	1	0	1	1	0	384	41.67 KHz
0	1	0	1	1	1	768	20.83 KHz
0	1	1	0	0	0	8	2 MHz
0	1	1	0	0	1	16	1 MHz
0	1	1	0	1	0	32	500 KHz
0	1	1	0	1	1	64	250 KHz
0	1	1	1	0	0	128	125 KHz
0	1	1	1	0	1	256	62.5 KHz
0	1	1	1	1	0	512	31.25 KHz
0	1	1	1	1	1	1024	15.63 KHz

Freescale Semiconductor, Inc.

Table 83 SPI Baud Rate Selection (16 MHz Bus Clock) (Continued)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Bus Clock Divisor	Baud Rate
1	0	0	0	0	0	10	1.6 MHz
1	0	0	0	0	1	20	800 KHz
1	0	0	0	1	0	40	400 KHz
1	0	0	0	1	1	80	200 KHz
1	0	0	1	0	0	160	100 KHz
1	0	0	1	0	1	320	50 KHz
1	0	0	1	1	0	640	25 KHz
1	0	0	1	1	1	1280	12.5 KHz
1	0	1	0	0	0	12	1.33 MHz
1	0	1	0	0	1	24	666.67 KHz
1	0	1	0	1	0	48	333.33 KHz
1	0	1	0	1	1	96	166.67 KHz
1	0	1	1	0	0	192	83.33 KHz
1	0	1	1	0	1	384	41.67 KHz
1	0	1	1	1	0	768	20.83 KHz
1	0	1	1	1	1	1536	10.42 KHz
1	1	0	0	0	0	14	1.14 MHz
1	1	0	0	0	1	28	571.43 KHz
1	1	0	0	1	0	56	285.71 KHz
1	1	0	0	1	1	112	142.86 KHz
1	1	0	1	0	0	224	71.43 KHz
1	1	0	1	0	1	448	35.71 KHz
1	1	0	1	1	0	896	17.86 KHz
1	1	0	1	1	1	1792	8.93 KHz
1	1	1	0	0	0	16	1 MHz
1	1	1	0	0	1	32	500 MHz
1	1	1	0	1	0	64	250 KHz
1	1	1	0	1	1	128	125 KHz
1	1	1	1	0	0	256	62.5 KHz
1	1	1	1	0	1	512	31.25 KHz
1	1	1	1	1	0	1024	15.63 KHz
1	1	1	1	1	1	2048	7.81 KHz

1. These Bus Clock Divisors are not supported in slave mode of SPI.

Freescale Semiconductor, Inc.

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

SPI Status Register (SPISR)

Address Offset: $00DB

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SPIF	0	SPTEF	MODF	0	0	0	0
Write:								
Reset:	0	0	1	0	0	0	0	0

= Unimplemented or Reserved

Read: anytime
Write: has no meaning or effect

SPIF — SPI Interrupt Flag

This bit is set after the eighth SCK cycle in a data transfer and is cleared by reading the SPISR register (with SPIF set) followed by a read access to the SPI data register.

1 = New data Copied to SPIDR
0 = Transfer not yet complete

SPTEF — SPI Transmit Empty Interrupt Flag

NOTE: *There is an errata information about the SPTEF flag. See MC9S12T64 Errata Sheet for details.*

This bit is set when there is room in the transmit data buffer. It is cleared by reading SPISR with SPTEF set, followed by writing a data value to the transmit buffer at SPIDR. SPISR must be read with SPTEF=1 before writing data to SPIDR or the SPIDR write will be ignored. SPTEF generates an SPTEF CPU interrupt request if the SPTIE bit in the SPICR1 is also set. SPTEF is automatically set when a data byte transfers from the transmit buffer into the transmit shift register. For an idle SPI (no data in the transmit buffer or the shift register and no transfer in progress), data written to SPIDR is transferred to the shifter almost immediately so SPTEF is set within two bus cycles allowing a second 8-bit data value to be queued into the transmit buffer. After completion of the transfer of the value in the shift register, the queued value from the transmit buffer will automatically move to the shifter and SPTEF will be set to indicate

Freescale Semiconductor, Inc.

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

there is room for new data in the transmit buffer. If no new data is waiting in the transmit buffer, SPTEF simply remains set and no data moves from the buffer to the shifter.

1 = SPI Data register empty
0 = SPI Data register not empty

NOTE: *Do not write to the SPI data register unless the SPTEF bit is high. Any such write to the SPI Data Register before reading SPTEF=1 is effectively ignored*

MODF — Mode Fault Flag

NOTE: *There is an errata information about the mode fault behavior. See MC9S12T64 Errata Sheet for details.*

This bit is set if the $\overline{SS}$ input becomes low while the SPI is configured as a master. The flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to the SPI control register 1. The MODF flag is set only if the MODFEN bit of SPICR2 register is set, Refer to MODFEN bit description in SPI Control Register 2 (SPICR2).

1 = Mode fault has occurred.
0 = Mode fault has not occurred.

SPI Data Register (SPIDR)

Address Offset: $00DD

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	2	Bit 0
Reset:	0	0	0	0	0	0	0	0

Read: anytime; normally read only after SPIF is set
Write: anytime; see SPTEF bit in SPISR register

The SPI Data register is both the input and output register for SPI data. A write to this register allows a data byte to be queued and transmitted. For a SPI configured as a master, a queued data byte is

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

Freescale Semiconductor, Inc.

transmitted immediately after the previous transmission has completed. The SPI Transmitter empty flag in SPISR indicates when the SPI data register is ready to accept new data.

NOTE: *Do not write to the SPI data register unless the SPTEF bit is high.*

Reading the data can occur anytime from after the SPIF is set to before the end of the next transfer. If the SPIF is not serviced by the end of the successive transfers, those data bytes are lost and the data within the SPIDR retains the first byte until SPIF is serviced.

NOTE: *After reset the content of the SPI Shift Register is undefined until a data byte is stored into SPIDR.*

Functional Description

The SPI module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or SPI operation can be interrupt driven.

The SPI system is enabled by setting the SPI enable (SPE) bit in SPI control register 1. While SPE is set, the four associated SPI port pins are dedicated to the SPI function as:

- Slave select ($\overline{SS}$)
- Serial clock (SCK)
- Master out/slave in (MOSI)
- Master in/slave out (MISO)

The main element of the SPI system is the SPI data register. The 8-bit data register in the master and the 8-bit data register in the slave are linked by the MOSI and MISO pins to form a distributed 16-bit register. When a data transfer operation is performed, this 16-bit register is serially shifted eight bit positions by the SCK clock from the master; data is exchanged between the master and the slave. Data written to the master SPI data register becomes the output data for the slave, and data read from the master SPI data register after a transfer operation is the input data from the slave.

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

Freescale Semiconductor, Inc.

A write to the SPI data register puts data into the transmit buffer if the previous transmission was complete. When a transfer is complete, received data is moved into a receive data register. Data may be read from this double-buffered system any time before the next transfer is complete. This 8-bit data register acts as the SPI receive data register for reads and as the SPI transmit data register for writes. A single SPI register address is used for reading data from the read data buffer and for writing data to the shifter.

The clock phase control bit (CPHA) and a clock polarity control bit (CPOL) in the SPI control register 1 select one of four possible clock formats to be used by the SPI system. The CPOL bit simply selects a non-inverted or inverted clock. The CPHA bit is used to accommodate two fundamentally different protocols by shifting the clock by a half cycle or by not shifting the clock (see Transmission Formats).

The SPI can be configured to operate as a master or as a slave. When MSTR in SPI control register1 is set, the master mode is selected; when the MSTR bit is clear, the slave mode is selected.

Master Mode

The SPI operates in master mode when the MSTR bit is set. Only a master SPI module can initiate transmissions. A transmission begins by writing to the master SPI data register. If the shift register is empty, the byte immediately transfers to the shift register. The byte begins shifting out on the MOSI pin under the control of the serial clock.

The SPR2, SPR1, and SPR0 baud rate selection bits in conjunction with the SPPR2, SPPR1, and SPPR0 baud rate preselection bits in the SPI baud rate register control the baud rate generator and determine the speed of the shift register. The SCK pin is the SPI clock output. Through the SCK pin, the baud rate generator of the master controls the shift register of the slave peripheral.

In master mode, the function of the serial data output pin (MOSI) and the serial data input pin (MISO) is determined by the SPC0 and MSTR control bits.

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

Freescale Semiconductor, Inc.

The $\overline{SS}$ pin is normally an input which should remain in the inactive high state. However, in the master mode, if both MODFEN bit and SSOE bit are set, then the $\overline{SS}$ pin is the slave select output.

The $\overline{SS}$ output becomes low during each transmission and is high when the SPI is in the idling state. If the $\overline{SS}$ input becomes low while the SPI is configured as a master, it indicates a mode fault error where more than one master may be trying to drive the MOSI and SCK lines simultaneously. In this case, the SPI immediately clears the output buffer enables associated with the MISO, MOSI (or MOMI), and SCK pins so that these pins become inputs. This mode fault error also clears the MSTR control bit and sets the mode fault (MODF) flag in the SPI status register. If the SPI interrupt enable bit (SPIE) is set when the MODF bit gets set, then an SPI interrupt sequence is also requested

NOTE: *There is an errata information about the mode fault behavior. See MC9S12T64 Errata Sheet for details.*

When a write to the SPI data register in the master occurs, there is a half SCK-cycle delay. After the delay, SCK is started within the master. The rest of the transfer operation differs slightly, depending on the clock format specified by the SPI clock phase bit, CPHA, in SPI control register 1 (see Transmission Formats).

Slave Mode

The SPI operates in slave mode when the MSTR bit in SPI control register1 is clear. In slave mode, SCK is the SPI clock input from the master, and $\overline{SS}$ is the slave select input. Before a data transmission occurs, the $\overline{SS}$ pin of the slave SPI must be at logic 0. $\overline{SS}$ must remain low until the transmission is complete.

In slave mode, the function of the serial data output pin (MISO) and serial data input pin (MOSI) is determined by the SPC0 bit in SPI control register 2 and the MSTR control bit. While in slave mode, the $\overline{SS}$ input controls the serial data output pin; if $\overline{SS}$ is high (not selected), the serial data output pin is high impedance, and, if $\overline{SS}$ is low the first bit in the SPI data register is driven out of the serial data output pin. Also, if the slave is not selected ($\overline{SS}$ is high), then the SCK input is ignored and no internal shifting of the SPI shift register takes place.

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

Freescale Semiconductor, Inc.

Although the SPI is capable of duplex operation, some SPI peripherals are capable of only receiving SPI data in a slave mode. For these simpler devices, there is no serial data out pin.

NOTE: *When peripherals with duplex capability are used, take care not to simultaneously enable two receivers whose serial outputs drive the same system slave's serial data output line.*

As long as no more than one slave device drives the system slave's serial data output line, it is possible for several slaves to receive the same transmission from a master, although the master would not receive return information from all of the receiving slaves.

If the CPHA bit in SPI control register 1 is clear, odd numbered edges on the SCK input cause the data at the serial data input pin to be latched. Even numbered edges cause the value previously latched from the serial data input pin to shift into the LSB of the SPI shifter.

If the CPHA bit is set, even numbered edges on the SCK input cause the data at the serial data input pin to be latched. Odd numbered edges cause the value previously latched from the serial data input pin to shift into the LSB of the SPI shifter.

NOTE: *In slave mode, the control bits CPHA and CPOL of the SPI should be configured only when SPI is disabled else it may lead to incorrect data transfer. If the SPI is disabled, e.g. for configuration purpose, the dedicated SPI pins become GPIO pins. Care must be taken to avoid driver collisions. As recommended action, GPIO pins should be configured as high impedance inputs, before disabling the SPI.*

When CPHA is set, the first edge is used to get the first data bit onto the serial data output pin. When CPHA is clear and the $\overline{SS}$ input is low (slave selected), the first bit of the SPI data is driven out of the serial data output pin. After the eighth shift, the transfer is considered complete and the received data is transferred into the SPI data register. To indicate transfer is complete, the SPIF flag in the SPI status register is set.

NOTE: *There is an errata information about CPHA=1 transfer format in slave mode. See MC9S12T64 Errata Sheet for details.*

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

Freescale Semiconductor, Inc.

Transmission Formats

During an SPI transmission, data is transmitted (shifted out serially) and received (shifted in serially) simultaneously. The serial clock (SCK) synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows selection of an individual slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. Optionally, on a master SPI device, the slave select line can be used to indicate multiple-master bus contention.

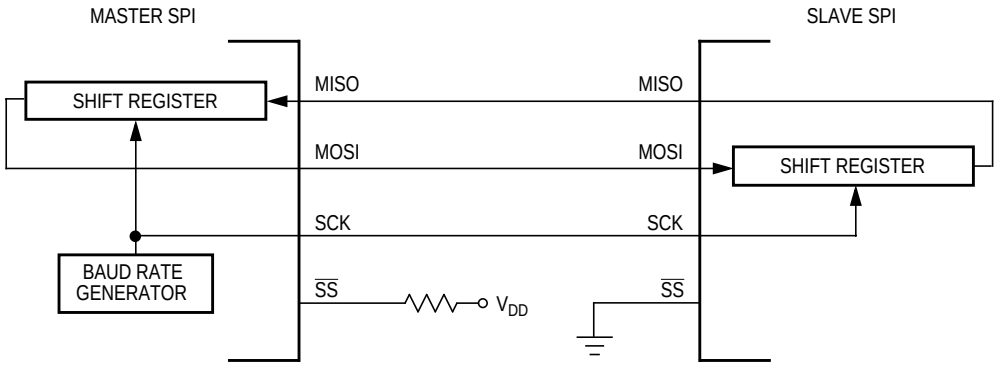


Figure 93 Master/Slave Transfer Block Diagram

Freescale Semiconductor, Inc.

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

Clock Phase and Polarity Controls

Using two bits in the SPI control register1, software selects one of four combinations of serial clock phase and polarity.

The CPOL clock polarity control bit specifies an active high or low clock and has no significant effect on the transmission format.

The CPHA clock phase control bit selects one of two fundamentally different transmission formats.

Clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

NOTE: *It is recommended that software writes to the SPI control register to change CPHA, CPOL or MSTR bits only in the idle state of the SPI. If these bits are changed during the transmission, data transmission gets corrupted.*

CPHA = 0 Transfer Format

The first edge on the SCK line is used to clock the first data bit of slave into the master and the first data bit of master into the slave. In some peripherals, the first bit of the slave's data is available at the slave data out pin as soon as the slave is selected. In this format, the first SCK edge is not issued until a half cycle into the 8-cycle transfer operation. The first edge of SCK is delayed a half cycle by clearing the CPHA bit.

The SCK output from the master remains in the inactive state for a half SCK period before the first edge appears. A half SCK cycle later, the second edge appears on the SCK line. When this second edge occurs, the value previously latched from the serial data input pin is shifted into the LSB of the shifter.

After this second edge, the next bit of the SPI master data is transmitted out of the serial data output pin of the master to the serial input pin on the slave. This process continues for a total of 16 edges on the SCK line, with data being latched on odd numbered edges and shifted on even numbered edges.

Freescale Semiconductor, Inc.

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

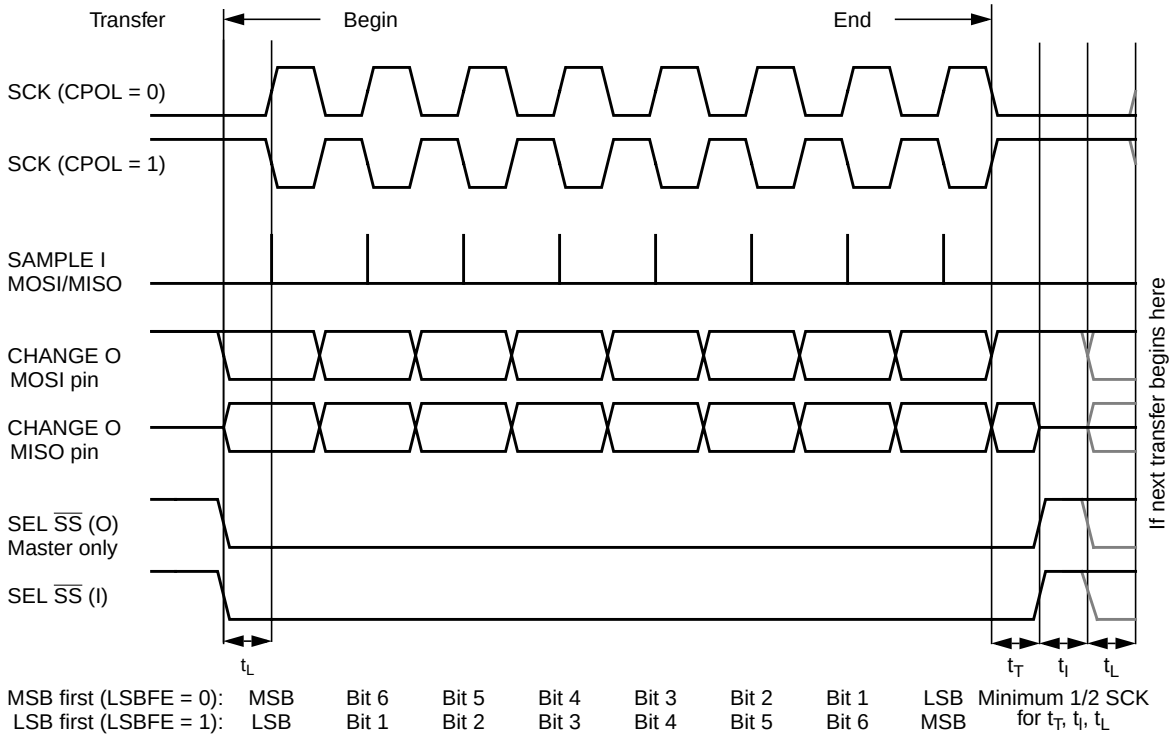
Data reception is double buffered. Data is shifted serially into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After the 16th (last) SCK edge:

- Data that was previously in the master SPI data register should now be in the slave data register and the data that was in the slave data register should be in the master.
- The SPIF flag in the SPI status register is set indicating that the transfer is complete.

Figure 94 is a timing diagram of an SPI transfer where CPHA = 0. SCK waveforms are shown for CPOL = 0 and CPOL = 1. The diagram may be interpreted as a master or slave timing diagram since the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave and the MOSI signal is the output from the master. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.

Freescale Semiconductor, Inc.



t_L = Minimum leading time before the first SCK edge
t_T = Minimum trailing time after the last SCK edge
t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time)
t_L, t_T, and t_I are guaranteed for the master mode and required for the slave mode.

Figure 94 SPI Clock Format 0 (CPHA = 0)

The $\overline{SS}$ line should be deasserted at least for minimum idle time (half SCK cycle) between the successive transfers ($\overline{SS}$ should not be tied low all the times in this mode). If $\overline{SS}$ is not deasserted between the successive transmission then the new byte written to the data register would not be transmitted, instead the last received byte would be transmitted.

CPHA = 1 Transfer Format

Some peripherals require the first SCK edge before the first data bit becomes available at the data out pin; the second edge clocks data into the system. In this format, the first SCK edge is issued by setting the CPHA bit at the beginning of the 8-cycle transfer operation.

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

The first edge of SCK occurs immediately after the half SCK clock cycle synchronization delay. This first edge commands the slave to transfer its most significant data bit to the serial data input pin of the master.

A half SCK cycle later, the second edge appears on the SCK pin. This is the latching edge for both the master and slave.

When the third edge occurs, the value previously latched from the serial data input pin is shifted into the LSB of the SPI shifter. After this edge, the next bit of the master data is coupled out of the serial data output pin of the master to the serial input pins on the slave.

This process continues for a total of 16 edges on the SCK line with data being latched on even numbered edges and shifting taking place on odd numbered edges.

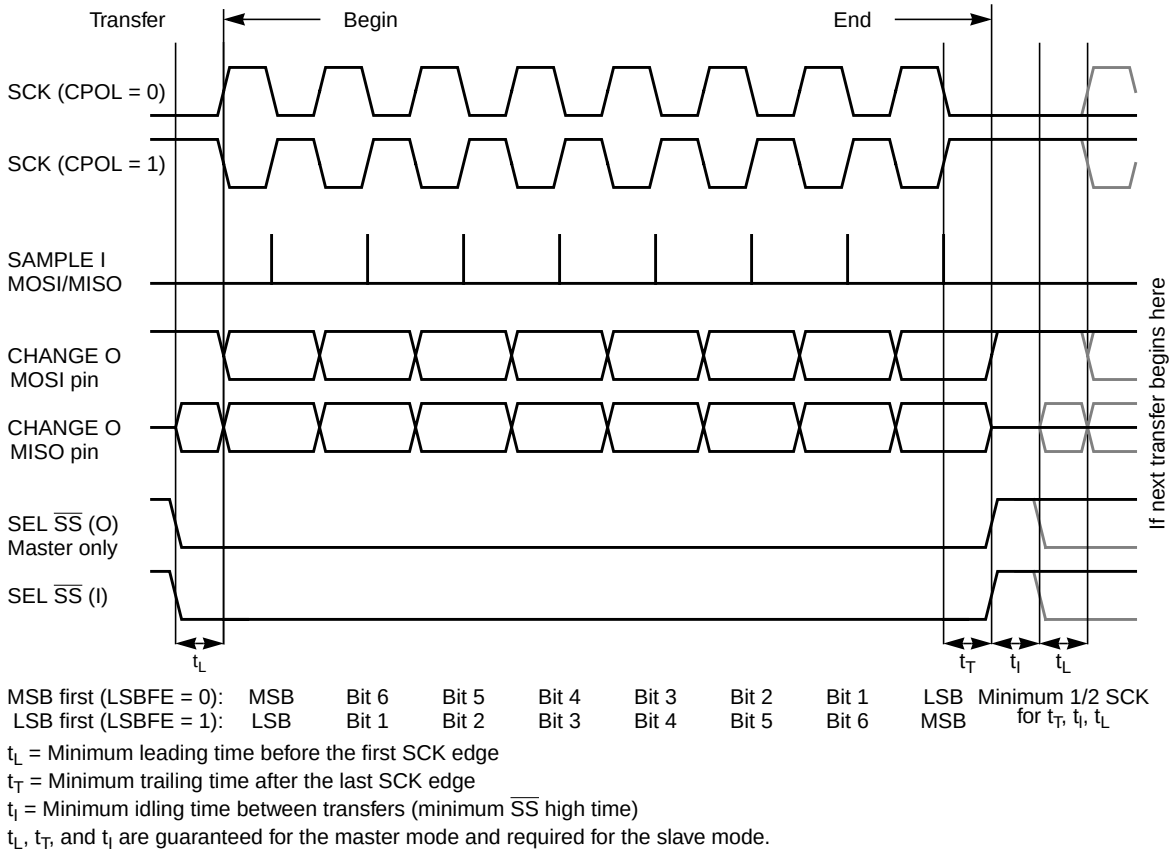
Data reception is double buffered; data is serially shifted into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After the 16th SCK edge:

- Data that was previously in the SPI data register of the master is now in the data register of the slave, and data that was in the data register of the slave is in the master.
- The SPIF flag bit in SPISR is set indicating that the transfer is complete.

Figure 95 shows two clocking variations for CPHA = 1. The diagram may be interpreted as a master or slave timing diagram since the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the slave select input to the slave. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.

Freescale Semiconductor, Inc.



t_L = Minimum leading time before the first SCK edge
t_T = Minimum trailing time after the last SCK edge
t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time)
t_L, t_T, and t_I are guaranteed for the master mode and required for the slave mode.

Figure 95 SPI Clock Format 1 (CPHA = 1)

The $\overline{SS}$ line can remain active low between successive transfers (can be tied low at all times). This format is sometimes preferred in systems having a single fixed master and a single slave that drive the MISO data line.

The SPI interrupt request flag (SPIF) is common to both the master and slave modes. SPIF gets set after the last SCK edge in a data transfer operation to indicate that the transfer is complete though transfer is actually complete half SCK cycle later.

SPI Baud Rate Generation

Baud rate generation consists of a series of divider stages. Six bits in the SPI baud rate register (SPPR2, SPPR1, SPPR0, SPR2, SPR1, and SPR0) determine the divisor to the bus clock which results in the SPI baud rate.

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

The SPI clock rate is determined by the product of the value in the baud rate preselection bits (SPPR2–SPPR0) and the value in the baud rate selection bits (SPR2–SPR0). The bus clock divisor equation is shown in Figure 96.

When all bits are clear (the default condition), the bus clock is divided by 2. When the selection bits (SPR2–SPR0) are 001 and the preselection bits (SPPR2–SPPR0) are 000, the bus clock divisor becomes 4. When the selection bits are 010, the bus clock divisor becomes 8, etc.

When the preselection bits are 001, the divisor determined by the selection bits is multiplied by 2. When the preselection bits are 010, the divisor is multiplied by 3, etc. See Table 83 for baud rate calculations for all bit conditions, based on a 16 MHz bus clock.The two sets of selects allows the clock to be divided by a non-power of two to achieve other baud rates such as divide by 6, divide by 10, etc.

The baud rate generator is activated only when the SPI is in the master mode and a serial transfer is taking place. In the other cases, the divider is disabled to decrease I_{DD} current.

$$BusClockDivisor = (SPPR + 1) \bullet 2^{(SPR + 1)}$$

Figure 96 Bus Clock Divisor Equation

Special Features

$\overline{SS}$ Output

The $\overline{SS}$ output feature automatically drives the $\overline{SS}$ pin low during transmission to select external devices and drives it high during idle to deselect external devices. When $\overline{SS}$ output is selected, the $\overline{SS}$ output pin is connected to the $\overline{SS}$ input pin of the external device.

The $\overline{SS}$ output is available only in master mode during normal SPI operation by asserting SSOE and MODFEN bit as shown in Table 81.

The mode fault feature is disabled while $\overline{SS}$ output is enabled.

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

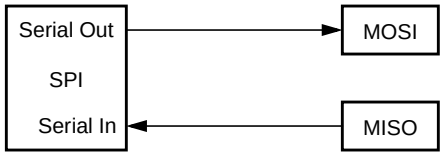
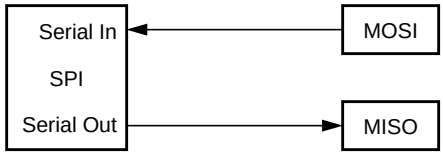
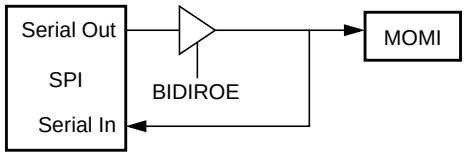
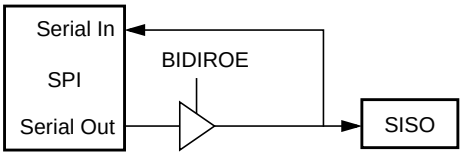
Freescale Semiconductor, Inc.

NOTE: *Care must be taken when using the $\overline{SS}$ output feature in a multimaster system since the mode fault feature is not available for detecting system errors between masters.*

Bidirectional Mode (MOMI or SISO)

The bidirectional mode is selected when the SPC0 bit is set in SPI control register 2 (see Normal Mode and Bidirectional Mode). In this mode, the SPI uses only one serial data pin for the interface with external device(s). The MSTR bit decides which pin to use. The MOSI pin becomes the serial data I/O (MOMI) pin for the master mode, and the MISO pin becomes serial data I/O (SISO) pin for the slave mode. The MISO pin in the master mode and MOSI pin in the slave mode are not used by the SPI in Bidirectional Mode.

Table 84 Normal Mode and Bidirectional Mode

When SPE = 1	Master Mode MSTR = 1	Slave Mode MSTR = 0
Normal Mode SPC0 = 0		
Bidirectional Mode SPC0 = 1		

The direction of each serial I/O pin depends on the BIDIROE bit. If the pin is configured as an output, serial data from the shift register is driven out on the pin. The same pin is also the serial input to the shift register.

The SCK is output for the master mode and input for the slave mode.

The $\overline{SS}$ is the input or output for the master mode, and it is always the input for the slave mode.

Freescale Semiconductor, Inc.

The bidirectional mode does not affect SCK and $\overline{SS}$ functions.

Error Conditions

The SPI has one error condition

- Mode fault error

Mode Fault Error

If the $\overline{SS}$ input becomes low while the SPI is configured as a master, it indicates a system error where more than one master may be trying to drive the MOSI and SCK lines simultaneously. This condition is not permitted in normal operation; the MODF bit in the SPI status register is set automatically provided the MODFEN bit is set.

In the special case where the MODFEN bit is cleared, the $\overline{SS}$ pin is a general purpose input/output pin for the SPI system configured in master mode. In this special case, the mode error function is inhibited and MODF remains cleared. In case the SPI system is configured as a slave, the $\overline{SS}$ pin is a dedicated input pin. Mode fault error doesn't occur in slave mode.

When a mode fault error occurs, the MSTR bit in control register SPICR1 is cleared, MODF bit in the status register is set and the output enable for the SCK, MISO and MOSI pins are de-asserted. So SCK, MISO and MOSI pins are forced to be high impedance inputs to avoid any possibility of conflict with another output driver.

NOTE: *There is an errata information about the mode fault behavior. See MC9S12T64 Errata Sheet for details.*

If the mode fault error occurs in the bidirectional mode for a SPI system configured in master mode, output enable of the MOMI (MOSI in bidirectional mode) is cleared if it was set but MISO (SISO) is not affected. No mode fault error occurs in the bidirectional mode for SPI system configured in slave mode.

The mode fault flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to SPI control register 1.

Freescale Semiconductor, Inc.

Low Power Mode Options

SPI in Run Mode

In run mode with the SPI system enable (SPE) bit in the SPI control register clear, the SPI system is in a low-power, disabled state. SPI registers can still be accessed, but clocks to the core of this module are disabled.

SPI in Wait Mode

SPI operation in wait mode depends upon the state of the SPISWAI bit in SPI control register 2.

- If SPISWAI is clear, the SPI operates normally when the CPU is in wait mode
- If SPISWAI is set, SPI clock generation ceases and the SPI module enters a power conservation state when the CPU is in wait mode.
  - If SPISWAI is set and the SPI is configured for master, any transmission and reception in progress stops at wait mode entry. The transmission and reception resumes when the SPI exits wait mode.
  - If SPISWAI is set and the SPI is configured as a slave, any transmission and reception in progress continues if the SCK continues to be driven from the master. This keeps the slave synchronized to the master and the SCK.

    If the master transmits several bytes while the slave is in wait mode, the slave will continue to send out bytes consistent with the its operation mode at the start of wait mode (i.e. If the slave is currently sending its SPIDR to the master, it will continue to send the same byte. Else if the slave is currently sending the last received byte from the master, it will continue to send each previous master byte).

NOTE: *Care must be taken when expecting data from a master while the slave is in wait or stop mode. Even though the shift register will continue to operate, the rest of the SPI is shut down (i.e. a SPIF interrupt will* ***not*** *be generated until exiting stop or wait mode). Also, the byte from the shift*

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

Freescale Semiconductor, Inc.

register will not be copied into the SPIDR register until after the slave SPI has exited wait or stop mode. A SPIF flag and SPIDR copy is only generated if wait mode is entered or exited during a transmission. If the slave enters wait mode in idle mode and exits wait mode in idle mode, neither a SPIF nor a SPIDR copy will occur.

SPI in Stop Mode

Stop mode is dependent on the system. The SPI enters stop mode when the bus clock is disabled (held high or low). If the SPI is in master mode and exchanging data when the CPU enters stop mode, the transmission is frozen until the CPU exits stop mode. After stop, data to and from the external SPI is exchanged correctly. In slave mode, the SPI will stay synchronized with the master.

The stop mode is equivalent to the wait mode with the SPISWAI bit set except that the stop mode is dependent on the system and cannot be controlled with the SPISWAI bit.

Reset Initialization

The reset values of registers and signals are described in Registers. All registers reset to a particular value.

- If a data transmission occurs in slave mode after reset without a write to SPIDR, it will transmit random data or the byte last received from the master before the reset.
- Reading from the SPIDR after reset will always read a byte of zeros.

Interrupts

This section describes interrupts originated by the Serial Peripheral Interface. The MCU must service the interrupt requests. Table 85 lists the three sources of interrupts generated by the SPI module. The SPI module communicates with the MCU through one interrupt port.

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

Freescale Semiconductor, Inc.

Table 85 SPI Interrupt Signals

Interrupt sources	Vector Address	Description
Mode Fault	$FFD8, $FFD9	Active high detect. MODF occurs when a logic zero occurs on the $\overline{SS}$ pin of a master SPI
Transfer Complete		Active high detect. SPIF occurs after the last SCK cycle in a data transfer operation to indicate that the transfer is complete.
SPI Transmitter Empty		Active high detect. SPTEF occurs when the SPI data register transfers a byte into the Transmit shift register.

Freescale Semiconductor, Inc.

Description of Interrupt Operation

The Serial Peripheral Interface only originates interrupt requests. The following is a description of how the Serial Peripheral Interface makes a request and how the MCU should acknowledge that request.

MODF Description

MODF occurs when the master detects an error on the $\overline{SS}$ pin. The master SPI must be configured for the MODF feature (see Table 81). Once MODF is set, the current transfer is halted and the following bit is changed:

- MSTR=0, The master bit in SPICR1 resets.

The MODF interrupt is reflected in the status register MODF flag. Clearing the flag will also clear the interrupt. This interrupt will stay active while the MODF flag is set. MODF has an automatic clearing process which is described in SPI Status Register (SPISR).

SPIF Description

SPIF occurs when the SPI receives/transmits the last SCK edge in a data transfer operation. Once SPIF is set, it does not clear until it is serviced. SPIF has an automatic clearing process which is described in SPI Status Register (SPISR). In the event that the SPIF is not serviced before the end of the next transfer (i.e. SPIF remains active throughout another transfer), the latter transfers will be ignored and no new data will be copied into the SPIDR.

SPTEF Description

SPTEF occurs when the SPI Data register transfers a byte into the transmit buffer Once SPTEF is set, it does not clear until it is serviced.

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

SPTEF has an automatic clearing process which is described in SPI Status Register (SPISR).

Freescale Semiconductor, Inc.

Analog to Digital Converter (ATD)

Contents

Overview 487
Features 487
Modes of Operation 488
Block Diagram 490
External Pin Descriptions 491
Register Map 492
Register Descriptions 494
Functional Description 511
Low Power Modes 514
Reset Initialization 515
Interrupts 515

Overview

The ATD module is an 8-channel, 10-bit, multiplexed input successive approximation analog-to-digital converter. Refer to device electrical specifications for ATD accuracy.

The block is designed to be upwards compatible with the 68HC11 standard 8-bit A/D converter. In addition, there are new operating modes that are unique to the HC12 design.

Features

- 8/10 Bit Resolution.
- 7 µsec, 10-Bit Single Conversion Time.

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

Freescale Semiconductor, Inc.

- Sample Buffer Amplifier.
- Programmable Sample Time.
- Left/Right Justified, Signed/Unsigned Result Data.
- External Trigger Control.
- Conversion Completion Interrupt Generation.
- Analog Input Multiplexer for 8 Analog Input Channels.
- Analog/Digital Input Pin Multiplexing.
- 1 to 8 Conversion Sequence Lengths.
- Continuous Conversion Mode.
- Multiple Channel Scans.

Modes of Operation

Conversion modes

There is software programmable selection between performing **single** or **continuous conversion** on a **single channel** or **multiple channels**.

The conversion modes for the ATD module are defined by the settings of three control bits and three control values. The control bits are ETRIGE in ATDCTL2, and MULT, SCAN in ATDCTL5. In brief, ETRIGE controls whether an external trigger is used to start a conversion sequence. MULT controls whether the sequence examines a single analog input channel or scans a number of different channels. SCAN determines if sequences are performed continuously. The control values are bits CC/CB/CA in ATDCTL5 which define the input channels to be examined; S8C/S4C/S2C/S1C in ATDCTL3 define the number of conversions in a sequence; SMP0/SMP1 in ATDCTL4 define the length of the sample time.

MCU Operating Modes

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

Freescale Semiconductor, Inc.

Run Mode

RUN mode for the ATD module is defined as the state where the ATD module is powered up and currently performing an A/D conversion. Complete assess to all control, status, and result registers is available. The module is consuming maximum power.

Wait Mode

Entering Wait Mode the ATD conversion either continues or aborts for low power depending on the logical value of the AWAIT bit in ATDCTL2 register.

Stop Mode

Entering Stop Mode causes all clocks to halt and thus the system is placed in a minimum power standby mode. This aborts any conversion sequence in progress. During recovery from Stop Mode, there must be a minimum delay for the Stop Recovery Time t_{SR} before initiating a new ATD conversion sequence.

Freeze Mode

The ATD module enters freeze mode when background debug mode (BDM) is active. In Freeze Mode the ATD module will behave according to the logical values of the FRZ1 and FRZ0 bits. This is useful for debugging and emulation.

Freescale Semiconductor, Inc.

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

Block Diagram

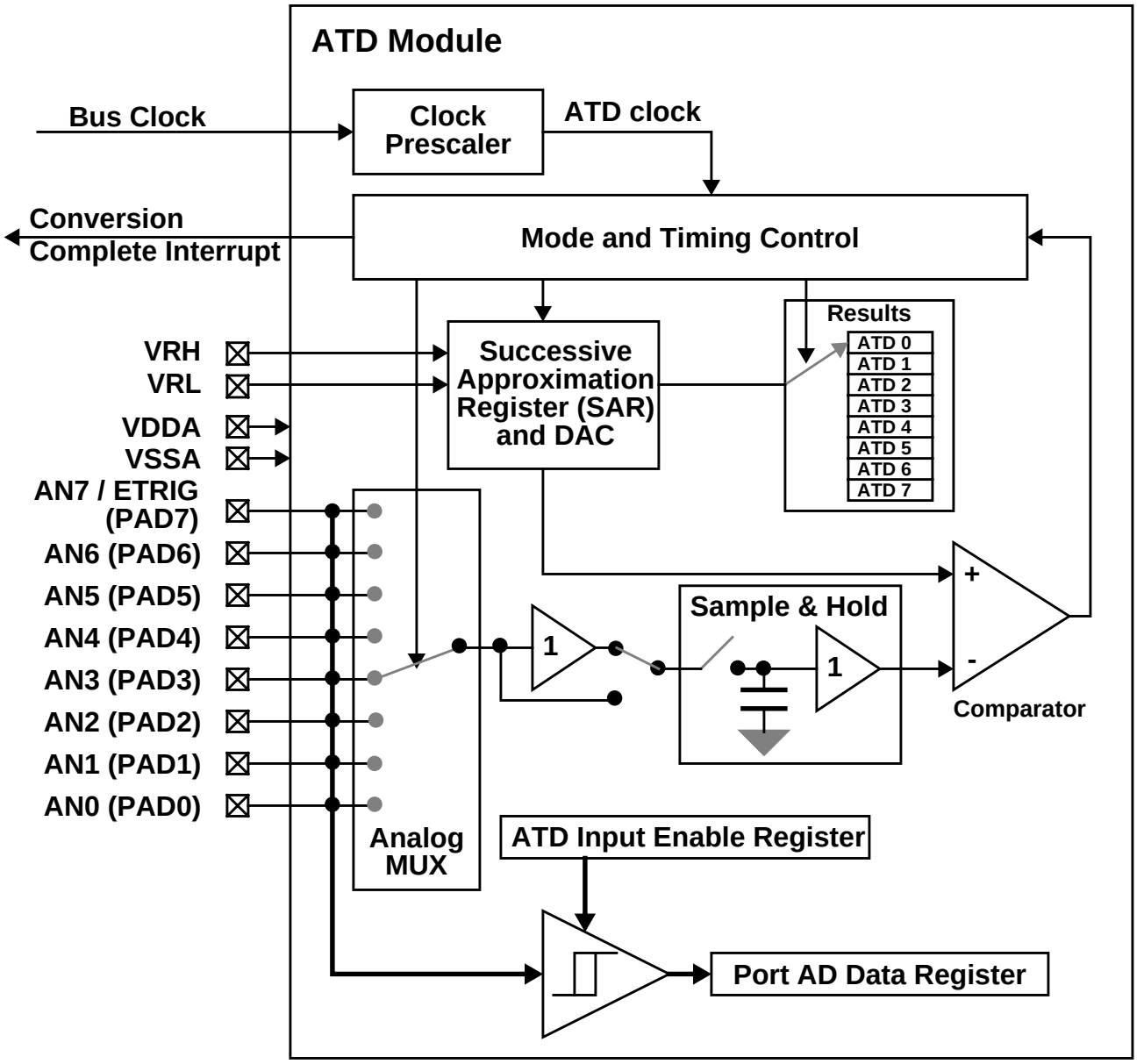


Figure 97 ATD Module Block Diagram

Freescale Semiconductor, Inc.

External Pin Descriptions

The ATD module has a total of 12 external pins.

AN7 / ETRIG (PAD7) This pin serves as the analog input Channel 7. It can be configured as an external trigger for the ATD conversion, or as general purpose digital input.

AN6 (PAD6) This pin serves as the analog input Channel 6. It can be configured as general purpose digital input.

AN5 (PAD5) This pin serves as the analog input Channel 5. It can be configured as general purpose digital input.

AN4 (PAD4) This pin serves as the analog input Channel 4. It can be configured as general purpose digital input.

AN3 (PAD3) This pin serves as the analog input Channel 3. It can be configured as general purpose digital input.

AN2 (PAD2) This pin serves as the analog input Channel 2. It can be configured as general purpose digital input.

AN1 (PAD1) This pin serves as the analog input Channel 1.It can be configured as general purpose digital input.

AN0 (PAD0) This pin serves as the analog input Channel 0. It can also be configured as general purpose digital input.

VRH, VRL VRH is the high reference voltage and VRL is the low reference voltage for ATD conversion.

Freescale Semiconductor, Inc.

VDDA, VSSA These pins are the power supplies for the analog circuitry of the ATD module block.

Register Map

Register Name		Bit 7	6	5	4	3	2	1	Bit 0	Address Offset
Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								$0080
	Write:									
Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								$0081
	Write:									
ATDCTL2	Read:	ADPU	AFFC	AWAI	ETRIGLE	ETRIGP	ETRIGE	ASCIE	ASCIF	$0082
	Write:									
ATDCTL3	Read:	0	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0	$0083
	Write:									
ATDCTL4	Read:	SRES8	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0	$0084
	Write:									
ATDCTL5	Read:	DJM	DSGN	SCAN	MULT	0	CC	CB	CA	$0085
	Write:									
ATDSTAT0	Read:	SCF	0	ETORF	FIFOR	0	CC2	CC1	CC0	$0086
	Write:									
Unimplemented	Read:	Reads to this register return unpredictable values.								$0087
	Write:									
Reserved for Factory Test	Read:	Reads to this register return unpredictable values.								$0088
	Write:									
ATDTEST1	Read:	Reads to these bits return unpredictable values.							SC	$0089
	Write:									
Unimplemented	Read:	Reads to this register return unpredictable values.								$008A
	Write:									
ATDSTAT1	Read:	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0	$008B
	Write:									
Unimplemented	Read:	Reads to this register return unpredictable values.								$008C
	Write:									
ATDDIEN	Read:	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0	$008D
	Write:									
Unimplemented	Read:	Reads to this register return unpredictable values.								$008E
	Write									

= Unimplemented or Reserved

Figure 98 ATD Register Map

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

Freescale Semiconductor, Inc.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0	Address Offset
PORTAD	Read:	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0	$008F
	Write:									
ATDDR0H	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	$0090
	Write:									
ATDDR0L	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	$0091
	Write:									
ATDDR1H	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	$0092
	Write:									
ATDDR1L	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	$0093
	Write:									
ATDDR2H	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	$0094
	Write:									
ATDDR2L	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	$0095
	Write:									
ATDDR3H	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	$0096
	Write:									
ATDDR3L	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	$0097
	Write:									
ATDDR4H	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	$0098
	Write:									
ATDDR4L	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	$0099
	Write:									
ATDDR5H	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	$009A
	Write:									
ATDDR5L	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	$009B
	Write:									
ATDDR6H	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	$009C
	Write:									
ATDDR6L	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	$009D
	Write:									
ATDDR7H	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	$009E
	Write:									
ATDDR7L	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	$009F
	Write:									

= Unimplemented or Reserved

Figure 98 ATD Register Map (Continued)

NOTE: *Register address: Base address (INITRG) + Address Offset*

NOTE: *Writing to Reserved Registers during special modes can alter functionality.*

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

Freescale Semiconductor, Inc.

Register Descriptions

The following subsections describe the bit-level arrangement and functionality of each register.

ATD Control Register 2 (ATDCTL2)

This register controls power down, interrupt and external trigger. Writes to this register will abort current conversion sequence but will not start a new sequence.

Address Offset: $0082

	Bit 15	14	13	12	11	10	9	Bit 8
Read:	ADPU	AFFC	AWAI	ETRIGLE	ETRIGP	ETRIGE	ASCIE	ASCIF
Write:								
Reset:	0	0	0	0	0	0	0	0

READ: anytime
WRITE: anytime
(except for Bit 8 – ASCIF, READ: any time, WRITE: not allowed)

ADPU — ATD Power Down

This bit provides on/off control over the ATD module block allowing reduced MCU power consumption. Because analog electronic is turned off when powered down, the ATD requires a recovery time period after ADPU bit is enabled.

1 = Normal ATD functionality
0 = Power down ATD

AFFC — ATD Fast Flag Clear All

1 = Changes all ATD conversion complete flags to a fast clear sequence. Any access to a result register will cause the associate CCF flag to clear automatically.
0 = ATD flag clearing operates normally (read the status register ATDSTAT1 before reading the result register to clear the associate CCF flag).

AWAI — ATD Power Down in Wait Mode

Freescale Semiconductor, Inc.

When entering Wait Mode this bit provides on/off control over the ATD module block allowing reduced MCU power. Because analog electronic is turned off when powered down, the ATD requires a recovery time period after exit from Wait mode.

1 = Power down ATD during Wait mode
0 = ATD continues to run in Wait mode

NOTE: *Although a conversion that was interrupted prior to entering WAIT mode is restarted after WAIT mode wake-up, the results are not reliable and should not be used by the user application.*

ETRIGLE — External Trigger Level/Edge Control

This bit controls the sensitivity of the external trigger signal. See Table 86 for details.

ETRIGP — External Trigger Polarity

This bit controls the polarity of the external trigger signal. See Table 86 for details.

Table 86 External Trigger Configurations

ETRIGLE	ETRIGP	External Trigger Sensitivity
0	0	falling edge
0	1	rising edge
1	0	low level
1	1	high level

ETRIGE — External Trigger Mode Enable

This bit enables the external trigger on ATD channel 7. The external trigger allows to synchronize sample and ATD conversions processes with external events.

1 = Enable external trigger
0 = Disable external trigger

NOTE: *The conversion results for the external trigger ATD channel 7 have no meaning while external trigger mode is enabled.*

ASCIE — ATD Sequence Complete Interrupt Enable

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

Freescale Semiconductor, Inc.

1 = ATD Interrupt will be requested whenever ASCIF=1 is set.
0 = ATD Sequence Complete interrupt requests are disabled.

NOTE: *There is an errata information about the ASCIF flag. See MC9S12 Errata Sheet for details.*

ASCIF — ATD Sequence Complete Interrupt Flag

If ASCIE=1 the ASCIF flag equals the SCF flag (see register ATDSTAT0 - page 505), else ASCIF reads zero. Writes have no effect.

1 = ATD sequence complete interrupt pending
0 = No ATD interrupt occurred

ATD Control Register 3 (ATDCTL3)

This register controls the conversion sequence length, FIFO for results registers and behavior in Freeze Mode. Writes to this register will abort current conversion sequence but will not start a new sequence.

Address Offset: $0083

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
Write:								
Reset:	0	0	1	0	0	0	0	0

Unimplemented or Reserved

S8C/S4C/S2C/S1C — Conversion Sequence Length

These bits control the number of conversions per sequence. Table 87 shows all combinations. At reset, S4C is set to 1 (sequence length is 4). This is to maintain software continuity to HC12 family.

Table 87 Conversion Sequence Length Coding.

S8C	S4C	S2C	S1C	Number of Conversions per Sequence
0	0	0	0	8
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4

Freescale Semiconductor, Inc.

Table 87 Conversion Sequence Length Coding. (Continued)

S8C	S4C	S2C	S1C	Number of Conversions per Sequence
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	X	X	X	8

FIFO — Result Register FIFO Mode

If this bit is zero (non-FIFO mode), the A/D conversion results map into the result registers based on the conversion sequence; the result of the first conversion appears in the first result register, the second result in the second result register, and so on.

If this bit is one (FIFO mode) the conversion counter is not reset at the beginning or end of a conversion sequence; conversion results are placed in consecutive result registers between sequences. The result register counter wraps around when it reaches the end of the result register file. The conversion counter value in ATDSTAT0 can be used to determine where in the result register file, the current conversion result will be placed.

Finally, which result registers hold valid data can be tracked using the conversion complete flags. Fast flag clear mode may or may not be useful in a particular application to track valid data.

1 = Conversion results are placed in consecutive result registers (wrap around at end).

0 = Conversion results are placed in the corresponding result register up to the selected sequence length.

FRZ1, FRZ0 — Background Debug Freeze Enable

When debugging an application, it is useful in many cases to have the ATD pause when a breakpoint (Freeze Mode) is encountered. These 2 bits determine how the ATD will respond to a breakpoint as shown in Table 88. Leakage onto the storage node and comparator

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

Freescale Semiconductor, Inc.

reference capacitors may compromise the accuracy of an immediately frozen conversion depending on the length of the freeze period.

Table 88 ATD Behavior in Freeze Mode (Breakpoint)

FRZ1	FRZ0	ATD RESPONSE
0	0	Continue conversion
0	1	Reserved
1	0	Finish current conversion, then freeze
1	1	Freeze Immediately

NOTE: *The ATD module enters freeze mode when background debug mode (BDM) is active. Refer to the Fast Background Debug Module (FBDM) section about the background debug mode.*

Freescale Semiconductor, Inc.

ATD Control Register 4 (ATDCTL4)

This register selects the conversion clock frequency, the length of the second phase of the sample time and the resolution of the A/D conversion (i.e.: 8-bits or 10-bits). Writes to this register will abort current conversion sequence but will not start a new sequence.

Address Offset: $0084

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SRES8	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0
Reset:	0	0	0	0	0	1	0	1

Unimplemented or Reserved

SRES8 — A/D Resolution Select

1 = 8-bit resolution selected.

0 = 10-bit resolution selected.

This bit selects the resolution of A/D conversion results as either 8 or 10 bits. The A/D converter has an accuracy of 10 bits. However, if low resolution is required, the conversion can be speeded up by selecting 8-bit resolution.

SMP0, SMP1 — Sample Time Select

These two bits select the length of the second phase of the sample time in units of ATD conversion clock cycles. Note that the ATD conversion clock period is itself a function of the prescaler value (bits PRS4-0). The sample time consists of two phases. The first phase is two ATD conversion clock cycles long and transfers the sample quickly (via the buffer amplifier) onto the A/D machine's storage node. The second phase attaches the external analog signal directly to the storage node for final charging and high accuracy. Table 89 lists the lengths available for the second sample phase.

Table 89 Sample Time Select

SMP1	SMP0	Length of Second Phase of Sample Time
0	0	2 A/D clock periods
0	1	4 A/D clock periods
1	0	8 A/D clock periods
1	1	16 A/D clock periods

Freescale Semiconductor, Inc.

PRS0, PRS1, PRS2, PRS3, PRS4 — ATD Clock Prescaler

These 5 bits are the binary value prescaler value PRS. The ATD conversion clock frequency is calculated as follows:

$$ATDconversionclock = \frac{[BusClock]}{[PRS + 1]} \times 0.5$$

Note that the maximum ATD conversion clock frequency is half the Bus Clock. The default (after reset) prescaler value is 5 which results in a default ATD conversion clock frequency that is Bus Clock divided by 12. Table 90 illustrates the divide-by operation and the appropriate range of bus clock.

Freescale Semiconductor, Inc.

Table 90 Clock Prescaler Values

Prescale Value	Total Divisor Value	Max Bus Clock[1]	Min Bus Clock[2]
00000	divide by 2	4 MHz	1 MHz
00001	divide by 4	8 MHz	2 MHz
00010	divide by 6	12 MHz	3 MHz
00011	divide by 8	16 MHz	4 MHz
00100	divide by 10	20 MHz	5 MHz
00101	divide by 12	24 MHz	6 MHz
00110	divide by 14	28 MHz	7 MHz
00111	divide by 16	32 MHz	8 MHz
01000	divide by 18	36 MHz	9 MHz
01001	divide by 20	40 MHz	10 MHz
01010	divide by 22	44 MHz	11 MHz
01011	divide by 24	48 MHz	12 MHz
01100	divide by 26	52 MHz	13 MHz
01101	divide by 28	56 MHz	14 MHz
01110	divide by 30	60 MHz	15 MHz
01111	divide by 32	64 MHz	16 MHz
10000	divide by 34	68 MHz	17 MHz
10001	divide by 36	72 MHz	18 MHz
10010	divide by 38	76 MHz	19 MHz
10011	divide by 40	80 MHz	20 MHz
10100	divide by 42	84 MHz	21 MHz
10101	divide by 44	88 MHz	22 MHz
10110	divide by 46	92 MHz	23 MHz
10111	divide by 48	96 MHz	24 MHz
11000	divide by 50	100 MHz	25 MHz
11001	divide by 52	104 MHz	26 MHz
11010	divide by 54	108 MHz	27 MHz
11011	divide by 56	112 MHz	28 MHz
11100	divide by 58	116 MHz	29 MHz
11101	divide by 60	120 MHz	30 MHz
11110	divide by 62	124 MHz	31 MHz
11111	divide by 64	128 MHz	32 MHz

1. Maximum ATD conversion clock frequency is 2MHz. The max. Bus clock frequency is computed from the max. ATD clock frequency times the indicated prescaler setting.

2. Minimum ATD clock frequency is 500KHz. The min. Bus clock frequency is computed from the min. ATD clock frequency times the indicated prescaler settling.

Freescale Semiconductor, Inc.

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

ATD Control Register 5 (ATDCTL5)

This register selects the type of conversion sequence and the analog input channels sampled. Writes to this register will abort current conversion sequence and start a new conversion sequence.

Address Offset: $0085

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	DJM	DSGN	SCAN	MULT	0	CC	CB	CA
Write:								
Reset:	0	0	0	0	0	0	0	0

Unimplemented or Reserved

Read: anytime

Write: anytime

DJM — Result Register Data Justification Mode

1 = Right justified mode.

0 = Left justified mode.

This bit controls justification of conversion data in the result registers. See *ATDDRx A/D Conversion Result Registers (ATDDR0–7)* in page 510 for details.

DSGN — Signed/Unsigned Result Data Mode

1 = Signed result register data select.

0 = Unsigned result register data select.

This bit selects between signed and unsigned conversion data representation in the result registers. Signed data is represented as 2’s complement. See *ATDDRx A/D Conversion Result Registers (ATDDR0–7)* in page 510 for details.

Table 91 summarizes the result data formats available and how they are set up using the control bits.

Table 92 illustrates the difference between the signed and unsigned, left justified output codes for an input signal range between 0 and 5.12 Volts.

Freescale Semiconductor, Inc.

Table 91 Result Data Formats Available.

SRES8	DJM	DSGN	Result Data Formats Description and Bus Bit Mapping
1	0	0	8-bit/left justified/unsigned - bits 8-15
1	0	1	8-bit/ left justified/signed - bits 8-15
1	1	X	8-bit/right justified/unsigned - bits 0-7
0	0	0	10-bit/left justified/unsigned - bits 6-15
0	0	1	10-bit/left justified/signed - bits 6-15
0	1	X	10-bit/right justified/unsigned - bits 0-9

Table 92 Left Justified, Signed and Unsigned ATD Output Codes.

Input Signal Vrl = 0 Volts Vrh = 5.12 Volts	Signed 8-Bit Codes	Unsigned 8-Bit Codes	Signed 10-Bit Codes	Unsigned 10-Bit Codes
5.120 Volts	7F	FF	7FC0	FFC0
5.100	7F	FF	7F00	FF00
5.080	7E	FE	7E00	FE00
2.580	01	81	0100	8100
2.560	00	80	0000	8000
2.540	FF	7F	FF00	7F00
0.020	81	01	8100	0100
0.000	80	00	8000	0000

SCAN — Continuous Conversion Sequence Mode

1 = Perform conversion sequences continuously.

0 = Perform a conversion sequence and return to idle mode.

The scan mode bit controls whether or not conversion sequences are performed continuously or only once.

MULT — Multi-Channel Sample Mode

1 = Sample across many channels.

0 = Sample only the specified channel.

When MULT is 0, the ATD sequence controller samples only from the specified analog input channel for an entire conversion sequence. The analog channel is selected by channel selection code (control bits CC/CB/CA located in ATDCTL5). When MULT is 1, the ATD sequence controller samples across channels. The number of

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

Freescale Semiconductor, Inc.

channels sampled is determined by the sequence length value (S8C, S4C, S2C, S1C). The first analog channel examined is determined by channel selection code (CC, CB, CA control bits); subsequent channels sampled in the sequence are determined by incrementing the channel selection code.

CC, CB, CA — Analog Input Channel Select Code

These bits select the analog input channel(s) whose signals are sampled and converted to digital codes. Table 93 lists the coding used to select the various analog input channels. In the case of single channel scans (MULT=0), this selection code specifies the channel to be examined. In the case of multi-channel scans (MULT=1), this selection code represents the first channel to be examined in the conversion sequence. Subsequent channels are determined by incrementing channel selection code; selection codes that reach the maximum value wrap around to the minimum value.

Table 93 Analog Input Channel Select Coding

CC	CB	CA	Analog Input Channel
0	0	0	AN0
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4
1	0	1	AN5
1	1	0	AN6
1	1	1	AN7

Freescale Semiconductor, Inc.

A/D Status Register (ATDSTAT0)

This read-only register contains the Sequence Complete Flag, overrun flags for external trigger and FIFO mode, and the conversion counter.

Address Offset: $0086

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SCF	0	ETORF	FIFOR	0	CC2	CC1	CC0
Write:								
Reset:	0	0	0	0	0	0	0	0

Read: Anytime

Write: Refer to each bit for individual write conditions

NOTE: *There is an errata information about the flags SCF, ETORF and FIFOR. See MC9S12T64 Errata Sheet for details.*

NOTE: *There is an errata information about the SCF flag. See MC9S12T64 Errata Sheet for details.*

SCF — Sequence Complete Flag

This flag is set upon completion of a conversion sequence. If conversion sequences are continuously performed (SCAN=1), the flag is set after each one is completed. This flag is cleared when one of the following occurs:
A) Write “1” to SCF
B) Write to ATDCTL5 (a new conversion sequence is started)
C) If AFFC=1 and read of a result register
1 = Conversion sequence has completed
0 = Conversion sequence not completed

ETORF — External Trigger Overrun Flag

While in edge trigger mode (ETRIGLE=0), if additional active edges are detected while a conversion sequence is in process the overrun flag is set. This flag is cleared when one of the following occurs:
A) Write “1” to ETORF
B) Write to ATDCTL2, ATDCTL3 or ATDCTL4 (a conversion sequence is aborted)
C) Write to ATDCTL5 (a new conversion sequence is started)

Freescale Semiconductor, Inc.

1 = External trigger overrun error has occurred
0 = No External trigger overrun error has occurred

FIFOR — FIFO Over Run Flag.

This bit indicates that a result register has been written to before its associated conversion complete flag (CCF) has been cleared. This flag is most useful when using the FIFO mode because the flag potentially indicates that result registers are out of sync with the input channels. However, it is also practical for non-FIFO modes, and indicates that a result register has been over written before it has been read (i.e. the old data has been lost). This flag is cleared when one of the following occurs:
A) Write “1” to FIFOR
B) Write to ATDCTL5 (a new conversion sequence is started)
1 = An over run condition exists
0 = No over run has occurred

CC2/CC1/CC0 — Conversion Counter

These 3 read-only bits are the binary value of the conversion counter. The conversion counter points to the result register that will receive the result of the current conversion. E.g. CC2=1, CC1=1, CC0=0 indicates that the result of the current conversion will be in ATD Result Register 6. If in non-FIFO mode (FIFO=0) the conversion counter is initialized to zero at the begin and end of the conversion sequence. If in FIFO mode (FIFO=1) the register counter is not initialized. The conversion counters wraps around when its maximum value is reached.

ATD Test Register 1 (ATDTEST1)

This register contains the SC bit used to enable special channel conversions.

Address Offset: $0089

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Reads to these bits return unpredictable values.							SC
Write:								
Reset:	0	0	0	0	0	0	0	0

Unimplemented or Reserved

Freescale Semiconductor, Inc.

Read: anytime

Write: anytime

SC - Special Channel Conversion Bit

If this bit is set, then special channel conversion can be selected using CC, CB, and CA of ATDCTL5 register. Table 94 lists the coding.
1 = Special channel conversions enabled
0 = Special channel conversions disabled

CAUTION: *Always write remaining bits of ATDTEST1 (Bits 7 to 1) zero when writing the SC bit. Not doing so might result in unpredictable ATD behavior.*

Table 94 Special Channel Select Coding

SC	CC	CB	CA	Analog Input Channel
1	0	X	X	Reserved
1	1	0	0	V_{RH}
1	1	0	1	V_{RL}
1	1	1	0	$(V_{RL}+V_{RH})/2$
1	1	1	1	Reserved

A/D Status Register (ATDSTAT1)

This read-only register contains the Conversion Complete Flags.

Address Offset: $008B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
Write:								
Reset:	0	0	0	0	0	0	0	0

Unimplemented or Reserved

Read: anytime

Write: only in special modes

NOTE: *There is an errata information about the CCF flags. See MC9S12T64 Errata Sheet for details.*

Freescale Semiconductor, Inc.

CCFx — Conversion Complete Flag x (x=7,6,5,4,3,2,1,0)

A conversion complete flag is set at the end of each conversion in a conversion sequence. The flags are associated with the conversion position in a sequence (and also the result register number). Therefore, CCF0 is set when the first conversion in a sequence is complete and the result is available in result register ATDDR0; CCF1 is set when the second conversion in a sequence is complete and the result is available in ATDDR1, and so forth. A flag CCFx (x=7,6,5,4,3,2,1,0) is cleared when one of the following occurs:
A) Write to ATDCTL5 (a new conversion sequence is started)
B) If AFFC=0 and read of ATDSTAT1 followed by read of result register ATDDRx
C) If AFFC=1 and read of result register ATDDRx

1 = Conversion number x has completed, result ready in ATDDRx
0 = Conversion number x not completed

ATD Input Enable Mask Register (ATDDIEN)

Address Offset: $008D

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	IEN 7	IEN 6	IEN 5	IEN 4	IEN 3	IEN 2	IEN 1	IEN 0
Reset:	0	0	0	0	0	0	0	0

Read: anytime

Write: anytime

IENx— ATD Digital Input Enable on channel x (x=7,6,5,4,3,2,1,0)

This bit controls the digital input buffer from the analog input pin (ANx) to PTADx data register.

1 = Enable digital input buffer to PTADx.
0 = Disable digital input buffer to PTADx

NOTE: *Setting this bit will enable the corresponding digital input buffer continuously. If this bit is set while simultaneously using it as an analog port, there is potentially increased power consumption because the digital input buffer maybe in the linear region.*

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

Freescale Semiconductor, Inc.

Port Data Register (PORTAD)

The data port associated with the ATD is input-only. The port pins are shared with the analog A/D inputs AN7-0.

Address Offset: $008F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTAD 7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
Write:								
Pin Function	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
Reset:	1	1	1	1	1	1	1	1
		Unimplemented or Reserved						

Read: anytime. If the digital input buffers are disabled (IENx=0), read returns a 1.

Write: anytime, no effect

The A/D input channels may be used for general purpose digital input.

PTADx — A/D Channel x (ANx) Digital Input (x= 7,6,5,4,3,2,1,0)

If the digital input buffer on the ANx pin is enabled (IENx=1) read returns the logic level on ANx pin (signal potentials not meeting V_{IL} or V_{IH} specifications will have an indeterminate value)).

If the digital input buffers are disabled (IENx=0), read returns a “1”.

Reset sets all PORTAD bits to “1”.

Freescale Semiconductor, Inc.

ATDDRx A/D Conversion Result Registers (ATDDR0–7)

Where: x = 0, 1, 2, 3, 4, 5, 6, 7

Left Justified Result Data

Address Offset: $0090, $0092, $0094, $0096, $0098, $009A, $009C, $009E

	ATDDRxH	Bit 15	14	13	12	11	10	9	Bit 8
10-bit data	Read:	Bit 9 MSB	8	7	6	5	4	3	Bit 2
	Write:								
8-bit data	Read:	Bit 7 MSB	6	5	4	3	2	1	Bit 0
	Write:								
	Reset	U	U	U	U	U	U	U	U

Address Offset: $0091, $0093, $0095, $0097, $0099, $009B, $009D, $009F

	ATDDRxL	Bit 7	6	5	4	3	2	1	Bit 0
10-bit data	Read:	Bit 1	Bit 0	0	0	0	0	0	0
	Write:								
8-bit data	Read:	U	U	0	0	0	0	0	0
	Write:								
	Reset	U	U	U	U	U	U	U	U

Right Justified Result Data

Address Offset: $0090, $0092, $0094, $0096, $0098, $009A, $009C, $009E

	ATDDRxH	Bit 15	14	13	12	11	10	9	Bit 8
10-bit data	Read:	0	0	0	0	0	0	Bit 9 MSB	Bit 8
	Write:								
8-bit data	Read:	0	0	0	0	0	0	0	0
	Write:								
	Reset	At reset the data format is left justified!							

Address Offset: $0091, $0093, $0095, $0097, $0099, $009B, $009D, $009F

	ATDDRxL	Bit 7	6	5	4	3	2	1	Bit 0
10-bit data	Read:	Bit 7	6	5	4	3	2	1	Bit 0
	Write:								
8-bit data	Read:	Bit 7 MSB	6	5	4	3	2	1	Bit 0
	Write:								
	Reset	At reset the data format is left justified!							

Unimplemented or Reserved

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

Freescale Semiconductor, Inc.

Read: Anytime

Write: Anytime, no effect.

The A/D conversion results are stored in 8 read-only result registers. The result data is formatted in the result registers based on two criteria. First there is left and right justification; this selection is made using the DJM control bit in ATDCTL5. Second there is signed and unsigned data; this selection is made using the DSGN control bit in ATDCTL5. Signed data is stored in 2's complement format and only exists in left justified format. Signed data selected for right justified format is ignored.

For 8-bit result data, the result data maps between the high (left justified) and low (right justified) order bytes of the result register. For 10-bit result data, the result data maps between bits 6–15 (left justified) and bits 0–9 (right justified) of the result register. Therefore for each bit in the SAR, there are 3 possible mappings of this bit into the result registers.

Functional Description

General

The ATD module is structured in an analog and a digital sub-block.

Analog Sub-block

The analog sub-block contains all analog electronics required to perform a single conversion. Separate power supplies VDDA and VSSA allow to isolate noise of other MCU circuitry from the analog sub-block.

Sample and Hold Machine

The Sample and Hold (S/H) Machine accepts analog signals from the external surroundings and stores them as capacitor charge on a storage node.

The sample process uses a two stage approach. During the first stage, the sample amplifier is used to quickly charge the storage node. The second stage connects the input directly to the storage node to complete the sample for high accuracy.

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

Freescale Semiconductor, Inc.

When not sampling, the sample and hold machine disables its own clocks. The analog electronics still draw their quiescent current. The power down (ADPU) bit must be set to disable both the digital clocks and the analog power consumption.

The input analog signals are unipolar and must fall within the potential range of VSSA to VDDA.

Analog Input Multiplexer

The analog input multiplexer connects one of the 8 external analog input channels to the sample and hold machine.

Sample Buffer Amplifier

The sample amplifier is used to buffer the input analog signal so that the storage node can be quickly charged to the sample potential.

Analog-to-Digital (A/D) Machine

The A/D Machine performs analog to digital conversions. The resolution is program selectable at either 8 or 10 bits. The A/D machine uses a successive approximation architecture. It functions by comparing the stored analog sample potential with a series of digitally generated analog potentials. By following a binary search algorithm, the A/D machine locates the approximating potential that is nearest to the sampled potential.

When not converting the A/D machine disables its own clocks. The analog electronics still draws quiescent current. The power down (ADPU) bit must be set to disable both the digital clocks and the analog power consumption.

Only analog input signals within the potential range of V_{RL} to V_{RH} (A/D reference potentials) will result in a non-railed digital output codes.

Digital Sub-block

This subsection explains some of the digital features in more detail. See register descriptions for all details.

External Trigger Input (ETRIG)

The external trigger feature allows the user to synchronize ATD conversions to the external environment events rather than relying on software to signal the ATD module when ATD conversions are to take place. The input signal (ATD channel 7) is programmable to be edge or level sensitive with polarity control.

Freescale Semiconductor, Inc.

Table 95 gives a brief description of the different combinations of control bits and their affect on the external trigger function.

Table 95 External Trigger Control Bits

ETRIGLE	ETRIGP	ETRIGE	SCAN	Description
X	X	0	0	Ignores external trigger. Performs one conversion sequence and stops.
X	X	0	1	Ignores external trigger. Performs continuous conversion sequences.
0	0	1	X	Falling edge triggered. Performs one conversion sequence per trigger.
0	1	1	X	Rising edge triggered. Performs one conversion sequence per trigger.
1	0	1	X	Trigger active low. Performs continuous conversions while trigger is active.
1	1	1	X	Trigger active high. Performs continuous conversions while trigger is active.

During a conversion, if additional active edges are detected the overrun error flag ETORF is set.

In either level or edge triggered modes, the first conversion begins when the trigger is received. In both cases, the maximum latency time is one Bus Clock cycle plus any skew or delay introduced by the trigger circuitry.

NOTE: *The conversion results for the external trigger ATD channel 7 have no meaning while external trigger mode is enabled.*

Once ETRIGE is enabled, conversions cannot be started by a write to ATDCTL5, but rather must be triggered externally.

If the level mode is active and the external trigger both de-asserts and re-asserts itself during a conversion sequence, this does not constitute an overrun. Therefore, the flag is not set. If the trigger is left asserted in

Freescale Semiconductor, Inc.

level mode while a sequence is completing, another sequence will be triggered immediately.

General Purpose Digital Input Port Operation

The input channel pins can be multiplexed between analog and digital data. As analog inputs, they are multiplexed and sampled to supply signals to the A/D converter. As digital inputs, they supply external input data that can be accessed through the digital port register PORTAD (input-only).

The analog/digital multiplex operation is performed in the input pads. The input pad is always connected to the analog inputs of the ATD module. The input pad signal is buffered to the digital port registers. This buffer can be turned on or off with the ATDDIEN register. This is important so that the buffer does not draw excess current when analog potentials are presented at its input.

Low Power Modes

The ATD module can be configured for lower MCU power consumption in 3 different ways:

- Stop Mode
- Wait Mode with AWAI=1
- Power down by writing ADPU=0 (Note that all ATD registers remain accessible.)

Note that the reset value for the ADPU bit is zero. Therefore, when this module is reset, it is reset into the power down state. Once the ATD module is configured for low power, it aborts any conversion sequence in progress. When ATD module powers up again (exit Stop Mode, exit Wait Mode with AWAI=1 or set ADPU=1), it requires a recovery time period.

Freescale Semiconductor, Inc.

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

Reset Initialization

The reset state of each individual bit is listed within the Register Description section (see Register Descriptions) which details the registers and their bit-fields.

Interrupts

The interrupt requested by the ATD module is listed in Table 96.

Table 96 ATD module Interrupt Vectors

Interrupt Source	Vector Address	CCR Mask	Local Enable
Sequence Complete Interrupt	$FFD2,$FFD3	I bit	ASCIE in ATDCTL2

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

Fast Background Debug Module (FBDM)

Contents

Overview 517
Features 517
Modes of Operation 519
Block Diagram 520
External Pin Descriptions 521
Register Map 523
Register Descriptions 524
Functional Description 528
Low-Power Options 546
Interrupts 546

Overview

The Fast Background Debug Mode (FBDM) module is a selectable single-wire or multi-wire, background debug system implemented in on-chip hardware for minimal CPU intervention.

This module is a super set of the BDM module and has added capability for selection of an SPI type interface in addition to the single-wire interface. This allows the interface to operate at a higher speed, through the use of three pins.

Features

- Single-wire communication with host development system
- Active out of reset in special single-chip mode

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

Freescale Semiconductor, Inc.

- Nine hardware commands using free cycles, if available, for minimal CPU intervention
- Hardware commands not requiring active BDM
- 15 firmware commands execute from on the standard BDM firmware lookup table
- Instruction tagging capability
- Software control of BDM operation during wait mode
- Software selectable clocks
- BDM disabled when secure feature is enabled.
- Selectable SPI type interface (matches Motorola standard SPI phase =1, polarity =1)
- In SPI mode, word reads are accelerated by sending next address while receiving data
- In SPI mode, the interface will always force a steal in order to make BDM instructions faster

Freescale Semiconductor, Inc.

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

Modes of Operation

BDM is available in all operating modes but must be enabled before firmware commands are executed.

Some on-chip peripherals have a control bit which allows suspending the peripheral function during background debug mode.

In special single-chip mode, background operation is enabled and active out of reset. This allows programming a system with blank memory.

BDM is also active out of special peripheral mode reset and can be turned off by clearing the BDMACT bit in the BDM status (BDMSTS) register. This allows testing of the BDM memory space as well as the user's memory space.

NOTE: *The BDM serial system should not be used in special peripheral mode since the CPU, which in other modes interfaces with the BDM to relinquish control of the bus during a free cycle or a steal operation, is not operating in this mode.*

Normal Operation BDM operates the same in all normal modes.

Special Operation

Special single-chip mode BDM is enabled and active immediately out of reset. This allows programming a system with blank memory.

Special peripheral mode BDM is enabled and active immediately out of reset. BDM can be disabled by clearing the BDMACT bit in the BDM status (BDMSTS) register. The BDM serial system should not be used in special peripheral mode.

Emulation Modes In emulation modes, the BDM operates as in all normal modes.

Freescale Semiconductor, Inc.

Block Diagram

The block diagrams of the BDM are shown in Figure 99 and Figure 100 below.

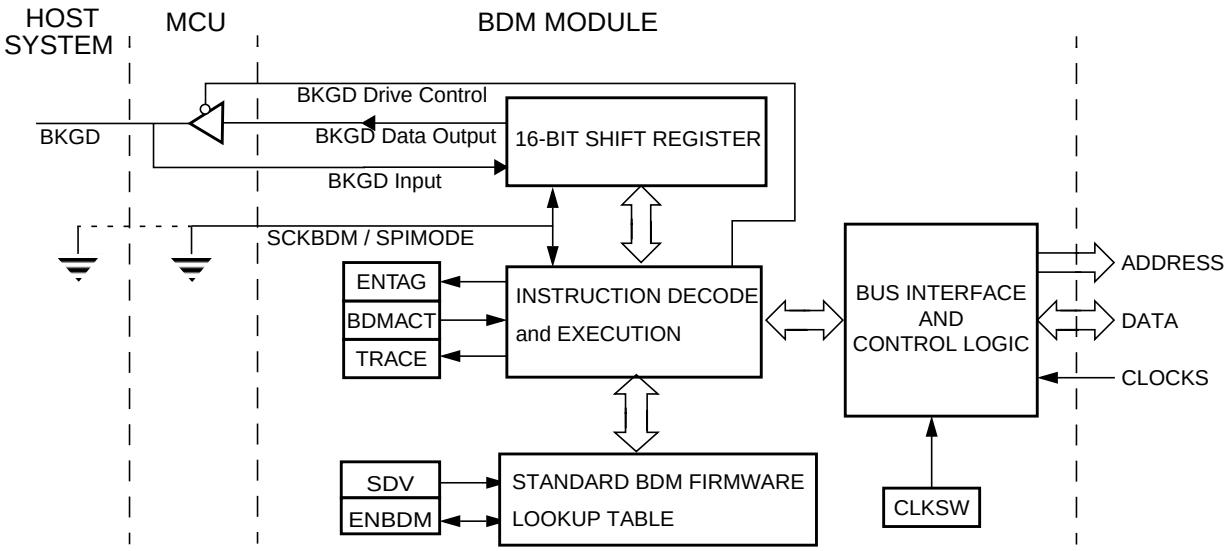


Figure 99 Block Diagram of BDM in Single Wire Mode

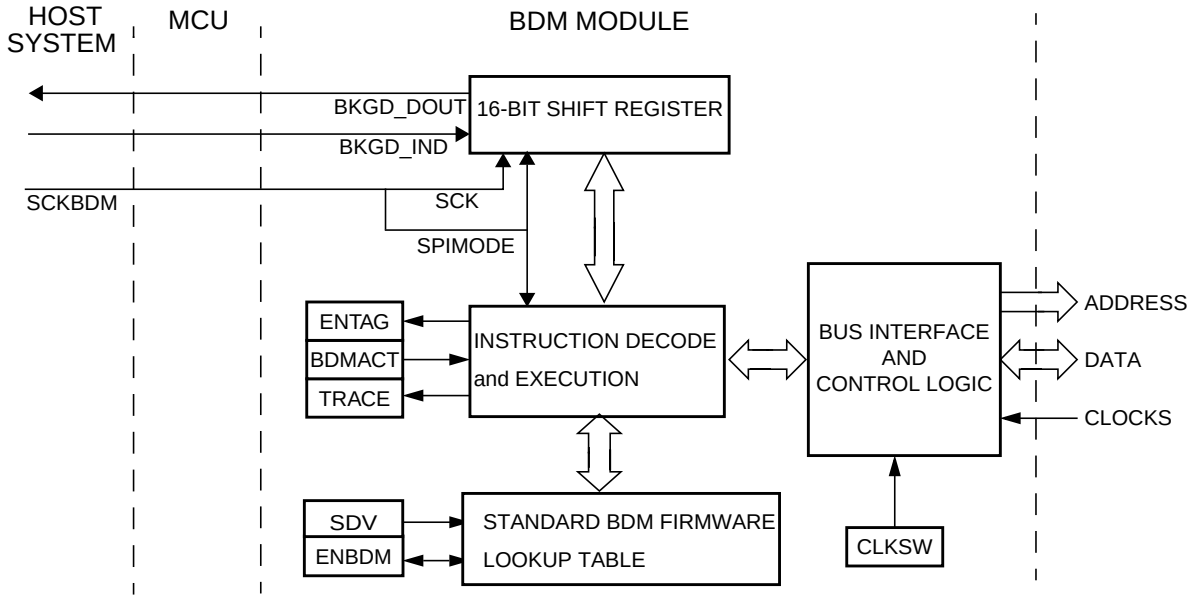


Figure 100 Block Diagram of BDM in SPI Mode

Freescale Semiconductor, Inc.

External Pin Descriptions

From a core standpoint there are five pins making up the FBDM interface. At the chip level these pins will likely be combined with other pins.

- BKGD / SI— Background interface pin, which becomes the SI (Serial data into the BDM) in SPI mode
- $\overline{\text{TAGHI}}$ — High byte instruction tagging pin
- $\overline{\text{TAGLO}}$ — Low byte instruction tagging pin
- SCKBDM / SPIMODE — Selects SPI interface or single wire interface at rising edge the $\overline{\text{RESET}}$ pin, if SPI mode is select, then this pin becomes the serial clock input
- SO — Serial data out of the BDM for SPI mode

Background Interface Pin (BKGD)

Debugging control logic communicates with external devices serially via the single-wire background interface pin (BKGD). During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the background debug mode. The serial data from the host system to the FBDM uses this pin in SPI mode.

High Byte Instruction Tagging Pin ($\overline{\text{TAGHI}}$)

This pin is used to tag the high byte of an instruction. When instruction tagging is on, a logic 0 at the falling edge of the external clock (ECLK) tags the high half of the instruction word being read into the instruction queue.

Low Byte Instruction Tagging Pin ($\overline{\text{TAGLO}}$)

This pin is used to tag the low byte of an instruction. When instruction tagging is on and low strobe is enabled, a logic 0 at the falling edge of the external clock (ECLK) tags the low half of the instruction word being read into the instruction queue.

Freescale Semiconductor, Inc.

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

FBDM Clock in SPI Mode (SCKBDM) and SPI Mode Select (SPIMODE)

While the part is in reset, this pin selects whether the FBDM will operate as a single-wire interface (SPIMODE = 0) or multi-wire SPI type interface (SPIMODE = 1). When reset is released this mode is locked in and cannot be changed until the next reset.

When the part is out of reset, if SPI mode has been selected, then this pin is the serial clock input (SCKBDM pin, SCK inside the module). The clock indicates when data should be shifted and sampled.

Serial Data Out of FBDM In SPI Mode (SO)

The serial data from the FBDM to the host system uses this output pin in SPI mode. If single wire mode is selected during reset, this pin will hold a driven low state.

Freescale Semiconductor, Inc.

Register Map

A summary of the registers associated with the BDM is shown in Figure 101 below. Registers are accessed by host-driven communications to the BDM hardware using READ_BD and WRITE_BD commands. Detailed descriptions of the registers and associated bits are given in the subsections that follow.

Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FF00	BDM reserved	Read:	X	X	X	X	X	X	0	0
		Write:								
$FF01	BDMSTS	Read:	ENBDM	BDMACT	ENTAG	SDV	TRACE	CLKSW	UNSEC	0
		Write:								
$FF02	BDM reserved	Read:	X	X	X	X	X	X	X	X
		Write:								
$FF03	BDM reserved	Read:	X	X	X	X	X	X	X	X
		Write:								
$FF04	BDM reserved	Read:	X	X	X	X	X	X	X	X
		Write:								
$FF05	BDM reserved	Read:	X	X	X	X	X	X	X	X
		Write:								
$FF06	BDMCCR	Read:	CCR7	CCR6	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0
		Write:								
$FF07	BDMINR	Read:	0	REG14	REG13	REG12	REG11	0	0	0
		Write:								

= Unimplemented X = Indeterminate

Figure 101 BDM Register Map Summary

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

Register Descriptions

BDM Status Register (BDMSTS)

Address: $FF01

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ENBDM	BDMACT	ENTAG	SDV	TRACE	CLKSW	UNSEC	0
Write:								
Reset:								
Special single-chip mode:	0	1	0	0	0	0	0	0
Special peripheral mode:	0	1	0	0	0	0	0	0
All other modes:	0	0	0	0	0	0	0	0

= Unimplemented

Read: All modes through BDM operation

Write: All modes but subject to the following:

- BDMACT can only be set by BDM hardware upon entry into BDM. It can only be cleared by the standard BDM firmware lookup table upon exit from BDM active mode.
- CLKSW can only be written via BDM hardware or standard BDM firmware write commands.
- All other bits, while writable via BDM hardware or standard BDM firmware write commands, should only be altered by the BDM hardware or standard firmware lookup table as part of BDM command execution.
- ENBDM should only be set via a BDM hardware command if the BDM firmware commands are needed. (This does not apply in Special Single Chip Mode).

ENBDM — Enable BDM

This bit controls whether the BDM is enabled or disabled. When enabled, BDM can be made active to allow firmware commands to be executed. When disabled, BDM cannot be made active but BDM hardware commands are still allowed.

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

Freescale Semiconductor, Inc.

1 = BDM enabled
0 = BDM disabled

NOTE: *ENBDM is set by the firmware immediately out of reset in special single-chip mode. In secured mode this bit will not be set by the firmware until after the FLASH erase verify test is complete.*

BDMACT — BDM active status

This bit becomes set upon entering BDM. The standard BDM firmware lookup table is then enabled and put into the memory map. BDMACT is cleared by a carefully timed store instruction in the standard BDM firmware as part of the exit sequence to return to user code and remove the BDM memory from the map.

1 = BDM active
0 = BDM not active

ENTAG — Tagging enable

This bit indicates whether instruction tagging in enabled or disabled. It is set when the TAGGO command is executed and cleared when BDM is entered. The serial system is disabled and the tag function enabled 16 cycles after this bit is written. BDM cannot process serial commands while tagging is active.

1 = Tagging enabled
0 = Tagging not enabled, or BDM active

SDV — Shift data valid

This bit is set and cleared by the BDM hardware. It is set after data has been transmitted as part of a firmware read command or after data has been received as part of a firmware write command. It is cleared when the next BDM command has been received or background debug mode is exited. SDV is used by the standard BDM firmware to control program flow execution.

1 = Data phase of command is complete
0 = Data phase of command not complete

Freescale Semiconductor, Inc.

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

TRACE — TRACE1 BDM firmware command is being executed

This bit gets set when a BDM TRACE1 firmware command is first recognized. It will stay set as long as continuous back-to-back TRACE1 commands are executed. This bit will get cleared when the next command that is not a TRACE1 command is recognized.

1 = TRACE1 command is being executed

0 = TRACE1 command is not being executed

CLKSW — Clock switch

The CLKSW bit controls which clock the BDM operates with. It is only writable from a hardware BDM command. A WRITE_BD_BYTE command to address $FF01 changes the CLKSW bit. The 150 cycle delay at the clock speed that is active during the data portion of the command will occur before the new clock source is guaranteed to be active. The start of the next BDM command uses the new clock for timing subsequent BDM communications. This clock is referred as target clock in this document.

1 = BDM system operates with bus clock rate

0 = BDM system operates with oscillator clock divided by 2 when the PLLSEL bit is set, otherwise BDM system operates with bus clock rate.

Table 97 Target Clock Selection Summary

CLKSW	PLLSEL	Target Clock
0	0	Bus Clock
0	1	Oscillator Clock / 2
1	—	Bus Clock

UNSEC — Unsecure

This bit is writable only in special single chip mode from the BDM secure firmware and always resets to 0. This bit is clear as secured mode is entered so that the secure BDM firmware lookup table is enabled and put into the memory map along with the standard BDM firmware lookup table.

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

Freescale Semiconductor, Inc.

The secure BDM firmware lookup table verifies that the FLASH is erased. This being the case, UNSEC signal is set. The BDM program jumps to start of the standard BDM firmware lookup table and the secure BDM firmware lookup table is turned off. If the erase test fails, the UNSEC bit will not be asserted.

1 = the part is in the unsecured mode
0 = the part is in the secured mode

WARNING: *When UNSEC is set, security is off and the user can change the state of the secure bits in the on-chip Flash EEPROM. Note that if the user does not change the state of the bits to "unsecured" mode, the system will be secured again when it is next taken out of reset.*

BDM CCR Holding Register (BDMCCR)

Address: $FF06

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	CCR7	CCR6	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0
Reset:	0	0	0	0	0	0	0	0

Read: All modes

Write: All modes

NOTE: *When BDM is made active, the CPU stores the value of the CCR register in the BDMCCR register. However, out of special single-chip reset, the BDMCCR is set to $D8 and not $D0 which is the reset value of the CCR register.*

CCR7–CCR0 — BDM CCR Holding Bits

When entering background debug mode, the BDM CCR holding register is used to save the contents of the condition code register of the user's program. It is also used for temporary storage in the standard BDM firmware mode. The BDM CCR holding register can be written to modify the CCR value.

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

Freescale Semiconductor, Inc.

BDM Internal Register Position Register (BDMINR)

Address: $FF07

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	REG14	REG13	REG12	REG11	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Read: All modes

Write: Never

REG14–REG11 — Internal register map position

These four bits show the state of the upper bits of the base address for the system's relocatable register block. BDMINR is a shadow of the INITRG register which maps the register block to any 2K byte space within the first 32K bytes of the 64K byte address space. If the register block size is 1K bytes, it will always occupy the first 1K bytes of the specified 2K byte space.

Functional Description

The BDM module receives and executes commands from a host via a single wire serial interface or via the BDM SPI interface. There are two types of BDM commands, namely, hardware commands and firmware commands.

Hardware commands are used to read and write target system memory locations and to enter active background debug mode (see BDM Hardware Commands). Target system memory includes all memory that is accessible by the CPU.

Firmware commands are used to read and write CPU resources and to exit from active background debug mode (see Standard BDM Firmware Commands). The CPU resources referred to are the accumulator (D), X

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

Freescale Semiconductor, Inc.

index register (X), Y index register (Y), stack pointer (SP), and program counter (PC).

Hardware commands can be executed at any time and in any mode excluding a few exceptions as highlighted in Modes of Operation below. Firmware commands can only be executed when the system is in active background debug mode (BDM).

SPI Mode

An SPI mode option has been added to this module. When SPI mode is selected, the single wire interface becomes three wires: serial slave data in (SI), serial slave data out (SO) and serial slave clock in (SCK). As in the single wire mode, the external system still initiates and controls all transfers. The dedicated pins allow faster transfer rates, up to one quarter the bus frequency per bit.

While in SPI mode, the delay time between an address and read data is shortened to always steal the next available cycle. On parts operating in single chip mode, this will be 8 target clock cycles (detailed explanation in Hardware Delay in SPI Mode). It will take 21 target clock cycles for parts with memory and / or peripherals on an external bus (to allow for misaligned, narrow bus and stretched accesses on the cycle before the steal and to allow for narrow bus and stretched accesses on the steal cycle).

Also while in SPI mode, there is one hardware command with added functionality. The read word instruction will take in a new address while the data is being output. Then another address can be sent in while the data from the previous address is sent out (full duplex operation). When the user has completed the operation, an $FFFF needs to be sent in as the address. This will quit the read word instruction and the next input will be a new command. Note that this is the only instruction with this operation enhancement.

Selecting SPI Mode

The core has an input (SCKBDM / SPIMODE) which selects whether the BDM will use a single wire interface or the SPI type interface. This selection occurs at the release of reset and cannot be changed until the next reset. SPIMODE = 1 during reset for selecting the SPI type interface and SPIMODE = 0 during reset for selecting the single wire interface.

Freescale Semiconductor, Inc.

Secured Mode

If the user resets into special single chip mode with the part secured, a secure BDM firmware lookup table is brought into the map along with the standard BDM firmware lookup table. The secure BDM firmware is higher priority than the standard BDM firmware. The secure BDM firmware verifies that the FLASH is erased. This being the case, the UNSEC bit is asserted (written to one). The BDM program jumps to start of the standard BDM firmware and the secure BDM firmware is turned off. If the FLASH do not verify as erased, the BDM firmware sets the ENBDM bit, without asserting UNSEC, and the firmware enters a loop. This causes the BDM hardware commands to become enabled, but does not enable the software commands. This allows the BDM hardware commands to be used to erase the FLASH.

Enabling and Activating BDM

The system must be in active BDM to execute standard BDM firmware commands. BDM can be activated only after being enabled. BDM is enabled by setting the ENBDM bit in the BDM status (BDMSTS) register. The ENBDM bit is set by writing to the BDM status (BDMSTS) register, via the single-wire interface or via the BDM SPI interface, using a hardware command such as WRITE_BD_BYTE.

After being enabled, BDM is activated by one of the following[1]:

- Hardware BACKGROUND command
- BDM external instruction tagging mechanism
- CPU BGND instruction
- Breakpoint sub-block's force or tag mechanism[2]

When BDM is activated, the CPU finishes executing the current instruction and then begins executing the firmware in the standard BDM firmware lookup table. When BDM is activated by the breakpoint sub-block, the type of breakpoint used determines if BDM becomes active before or after execution of the next instruction.

---

1. BDM is enabled and active immediately out of special single-chip reset (see Special Operation).

2. This method is only available on systems that have a a Breakpoint sub-block.

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

Freescale Semiconductor, Inc.

NOTE: *If an attempt is made to activate BDM before being enabled, the CPU resumes normal instruction execution after a brief delay. If BDM is not enabled, any hardware BACKGROUND commands issued are ignored by the BDM and the CPU is not delayed.*

In active BDM, the BDM registers and standard BDM firmware lookup table are mapped to addresses $FF00 to $FFFF. BDM registers are mapped to addresses $FF00 to $FF07. The BDM module uses these registers which are readable anytime by the BDM module, not user programs.

NOTE: *When the background debug mode is activated, the PWM, ECT and ATD modules enter freeze mode and control bits determine if the module is stopped during active BDM. See each module section about freeze mode.*

BDM Hardware Commands

Hardware commands are used to read and write target system memory locations and to enter active background debug mode. Target system memory includes all memory that is accessible by the CPU such as on-chip RAM, EEPROM, Flash EEPROM, I/O and control registers, and all external memory.

Hardware commands are executed with minimal or no CPU intervention and do not require the MCU to be in active BDM for execution, although, they can still be executed in this mode. When executing a hardware command in single wire mode, the BDM sub-block waits for a free CPU bus cycle so that the background access does not disturb the running application programs. If a free cycle is not found within 128 clock cycles, the CPU is momentarily frozen so that the BDM module can steal a cycle. When the BDM module finds a free cycle, the operation does not intrude on normal CPU operation provided that it can be completed in a single cycle. However, if an operation requires multiple cycles, CPU clocks are frozen until the operation is complete, even though the BDM module found a free cycle.

Freescale Semiconductor, Inc.

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

The BDM hardware commands are listed in Table 98.

Table 98 Hardware Commands

Command	Opcode (hex)	Data	Description
BACKGROUND	90	None	Enter background debug mode if firmware is enabled.
READ_BD_BYTE	E4	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table in map. Odd address data on low byte; even address data on high byte
READ_BD_WORD	EC	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table in map. Must be aligned access.
READ_BYTE	E0	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table out of map. Odd address data on low byte; even address data on high byte
READ_WORD (single wire mode)	E8	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table out of map. Must be aligned access.
READ_WORD (SPI mode)	E8	16-bit address 16-bit data out / next address	Read from memory with standard BDM firmware lookup table out of map. Must be aligned access. SPI interface only: next address must be sent in with data out, use address = $FFFF to end read word operation.
WRITE_BD_BYTE	C4	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table in map. Odd address data on low byte; even address data on high byte
WRITE_BD_WORD	CC	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table in map. Must be aligned access
WRITE_BYTE	C0	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table out of map. Odd address data on low byte; even address data on high byte
WRITE_WORD	C8	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table out of map. Must be aligned access.

The READ_BD and WRITE_BD commands are used for reading the on standard BDM firmware lookup table locations and for reading and writing to the BDM register locations. These locations are not normally in the system memory map but share addresses with the application in memory.

To distinguish between physical memory locations that share the same address, BDM memory resources are enabled just for the READ_BD and WRITE_BD access cycle. This allows the BDM system to access

Freescale Semiconductor, Inc.

BDM locations unobtrusively, even if the addresses conflict with the application memory map.

Standard BDM Firmware Commands

Firmware commands are used to access and manipulate CPU resources. The system must be in active BDM to execute standard BDM firmware commands (see Enabling and Activating BDM). Normal instruction execution is suspended while the CPU executes the firmware located in the standard BDM firmware lookup table. The hardware command BACKGROUND is the usual way to activate BDM.

As the system enters active BDM, the standard BDM firmware lookup table and BDM registers become visible in the on-chip memory map at $FF00-$FFFF, and the CPU begins executing the standard BDM firmware. The standard BDM firmware watches for serial commands and executes them as they are received. The firmware commands are shown in Table 99.

Freescale Semiconductor, Inc.

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

Table 99 Firmware Commands

Command	Opcode (hex)	Data	Description
READ_NEXT	62	16-bit data out	Increment X by 2 (X = X + 2), then read word X points to.
READ_PC	63	16-bit data out	Read program counter.
READ_D	64	16-bit data out	Read D accumulator.
READ_X	65	16-bit data out	Read X index register.
READ_Y	66	16-bit data out	Read Y index register.
READ_SP	67	16-bit data out	Read stack pointer.
WRITE_NEXT	42	16-bit data in	Increment X by 2 (X=X+2), then write word to location pointed to by X.
WRITE_PC	43	16-bit data in	Write program counter.
WRITE_D	44	16-bit data in	Write D accumulator.
WRITE_X	45	16-bit data in	Write X index register.
WRITE_Y	46	16-bit data in	Write Y index register.
WRITE_SP	47	16-bit data in	Write stack pointer.
GO	08	none	Go to user program.
TRACE1	10	none	Execute one user instruction then return to active BDM.
TAGGO	18	none	Enable tagging and go to user program.

Freescale Semiconductor, Inc.

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

BDM Command Structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.

NOTE: *8-bit reads return 16-bits of data, of which, only one byte will contain valid data. If reading an even address, the valid data will appear in the MSB. If reading an odd address, the valid data will appear in the LSB.*

NOTE: *16-bit misaligned reads and writes are not allowed. If attempted, the BDM module will ignore the least significant bit of the address and will assume an even address from the remaining bits.*

Hardware Commands in Single Wire Mode

For hardware data read commands, the external host must wait 150 target clock cycles[1] after sending the address before attempting to obtain the read data. This is to be certain that valid data is available in the BDM shift register, ready to be shifted out. For hardware write commands, the external host must wait 150 target clock cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed. The 150 target clock cycle delay in both cases includes the maximum 128 cycle delay that can be incurred as the BDM waits for a free cycle before stealing a cycle.

Hardware Commands in SPI Mode

For hardware data read commands, the external host must wait the hardware delay after sending the address before attempting to obtain the read data. This is to be certain that valid data is available in the BDM shift register, ready to be shifted out. For hardware write commands, the external host must wait the hardware delay after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed. Refer to Hardware Delay in SPI Mode in page 539.

1. Target clock cycles are cycles measured using the target system's serial clock rate. See BDM Serial Interface and BDM Status Register (BDMSTS) for information on how serial clock rate is selected.

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

Freescale Semiconductor, Inc.

Firmware commands

For firmware read commands, the external host must wait 32 target clock cycles after sending the command opcode before attempting to obtain the read data. If the access is external with a narrow bus access (+1 cycle) and / or a stretch (+1, 2 or 3 cycles), up to an additional 7 cycles could be needed if both occur (39 target clock cycles total). This allows enough time for the requested data to be made available in the BDM shift register, ready to be shifted out. For firmware write commands, the external host must wait 32 target clock cycles after sending the data to be written, before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed.

The external host should wait 64 target clock cycles after a TRACE1 or GO command before starting any new serial command. This is to allow the CPU to exit gracefully from the standard BDM firmware lookup table and resume execution of the user code. Disturbing the BDM shift register prematurely may adversely affect the exit from the standard BDM firmware lookup table.

Figure 102 represents the BDM command structure in single wire mode. The command blocks illustrate a series of eight bit times starting with a falling edge. The bar across the top of the blocks indicates that the BKGD line idles in the high state. The minimum time for an 8-bit command is 8×16 target clock cycles.

Freescale Semiconductor, Inc.

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

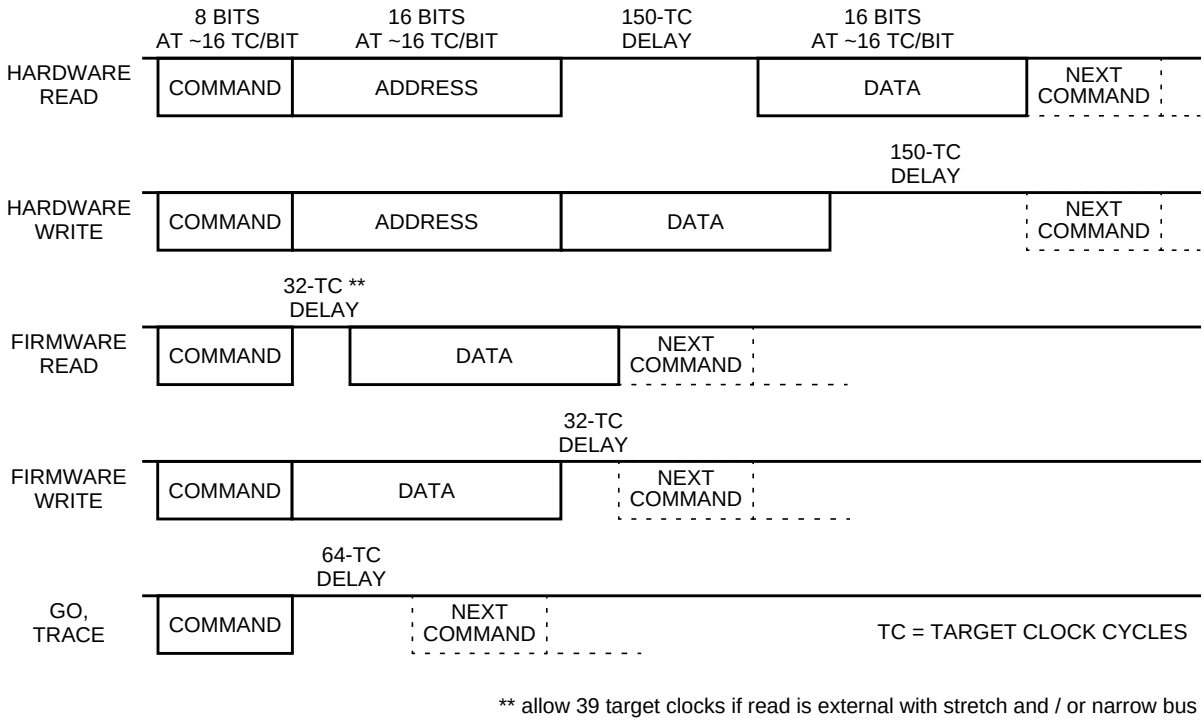


Figure 102 BDM Command Structure - Single Wire Mode

Figure 103 represents the BDM command structure in SPI mode. The command blocks illustrate a series of eight bit times starting with a falling edge. The bar across the top of the blocks indicates that the SI and SO lines idle in the high state. The minimum time for an 8-bit command is 8 × 4 target clock cycles.

Freescale Semiconductor, Inc.

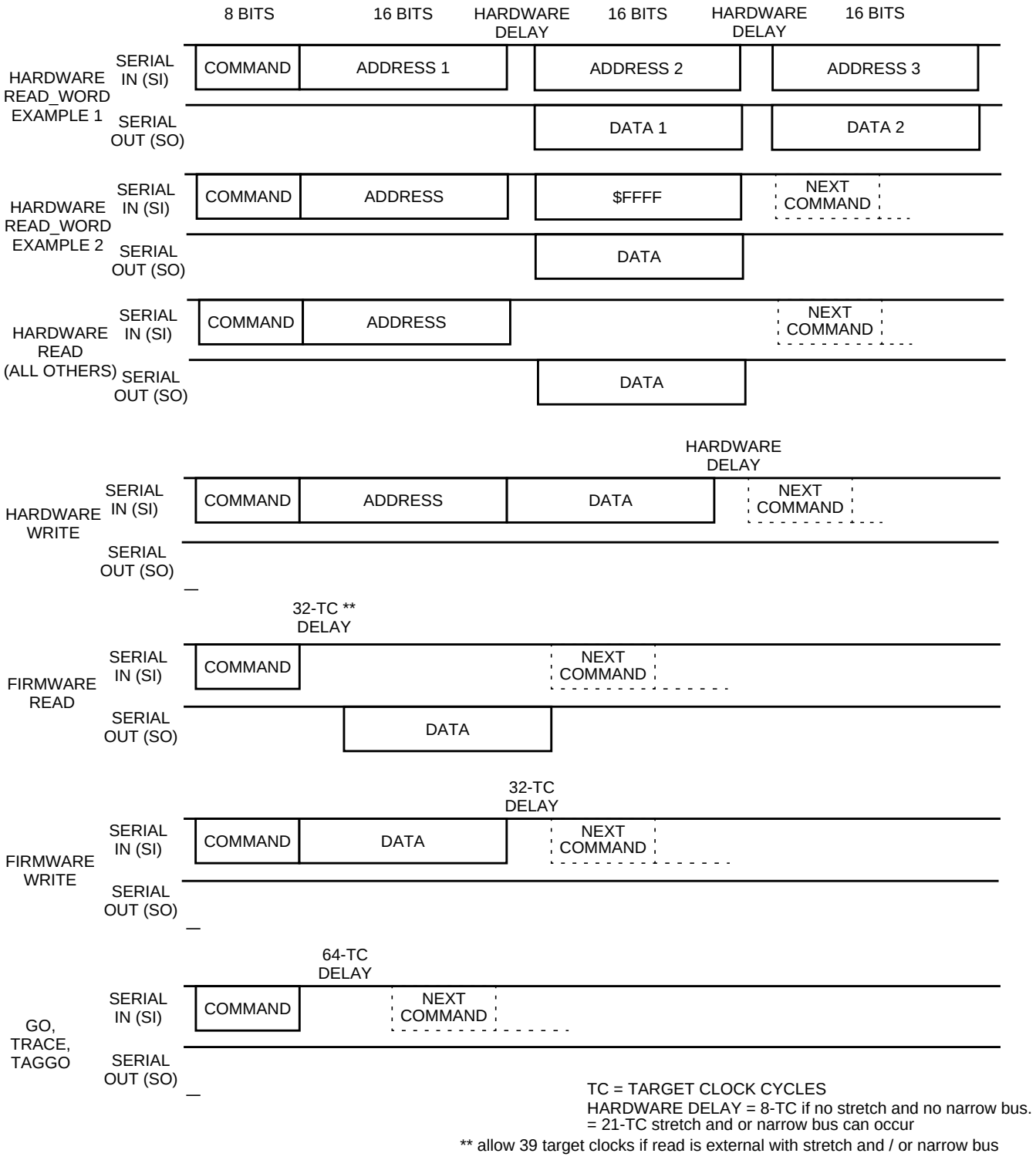


Figure 103 BDM Command Structure - SPI Mode

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

Freescale Semiconductor, Inc.

Hardware Delay in SPI Mode

A hardware delay is required between serial transfers when the BDM needs to access the bus to complete an operation. For example, on a hardware read a hardware delay is required after the address is sent so that the data can be accessed before it is returned. On a hardware write, a hardware delay is required after the write data is sent so that the write occurs before the next command is sent.

In SPI mode, the hardware delay allows the BDM to synchronize to the incoming serial data, load up the appropriate registers for the bus access, synchronize with the CPU to steal the bus, complete the bus access and for a read, put the read data into the shifter to return it.

The hardware delay takes into account the part of the operation that occurs between the rising edges of the serial clock. At the fastest transfer rate this is a total of eight target clocks. At the fastest rate, the user must wait eight target clocks which is the hardware delay. If SCKBDM was running slower, at 12 target clocks per bit, the hardware delay reduces to zero for single chip mode and to 13 for expanded mode (worst case).

This can be determined with the equations:

- For internal-only accesses
  Hardware Delay = 12TC – (SCKBDM PERIOD)
- For external accesses allowing for stretch and wide bus
  Hardware Delay = 21TC – (SCKBDM PERIOD)
- For external accesses allowing for stretch and narrow bus
  Hardware Delay = 25TC – (SCKBDM PERIOD)

Where: If the hardware delay goes negative, it is zero.

For example, if the SPI is running at 8 target clocks per bit, we get a 4 TC delay for internal accesses, a 13 TC delay for external accesses with a wide bus and stretch, and a 17 TC delay for external accesses with a narrow bus and stretch.

BDM Serial Interface

The BDM module communicates with external devices serially via the MCU mode control BKGD pin. During reset, this pin is a mode select input which selects between normal and special modes of operation.

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

Freescale Semiconductor, Inc.

After reset, this pin becomes the dedicated serial interface pin for the BDM module.

The BDM serial interface is timed using the clock selected by the CLKSW bit in the status register (see BDM Status Register (BDMSTS)). This clock will be referred to as the target clock in the following explanation.

The BDM serial interface uses a clocking scheme in which the external host generates a falling edge on the BKGD pin to indicate the start of each bit time. This falling edge is sent for every bit whether data is transmitted or received. Data is transferred most significant bit (MSB) first at 16 target clock cycles per bit. The interface times out if 512 clock cycles occur between falling edges from the host.

The BKGD pin is a pseudo open-drain pin and has an weak on-chip active pull-up that is enabled at all times. It is assumed that there is an external pullup and that drivers connected to BKGD do not typically drive the high level. Since R-C rise time could be unacceptably long, the target system and host provide brief driven-high (speedup) pulses to drive BKGD to a logic 1. The source of this speedup pulse is the host for transmit cases and the target for receive cases.

Diagrams in Figure 104, Figure 105, and Figure 106 show timing of bit-time cases in single wire mode. Each case begins when the host drives the BKGD pin low to generate a falling edge. Since the host and target are operating from separate clocks, it can take the target MCU up to one full clock cycle to recognize this edge. The target measures delays from this perceived start of the bit time while the host measures delays from the point it actually drove BKGD low to start the bit up to one target clock cycle earlier. Synchronization between the host and target is established in this manner at the start of every bit time.

Figure 104 shows an external host transmitting a logic 1 and transmitting a logic 0 to the BKGD pin of a target system. The host is asynchronous to the target, so there is up to a one clock-cycle delay from the host-generated falling edge to where the target recognizes this edge as the beginning of the bit time. Ten target clock cycles later, the target senses the bit level on the BKGD pin. Internal glitch detect logic requires

Freescale Semiconductor, Inc.

the pin be driven high no later that eight target clock cycles after the falling edge for a logic 1 transmission.

Since the host drives the high speedup pulses in these two cases, the rising edges look like digitally driven signals.

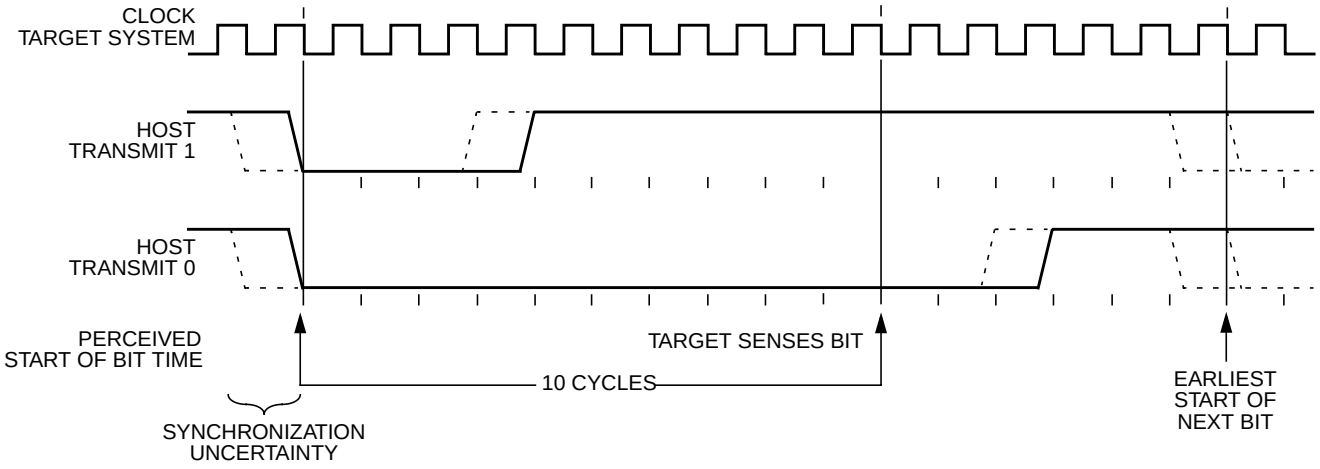


Figure 104 BDM Host-to-Target Serial Bit Timing

The receive cases are more complicated. Figure 105 shows the host receiving a logic 1 from the target MCU. Since the host is asynchronous to the target MCU, there is up to one clock-cycle delay from the host-generated falling edge on BKGD to the perceived start of the bit time in the target MCU. The host holds the BKGD pin low long enough for the target to recognize it (at least two target clock cycles). The host must release the low drive before the target MCU drives a brief high speedup pulse seven target clock cycles after the perceived start of the bit time. The host should sample the bit level about 10 target clock cycles after it started the bit time.

Freescale Semiconductor, Inc.

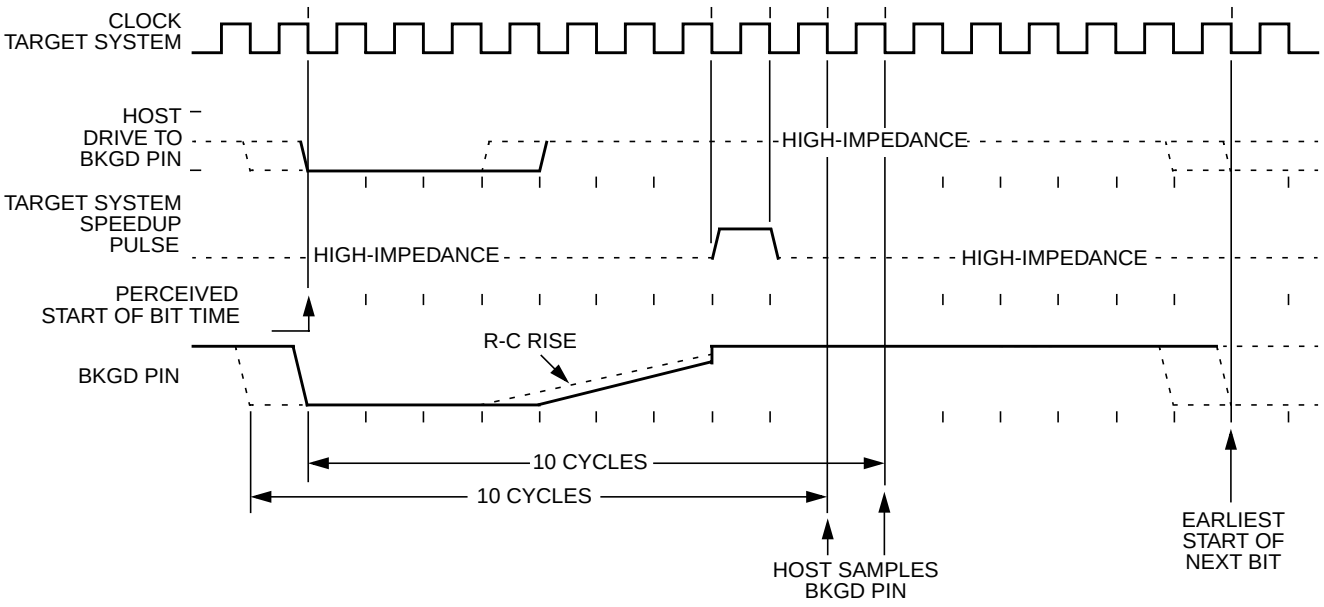


Figure 105 BDM Target-to-Host Serial Bit Timing (Logic 1)

Figure 106 shows the host receiving a logic 0 from the target MCU. Since the host is asynchronous to the target MCU, there is up to a one clock-cycle delay from the host-generated falling edge on BKGD to the start of the bit time as perceived by the target MCU. The host initiates the bit time but the target MCU finishes it. Since the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 target clock cycles then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 target clock cycles after starting the bit time.

Freescale Semiconductor, Inc.

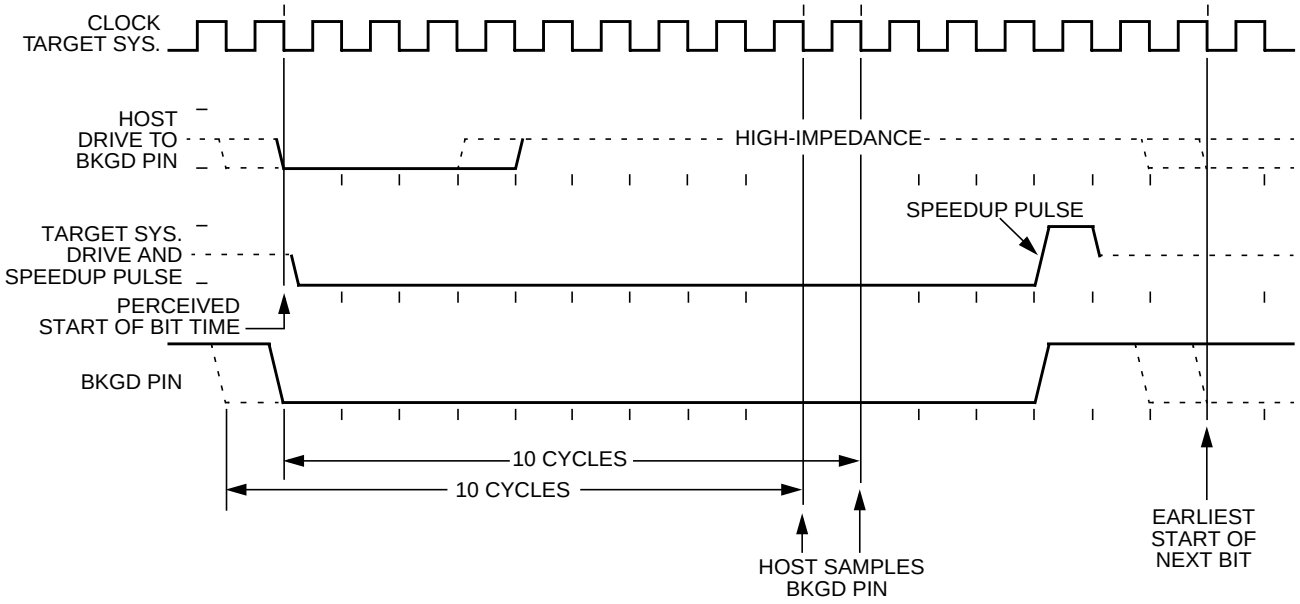


Figure 106 BDM Target-to-Host Serial Bit Timing (Logic 0)

BDM Serial Interface in SPI mode

An eight bit transfer in SPI mode is shown in the diagram (see Figure 107). The data is shifted (changes) on the falling edges of SCK and it is sampled (registered) on the rising edges of SCK. The SCK clock is always driven into the BDM and initiates a transfer. It's idle state is one. Commands, addresses and write data are driven into the BDM on SI and read data is driven out of the BDM on SO.

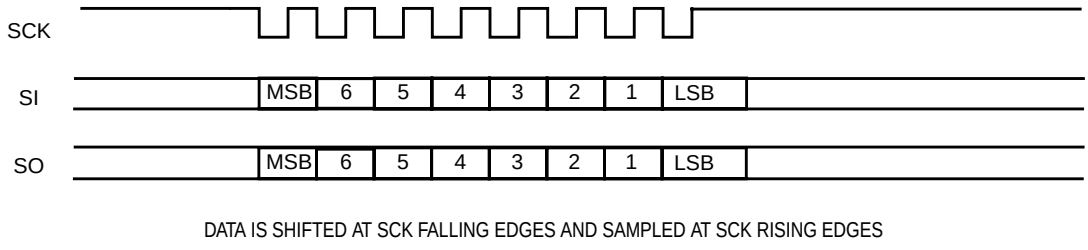


Figure 107 8-Bit Data Transfer in SPI Mode

Timing specifications for the SPI mode interface are described in the table (see Table 100). SCK must be no faster than 4 target clocks. When

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

running at the fastest rate, it is important that SCK be a 50% duty cycle because both edges of SCK are used (sampling and shifting).

Table 100 SPI Mode Timing

Characteristic	Minimum	Maximum
SCK period	4 target clocks	500 target clocks
SCK high time	2 target clocks	250 target clocks
SCK low time	2 target clocks	250 target clocks
SI data valid before rising edge of SCK	0.5 target clocks	-
SI data hold after rising edge of SCK	2 target clocks	-
SO data valid before rising edge of SCK	0.5 target clocks	-
SO data hold after rising edge of SCK	2 target clocks	-

Instruction Tracing

When a TRACE1 command is issued to the BDM module in active BDM, the CPU exits the standard BDM firmware and executes a single instruction in the user code. Once this has occurred, the CPU is forced to return to the standard BDM firmware and the BDM module is active and ready to receive a new command. If the TRACE1 command is issued again, the next user instruction will be executed. This facilitates stepping or tracing through the user code one instruction at a time.

If an interrupt is pending when a TRACE1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. Once back in standard BDM firmware execution, the program counter points to the first instruction in the interrupt service routine.

Instruction Tagging

The instruction queue and cycle-by-cycle CPU activity are reconstructible in real time or from trace history that is captured by a logic analyzer. However, the reconstructed queue cannot be used to stop the CPU at a specific instruction, because execution already has begun by the time an operation is visible outside the system. A separate instruction tagging mechanism is provided for this purpose.

Freescale Semiconductor, Inc.

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

The tag follows program information as it advances through the instruction queue. When a tagged instruction reaches the head of the queue, the CPU enters active BDM rather than executing the instruction.

NOTE: *Tagging is disabled when BDM becomes active and BDM serial commands are not processed while tagging is active.*

Executing the BDM TAGGO command configures two MCU pins for tagging. On MCUs with an [Muxed] External Bus Interface (MEBI or EBI), the $\overline{\text{TAGLO}}$ signal shares a pin with the $\overline{\text{LSTRB}}$ signal, and the $\overline{\text{TAGHI}}$ signal shares a pin with the BKGD signal. On MCUs with a Development Tools Interface (DTI), $\overline{\text{TAGLO}}$ will generally be bonded out separately.

WARNING: ***If tagging is used in SPI mode, the BKGD pin needs to be brought to a high level soon after the taggo command is sent. Since the BKGD pin is used for $\overline{\text{TAGHI}}$, it must be high within 15 target cycles, or the next instruction may get tagged.***

Table 101 shows the functions of the two tagging pins. The pins operate independently, that is, the state of one pin does not affect the function of the other. The presence of logic level 0 on either pin at the fall of the external clock (ECLK) performs the indicated function. High tagging is allowed in all modes. On MCUs with a [Muxed] External Bus Interface (MEBI or EBI), low tagging is allowed only when low strobe is enabled (LSTRB is allowed only in wide expanded modes and emulation expanded narrow mode). On MCUs with a Development Tools Interface (DTI), low tagging is allowed in all modes.

Table 101 Tag Pin Function

TAGHI	TAGLO	Tag
1	1	No tag
1	0	Low byte
0	1	High byte
0	0	Both bytes

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

Low-Power Options

Run Mode The BDM does not include disable controls that would conserve power during run mode.

Wait Mode The BDM cannot be used in wait mode if the system disables the clocks to the BDM.

Stop Mode The BDM is completely shutdown in stop mode.

Interrupts

The BDM does not generate interrupt requests.

Freescale Semiconductor, Inc.

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

Breakpoint (BKP)

Contents

Overview 547
Features 548
Block Diagram 550
External Pin Descriptions 552
Register Map 552
Register Descriptions 553
Breakpoint Priority 560
Reset Initialization 560
Interrupts 560

Overview

The Breakpoint sub-block of the Core provides for hardware breakpoints that are used to debug software on the CPU by comparing actual address and data values to predetermined data in setup registers. A successful comparison will place the CPU in Background Debug Mode or initiate a software interrupt (SWI).

The Breakpoint Module contains two modes of operation:

1. Dual Address Mode, where a match on either of two addresses will cause the system to enter Background Debug Mode or initiate a Software Interrupt (SWI).
2. Full Breakpoint Mode, where a match on address and data will cause the system to enter Background Debug Mode or initiate a Software Interrupt (SWI).

There are two types of breakpoints, forced and tagged. Forced breakpoints occur at the next instruction boundary if a match occurs and

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

Freescale Semiconductor, Inc.

tagged breakpoints allow for breaking just before a specific instruction executes. Tagged breakpoints will only occur on addresses. Tagging on data is not allowed; however, if this occurs, nothing will happen within the BKP.

The BKP allows breaking within a 256 byte address range and/or within expanded memory. It allows matching of the data as well as address matching and to match 8 bit or 16 bit data. Forced breakpoints can match on a read or a write cycle.

Freescale Semiconductor, Inc.

Features

- Full or Dual Breakpoint Mode
  - Compare on address and data (Full)
  - Compare on either of two addresses (Dual)
- BDM or SWI Breakpoint
  - Enter BDM on breakpoint (BDM)
  - Execute SWI on breakpoint (SWI)
- Tagged or Forced Breakpoint
  - Break just before a specific instruction will begin execution (TAG)
  - Break on the first instruction boundary after a match occurs (Force)
- Single, Range, or Page address compares
  - Compare on address (Single)
  - Compare on address 256 byte (Range)
  - Compare on any 16K Page (Page)
- Compare address on read or write on forced breakpoints
- High and/or low byte data compares

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

Modes of Operation

The Breakpoint can operate in Dual Address Mode or Full Breakpoint Mode. Each of these modes is discussed in the subsections below.

Dual Address Mode

When Dual Address Mode is enabled, two address breakpoints can be set. Each breakpoint can cause the system to enter Background Debug Mode or to initiate a software interrupt based upon the state of the BKBDM bit in the BKPCT0 Register being logic one or logic zero, respectively. BDM requests have a higher priority than the SWI requests. No data breakpoints are allowed in this mode.

The BKTAG bit in the BKPCT0 register selects whether the breakpoint mode is force or tag. The BKxMBH:L bits in the BKPCT1 register select whether or not the breakpoint is matched exactly or is a range breakpoint. They also select whether the address is matched on the high byte, low byte, both bytes, and/or memory expansion. The BKxRW and BKxRWE bits in the BKPCT1 register select whether the type of bus cycle to match is a read, write, or both when performing forced breakpoints.

Full Breakpoint Mode

Full Breakpoint Mode requires a match on address and data for a breakpoint to occur. Upon a successful match, the system will enter Background Debug Mode or initiate a software interrupt based upon the state of the BKBDM bit in the BKPCT0 Register being logic one or logic zero, respectively. The BDM requests have a higher priority than the SWI requests. R/W matches are also allowed in this mode.

The BKTAG bit in the BKPCT0 register selects whether the breakpoint mode is forced or tagged. If the BKTAG bit is set in BKPCT0, then only address is matched, data is ignored. The BK0MBH:L bits in the BKPCT1 register select whether or not the breakpoint is matched exactly, is a range breakpoint, or is in page space. The BK1MBH:L bits in the BKPCT1 register select whether the data is matched on the high byte, low byte, or both bytes. The BK0RW and BK0RWE bits in the BKPCT1 register select whether the type of bus cycle to match is a read or a write

Freescale Semiconductor, Inc.

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

when performing forced breakpoints. BK1RW and BK1RWE bits in the BKPCT1 register are not used in Full Breakpoint Mode.

Block Diagram

A block diagram of the Breakpoint sub-block is shown in Figure 108 below. The Breakpoint contains three main sub-blocks: the Register Block, the Compare Block and the Control Block. The Register Block consists of the eight registers that make up the Breakpoint register space. The Compare Block performs all required address and data signal comparisons. The Control Block generates the signals for the CPU for the tag high, tag low, force SWI and force BDM functions. In addition, it generates the register read and write signals and the comparator block enable signals.

NOTE: *There is a two cycle latency for address compares for forces, a two cycle latency for write data compares, and a three cycle latency for read data compares.*

Freescale Semiconductor, Inc.

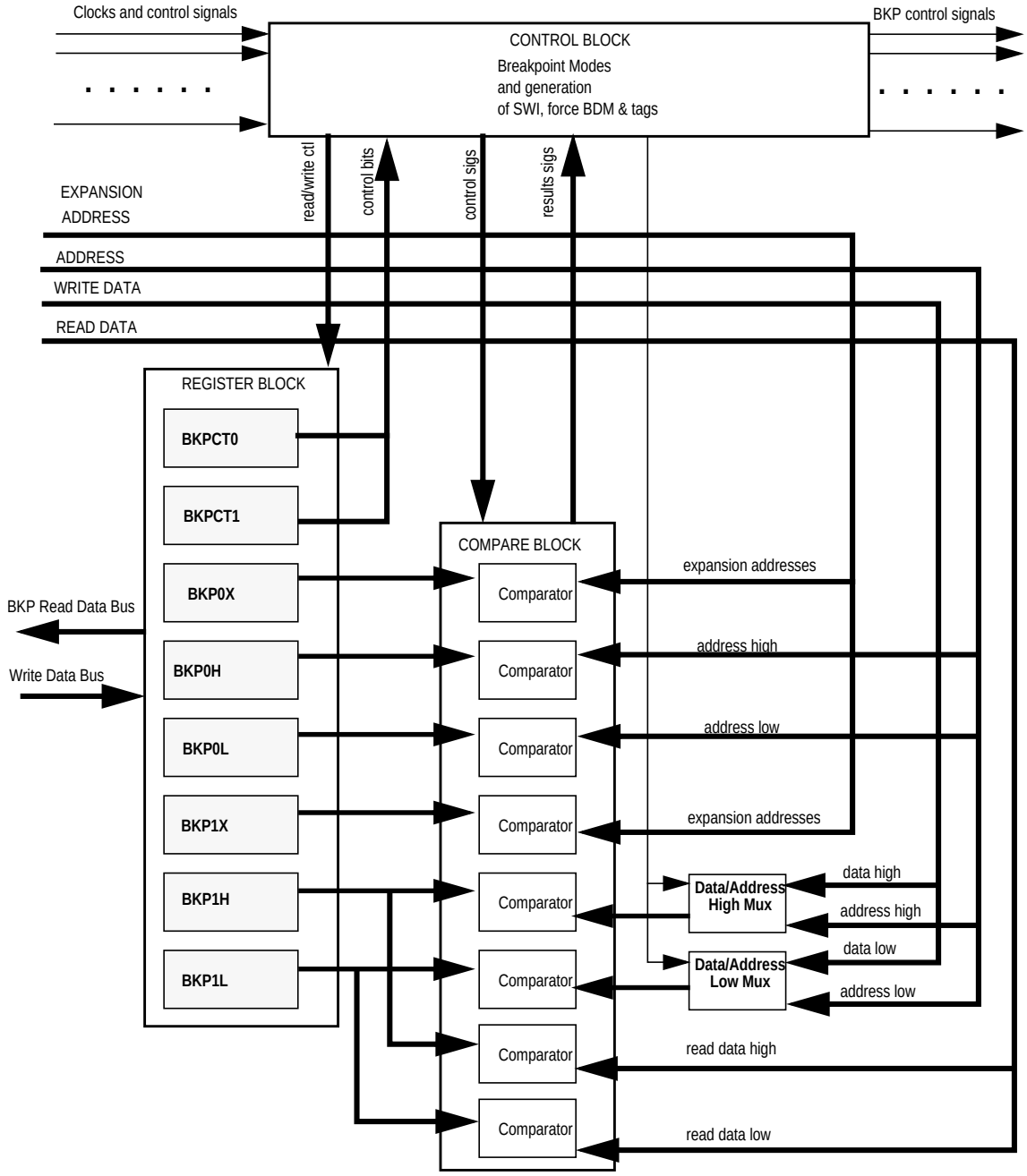


Figure 108 Breakpoint Block Diagram

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

Freescale Semiconductor, Inc.

External Pin Descriptions

The BKP sub-block does not access any external pins.

Register Map

Register Name		Bit 7	6	5	4	3	2	1	Bit 0	Address Offset
BKPCT0	Read:	BKEN	BKFULL	BKBDM	BKTAG	0	0	0	0	$0028
	Write:									
BKPCT1	Read:	BK0MBH	BK0MBL	BK1MBH	BK1MBL	BK0RWE	BK0RW	BK1RWE	BK1RW	$0029
	Write:									
BKP0X	Read:	0	0	BK0V5	BK0V4	BK0V3	BK0V2	BK0V1	BK0V0	$002A
	Write:									
BKP0H	Read:	Bit 15	14	13	12	11	10	9	Bit 8	$002B
	Write:									
BKP0L	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$002C
	Write:									
BKP1X	Read:	0	0	BK1V5	BK1V4	BK1V3	BK1V2	BK1V1	BK1V0	$002D
	Write:									
BKP1H	Read:	Bit 15	14	13	12	11	10	9	Bit 8	$002E
	Write:									
BKP1L	Read:	Bit 7	6	5	4	3	2	1	Bit 0	$002F
	Write:									

= Unimplemented or reserved

Figure 109 Breakpoint Register Map

NOTE: *Register Address = Base Address (INITRG) + Address Offset*

Freescale Semiconductor, Inc.

Register Descriptions

There are eight 8-bit registers in the BKP module.

NOTE: *All bits of all registers in this module are completely synchronous to internal clocks during a register read.*

Breakpoint Control Register 0 (BKPCT0)

This register is used to set the breakpoint modes.

Read and write: anytime.

Address Offset $0028

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BKEN	BKFULL	BKBDM	BKTAG	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

BKEN — Breakpoint Enable

This bit enables the module.

1 = Breakpoint module on.
0 = Breakpoint module off.

BKFULL— Full Breakpoint Mode Enable

This bit controls whether the breakpoint module is in Dual Address Mode or Full Breakpoint Mode

1 = Full Breakpoint Mode enabled.
0 = Dual Address Mode enabled.

BKBDM — Breakpoint Background Debug Mode Enable

This bit determines if the breakpoint causes the system to enter Background Debug Mode (BDM) or initiate a Software Interrupt (SWI)

1 = Go to BDM on a compare.
0 = Go to Software Interrupt on a compare.

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

Freescale Semiconductor, Inc.

BKTAG — Breakpoint on Tag

This bit controls whether the breakpoint will cause a break on the next instruction boundary (force) or on a match that will be an executable opcode (tagged). Non-executed opcodes cannot cause a tagged breakpoint

1 = On match, break if the match is an instruction that will be executed (tagged).

0 = On match, break at the next instruction boundary (force).

Breakpoint Control Register 1 (BKPCT1)

This register is used to control the breakpoint logic that resides within the Core.

Read and write: anytime.

Address Offset $0029

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BK0MBH	BK0MBL	BK1MBH	BK1MBL	BK0RWE	BK0RW	BK1RWE	BK1RW
Reset:	0	0	0	0	0	0	0	0

BK0MBH:BK0MBL — Breakpoint Mask High Byte and Low Byte for First Address

In Dual Address or Full Breakpoint Mode, these bits may be used to mask (disable) the comparison of the high and low bytes of the first address breakpoint. The functionality is as given in Table 102 below.

Table 102 Breakpoint Mask Bits for First Address

BK0MBH:BK0MBL	Address Compare	BKP0X	BKP0H	BKP0L
x:0	Full Address Compare	Yes(1)	Yes	Yes
0:1	256 byte Address Range	Yes(1)	Yes	No
1:1	16K byte Address Range	Yes(1)	No	No

1. If page is selected.

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

Freescale Semiconductor, Inc.

The x:0 case is for a Full Address Compare. When a program page is selected, the full address compare will be based on bits for a 20-bit compare. The registers used for the compare are {BKP0X[5:0],BKP0H[5:0],BKP0L[7:0]}. When a program page is not selected the full address compare will be based on bits for a 16-bit compare. The registers used for the compare are {BKP0H[7:0],BKP0L[7:0]}.

The 1:0 case is not sensible because it would ignore the high order address and compare the low order and expansion addresses. Logic forces this case to compare all address lines (effectively ignoring the BK0MBH control bit).

The 1:1 case is useful for triggering a breakpoint on any access to a particular expansion page. This only makes sense if a program page is being accessed so that the breakpoint trigger will only occur if BKP0X compares.

BK1MBH:BK1MBL — Breakpoint Mask High Byte and Low Byte of Data (Second Address)

In Dual Address Mode, these bits may be used to mask (disable) the comparison of the high and/or low bytes of the second address breakpoint. The functionality is as given in Table 103 below.

Table 103 Breakpoint Mask Bits for Second Address (Dual Address Mode)

BK1MBH:BK1MBL	Address Compare	BKP1X	BKP1H	BKP1L
x:0	Full Address Compare	Yes(1)	Yes	Yes
0:1	256 byte Address Range	Yes(1)	Yes	No
1:1	16K byte Address Range	Yes(1)	No	No

1. If page is selected.

The x:0 case is for a Full Address Compare. When a program page is selected the full address compare will be based on bits for a 20-bit compare. The registers used for the compare are {BKP1X[5:0],BKP1H[5:0],BKP1L[7:0]}. When a program page is not selected the full address compare will be based on bits for a 16-bit compare. The registers used for the compare are {BKP1H[7:0],BKP1L[7:0]}.

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

Freescale Semiconductor, Inc.

The 1:0 case is not sensible because it would ignore the high order address and compare the low order and expansion addresses. Logic forces this case to compare all address lines (effectively ignoring the BK1MBH control bit).

The 1:1 case is useful for triggering a breakpoint on any access to a particular expansion page. This only makes sense if a program page is being accessed so that the breakpoint trigger will only occur if BKP1X compares.

In Full Breakpoint Mode, these bits may be used to mask (disable) the comparison of the high and/or low bytes of the data breakpoint. The functionality is as given in Table 104 below.

Table 104 Breakpoint Mask Bits for Data Breakpoints (Full Breakpoint Mode)

BK1MBH:BK1MBL	Data Compare	BKP1X	BKP1H	BKP1L
0:0	High and Low Byte Compare	No(1)	Yes	Yes
0:1	High Byte	No(1)	Yes	No
1:0	Low Byte	No(1)	No	Yes
1:1	No Compare	No(1)	No	No

1. Expansion addresses for breakpoint 1 are not available in this mode.

BK0RWE — R/$\overline{W}$ Compare Enable

Enables the comparison of the R/$\overline{W}$ signal for first address breakpoint. This bit is not useful in tagged breakpoints.

1 = R/$\overline{W}$ is used in comparisons.

0 = R/$\overline{W}$ is not used in the comparisons.

BK0RW— R/$\overline{W}$ Compare Value

When BK0RWE=1, this bit determines the type of bus cycle to match on first address breakpoint. When BK0RWE=0, this bit has no effect.

1 = Read cycle will be matched

0 = Write cycle will be matched.

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

Freescale Semiconductor, Inc.

BK1RWE — R/$\overline{W}$ Compare Enable

In Dual Address Mode, this bit enables the comparison of the R/$\overline{W}$ signal to further specify what causes a match for the second address breakpoint. This bit is not useful on tagged breakpoints or in Full Breakpoint Mode and is therefore a don't care.

1 = R/$\overline{W}$ is used in comparisons.
0 = R/$\overline{W}$ is not used in comparisons.

BK1RW — R/$\overline{W}$ Compare Value

When BK1RWE=1, this bit determines the type of bus cycle to match on the second address breakpoint.When BK1RWE=0, this bit has no effect.

1 = Read cycle will be matched.
0 = Write cycle will be matched.

Freescale Semiconductor, Inc.

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

Breakpoint (BKP)

Breakpoint First Address Expansion Register (BKP0X)

This register contains the data to be matched against expansion address lines for the first address breakpoint when a page is selected.

Read and write: anytime.

Address Offset $002A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	BK0V5	BK0V4	BK0V3	BK0V2	BK0V1	BK0V0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

BK0V[5:0] — First Address Breakpoint Expansion Address Value

Breakpoint First Address High Byte Register (BKP0H)

This register is used to set the breakpoint when compared against the high byte of the address.

Read and write: anytime

Address Offset $002B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	14	13	12	11	10	9	Bit 8
Write:								
Reset:	0	0	0	0	0	0	0	0

Breakpoint First Address Low Byte Register (BKP0L)

This register is used to set the breakpoint when compared against the low byte of the address.

Read and write: anytime

Address Offset $002C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	0	0	0	0	0	0	0	0

Freescale Semiconductor, Inc.

Breakpoint Second Address Expansion Register(BKP1X)

In Dual Address Mode, this register contains the data to be matched against expansion address lines for the second address breakpoint when a page is selected. In Full Breakpoint Mode, this register is not used.

Read and write: anytime

Address Offset $002D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	BK1V5	BK1V4	BK1V3	BK1V2	BK1V1	BK1V0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Reserved or unimplemented

BK1V[5:0] — Second Address Breakpoint Expansion Address Value

Breakpoint Data (Second Address) High Byte Register (BKP1H)

In Dual Address Mode, this register is used to compare against the high order address lines. In Full Breakpoint Mode, this register is used to compare against the high order data lines.

Read and write: anytime

Address Offset $002E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	14	13	12	11	10	9	Bit 8
Write:								
Reset:	0	0	0	0	0	0	0	0

Breakpoint Data (Second Address) Low Byte Register (BKP1L)

In Dual Address Mode, this register is used to compare against the low order address lines. In Full Breakpoint Mode, this register is used to compare against the low order data lines.

Read and write: anytime

Address Offset $002F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	0	0	0	0	0	0	0	0

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

Freescale Semiconductor, Inc.

Breakpoint Priority

Breakpoint operation is first determined by the state of BDM. If BDM is already active, meaning the CPU is executing out of BDM firmware, Breakpoints are not allowed. In addition, while in BDM trace mode, tagging into BDM is not allowed. If BDM is not active, the Breakpoint will give priority to BDM requests over SWI requests. This condition applies to both forced and tagged breakpoints.

In all cases, BDM related breakpoints will have priority over those generated by the Breakpoint sub-block. This priority includes breakpoints enabled by the $\overline{\text{TAGLO}}$ and $\overline{\text{TAGHI}}$ external pins of the system that interface with the BDM directly and whose signal information passes through and is used by the Breakpoint sub-block.enabled.

NOTE: *BDM should not be entered from a breakpoint unless the ENABLE bit is set in the BDM. Even if the ENABLE bit in the BDM is negated, the CPU actually executes the BDM ROM code. It checks the ENABLE and returns if enable is not set. If the BDM is not serviced by the monitor, then the breakpoint would be re-asserted when the BDM returns to normal CPU flow.*

There is no hardware to enforce restriction of breakpoint operation if the BDM is not enabled.

Reset Initialization

The reset state of each individual bit is listed within the Register Description section, which details the registers and their bit-fields. All registers are reset by the system reset.

Interrupts

The BKP sub-block causes the CPU to generate SWI interrupts when BKBDM bit is equal to zero.

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

Freescale Semiconductor, Inc.

Electrical Characteristics

Contents

General 561
ATD Characteristics 569
Flash EEPROM Characteristics 573
Voltage Regulator Characteristics. 577
Reset, Oscillator and PLL Characteristics. 578
SPI Timing. 582
External Bus Timing 587

General

NOTE: *The electrical characteristics given in this section are preliminary and should be used as a guide only. Values cannot be guaranteed by Motorola and are subject to change without notice.*

This section contains the most accurate electrical information for the MC9S12T64 microcontroller available at the time of publication. The information should be considered **PRELIMINARY** and is subject to change.

This introduction is intended to give an overview on several common topics like power supply, current injection etc.

Power Supply

The MC9S12T64 utilizes several pins to supply power to the I/O ports, A/D converter, oscillator and PLL as well as the digital core.

The VDDA, VSSA pair supplies the A/D converter and the resistor ladder of the internal voltage regulator.

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

Freescale Semiconductor, Inc.

The VDDX, VSSX, VDDR and VSSR pairs supply the I/O pins. VDDR supplies also the internal voltage regulator.

VDD1, VSS1, VDD2 and VSS2 are the supply pins for the digital logic, VDDPLL, VSSPLL supply the oscillator and the PLL.

VSS1 and VSS2 are internally connected by metal.

VDDA, VDDX, VDDR as well as VSSA, VSSX, VSSR are connected by anti-parallel diodes for ESD protection.

NOTE: *In the following context VDD5 is used for either VDDA, VDDR and VDDX; VSS5 is used for either VSSA, VSSR and VSSX unless otherwise noted.*
IDD5 denotes the sum of the currents flowing into the VDDA, VDDX and VDDR pins.
VDD is used for VDD1, VDD2 and VDDPLL, VSS is used for VSS1, VSS2 and VSSPLL.
IDD is used for the sum of the currents flowing into VDD1 and VDD2.

Pins

There are four groups of functional pins.

5V I/O pins

Those I/O pins have a nominal level of 5V. This class of pins is comprised of all port I/O pins, the analog inputs, BKGD and the $\overline{\text{RESET}}$ pins.The internal structure of all those pins is identical, however some of the functionality may be disabled. E.g. for the analog inputs the output drivers, pull-up and pull-down resistors are disabled permanently.

Analog Reference

This group is made up by the VRH and VRL pins.

Oscillator

The pins XFC, EXTAL, XTAL dedicated to the oscillator have a nominal 2.5V level. They are supplied by VDDPLL.

TEST

This pin is used for production testing only.

VREGEN

This pin is used to enable the on chip voltage regulator.

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

Freescale Semiconductor, Inc.

Current Injection

Power supply must maintain regulation within operating V_{DD5} or V_{DD} range during instantaneous and operating maximum current conditions. If positive injection current ($V_{in} > V_{DD5}$) is greater than I_{DD5}, the injection current may flow out of VDD5 and could result in external power supply going out of regulation. Ensure external VDD5 load will shunt current greater than maximum injection current. This will be the greatest risk when the MCU is not consuming power; e.g. if no system clock is present, or if clock rate is very low which would reduce overall power consumption.

Absolute Maximum Ratings

Absolute maximum ratings are stress ratings only. A functional operation under or outside those maxima is not guaranteed. Stress beyond those limits may affect the reliability or cause permanent damage of the device.

This device contains circuitry protecting against damage due to high static voltage or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum-rated voltages to this high-impedance circuit. Reliability of

Freescale Semiconductor, Inc.

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

operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS5} or V_{DD5}).

Table 105 Absolute Maximum Ratings(1)

Num	Rating	Symbol	Min	Max	Unit
1	I/O, Regulator and Analog Supply Voltage	V_{DD5}	–0.3	6.0	V
2	Digital Logic Supply Voltage(2)	V_{DD}	–0.3	3.0	V
3	PLL Supply Voltage (2)	V_{DDPLL}	–0.3	3.0	V
4	Voltage difference VDDX to VDDR and VDDA	Δ_{VDDX}	–0.3	0.3	V
5	Voltage difference VSSX to VSSR and VSSA	Δ_{VSSX}	–0.3	0.3	V
6	Digital I/O Input Voltage	V_{IN}	–0.3	6.0	V
7	Analog Reference	V_{RH}, V_{RL}	–0.3	6.0	V
8	XFC, EXTAL, XTAL inputs	V_{ILV}	–0.3	3.0	V
9	TEST input	V_{TEST}	–0.3	10.0	V
10	Instantaneous Maximum Current Single pin limit for all digital I/O pins (3)	I_D	–25	+25	mA
11	Instantaneous Maximum Current Single pin limit for XFC, EXTAL, XTAL(4)	I_{DL}	–25	+25	mA
12	Instantaneous Maximum Current Single pin limit for TEST(5)	I_{DT}	–0.25	0	mA
13	Storage Temperature Range	T_{stg}	–65	155	˚C

1. Functional operation beyond absolute maximum ratings might damage the device.
2. The device contains an internal voltage regulator to generate the logic and PLL supply out of the I/O supply. The absolute maximum ratings apply when the device is powered from an external source.
3. All digital I/O pins are internally clamped to V_{SSX} and V_{DDX}, V_{SSR} and V_{DDR} or V_{SSA} and V_{DDA}.
4. Those pins are internally clamped to V_{SSPLL} and V_{DDPLL}
5. This pin is clamped low to V_{SSPLL}, but not clamped high. This pin must be tied low in applications.

Operating Conditions

This chapter describes the operating conditions of the device. Unless otherwise noted those conditions apply to all the following data.

NOTE: *Please refer to the temperature rating of the device (C, V, M) with regards to the ambient temperature T_A and the junction temperature T_J.*

Freescale Semiconductor, Inc.

For power dissipation calculations refer to the subsection Power Dissipation and Thermal Characteristics.

Table 106 Operating Conditions

Rating	Symbol	Min	Typ	Max	Unit
I/O, Regulator and Analog Supply Voltage	V_{DD5}	4.75	5	5.25	V
Digital Logic Supply Voltage[1]	V_{DD}	2.25	2.5	2.75	V
PLL Supply Voltage [2]	V_{DDPLL}	2.25	2.5	2.75	V
Voltage Difference VDDX to VDDR and VDDA	Δ_{VDDX}	–0.1	0	0.1	V
Voltage Difference VSSX to VSSR and VSSA	Δ_{VSSX}	–0.1	0	0.1	V
Oscillator	f_{osc}	2	—	16	MHz
Bus Frequency	f_{bus} ($1/t_{bus}$)	1	—	16.0	MHz
MC9S12T64**C** Operating Ambient Temperature Range [2] Operating Junction Temperature Range	 T_A T_J	 –40 –40	 27 —	 85 100	˚C
MC9S12T64**V** Operating Ambient Temperature Range [2] Operating Junction Temperature Range	 T_A T_J	 –40 –40	 27 —	 105 120	˚C
MC9S12T64**M** Operating Ambient Temperature Range [2] Operating Junction Temperature Range	 T_A T_J	 –40 –40	 27 —	 125 140	˚C

1. The device contains an internal voltage regulator to generate the logic and PLL supply out of the I/O supply. The absolute maximum ratings apply when this regulator is disabled and the device is powered from an external source.

2. Please refer to the subsection *Power Dissipation and Thermal Characteristics* for more details about the relation between ambient temperature T_A and device junction temperature T_J.

Power Dissipation and Thermal Characteristics

Power dissipation and thermal characteristics are closely related. The user must assure that the maximum operating junction temperature is not exceeded. The average chip-junction temperature (T_J) in ˚C can be obtained from:

$$T_J = T_A + (P_D \bullet \Theta_{JA})$$

T_J = Junction Temperature, [°C]

T_A = Ambient Temperature, [°C]

P_D = Total Chip Power Dissipation, [W]

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

Freescale Semiconductor, Inc.

Θ_{JA} = Package Thermal Resistance, [°C/W]

The total power dissipation can be calculated from:

$$P_D = P_{INT} + P_{IO}$$

P_{INT} = Chip Internal Power Dissipation, [W]

Two cases with internal voltage regulator enabled and disabled must be considered:

1. Internal Voltage Regulator disabled

$$P_{INT} = I_{DD} \cdot V_{DD} + I_{DDPLL} \cdot V_{DDPLL} + I_{DDA} \cdot V_{DDA}$$

$$P_{IO} = \sum_i R_{DSON} \cdot I_{IO_i}^2$$

P_{IO} is the sum of all output currents on I/O ports associated with VDDX and VDDR.

For R_{DSON} is valid:

$$R_{DSON} = \frac{V_{OL}}{I_{OL}}; \text{for outputs driven low}$$

respectively

$$R_{DSON} = \frac{V_{DD5} - V_{OH}}{I_{OH}}; \text{for outputs driven high}$$

2. Internal voltage regulator enabled

$$P_{INT} = I_{DDR} \cdot V_{DDR} + I_{DDA} \cdot V_{DDA}$$

I_{DDR} is the current shown in Table 109 and not the overall current flowing into VDDR, which additionally contains the current flowing into the external loads with output high.

$$P_{IO} = \sum_i R_{DSON} \cdot I_{IO_i}^2$$

P_{IO} is the sum of all output currents on I/O ports associated with VDDX and VDDR.

Freescale Semiconductor, Inc.

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

Table 107 Thermal Package Characteristics(1)

Num	Rating	Symbol	Min	Typ	Max	Unit
1	Thermal Resistance	θ_{JA}	—	—	51	˚C/W

1. The values for thermal resistance are achieved by package simulations

I/O Characteristics

This section describes the characteristics of all 5V I/O pins. All parameters are not always applicable, e.g. not all pins feature pull up/down resistances.

Table 108 5V I/O Characteristics

Conditions are shown in Table 106 unless otherwise noted						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Input High Voltage	V_{IH}	$0.70*V_{DD5}$	—	V_{DD5} + 0.3	V
2	Input Low Voltage	V_{IL}	V_{SS5} – 0.3	—	$0.30*V_{DD5}$	V
3	Input Hysteresis	V_{HYS}		250		mV
4	Input Leakage Current (pins in high ohmic input mode)(1) $V_{in} = V_{DD5}$ or V_{SS5}	I_{in}	–2.5	—	2.5	µA
5	Output High Voltage (pins in output mode) Partial Drive I_{OH} = –1.25mA Full Drive I_{OH} = –3mA	V_{OH}	V_{DD5} – 0.8	—	—	V
6	Output Low Voltage (pins in output mode) Partial Drive I_{OL} = +1.25mA Full Drive I_{OL} = +3mA	V_{OL}	—	—	0.8	V
7	Internal Pull Up Device Current, tested at V_{IL} Max.	I_{PUL}	—	—	–130	µA
8	Internal Pull Up Device Current, tested at V_{IH} Min.	I_{PUH}	–10	—	—	µA
9	Internal Pull Down Device Current, tested at V_{IH} Min.	I_{PDH}	—	—	130	µA
10	Internal Pull Down Device Current, tested at V_{IL} Max.	I_{PDL}	10	—	—	µA
11	Input Capacitance	C_{in}		7	—	pF
12	Injection current(2) Single Pin limit Total Device Limit. Sum of all injected currents	 I_{ICS} I_{ICP}	 –2.5 –25	—	 2.5 25	mA

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

Freescale Semiconductor, Inc.

1. Maximum leakage current occurs at maximum operating temperature. Current decreases by approximately one-half for each 8 C to 12 C in the temperature range from 50˚C to 125˚C.
2. Refer to the subsection Current Injection (page 563), for more details

Supply Currents

This section describes the current consumption characteristics of the device as well as the conditions for the measurements.

Measurement Conditions

All measurements are without output loads. Unless otherwise noted the currents are measured in single chip mode, internal voltage regulator enabled and at 16MHz bus frequency using a 4MHz oscillator in Colpitts mode. Production testing is performed using a square wave signal at the EXTAL input.

Additional Remarks

In expanded modes the currents flowing in the system are highly dependent on the load at the address, data and control signals as well as on the duty cycle of those signals. No generally applicable numbers can be given. A very good estimate is to take the single chip currents and add the currents due to the external loads.

Table 109 Supply Current Characteristics

Conditions are shown in Table 106 unless otherwise noted						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Run supply currents Single Chip Mode	I_{DD5}			30	mA
2	Wait Supply current All modules enabled, PLL on Only RTI enabled, PLL off	I_{DDW}			25 10	mA
3	Pseudo Stop Current (1) RTI and COP enabled RTI and COP disabled	I_{DDPS}		500 250	7,000 6,500	µA
4	Stop Current	I_{DDS}		60	6,000	µA

1. PLL off

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

Freescale Semiconductor, Inc.

ATD Characteristics

This section describes the characteristics of the analog to digital converter.

ATD Operating Characteristics

The Table 110 shows conditions under which the ATD operates.

The following constraints exist to obtain full-scale, full range results: $V_{SSA} \le V_{RL} \le V_{IN} \le V_{RH} \le V_{DDA}$. This constraint exists since the sample buffer amplifier can not drive beyond the power supply levels that it ties to. If the input level goes outside of this range it will effectively be clipped.

Table 110 ATD Operating Characteristics

Conditions are shown in Table 106 unless otherwise noted						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Reference Potential Low High	 V_{RL} V_{RH}	 V_{SSA} $V_{DDA}/2$		 $V_{DDA}/2$ V_{DDA}	 V V
2	Differential Reference Voltage(1)	$V_{RH}-V_{RL}$	4.75	5.0	5.25	V
3	ATD Clock Frequency	f_{ATDCLK}	0.5		2.0	MHz
4	ATD 10-Bit Conversion Period Clock Cycles(2) Conv, Time at 2.0MHz ATD Clock f_{ATDCLK}	 N_{CONV10} T_{CONV10}	 14 7		 28 14	 Cycles µs
5	ATD 8-Bit Conversion Period Clock Cycles(2) Conv, Time at 2.0MHz ATD Clock f_{ATDCLK}	 N_{CONV8} T_{CONV8}	 12 6		 26 13	 Cycles µs
5	Stop Recovery Time (V_{DDA}=5.0 Volts)	t_{SR}			20	µs
6	Reference Supply current	I_{REF}			0.75	mA

1. Full accuracy is not guaranteed when differential voltage is less than 4.75V
2. The minimum time assumes a final sample period of 2 ATD clocks cycles while the maximum time assumes a final sample period of 16 ATD clocks.

Factors Influencing Accuracy

Two factors - source resistance and source capacitance - have an influence on the accuracy of the ATD.

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

Freescale Semiconductor, Inc.

Source Resistance: Due to the input pin leakage current as specified in Table 108 in conjunction with the source resistance there will be a voltage drop from the signal source to the ATD input. The maximum source resistance R_S specifies results in an error of less than 1/2 LSB (2.5mV) at the maximum leakage current. If device or operating conditions are less than worst case or leakage-induced error is acceptable, larger values of source resistance is allowed.

Source Capacitance When sampling an additional internal capacitor is switched to the input. This can cause a voltage drop due to charge sharing with the external and the pin capacitance. For a maximum sampling error of the input voltage ≤ 1LSB, then the external filter capacitor, $C_f \geq 1024 * (C_{INS} - C_{INN})$.

Table 111 ATD Electrical Characteristics

Conditions are shown in Table 106 unless otherwise noted						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Max input Source Resistance	R_S	—	—	1	KΩ
2	Total Input Capacitance Non Sampling Sampling	 C_{INN} C_{INS}			 10 15	pF
3	Disruptive Analog Input Current	I_{NA}	–2.5		2.5	mA

ATD accuracy Table 112 specifies the ATD conversion performance excluding any errors due to current injection, input capacitance and source resistance.

Freescale Semiconductor, Inc.

Table 112 ATD Conversion Performance

Conditions are shown in Table 106 unless otherwise noted $V_{REF} = V_{RH} - V_{RL} = 5.12V$. Resulting to one 8 bit count = 20mV and one 10 bit count = 5mV $f_{ATDCLK} = 2.0MHz$						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	10-Bit Resolution	LSB		5		mV
2	10-Bit Differential Nonlinearity	DNL	–1		1	Counts
3	10-Bit Integral Nonlinearity	INL	–2.5	±1.5	2.5	Counts
4	10-Bit Absolute Error(1)	AE	–3	±2.0	3	Counts
5	8-Bit Resolution	LSB		20		mV
6	8-Bit Differential Nonlinearity	DNL	–0.5		0.5	Counts
7	8-Bit Integral Nonlinearity	INL	–1.0	±0.5	1.0	Counts
8	8-Bit Absolute Error(1)	AE	–1.5	±1.0	1.5	Counts

1. These values include quantization error which is inherently 1/2 count for any A/D converter.

For the following definitions see also Figure 110.

Differential Non-Linearity (DNL) is defined as the difference between two adjacent switching steps.

$$DNL(i) = \frac{V_i - V_{i-1}}{1LSB} - 1$$

The Integral Non-Linearity (INL) is defined as the sum of all DNLs:

$$INL(n) = \sum_{i=1}^{n} DNL(i) = \frac{V_n - V_0}{1LSB} - n$$

Freescale Semiconductor, Inc.

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

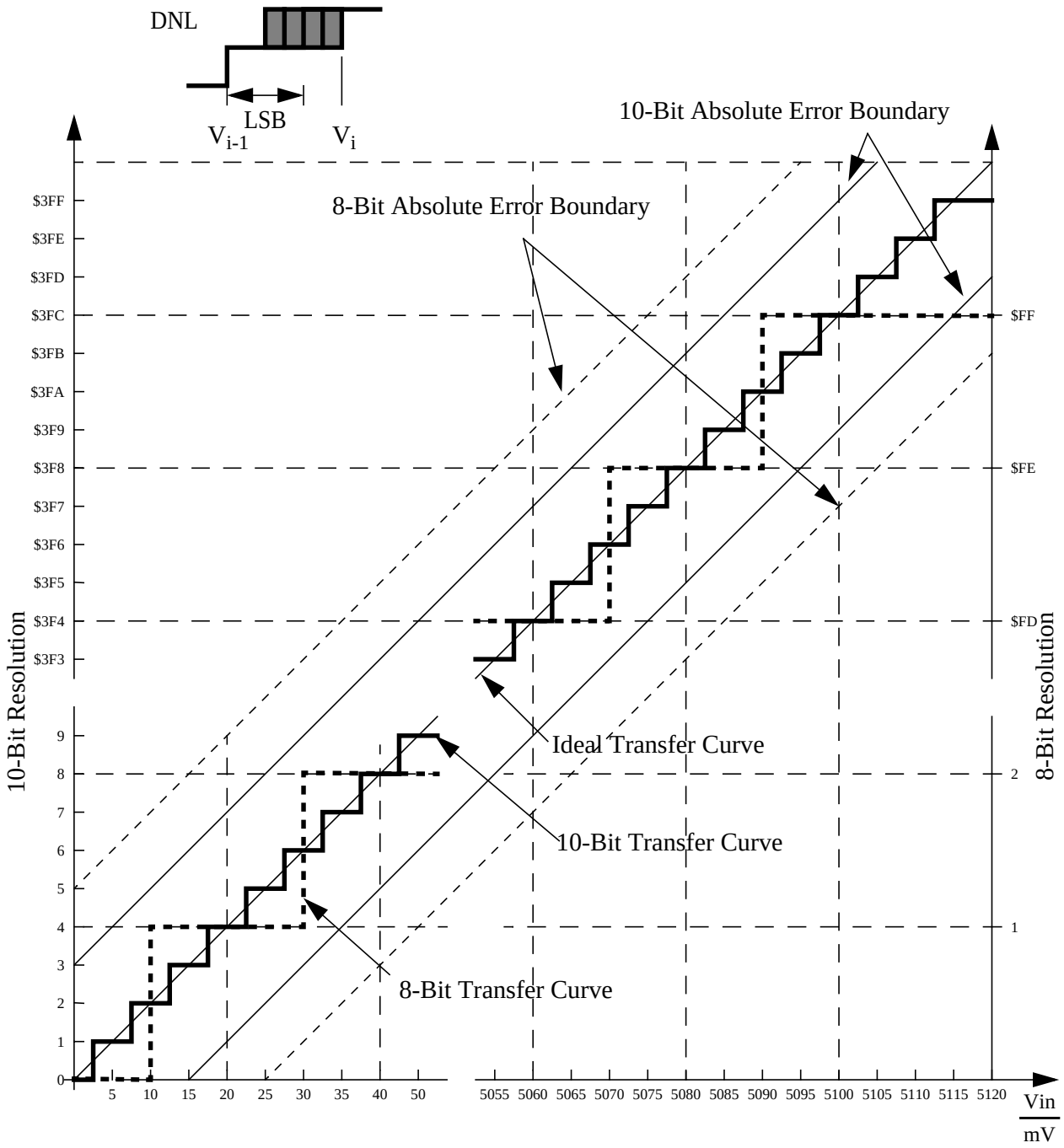


Figure 110 ATD Accuracy Definitions

NOTE: *Figure 110 shows only definitions, for specification values refer to Table 112.*

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

Freescale Semiconductor, Inc.

Flash EEPROM Characteristics

Flash EEPROM timing

The time base for all Flash EEPROM program or erase operations is derived from the oscillator. A minimum oscillator frequency f_{NVMOSC} is required for performing program or erase operations. The Flash EEPROM module does not have any means to monitor the frequency and will not prevent program or erase operation at frequencies above or below the specified minimum. Attempting to program or erase the Flash EEPROM module at a lower frequency a full program or erase transition is not assured.

The Flash EEPROM program and erase operations are timed using a clock derived from the oscillator using the FCLKDIV register. The frequency of this clock must be set within the limits specified as f_{NVMOP}.

The minimum program and erase times shown in Table 113 are calculated for maximum f_{NVMOP} and maximum f_{bus}. The maximum times are calculated for minimum f_{NVMOP} and minimum f_{bus}.

Single Word Programming

The programming time for single word programming is dependant on the bus frequency as a well as on the frequency f_{NVMOP} and can be calculated according to the following formula.

$$t_{swpgm} = 9 \cdot \frac{1}{f_{NVMOP}} + 25 \cdot \frac{1}{f_{bus}}$$

Burst Programming

This applies only to the Flash where up to 32 words in a row can be programmed consecutively using burst programming by keeping the command pipeline filled. The time to program a consecutive word can be calculated as:

$$t_{bwpgm} = 4 \cdot \frac{1}{f_{NVMOP}} + 9 \cdot \frac{1}{f_{bus}}$$

The time to program a whole row is:

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

Freescale Semiconductor, Inc.

$$t_{brpgm} = t_{swpgm} + 31 \cdot t_{bwpgm}$$

Burst programming is more than 2 times faster than single word programming.

Sector Erase

Erasing a 512 byte Flash sector or a 4 byte EEPROM sector takes:

$$t_{era} \approx 4000 \cdot \frac{1}{f_{NVMOP}}$$

The setup times can be ignored for this operation.

Mass Erase

Erasing a NVM block takes:

$$t_{mass} \approx 20000 \cdot \frac{1}{f_{NVMOP}}$$

The setup times can be ignored for this operation.

Blank Check

The time it takes to perform a blank check on the Flash is dependent on the location of the first non-blank word starting at relative address zero. It takes one bus cycle per word to verify plus a setup of the command.

$$t_{check} \cong location \cdot t_{bus} + 10 \cdot t_{bus}$$

Freescale Semiconductor, Inc.

Table 113 NVM Timing Characteristics

Conditions are shown in Table 106 unless otherwise noted						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	External Oscillator Clock	f_{NVMOSC}	2		16 (1)	MHz
2	Bus Frequency for Programming or Erase Operations	f_{NVMBUS}	1			MHz
3	Operating Frequency	f_{NVMOP}	150		200	kHz
4	Single Word Programming Time	t_{swpgm}	46 (2)		85 (3)	µs
5	Flash Burst Programming Time for Consecutive Word	t_{bwpgm}	20.4 (2)		35.7 (3)	µs
6	Flash Burst Programming Time for 32 Words	t_{brpgm}	678.4 (2)		1190.7 (3)	µs
7	Sector Erase Time	t_{era}	20 (4)		26.7 (3)	ms
8	Mass Erase Time	t_{mass}	100 (4)		133 (3)	ms
9	Blank Check Time per Block	t_{check}	11(5)		32778(6)	t_{bus}

1. Restrictions for oscillator in crystal mode apply!
2. Minimum Programming times are achieved under maximum NVM operating frequency f_{NVMOP} and maximum bus frequency f_{bus}.
3. Maximum Erase and Programming times are achieved under particular combinations of f_{NVMOP} and bus frequency f_{bus}. Refer to formulae in subsections from Single Word Programming through Mass Erase for guidance.
4. Minimum Erase times are achieved under maximum NVM operating frequency f_{NVMOP}.
5. Minimum time. If first word in the array is not blank.
6. Maximum time to complete check on erased block.

NVM Reliability

The reliability of the NVM blocks is guaranteed by stress test during qualification, constant process monitors and burn-in to screen early life failures.

The program/erase cycle count on the sector is incremented every time a sector or mass erase event is executed.

NOTE: *All values shown in Table 114 are target values and subject to further extensive characterization.*

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

Freescale Semiconductor, Inc.

Table 114 NVM Reliability Characteristics

Conditions are shown in Table 106 unless otherwise noted				
Num	**Rating**	**Symbol**		**Unit**
1	Data Retention at an average junction temperature of T_{Javg} = 65˚C	t_{NVMRET}	10	Years
2	Flash number of Program/Erase cycles	n_{FLPE}	100	Cycles

Freescale Semiconductor, Inc.

Voltage Regulator Characteristics

The on-chip voltage regulator is intended to supply the internal logic and oscillator circuits. No external DC load is allowed.

Table 115 Voltage Regulator Recommended Load Capacitance

Rating	Symbol	Min	Typ	Max	Unit
Load Capacitance on VDD1, 2	C_{LVDD}		220		nF
Load Capacitance on VDDPLL	C_{LVPLL}		220		nF

Freescale Semiconductor, Inc.

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

Reset, Oscillator and PLL Characteristics

This section summarizes the electrical characteristics of the various startup scenarios for Oscillator and Phase-Locked-Loop (PLL).

Startup

Table 116 summarizes several startup characteristics explained in this section. Detailed description of the startup behavior can be found in the Clocks and Reset Generator (CRG) section in page 271.

Table 116 Startup Characteristics

Conditions are shown in Table 106 unless otherwise noted						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Reset input pulse width, minimum input time	PW_{RSTL}	2			t_{osc}
2	Startup from Reset	n_{RST}	192		196	n_{osc}
3	Interrupt pulse width, $\overline{IRQ}$ edge-sensitive mode	PW_{IRQ}	20			ns
4	Wait recovery startup time	t_{WRS}			14	t_{bus}
5	Low-voltage detector reset release level	V_{LVRR}			4.75	V
6	Low voltage detection level	V_{LVR}	3.9			V
7	Low-voltage detector reset/recover hysteresis	V_{LVHYS}		200		mV
8	POR rearm level (1)	V_{POR}	0		100	mV
9	POR rise time ramp rate (2)	R_{POR}			0.035	V/ms

1. Maximum is highest voltage that POR is guaranteed.
2. If minimum V_{DD} is not reached before the internal POR reset is released, $\overline{RESET}$ must be driven low externally until minimum V_{DD} is reached.

POR and Low-voltage detector (LVD) Reset

The release level V_{LVRR} and the detection level V_{LVR} are derived from the V_{DDR}, while the rearm level V_{POR} is derived from the V_{DD}. They are also valid if the device is powered externally.

External Reset

When external reset is asserted for a time greater than PW_{RSTL} the CRG module generates an internal reset, and the CPU starts fetching the

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

Freescale Semiconductor, Inc.

reset vector without doing a clock quality check, if there was an oscillation before reset.

Stop Recovery

Out of STOP the controller can be woken up by an external interrupt. A clock quality check as after POR is performed before releasing the clocks to the system.

Pseudo Stop and Wait Recovery

The recovery from Pseudo STOP and Wait are essentially the same since the oscillator was not stopped in both modes. The controller can be woken up by internal or external interrupts. After t_{wrs} the CPU starts fetching the interrupt vector.

Oscillator

The device features internal Colpitts and Pierce oscillators. By asserting the $\overline{\text{XCLKS}}$ input during reset the Colpitts oscillator can be bypassed allowing the input of a square wave or the Pierce oscillator. Before asserting the oscillator to the internal system clocks the quality of the oscillation is checked for each start from either power-on, LVD reset, STOP or oscillator fail. t_{CQOUT} specifies the maximum time before switching to the internal self clock mode after POR/LVDR or STOP if a proper oscillation is not detected. The fastest startup time possible is given by n_{uposc}. The quality check also determines the minimum oscillator start-up time t_{UPOSC}. The device also features a clock monitor. A Clock Monitor Failure is asserted if the frequency of the incoming clock signal is below the Clock Monitor Failure Assert Frequency f_{CMFA}.

Freescale Semiconductor, Inc.

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

Table 117 Oscillator Characteristics (1)

Conditions are shown in Table 106 unless otherwise noted						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Crystal oscillator range (Colpitts)	f_{OSC}	2		16	MHz
2	Startup Current	i_{OSC}	100			µA
3	Oscillator start-up time (2)	t_{UPOSC}		8 (3)	400 (4)	ms
4	Clock Quality check time-out	t_{CQOUT}	0.45		2.5	s
5	Clock Monitor Failure Assert Frequency	f_{CMFA}	50	100	200	KHz
6	External square wave input frequency(5)	f_{EXT}	2		32	MHz
7	External square wave pulse width low	t_{EXTL}	30			ns
8	External square wave pulse width high	t_{EXTH}	30			ns
9	External square wave rise time	t_{EXTR}			1	ns
10	External square wave fall time	t_{EXTF}			1	ns
11	Input Capacitance (EXTAL, XTAL inputs)	C_{IN}		9		pF
12	DC Operating Bias in Colpitts Configuration on EXTAL Pin	V_{DCBIAS}		1.1		V

1. When Colpitts Oscillator is selected during reset.
2. The oscillator start-up time is measured as the time necessary to detect a stable ECLK on pin PE4 after POR.
3. f_{osc} = 4MHz, C = 22pF
4. Maximum value is for extreme cases using high Q, low frequency crystals
5. $\overline{XCLKS}$ =0 during reset

Freescale Semiconductor, Inc.

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

Phase Locked Loop

The oscillator provides the reference clock for the PLL. The PLL´s Voltage Controlled Oscillator (VCO) is also the system clock source in self clock mode.

XFC Component Selection

This section describes the selection of the XFC components to achieve a good filter characteristics.

Motorola suggest the use of the following values R= 2.2 KΩ, Cs=33 nF, and Cp = 3.3 nF when f_{REF} = 4MHz, f_{BUS} = 16MHz, f_{VCO} = 32MHz, REFDV = #$00, and SYNR = #$03. See Figure 111. These values are preliminary and should be used as a guide only. Values cannot be guaranteed by Motorola and are subject to change without notice.

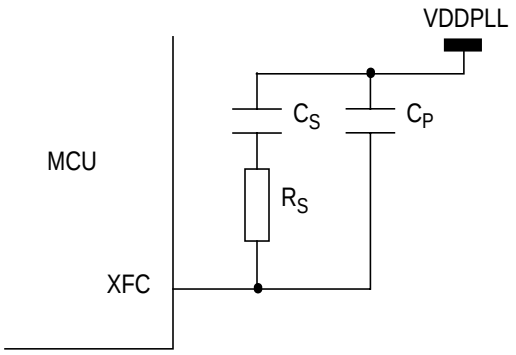


Figure 111 PLL Loop Filter Connections

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

Table 118 PLL Characteristics

Conditions are shown in Table 106 unless otherwise noted						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Self Clock Mode frequency	f_{SCM}		3		MHz
2	VCO locking range	f_{VCO}	8		32	MHz
3	PLL Lock Time (1)	t_{lock}		1		ms
4	Jitter (2) Short term (5µs interval) Long term (500µs interval)	 j_{short} j_{long}	 –0.26 –0.05		 0.26 0.05	%

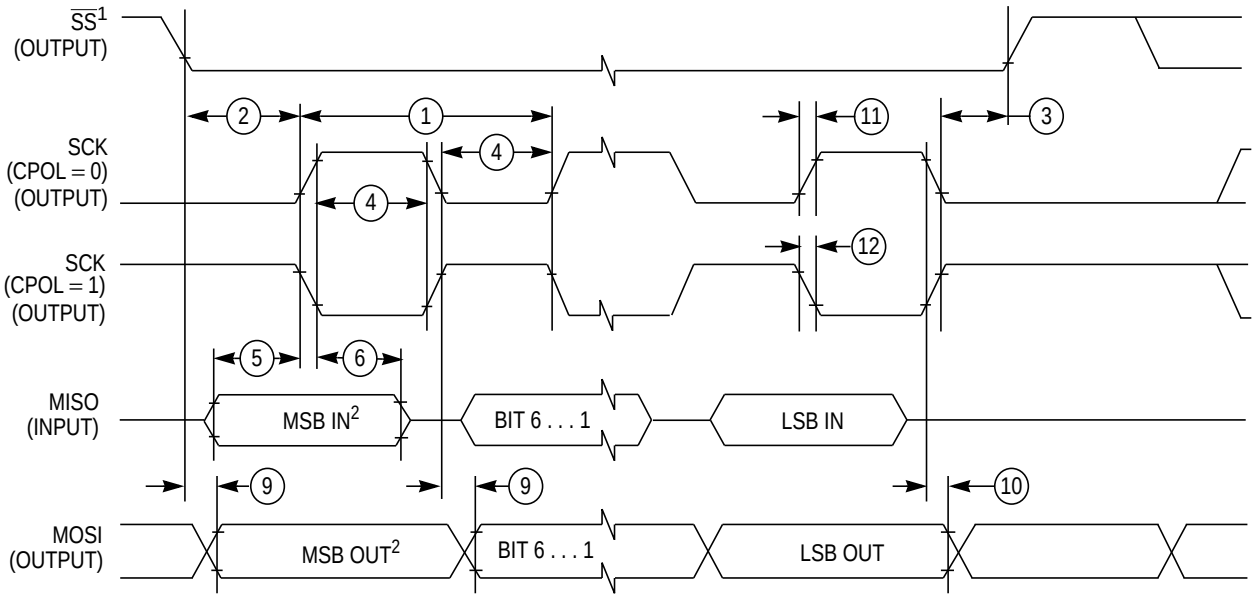
1. Lock time is defined as the time necessary for the PLL to have its LOCK bit asserted after its multiplication factor $2 \cdot (SYNR + 1)/(REVDF + 1)$ was changed.
2. Jitter is defined as the average deviation from the programmed target frequency measured over an specific time interval (5 µs or 500 µs) at maximum f_{VCO}.

SPI Timing

Master Mode

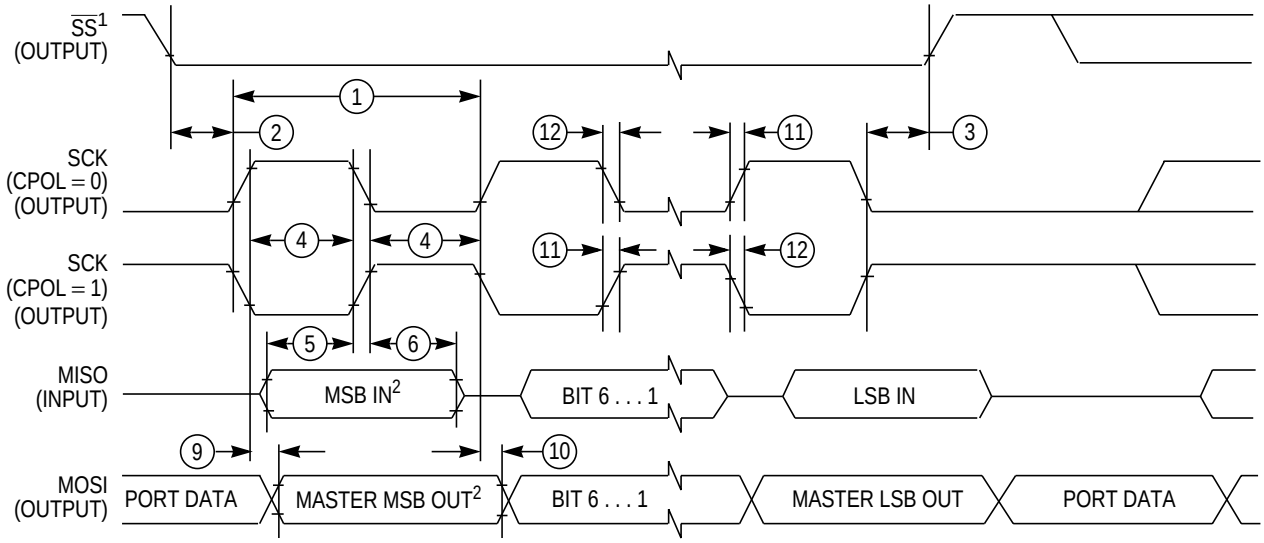
Figure 112 and Figure 113 illustrate the master mode timing. Timing values are shown in Table 119.

Freescale Semiconductor, Inc.



1. If configured as output.
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

Figure 112 SPI Master Timing (CPHA = 0)



1. If configured as output
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

Figure 113 SPI Master Timing (CPHA =1)

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

Freescale Semiconductor, Inc.

Table 119 SPI Master Mode Timing Characteristics(1)

Conditions are shown in Table 106 unless otherwise noted, CLOAD = 200pF on all outputs						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Operating Frequency	f_{op}	DC		1/4	f_{bus}
1	SCK Period t_{sck} = 1/ f_{op}	t_{sck}	4		2048	t_{bus}
2	Enable Lead Time	t_{lead}	1/2		—	t_{sck}
3	Enable Lag Time	t_{lag}	1/2			t_{sck}
4	Clock (SCK) High or Low Time	t_{wsck}	t_{bus} – 30		1024 t_{bus}	ns
5	Data Setup Time (Inputs)	t_{su}	25			ns
6	Data Hold Time (Inputs)	t_{hi}	0			ns
9	Data Valid (after SCK Edge)	t_{v}			25	ns
10	Data Hold Time (Outputs)	t_{ho}	0			ns
11	Rise Time Inputs and Outputs	t_{r}			25	ns
12	Fall Time Inputs and Outputs	t_{f}			25	ns

1. The numbers 7, 8 in the column labeled “Num” are missing. This has been done on purpose to be consistent between the Master and the Slave timing shown in Table 120.

Slave Mode

Figure 114 and Figure 115 illustrate the slave mode timing. Timing values are shown in Table 120.

Freescale Semiconductor, Inc.

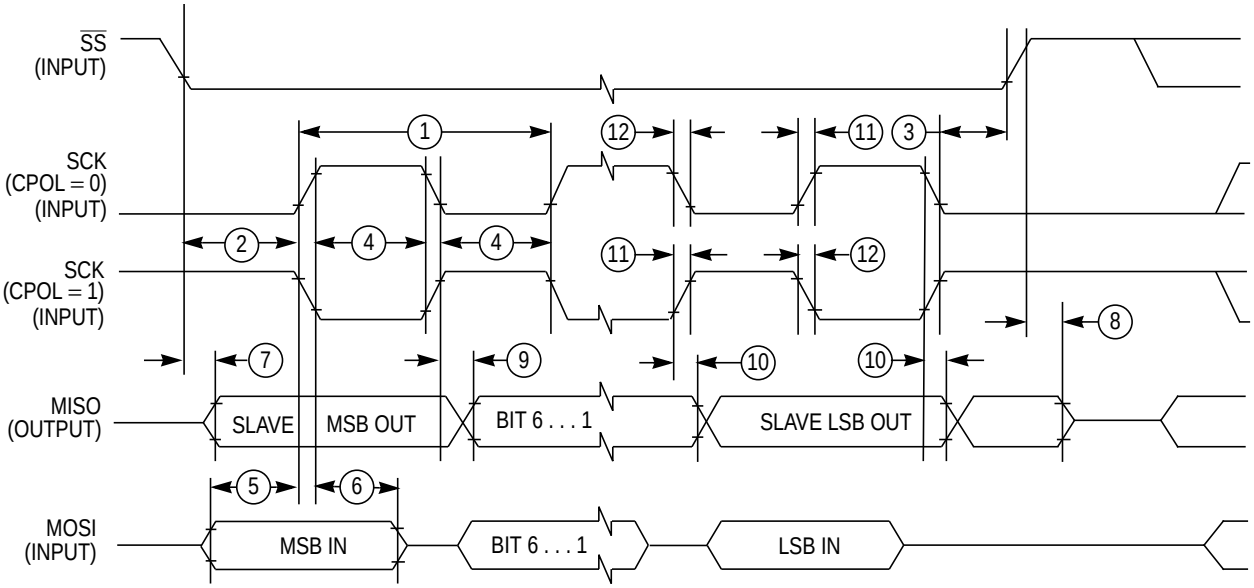


Figure 114 SPI Slave Timing (CPHA = 0)

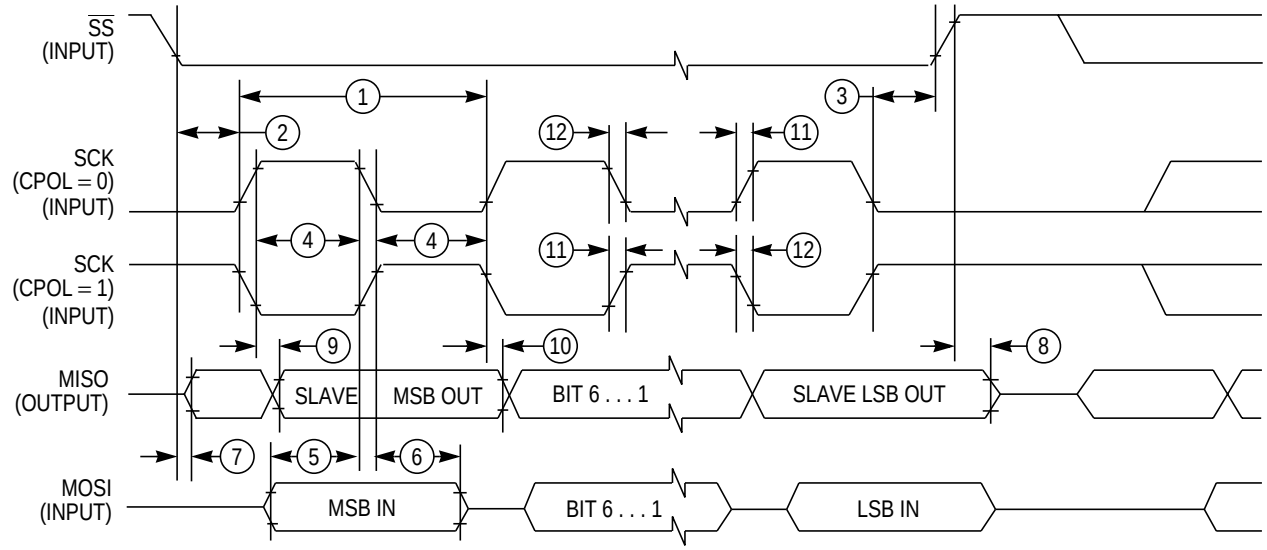


Figure 115 SPI Slave Timing (CPHA =1)

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

Freescale Semiconductor, Inc.

Table 120 SPI Slave Mode Timing Characteristics

Conditions are shown in Table 106 unless otherwise noted, CLOAD = 200pF on all outputs						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Operating Frequency	f_{op}	DC		1/4	f_{bus}
1	SCK Period $f_{sck} = 1 / f_{op}$	t_{sck}	4		2048	t_{bus}
2	Enable Lead Time	t_{lead}	1			t_{bus}
3	Enable Lag Time	t_{lag}	1			t_{bus}
4	Clock (SCK) High or Low Time	t_{wsck}	$t_{bus} - 30$			ns
5	Data Setup Time (Inputs)	t_{su}	25			ns
6	Data Hold Time (Inputs)	t_{hi}	25			ns
7	Slave Access Time	t_a			1	t_{bus}
8	Slave MISO Disable Time	t_{dis}			1	t_{bus}
9	Data Valid (after SCK Edge)	t_v			25	ns
10	Data Hold Time (Outputs)	t_{ho}	0			ns
11	Rise Time Inputs and Outputs	t_r			25	ns
12	Fall Time Inputs and Outputs	t_f			25	ns

Freescale Semiconductor, Inc.

External Bus Timing

A timing diagram of the external multiplexed-bus is illustrated in Figure 116 with the actual timing values shown on table Table 121. All major bus signals are included in the diagram. While both a data write and data read cycle are shown, only one or the other would occur on a particular bus cycle.

General Muxed Bus Timing

The expanded bus timings are highly dependent on the load conditions. The timing parameters shown assume a balanced load across all outputs.

Freescale Semiconductor, Inc.

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

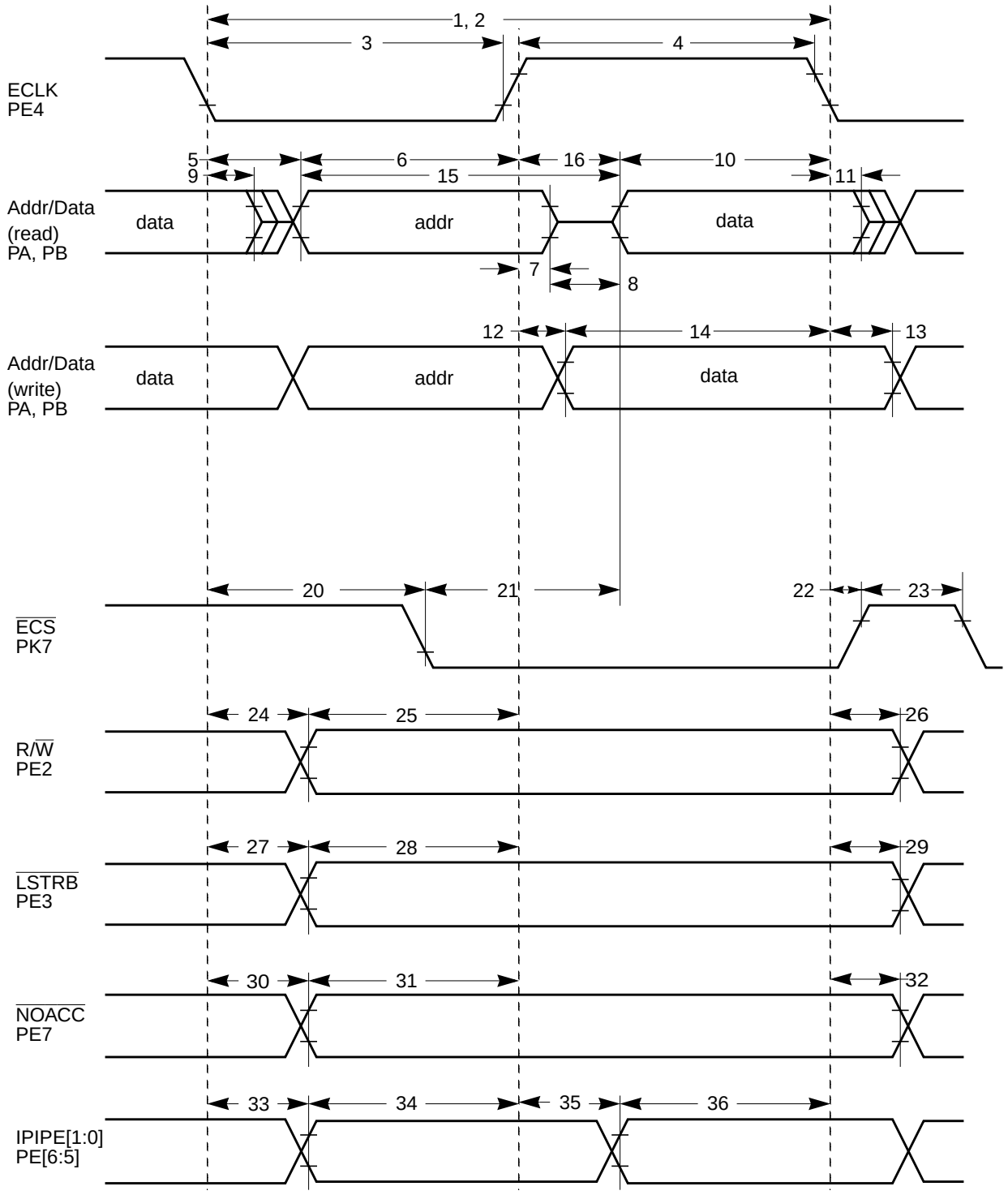


Figure 116 General External Bus Timing

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

Freescale Semiconductor, Inc.

Table 121 Expanded Bus Timing Characteristics – 16MHz

Conditions are shown in Table 106 unless otherwise noted, C_{LOAD} = 50pF						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Frequency of operation (E-clock)	f_o	0		16.0	MHz
2	Cycle time	t_{cyc}	62.5			ns
3	Pulse width, E low	PW_{EL}	29			ns
4	Pulse width, E high(1)	PW_{EH}	29			ns
5	Address delay time	t_{AD}			18	ns
6	Address valid time to E rise (PW_{EL}–t_{AD})	t_{AV}	11			ns
7	Muxed address hold time	t_{MAH}	–5			ns
8	Address hold to data valid	t_{AHDS}	9			ns
9	Data hold to address	t_{DHA}	–2			ns
10	Read data setup time	t_{DSR}	25			ns
11	Read data hold time	t_{DHR}	–5			ns
12	Write data delay time	t_{DDW}			16	ns
13	Write data hold time	t_{DHW}	–3			ns
14	Write data setup time(1) (PW_{EH}–t_{DDW})	t_{DSW}	13			ns
15	Address access time(1) (t_{cyc}–t_{AD}–t_{DSR})	t_{ACCA}	19			ns
16	E high access time(1) (PW_{EH}–t_{DSR})	t_{ACCE}	4			ns
20	Chip select delay time	t_{CSD}			20	ns
21	Chip select access time(1) (t_{cyc}–t_{CSD}–t_{DSR})	t_{ACCS}	17			ns
22	Chip select hold time	t_{CSH}	–5			ns
23	Chip select negated time	t_{CSN}	11			ns
24	Read/write delay time	t_{RWD}			15	ns
25	Read/write valid time to E rise (PW_{EL}–t_{RWD})	t_{RWV}	14			ns
26	Read/write hold time	t_{RWH}	–5			ns
27	Low strobe delay time	t_{LSD}			15	ns
28	Low strobe valid time to E rise (PW_{EL}–t_{LSD})	t_{LSV}	14			ns
29	Low strobe hold time	t_{LSH}	–5			ns
30	NOACC strobe delay time	t_{NOD}			14	ns
31	NOACC valid time to E rise (PW_{EL}–t_{NOD})	t_{NOV}	15			ns
32	NOACC hold time	t_{NOH}	–6			ns
33	ECLK Low to IPIPE[1:0] delay time	t_{P0D}	0		12	ns
34	IPIPE[1:0] valid time to E rise (PW_{EL}–t_{P0D})	t_{P0V}	16			ns

Freescale Semiconductor, Inc.

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

Table 121 Expanded Bus Timing Characteristics – 16MHz (Continued)

Conditions are shown in Table 106 unless otherwise noted, C_{LOAD} = 50pF						
35	ECLK High to IPIPE[1:0] delay time[(1)] (PW_{EH}–t_{P1V})	t_{P1D}	0		40	ns
36	IPIPE[1:0] valid time to E fall	t_{P1V}	16			ns

1. Affected by clock stretch: add N x t_{cyc} where N=0,1,2 or 3, depending on the number of clock stretches.

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

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

Freescale Semiconductor, Inc.

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

Glossary

A — See “accumulators (A and B or D).”

accumulators (A and B or D) — Two 8-bit (A and B) or one 16-bit (D) general-purpose registers in the CPU. The CPU uses the accumulators to hold operands and results of arithmetic and logic operations.

acquisition mode — A mode of PLL operation with large loop bandwidth. Also see “tracking mode”.

address bus — The set of wires that the CPU or DMA uses to read and write memory locations.

addressing mode — The way that the CPU determines the operand address for an instruction. The M68HC12 CPU has 15 addressing modes.

ALU — See “arithmetic logic unit (ALU).”

analogue-to-digital converter (ATD) — The ATD module is an 8-channel, multiplexed-input successive-approximation analog-to-digital converter.

arithmetic logic unit (ALU) — The portion of the CPU that contains the logic circuitry to perform arithmetic, logic, and manipulation operations on operands.

asynchronous — Refers to logic circuits and operations that are not synchronized by a common reference signal.

ATD — See “analogue-to-digital converter”.

B — See “accumulators (A and B or D).”

baud rate — The total number of bits transmitted per unit of time.

BCD — See “binary-coded decimal (BCD).”

binary — Relating to the base 2 number system.

binary number system — The base 2 number system, having two digits, 0 and 1. Binary arithmetic is convenient in digital circuit design because digital circuits have two permissible voltage levels, low and high. The binary digits 0 and 1 can be interpreted to correspond to the two digital voltage levels.

Freescale Semiconductor, Inc.

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

Glossary

binary-coded decimal (BCD) — A notation that uses 4-bit binary numbers to represent the 10 decimal digits and that retains the same positional structure of a decimal number. For example,

234 (decimal) = 0010 0011 0100 (BCD)

bit — A binary digit. A bit has a value of either logic 0 or logic 1.

branch instruction — An instruction that causes the CPU to continue processing at a memory location other than the next sequential address.

break module — The break module allows software to halt program execution at a programmable point in order to enter a background routine.

breakpoint — A number written into the break address registers of the break module. When a number appears on the internal address bus that is the same as the number in the break address registers, the CPU executes the software interrupt instruction (SWI).

break interrupt — A software interrupt caused by the appearance on the internal address bus of the same value that is written in the break address registers.

bus — A set of wires that transfers logic signals.

bus clock — See “CPU clock”.

byte — A set of eight bits.

CAN — See “Motorola scalable CAN.”

CCR — See “condition code register.”

central processor unit (CPU) — The primary functioning unit of any computer system. The CPU controls the execution of instructions.

CGM — See “clock generator module (CGM).”

clear — To change a bit from logic 1 to logic 0; the opposite of set.

clock — A square wave signal used to synchronize events in a computer.

Freescale Semiconductor, Inc.

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

clock generator module (CGM) — The CGM module generates a base clock signal from which the system clocks are derived. The CGM may include a crystal oscillator circuit and/or phase-locked loop (PLL) circuit.

comparator — A device that compares the magnitude of two inputs. A digital comparator defines the equality or relative differences between two binary numbers.

computer operating properly module (COP) — A counter module that resets the MCU if allowed to overflow.

condition code register (CCR) — An 8-bit register in the CPU that contains the interrupt mask bit and five bits that indicate the results of the instruction just executed.

control bit — One bit of a register manipulated by software to control the operation of the module.

control unit — One of two major units of the CPU. The control unit contains logic functions that synchronize the machine and direct various operations. The control unit decodes instructions and generates the internal control signals that perform the requested operations. The outputs of the control unit drive the execution unit, which contains the arithmetic logic unit (ALU), CPU registers, and bus interface.

COP — See “computer operating properly module (COP).”

CPU — See “central processor unit (CPU).”

CPU12 — The CPU of the MC68HC12 Family.

CPU clock — Bus clock select bits BCSP and BCSS in the clock select register (CLKSEL) determine which clock drives SYSCLK for the main system, including the CPU and buses. When EXTALi drives the SYSCLK, the CPU or bus clock frequency (f_o) is equal to the EXTALi frequency divided by 2.

CPU cycles — A CPU cycle is one period of the internal bus clock, normally derived by dividing a crystal oscillator source by two or more so the high and low times will be equal. The length of time required to execute an instruction is measured in CPU clock cycles.

Freescale Semiconductor, Inc.

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

Glossary

CPU registers — Memory locations that are wired directly into the CPU logic instead of being part of the addressable memory map. The CPU always has direct access to the information in these registers. The CPU registers in an M68HC12 are:

- A (8-bit accumulator)
- B (8-bit accumulator)
  - D (16-bit accumulator formed by concatenation of accumulators A and B)
- IX (16-bit index register)
- IY (16-bit index register)
- SP (16-bit stack pointer)
- PC (16-bit program counter)
- CCR (8-bit condition code register)

cycle time — The period of the operating frequency: $t_{CYC} = 1/f_{OP}$.

D — See “accumulators (A and B or D).”

decimal number system — Base 10 numbering system that uses the digits zero through nine.

duty cycle — A ratio of the amount of time the signal is on versus the time it is off. Duty cycle is usually represented by a percentage.

ECT — See “enhanced capture timer.”

EEPROM — Electrically erasable, programmable, read-only memory. A nonvolatile type of memory that can be electrically erased and reprogrammed.

EPROM — Erasable, programmable, read-only memory. A nonvolatile type of memory that can be erased by exposure to an ultraviolet light source and then reprogrammed.

enhanced capture timer (ECT) — The HC12 Enhanced Capture Timer module has the features of the HC12 Standard Timer module enhanced by additional features in order to enlarge the field of applications.

exception — An event such as an interrupt or a reset that stops the sequential execution of the instructions in the main program.

fetch — To copy data from a memory location into the accumulator.

firmware — Instructions and data programmed into nonvolatile memory.

Freescale Semiconductor, Inc.

flash EEPROM — Electrically erasable, programmable, read-only memory. A nonvolatile type of memory that can be electrically erased and reprogrammed. Does not support byte or word erase.

free-running counter — A device that counts from zero to a predetermined number, then rolls over to zero and begins counting again.

full-duplex transmission — Communication on a channel in which data can be sent and received simultaneously.

hexadecimal — Base 16 numbering system that uses the digits 0 through 9 and the letters A through F.

high byte — The most significant eight bits of a word.

illegal address — An address not within the memory map

illegal opcode — A nonexistent opcode.

index registers (IX and IY) — Two 16-bit registers in the CPU. In the indexed addressing modes, the CPU uses the contents of IX or IY to determine the effective address of the operand. IX and IY can also serve as a temporary data storage locations.

input/output (I/O) — Input/output interfaces between a computer system and the external world. A CPU reads an input to sense the level of an external signal and writes to an output to change the level on an external signal.

instructions — Operations that a CPU can perform. Instructions are expressed by programmers as assembly language mnemonics. A CPU interprets an opcode and its associated operand(s) and instruction.

interrupt — A temporary break in the sequential execution of a program to respond to signals from peripheral devices by executing a subroutine.

interrupt request — A signal from a peripheral to the CPU intended to cause the CPU to execute a subroutine.

I/O — See “input/output (I/0).”

jitter — Short-term signal instability.

latch — A circuit that retains the voltage level (logic 1 or logic 0) written to it for as long as power is applied to the circuit.

latency — The time lag between instruction completion and data movement.

least significant bit (LSB) — The rightmost digit of a binary number.

Freescale Semiconductor, Inc.

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

Glossary

logic 1 — A voltage level approximately equal to the input power voltage (V_{DD}).

logic 0 — A voltage level approximately equal to the ground voltage (V_{SS}).

low byte — The least significant eight bits of a word.

M68HC12 — A Motorola family of 16-bit MCUs.

mark/space — The logic 1/logic 0 convention used in formatting data in serial communication.

mask — 1. A logic circuit that forces a bit or group of bits to a desired state. 2. A photomask used in integrated circuit fabrication to transfer an image onto silicon.

MCU — Microcontroller unit. See “microcontroller.”

memory location — Each M68HC12 memory location holds one byte of data and has a unique address. To store information in a memory location, the CPU places the address of the location on the address bus, the data information on the data bus, and asserts the write signal. To read information from a memory location, the CPU places the address of the location on the address bus and asserts the read signal. In response to the read signal, the selected memory location places its data onto the data bus.

memory map — A pictorial representation of all memory locations in a computer system.

MI-Bus — See “Motorola interconnect bus”.

microcontroller — Microcontroller unit (MCU). A complete computer system, including a CPU, memory, a clock oscillator, and input/output (I/O) on a single integrated circuit.

modulo counter — A counter that can be programmed to count to any number from zero to its maximum possible modulus.

most significant bit (MSB) — The leftmost digit of a binary number.

Motorola interconnect bus (MI-Bus) — The Motorola Interconnect Bus (MI Bus) is a serial communications protocol which supports distributed real-time control efficiently and with a high degree of noise immunity.

Motorola scalable CAN (msCAN) — The Motorola scalable controller area network is a serial communications protocol that efficiently supports distributed real-time control with a very high level of data integrity.

msCAN — See “Motorola scalable CAN”.

MSI — See “multiple serial interface”.

multiple serial interface — A module consisting of multiple independent serial I/O sub-systems, e.g. two SCI and one SPI.

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

Freescale Semiconductor, Inc.

multiplexer — A device that can select one of a number of inputs and pass the logic level of that input on to the output.

nibble — A set of four bits (half of a byte).

object code — The output from an assembler or compiler that is itself executable machine code, or is suitable for processing to produce executable machine code.

opcode — A binary code that instructs the CPU to perform an operation.

open-drain — An output that has no pullup transistor. An external pullup device can be connected to the power supply to provide the logic 1 output voltage.

operand — Data on which an operation is performed. Usually a statement consists of an operator and an operand. For example, the operator may be an add instruction, and the operand may be the quantity to be added.

oscillator — A circuit that produces a constant frequency square wave that is used by the computer as a timing and sequencing reference.

OTPROM — One-time programmable read-only memory. A nonvolatile type of memory that cannot be reprogrammed.

overflow — A quantity that is too large to be contained in one byte or one word.

page zero — The first 256 bytes of memory (addresses $0000–$00FF).

parity — An error-checking scheme that counts the number of logic 1s in each byte transmitted. In a system that uses odd parity, every byte is expected to have an odd number of logic 1s. In an even parity system, every byte should have an even number of logic 1s. In the transmitter, a parity generator appends an extra bit to each byte to make the number of logic 1s odd for odd parity or even for even parity. A parity checker in the receiver counts the number of logic 1s in each byte. The parity checker generates an error signal if it finds a byte with an incorrect number of logic 1s.

PC — See “program counter (PC).”

peripheral — A circuit not under direct CPU control.

phase-locked loop (PLL) — A clock generator circuit in which a voltage controlled oscillator produces an oscillation which is synchronized to a reference signal.

PLL — See “phase-locked loop (PLL).”

pointer — Pointer register. An index register is sometimes called a pointer register because its contents are used in the calculation of the address of an operand, and therefore points to the operand.

Freescale Semiconductor, Inc.

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

Glossary

polarity — The two opposite logic levels, logic 1 and logic 0, which correspond to two different voltage levels, V_{DD} and V_{SS}.

polling — Periodically reading a status bit to monitor the condition of a peripheral device.

port — A set of wires for communicating with off-chip devices.

prescaler — A circuit that generates an output signal related to the input signal by a fractional scale factor such as 1/2, 1/8, 1/10 etc.

program — A set of computer instructions that cause a computer to perform a desired operation or operations.

program counter (PC) — A 16-bit register in the CPU. The PC register holds the address of the next instruction or operand that the CPU will use.

pull — An instruction that copies into the accumulator the contents of a stack RAM location. The stack RAM address is in the stack pointer.

pullup — A transistor in the output of a logic gate that connects the output to the logic 1 voltage of the power supply.

pulse-width — The amount of time a signal is on as opposed to being in its off state.

pulse-width modulation (PWM) — Controlled variation (modulation) of the pulse width of a signal with a constant frequency.

push — An instruction that copies the contents of the accumulator to the stack RAM. The stack RAM address is in the stack pointer.

PWM period — The time required for one complete cycle of a PWM waveform.

RAM — Random access memory. All RAM locations can be read or written by the CPU. The contents of a RAM memory location remain valid until the CPU writes a different value or until power is turned off.

RC circuit — A circuit consisting of capacitors and resistors having a defined time constant.

read — To copy the contents of a memory location to the accumulator.

register — A circuit that stores a group of bits.

reserved memory location — Writing to a reserved location has no effect. Reading a reserved location always returns ZERO.

reset — To force a device to a known condition.

Freescale Semiconductor, Inc.

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

ROM — Read-only memory. A type of memory that can be read but cannot be changed (written). The contents of ROM must be specified before manufacturing the MCU.

SCI — See “serial communication interface module (SCI).”

serial — Pertaining to sequential transmission over a single line.

serial communications interface module (SCI) — A module that supports asynchronous communication.

serial peripheral interface module (SPI) — A module that supports synchronous communication.

set — To change a bit from logic 0 to logic 1; opposite of clear.

shift register — A chain of circuits that can retain the logic levels (logic 1 or logic 0) written to them and that can shift the logic levels to the right or left through adjacent circuits in the chain.

signed — A binary number notation that accommodates both positive and negative numbers. The most significant bit is used to indicate whether the number is positive or negative, normally logic 0 for positive and logic 1 for negative. The other seven bits indicate the magnitude of the number.

software — Instructions and data that control the operation of a microcontroller.

software interrupt (SWI) — An instruction that causes an interrupt and its associated vector fetch.

SPI — See “serial peripheral interface module (SPI).”

stack — A portion of RAM reserved for storage of CPU register contents and subroutine return addresses.

stack pointer (SP) — A 16-bit register in the CPU containing the address of the next available storage location on the stack.

start bit — A bit that signals the beginning of an asynchronous serial transmission.

status bit — A register bit that indicates the condition of a device.

stop bit — A bit that signals the end of an asynchronous serial transmission.

Freescale Semiconductor, Inc.

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

Glossary

subroutine — A sequence of instructions to be used more than once in the course of a program. The last instruction in a subroutine is a return from subroutine (RTS) instruction. At each place in the main program where the subroutine instructions are needed, a jump or branch to subroutine (JSR or BSR) instruction is used to call the subroutine. The CPU leaves the flow of the main program to execute the instructions in the subroutine. When the RTS instruction is executed, the CPU returns to the main program where it left off.

synchronous — Refers to logic circuits and operations that are synchronized by a common reference signal.

timer — A module used to relate events in a system to a point in time.

toggle — To change the state of an output from a logic 0 to a logic 1 or from a logic 1 to a logic 0.

tracking mode — A mode of PLL operation with narrow loop bandwidth. Also see ‘acquisition mode.’

two’s complement — A means of performing binary subtraction using addition techniques. The most significant bit of a two’s complement number indicates the sign of the number (1 indicates negative). The two’s complement negative of a number is obtained by inverting each bit in the number and then adding 1 to the result.

unbuffered — Utilizes only one register for data; new data overwrites current data.

unimplemented memory location — A memory location that is not used. Writing to an unimplemented location has no effect. Reading an unimplemented location returns an unpredictable value.

variable — A value that changes during the course of program execution.

VCO — See “voltage-controlled oscillator.”

vector — A memory location that contains the address of the beginning of a subroutine written to service an interrupt or reset.

voltage-controlled oscillator (VCO) — A circuit that produces an oscillating output signal of a frequency that is controlled by a dc voltage applied to a control input.

waveform — A graphical representation in which the amplitude of a wave is plotted against time.

wired-OR — Connection of circuit outputs so that if any output is high, the connection point is high.

word — A set of two bytes (16 bits).

write — The transfer of a byte of data from the CPU to a memory location.

Freescale Semiconductor, Inc.

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

Literature Updates

This document contains the latest data available at publication time. For updates, contact one of the centers listed below:

Literature Distribution Centers

Order literature by mail or phone.

USA/Europe

Motorola Literature Distribution
P.O. Box 5405
Denver, Colorado, 80217
Phone 1-303-675-2140

US & Canada only

http://sps.motorola.com/mfax

Japan

Motorola Japan Ltd.
Tatsumi-SPD-JLDC
Toshikatsu Otsuki
6F Seibu-Butsuryu Center
3-14-2 Tatsumi Koto-Ku
Tokyo 135, Japan
Phone 03-3521-8315

Freescale Semiconductor, Inc.

Hong Kong

Motorola Semiconductors H.K. Ltd.
8B Tai Ping Industrial Park
51 Ting Kok Road
Tai Po, N.T., Hong Kong
Phone 852-26629298

Customer Focus Center

1-800-521-6274

Mfax

To access this worldwide faxing service call or contact by electronic mail or the internet:

RMFAX0@email.sps.mot.com
TOUCH-TONE 1-602-244-6609
http://sps.motorola.com/mfax

Motorola SPS World Marketing World Wide Web Server

Use the Internet to access Motorola's World Wide Web server. Use the following URL:

http://design-net.com

Microcontroller Division's Web Site

Directly access the Microcontroller Division's web site with the following URL:

http://design-net.com/csic/CSIC_home.html

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 MC9S12T64MPKE16 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