MC68HC908SR12
MC68HC08SR12

Data Sheet

M68HC08
Microcontrollers

MC68HC908SR12
Rev. 5.0
07/2004

freescale.com

freescale™
semiconductor

MC68HC908SR12
MC68HC08SR12

Data Sheet

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://www.freescale.com

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc.
This product incorporates SuperFlash® technology licensed from SST.
© Freescale Semiconductor, Inc., 2004

Revision History

Date	Revision Level	Description	Page Number(s)
July 2004	5	**Table 24-2 . Operating Range** and **Table 24-11 . 3V ADC Electrical Characteristics** — changed minimum V_{DD} for ADC operation to 3V.	373, 381
		15.8.4 ADC Auto-Scan Mode Data Registers (ADRL1–ADRL3) — Corrected ADRL–ADRL3 register bits.	248
February, 2002	4	PTB0/SDA0, PTB1/SCL0, PTB2/SDA1/TxD, and PTB3/SCL1/RxD pins — clarified these open-drain pins throughout this document.	323, 254, 293
		8.4.6 Programming the PLL — deleted redundant step in programming the PLL.	120
		Figure 10-1 . Monitor Mode Circuit — corrected connections for PTA1 and PTA2.	167
		Table 10-1 . Monitor Mode Signal Requirements and Options — clarified clock input requirements for monitor mode entry.	169
		Section 11. Timer Interface Module (TIM) — timer discrepancies corrected throughout this section.	181
		18.5.1 Port C Data Register (PTC) and **18.5.2 Data Direction Register C (DDRC)** — added notes for PTC6 and PTC7 on 42-pin package.	327, 329
		Figure 19-3 . IRQ2 Block Diagram and **19.5 IRQ1 and IRQ2 Pins** — corrected $\overline{IRQ2}$ for BIH and BIL instructions.	338, 339
		Table 24-4 . 5V DC Electrical Characteristics and **Table 24-5 . 3V DC Electrical Characteristics** — added additional I_{DD} measurements.	374, 376
		Table 24-13 . Current Detection Electrical Characteristics — updated trip point values.	382
		Appendix A. MC68HC08SR12 — added appendix for ROM part: MC68HC08SR12.	393

List of Sections

Section 1. General Description 35
Section 2. Memory Map 45
Section 3. Random-Access Memory (RAM) 61
Section 4. FLASH Memory 63
Section 5. Configuration and Mask Option Registers (CONFIG & MOR). 73
Section 6. Central Processor Unit (CPU) 81
Section 7. Oscillator (OSC) 101
Section 8. Clock Generator Module (CGM). 111
Section 9. System Integration Module (SIM) 141
Section 10. Monitor ROM (MON) 165
Section 11. Timer Interface Module (TIM) 181
Section 12. Timebase Module (TBM). 205
Section 13. Pulse Width Modulator (PWM). 211
Section 14. Analog Module 221
Section 15. Analog-to-Digital Converter (ADC) 231
Section 16. Serial Communications Interface (SCI). . .251
Section 17. Multi-Master IIC Interface (MMIIC) 291
Section 18. Input/Output (I/O) Ports 317
Section 19. External Interrupt (IRQ) 335

List of Sections

Section 20. Keyboard Interrupt Module (KBI). 343
Section 21. Computer Operating Properly (COP) 351
Section 22. Low-Voltage Inhibit (LVI) 357
Section 23. Break Module (BRK) 363
Section 24. Electrical Specifications. 371
Section 25. Mechanical Specifications 387
Section 26. Ordering Information 391
Appendix A. MC68HC08SR12 393

Table of Contents

Section 1. General Description

1.1 Contents 35

1.2 Introduction 36

1.3 Features 36

1.4 MCU Block Diagram 38

1.5 Pin Assignments 40

1.6 Pin Functions 41
1.6.1 Power Supply Pins (V_{DD} and V_{SS}) 42
1.6.2 Oscillator Pins (OSC1 and OSC2) 42
1.6.3 External Reset Pin ($\overline{RST}$) 43
1.6.4 External Interrupt Pin ($\overline{IRQ1}$) 43
1.6.5 Analog Power Supply Pin (V_{DDA}) 43
1.6.6 Analog Ground Pin (V_{SSA}) 43
1.6.7 ADC Voltage Low Reference Pin (V_{REFL}) 43
1.6.8 ADC Voltage High Reference Pin (V_{REFH}) 43
1.6.9 External Filter Capacitor Pin (CGMXFC) 43
1.6.10 Analog Input Pins (OPIN1/ATD0, OPIN2/ATD1, V_{SSAM}) . . . 44
1.6.11 Port A Input/Output (I/O) Pins (PTA7–PTA0) 44
1.6.12 Port B I/O Pins (PTB6–PTB0) 44
1.6.13 Port C I/O Pins (PTC7–PTC0) 44
1.6.14 Port D I/O Pins (PTD7/KBI7–PTD0/KBI0) 44

Section 2. Memory Map

2.1 Contents 45

2.2 Introduction 45

2.3 Unimplemented Memory Locations 45

2.4 Reserved Memory Locations 46
2.5 Input/Output (I/O) Section. 46

Section 3. Random-Access Memory (RAM)

3.1 Contents 61
3.2 Introduction. 61
3.3 Functional Description 61

Section 4. FLASH Memory

4.1 Contents 63
4.2 Introduction. 63
4.3 Functional Description 64
4.4 FLASH Control Register 65
4.5 FLASH Page Erase Operation 66
4.6 FLASH Mass Erase Operation 67
4.7 FLASH Program Operation. 68
4.8 FLASH Protection. 70
4.8.1 FLASH Block Protect Register 70

Section 5. Configuration and Mask Option Registers (CONFIG & MOR)

5.1 Contents 73
5.2 Introduction. 73
5.3 Functional Description 74
5.4 Configuration Register 1 (CONFIG1) 75
5.5 Configuration Register 2 (CONFIG2) 77
5.6 Mask Option Register (MOR) 79

Section 6. Central Processor Unit (CPU)

6.1 Contents 81
6.2 Introduction 81
6.3 Features 82
6.4 CPU Registers 82
6.4.1 Accumulator 83
6.4.2 Index Register 84
6.4.3 Stack Pointer 84
6.4.4 Program Counter 85
6.4.5 Condition Code Register 85
6.5 Arithmetic/Logic Unit (ALU) 88
6.6 Low-Power Modes 88
6.6.1 Wait Mode 88
6.6.2 Stop Mode 89
6.7 CPU During Break Interrupts 89
6.8 Instruction Set Summary 89
6.9 Opcode Map 89

Section 7. Oscillator (OSC)

7.1 Contents 101
7.2 Introduction 102
7.3 Clock Selection 103
7.3.1 CGM Reference Clock Selection 104
7.3.2 TBM Reference Clock Selection 105
7.4 Internal Oscillator 105
7.5 RC Oscillator 106
7.6 X-tal Oscillator 107
7.7 I/O Signals 108
7.7.1 Crystal Amplifier Input Pin (OSC1) 108
7.7.2 Crystal Amplifier Output Pin (OSC2) 109

7.7.3 Oscillator Enable Signal (SIMOSCEN) 109
7.7.4 CGM Oscillator Clock (CGMXCLK) 109
7.7.5 CGM Reference Clock (CGMRCLK) 109
7.7.6 Oscillator Clock to Time Base Module (OSCCLK) 109

7.8 Low Power Modes 109
7.8.1 Wait Mode 110
7.8.2 Stop Mode 110

7.9 Oscillator During Break Mode 110

Section 8. Clock Generator Module (CGM)

8.1 Contents 111

8.2 Introduction 112

8.3 Features 113

8.4 Functional Description 113
8.4.1 Oscillator Module 116
8.4.2 Phase-Locked Loop Circuit (PLL) 116
8.4.3 PLL Circuits 116
8.4.4 Acquisition and Tracking Modes 118
8.4.5 Manual and Automatic PLL Bandwidth Modes 118
8.4.6 Programming the PLL 120
8.4.7 Special Programming Exceptions 124
8.4.8 Base Clock Selector Circuit 124
8.4.9 CGM External Connections 125

8.5 I/O Signals 125
8.5.1 External Filter Capacitor Pin (CGMXFC) 126
8.5.2 PLL Analog Power Pin (V_{DDA}) 126
8.5.3 PLL Analog Ground Pin (V_{SSA}) 126
8.5.4 Oscillator Output Frequency Signal (CGMXCLK) 126
8.5.5 CGM Reference Clock (CGMRCLK) 126
8.5.6 CGM VCO Clock Output (CGMVCLK) 127
8.5.7 CGM Base Clock Output (CGMOUT) 127
8.5.8 CGM CPU Interrupt (CGMINT) 127

8.6 CGM Registers 127
8.6.1 PLL Control Register 128

8.6.2 PLL Bandwidth Control Register 130
8.6.3 PLL Multiplier Select Registers 132
8.6.4 PLL VCO Range Select Register 133
8.6.5 PLL Reference Divider Select Register 134

8.7 Interrupts 135

8.8 Special Modes 135
8.8.1 Wait Mode 135
8.8.2 Stop Mode 136
8.8.3 CGM During Break Interrupts 136

8.9 Acquisition/Lock Time Specifications 137
8.9.1 Acquisition/Lock Time Definitions 137
8.9.2 Parametric Influences on Reaction Time 137
8.9.3 Choosing a Filter 139

Section 9. System Integration Module (SIM)

9.1 Contents 141

9.2 Introduction 142

9.3 SIM Bus Clock Control and Generation 144
9.3.1 Bus Timing 145
9.3.2 Clock Start-up from POR or LVI Reset 145
9.3.3 Clocks in Stop Mode and Wait Mode 146

9.4 Reset and System Initialization 146
9.4.1 External Pin Reset 147
9.4.2 Active Resets from Internal Sources 147
9.4.2.1 Power-On Reset 148
9.4.2.2 Computer Operating Properly (COP) Reset 149
9.4.2.3 Illegal Opcode Reset 150
9.4.2.4 Illegal Address Reset 150
9.4.2.5 Low-Voltage Inhibit (LVI) Reset 150
9.4.2.6 Monitor Mode Entry Module Reset 150

9.5 SIM Counter 151
9.5.1 SIM Counter During Power-On Reset 151
9.5.2 SIM Counter During Stop Mode Recovery 151
9.5.3 SIM Counter and Reset States 151

9.6 Exception Control 152
9.6.1 Interrupts 152
9.6.1.1 Hardware Interrupts 154
9.6.1.2 SWI Instruction 155
9.6.1.3 Interrupt Status Registers 155
9.6.1.4 Interrupt Status Register 1 155
9.6.1.5 Interrupt Status Register 2 157
9.6.1.6 Interrupt Status Register 3 157
9.6.2 Reset 158
9.6.3 Break Interrupts 158
9.6.4 Status Flag Protection in Break Mode 158

9.7 Low-Power Modes 159
9.7.1 Wait Mode 159
9.7.2 Stop Mode 160

9.8 SIM Registers 161
9.8.1 SIM Break Status Register 162
9.8.2 SIM Reset Status Register 163
9.8.3 SIM Break Flag Control Register 164

Section 10. Monitor ROM (MON)

10.1 Contents 165

10.2 Introduction 165

10.3 Features 166

10.4 Functional Description 166
10.4.1 Entering Monitor Mode 168
10.4.2 Data Format 172
10.4.3 Break Signal 172
10.4.4 Baud Rate 173
10.4.5 Commands 173

10.5 Security 178

Section 11. Timer Interface Module (TIM)

11.1 Contents 181
11.2 Introduction 182
11.3 Features 182
11.4 Pin Name Conventions 183
11.5 Functional Description 183
11.5.1 TIM Counter Prescaler 187
11.5.2 Input Capture 187
11.5.3 Output Compare 188
11.5.3.1 Unbuffered Output Compare 188
11.5.3.2 Buffered Output Compare 189
11.5.4 Pulse Width Modulation (PWM) 189
11.5.4.1 Unbuffered PWM Signal Generation 190
11.5.4.2 Buffered PWM Signal Generation 191
11.5.4.3 PWM Initialization 192
11.6 Interrupts 193
11.7 Low-Power Modes 193
11.7.1 Wait Mode 194
11.7.2 Stop Mode 194
11.8 TIM During Break Interrupts 194
11.9 I/O Signals 195
11.10 I/O Registers 195
11.10.1 TIM Status and Control Register 196
11.10.2 TIM Counter Registers 198
11.10.3 TIM Counter Modulo Registers 199
11.10.4 TIM Channel Status and Control Registers 200
11.10.5 TIM Channel Registers 203

Section 12. Timebase Module (TBM)

12.1 Contents 205
12.2 Introduction 205

12.3 Features 205
12.4 Functional Description 206
12.5 Timebase Register Description. 207
12.6 Interrupts. 208
12.7 Low-Power Modes 209
12.7.1 Wait Mode 209
12.7.2 Stop Mode 209

Section 13. Pulse Width Modulator (PWM)

13.1 Contents 211
13.2 Introduction. 211
13.3 Features 212
13.4 PWM Period and Resolution. 214
13.5 PWM Automatic Phase Control 215
13.6 Low-Power Modes 216
13.7 Wait Mode. 216
13.8 Stop Mode 216
13.9 I/O Signals 217
13.10 I/O Registers. 217
13.10.1 PWM Control Register (PWMCR) 217
13.10.2 PWM Clock Control Register (PWMCCR) 218
13.10.3 PWM Data Registers (PWMDR0–PWMDR2) 219
13.10.4 PWM Phase Control Register 220

Section 14. Analog Module

14.1 Contents 221
14.2 Introduction. 221
14.3 Features 222
14.4 Functional Description 223

14.4.1 On-Chip Temperature Sensor 223
14.4.2 Two-Stage Amplifier 224
14.4.3 Amplifier Response Time 224
14.4.4 Current Flow Detection Amplifier 225
14.4.5 Current Flow Detect Output 225

14.5 Interrupts 225

14.6 Low-Power Modes 225
14.6.1 Wait Mode 225
14.6.2 Stop Mode 225

14.7 Analog Module I/O Registers 226
14.7.1 Analog Module Control Register (AMCR) 226
14.7.2 Analog Module Gain Control Register (AMGCR) 227
14.7.3 Analog Module Status and Control Register (AMSCR) . . . 228

Section 15. Analog-to-Digital Converter (ADC)

15.1 Contents 231

15.2 Introduction 232

15.3 Features 232

15.4 Functional Description 234
15.4.1 ADC Port I/O Pins 234
15.4.2 Voltage Conversion 235
15.4.3 Conversion Time 236
15.4.4 Continuous Conversion 237
15.4.5 Auto-scan Mode 237
15.4.6 Result Justification 238
15.4.7 Data Register Interlocking 239
15.4.8 Monotonicity 239

15.5 Interrupts 239

15.6 Low-Power Modes 239
15.6.1 Wait Mode 239
15.6.2 Stop Mode 240

15.7 I/O Signals 240
15.7.1 ADC Voltage In (V_{ADIN}) 240

15.7.2 ADC Analog Power Pin (V_{DDA}) 240
15.7.3 ADC Analog Ground Pin (V_{SSA}) 240
15.7.4 ADC Voltage Reference High Pin (V_{REFH}) 241
15.7.5 ADC Voltage Reference Low Pin (V_{REFL}) 241

15.8 I/O Registers 241
15.8.1 ADC Status and Control Register 242
15.8.2 ADC Clock Control Register 244
15.8.3 ADC Data Register 0 (ADRH0 and ADRL0) 246
15.8.4 ADC Auto-Scan Mode Data Registers (ADRL1–ADRL3) . . 248
15.8.5 ADC Auto-Scan Control Register (ADASCR) 248

Section 16. Serial Communications Interface (SCI)

16.1 Contents 251

16.2 Introduction 252

16.3 Features 252

16.4 Pin Name Conventions 254

16.5 Functional Description 254
16.5.1 Data Format 257
16.5.2 Transmitter 257
16.5.2.1 Character Length 259
16.5.2.2 Character Transmission 259
16.5.2.3 Break Characters 260
16.5.2.4 Idle Characters 260
16.5.2.5 Inversion of Transmitted Output 261
16.5.2.6 Transmitter Interrupts 261
16.5.3 Receiver 262
16.5.3.1 Character Length 262
16.5.3.2 Character Reception 262
16.5.3.3 Data Sampling 264
16.5.3.4 Framing Errors 266
16.5.3.5 Baud Rate Tolerance 266
16.5.3.6 Receiver Wakeup 269
16.5.3.7 Receiver Interrupts 270
16.5.3.8 Error Interrupts 270

16.6 Low-Power Modes 271
16.6.1 Wait Mode 271
16.6.2 Stop Mode 271

16.7 SCI During Break Module Interrupts 272

16.8 I/O Signals 272
16.8.1 TxD (Transmit Data) 272
16.8.2 RxD (Receive Data) 273

16.9 I/O Registers 273
16.9.1 SCI Control Register 1 273
16.9.2 SCI Control Register 2 276
16.9.3 SCI Control Register 3 279
16.9.4 SCI Status Register 1 282
16.9.5 SCI Status Register 2 286
16.9.6 SCI Data Register 287
16.9.7 SCI Baud Rate Register 288

Section 17. Multi-Master IIC Interface (MMIIC)

17.1 Contents 291

17.2 Introduction 292

17.3 Features 293

17.4 I/O Pins 293

17.5 Multi-Master IIC System Configuration 295

17.6 Multi-Master IIC Bus Protocol 295
17.6.1 START Signal 296
17.6.2 Slave Address Transmission 296
17.6.3 Data Transfer 296
17.6.4 Repeated START Signal 297
17.6.5 STOP Signal 297
17.6.6 Arbitration Procedure 297
17.6.7 Clock Synchronization 298
17.6.8 Handshaking 298
17.6.9 Packet Error Code 299

17.7 MMIIC I/O Registers 299

17.7.1 MMIIC Address Register (MMADR) 299
17.7.2 MMIIC Control Register 1 (MMCR1) 301
17.7.3 MMIIC Control Register 2 (MMCR2) 303
17.7.4 MMIIC Status Register (MMSR) 305
17.7.5 MMIIC Data Transmit Register (MMDTR) 307
17.7.6 MMIIC Data Receive Register (MMDRR) 308
17.7.7 MMIIC CRC Data Register (MMCRCDR) 309
17.7.8 MMIIC Frequency Divider Register (MMFDR) 310

17.8 Program Algorithm 311
17.8.1 Data Sequence 312

17.9 SMBus Protocols with PEC and without PEC 313
17.9.1 Quick Command 313
17.9.2 Send Byte 313
17.9.3 Receive Byte 313
17.9.4 Write Byte/Word 314
17.9.5 Read Byte/Word 314
17.9.6 Process Call 315
17.9.7 Block Read/Write 315

17.10 SMBus Protocol Implementation 316

Section 18. Input/Output (I/O) Ports

18.1 Contents 317

18.2 Introduction 317

18.3 Port A 320
18.3.1 Port A Data Register (PTA) 320
18.3.2 Data Direction Register A (DDRA) 321
18.3.3 Port A LED Control Register (LEDA) 323

18.4 Port B 323
18.4.1 Port B Data Register (PTB) 324
18.4.2 Data Direction Register B (DDRB) 325

18.5 Port C 327
18.5.1 Port C Data Register (PTC) 327
18.5.2 Data Direction Register C (DDRC) 329
18.5.3 Port C LED Control Register (LEDC) 330

18.6 Port D 331
18.6.1 Port D Data Register (PTD) 331
18.6.2 Data Direction Register D (DDRD) 332

Section 19. External Interrupt (IRQ)

19.1 Contents 335
19.2 Introduction 335
19.3 Features 335
19.4 Functional Description 336
19.5 $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$ Pins 338
19.6 IRQ Module During Break Interrupts 339
19.7 IRQ Registers 340
19.7.1 IRQ1 Status and Control Register 340
19.7.2 IRQ2 Status and Control Register 341

Section 20. Keyboard Interrupt Module (KBI)

20.1 Contents 343
20.2 Introduction 343
20.3 Features 344
20.4 I/O Pins 344
20.5 Functional Description 345
20.5.1 Keyboard Initialization 347
20.6 Keyboard Interrupt Registers 347
20.6.1 Keyboard Status and Control Register 348
20.6.2 Keyboard Interrupt Enable Register 349
20.7 Low-Power Modes 349
20.7.1 Wait Mode 349
20.7.2 Stop Mode 349
20.10 Keyboard Module During Break Interrupts 350

Section 21. Computer Operating Properly (COP)

21.1 Contents 351
21.2 Introduction 351
21.3 Functional Description 352
21.4 I/O Signals 353
21.4.1 ICLK 353
21.4.2 STOP Instruction 353
21.4.3 COPCTL Write 353
21.4.4 Power-On Reset 353
21.4.5 Internal Reset 354
21.4.6 Reset Vector Fetch 354
21.4.7 COPD (COP Disable) 354
21.4.8 COPRS (COP Rate Select) 354
21.5 COP Control Register 355
21.6 Interrupts 355
21.7 Monitor Mode 355
21.8 Low-Power Modes 355
21.8.1 Wait Mode 356
21.8.2 Stop Mode 356
21.9 COP Module During Break Mode 356

Section 22. Low-Voltage Inhibit (LVI)

22.1 Contents 357
22.2 Introduction 357
22.3 Features 357
22.4 Functional Description 358
22.4.1 Polled LVI Operation 359
22.4.2 Forced Reset Operation 360
22.4.3 Voltage Hysteresis Protection 360
22.4.4 LVI Trip Selection 360
22.5 LVI Status Register 361

22.6 LVI Interrupts 361
22.7 Low-Power Modes 361
22.7.1 Wait Mode 362
22.7.2 Stop Mode 362

Section 23. Break Module (BRK)

23.1 Contents 363
23.2 Introduction 363
23.3 Features 364
23.4 Functional Description 364
23.4.1 Flag Protection During Break Interrupts 366
23.4.2 CPU During Break Interrupts 366
23.4.3 TIM1 and TIM2 During Break Interrupts 366
23.4.4 COP During Break Interrupts 366
23.5 Low-Power Modes 366
23.5.1 Wait Mode 366
23.5.2 Stop Mode 367
23.6 Break Module Registers 367
23.6.1 Break Status and Control Register 367
23.6.2 Break Address Registers 368
23.6.3 SIM Break Status Register 368
23.6.4 SIM Break Flag Control Register 370

Section 24. Electrical Specifications

24.1 Contents 371
24.2 Introduction 372
24.3 Absolute Maximum Ratings 372
24.4 Functional Operating Range 373
24.5 Thermal Characteristics 373
24.6 5.0V DC Electrical Characteristics 374
24.7 3.0V DC Electrical Characteristics 376

24.8 5.0V Control Timing 377
24.9 3.0V Control Timing 377
24.10 5.0V Oscillator Characteristics 378
24.11 3.0V Oscillator Characteristics 379
24.12 5.0V ADC Electrical Characteristics 380
24.13 3.0V ADC Electrical Characteristics 381
24.14 Analog Module Electrical Characteristics 382
24.14.1 Temperature Sensor Electrical Characteristics 382
24.14.2 Current Detection Electrical Characteristics. 382
24.14.3 Two-Stage Amplifier Electrical Characteristics. 382
24.15 Timer Interface Module Characteristics 383
24.16 MMIIC Electrical Characteristics. 383
24.17 CGM Electrical Specifications. 385
24.18 FLASH Memory Characteristics 386

Section 25. Mechanical Specifications

25.1 Contents 387
25.2 Introduction. 387
25.3 48-Pin Plastic Low Quad Flat Pack (LQFP) 388
25.4 42-Pin Shrink Dual In-Line Package (SDIP). 389

Section 26. Ordering Information

26.1 Contents 391
26.2 Introduction. 391
26.3 MC Order Numbers 391

Appendix A. MC68HC08SR12

A.1 Contents 393
A.2 Introduction 394
A.3 MCU Block Diagram 394
A.4 Memory Map 394
A.5 Mask Option Register 397
A.6 Reserved Registers 397
A.7 Monitor ROM 397
A.8 Electrical Specifications 397
A.8.1 5.0V DC Electrical Characteristics 398
A.8.2 3.0V DC Electrical Characteristics 399
A.8.3 Memory Characteristics 401
A.9 ROM Order Numbers 401

Table of Contents

List of Figures

Figure	Title	Page
1-1	MC68HC908SR12 Block Diagram	39
1-2	48-Pin LQFP Pin Assignments	40
1-3	42-Pin SDIP Pin Assignment	41
1-4	Power Supply Bypassing	42
2-1	Memory Map	47
2-2	Control, Status, and Data Registers	48
4-1	FLASH I/O Register Summary	64
4-2	FLASH Control Register (FLCR)	65
4-3	FLASH Programming Flowchart	69
4-4	FLASH Block Protect Register (FLBPR)	70
4-5	FLASH Block Protect Start Address	70
5-1	CONFIG and MOR Register Summary	74
5-2	Configuration Register 1 (CONFIG1)	75
5-3	Configuration Register 2 (CONFIG2)	77
5-4	Mask Option Register (MOR)	79
6-1	CPU Registers	83
6-2	Accumulator (A)	83
6-3	Index Register (H:X)	84
6-4	Stack Pointer (SP)	85
6-5	Program Counter (PC)	85
6-6	Condition Code Register (CCR)	86
7-1	Oscillator Module Block Diagram	103
7-2	Mask Option Register (MOR)	104
7-3	Configuration Register 2 (CONFIG2)	105
7-4	Internal Oscillator	106

List of Figures

Figure	Title	Page
7-5	RC Oscillator	107
7-6	Crystal Oscillator	108
8-1	CGM Block Diagram	114
8-2	CGM I/O Register Summary	115
8-3	CGM External Connections	125
8-4	PLL Control Register (PCTL)	128
8-5	PLL Bandwidth Control Register (PBWCR)	131
8-6	PLL Multiplier Select Register High (PMSH)	132
8-7	PLL Multiplier Select Register Low (PMSL)	132
8-8	PLL VCO Range Select Register (PMRS)	133
8-9	PLL Reference Divider Select Register (PMDS)	134
8-10	PLL Filter	139
9-1	SIM Block Diagram	143
9-2	SIM I/O Register Summary	144
9-3	CGM Clock Signals	145
9-4	External Reset Timing	147
9-5	Internal Reset Timing	148
9-6	Sources of Internal Reset	148
9-7	POR Recovery	149
9-8	Interrupt Entry Timing	152
9-9	Interrupt Recovery Timing	152
9-10	Interrupt Processing	153
9-11	Interrupt Recognition Example	154
9-12	Interrupt Status Register 1 (INT1)	155
9-13	Interrupt Status Register 2 (INT2)	157
9-14	Interrupt Status Register 3 (INT3)	157
9-15	Wait Mode Entry Timing	159
9-16	Wait Recovery from Interrupt or Break	160
9-17	Wait Recovery from Internal Reset	160
9-18	Stop Mode Entry Timing	161
9-19	Stop Mode Recovery from Interrupt or Break	161
9-20	SIM Break Status Register (SBSR)	162
9-21	SIM Reset Status Register (SRSR)	163
9-22	SIM Break Flag Control Register (SBFCR)	164

Figure	Title	Page

10-1 Monitor Mode Circuit 167
10-2 Low-Voltage Monitor Mode Entry Flowchart 171
10-3 Monitor Data Format 172
10-4 Break Transaction 172
10-5 Read Transaction 174
10-6 Write Transaction 174
10-7 Stack Pointer at Monitor Mode Entry 178
10-8 Monitor Mode Entry Timing 179

11-1 TIM Block Diagram 184
11-2 TIM I/O Register Summary 185
11-3 PWM Period and Pulse Width 190
11-4 TIM Status and Control Register (TSC) 196
11-5 TIM Counter Registers High (TCNTH) 198
11-6 TIM Counter Registers Low (TCNTL) 198
11-7 TIM Counter Modulo Register High (TMODH) 199
11-8 TIM Counter Modulo Register Low (TMODL) 199
11-9 TIM Channel 0 Status and Control Register (TSC0) 200
11-10 TIM Channel 1 Status and Control Register (TSC1) 200
11-11 CHxMAX Latency 203
11-12 TIM Channel 0 Register High (TCH0H) 204
11-13 TIM Channel 0 Register Low (TCH0L) 204
11-14 TIM Channel 1 Register High (TCH1H) 204
11-15 TIM Channel 1 Register Low (TCH1L) 204

12-1 Timebase Block Diagram 206
12-2 Timebase Control Register (TBCR) 207

13-1 PWM I/O Register Summary 212
13-2 PWM Block Diagram 213
13-3 PWM Output Waveforms 214
13-4 PWM Automatic Phase Control 215
13-5 PWM Control Register (PWMCR) 217
13-6 PWM Clock Control Register (PWMCCR) 218
13-7 PWM Data Register 0 (PWMDR0) 219
13-8 PWM Data Register 1 (PWMDR1) 219

List of Figures

Figure	Title	Page
13-9	PWM Data Register 2 (PWMDR2)	219
13-10	PWM Phase Control Register (PWMPCR)	220
14-1	Analog Module Block Diagram	222
14-2	Analog Module I/O Register Summary	223
14-3	Analog Module Control Register (AMCR)	226
14-4	Analog Module Gain Control Register (AMGCR)	227
14-5	Analog Module Status and Control Register (AMSCR)	229
15-1	ADC I/O Register Summary	233
15-2	ADC Block Diagram	235
15-3	ADC Status and Control Register (ADSCR)	242
15-4	ADC Clock Control Register (ADICLK)	244
15-5	ADRH0 and ADRL0 in 8-Bit Truncated Mode	246
15-6	ADRH0 and ADRL0 in Right Justified Mode	246
15-7	ADRH0 and ADRL0 in Left Justified Mode	247
15-8	ADRH0 and ADRL0 in Left Justified Sign Data Mode	247
15-9	ADC Data Register Low 1 to 3 (ADRL1–ADRL3)	248
15-10	ADC Scan Control Register (ADASCR)	248
16-1	SCI Module Block Diagram	255
16-2	SCI I/O Register Summary	256
16-3	SCI Data Formats	257
16-4	SCI Transmitter	258
16-5	SCI Receiver Block Diagram	263
16-6	Receiver Data Sampling	264
16-7	Slow Data	267
16-8	Fast Data	268
16-9	SCI Control Register 1 (SCC1)	274
16-10	SCI Control Register 2 (SCC2)	277
16-11	SCI Control Register 3 (SCC3)	279
16-12	SCI Status Register 1 (SCS1)	282
16-13	Flag Clearing Sequence	285
16-14	SCI Status Register 2 (SCS2)	286
16-15	SCI Data Register (SCDR)	287
16-16	SCI Baud Rate Register (SCBR)	288

Figure	Title	Page

17-1 MMIIC I/O Register Summary 294
17-2 Multi-Master IIC Bus Transmission Signal Diagram 295
17-3 Clock Synchronization 298
17-4 MMIIC Address Register (MMADR) 299
17-5 MMIIC Control Register 1 (MMCR1) 301
17-6 MMIIC Control Register 2 (MMCR2) 303
17-7 MMIIC Status Register (MMSR) 305
17-8 MMIIC Data Transmit Register (MMDTR) 307
17-9 MMIIC Data Receive Register (MMDRR) 308
17-10 MMIIC CRC Data Register (MMCRCDR) 309
17-11 MMIIC Frequency Divider Register (MMFDR) 310
17-12 Data Transfer Sequences for Master/Slave Transmit/Receive Modes 312
17-13 Quick Command 313
17-14 Send Byte 313
17-15 Receive Byte 313
17-16 Write Byte/Word 314
17-17 Read Byte/Word 314
17-18 Process Call 315
17-19 Block Read/Write 315
17-20 SMBus Protocol Implementation 316

18-1 I/O Port Register Summary 318
18-2 Port A Data Register (PTA) 320
18-3 Data Direction Register A (DDRA) 321
18-4 Port A I/O Circuit 322
18-5 Port A LED Control Register (LEDA) 323
18-6 Port B Data Register (PTB) 324
18-7 Data Direction Register B (DDRB) 325
18-8 Port B I/O Circuit 326
18-9 Port C Data Register (PTC) 327
18-10 Data Direction Register B (DDRB) 329
18-11 Port C I/O Circuit 329
18-12 Port A LED Control Register (LEDA) 330
18-13 Port D Data Register (PTD) 331
18-14 Data Direction Register D (DDRD) 332
18-15 Port D I/O Circuit 332

List of Figures

Figure	Title	Page
19-1	External Interrupt I/O Register Summary	336
19-2	IRQ1 Block Diagram	337
19-3	IRQ2 Block Diagram	338
19-4	IRQ1 Status and Control Register (INTSCR1)	340
19-5	IRQ2 Status and Control Register (INTSCR2)	341
20-1	KBI I/O Register Summary	344
20-2	Keyboard Interrupt Block Diagram	345
20-3	Keyboard Status and Control Register (KBSCR)	348
20-4	Keyboard Interrupt Enable Register (KBIER)	349
21-1	COP Block Diagram	352
21-2	Configuration Register 1 (CONFIG1)	354
21-3	COP Control Register (COPCTL)	355
22-1	LVI I/O Register Summary	358
22-2	LVI Module Block Diagram	358
22-3	LVI Status Register	361
23-1	Break Module Block Diagram	365
23-2	Break Module I/O Register Summary	365
23-3	Break Status and Control Register (BRKSCR)	367
23-4	Break Address Register High (BRKH)	368
23-5	Break Address Register Low (BRKL)	368
23-6	SIM Break Status Register (SBSR)	369
23-7	SIM Break Flag Control Register (SBFCR)	370
24-1	RC vs. Bus Frequency (5V @25°C)	378
24-2	RC vs. Bus Frequency (3V @25°C)	379
24-3	MMIIC Signal Timings	383
25-1	48-Pin LQFP (Case #932-02)	388
25-2	42-Pin SDIP (Case #858-01)	389
A-1	MC68HC08SR12 Block Diagram	395
A-2	MC68HC08SR12 Memory Map	396

List of Tables

Table	Title	Page
2-1	Vector Addresses	60
5-1	CGMXCLK Clock Selection	80
6-1	Instruction Set Summary	90
6-2	Opcode Map	99
7-1	CGMXCLK Clock Selection	104
7-2	Timebase Module Reference Clock Selection	105
8-1	Numeric Examples	123
8-3	VPR1 and VPR0 Programming	130
8-2	PRE1 and PRE0 Programming	130
9-1	Signal Name Conventions	143
9-2	PIN Bit Set Timing	147
9-3	Interrupt Sources	156
10-1	Monitor Mode Signal Requirements and Options	169
10-2	Mode Differences	172
10-3	Monitor Baud Rate Selection	173
10-4	READ (Read Memory) Command	175
10-5	WRITE (Write Memory) Command	175
10-7	IWRITE (Indexed Write) Command	176
10-6	IREAD (Indexed Read) Command	176
10-8	READSP (Read Stack Pointer) Command	177
10-9	RUN (Run User Program) Command	177
11-1	Pin Name Conventions	183
11-2	Prescaler Selection	197

List of Tables

Table	Title	Page
11-3	Mode, Edge, and Level Selection	202
12-1	Timebase Rate Selection for OSCCLK = 32.768 kHz	207
13-1	PTC0 Pin Configuration	218
13-2	PWM Counter Clock Prescaler Selection	219
14-1	Analog Module Power Control	226
14-2	Amplifier Channel Select Control bits	227
14-3	Analog Module Gain Values	228
14-4	Analog Module Clock Divider Select	229
15-1	MUX Channel Select	243
15-2	ADC Clock Divide Ratio	244
15-3	ADC Mode Select	245
15-4	Auto-scan Mode Channel Select	248
16-1	Pin Name Conventions	254
16-2	Start Bit Verification	265
16-3	Data Bit Recovery	265
16-4	Stop Bit Recovery	266
16-5	Character Format Selection	276
16-6	SCI Baud Rate Prescaling	288
16-7	SCI Baud Rate Selection	289
16-8	SCI Baud Rate Selection Examples	290
17-1	Pin Name Conventions	294
17-2	MMIIC Baud Rate Selection	310
18-1	Port Control Register Bits Summary	319
18-2	Port A Pin Functions	322
18-3	PTB2 and PTB3 Pin Configurations	325
18-4	Port B Pin Functions	326
18-5	PTC0 Pin Configuration	328
18-6	Port C Pin Functions	330
18-7	Port D Pin Functions	333

Table	Title	Page
20-1	Pin Name Conventions	344
22-1	LVIOUT Bit Indication	361
24-1	Absolute Maximum Ratings	372
24-2	Operating Range	373
24-3	Thermal Characteristics	373
24-4	5V DC Electrical Characteristics	374
24-5	3V DC Electrical Characteristics	376
24-6	5V Control Timing	377
24-7	3V Control Timing	377
24-8	5V Oscillator Specifications	378
24-9	3V Oscillator Specifications	379
24-10	5V ADC Electrical Characteristics	380
24-11	3V ADC Electrical Characteristics	381
24-12	Temperature Sensor Electrical Characteristics	382
24-13	Current Detection Electrical Characteristics	382
24-14	Two-Stage Amplifier Electrical Characteristics	382
24-15	MMIIC DC Electrical Characteristics	383
24-16	MMIIC Interface Input/Output Signal Timing	384
24-17	FLASH Memory Electrical Characteristics	386
26-1	MC Order Numbers	391
A-1	Summary of MC68HC08SR12 and MC68HC908SR12 Differences	394
A-2	5V DC Electrical Characteristics	398
A-3	3V DC Electrical Characteristics	399
A-4	MC68HC08SR12 Order Numbers	401

List of Tables

Section 1. General Description

1.1 Contents

1.2 Introduction 36

1.3 Features 36

1.4 MCU Block Diagram 38

1.5 Pin Assignments 40

1.6 Pin Functions 41
1.6.1 Power Supply Pins (V_{DD} and V_{SS}) 42
1.6.2 Oscillator Pins (OSC1 and OSC2) 42
1.6.3 External Reset Pin ($\overline{RST}$) 43
1.6.4 External Interrupt Pin ($\overline{IRQ1}$) 43
1.6.5 Analog Power Supply Pin (V_{DDA}) 43
1.6.6 Analog Ground Pin (V_{SSA}) 43
1.6.7 ADC Voltage Low Reference Pin (V_{REFL}) 43
1.6.8 ADC Voltage High Reference Pin (V_{REFH}) 43
1.6.9 External Filter Capacitor Pin (CGMXFC) 43
1.6.10 Analog Input Pins (OPIN1/ATD0, OPIN2/ATD1, V_{SSAM}) . . . 44
1.6.11 Port A Input/Output (I/O) Pins (PTA7–PTA0) 44
1.6.12 Port B I/O Pins (PTB6–PTB0) 44
1.6.13 Port C I/O Pins (PTC7–PTC0) 44
1.6.14 Port D I/O Pins (PTD7/KBI7–PTD0/KBI0) 44

1.2 Introduction

The MC68HC908SR12 is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). The M68HC08 Family is based on the customer-specified integrated circuit (CSIC) design strategy. All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

1.3 Features

Features of the MC68HC908SR12 include the following:

- High-performance M68HC08 architecture
- Fully upward-compatible object code with M6805, M146805, and M68HC05 Families
- Maximum internal bus frequency:
  - 8-MHz at 5V operating voltage
  - 4-MHz at 3V operating voltage
- Clock input options:
  - RC-oscillator
  - 32kHz crystal-oscillator with 32MHz internal phase-lock-loop
- 12k-bytes user program FLASH memory with security[1] feature
- 512 bytes of on-chip RAM
- Two 16-bit, 2-channel timer interface modules (TIM1 and TIM2) with selectable input capture, output compare, and PWM capability on each channel
- Timebase module
- 3-channel, 8-bit high speed PWM (125kHz) with independent counters and automatic phase control
- Serial communications interface module (SCI)

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.

- System Management Bus (SMBus), version 1.0/1.1 (Multi-master IIC bus)
- 14-channel, 10-bit analog-to-digital converter (ADC), with auto-scan mode for 4 channels
- Current sensor with programmable amplifier
- Temperature sensor (–20°C to +70°C)
- $\overline{\text{IRQ1}}$ external interrupt pin with integrated pullup
- $\overline{\text{IRQ2}}$ external interrupt pin with programmable pullup
- 8-bit keyboard wakeup port with integrated pullup
- 31 general-purpose input/output (I/O) pins and 2 dedicated pins:
  - 31 shared-function I/O pins
  - Two dedicated analog input pins
- Low-power design (fully static with Stop and Wait modes)
- Master reset pin (with integrated pullup) and power-on reset
- System protection features
  - Optional computer operating properly (COP) reset
  - Low-voltage detection with optional reset
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- 48-pin low quad flat pack (LQFP) and 42-pin shrink dual-in-line package (SDIP)
- Specific features of the MC68HC908SR12 in 42-pin SDIP are:
  - 29 general-purpose I/Os only
  - 11-channel ADC only

Features of the CPU08 include the following:

- Enhanced HC05 programming model
- Extensive loop control functions
- 16 addressing modes (eight more than the HC05)

- 16-bit Index register and stack pointer
- Memory-to-memory data transfers
- Fast 8 × 8 multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Efficient C language support

1.4 MCU Block Diagram

Figure 1-1 shows the structure of the MC68HC908SR12.

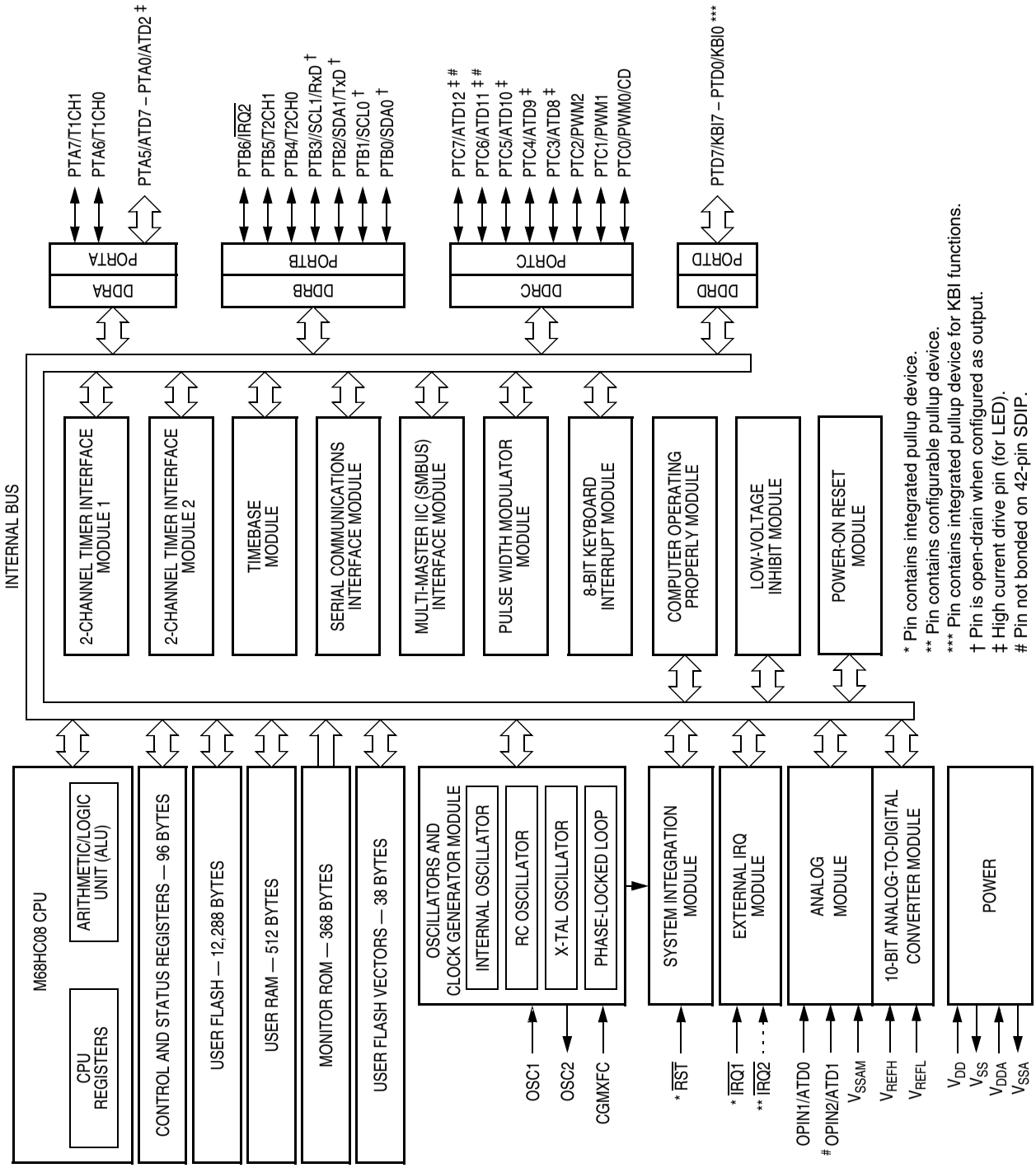
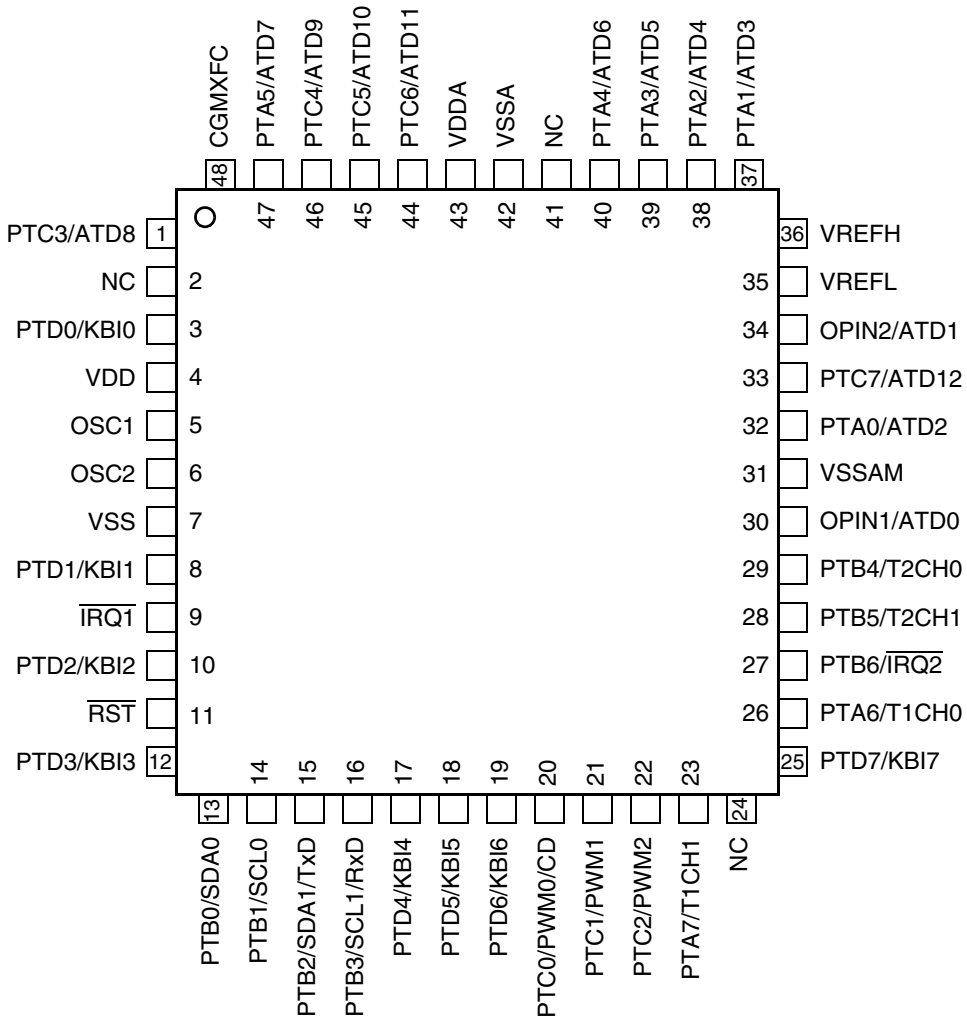


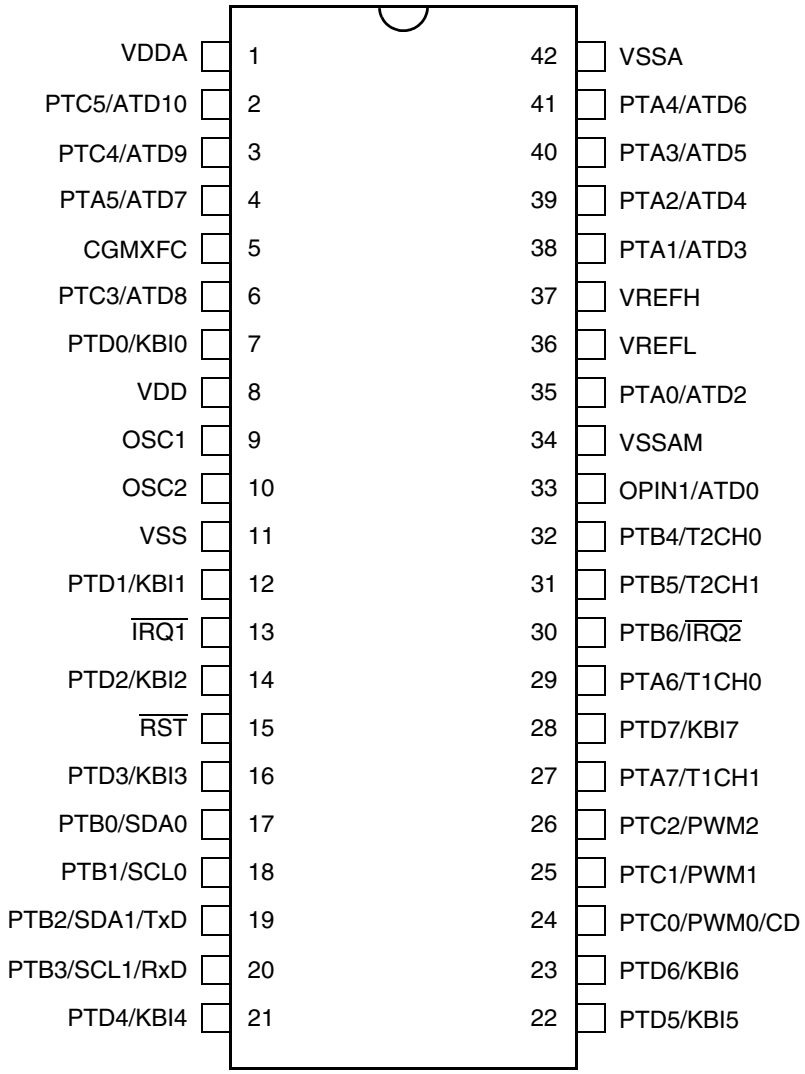
Figure 1-1. MC68HC908SR12 Block Diagram

1.5 Pin Assignments



NC: No connection

Figure 1-2. 48-Pin LQFP Pin Assignments



Pins not available on 42-pin package	Internal connection
OPIN2/ATD1	Unconnected
PTC6/ATD11	Unconnected
PTC7/ATD12	Unconnected

Figure 1-3. 42-Pin SDIP Pin Assignment

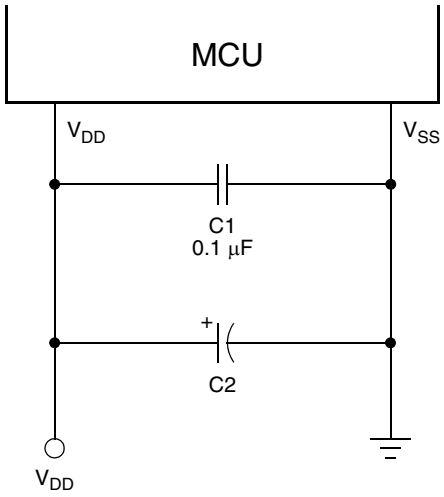
1.6 Pin Functions

Description of pin functions are provided here.

1.6.1 Power Supply Pins (V_{DD} and V_{SS})

V_{DD} and V_{SS} are the power supply and ground pins. The MCU operates from a single power supply.

Fast signal transitions on MCU pins place high, short-duration current demands on the power supply. To prevent noise problems, take special care to provide power supply bypassing at the MCU as **Figure 1-4** shows. Place the C1 bypass capacitor as close to the MCU as possible. Use a high-frequency-response ceramic capacitor for C1. C2 is an optional bulk current bypass capacitor for use in applications that require the port pins to source high current levels.



NOTE: Component values shown represent typical applications.

Figure 1-4. Power Supply Bypassing

V_{SS} must be grounded for proper MCU operation.

1.6.2 Oscillator Pins (OSC1 and OSC2)

The OSC1 and OSC2 pins are the connections for the on-chip oscillator circuit. See **Section 7. Oscillator (OSC)** and **Section 8. Clock Generator Module (CGM)**.

1.6.3 External Reset Pin ($\overline{\text{RST}}$)

A logic 0 on the $\overline{\text{RST}}$ pin forces the MCU to a known start-up state. $\overline{\text{RST}}$ is bidirectional, allowing a reset of the entire system. It is driven low when any internal reset source is asserted. This pin contains an internal pullup resistor. See **Section 9. System Integration Module (SIM)**.

1.6.4 External Interrupt Pin ($\overline{\text{IRQ1}}$)

$\overline{\text{IRQ1}}$ is an asynchronous external interrupt pin. This pin contains an internal pullup resistor. See **Section 19. External Interrupt (IRQ)**.

1.6.5 Analog Power Supply Pin (V_{DDA})

V_{DDA} is the power supply pin for the analog circuits of the MCU.

1.6.6 Analog Ground Pin (V_{SSA})

V_{SSA} is the power supply ground pin for the analog circuits of the MCU. It should be decoupled as per the V_{SS} digital ground pin.

1.6.7 ADC Voltage Low Reference Pin (V_{REFL})

V_{REFL} is the voltage input pin for the ADC voltage low reference. See **Section 15. Analog-to-Digital Converter (ADC)**.

1.6.8 ADC Voltage High Reference Pin (V_{REFH})

V_{REFH} is the voltage input pin for the ADC voltage high reference. See **Section 15. Analog-to-Digital Converter (ADC)**.

1.6.9 External Filter Capacitor Pin (CGMXFC)

CGMXFC is an external filter capacitor connection for the CGM. See **Section 8. Clock Generator Module (CGM)**.

1.6.10 Analog Input Pins (OPIN1/ATD0, OPIN2/ATD1, V_{SSAM})

OPIN1/ATD0 and OPIN2/ATD1 are input pins to the analog module and ADC and V_{SSAM} is the negative reference input. See **Section 14. Analog Module** and **Section 15. Analog-to-Digital Converter (ADC)**.

1.6.11 Port A Input/Output (I/O) Pins (PTA7–PTA0)

PTA7–PTA0 are special function, bidirectional port pins. PTA7/T1CH1–PTA6/T1CH0 are shared with the TIM1, and PTA5/ATD7–PTA0/ATD2 are shared with the ADC. See **Section 18. Input/Output (I/O) Ports**, **Section 11. Timer Interface Module (TIM)**, and **Section 15. Analog-to-Digital Converter (ADC)**.

1.6.12 Port B I/O Pins (PTB6–PTB0)

PTB6–PTB0 are special function, bidirectional port pins. PTB6/$\overline{\text{IRQ2}}$ is shared with the IRQ2 input, PTB5/T2CH1–PTB4/T2CH0 are shared with the TIM2, PTB3/SCL1/RxD–PTB2/SDA1/TxD are shared with the MMIIC and SCI, and PTB1/SCL0–PTB0/SDA0 are shared with the MMIIC. See **Section 18. Input/Output (I/O) Ports**, **Section 19. External Interrupt (IRQ)**, **Section 11. Timer Interface Module (TIM)**, **Section 16. Serial Communications Interface (SCI)**, and **Section 17. Multi-Master IIC Interface (MMIIC)**.

1.6.13 Port C I/O Pins (PTC7–PTC0)

PTC7–PTC0 are special function, bidirectional port pins. PTC7/ATD12–PTC3/ATD8 are shared with the ADC, PTC2/PWM2–PTC1/PWM1 are shared with the PWM, and PTC0/PWM0/CD is shared with the PWM and analog module. See **Section 18. Input/Output (I/O) Ports**, **Section 15. Analog-to-Digital Converter (ADC)**, **Section 13. Pulse Width Modulator (PWM)**, and **Section 14. Analog Module**.

1.6.14 Port D I/O Pins (PTD7/KBI7–PTD0/KBI0)

PTD7–PTD0 are general-purpose bidirectional port pins with keyboard wakeup function. See **Section 18. Input/Output (I/O) Ports** and **Section 20. Keyboard Interrupt Module (KBI)**.

Section 2. Memory Map

2.1 Contents

2.2 Introduction 45
2.3 Unimplemented Memory Locations 45
2.4 Reserved Memory Locations 46
2.5 Input/Output (I/O) Section 46

2.2 Introduction

The CPU08 can address 64k-bytes of memory space. The memory map, shown in **Figure 2-1**, includes:

- 12,288 bytes of user FLASH memory
- 512 bytes of random-access memory (RAM)
- 38 bytes of user-defined vectors
- 368 bytes of monitor ROM

2.3 Unimplemented Memory Locations

Accessing an unimplemented location can cause an illegal address reset. In the memory map (**Figure 2-1**) and in register figures in this document, unimplemented locations are shaded.

2.4 Reserved Memory Locations

Accessing a reserved location can have unpredictable effects on MCU operation. In the **Figure 2-1** and in register figures in this document, reserved locations are marked with the word Reserved or with the letter R.

2.5 Input/Output (I/O) Section

Most of the control, status, and data registers are in the zero page $0000–$005F. Additional I/O registers have the following addresses:

- $FE00; SIM break status register, SBSR
- $FE01; SIM reset status register, SRSR
- $FE03; SIM break flag control register, SBFCR
- $FE04; Interrupt status register 1, INT1
- $FE05; Interrupt status register 2, INT2
- $FE06; Interrupt status register 3, INT3
- $FE07; Reserved
- $FE08; FLASH control register, FLCR
- $FE09; FLASH block protect register, FLBPR
- $FE0A; Reserved
- $FE0B; Reserved
- $FE0C; break address register high, BRKH
- $FE0D; break address register low, BRKL
- $FE0E; break status and control register, BRKSCR
- $FE0F; LVI status register, LVISR
- $FF80; Mask option register, MOR
- $FFFF; COP control register, COPCTL

Data registers are shown in **Figure 2-2**, **Table 2-1** is a list of vector locations.

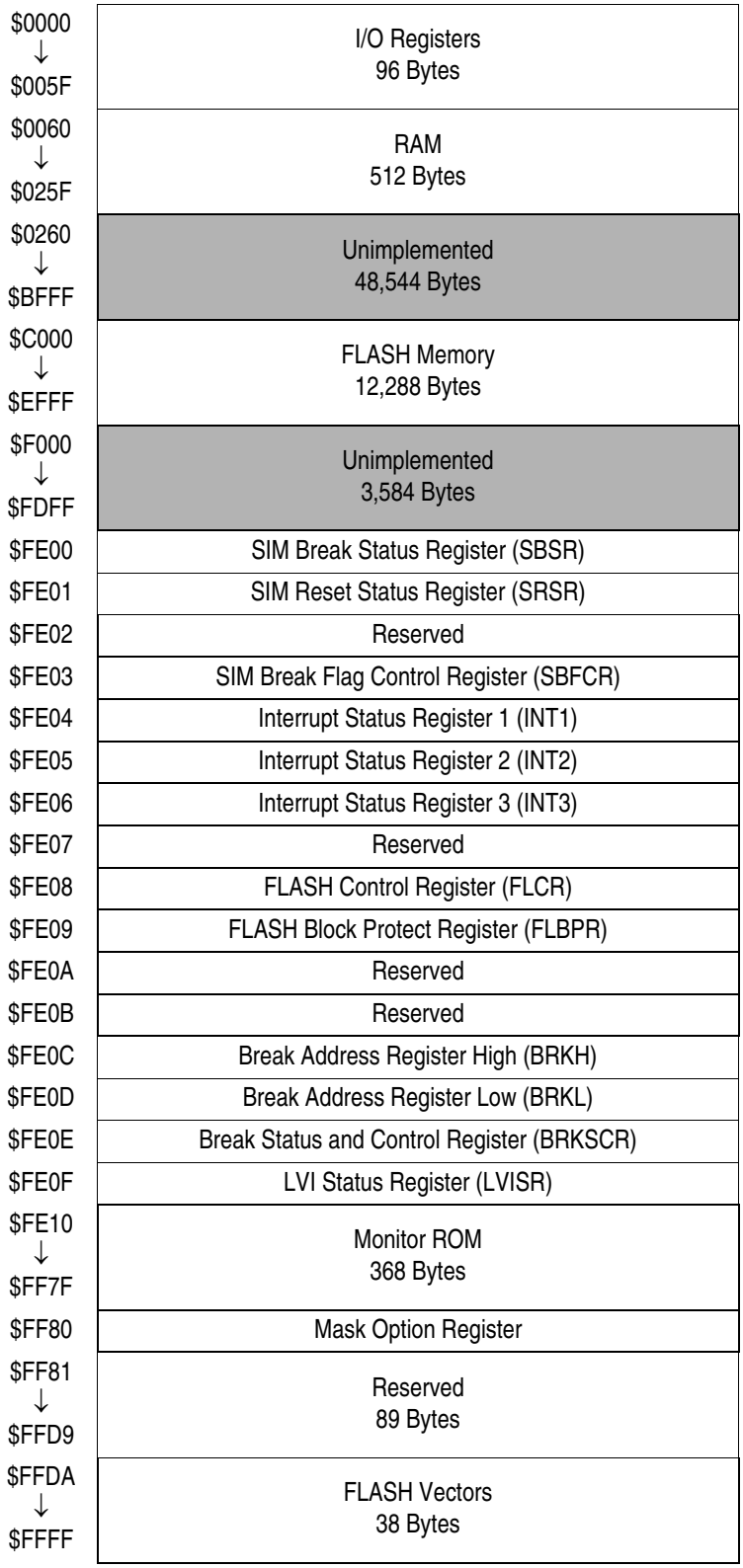


Figure 2-1. Memory Map

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA)	Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Reset:	U	U	U	U	U	U	U	U
$0001	Port B Data Register (PTB)	Read:	0	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:	0	U	U	U	U	U	U	U
$0002	Port C Data Register (PTC)	Read: Write:	PTC7	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
		Reset:	U	U	U	U	U	U	U	U
$0003	Port D Data Register (PTD)	Read: Write:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Reset:	U	U	U	U	U	U	U	U
$0004	Data Direction Register A (DDRA)	Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB)	Read:	0	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0006	Data Direction Register C (DDRC)	Read: Write:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Reset:	0	0	0	0	0	0	0	0
$0007	Data Direction Register D (DDRD)	Read: Write:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Reset:	0	0	0	0	0	0	0	0
$0008	Unimplemented	Read: Write:								
		Reset:								
$0009	Unimplemented	Read: Write:								
		Reset:								

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 1 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000A	Unimplemented	Read:								
		Write:								
		Reset:								
$000B	Unimplemented	Read:								
		Write:								
		Reset:								
$000C	Port-A LED Control Register (LEDA)	Read:	0	0	LEDA5	LEDA4	LEDA3	LEDA2	LEDA1	LEDA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000D	Port-C LED Control Register (LEDC)	Read:	LEDC7	LEDC6	LEDC5	LEDC4	LEDC3	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000E	Analog Module Control Register (AMCR)	Read:	PWR1	PWR0	OPCH1	OPCH0	AMIEN	DO2	DO1	DO0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000F	Analog Module Gain Control Register (AMGCR)	Read:	GAINB3	GAINB2	GAINB1	GAINB0	GAINA3	GAINA2	GAINA1	GAINA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0010	Analog Module Status and Control Register (AMSCR)	Read:	AMCDIV1	AMCDIV0	0	OPIF	0	DOF	0	CDIF
		Write:			OPIFR				CDIFR	
		Reset:	0	0	U	0	0	0	U	0
$0011	Unimplemented	Read:								
		Write:								
		Reset:								
$0012	Unimplemented	Read:								
		Write:								
		Reset:								
$0013	SCI Control Register 1 (SCC1)	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Write:								
		Reset:	0	0	0	0	0	0	0	0

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 2 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0014	SCI Control Register 2 (SCC2)	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0015	SCI Control Register 3 (SCC3)	Read:	R8	T8	DMARE	DMATE	ORIE	NEIE	FEIE	PEIE
		Write:								
		Reset:	U	U	0	0	0	0	0	0
$0016	SCI Status Register 1 (SCS1)	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:								
		Reset:	1	1	0	0	0	0	0	0
$0017	SCI Status Register 2 (SCS2)	Read:	0	0	0	0	0	0	BKF	RPF
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0018	SCI Data Register (SCDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	U	U	U	U	U	U	U	U
$0019	SCI Baud Rate Register (SCBR)	Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:								
		Reset:	0	0	0	0		0	0	0
$001A	Keyboard Status and Control Register (KBSCR)	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$001B	Keyboard Interrupt Enable Register (KBIER)	Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001C	IRQ2 Status and Control Register (INTSCR2)	Read:	0	PTBPUE6	0	0	IRQ2F	0	IMASK2	MODE2
		Write:						ACK2		
		Reset:	0	0	0	0	0	0	0	0
$001D	Configuration Register 2 (CONFIG2)†	Read:	STOP_ ICLKEN	STOP_ RCLKEN	STOP_ XCLKEN	OSCCLK1	OSCCLK0	0	CDOEN	SCIBDSRC
		Write:								
		Reset:	0	0	0	0	0	0	0	0

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 3 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001E	IRQ1 Status and Control Register (INTSCR1)	Read:	0	0	0	0	IRQ1F	0	IMASK1	MODE1
		Write:						ACK1		
		Reset:	0	0	0	0	0	0	0	0
$001F	Configuration Register 1 (CONFIG1)†	Read:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3	SSREC	STOP	COPD
		Write:								
		Reset:	0	0	0	0	0	0	0	0

† One-time writable register after each reset.

$0020	Timer 1 Status and Control Register (T1SC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0021	Timer 1 Counter Register High (T1CNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0022	Timer 1 Counter Register Low (T1CNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0023	Timer 1 Counter Modulo Register High (T1MODH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0024	Timer 1 Counter Modulo Register Low (T1MODL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0025	Timer 1 Channel 0 Status and Control Register (T1SC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0026	Timer 1 Channel 0 Register High (T1CH0H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	X	X	X	X	X	X	X	X

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0027	Timer 1 Channel 0 Register Low (T1CH0L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	X	X	X	X	X	X	X	X
$0028	Timer 1 Channel 1 Status and Control Register (T1SC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0029	Timer 1 Channel 1 Register High (T1CH1H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	X	X	X	X	X	X	X	X
$002A	Timer 1 Channel 1 Register Low (T1CH1L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	X	X	X	X	X	X	X	X
$002B	Timer 2 Status and Control Register (T2SC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$002C	Timer 2 Counter Register High (T2CNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$002D	Timer 2 Counter Register Low (T2CNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$002E	Timer 2 Counter Modulo Register High (T2MODH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$002F	Timer 2 Counter Modulo Register Low (T2MODL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0030	Timer 2 Channel 0 Status and Control Register (T2SC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 5 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0031	Timer 2 Channel 0 Register High (T2CH0H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	X	X	X	X	X	X	X	X
$0032	Timer 2 Channel 0 Register Low (T2CH0L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	X	X	X	X	X	X	X	X
$0033	Timer 2 Channel 1 Status and Control Register (T2SC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0034	Timer 2 Channel 1 Register High (T2CH1H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	X	X	X	X	X	X	X	X
$0035	Timer 2 Channel 1 Register Low (T2CH1L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	X	X	X	X	X	X	X	X
$0036	PLL Control Register (PTCL)	Read:	PLLIE	PLLF	PLLON	BCS	PRE1	PRE0	VPR1	VPR0
		Write:								
		Reset:	0	0	1	0	0	0	0	0
$0037	PLL Bandwidth Control Register (PBWC)	Read:	AUTO	LOCK	$\overline{\text{ACQ}}$	0	0	0	0	R
		Write:								
		Reset:	0	0	0	0	0	0	0	
$0038	PLL Multiplier Select Register High (PMSH)	Read:	0	0	0	0	MUL11	MUL10	MUL9	MUL8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0039	PLL Multiplier Select Register Low (PMSL)	Read: Write:	MUL7	MUL6	MUL5	MUL4	MUL3	MUL2	MUL1	MUL0
		Reset:	0	1	0	0	0	0	0	0
$003A	PLL VCO Range Select Register (PMRS)	Read: Write:	VRS7	VRS6	VRS5	VRS4	VRS3	VRS2	VRS1	VRS0
		Reset:	0	1	0	0	0	0	0	0

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 6 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$003B	PLL Reference Divider Select Register (PMDS)	Read:	0	0	0	0	RDS3	RDS2	RDS1	RDS0
		Write:								
		Reset:	0	0	0	0	0	0	0	1
$003C	Unimplemented	Read:								
		Write:								
		Reset:								
$003D	Unimplemented	Read:								
		Write:								
		Reset:								
$003E	Unimplemented	Read:								
		Write:								
		Reset:								
$003F	Unimplemented	Read:								
		Write:								
		Reset:								
$0040	Unimplemented	Read:								
		Write:								
		Reset:								
$0041	Unimplemented	Read:								
		Write:								
		Reset:								
$0042	Unimplemented	Read:								
		Write:								
		Reset:								
$0043	Unimplemented	Read:								
		Write:								
		Reset:								
$0044	Unimplemented	Read:								
		Write:								
		Reset:								

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 7 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0045	Unimplemented	Read:								
		Write:								
		Reset:								
$0046	Timebase Control Register (TBCR)	Read:	TBIF	TBR2	TBR1	TBR0	0	TBIE	TBON	R
		Write:					TACK			
		Reset:	0	0	0	0	0	0	0	
$0047	Unimplemented	Read:								
		Write:								
		Reset:								
$0048	MMIIC Address Register (MMADR)	Read:	MMAD7	MMAD6	MMAD5	MMAD4	MMAD3	MMAD2	MMAD1	MMEXTAD
		Write:								
		Reset:	1	0	1	0	0	0	0	0
$0049	MMIIC Control Register 1 (MMCR1)	Read:	MMEN	MMIEN	0	0	MMTXAK	REPSEN	MMCRCBYTE	SDASCL1
		Write:			MMCLRBB					
		Reset:	0	0	0	0	0	0	0	0
$004A	MMIIC Control Register 2 (MMCR2)	Read:	MMALIF	MMNAKIF	MMBB	MMAST	MMRW	0	0	MMCRCEF
		Write:	0	0						
		Reset:	0	0	0	0	0	0	0	Unaffected
$004B	MMIIC Status Register (MMSR)	Read:	MMRXIF	MMTXIF	MMATCH	MMSRW	MMRXAK	MMCRCBF	MMTXBE	MMRXBF
		Write:	0	0						
		Reset:	0	0	0	0	1	0	1	0
$004C	MMIIC Data Transmit Register (MMDTR)	Read:	MMTD7	MMTD6	MMTD5	MMTD4	MMTD3	MMTD2	MMTD1	MMTD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$004D	MMIIC Data Receive Register (MDDRR)	Read:	MMRD7	MMRD6	MMRD5	MMRD4	MMRD3	MMRD2	MMRD1	MMRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$004E	MMIIC CRC Data Register (MMCRDR)	Read:	MMCRCD7	MMCRCD6	MMCRCD5	MMCRCD4	MMCRCD3	MMCRCD2	MMCRCD1	MMCRCD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 8 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$004F	MMIIC Frequency Divider Register (MMFDR)	Read:	0	0	0	0	0	MMBR2	MMBR1	MMBR0
		Write:								
		Reset:	0	0	0	0	0	1	0	0
$0050	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$0051	PWM Control Register (PWMCR)	Read:	PWMEN2	PWMEN1	PWMEN0	0	0	PCH2	PCH1	PCH0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0052	PWM Clock Control Register (PWMCCR)	Read:	PCLKSEL	0	0	0	0	0	PCLK1	PCLK0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0053	PWM Data Register 0 (PWMDR0)	Read:	0PWMD7	0PWMD6	0PWMD5	0PWMD4	0PWMD3	0PWMD2	0PWMD1	0PWMD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0054	PWM Data Register 1 (PWMDR1)	Read:	1PWMD7	1PWMD6	1PWMD5	1PWMD4	1PWMD3	1PWMD2	1PWMD1	1PWMD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0055	PWM Data Register 2 (PWMDR2)	Read:	2PWMD7	2PWMD6	2PWMD5	2PWMD4	2PWMD3	2PWMD2	2PWMD1	2PWMD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0056	PWM Phase Control Register (PWMPCR)	Read:	PHEN	PHD6	PHD5	PHD4	PHD3	PHD2	PHD1	PHD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0057	ADC Status and Control Register (ADSCR)	Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
		Write:								
		Reset:	0	0	0	1	1	1	1	1
$0058	ADC Clock Control Register (ADICLK)	Read:	ADIV2	ADIV1	ADIV0	ADICLK	MODE1	MODE0	0	0
		Write:								R
		Reset:	0	0	0	0	0	1	0	0

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 9 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0059	ADC Data Register High 0 (ADRH0)	Read:	ADx	ADx	ADx	ADx	ADx	ADx	ADx	ADx
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005A	ADC Data Register Low 0 (ADRL0)	Read:	ADx	ADx	ADx	ADx	ADx	ADx	ADx	ADx
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005B	ADC Data Register Low 1 (ADRL1)	Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005C	ADC Data Register Low 2 (ADRL3)	Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005D	ADC Data Register Low 3 (ADRL3)	Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005E	ADC Auto-scan Control Register (ADASCR)	Read:	0	0	0	0	0	AUTO1	AUTO0	ASCAN
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$005F	Unimplemented	Read:								
		Write:								
		Reset:								
$FE00	SIM Break Status Register (SBSR)	Read:	R	R	R	R	R	R	SBSW	R
		Write:							Note	
		Reset:							0	
$FE01	SIM Reset Status Register (SRSR)	Read:	POR	PIN	COP	ILOP	ILAD	0	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0

Note: Writing a logic 0 clears SBSW.

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 10 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE02	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$FE03	SIM Break Flag Control Register (SBFCR)	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE04	Interrupt Status Register 1 (INT1)	Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE05	Interrupt Status Register 2 (INT2)	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE06	Interrupt Status Register 3 (INT3)	Read:	0	0	0	0	0	IF17	IF16	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE07	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$FE08	FLASH Control Register (FLCR)	Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE09	FLASH Block Protect Register (FLBPR)	Read:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0A	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$FE0B	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 11 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE0C	Break Address Register High (BRKH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0D	Break Address Register Low (BRKL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0E	Break Status and Control Register (BRKSCR)	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0F	Low-Voltage Inhibit Status Register (LVISR)	Read:	LVIOUT	0	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FF80	Mask Option Register (MOR)*	Read:	OSCSEL1	OSCSEL0	R	R	R	R	R	R
		Write:								
		Erased:	1	1	1	1	1	1	1	1
		Reset:	U	U	U	U	U	U	U	U

* MOR is a non-volatile FLASH register; write by programming.

$FFFF	COP Control Register (COPCTL)	Read:	Low byte of reset vector							
		Write:	Writing clears COP counter (any value)							
		Reset:	U	U	U	U	U	U	U	U

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 12 of 12)

Table 2-1. Vector Addresses

Vector Priority	Vector	Address	Vector
Lowest	IF17	$FFDA	Timebase Module Interrupt Vector (High)
		$FFDB	Timebase Module Interrupt Vector (Low)
	IF16	$FFDC	Analog Module Interrupt Vector (High)
		$FFDD	Analog Module Interrupt Vector (Low)
	IF15	$FFDE	ADC Conversion Complete Vector (High)
		$FFDF	ADC Conversion Complete Vector (Low)
	IF14	$FFE0	Keyboard Vector (High)
		$FFE1	Keyboard Vector (Low)
	IF13	$FFE2	SCI Transmit Vector (High)
		$FFE3	SCI Transmit Vector (Low)
	IF12	$FFE4	SCI Receive Vector (High)
		$FFE5	SCI Receive Vector (Low)
	IF11	$FFE6	SCI Error Vector (High)
		$FFE7	SCI Error Vector (Low)
	IF10	$FFE8	MMIIC Interrupt Vector (High)
		$FFE9	MMIIC Interrupt Vector (Low)
	IF9	$FFEA	TIM2 Overflow Vector (High)
		$FFEB	TIM2 Overflow Vector (Low)
	IF8	$FFEC	TIM2 Channel 1 Vector (High)
		$FFED	TIM2 Channel 1 Vector (Low)
	IF7	$FFEE	TIM2 Channel 0 Vector (High)
		$FFEF	TIM2 Channel 0 Vector (Low)
	IF6	$FFF0	TIM1 Overflow Vector (High)
		$FFF1	TIM1 Overflow Vector (Low)
	IF5	$FFF2	TIM1 Channel 1 Vector (High)
		$FFF3	TIM1 Channel 1 Vector (Low)
	IF4	$FFF4	TIM1 Channel 0 Vector (High)
		$FFF5	TIM1 Channel 0 Vector (Low)
	IF3	$FFF6	PLL Vector (High)
		$FFF7	PLL Vector (Low)
	IF2	$FFF8	$\overline{\text{IRQ2}}$ Vector (High)
		$FFF9	$\overline{\text{IRQ2}}$ Vector (Low)
	IF1	$FFFA	$\overline{\text{IRQ1}}$ Vector (High)
		$FFFB	$\overline{\text{IRQ1}}$ Vector (Low)
	—	$FFFC	SWI Vector (High)
		$FFFD	SWI Vector (Low)
	—	$FFFE	Reset Vector (High)
Highest		$FFFF	Reset Vector (Low)

Section 3. Random-Access Memory (RAM)

3.1 Contents

3.2 Introduction 61
3.3 Functional Description 61

3.2 Introduction

This section describes the 512 bytes of RAM (random-access memory).

3.3 Functional Description

Addresses $0060 through $025F are RAM locations. The location of the stack RAM is programmable. The 16-bit stack pointer allows the stack to be anywhere in the 64K-byte memory space.

NOTE: *For correct operation, the stack pointer must point only to RAM locations.*

Within page zero are 160 bytes of RAM. Because the location of the stack RAM is programmable, all page zero RAM locations can be used for I/O control and user data or code. When the stack pointer is moved from its reset location at $00FF out of page zero, direct addressing mode instructions can efficiently access all page zero RAM locations. Page zero RAM, therefore, provides ideal locations for frequently accessed global variables.

Before processing an interrupt, the CPU uses five bytes of the stack to save the contents of the CPU registers.

NOTE: *For M6805 compatibility, the H register is not stacked.*

During a subroutine call, the CPU uses two bytes of the stack to store the return address. The stack pointer decrements during pushes and increments during pulls.

NOTE: *Be careful when using nested subroutines. The CPU may overwrite data in the RAM during a subroutine or during the interrupt stacking operation.*

Section 4. FLASH Memory

4.1 Contents

4.2 Introduction 63
4.3 Functional Description 64
4.4 FLASH Control Register 65
4.5 FLASH Page Erase Operation 66
4.6 FLASH Mass Erase Operation 67
4.7 FLASH Program Operation. 68
4.8 FLASH Protection 70
4.8.1 FLASH Block Protect Register 70

4.2 Introduction

This section describes the operation of the embedded FLASH memory. This memory can be read, programmed, and erased from a single external supply. The program and erase operations are enabled through the use of an internal charge pump.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE08	FLASH Control Register (FLCR)	Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE09	FLASH Block Protect Register (FLBPR)	Read:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 4-1. FLASH I/O Register Summary

4.3 Functional Description

The FLASH memory consists of an array of 12,288 bytes for user memory plus a block of 38 bytes for user interrupt vectors and one byte for the mask option register. *An erased bit reads as logic 1 and a programmed bit reads as a logic 0*. The FLASH memory page size is defined as 128 bytes, and is the minimum size that can be erased in a page erase operation. Program and erase operations are facilitated through control bits in FLASH control register (FLCR). The address ranges for the FLASH memory are:

- $C000–$EFFF; user memory, 12,288 bytes
- $FFDA–$FFFF; user interrupt vectors, 38 bytes
- $FF80; mask option register

Programming tools are available from Freescale. Contact your local Freescale representative for more information.

NOTE: *A security feature prevents viewing of the FLASH contents.*[1]

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.

4.4 FLASH Control Register

The FLASH control register (FLCR) controls FLASH program and erase operations.

Address: $FE08

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 4-2. FLASH Control Register (FLCR)

HVEN — High Voltage Enable Bit

This read/write bit enables the charge pump to drive high voltages for program and erase operations in the array. HVEN can only be set if either PGM = 1 or ERASE = 1 and the proper sequence for program or erase is followed.

1 = High voltage enabled to array and charge pump on
0 = High voltage disabled to array and charge pump off

MASS — Mass Erase Control Bit

This read/write bit configures the memory for mass erase operation or block erase operation when the ERASE bit is set.

1 = Mass Erase operation selected
0 = Block Erase operation selected

ERASE — Erase Control Bit

This read/write bit configures the memory for erase operation. ERASE is interlocked with the PGM bit such that both bits cannot be equal to 1 or set to 1 at the same time.

1 = Erase operation selected
0 = Erase operation not selected

PGM — Program Control Bit

This read/write bit configures the memory for program operation. PGM is interlocked with the ERASE bit such that both bits cannot be equal to 1 or set to 1 at the same time.

1 = Program operation selected
0 = Program operation not selected

4.5 FLASH Page Erase Operation

Use the following procedure to erase a page of FLASH memory. A page consists of 128 consecutive bytes starting from addresses $xx00 or $xx80. The 38-byte user interrupt vectors area also forms a page. *The 38-byte user interrupt vectors cannot be erased by the page erase operation because of security reasons. Mass erase is required to erase this page.*

1. Set the ERASE bit and clear the MASS bit in the FLASH control register.
2. Write any data to any FLASH address within the page address range desired.
3. Wait for a time, t_{nvs} (10 μs).
4. Set the HVEN bit.
5. Wait for a time, t_{Erase} (1 ms).
6. Clear the ERASE bit.
7. Wait for a time, t_{nvh} (5 μs).
8. Clear the HVEN bit.
9. After time, t_{rcv} (1 μs), the memory can be accessed again in read mode.

NOTE: *Programming and erasing of FLASH locations cannot be performed by executing code from the FLASH memory; the code must be executed from RAM. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps.*

4.6 FLASH Mass Erase Operation

Use the following procedure to erase the entire FLASH memory to read as logic 1:

1. Set both the ERASE bit and the MASS bit in the FLASH control register.
2. Write any data to any FLASH address within the FLASH memory address range.
3. Wait for a time, t_{nvs} (10 μs).
4. Set the HVEN bit.
5. Wait for a time t_{MErase} (4 ms).
6. Clear the ERASE bit.
7. Wait for a time, t_{nvhl} (100 μs).
8. Clear the HVEN bit.
9. After time, t_{rcv} (1 μs), the memory can be accessed again in read mode.

NOTE: *Programming and erasing of FLASH locations cannot be performed by executing code from the FLASH memory; the code must be executed from RAM. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps.*

4.7 FLASH Program Operation

Programming of the FLASH memory is done on a row basis. A row consists of 64 consecutive bytes starting from addresses $xx00, $xx40, $xx80, or $xxC0. The procedure for programming a row of the FLASH memory is outlined below:

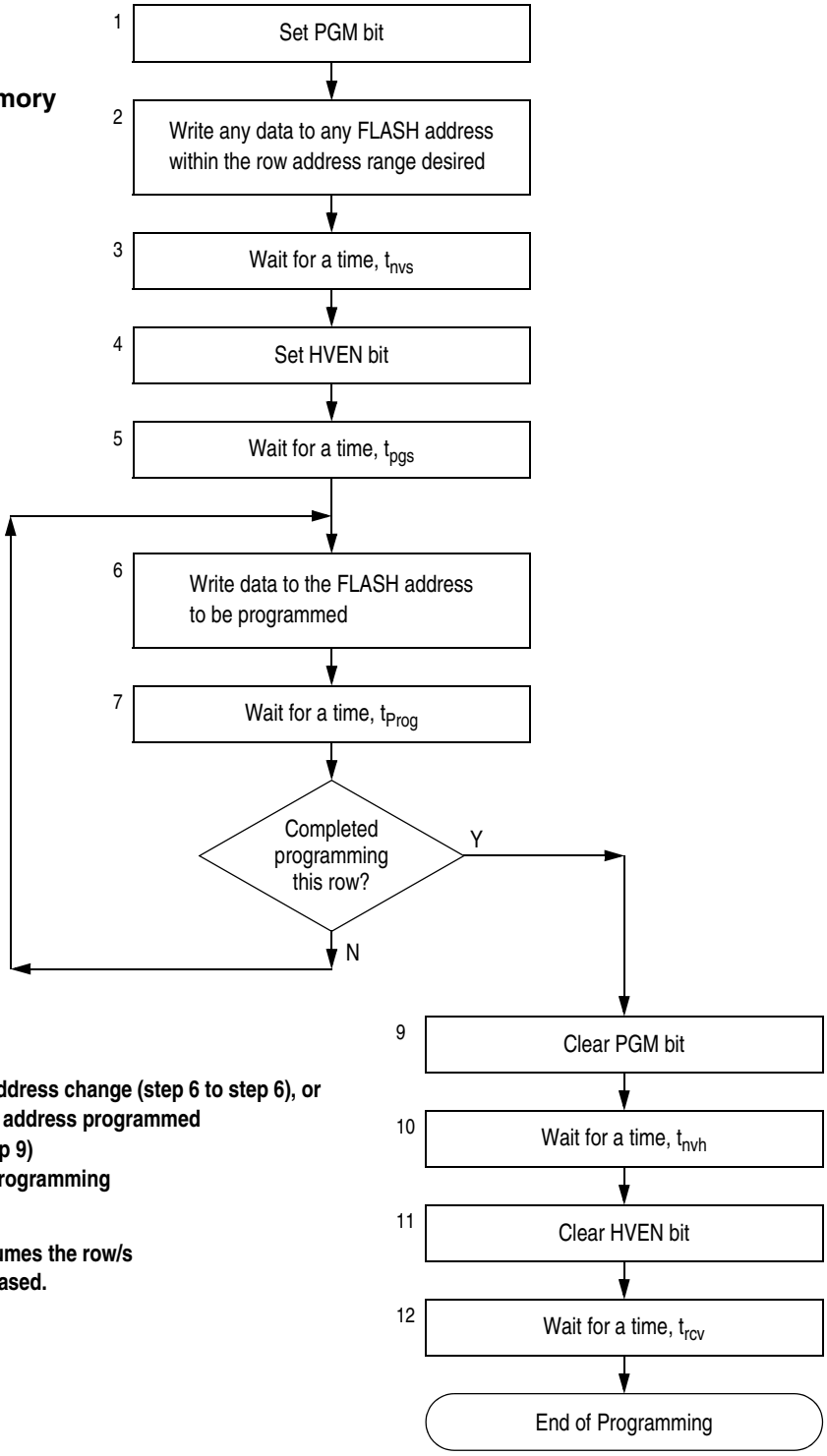
1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
2. Write any data to any FLASH address within the row address range desired.
3. Wait for a time, t_{nvs} (10 μs).
4. Set the HVEN bit.
5. Wait for a time, t_{pgs} (5 μs).
6. Write data to the FLASH address to be programmed.
7. Wait for time, t_{Prog} (30 μs).
8. Repeat step 6 and 7 until all the bytes within the row are programmed.
9. Clear the PGM bit.
10. Wait for time, t_{nvh} (5 μs).
11. Clear the HVEN bit.
12. After time, t_{rcv} (1 μs), the memory can be accessed again in read mode.

This program sequence is repeated throughout the memory until all data is programmed.

NOTE: *Programming and erasing of FLASH locations cannot be performed by executing code from the FLASH memory; the code must be executed from RAM. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps. Do not exceed t_{Prog} maximum. See* ***24.18 FLASH Memory Characteristics****.*

Figure 4-3 shows a flowchart representation for programming the FLASH memory.

Algorithm for programming a row (64 bytes) of FLASH memory



NOTE:

The time between each FLASH address change (step 6 to step 6), or the time between the last FLASH address programmed to clearing PGM bit (step 6 to step 9) must not exceed the maximum programming time, t_{Prog} max.

This row program algorithm assumes the row/s to be programmed are initially erased.

Figure 4-3. FLASH Programming Flowchart

4.8 FLASH Protection

Due to the ability of the on-board charge pump to erase and program the FLASH memory in the target application, provision is made to protect pages of memory from unintentional erase or program operations due to system malfunction. This protection is done by use of a FLASH block protect register (FLBPR). The FLBPR determines the range of the FLASH memory which is to be protected. The range of the protected area starts from a location defined by FLBPR and ends to the bottom of the FLASH memory ($FFFF). When the memory is protected, the HVEN bit cannot be set in either ERASE or PROGRAM operations.

NOTE: *When the FLBPR is cleared (all 0's), the entire FLASH memory is protected from being programmed and erased. When all the bits are set, the entire FLASH memory is accessible for program and erase.*

4.8.1 FLASH Block Protect Register

The FLASH block protect register is implemented as an 8-bit I/O register. The content of this register determine the starting location of the protected range within the FLASH memory.

Address: $FE09

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
Reset:	0	0	0	0	0	0	0	0

Figure 4-4. FLASH Block Protect Register (FLBPR)

BPR[7:0] — FLASH Block Protect Register Bit7 to Bit 0

BPR[7:1] represent bits [13:7] of a 16-bit memory address. Bits [15:14] are logic 1s and bits [6:0] are logic 0s.

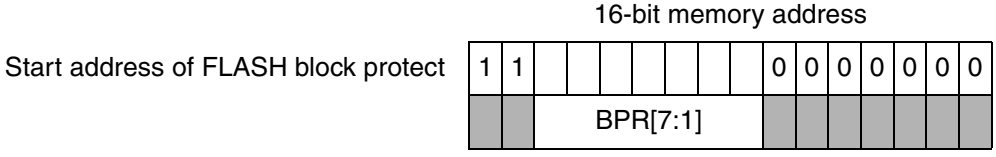


Figure 4-5. FLASH Block Protect Start Address

BPR0 is used only for BPR[7:0] = $FF, for no block protection.

The resultant 16-bit address is used for specifying the start address of the FLASH memory for block protection. The FLASH is protected from this start address to the end of FLASH memory, at $FFFF. With this mechanism, the protect start address can be XX00 or XX80 (at page boundaries) within the FLASH memory.

Examples of protect start address:

BPR[7:0]	Start of Address of Protect Range
$00 or $01	$C000 (11**00 0000 0**000 0000) The entire FLASH memory is protected.
$02 or $03	$C080 (11**00 0000 1**000 0000)
$04 or $05	$C100 (11**00 0001 0**000 0000)
$06 or $07	$C180 (11**00 0001 1**000 0000)
$08 or $09	$C200 (11**00 0010 0**000 0000)
and so on...	
$F8 or $F9	$FE00 (11**11 1110 0**000 0000)
$FA or $FB	$FE80 (11**11 1110 1**000 0000)
$FC or $FD	$FF00 (11**11 1111 0**000 0000)
$FE	$FF80 (11**11 1111 1**000 0000)
$FF	The entire FLASH memory is not protected.

Note:
The end address of the protected range is always $FFFF.

Section 5. Configuration and Mask Option Registers (CONFIG & MOR)

5.1 Contents

5.2 Introduction 73
5.3 Functional Description 74
5.4 Configuration Register 1 (CONFIG1) 75
5.5 Configuration Register 2 (CONFIG2) 77
5.6 Mask Option Register (MOR) 79

5.2 Introduction

This section describes the configuration registers, CONFIG1 and CONFIG2; and the mask option register, MOR.

The configuration registers enable or disable these options:

- Computer operating properly module (COP)
- COP timeout period ($2^{18} – 2^4$ or $2^{13} – 2^4$ ICLK cycles)
- Low-voltage inhibit (LVI) module power
- LVI module reset
- LVI module in stop mode
- LVI module voltage trip point selection
- STOP instruction
- Stop mode recovery time (32 ICLK cycles or 4096 ICLK cycles)
- Oscillator (internal, RC, and crystal) during stop mode
- Serial communications interface clock source (CGMXCLK or f_{BUS})
- Current detect output pin

The mask option register selects one of the following oscillator options as the MCU reference clock:

- Internal oscillator
- RC oscillator
- Crystal oscillator

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001D	Configuration Register 2 (CONFIG2)†	Read:	STOP_ ICLKEN	STOP_ RCLKEN	STOP_ XCLKEN	OSCCLK1	OSCCLK0	0	CDOEN	SCIBDSRC
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001F	Configuration Register 1 (CONFIG1)†	Read:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3	SSREC	STOP	COPD
		Write:								
		Reset:	0	0	0	0	0††	0	0	0
$FF80	Mask Option Register (MOR)*	Read:	OSCSEL1	OSCSEL0	R	R	R	R	R	R
		Write:								
		Erased:	1	1	1	1	1	1	1	1
* FLASH register.		Reset:	U	U	U	U	U	U	U	U

† One-time writable register after each reset.

†† Reset by POR only.

= Unimplemented R = Reserved

Figure 5-1. CONFIG and MOR Register Summary

5.3 Functional Description

The configuration registers and the mask option register are used in the initialization of various options. These two types of registers are configured differently:

- Configuration registers — Write-once registers after reset
- Mask option register — FLASH register (write by programming)

The configuration registers can be written once after each reset. All of the configuration register bits are cleared during reset. Since the various options affect the operation of the MCU, it is recommended that these registers be written immediately after reset. The configuration registers are located at $001D and $001F. The configurations register may be read at anytime.

NOTE: *The options except LVI5OR3 are one-time writable by the user after each reset. The LVI5OR3 bit is one-time writable by the user only after each POR (power-on reset). The CONFIG registers are not in the FLASH memory but are special registers containing one-time writable latches after each reset. Upon a reset, the CONFIG registers default to predetermined settings as shown in* ***Figure 5-2*** *and* ***Figure 5-3****.*

The mask option register (MOR) is used for selecting one of the three clock options for the MCU. The MOR is a byte located in FLASH memory, and is written to by a FLASH programming routine.

5.4 Configuration Register 1 (CONFIG1)

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3	SSREC	STOP	COPD
Reset:	0	0	0	0	0*	0	0	0

* Reset by POR only.

Figure 5-2. Configuration Register 1 (CONFIG1)

COPRS — COP Rate Select

COPRS selects the COP time-out period. Reset clears COPRS. (See **Section 21. Computer Operating Properly (COP)**.)

1 = COP time out period = $2^{13} – 2^4$ ICLK cycles
0 = COP time out period = $2^{18} – 2^4$ ICLK cycles

LVISTOP — LVI Enable in Stop Mode

When the LVIPWRD bit is clear, setting the LVISTOP bit enables the LVI to operate during stop mode. Reset clears LVISTOP. (See **Section 22. Low-Voltage Inhibit (LVI)**.)

1 = LVI enabled during stop mode
0 = LVI disabled during stop mode

LVIRSTD — LVI Reset Disable

LVIRSTD disables the reset signal from the LVI module. (See **Section 22. Low-Voltage Inhibit (LVI)**.)

1 = LVI module resets disabled
0 = LVI module resets enabled

LVIPWRD — LVI Power Disable Bit

LVIPWRD disables the LVI module. (See **Section 22. Low-Voltage Inhibit (LVI)**.)

1 = LVI module power disabled
0 = LVI module power enabled

LVI5OR3 — LVI 5V or 3V Operating Mode

LVI5OR3 selects the voltage operating mode of the LVI module. (See **Section 22. Low-Voltage Inhibit (LVI)**.) The voltage mode selected for the LVI should match the operating V_{DD}. See **Section 24. Electrical Specifications** for the LVI voltage trip points for each of the modes.

1 = LVI operates in 5V mode
0 = LVI operates in 3V mode

SSREC — Short Stop Recovery

SSREC enables the CPU to exit stop mode with a delay of 32 ICLK cycles instead of a 4096 ICLK cycle delay.

1 = Stop mode recovery after 32 ICLK cycles
0 = Stop mode recovery after 4096 ICLK cycles

NOTE: *Exiting stop mode by pulling reset will result in the long stop recovery.*

If using an external crystal oscillator, and it is disabled during stop mode (STOP_XCLKEN=0), do not set the SSREC bit.

NOTE: *When the LVI is disabled in stop mode (LVISTOP=0), the system stabilization time for long stop recovery (4096 ICLK cycles) gives a delay longer than the LVI's turn-on time. There is no period where the MCU is not protected from a low power condition. However, when using the short stop recovery configuration option, the 32 ICLK delay is less than the LVI's turn-on time and there exists a period in start-up where the LVI is not protecting the MCU.*

STOP — STOP Instruction Enable

STOP enables the STOP instruction.

1 = STOP instruction enabled

0 = STOP instruction treated as illegal opcode

COPD — COP Disable Bit

COPD disables the COP module. (See **Section 21. Computer Operating Properly (COP)**.)

1 = COP module disabled

0 = COP module enabled

5.5 Configuration Register 2 (CONFIG2)

Address: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	STOP_ ICLKEN	STOP_ RCLKEN	STOP_ XCLKEN	OSCCLK1	OSCCLK0	0	CDOEN	SCIBDSRC
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 5-3. Configuration Register 2 (CONFIG2)

STOP_ICLKEN — Internal Oscillator Stop Mode Disable

STOP_ICLKEN disables the internal oscillator during stop mode. Setting the STOP_ICLKEN bit disables the oscillator during stop mode. (See **7.4 Internal Oscillator**).

Reset clears this bit.

1 = Internal oscillator disabled during stop mode

0 = Internal oscillator enabled to operate during stop mode

STOP_RCLKEN — RC Oscillator Stop Mode Enable

STOP_RCLKEN enables the RC oscillator to continue operating during stop mode. Setting the STOP_RCLKEN bit allows the oscillator to operate continuously even during stop mode. This is useful for driving the timebase module to allow it to generate periodic wake up while in stop mode. (See **Section 8. Clock Generator Module (CGM)** and subsection **8.8.2 Stop Mode**.)

Reset clears this bit.

1 = RC oscillator enabled to operate during stop mode

0 = RC oscillator disabled during stop mode

STOP_XCLKEN — Crystal Oscillator Stop Mode Enable

STOP_XCLKEN enables the crystal (x-tal) oscillator to continue operating during stop mode. Setting the STOP_XCLKEN bit allows the x-tal oscillator to operate continuously even during stop mode. This is useful for driving the timebase module to allow it to generate periodic wake up while in stop mode. (See **Section 8. Clock Generator Module (CGM)** and subsection **8.8.2 Stop Mode**.) Reset clears this bit.

1 = X-tal oscillator enabled to operate during stop mode
0 = X-tal oscillator disabled during stop mode

OSCCLK1, OSCCLK0 — Oscillator Output Control Bits

OSCCLK1 and OSCCLK0 select which oscillator output to be driven out as OSCCLK to the timebase module (TBM). Reset clears these two bits.

OSCCLK1	OSCCLK0	Timebase Clock Source
0	0	Internal oscillator (ICLK)
0	1	RC oscillator (RCCLK)
1	0	X-tal oscillator (XTAL)
1	1	Not used

CDOEN — Current-Flow Detect Output Enable

CDOEN enables the port pin PC0/PWM0/CD as the CD output pin for the current detect flag (CDIF) from the analog module. Reset clears the CDOEN bit.

1 = PCO/PWMO/CD pin enabled as CD output pin,
PTC0 and PWM0 functions are disabled.
0 = PTC0/PWM/CD pin disabled as CD output pin,
PTC0 or PWM0 functions are available; see **18.5.1 Port C Data Register (PTC)**.

SCIBDSRC — SCI Baud Rate Clock Source

SCIBDSRC selects the clock source used for the SCI. The setting of this bit affects the frequency at which the SCI operates.

1 = Internal data bus clock, f_{BUS}, is used as clock source for SCI

0 = Oscillator clock, CGMXCLK, is used as clock source for SCI

5.6 Mask Option Register (MOR)

The mask option register (MOR) is used for selecting one of the three clock options for the MCU. The MOR is a byte located in FLASH memory, and is written to by a FLASH programming routine.

Address: $FF80

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	OSCSEL1	OSCSEL0	R	R	R	R	R	R
Erased:	1	1	1	1	1	1	1	1
Reset:	U	U	U	U	U	U	U	U

R = Reserved

Figure 5-4. Mask Option Register (MOR)

OSCSEL1, OSCSEL0 — Oscillator Selection Bits

OSCSEL1 and OSCSEL0 select which oscillator is used for the MCU CGMXCLK clock. The erase state of these two bits is logic 1. These bits are unaffected by reset. (See **Table 5-1**).

Bits 5–0 — Should be left as 1's.

Table 5-1. CGMXCLK Clock Selection

OSCSEL1	OSCSEL0	CGMXCLK	OSC2 pin	Comments
0	0	—	—	Not used
0	1	ICLK	f_{BUS}	Internal oscillator generates the CGMXCLK.
1	0	RCCLK	f_{BUS}	RC oscillator generates the CGMXCLK. Internal oscillator is available after each POR or reset.
1	1	X-TAL	Inverting output of XTAL	X-tal oscillator generates the CGMXCLK. Internal oscillator is available after each POR or reset.

NOTE: *The internal oscillator is a free running oscillator and is available after each POR or reset. It is turned-off in stop mode by clearing the STOP_ICLKEN bit in CONFIG2.*

Section 6. Central Processor Unit (CPU)

6.1 Contents

6.2 Introduction 81
6.3 Features 82
6.4 CPU Registers 82
6.4.1 Accumulator 83
6.4.2 Index Register 84
6.4.3 Stack Pointer 84
6.4.4 Program Counter 85
6.4.5 Condition Code Register 85
6.5 Arithmetic/Logic Unit (ALU) 88
6.6 Low-Power Modes 88
6.6.1 Wait Mode 88
6.6.2 Stop Mode 89
6.7 CPU During Break Interrupts 89
6.8 Instruction Set Summary 89
6.9 Opcode Map 89

6.2 Introduction

The M68HC08 CPU (central processor unit) is an enhanced and fully object-code-compatible version of the M68HC05 CPU. The *CPU08 Reference Manual* (Freescale document order number CPU08RM/AD) contains a description of the CPU instruction set, addressing modes, and architecture.

6.3 Features

- Object code fully upward-compatible with M68HC05 Family
- 16-bit stack pointer with stack manipulation instructions
- 16-bit index register with x-register manipulation instructions
- 8-MHz CPU internal bus frequency
- 64K-byte program/data memory space
- 16 addressing modes
- Memory-to-memory data moves without using accumulator
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- Enhanced binary-coded decimal (BCD) data handling
- Modular architecture with expandable internal bus definition for extension of addressing range beyond 64K-bytes
- Low-power stop and wait modes

6.4 CPU Registers

Figure 6-1 shows the five CPU registers. CPU registers are not part of the memory map.

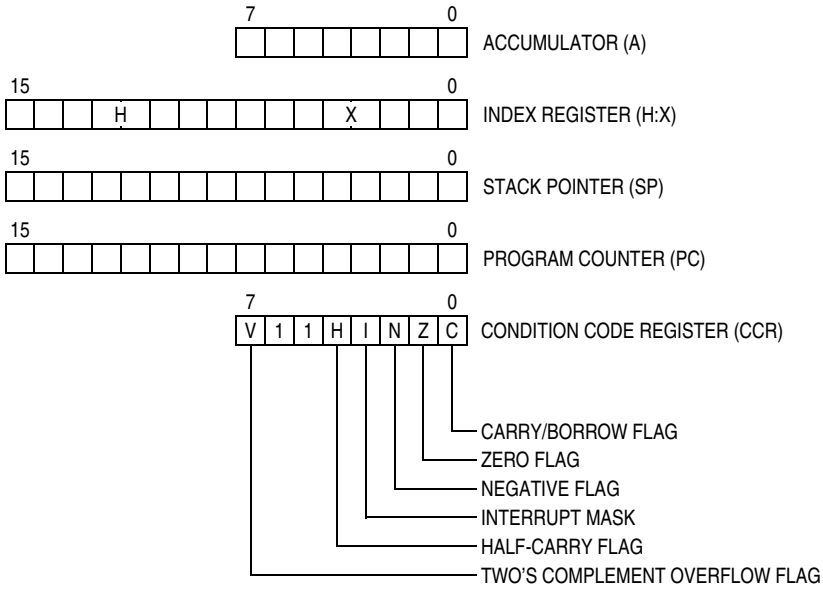


Figure 6-1. CPU Registers

6.4.1 Accumulator

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and the results of arithmetic/logic operations.

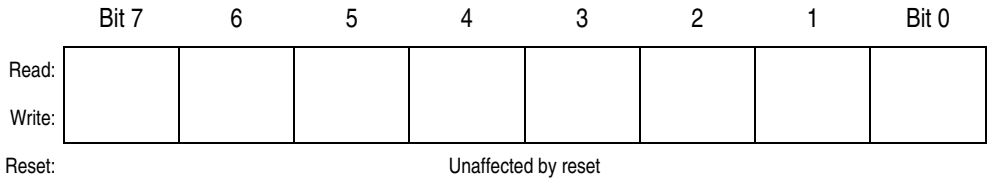


Figure 6-2. Accumulator (A)

6.4.2 Index Register

The 16-bit index register allows indexed addressing of a 64K-byte memory space. H is the upper byte of the index register, and X is the lower byte. H:X is the concatenated 16-bit index register.

In the indexed addressing modes, the CPU uses the contents of the index register to determine the conditional address of the operand.

The index register can serve also as a temporary data storage location.

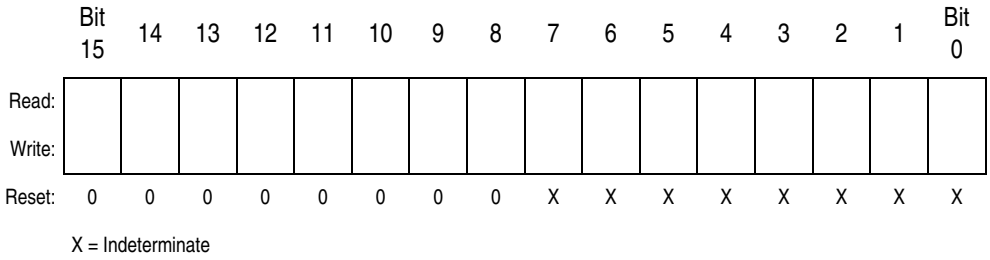


Figure 6-3. Index Register (H:X)

6.4.3 Stack Pointer

The stack pointer is a 16-bit register that contains the address of the next location on the stack. During a reset, the stack pointer is preset to $00FF. The reset stack pointer (RSP) instruction sets the least significant byte to $FF and does not affect the most significant byte. The stack pointer decrements as data is pushed onto the stack and increments as data is pulled from the stack.

In the stack pointer 8-bit offset and 16-bit offset addressing modes, the stack pointer can function as an index register to access data on the stack. The CPU uses the contents of the stack pointer to determine the conditional address of the operand.

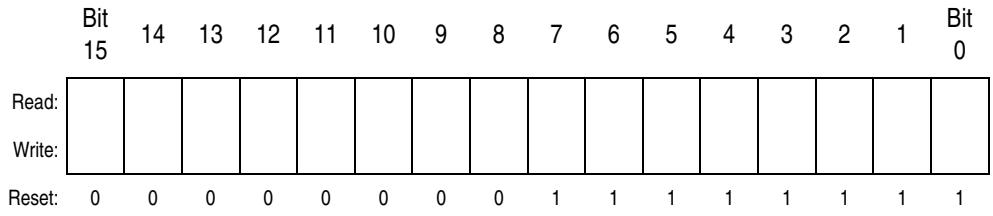


Figure 6-4. Stack Pointer (SP)

NOTE: *The location of the stack is arbitrary and may be relocated anywhere in RAM. Moving the SP out of page 0 ($0000 to $00FF) frees direct address (page 0) space. For correct operation, the stack pointer must point only to RAM locations.*

6.4.4 Program Counter

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched.

Normally, the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

During reset, the program counter is loaded with the reset vector address located at $FFFE and $FFFF. The vector address is the address of the first instruction to be executed after exiting the reset state.

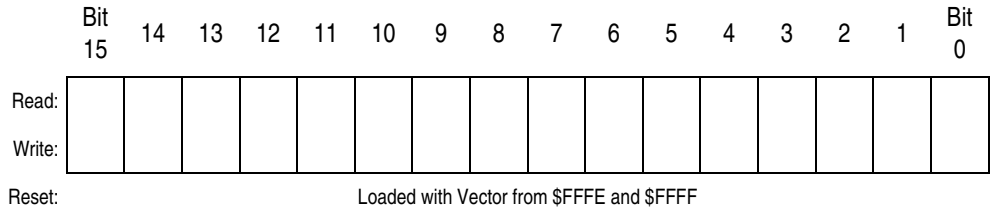


Figure 6-5. Program Counter (PC)

6.4.5 Condition Code Register

The 8-bit condition code register contains the interrupt mask and five flags that indicate the results of the instruction just executed. Bits 6 and

5 are set permanently to logic 1. The following paragraphs describe the functions of the condition code register.

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

X = Indeterminate

Figure 6-6. Condition Code Register (CCR)

V — Overflow Flag

The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag.

1 = Overflow
0 = No overflow

H — Half-Carry Flag

The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C flags to determine the appropriate correction factor.

1 = Carry between bits 3 and 4
0 = No carry between bits 3 and 4

I — Interrupt Mask

When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the interrupt vector is fetched.

1 = Interrupts disabled
0 = Interrupts enabled

NOTE: *To maintain M6805 Family compatibility, the upper byte of the index register (H) is not stacked automatically. If the interrupt service routine modifies H, then the user must stack and unstack H using the PSHH and PULH instructions.*

After the I bit is cleared, the highest-priority interrupt request is serviced first.

A return-from-interrupt (RTI) instruction pulls the CPU registers from the stack and restores the interrupt mask from the stack. After any reset, the interrupt mask is set and can be cleared only by the clear interrupt mask software instruction (CLI).

N — Negative flag

The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result.

1 = Negative result
0 = Non-negative result

Z — Zero flag

The CPU sets the zero flag when an arithmetic operation, logic operation, or data manipulation produces a result of $00.

1 = Zero result
0 = Non-zero result

C — Carry/Borrow Flag

The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some instructions — such as bit test and branch, shift, and rotate — also clear or set the carry/borrow flag.

1 = Carry out of bit 7

0 = No carry out of bit 7

6.5 Arithmetic/Logic Unit (ALU)

The ALU performs the arithmetic and logic operations defined by the instruction set.

Refer to the *CPU08 Reference Manual* (Freescale document order number CPU08RM/AD) for a description of the instructions and addressing modes and more detail about the architecture of the CPU.

6.6 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

6.6.1 Wait Mode

The WAIT instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling interrupts. After exit from wait mode by interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

6.6.2 Stop Mode

The STOP instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling external interrupts. After exit from stop mode by external interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

After exiting stop mode, the CPU clock begins running after the oscillator stabilization delay.

6.7 CPU During Break Interrupts

If a break module is present on the MCU, the CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with $FFFC:$FFFD or with $FEFC:$FEFD in monitor mode

The break interrupt begins after completion of the CPU instruction in progress. If the break address register match occurs on the last cycle of a CPU instruction, the break interrupt begins immediately.

A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation if the break interrupt has been deasserted.

6.8 Instruction Set Summary

6.9 Opcode Map

See **Table 6-2**.

Table 6-1. Instruction Set Summary

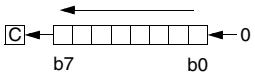
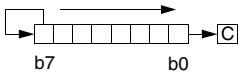
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X ADC *opr*,SP ADC *opr*,SP	Add with Carry	A ← (A) + (M) + (C)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A9 B9 C9 D9 E9 F9 9EE9 9ED9	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X ADD *opr*,SP ADD *opr*,SP	Add without Carry	A ← (A) + (M)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	AB BB CB DB EB FB 9EEB 9EDB	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
AIS #*opr*	Add Immediate Value (Signed) to SP	SP ← (SP) + (16 « M)	–	–	–	–	–	–	IMM	A7	ii	2
AIX #*opr*	Add Immediate Value (Signed) to H:X	H:X ← (H:X) + (16 « M)	–	–	–	–	–	–	IMM	AF	ii	2
AND #*opr* AND *opr* AND *opr* AND *opr*,X AND *opr*,X AND ,X AND *opr*,SP AND *opr*,SP	Logical AND	A ← (A) & (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A4 B4 C4 D4 E4 F4 9EE4 9ED4	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X ASL *opr*,SP	Arithmetic Shift Left (Same as LSL)		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR *opr*,X ASR *opr*,SP	Arithmetic Shift Right		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	37 47 57 67 77 9E67	dd ff ff	4 1 1 4 3 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + rel ? (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BCLR *n*, *opr*	Clear Bit n in M	Mn ← 0	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	4 4 4 4 4 4 4 4

Table 6-1. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? (C) = 1	–	–	–	–	–	–	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? (Z) = 1	–	–	–	–	–	–	REL	27	rr	3
BGE *opr*	Branch if Greater Than or Equal To (Signed Operands)	PC ← (PC) + 2 + *rel* ? (N ⊕ V) = 0	–	–	–	–	–	–	REL	90	rr	3
BGT *opr*	Branch if Greater Than (Signed Operands)	PC ← (PC) + 2 + *rel* ? (Z) | (N ⊕ V) = 0	–	–	–	–	–	–	REL	92	rr	3
BHCC *rel*	Branch if Half Carry Bit Clear	PC ← (PC) + 2 + *rel* ? (H) = 0	–	–	–	–	–	–	REL	28	rr	3
BHCS *rel*	Branch if Half Carry Bit Set	PC ← (PC) + 2 + *rel* ? (H) = 1	–	–	–	–	–	–	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? (C) | (Z) = 0	–	–	–	–	–	–	REL	22	rr	3
BHS *rel*	Branch if Higher or Same (Same as BCC)	PC ← (PC) + 2 + *rel* ? (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BIH *rel*	Branch if $\overline{\text{IRQ}}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 1	–	–	–	–	–	–	REL	2F	rr	3
BIL *rel*	Branch if $\overline{\text{IRQ}}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 0	–	–	–	–	–	–	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X BIT *opr*,SP BIT *opr*,SP	Bit Test	(A) & (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A5 B5 C5 D5 E5 F5 9EE5 9ED5	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
BLE *opr*	Branch if Less Than or Equal To (Signed Operands)	PC ← (PC) + 2 + *rel* ? (Z) | (N ⊕ V) = 1	–	–	–	–	–	–	REL	93	rr	3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? (C) = 1	–	–	–	–	–	–	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? (C) | (Z) = 1	–	–	–	–	–	–	REL	23	rr	3
BLT *opr*	Branch if Less Than (Signed Operands)	PC ← (PC) + 2 + *rel* ? (N ⊕ V) =1	–	–	–	–	–	–	REL	91	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? (I) = 0	–	–	–	–	–	–	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? (N) = 1	–	–	–	–	–	–	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? (I) = 1	–	–	–	–	–	–	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? (Z) = 0	–	–	–	–	–	–	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? (N) = 0	–	–	–	–	–	–	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel*	–	–	–	–	–	–	REL	20	rr	3

Table 6-1. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
BRCLR *n,opr,rel*	Branch if Bit *n* in M Clear	PC ← (PC) + 3 + *rel* ? (Mn) = 0	–	–	–	–	–	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2	–	–	–	–	–	–	REL	21	rr	3
BRSET *n,opr,rel*	Branch if Bit *n* in M Set	PC ← (PC) + 3 + *rel* ? (Mn) = 1	–	–	–	–	–	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BSET *n,opr*	Set Bit *n* in M	Mn ← 1	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	4 4 4 4 4 4 4 4
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	–	–	–	–	–	–	REL	AD	rr	4
CBEQ *opr,rel* CBEQA #*opr,rel* CBEQX #*opr,rel* CBEQ *opr*,X+,*rel* CBEQ X+,*rel* CBEQ *opr*,SP,*rel*	Compare and Branch if Equal	PC ← (PC) + 3 + rel ? (A) – (M) = $00 PC ← (PC) + 3 + rel ? (A) – (M) = $00 PC ← (PC) + 3 + rel ? (X) – (M) = $00 PC ← (PC) + 3 + rel ? (A) – (M) = $00 PC ← (PC) + 2 + rel ? (A) – (M) = $00 PC ← (PC) + 4 + rel ? (A) – (M) = $00	–	–	–	–	–	–	DIR IMM IMM IX1+ IX+ SP1	31 41 51 61 71 9E61	dd rr ii rr ii rr ff rr rr ff rr	5 4 4 5 4 6
CLC	Clear Carry Bit	C ← 0	–	–	–	–	–	0	INH	98		1
CLI	Clear Interrupt Mask	I ← 0	–	–	0	–	–	–	INH	9A		2
CLR *opr* CLRA CLRX CLRH CLR *opr*,X CLR ,X CLR *opr*,SP	Clear	M ← $00 A ← $00 X ← $00 H ← $00 M ← $00 M ← $00 M ← $00	0	–	–	0	1	–	DIR INH INH INH IX1 IX SP1	3F 4F 5F 8C 6F 7F 9E6F	dd ff ff	3 1 1 1 3 2 4

Table 6-1. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X CMP *opr*,SP CMP *opr*,SP	Compare A with M	(A) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A1 B1 C1 D1 E1 F1 9EE1 9ED1	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
COM *opr* COMA COMX COM *opr*,X COM ,X COM *opr*,SP	Complement (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (M) X ← ($\overline{X}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	0	–	–	↕	↕	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd ff ff	4 1 1 4 3 5
CPHX #*opr* CPHX *opr*	Compare H:X with M	(H:X) – (M:M + 1)	↕	–	–	↕	↕	↕	IMM DIR	65 75	ii ii+1 dd	3 4
CPX #*opr* CPX *opr* CPX *opr* CPX ,X CPX *opr*,X CPX *opr*,X CPX *opr*,SP CPX *opr*,SP	Compare X with M	(X) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A3 B3 C3 D3 E3 F3 9EE3 9ED3	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
DAA	Decimal Adjust A	$(A)_{10}$	U	–	–	↕	↕	↕	INH	72		2
DBNZ *opr,rel* DBNZA *rel* DBNZX *rel* DBNZ *opr*,X,*rel* DBNZ X,*rel* DBNZ *opr*,SP,*rel*	Decrement and Branch if Not Zero	A ← (A) –1 or M ← (M) –1 or X← (X) –1 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 4 + *rel* ? (result) ≠ 0	–	–	–	–	–	–	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	5 3 3 5 4 6
DEC *opr* DECA DECX DEC *opr*,X DEC ,X DEC *opr*,SP	Decrement	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1 M ← (M) – 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff	4 1 1 4 3 5
DIV	Divide	A ← (H:A)/(X) H ← Remainder	–	–	–	–	↕	↕	INH	52		7

Table 6-1. Instruction Set Summary (Continued)

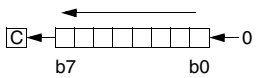
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X EOR *opr*,SP EOR *opr*,SP	Exclusive OR M with A	A ← (A ⊕ M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A8 B8 C8 D8 E8 F8 9EE8 9ED8	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
INC *opr* INCA INCX INC *opr*,X INC ,X INC *opr*,SP	Increment	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1 M ← (M) + 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff	4 1 1 4 3 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff 	2 3 4 3 2
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + *n* (*n* = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Unconditional Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff 	4 5 6 5 4
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X LDA *opr*,SP LDA *opr*,SP	Load A from M	A ← (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A6 B6 C6 D6 E6 F6 9EE6 9ED6	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LDHX #*opr* LDHX *opr*	Load H:X from M	H:X ← (M:M + 1)	0	–	–	↕	↕	–	IMM DIR	45 55	ii jj dd	3 4
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X LDX *opr*,SP LDX *opr*,SP	Load X from M	X ← (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	AE BE CE DE EE FE 9EEE 9EDE	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LSL *opr* LSLA LSLX LSL *opr*,X (Same as ASL) LSL ,X LSL *opr*,SP	Logical Shift Left  		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5

Table 6-1. Instruction Set Summary (Continued)

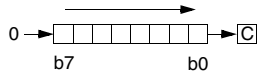
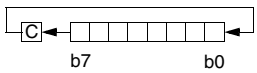
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
LSR *opr* LSRA LSR*X* LSR *opr*,X LSR ,X LSR *opr*,SP	Logical Shift Right		↕	–	–	0	↕	↕	DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff	4 1 1 4 3 5
MOV *opr,opr* MOV *opr,*X+ MOV #*opr,opr* MOV X+*,opr*	Move	$(M)_{Destination} \leftarrow (M)_{Source}$ H:X ← (H:X) + 1 (IX+D, DIX+)	0	–	–	↕	↕	–	DD DIX+ IMD IX+D	4E 5E 6E 7E	dd dd dd ii dd dd	5 4 4 4
MUL	Unsigned multiply	X:A ← (X) × (A)	–	0	–	–	–	0	INH	42		5
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X NEG *opr*,SP	Negate (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	30 40 50 60 70 9E60	dd ff ff	4 1 1 4 3 5
NOP	No Operation	None	–	–	–	–	–	–	INH	9D		1
NSA	Nibble Swap A	A ← (A[3:0]:A[7:4])	–	–	–	–	–	–	INH	62		3
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X ORA *opr*,SP ORA *opr*,SP	Inclusive OR A and M	A ← (A) | (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	AA BA CA DA EA FA 9EEA 9EDA	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
PSHA	Push A onto Stack	Push (A); SP ← (SP) – 1	–	–	–	–	–	–	INH	87		2
PSHH	Push H onto Stack	Push (H); SP ← (SP) – 1	–	–	–	–	–	–	INH	8B		2
PSHX	Push X onto Stack	Push (X); SP ← (SP) – 1	–	–	–	–	–	–	INH	89		2
PULA	Pull A from Stack	SP ← (SP + 1); Pull (A)	–	–	–	–	–	–	INH	86		2
PULH	Pull H from Stack	SP ← (SP + 1); Pull (H)	–	–	–	–	–	–	INH	8A		2
PULX	Pull X from Stack	SP ← (SP + 1); Pull (X)	–	–	–	–	–	–	INH	88		2
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X ROL *opr*,SP	Rotate Left through Carry		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	39 49 59 69 79 9E69	dd ff ff	4 1 1 4 3 5

Table 6-1. Instruction Set Summary (Continued)

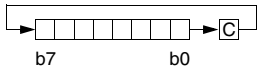
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ROR *opr* RORA RORX ROR *opr*,X ROR ,X ROR *opr*,SP	Rotate Right through Carry		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	4 1 1 4 3 5
RSP	Reset Stack Pointer	SP ← $FF	–	–	–	–	–	–	INH	9C		1
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	↕	INH	80		7
RTS	Return from Subroutine	SP ← SP + 1; Pull (PCH) SP ← SP + 1; Pull (PCL)	–	–	–	–	–	–	INH	81		4
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X SBC *opr*,SP SBC *opr*,SP	Subtract with Carry	A ← (A) – (M) – (C)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A2 B2 C2 D2 E2 F2 9EE2 9ED2	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
SEC	Set Carry Bit	C ← 1	–	–	–	–	–	1	INH	99		1
SEI	Set Interrupt Mask	I ← 1	–	–	1	–	–	–	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X STA *opr*,SP STA *opr*,SP	Store A in M	M ← (A)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh ll ee ff ff ff ee ff	3 4 4 3 2 4 5
STHX *opr*	Store H:X in M	(M:M + 1) ← (H:X)	0	–	–	↕	↕	–	DIR	35	dd	4
STOP	Enable $\overline{\text{IRQ}}$ Pin; Stop Oscillator	I ← 0; Stop Oscillator	–	–	0	–	–	–	INH	8E		1
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X STX *opr*,SP STX *opr*,SP	Store X in M	M ← (X)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP1 SP2	BF CF DF EF FF 9EEF 9EDF	dd hh ll ee ff ff ff ee ff	3 4 4 3 2 4 5

Table 6-1. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X SUB *opr*,SP SUB *opr*,SP	Subtract	A ← (A) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A0 B0 C0 D0 E0 F0 9EE0 9ED0	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	–	–	1	–	–	–	INH	83		9
TAP	Transfer A to CCR	CCR ← (A)	↕	↕	↕	↕	↕	↕	INH	84		2
TAX	Transfer A to X	X ← (A)	–	–	–	–	–	–	INH	97		1
TPA	Transfer CCR to A	A ← (CCR)	–	–	–	–	–	–	INH	85		1
TST *opr* TSTA TSTX TST *opr*,X TST ,X TST *opr*,SP	Test for Negative or Zero	(A) – $00 or (X) – $00 or (M) – $00	0	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3D 4D 5D 6D 7D 9E6D	dd ff ff	3 1 1 3 2 4
TSX	Transfer SP to H:X	H:X ← (SP) + 1	–	–	–	–	–	–	INH	95		2
TXA	Transfer X to A	A ← (X)	–	–	–	–	–	–	INH	9F		1
TXS	Transfer H:X to SP	(SP) ← (H:X) – 1	–	–	–	–	–	–	INH	94		2

Table 6-1. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				

A	Accumulator	*n*	Any bit
C	Carry/borrow bit	*opr*	Operand (one or two bytes)
CCR	Condition code register	PC	Program counter
dd	Direct address of operand	PCH	Program counter high byte
dd rr	Direct address of operand and relative offset of branch instruction	PCL	Program counter low byte
DD	Direct to direct addressing mode	REL	Relative addressing mode
DIR	Direct addressing mode	*rel*	Relative program counter offset byte
DIX+	Direct to indexed with post increment addressing mode	rr	Relative program counter offset byte
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	SP1	Stack pointer, 8-bit offset addressing mode
EXT	Extended addressing mode	SP2	Stack pointer 16-bit offset addressing mode
ff	Offset byte in indexed, 8-bit offset addressing	SP	Stack pointer
H	Half-carry bit	U	Undefined
H	Index register high byte	V	Overflow bit
hh ll	High and low bytes of operand address in extended addressing	X	Index register low byte
I	Interrupt mask	Z	Zero bit
ii	Immediate operand byte	&	Logical AND
IMD	Immediate source to direct destination addressing mode	|	Logical OR
IMM	Immediate addressing mode	⊕	Logical EXCLUSIVE OR
INH	Inherent addressing mode	( )	Contents of
IX	Indexed, no offset addressing mode	–( )	Negation (two's complement)
IX+	Indexed, no offset, post increment addressing mode	#	Immediate value
IX+D	Indexed with post increment to direct addressing mode	«	Sign extend
IX1	Indexed, 8-bit offset addressing mode	←	Loaded with
IX1+	Indexed, 8-bit offset, post increment addressing mode	?	If
IX2	Indexed, 16-bit offset addressing mode	:	Concatenated with
M	Memory location	↕	Set or cleared
N	Negative bit	—	Not affected

Table 6-2. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write						Control		Register/Memory							
	DIR	DIR	REL	DIR	INH	INH	IX1	SP1	IX	INH	INH	IMM	DIR	EXT	IX2	SP2	IX1	SP1	IX
MSB / LSB	0	1	2	3	4	5	6	9E6	7	8	9	A	B	C	D	9ED	E	9EE	F
0	5 BRSET0 3 DIR	4 BSET0 2 DIR	3 BRA 2 REL	4 NEG 2 DIR	1 NEGA 1 INH	1 NEGX 1 INH	4 NEG 2 IX1	5 NEG 3 SP1	3 NEG 1 IX	7 RTI 1 INH	3 BGE 2 REL	2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	4 SUB 3 IX2	5 SUB 4 SP2	3 SUB 2 IX1	4 SUB 3 SP1	2 SUB 1 IX
1	5 BRCLR0 3 DIR	4 BCLR0 2 DIR	3 BRN 2 REL	5 CBEQ 3 DIR	4 CBEQA 3 IMM	4 CBEQX 3 IMM	5 CBEQ 3 IX1+	6 CBEQ 4 SP1	4 CBEQ 2 IX+	4 RTS 1 INH	3 BLT 2 REL	2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	4 CMP 3 IX2	5 CMP 4 SP2	3 CMP 2 IX1	4 CMP 3 SP1	2 CMP 1 IX
2	5 BRSET1 3 DIR	4 BSET1 2 DIR	3 BHI 2 REL		5 MUL 1 INH	7 DIV 1 INH	3 NSA 1 INH		2 DAA 1 INH		3 BGT 2 REL	2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	4 SBC 3 IX2	5 SBC 4 SP2	3 SBC 2 IX1	4 SBC 3 SP1	2 SBC 1 IX
3	5 BRCLR1 3 DIR	4 BCLR1 2 DIR	3 BLS 2 REL	4 COM 2 DIR	1 COMA 1 INH	1 COMX 1 INH	4 COM 2 IX1	5 COM 3 SP1	3 COM 1 IX	9 SWI 1 INH	3 BLE 2 REL	2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	4 CPX 3 IX2	5 CPX 4 SP2	3 CPX 2 IX1	4 CPX 3 SP1	2 CPX 1 IX
4	5 BRSET2 3 DIR	4 BSET2 2 DIR	3 BCC 2 REL	4 LSR 2 DIR	1 LSRA 1 INH	1 LSRX 1 INH	4 LSR 2 IX1	5 LSR 3 SP1	3 LSR 1 IX	2 TAP 1 INH	2 TXS 1 INH	2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	4 AND 3 IX2	5 AND 4 SP2	3 AND 2 IX1	4 AND 3 SP1	2 AND 1 IX
5	5 BRCLR2 3 DIR	4 BCLR2 2 DIR	3 BCS 2 REL	4 STHX 2 DIR	3 LDHX 3 IMM	4 LDHX 2 DIR	3 CPHX 3 IMM		4 CPHX 2 DIR	1 TPA 1 INH	2 TSX 1 INH	2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	4 BIT 3 IX2	5 BIT 4 SP2	3 BIT 2 IX1	4 BIT 3 SP1	2 BIT 1 IX
6	5 BRSET3 3 DIR	4 BSET3 2 DIR	3 BNE 2 REL	4 ROR 2 DIR	1 RORA 1 INH	1 RORX 1 INH	4 ROR 2 IX1	5 ROR 3 SP1	3 ROR 1 IX	2 PULA 1 INH		2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	4 LDA 3 IX2	5 LDA 4 SP2	3 LDA 2 IX1	4 LDA 3 SP1	2 LDA 1 IX
7	5 BRCLR3 3 DIR	4 BCLR3 2 DIR	3 BEQ 2 REL	4 ASR 2 DIR	1 ASRA 1 INH	1 ASRX 1 INH	4 ASR 2 IX1	5 ASR 3 SP1	3 ASR 1 IX	2 PSHA 1 INH	1 TAX 1 INH	2 AIS 2 IMM	3 STA 2 DIR	4 STA 3 EXT	4 STA 3 IX2	5 STA 4 SP2	3 STA 2 IX1	4 STA 3 SP1	2 STA 1 IX
8	5 BRSET4 3 DIR	4 BSET4 2 DIR	3 BHCC 2 REL	4 LSL 2 DIR	1 LSLA 1 INH	1 LSLX 1 INH	4 LSL 2 IX1	5 LSL 3 SP1	3 LSL 1 IX	2 PULX 1 INH	1 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	4 EOR 3 IX2	5 EOR 4 SP2	3 EOR 2 IX1	4 EOR 3 SP1	2 EOR 1 IX
9	5 BRCLR4 3 DIR	4 BCLR4 2 DIR	3 BHCS 2 REL	4 ROL 2 DIR	1 ROLA 1 INH	1 ROLX 1 INH	4 ROL 2 IX1	5 ROL 3 SP1	3 ROL 1 IX	2 PSHX 1 INH	1 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	4 ADC 3 IX2	5 ADC 4 SP2	3 ADC 2 IX1	4 ADC 3 SP1	2 ADC 1 IX
A	5 BRSET5 3 DIR	4 BSET5 2 DIR	3 BPL 2 REL	4 DEC 2 DIR	1 DECA 1 INH	1 DECX 1 INH	4 DEC 2 IX1	5 DEC 3 SP1	3 DEC 1 IX	2 PULH 1 INH	2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	4 ORA 3 IX2	5 ORA 4 SP2	3 ORA 2 IX1	4 ORA 3 SP1	2 ORA 1 IX
B	5 BRCLR5 3 DIR	4 BCLR5 2 DIR	3 BMI 2 REL	5 DBNZ 3 DIR	3 DBNZA 2 INH	3 DBNZX 2 INH	5 DBNZ 3 IX1	6 DBNZ 4 SP1	4 DBNZ 2 IX	2 PSHH 1 INH	2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	4 ADD 3 IX2	5 ADD 4 SP2	3 ADD 2 IX1	4 ADD 3 SP1	2 ADD 1 IX
C	5 BRSET6 3 DIR	4 BSET6 2 DIR	3 BMC 2 REL	4 INC 2 DIR	1 INCA 1 INH	1 INCX 1 INH	4 INC 2 IX1	5 INC 3 SP1	3 INC 1 IX	1 CLRH 1 INH	1 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2		3 JMP 2 IX1		2 JMP 1 IX
D	5 BRCLR6 3 DIR	4 BCLR6 2 DIR	3 BMS 2 REL	3 TST 2 DIR	1 TSTA 1 INH	1 TSTX 1 INH	3 TST 2 IX1	4 TST 3 SP1	2 TST 1 IX		1 NOP 1 INH	4 BSR 2 REL	4 JSR 2 DIR	5 JSR 3 EXT	6 JSR 3 IX2		5 JSR 2 IX1		4 JSR 1 IX
E	5 BRSET7 3 DIR	4 BSET7 2 DIR	3 BIL 2 REL		5 MOV 3 DD	4 MOV 2 DIX+	4 MOV 3 IMD		4 MOV 2 IX+D	1 STOP 1 INH	*	2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	4 LDX 3 IX2	5 LDX 4 SP2	3 LDX 2 IX1	4 LDX 3 SP1	2 LDX 1 IX
F	5 BRCLR7 3 DIR	4 BCLR7 2 DIR	3 BIH 2 REL	3 CLR 2 DIR	1 CLRA 1 INH	1 CLRX 1 INH	3 CLR 2 IX1	4 CLR 3 SP1	2 CLR 1 IX	1 WAIT 1 INH	1 TXA 1 INH	2 AIX 2 IMM	3 STX 2 DIR	4 STX 3 EXT	4 STX 3 IX2	5 STX 4 SP2	3 STX 2 IX1	4 STX 3 SP1	2 STX 1 IX

INH Inherent
IMM Immediate
DIR Direct
EXT Extended
DD Direct-Direct
IX+D Indexed-Direct
REL Relative
IX Indexed, No Offset
IX1 Indexed, 8-Bit Offset
IX2 Indexed, 16-Bit Offset
IMD Immediate-Direct
DIX+ Direct-Indexed
SP1 Stack Pointer, 8-Bit Offset
SP2 Stack Pointer, 16-Bit Offset
IX+ Indexed, No Offset with Post Increment
IX1+ Indexed, 1-Byte Offset with Post Increment

*Pre-byte for stack pointer indexed instructions

Low Byte of Opcode in Hexadecimal

MSB / LSB	0	
0	5 BRSET0 3 DIR	Cycles Opcode Mnemonic Number of Bytes / Addressing Mode

High Byte of Opcode in Hexadecimal

Section 7. Oscillator (OSC)

7.1 Contents

7.2 Introduction 102

7.3 Clock Selection 103
7.3.1 CGM Reference Clock Selection 104
7.3.2 TBM Reference Clock Selection 105

7.4 Internal Oscillator 105

7.5 RC Oscillator 106

7.6 X-tal Oscillator 107

7.7 I/O Signals 108
7.7.1 Crystal Amplifier Input Pin (OSC1) 108
7.7.2 Crystal Amplifier Output Pin (OSC2) 109
7.7.3 Oscillator Enable Signal (SIMOSCEN) 109
7.7.4 CGM Oscillator Clock (CGMXCLK) 109
7.7.5 CGM Reference Clock (CGMRCLK) 109
7.7.6 Oscillator Clock to Time Base Module (OSCCLK) 109

7.8 Low Power Modes 109
7.8.1 Wait Mode 110
7.8.2 Stop Mode 110

7.9 Oscillator During Break Mode 110

7.2 Introduction

The oscillator module provides the reference clock for the clock generator module (CGM), the timebase module (TBM), and other MCU sub-systems.

The oscillator module consist of three types of oscillator circuits:

- Internal oscillator
- RC oscillator
- Crystal (x-tal) oscillator

The reference clock for the CGM and other MCU sub-systems is selected by:

- MC68HC908SR12 — FLASH device — oscillator selected by programming the mask option register located at $FF80.
- MC68HC08SR12 — ROM device — oscillator selected by ROM-mask layer at factory.

The reference clock for the timebase module (TBM) is selected by the two bits, OSCCLK1 and OSCCLK0, in the CONFIG2 register.

The internal oscillator runs continuously after a POR or reset, and is always available. The RC and crystal oscillator cannot run concurrently; one is disabled while the other is selected; because the RC and x-tal circuits share the same OSC1 pin.

Figure 7-1. shows the block diagram of the oscillator module.

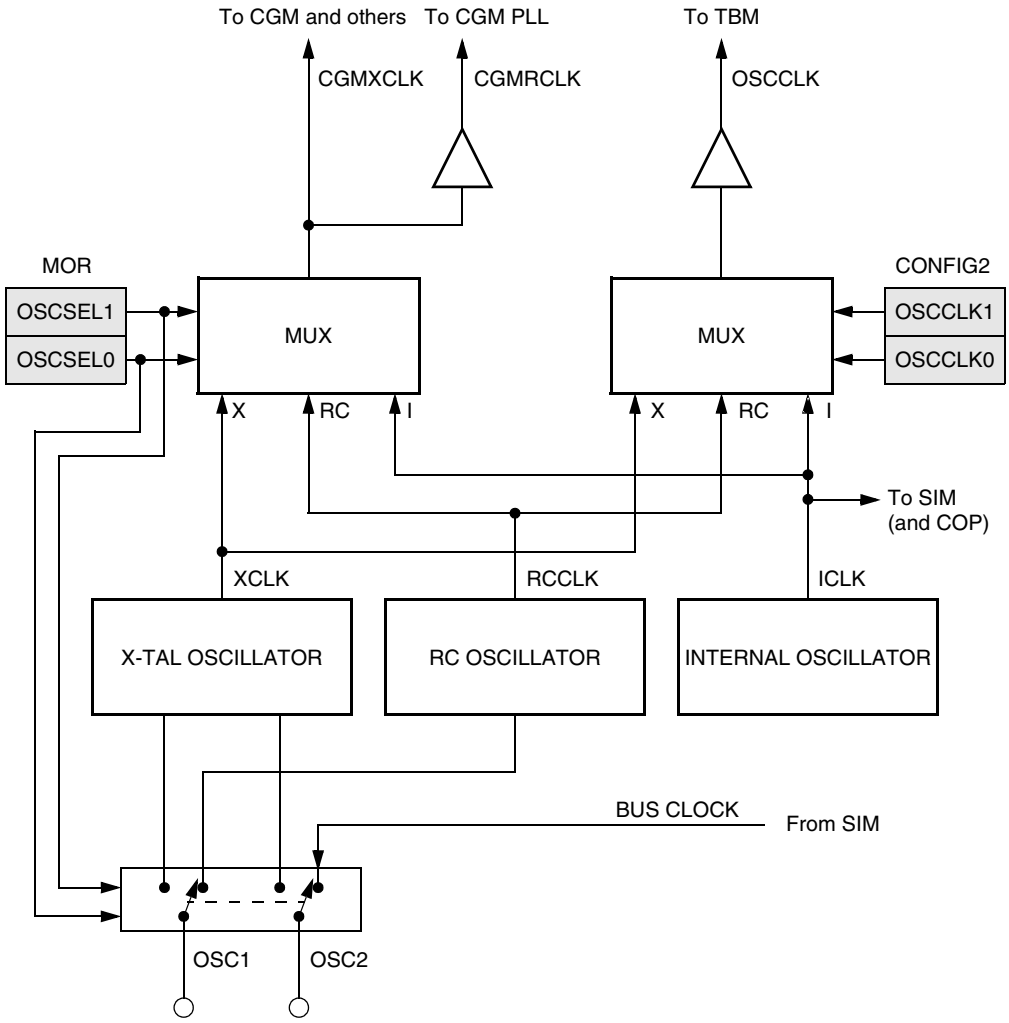


Figure 7-1. Oscillator Module Block Diagram

7.3 Clock Selection

Reference clocks are selectable for the following sub-systems:

- CGMXCLK and CGMRCLK — Reference clock for clock generator module (CGM) and other MCU sub-systems other than TBM and COP. This is the main reference clock for the MCU.
- OSCCLK — Reference clock for timebase module (TBM).

7.3.1 CGM Reference Clock Selection

The clock generator module (CGM) reference clock (CGMXCLK) is the reference clock input to the MCU. It is selected by programming two bits in a FLASH memory location; the mask option register (MOR), at $FF80. See **5.6 Mask Option Register (MOR)**.

NOTE: *On the ROM device, the oscillator is selected by a ROM-mask layer at factory.*

Address: $FF80

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	OSCSEL1	OSCSEL0	R	R	R	R	R	R
Write:								
Erased:	1	1	1	1	1	1	1	1
Reset:	U	U	U	U	U	U	U	U

R = Reserved

Figure 7-2. Mask Option Register (MOR)

Table 7-1. CGMXCLK Clock Selection

OSCSEL1	OSCSEL0	CGMXCLK	OSC2 Pin	Comments
0	0	—	—	Not used
0	1	ICLK	f_{BUS}	Internal oscillator generates the CGMXCLK.
1	0	RCCLK	f_{BUS}	RC oscillator generates the CGMXCLK. Internal oscillator is available after each POR or reset.
1	1	XCLK	Inverting output of X-TAL	X-tal oscillator generates the CGMXCLK. Internal oscillator is available after each POR or reset.

NOTE: *The internal oscillator is a free running oscillator and is available after each POR or reset. It is turned-off in stop mode by clearing the STOP_ICLKEN bit in CONFIG2.*

7.3.2 TBM Reference Clock Selection

The timebase module reference clock (OSCCLK) is selected by configuring two bits in the CONFIG2 register, at $001D. See **5.5 Configuration Register 2 (CONFIG2)**.

Address: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	STOP_ ICLKEN	STOP_ RCLKEN	STOP_ XCLKEN	OSCCLK1	OSCCLK0	0	CDOEN	SCIBDSRC
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 7-3. Configuration Register 2 (CONFIG2)

Table 7-2. Timebase Module Reference Clock Selection

OSCCLK1	OSCCLK0	Timebase Clock Source
0	0	Internal oscillator (ICLK)
0	1	RC oscillator (RCCLK)
1	0	X-tal oscillator (XCLK)
1	1	Not used

NOTE: *The RCCLK or XCLK is only available if that clock is selected as the CGM reference clock, whereas the ICLK is always available.*

7.4 Internal Oscillator

The internal oscillator clock (ICLK) is a free running 24kHz clock that requires no external components. It can be selected as the CGMXCLK for the CGM and MCU sub-systems; and the OSCCLK clock for the TBM. The ICLK is also the reference clock input to the computer operating properly (COP) module.

Due to the simplicity of the internal oscillator, it does not have the accuracy and stability of the RC oscillator or the x-tal oscillator. Therefore, the ICLK is not suitable where an accurate bus clock is required and it should not be used as the CGMRCLK to the CGM PLL.

The internal oscillator by default is always available and is free running after POR or reset. It can be stopped in Stop mode by setting the STOP_ICLKEN bit before executing the STOP instruction.

Figure 7-4 shows the logical representation of components of the internal oscillator circuitry.

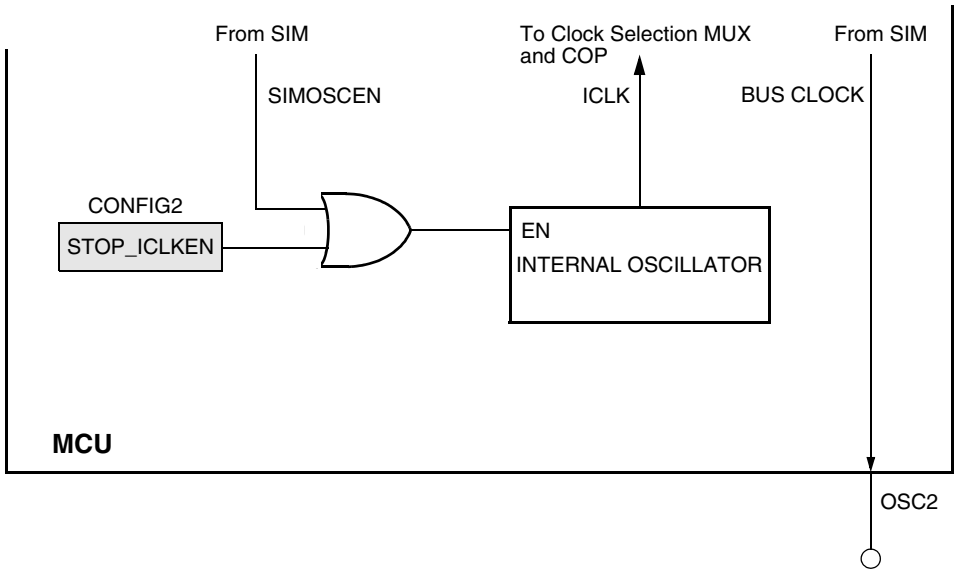


Figure 7-4. Internal Oscillator

7.5 RC Oscillator

The RC oscillator circuit is designed for use with external R and C to provide a clock source with tolerance less than 10%.

In its typical configuration, the RC oscillator requires two external components, one R and one C. Component values should have a tolerance of 1% or less, to obtain a clock source with less than 10% tolerance. The oscillator configuration uses two components:

- C_{EXT}
- R_{EXT}

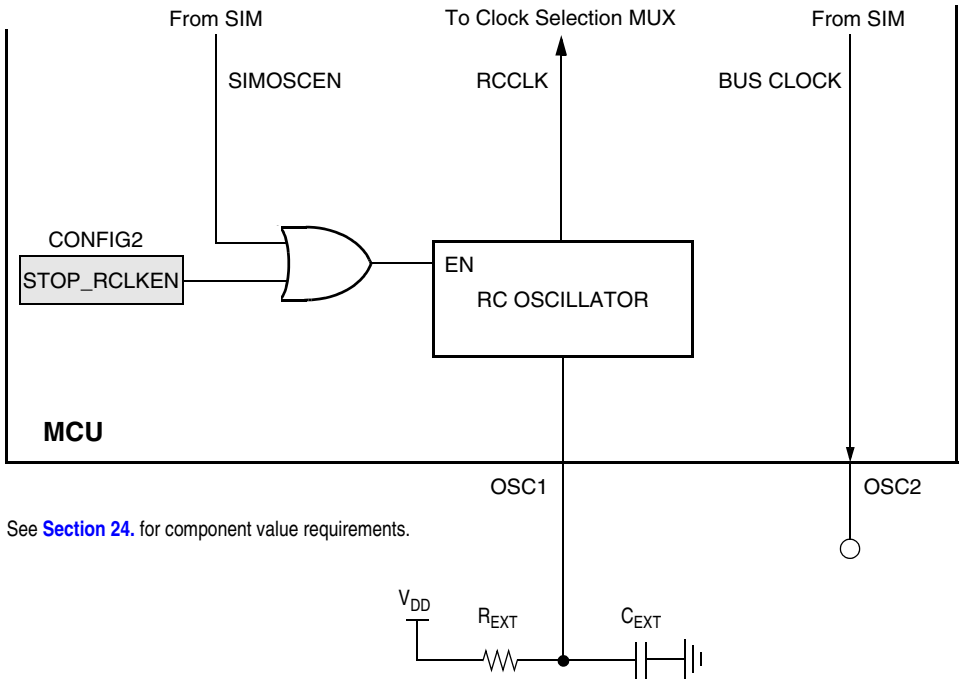


Figure 7-5. RC Oscillator

7.6 X-tal Oscillator

The X-tal oscillator circuit is designed for use with an external crystal or ceramic resonator to provide an accurate clock source.

In its typical configuration, the X-tal oscillator is connected in a Pierce oscillator configuration, as shown in **Figure 7-6**. This figure shows only the logical representation of the internal components and may not represent actual circuitry. The oscillator configuration uses five components:

- Crystal, X_1
- Fixed capacitor, C_1
- Tuning capacitor, C_2 (can also be a fixed capacitor)
- Feedback resistor, R_B
- Series resistor, R_S (optional)

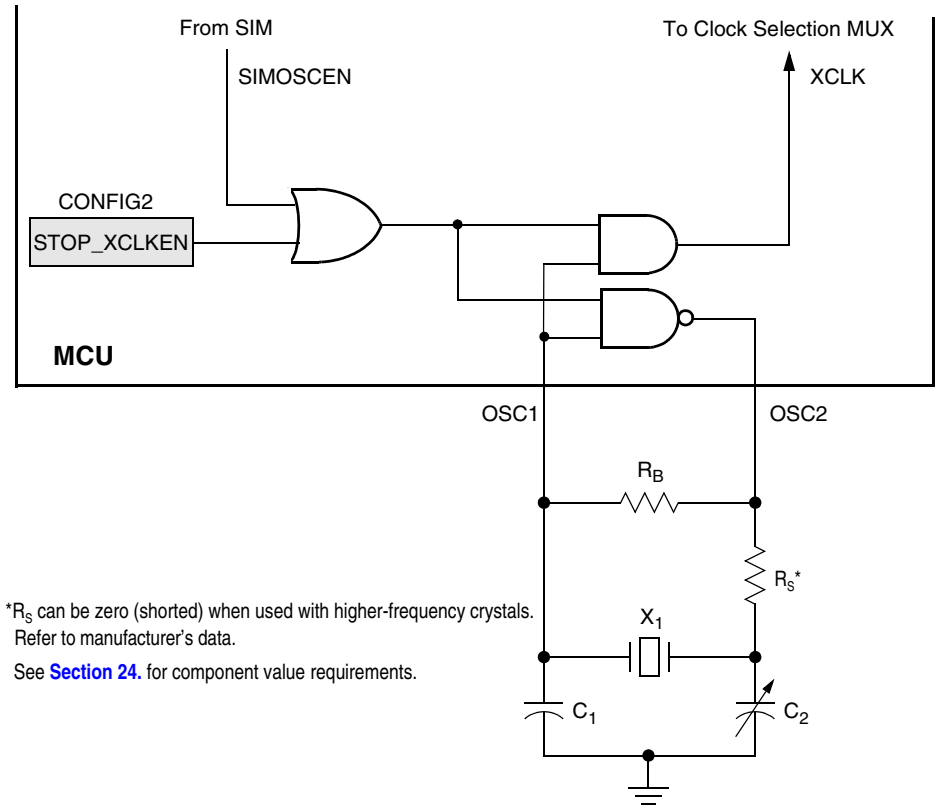


Figure 7-6. Crystal Oscillator

The series resistor (R_S) is included in the diagram to follow strict Pierce oscillator guidelines and may not be required for all ranges of operation, especially with high frequency crystals. Refer to the crystal manufacturer's data for more information.

7.7 I/O Signals

The following paragraphs describe the oscillator I/O signals.

7.7.1 Crystal Amplifier Input Pin (OSC1)

OSC1 pin is an input to the crystal oscillator amplifier or the input to the RC oscillator circuit.

7.7.2 Crystal Amplifier Output Pin (OSC2)

When the x-tal oscillator is selected, OSC2 pin is the output of the crystal oscillator inverting amplifier.

When the RC oscillator or internal oscillator is selected, OSC2 pin is the output of the internal bus clock.

7.7.3 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal from the system integration module (SIM) enables/disables the x-tal oscillator, the RC-oscillator, or the internal oscillator circuit.

7.7.4 CGM Oscillator Clock (CGMXCLK)

The CGMXCLK clock is output from the x-tal oscillator, RC oscillator or the internal oscillator. This clock drives to CGM and other MCU sub-systems.

7.7.5 CGM Reference Clock (CGMRCLK)

This is buffered signal of CGMXCLK, it is used by the CGM as the phase-locked-loop (PLL) reference clock.

7.7.6 Oscillator Clock to Time Base Module (OSCCLK)

The OSCCLK is the reference clock that drives the timebase module. See **Section 12. Timebase Module (TBM)**.

7.8 Low Power Modes

The WAIT and STOP instructions put the MCU in low-power consumption standby modes.

7.8.1 Wait Mode

The WAIT instruction has no effect on the oscillator module. CGMXCLK continues to drive to the clock generator module, and OSCCLK continues to drive the timebase module.

7.8.2 Stop Mode

The STOP instruction disables the x-tal or the RC oscillator circuit, and hence the CGMXCLK clock stops running. For continuous x-tal or RC oscillator operation in stop mode, set the STOP_XCLKEN (for x-tal) or STOP_RCLKEN (for RC) bit to logic 1 before entering stop mode.

The internal oscillator clock continues operation in stop mode. It can be disabled by setting the STOP_ICLKEN bit to logic 1 before entering stop mode.

7.9 Oscillator During Break Mode

The oscillator continues to drive CGMXCLK when the device enters the break state.

Section 8. Clock Generator Module (CGM)

8.1 Contents

8.2 Introduction 112

8.3 Features 113

8.4 Functional Description 113
8.4.1 Oscillator Module 116
8.4.2 Phase-Locked Loop Circuit (PLL) 116
8.4.3 PLL Circuits 116
8.4.4 Acquisition and Tracking Modes 118
8.4.5 Manual and Automatic PLL Bandwidth Modes 118
8.4.6 Programming the PLL 120
8.4.7 Special Programming Exceptions 124
8.4.8 Base Clock Selector Circuit 124
8.4.9 CGM External Connections 125

8.5 I/O Signals 125
8.5.1 External Filter Capacitor Pin (CGMXFC) 126
8.5.2 PLL Analog Power Pin (V_{DDA}) 126
8.5.3 PLL Analog Ground Pin (V_{SSA}) 126
8.5.4 Oscillator Output Frequency Signal (CGMXCLK) 126
8.5.5 CGM Reference Clock (CGMRCLK) 126
8.5.6 CGM VCO Clock Output (CGMVCLK) 127
8.5.7 CGM Base Clock Output (CGMOUT) 127
8.5.8 CGM CPU Interrupt (CGMINT) 127

8.6 CGM Registers 127
8.6.1 PLL Control Register 128
8.6.2 PLL Bandwidth Control Register 130
8.6.3 PLL Multiplier Select Registers 132
8.6.4 PLL VCO Range Select Register 133
8.6.5 PLL Reference Divider Select Register 134

8.7 Interrupts 135

8.8 Special Modes 135
8.8.1 Wait Mode 135
8.8.2 Stop Mode 136
8.8.3 CGM During Break Interrupts 136

8.9 Acquisition/Lock Time Specifications 137
8.9.1 Acquisition/Lock Time Definitions 137
8.9.2 Parametric Influences on Reaction Time 137
8.9.3 Choosing a Filter 139

8.2 Introduction

This section describes the clock generator module (CGM). The CGM generates the base clock signal, CGMOUT, which is based on either the oscillator clock divided by two or the divided phase-locked loop (PLL) clock, CGMPCLK, divided by two. CGMOUT is the clock from which the SIM derives the system clocks, including the bus clock, which is at a frequency of CGMOUT÷2. The PLL clock, CGMVCLK (an integer multiple of CGMPCLK) provides clock reference for the PWM and analog modules.

The PLL is a frequency generator designed for use with a low frequency crystal (typically 32.768kHz) to generate a base frequency and dividing to a maximum bus frequency of 8MHz.

8.3 Features

Features of the CGM include:

- Phase-locked loop with output frequency in integer multiples of an integer dividend of the crystal reference
- Low-frequency crystal operation with low-power operation and high-output frequency resolution
- Programmable prescaler for power-of-two increases in frequency
- Programmable hardware voltage-controlled oscillator (VCO) for low-jitter operation
- Automatic bandwidth control mode for low-jitter operation
- Automatic frequency lock detector
- CPU interrupt on entry or exit from locked condition
- Configuration register bit to allow oscillator operation during stop mode

8.4 Functional Description

The CGM consists of three major sub-modules:

- Oscillator module — The oscillator module generates the constant reference frequency clock, CGMRCLK (buffered CGMXCLK).
- Phase-locked loop (PLL) — The PLL generates the programmable VCO frequency clock, CGMVCLK, and the divided, CGMPCLK. The CGMVCLK provides the input reference clock to the PWM and analog modules.
- Base clock selector circuit — This software-controlled circuit selects either CGMXCLK divided by two or the divided VCO clock, CGMPCLK, divided by two as the base clock, CGMOUT. The SIM derives the system clocks from either CGMOUT or CGMXCLK.

Figure 8-1 shows the structure of the CGM.

Figure 8-2 is a summary of the CGM registers.

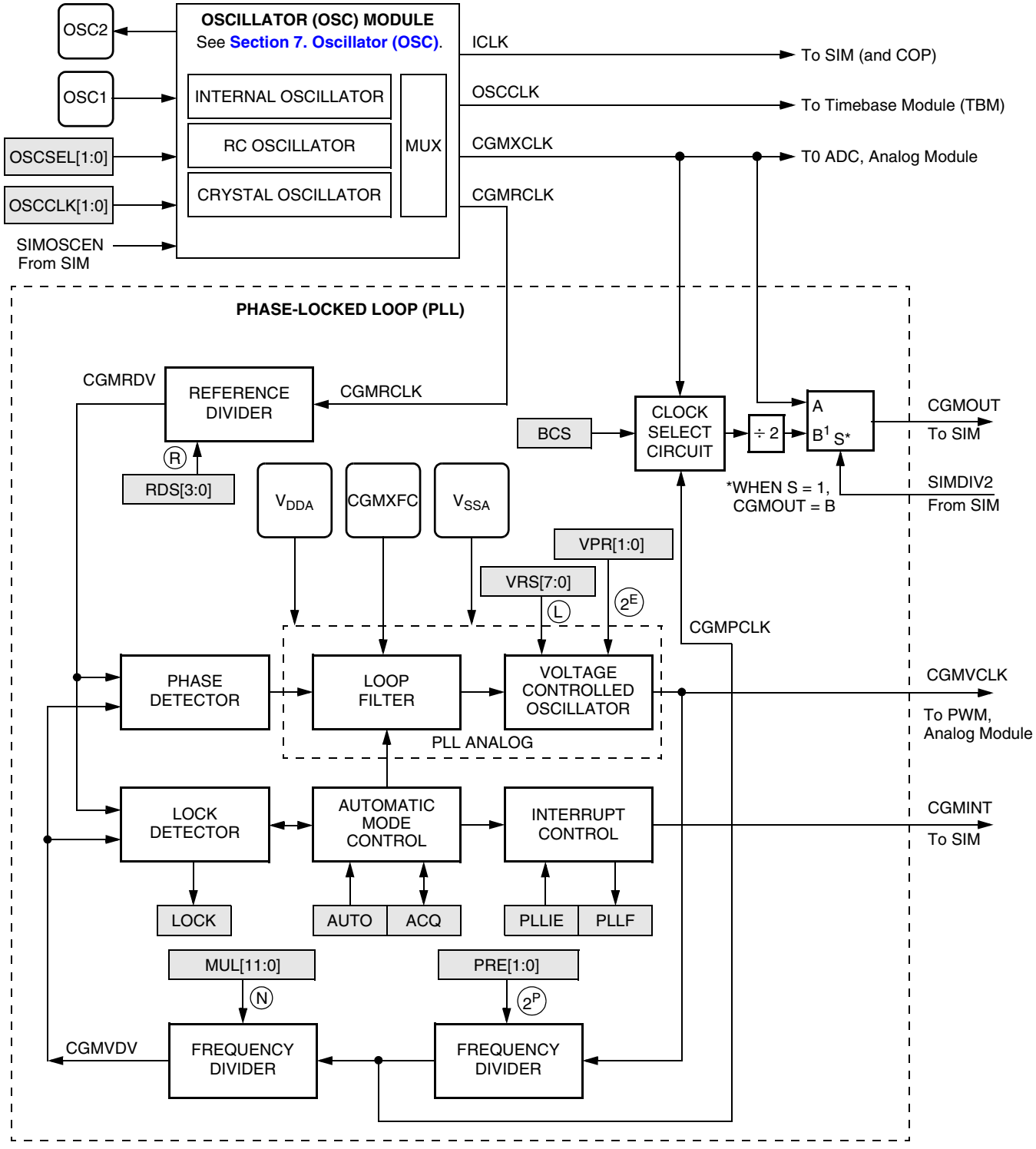


Figure 8-1. CGM Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0036	PLL Control Register (PTCL)	Read:	PLLIE	PLLF	PLLON	BCS	PRE1	PRE0	VPR1	VPR0
		Write:								
		Reset:	0	0	1	0	0	0	0	0
$0037	PLL Bandwidth Control Register (PBWC)	Read:	AUTO	LOCK	$\overline{ACQ}$	0	0	0	0	R
		Write:								
		Reset:	0	0	0	0	0	0	0	
$0038	PLL Multiplier Select Register High (PMSH)	Read:	0	0	0	0	MUL11	MUL10	MUL9	MUL8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0039	PLL Multiplier Select Register Low (PMSL)	Read:	MUL7	MUL6	MUL5	MUL4	MUL3	MUL2	MUL1	MUL0
		Write:								
		Reset:	0	1	0	0	0	0	0	0
$003A	PLL VCO Range Select Register (PMRS)	Read:	VRS7	VRS6	VRS5	VRS4	VRS3	VRS2	VRS1	VRS0
		Write:								
		Reset:	0	1	0	0	0	0	0	0
$003B	PLL Reference Divider Select Register (PMDS)	Read:	0	0	0	0	RDS3	RDS2	RDS1	RDS0
		Write:								
		Reset:	0	0	0	0	0	0	0	1

= Unimplemented R = Reserved

NOTES:
1. When AUTO = 0, PLLIE is forced clear and is read-only.
2. When AUTO = 0, PLLF and LOCK read as clear.
3. When AUTO = 1, $\overline{ACQ}$ is read-only.
4. When PLLON = 0 or VRS7:VRS0 = $0, BCS is forced clear and is read-only.
5. When PLLON = 1, the PLL programming register is read-only.
6. When BCS = 1, PLLON is forced set and is read-only.

Figure 8-2. CGM I/O Register Summary

8.4.1 Oscillator Module

The oscillator module provides two clock outputs CGMXCLK and CGMRCLK to the CGM module. CGMXCLK when selected, is driven to SIM module to generate the system bus clock. CGMRCLK is used by the phase-lock-loop to provide a higher frequency system bus clock and the reference clock for the PWM and analog modules. The oscillator module also provides the reference clock for the timebase module (TBM). See **Section 7. Oscillator (OSC)** for detailed oscillator circuit description. See **Section 12. Timebase Module (TBM)** for detailed description on TBM. See **Section 13. Pulse Width Modulator (PWM)** for detailed description on PWM module.

8.4.2 Phase-Locked Loop Circuit (PLL)

The PLL is a frequency generator that can operate in either acquisition mode or tracking mode, depending on the accuracy of the output frequency. The PLL can change between acquisition and tracking modes either automatically or manually.

8.4.3 PLL Circuits

The PLL consists of these circuits:

- Voltage-controlled oscillator (VCO)
- Reference divider
- Frequency pre-scaler
- Modulo VCO frequency divider
- Phase detector
- Loop filter
- Lock detector

The operating range of the VCO is programmable for a wide range of frequencies and for maximum immunity to external noise, including supply and CGMXFC noise. The VCO frequency is bound to a range from roughly one-half to twice the center-of-range frequency, f_{VRS}. Modulating the voltage on the CGMXFC pin changes the frequency within this range. By design, f_{VRS} is equal to the nominal center-of-range frequency, f_{NOM}, (38.4 kHz) times a linear factor, L, and a power-of-two factor, E, or $(L \times 2^E)f_{NOM}$.

CGMRCLK is the PLL reference clock, a buffered version of CGMXCLK. CGMRCLK runs at a frequency, f_{RCLK}, and is fed to the PLL through a programmable modulo reference divider, which divides f_{RCLK} by a factor, R. The divider's output is the final reference clock, CGMRDV, running at a frequency, $f_{RDV} = f_{RCLK}/R$. With an external crystal (30kHz–100kHz), always set R = 1 for specified performance. With an external high-frequency clock source, use R to divide the external frequency to between 30kHz and 100kHz.

The VCO's output clock, CGMVCLK, running at a frequency, f_{VCLK}, is fed back through a programmable pre-scaler divider and a programmable modulo divider. The pre-scaler divides the VCO clock by a power-of-two factor P (the CGMPCLK) and the modulo divider reduces the VCO clock by a factor, N. The dividers' output is the VCO feedback clock, CGMVDV, running at a frequency, $f_{VDV} = f_{VCLK}/(N \times 2^P)$. (See **8.4.6 Programming the PLL** for more information.)

The phase detector then compares the VCO feedback clock, CGMVDV, with the final reference clock, CGMRDV. A correction pulse is generated based on the phase difference between the two signals. The loop filter then slightly alters the DC voltage on the external capacitor connected to CGMXFC based on the width and direction of the correction pulse. The filter can make fast or slow corrections depending on its mode, described in **8.4.4 Acquisition and Tracking Modes**. The value of the external capacitor and the reference frequency determines the speed of the corrections and the stability of the PLL.

The lock detector compares the frequencies of the VCO feedback clock, CGMVDV, and the final reference clock, CGMRDV. Therefore, the speed of the lock detector is directly proportional to the final reference frequency, f_{RDV}. The circuit determines the mode of the PLL and the lock condition based on this comparison.

8.4.4 Acquisition and Tracking Modes

The PLL filter is manually or automatically configurable into one of two 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 $\overline{\text{ACQ}}$ bit is clear in the PLL bandwidth control register. (See **8.6.2 PLL Bandwidth Control Register**.)
- Tracking mode — 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, such as when the PLL is selected as the base clock source. (See **8.4.8 Base Clock Selector Circuit**.) The PLL is automatically in tracking mode when not in acquisition mode or when the $\overline{\text{ACQ}}$ bit is set.

8.4.5 Manual and Automatic PLL Bandwidth Modes

The PLL can change the bandwidth or operational mode of the loop filter manually or automatically. Automatic mode is recommended for most users.

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 VCO clock, CGMVCLK, is safe to use as the source for the base clock, CGMOUT. (See **8.6.2 PLL Bandwidth Control Register**.) If PLL interrupts are enabled, the software can wait for a PLL interrupt request and then check the LOCK bit. If interrupts are disabled, software can poll the LOCK bit continuously (during PLL start-up, usually) or at periodic intervals. In either case, when the LOCK bit is set, the VCO clock is safe to use as the source for the base clock. (See **8.4.8 Base Clock Selector Circuit**.) If the VCO is selected as the source for the base clock 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. (See **8.7 Interrupts** for information and precautions on using interrupts.)

The following conditions apply when the PLL is in automatic bandwidth control mode:

- The $\overline{ACQ}$ bit (See **8.6.2 PLL Bandwidth Control Register**.) is a read-only indicator of the mode of the filter. (See **8.4.4 Acquisition and Tracking Modes**.)
- The $\overline{ACQ}$ bit is set when the VCO frequency is within a certain tolerance and is cleared when the VCO frequency is out of a certain tolerance. (See **8.9 Acquisition/Lock Time Specifications** for more information.)
- The LOCK bit is a read-only indicator of the locked state of the PLL.
- The LOCK bit is set when the VCO frequency is within a certain tolerance and is cleared when the VCO frequency is out of a certain tolerance. (See **8.9 Acquisition/Lock Time Specifications** for more information.)
- CPU interrupts can occur if enabled (PLLIE = 1) when the PLL's lock condition changes, toggling the LOCK bit. (See **8.6.1 PLL Control Register**.)

The PLL also may 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 f_{BUSMAX}.

The following conditions apply when in manual mode:

- $\overline{ACQ}$ is a writable control bit that controls the mode of the filter. Before turning on the PLL in manual mode, the $\overline{ACQ}$ bit must be clear.
- Before entering tracking mode ($\overline{ACQ}$ = 1), software must wait a given time, t_{ACQ} (See **8.9 Acquisition/Lock Time Specifications**.), after turning on the PLL by setting PLLON in the PLL control register (PCTL).
- Software must wait a given time, t_{AL}, after entering tracking mode before selecting the PLL as the clock source to CGMOUT (BCS = 1).
- The LOCK bit is disabled.
- CPU interrupts from the CGM are disabled.

8.4.6 Programming the PLL

The following procedure shows how to program the PLL.

NOTE: *The round function in the following equations means that the real number should be rounded to the nearest integer number.*

1. Choose the desired bus frequency, f_{BUSDES}, or the desired VCO frequency, $f_{VCLKDES}$; and then solve for the other.
   The relationship between f_{BUS} and f_{VCLK} is governed by the equation:

   $$f_{VCLK} = 2^P \times f_{CGMPCLK} = 2^P \times 4 \times f_{BUS}$$

   where P is the power of two multiplier, and can be 0, 1, 2, or 3
2. Choose a practical PLL reference frequency, f_{RCLK}, and the reference clock divider, R. Typically, the reference is 32.768kHz and R = 1.

   Frequency errors to the PLL are corrected at a rate of f_{RCLK}/R. For stability and lock time reduction, this rate must be as fast as possible. The VCO frequency must be an integer multiple of this rate.

The relationship between the VCO frequency, f_{VCLK}, and the reference frequency, f_{RCLK}, is

$$f_{VCLK} = \frac{2^P N}{R}(f_{RCLK})$$

where N is the integer range multiplier, between 1 and 4095.

In cases where desired bus frequency has some tolerance, choose f_{RCLK} to a value determined either by other module requirements (such as modules which are clocked by CGMXCLK), cost requirements, or ideally, as high as the specified range allows. See **Section 24. Electrical Specifications**. Choose the reference divider, R = 1.

When the tolerance on the bus frequency is tight, choose f_{RCLK} to an integer divisor of f_{BUSDES}, and R = 1. If f_{RCLK} cannot meet this requirement, use the following equation to solve for R with practical choices of f_{RCLK}, and choose the f_{RCLK} that gives the lowest R.

$$R = \text{round}\left[R_{MAX} \times \left\{\left(\frac{f_{VCLKDES}}{f_{RCLK}}\right) - \text{integer}\left(\frac{f_{VCLKDES}}{f_{RCLK}}\right)\right\}\right]$$

3. Calculate N:

$$N = \text{round}\left(\frac{R \times f_{VCLKDES}}{f_{RCLK} \times 2^P}\right)$$

4. Calculate and verify the adequacy of the VCO and bus frequencies f_{VCLK} and f_{BUS}.

$$f_{VCLK} = \frac{2^P N}{R}(f_{RCLK})$$

$$f_{BUS} = \frac{f_{VCLK}}{2^P \times 4}$$

5. Select the VCO's power-of-two range multiplier E, according to this table:

Frequency Range	E
$0 < f_{VCLK} < 9{,}830{,}400$	0
$9{,}830{,}400 \leq f_{VCLK} < 19{,}660{,}800$	1
$19{,}660{,}800 \leq f_{VCLK} < 39{,}321{,}600$	2

NOTE: Do not program E to a value of 3.

6. Select a VCO linear range multiplier, L, where f_{NOM} = 38.4kHz

$$L = \text{round}\left(\frac{f_{VCLK}}{2^E \times f_{NOM}}\right)$$

7. Calculate and verify the adequacy of the VCO programmed center-of-range frequency, f_{VRS}. The center-of-range frequency is the midpoint between the minimum and maximum frequencies attainable by the PLL.

$$f_{VRS} = (L \times 2^E) f_{NOM}$$

For proper operation,

$$|f_{VRS} - f_{VCLK}| \leq \frac{f_{NOM} \times 2^E}{2}$$

8. Verify the choice of P, R, N, E, and L by comparing f_{VCLK} to f_{VRS} and $f_{VCLKDES}$. For proper operation, f_{VCLK} must be within the application's tolerance of $f_{VCLKDES}$, and f_{VRS} must be as close as possible to f_{VCLK}.

NOTE: *Exceeding the recommended maximum bus frequency or VCO frequency can crash the MCU.*

9. Program the PLL registers accordingly:
   a. In the PRE bits of the PLL control register (PCTL), program the binary equivalent of P.
   b. In the VPR bits of the PLL control register (PCTL), program the binary equivalent of E.
   c. In the PLL multiplier select register low (PMSL) and the PLL multiplier select register high (PMSH), program the binary equivalent of N.
   d. In the PLL VCO range select register (PMRS), program the binary coded equivalent of L.
   e. In the PLL reference divider select register (PMDS), program the binary coded equivalent of R.

NOTE: *The values for P, E, N, L, and R can only be programmed when the PLL is off (PLLON = 0).*

Table 8-1 provides numeric examples (numbers are in hexadecimal notation):

Table 8-1. Numeric Examples

CGMVCLK	CGMPCLK	f_{BUS}	f_{RCLK}	R	N	P	E	L
8.0 MHz	8.0 MHz	2.0 MHz	32.768 kHz	1	F5	0	0	D1
9.8304 MHz	9.8304 MHz	2.4576 MHz	32.768 kHz	1	12C	0	1	80
10.0 MHz	10.0 MHz	2.5 MHz	32.768 kHz	1	132	0	1	83
16 MHz	16 MHz	4.0 MHz	32.768 kHz	1	1E9	0	1	D1
19.6608 MHz	19.6608 MHz	4.9152 MHz	32.768 kHz	1	258	0	2	80
20 MHz	20 MHz	5.0 MHz	32.768 kHz	1	263	0	2	82
29.4912 MHz	29.4912 MHz	7.3728 MHz	32.768 kHz	1	384	0	2	C0
32 MHz	32 MHz	8.0 MHz	32.768 kHz	1	3D1	0	2	D0
32 MHz	16 MHz	4.0 MHz	32.768 kHz	1	1E9	1	2	D0
32 MHz	8 MHz	2.0 MHz	32.768 kHz	1	F5	2	2	D0
32 MHz	4 MHz	1.0 MHz	32.768 kHz	1	7B	3	2	D0

8.4.7 Special Programming Exceptions

The programming method described in **8.4.6 Programming the PLL** does not account for three possible exceptions. A value of 0 for R, N, or L is meaningless when used in the equations given. To account for these exceptions:

- A 0 value for R or N is interpreted exactly the same as a value of 1.
- A 0 value for L disables the PLL and prevents its selection as the source for the base clock.

(See **8.4.8 Base Clock Selector Circuit**.)

8.4.8 Base Clock Selector Circuit

This circuit is used to select either the oscillator clock, CGMXCLK, or the divided VCO clock, CGMPCLK, as the source of the base clock, CGMOUT. The two input clocks go through a transition control circuit that waits up to three CGMXCLK cycles and three CGMPCLK cycles to change from one clock source to the other. During this time, CGMOUT is held in stasis. The output of the transition control circuit is then divided by two to correct the duty cycle. Therefore, the bus clock frequency, which is one-half of the base clock frequency, is one-fourth the frequency of the selected clock (CGMXCLK or CGMPCLK).

The BCS bit in the PLL control register (PCTL) selects which clock drives CGMOUT. The divided VCO clock cannot be selected as the base clock source if the PLL is not turned on. The PLL cannot be turned off if the divided VCO clock is selected. The PLL cannot be turned on or off simultaneously with the selection or deselection of the divided VCO clock. The divided VCO clock also cannot be selected as the base clock source if the factor L is programmed to a 0. This value would set up a condition inconsistent with the operation of the PLL, so that the PLL would be disabled and the oscillator clock would be forced as the source of the base clock.

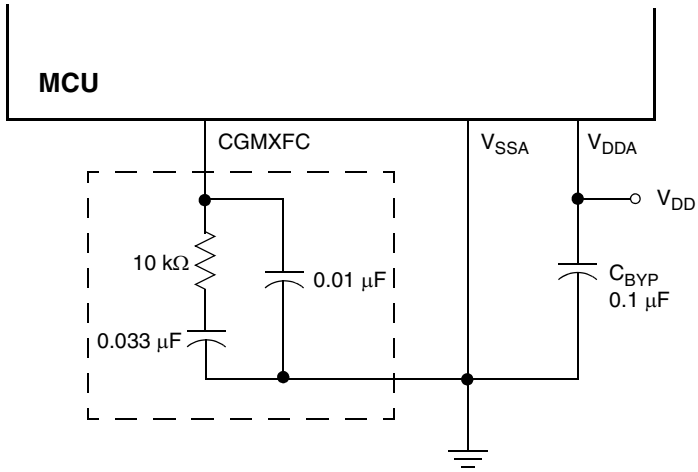
8.4.9 CGM External Connections

In its typical configuration, the CGM requires up to four external components.

Figure 8-3 shows the external components for the PLL:

- Bypass capacitor, C_{BYP}
- Filter network

Care should be taken with PCB routing in order to minimize signal cross talk and noise. (See **8.9 Acquisition/Lock Time Specifications** for routing information, filter network and its effects on PLL performance.)



Note: Filter network in box can be replaced with a 0.47µF capacitor, but will degrade stability.

Figure 8-3. CGM External Connections

8.5 I/O Signals

The following paragraphs describe the CGM I/O signals.

8.5.1 External Filter Capacitor Pin (CGMXFC)

The CGMXFC pin is required by the loop filter to filter out phase corrections. An external filter network is connected to this pin. (See **Figure 8-3**.)

NOTE: *To prevent noise problems, the filter network should be placed as close to the CGMXFC pin as possible, with minimum routing distances and no routing of other signals across the network.*

8.5.2 PLL Analog Power Pin (V_{DDA})

V_{DDA} is a power pin used by the analog portions of the PLL. Connect the V_{DDA} pin to the same voltage potential as the V_{DD} pin.

NOTE: *Route V_{DDA} carefully for maximum noise immunity and place bypass capacitors as close as possible to the package.*

8.5.3 PLL Analog Ground Pin (V_{SSA})

V_{SSA} is a ground pin used by the analog portions of the PLL. Connect the V_{SSA} pin to the same voltage potential as the V_{SS} pin.

NOTE: *Route V_{SSA} carefully for maximum noise immunity and place bypass capacitors as close as possible to the package.*

8.5.4 Oscillator Output Frequency Signal (CGMXCLK)

CGMXCLK is the oscillator output signal. It runs at the full speed of the oscillator, and is generated directly from the crystal oscillator circuit, the RC oscillator circuit, or the internal oscillator circuit.

8.5.5 CGM Reference Clock (CGMRCLK)

CGMRCLK is a buffered version of CGMXCLK, this clock is the reference clock for the phase-locked-loop circuit.

8.5.6 CGM VCO Clock Output (CGMVCLK)

CGMVCLK is the clock output from the VCO. This clock can be used by the pulse width modulator (PWM) module to generate high frequency PWM signals. This clock is also used by the analog module as a reference for signal sampling.

8.5.7 CGM Base Clock Output (CGMOUT)

CGMOUT is the clock output of the CGM. This signal goes to the SIM, which generates the MCU clocks. CGMOUT is a 50 percent duty cycle clock running at twice the bus frequency. CGMOUT is software programmable to be either the oscillator output, CGMXCLK, divided by two or the divided VCO clock, CGMPCLK, divided by two.

8.5.8 CGM CPU Interrupt (CGMINT)

CGMINT is the interrupt signal generated by the PLL lock detector.

8.6 CGM Registers

The following registers control and monitor operation of the CGM:

- PLL control register (PCTL)
  (See **8.6.1 PLL Control Register**.)
- PLL bandwidth control register (PBWC)
  (See **8.6.2 PLL Bandwidth Control Register**.)
- PLL multiplier select registers (PMSH and PMSL)
  (See **8.6.3 PLL Multiplier Select Registers**.)
- PLL VCO range select register (PMRS)
  (See **8.6.4 PLL VCO Range Select Register**.)
- PLL reference divider select register (PMDS)
  (See **8.6.5 PLL Reference Divider Select Register**.)

8.6.1 PLL Control Register

The PLL control register (PCTL) contains the interrupt enable and flag bits, the on/off switch, the base clock selector bit, the prescaler bits, and the VCO power-of-two range selector bits.

Address: $0036

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PLLIE	PLLF	PLLON	BCS	PRE1	PRE0	VPR1	VPR0
Write:								
Reset:	0	0	1	0	0	0	0	0

= Unimplemented

Figure 8-4. PLL Control Register (PCTL)

PLLIE — PLL Interrupt Enable Bit

This read/write bit enables the PLL to generate an interrupt request when the LOCK bit toggles, setting the PLL flag, PLLF. When the AUTO bit in the PLL bandwidth control register (PBWC) is clear, PLLIE cannot be written and reads as logic 0. Reset clears the PLLIE bit.

1 = PLL interrupts enabled
0 = PLL interrupts disabled

PLLF — PLL Interrupt Flag Bit

This read-only bit is set whenever the LOCK bit toggles. PLLF generates an interrupt request if the PLLIE bit also is set. PLLF always reads as logic 0 when the AUTO bit in the PLL bandwidth control register (PBWC) is clear. Clear the PLLF bit by reading the PLL control register. Reset clears the PLLF bit.

1 = Change in lock condition
0 = No change in lock condition

NOTE: *Do not inadvertently clear the PLLF bit. Any read or read-modify-write operation on the PLL control register clears the PLLF bit.*

PLLON — PLL On Bit

This read/write bit activates the PLL and enables the VCO clock, CGMVCLK. PLLON cannot be cleared if the VCO clock is driving the base clock, CGMOUT (BCS = 1). (See **8.4.8 Base Clock Selector Circuit**.) Reset sets this bit so that the loop can stabilize as the MCU is powering up.

1 = PLL on
0 = PLL off

BCS — Base Clock Select Bit

This read/write bit selects either the oscillator output, CGMXCLK, or the divided VCO clock, CGMPCLK, as the source of the CGM output, CGMOUT. CGMOUT frequency is one-half the frequency of the selected clock. BCS cannot be set while the PLLON bit is clear. After toggling BCS, it may take up to three CGMXCLK and three CGMPCLK cycles to complete the transition from one source clock to the other. During the transition, CGMOUT is held in stasis. (See **8.4.8 Base Clock Selector Circuit**.) Reset clears the BCS bit.

1 = CGMPCLK divided by two drives CGMOUT
0 = CGMXCLK divided by two drives CGMOUT

NOTE: *PLLON and BCS have built-in protection that prevents the base clock selector circuit from selecting the VCO clock as the source of the base clock if the PLL is off. Therefore, PLLON cannot be cleared when BCS is set, and BCS cannot be set when PLLON is clear. If the PLL is off (PLLON = 0), selecting CGMPCLK requires two writes to the PLL control register. (See **8.4.8 Base Clock Selector Circuit**.)*

PRE1 and PRE0 — Prescaler Program Bits

These read/write bits control a prescaler that selects the prescaler power-of-two multiplier, P. (See **8.4.3 PLL Circuits** and **8.4.6 Programming the PLL**.) PRE1 and PRE0 cannot be written when the PLLON bit is set. Reset clears these bits.

These prescaler bits affects the relationship between the VCO clock and the final system bus clock.

Table 8-2. PRE1 and PRE0 Programming

PRE1 and PRE0	P	Prescaler Multiplier
00	0	1
01	1	2
10	2	4
11	3	8

VPR1 and VPR0 — VCO Power-of-Two Range Select Bits

These read/write bits control the VCO's hardware power-of-two range multiplier E that, in conjunction with L (See **8.4.3 PLL Circuits**, **8.4.6 Programming the PLL**, and **8.6.4 PLL VCO Range Select Register**.) controls the hardware center-of-range frequency, f_{VRS}. VPR1:VPR0 cannot be written when the PLLON bit is set. Reset clears these bits.

Table 8-3. VPR1 and VPR0 Programming

VPR1 and VPR0	E	VCO Power-of-Two Range Multiplier
00	0	1
01	1	2
10	2	4

NOTE: Do not program E to a value of 3.

8.6.2 PLL Bandwidth Control Register

The PLL bandwidth control register (PBWC):

- Selects automatic or manual (software-controlled) bandwidth control mode
- Indicates when the PLL is locked
- In automatic bandwidth control mode, indicates when the PLL is in acquisition or tracking mode
- In manual operation, forces the PLL into acquisition or tracking mode

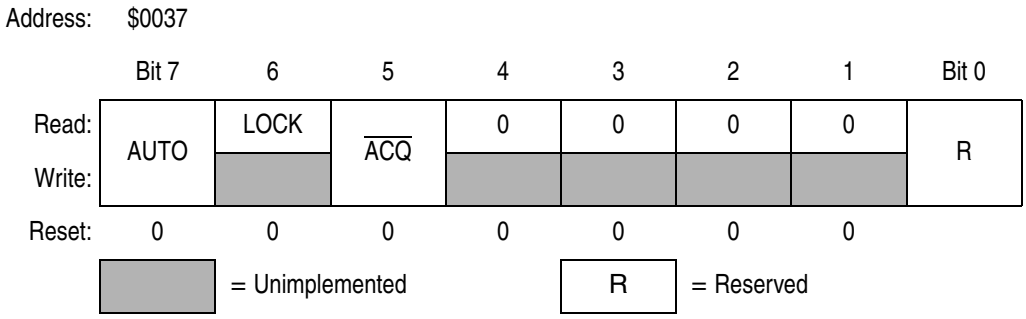


Figure 8-5. PLL Bandwidth Control Register (PBWCR)

AUTO — Automatic Bandwidth Control Bit

This read/write bit selects automatic or manual bandwidth control. When initializing the PLL for manual operation (AUTO = 0), clear the $\overline{ACQ}$ bit before turning on the PLL. Reset clears the AUTO bit.

1 = Automatic bandwidth control
0 = Manual bandwidth control

LOCK — Lock Indicator Bit

When the AUTO bit is set, LOCK is a read-only bit that becomes set when the VCO clock, CGMVCLK, is locked (running at the programmed frequency). When the AUTO bit is clear, LOCK reads as logic 0 and has no meaning. The write one function of this bit is reserved for test, so this bit must *always* be written a 0. Reset clears the LOCK bit.

1 = VCO frequency correct or locked
0 = VCO frequency incorrect or unlocked

$\overline{ACQ}$ — Acquisition Mode Bit

When the AUTO bit is set, $\overline{ACQ}$ is a read-only bit that indicates whether the PLL is in acquisition mode or tracking mode. When the AUTO bit is clear, $\overline{ACQ}$ is a read/write bit that controls whether the PLL is in acquisition or tracking mode.

In automatic bandwidth control mode (AUTO = 1), the last-written value from manual operation is stored in a temporary location and is recovered when manual operation resumes. Reset clears this bit, enabling acquisition mode.

1 = Tracking mode
0 = Acquisition mode

8.6.3 PLL Multiplier Select Registers

The PLL multiplier select registers (PMSH and PMSL) contain the programming information for the modulo feedback divider.

Address: $0038

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	MUL11	MUL10	MUL9	MUL8
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 8-6. PLL Multiplier Select Register High (PMSH)

Address: $0039

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MUL7	MUL6	MUL5	MUL4	MUL3	MUL2	MUL1	MUL0
Write:								
Reset:	0	1	0	0	0	0	0	0

Figure 8-7. PLL Multiplier Select Register Low (PMSL)

MUL[11:0] — Multiplier Select Bits

These read/write bits control the modulo feedback divider that selects the VCO frequency multiplier N. (See **8.4.3 PLL Circuits** and **8.4.6 Programming the PLL**.) A value of $0000 in the multiplier select registers configure the modulo feedback divider the same as a value of $0001. Reset initializes the registers to $0040 for a default multiply value of 64.

NOTE: *The multiplier select bits have built-in protection such that they cannot be written when the PLL is on (PLLON = 1).*

8.6.4 PLL VCO Range Select Register

The PLL VCO range select register (PMRS) contains the programming information required for the hardware configuration of the VCO.

Address: $003A

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	VRS7	VRS6	VRS5	VRS4	VRS3	VRS2	VRS1	VRS0
Reset:	0	1	0	0	0	0	0	0

Figure 8-8. PLL VCO Range Select Register (PMRS)

VRS[7:0] — VCO Range Select Bits

These read/write bits control the hardware center-of-range linear multiplier L which, in conjunction with E (See **8.4.3 PLL Circuits**, **8.4.6 Programming the PLL**, and **8.6.1 PLL Control Register**.), controls the hardware center-of-range frequency, f_{VRS}. VRS[7:0] cannot be written when the PLLON bit in the PCTL is set. (See **8.4.7 Special Programming Exceptions**.) A value of $00 in the VCO range select register disables the PLL and clears the BCS bit in the PLL control register (PCTL). (See **8.4.8 Base Clock Selector Circuit** and **8.4.7 Special Programming Exceptions**.). Reset initializes the register to $40 for a default range multiply value of 64.

NOTE: *The VCO range select bits have built-in protection such that they cannot be written when the PLL is on (PLLON = 1) and such that the VCO clock cannot be selected as the source of the base clock (BCS = 1) if the VCO range select bits are all clear.*

The PLL VCO range select register must be programmed correctly. Incorrect programming can result in failure of the PLL to achieve lock.

8.6.5 PLL Reference Divider Select Register

The PLL reference divider select register (PMDS) contains the programming information for the modulo reference divider.

Address: $003B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	RDS3	RDS2	RDS1	RDS0
Write:								
Reset:	0	0	0	0	0	0	0	1

= Unimplemented

Figure 8-9. PLL Reference Divider Select Register (PMDS)

RDS[3:0] — Reference Divider Select Bits

These read/write bits control the modulo reference divider that selects the reference division factor, R. (See **8.4.3 PLL Circuits** and **8.4.6 Programming the PLL**.) RDS[3:0] cannot be written when the PLLON bit in the PCTL is set. A value of $00 in the reference divider select register configures the reference divider the same as a value of $01. (See **8.4.7 Special Programming Exceptions**.) Reset initializes the register to $01 for a default divide value of 1.

NOTE: *The reference divider select bits have built-in protection such that they cannot be written when the PLL is on (PLLON = 1).*

NOTE: *The default divide value of 1 is recommended for all applications.*

8.7 Interrupts

When the AUTO bit is set in the PLL bandwidth control register (PBWC), the PLL can generate a CPU interrupt request every time the LOCK bit changes state. The PLLIE bit in the PLL control register (PCTL) enables CPU interrupts from the PLL. PLLF, the interrupt flag in the PCTL, becomes set whether interrupts are enabled or not. When the AUTO bit is clear, CPU interrupts from the PLL are disabled and PLLF reads as logic 0.

Software should read the LOCK bit after a PLL interrupt request to see if the request was due to an entry into lock or an exit from lock. When the PLL enters lock, the divided VCO clock, CGMPCLK, divided by two can be selected as the CGMOUT source by setting BCS in the PCTL. When the PLL exits lock, the VCO clock frequency is corrupt, and appropriate precautions should be taken. If the application is not frequency sensitive, interrupts should be disabled to prevent PLL interrupt service routines from impeding software performance or from exceeding stack limitations.

NOTE: *Software can select the CGMPCLK divided by two as the CGMOUT source even if the PLL is not locked (LOCK = 0). Therefore, software should make sure the PLL is locked before setting the BCS bit.*

8.8 Special Modes

The WAIT instruction puts the MCU in low power-consumption standby modes.

8.8.1 Wait Mode

The WAIT instruction does not affect the CGM. Before entering wait mode, software can disengage and turn off the PLL by clearing the BCS and PLLON bits in the PLL control register (PCTL) to save power. Less power-sensitive applications can disengage the PLL without turning it off, so that the PLL clock is immediately available at WAIT exit. This would be the case also when the PLL is to wake the MCU from wait mode, such as when the PLL is first enabled and waiting for LOCK or LOCK is lost.

8.8.2 Stop Mode

If the oscillator stop mode enable bit (STOP_ICLKEN, STOP_RCLKEN, or STOP_XCLKEN in CONFIG2 register) for the selected oscillator is configured to disabled the oscillator in stop mode, then the STOP instruction disables the CGM (oscillator and phase locked loop) and holds low all CGM outputs (CGMOUT, CGMVCLK, CGMPCLK, and CGMINT).

If the STOP instruction is executed with the divided VCO clock, CGMPCLK, divided by two driving CGMOUT, the PLL automatically clears the BCS bit in the PLL control register (PCTL), thereby selecting the oscillator clock, CGMXCLK, divided by two as the source of CGMOUT. When the MCU recovers from STOP, the crystal clock divided by two drives CGMOUT and BCS remains clear.

If the oscillator stop mode enable bit is configured for continuous oscillator operation in stop mode, then the phase locked loop is shut off but the CGMXCLK will continue to drive the SIM and other MCU sub-systems.

8.8.3 CGM During Break Interrupts

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state. (See **9.8.3 SIM Break Flag Control Register**.)

To allow software to clear status bits during a break interrupt, write a logic 1 to the BCFE bit. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect the PLLF bit during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), software can read and write the PLL control register during the break state without affecting the PLLF bit.

8.9 Acquisition/Lock Time Specifications

The acquisition and lock times of the PLL are, in many applications, the most critical PLL design parameters. Proper design and use of the PLL ensures the highest stability and lowest acquisition/lock times.

8.9.1 Acquisition/Lock Time Definitions

Typical control systems refer to the acquisition time or lock time as the reaction time, within specified tolerances, of the system to a step input. In a PLL, the step input occurs when the PLL is turned on or when it suffers a noise hit. The tolerance is usually specified as a percent of the step input or when the output settles to the desired value plus or minus a percent of the frequency change. Therefore, the reaction time is constant in this definition, regardless of the size of the step input. For example, consider a system with a 5 percent acquisition time tolerance. If a command instructs the system to change from 0Hz to 1MHz, the acquisition time is the time taken for the frequency to reach 1MHz ±50kHz. 50kHz = 5% of the 1MHz step input. If the system is operating at 1MHz and suffers a –100kHz noise hit, the acquisition time is the time taken to return from 900kHz to 1MHz ±5kHz. 5kHz = 5% of the 100kHz step input.

Other systems refer to acquisition and lock times as the time the system takes to reduce the error between the actual output and the desired output to within specified tolerances. Therefore, the acquisition or lock time varies according to the original error in the output. Minor errors may not even be registered. Typical PLL applications prefer to use this definition because the system requires the output frequency to be within a certain tolerance of the desired frequency regardless of the size of the initial error.

8.9.2 Parametric Influences on Reaction Time

Acquisition and lock times are designed to be as short as possible while still providing the highest possible stability. These reaction times are not constant, however. Many factors directly and indirectly affect the acquisition time.

The most critical parameter which affects the reaction times of the PLL is the reference frequency, f_{RDV}. This frequency is the input to the phase detector and controls how often the PLL makes corrections. For stability, the corrections must be small compared to the desired frequency, so several corrections are required to reduce the frequency error. Therefore, the slower the reference the longer it takes to make these corrections. This parameter is under user control via the choice of crystal frequency f_{XCLK} and the R value programmed in the reference divider. (See **8.4.3 PLL Circuits**, **8.4.6 Programming the PLL**, and **8.6.5 PLL Reference Divider Select Register**.)

Another critical parameter is the external filter network. The PLL modifies the voltage on the VCO by adding or subtracting charge from capacitors in this network. Therefore, the rate at which the voltage changes for a given frequency error (thus change in charge) is proportional to the capacitance. The size of the capacitor also is related to the stability of the PLL. If the capacitor is too small, the PLL cannot make small enough adjustments to the voltage and the system cannot lock. If the capacitor is too large, the PLL may not be able to adjust the voltage in a reasonable time. (See **8.9.3 Choosing a Filter**.)

Also important is the operating voltage potential applied to V_{DDA}. The power supply potential alters the characteristics of the PLL. A fixed value is best. Variable supplies, such as batteries, are acceptable if they vary within a known range at very slow speeds. Noise on the power supply is not acceptable, because it causes small frequency errors which continually change the acquisition time of the PLL.

Temperature and processing also can affect acquisition time because the electrical characteristics of the PLL change. The part operates as specified as long as these influences stay within the specified limits. External factors, however, can cause drastic changes in the operation of the PLL. These factors include noise injected into the PLL through the filter capacitor, filter capacitor leakage, stray impedances on the circuit board, and even humidity or circuit board contamination.

8.9.3 Choosing a Filter

As described in **8.9.2 Parametric Influences on Reaction Time**, the external filter network is critical to the stability and reaction time of the PLL. The PLL is also dependent on reference frequency and supply voltage.

Either of the filter networks in **Figure 8-10** is recommended when using a 32.768kHz reference clock (CGMRCLK). **Figure 8-10 (a)** is used for applications requiring better stability. **Figure 8-10 (b)** is used in low-cost applications where stability is not critical.

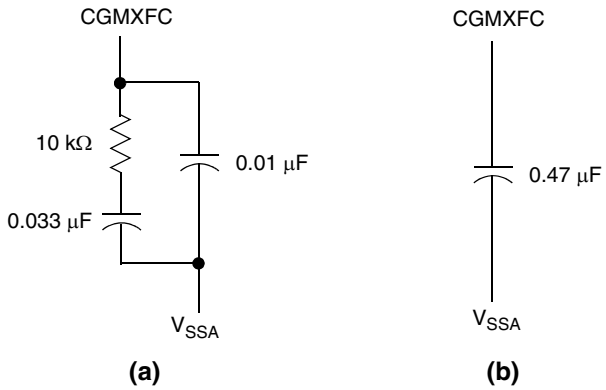


Figure 8-10. PLL Filter

Section 9. System Integration Module (SIM)

9.1 Contents

9.2 Introduction 142

9.3 SIM Bus Clock Control and Generation 144
9.3.1 Bus Timing 145
9.3.2 Clock Start-up from POR or LVI Reset 145
9.3.3 Clocks in Stop Mode and Wait Mode 146

9.4 Reset and System Initialization 146
9.4.1 External Pin Reset 147
9.4.2 Active Resets from Internal Sources 147
9.4.2.1 Power-On Reset 148
9.4.2.2 Computer Operating Properly (COP) Reset 149
9.4.2.3 Illegal Opcode Reset 150
9.4.2.4 Illegal Address Reset 150
9.4.2.5 Low-Voltage Inhibit (LVI) Reset 150
9.4.2.6 Monitor Mode Entry Module Reset 150

9.5 SIM Counter 151
9.5.1 SIM Counter During Power-On Reset 151
9.5.2 SIM Counter During Stop Mode Recovery 151
9.5.3 SIM Counter and Reset States 151

9.6 Exception Control 152
9.6.1 Interrupts 152
9.6.1.1 Hardware Interrupts 154
9.6.1.2 SWI Instruction 155
9.6.1.3 Interrupt Status Registers 155
9.6.1.4 Interrupt Status Register 1 155
9.6.1.5 Interrupt Status Register 2 157
9.6.1.6 Interrupt Status Register 3 157
9.6.2 Reset 158
9.6.3 Break Interrupts 158

9.6.4 Status Flag Protection in Break Mode 158

9.7 Low-Power Modes 159
9.7.1 Wait Mode 159
9.7.2 Stop Mode 160

9.8 SIM Registers 161
9.8.1 SIM Break Status Register 162
9.8.2 SIM Reset Status Register 163
9.8.3 SIM Break Flag Control Register 164

9.2 Introduction

This section describes the system integration module (SIM). Together with the CPU, the SIM controls all MCU activities. A block diagram of the SIM is shown in **Figure 9-1**. **Table 9-1** is a summary of the SIM input/output (I/O) registers. The SIM is a system state controller that coordinates CPU and exception timing. The SIM is responsible for:

- Bus clock generation and control for CPU and peripherals:
  - Stop/wait/reset/break entry and recovery
  - Internal clock control
- Master reset control, including power-on reset (POR) and COP timeout
- Interrupt control:
  - Acknowledge timing
  - Arbitration control timing
  - Vector address generation
- CPU enable/disable timing
- Modular architecture expandable to 128 interrupt sources

Table 9-1 shows the internal signal names used in this section.

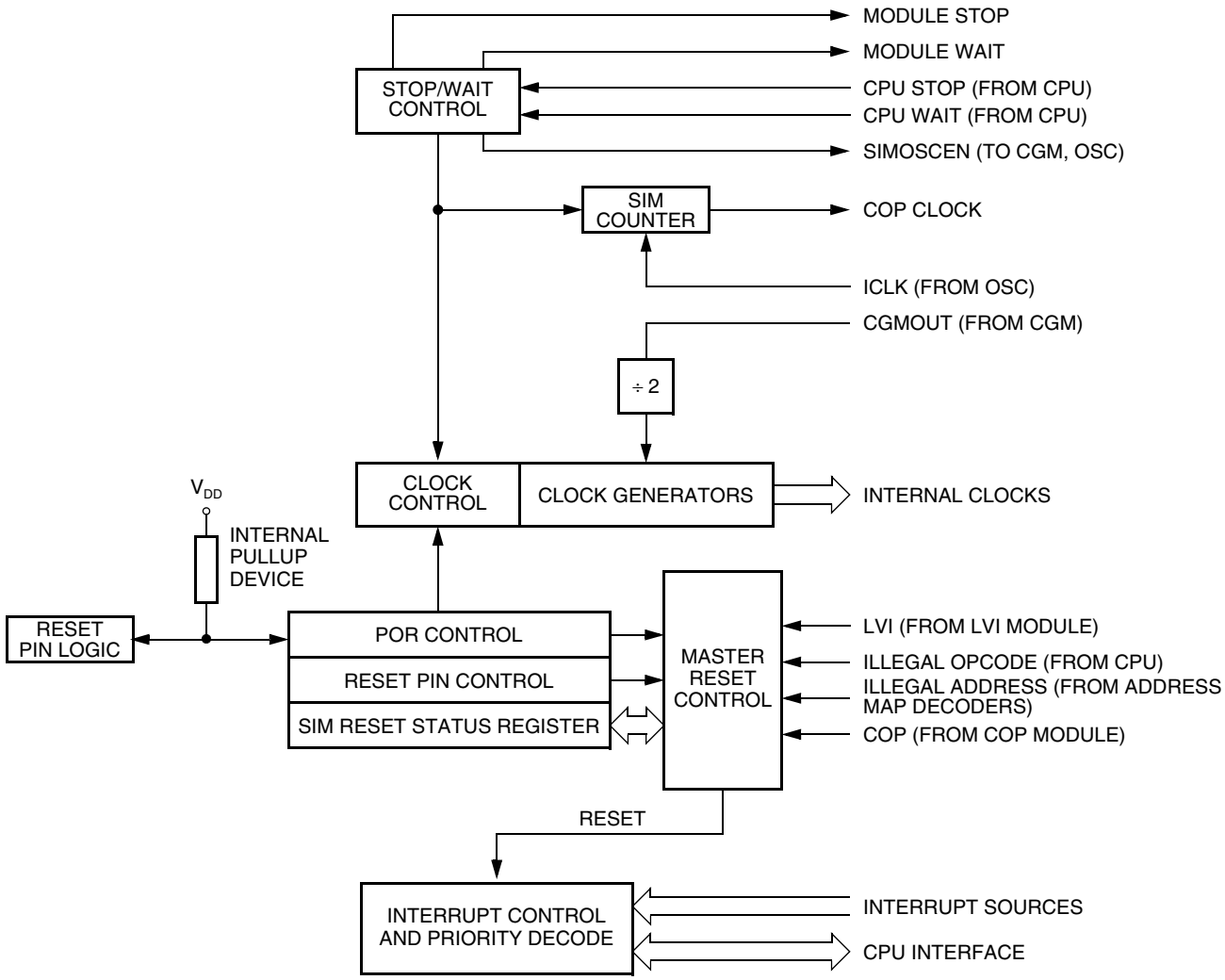


Figure 9-1. SIM Block Diagram

Table 9-1. Signal Name Conventions

Signal Name	Description
ICLK	Internal oscillator clock
CGMXCLK	Selected oscillator clock from oscillator module
CGMVCLK, CGMPCLK	PLL output and the divided PLL output
CGMOUT	CGMPCLK-based or oscillator-based clock output from CGM module (Bus clock = CGMOUT ÷ 2)
IAB	Internal address bus
IDB	Internal data bus
PORRST	Signal from the power-on reset module to the SIM
IRST	Internal reset signal
R/$\overline{W}$	Read/write signal

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE00	SIM Break Status Register (SBSR)	Read:	R	R	R	R	R	R	SBSW	R
		Write:							Note	
		Reset:							0	
Note: Writing a logic 0 clears SBSW.										
$FE01	SIM Reset Status Register (SRSR)	Read:	POR	PIN	COP	ILOP	ILAD	0	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
$FE03	SIM Break Flag Control Register (SBFCR)	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE04	Interrupt Status Register 1 (INT1)	Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE05	Interrupt Status Register 2 (INT2)	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE06	Interrupt Status Register 3 (INT3)	Read:	0	0	0	0	0	IF17	IF16	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
			= Unimplemented				R	= Reserved		

Figure 9-2. SIM I/O Register Summary

9.3 SIM Bus Clock Control and Generation

The bus clock generator provides system clock signals for the CPU and peripherals on the MCU. The system clocks are generated from an incoming clock, CGMOUT, as shown in **Figure 9-3**. This clock can come from either the oscillator module or from the on-chip PLL. (See **Section 8. Clock Generator Module (CGM)**.)

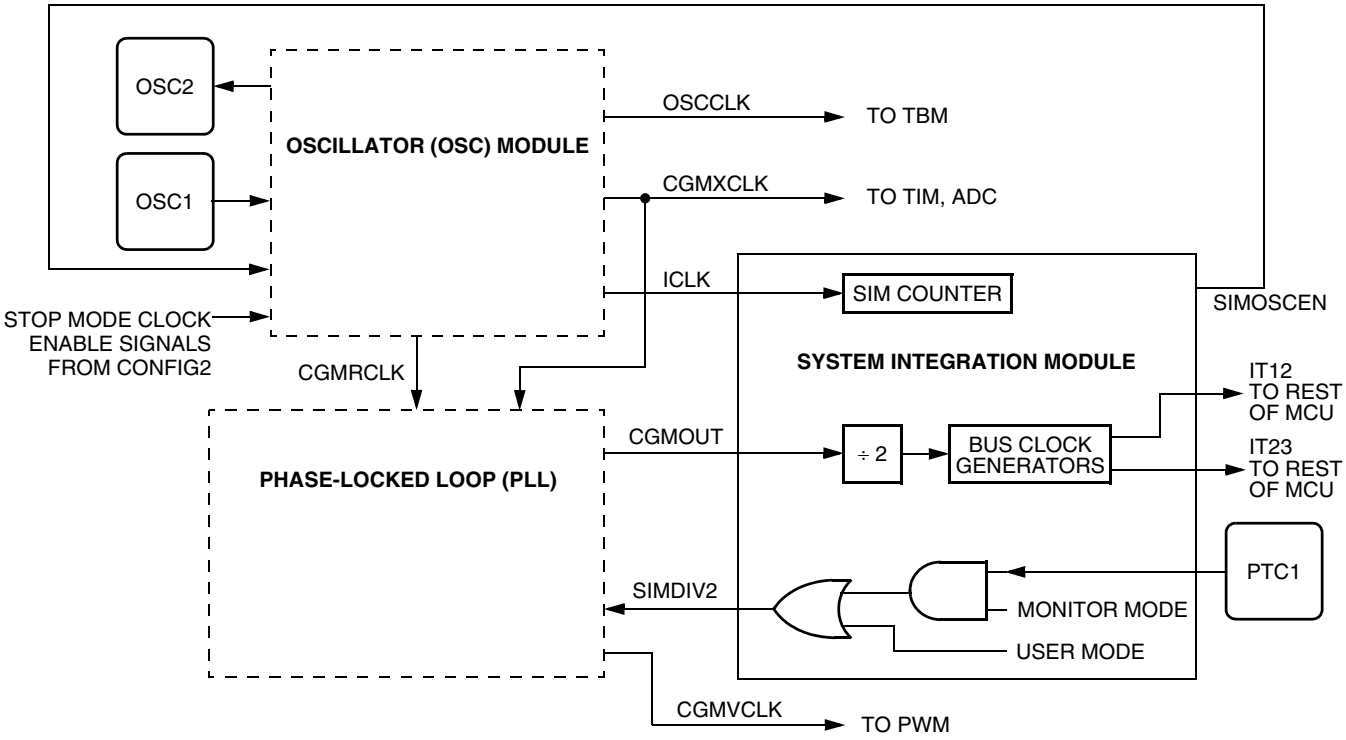


Figure 9-3. CGM Clock Signals

9.3.1 Bus Timing

In user mode, the internal bus frequency is either the oscillator output (CGMXCLK) divided by four or the divided PLL output (CGMPCLK) divided by four.

9.3.2 Clock Start-up from POR or LVI Reset

When the power-on reset module or the low-voltage inhibit module generates a reset, the clocks to the CPU and peripherals are inactive and held in an inactive phase until after the 4096 ICLK cycle POR timeout has completed. The $\overline{\text{RST}}$ pin is driven low by the SIM during this entire period. The IBUS clocks start upon completion of the timeout.

9.3.3 Clocks in Stop Mode and Wait Mode

Upon exit from stop mode by an interrupt, break, or reset, the SIM allows ICLK to clock the SIM counter. The CPU and peripheral clocks do not become active until after the stop delay timeout. This timeout is selectable as 4096 or 32 ICLK cycles. (See **9.7.2 Stop Mode**.)

In wait mode, the CPU clocks are inactive. The SIM also produces two sets of clocks for other modules. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

9.4 Reset and System Initialization

The MCU has these reset sources:

- Power-on reset module (POR)
- External reset pin ($\overline{\text{RST}}$)
- Computer operating properly module (COP)
- Low-voltage inhibit module (LVI)
- Illegal opcode
- Illegal address

All of these resets produce the vector $FFFE:$FFFF ($FEFE:$FEFF in monitor mode) and assert the internal reset signal (IRST). IRST causes all registers to be returned to their default values and all modules to be returned to their reset states.

An internal reset clears the SIM counter (see **9.5 SIM Counter**), but an external reset does not. Each of the resets sets a corresponding bit in the SIM reset status register (SRSR). (See **9.8 SIM Registers**.)

9.4.1 External Pin Reset

The $\overline{RST}$ pin circuit includes an internal pull-up device. Pulling the asynchronous $\overline{RST}$ pin low halts all processing. The PIN bit of the SIM reset status register (SRSR) is set as long as $\overline{RST}$ is held low for a minimum of 67 ICLK cycles, assuming that neither the POR nor the LVI was the source of the reset. See **Table 9-2** for details.
Figure 9-4 shows the relative timing.

Table 9-2. PIN Bit Set Timing

Reset Type	Number of Cycles Required to Set PIN
POR/LVI	4163 (4096 + 64 + 3)
All others	67 (64 + 3)

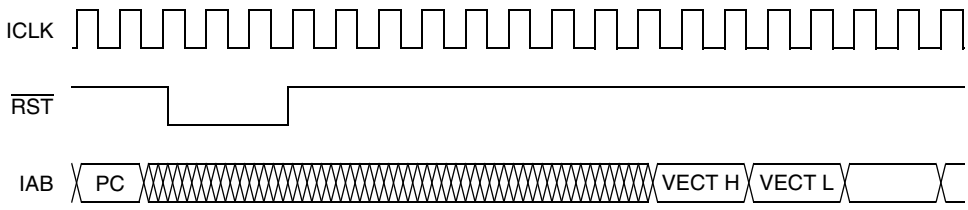


Figure 9-4. External Reset Timing

9.4.2 Active Resets from Internal Sources

All internal reset sources actively pull the $\overline{RST}$ pin low for 32 ICLK cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles (see Figure 9-5). An internal reset can be caused by an illegal address, illegal opcode, COP timeout, LVI, or POR (see Figure 9-6).

NOTE: *For LVI or POR resets, the SIM cycles through 4096 + 32 ICLK cycles during which the SIM forces the $\overline{RST}$ pin low. The internal reset signal then follows the sequence from the falling edge of $\overline{RST}$ shown in **Figure 9-5**.*

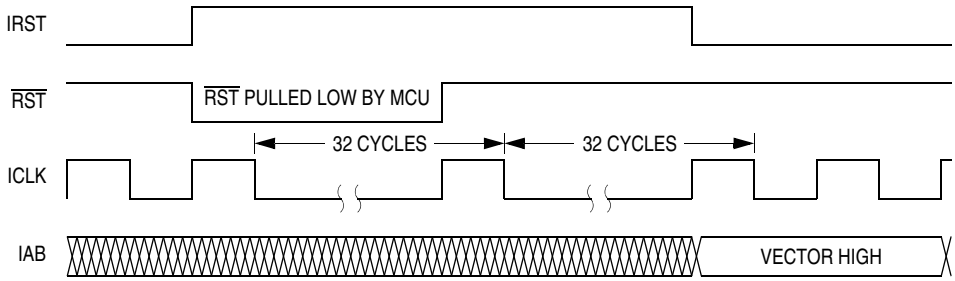


Figure 9-5. Internal Reset Timing

The COP reset is asynchronous to the bus clock.

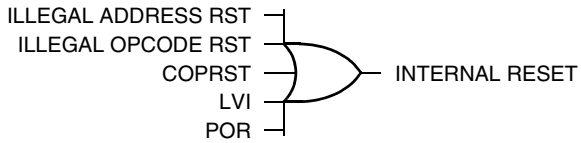


Figure 9-6. Sources of Internal Reset

The active reset feature allows the part to issue a reset to peripherals and other chips within a system built around the MCU.

9.4.2.1 Power-On Reset

When power is first applied to the MCU, the power-on reset module (POR) generates a pulse to indicate that power-on has occurred. The external reset pin ($\overline{RST}$) is held low while the SIM counter counts out 4096 + 32 ICLK cycles. Thirty-two ICLK cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur.

At power-on, these events occur:

- A POR pulse is generated.
- The internal reset signal is asserted.
- The SIM enables CGMOUT.
- Internal clocks to the CPU and modules are held inactive for 4096 ICLK cycles to allow stabilization of the oscillator.
- The $\overline{RST}$ pin is driven low during the oscillator stabilization time.
- The POR bit of the SIM reset status register (SRSR) is set and all other bits in the register are cleared.

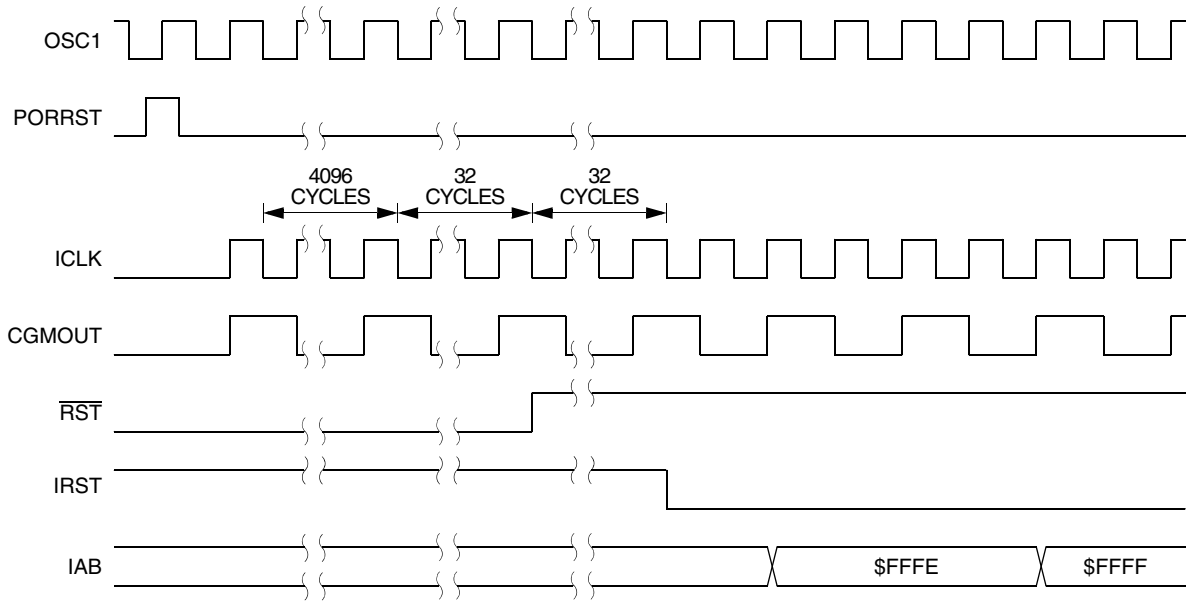


Figure 9-7. POR Recovery

9.4.2.2 Computer Operating Properly (COP) Reset

An input to the SIM is reserved for the COP reset signal. The overflow of the COP counter causes an internal reset and sets the COP bit in the SIM reset status register (SRSR). The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

To prevent a COP module timeout, write any value to location $FFFF. Writing to location $FFFF clears the COP counter and bits 12 through 5 of the SIM counter. The SIM counter output, which occurs at least every $2^{13} – 2^4$ ICLK cycles, drives the COP counter. The COP should be serviced as soon as possible out of reset to guarantee the maximum amount of time before the first timeout.

The COP module is disabled if the $\overline{\text{RST}}$ pin or the $\overline{\text{IRQ1}}$ pin is held at V_{TST} while the MCU is in monitor mode. The COP module can be disabled only through combinational logic conditioned with the high voltage signal on the $\overline{\text{RST}}$ or the $\overline{\text{IRQ1}}$ pin. This prevents the COP from becoming disabled as a result of external noise. During a break state, V_{TST} on the $\overline{\text{RST}}$ pin disables the COP module.

9.4.2.3 Illegal Opcode Reset

The SIM decodes signals from the CPU to detect illegal instructions. An illegal instruction sets the ILOP bit in the SIM reset status register (SRSR) and causes a reset.

If the stop enable bit, STOP, in the mask option register is logic 0, the SIM treats the STOP instruction as an illegal opcode and causes an illegal opcode reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

9.4.2.4 Illegal Address Reset

An opcode fetch from an unmapped address generates an illegal address reset. The SIM verifies that the CPU is fetching an opcode prior to asserting the ILAD bit in the SIM reset status register (SRSR) and resetting the MCU. A data fetch from an unmapped address does not generate a reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

9.4.2.5 Low-Voltage Inhibit (LVI) Reset

The low-voltage inhibit module (LVI) asserts its output to the SIM when the V_{DD} voltage falls to the LVI_{TRIPF} voltage. The LVI bit in the SIM reset status register (SRSR) is set, and the external reset pin ($\overline{\text{RST}}$) is held low while the SIM counter counts out 4096 + 32 ICLK cycles. Thirty-two ICLK cycles later, the CPU is released from reset to allow the reset vector sequence to occur. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

9.4.2.6 Monitor Mode Entry Module Reset

The monitor mode entry module reset asserts its output to the SIM when monitor mode is entered in the condition where the reset vectors are blank ($FF). (See **Section 10. Monitor ROM (MON)**.) When MODRST gets asserted, an internal reset occurs. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

9.5 SIM Counter

The SIM counter is used by the power-on reset module (POR) and in stop mode recovery to allow the oscillator time to stabilize before enabling the internal bus (IBUS) clocks. The SIM counter also serves as a prescaler for the computer operating properly module (COP). The SIM counter overflow supplies the clock for the COP module. The SIM counter is 12 bits long and is clocked by the falling edge of ICLK.

9.5.1 SIM Counter During Power-On Reset

The power-on reset module (POR) detects power applied to the MCU. At power-on, the POR circuit asserts the signal PORRST. Once the SIM is initialized, it enables the clock generation module (CGM) to drive the bus clock state machine.

9.5.2 SIM Counter During Stop Mode Recovery

The SIM counter also is used for stop mode recovery. The STOP instruction clears the SIM counter. After an interrupt, break, or reset, the SIM senses the state of the short stop recovery bit, SSREC, in the configuration register 1 (CONFIG1). If the SSREC bit is a logic 1, then the stop recovery is reduced from the normal delay of 4096 ICLK cycles down to 32 ICLK cycles. This is ideal for applications using canned oscillators that do not require long start-up times from stop mode. External crystal applications should use the full stop recovery time, that is, with SSREC cleared.

9.5.3 SIM Counter and Reset States

External reset has no effect on the SIM counter. (See **9.7.2 Stop Mode** for details.) The SIM counter is free-running after all reset states. (See **9.4.2 Active Resets from Internal Sources** for counter control and internal reset recovery sequences.)

9.6 Exception Control

Normal, sequential program execution can be changed in three different ways:

- Interrupts:
  - Maskable hardware CPU interrupts
  - Non-maskable software interrupt instruction (SWI)
- Reset
- Break interrupts

9.6.1 Interrupts

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. **Figure 9-8** shows interrupt entry timing, and **Figure 9-9** shows interrupt recovery timing.

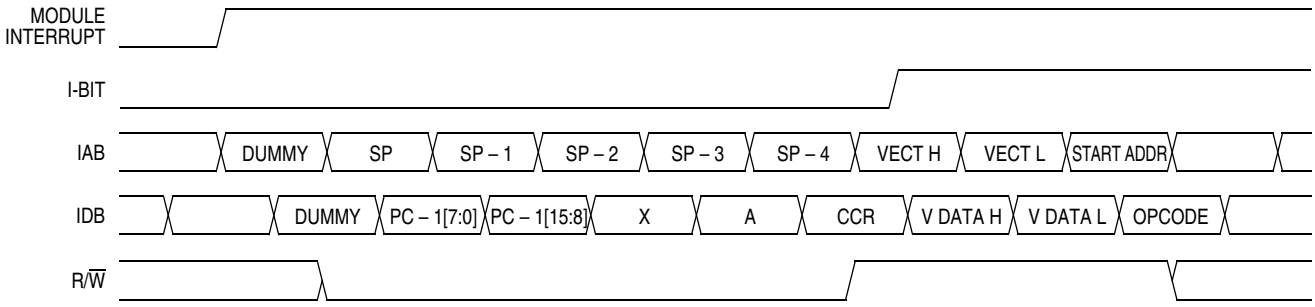


Figure 9-8. Interrupt Entry Timing

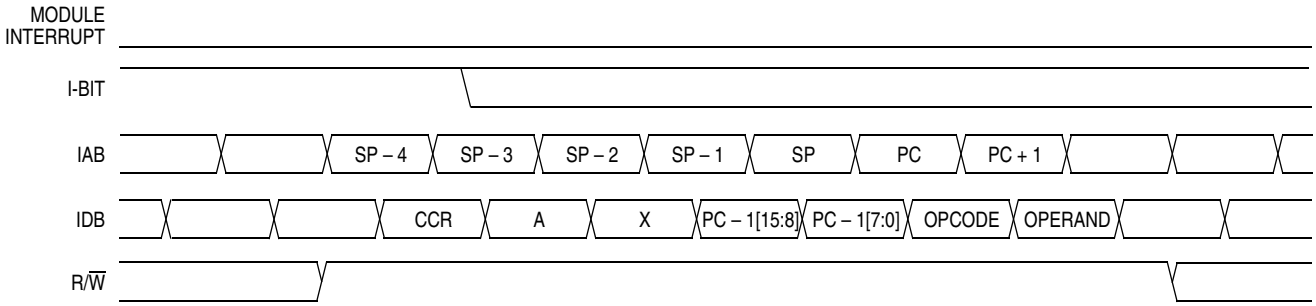


Figure 9-9. Interrupt Recovery Timing

Interrupts are latched, and arbitration is performed in the SIM at the start of interrupt processing. The arbitration result is a constant that the CPU uses to determine which vector to fetch. Once an interrupt is latched by the SIM, no other interrupt can take precedence, regardless of priority, until the latched interrupt is serviced (or the I bit is cleared).
(See **Figure 9-10**.)

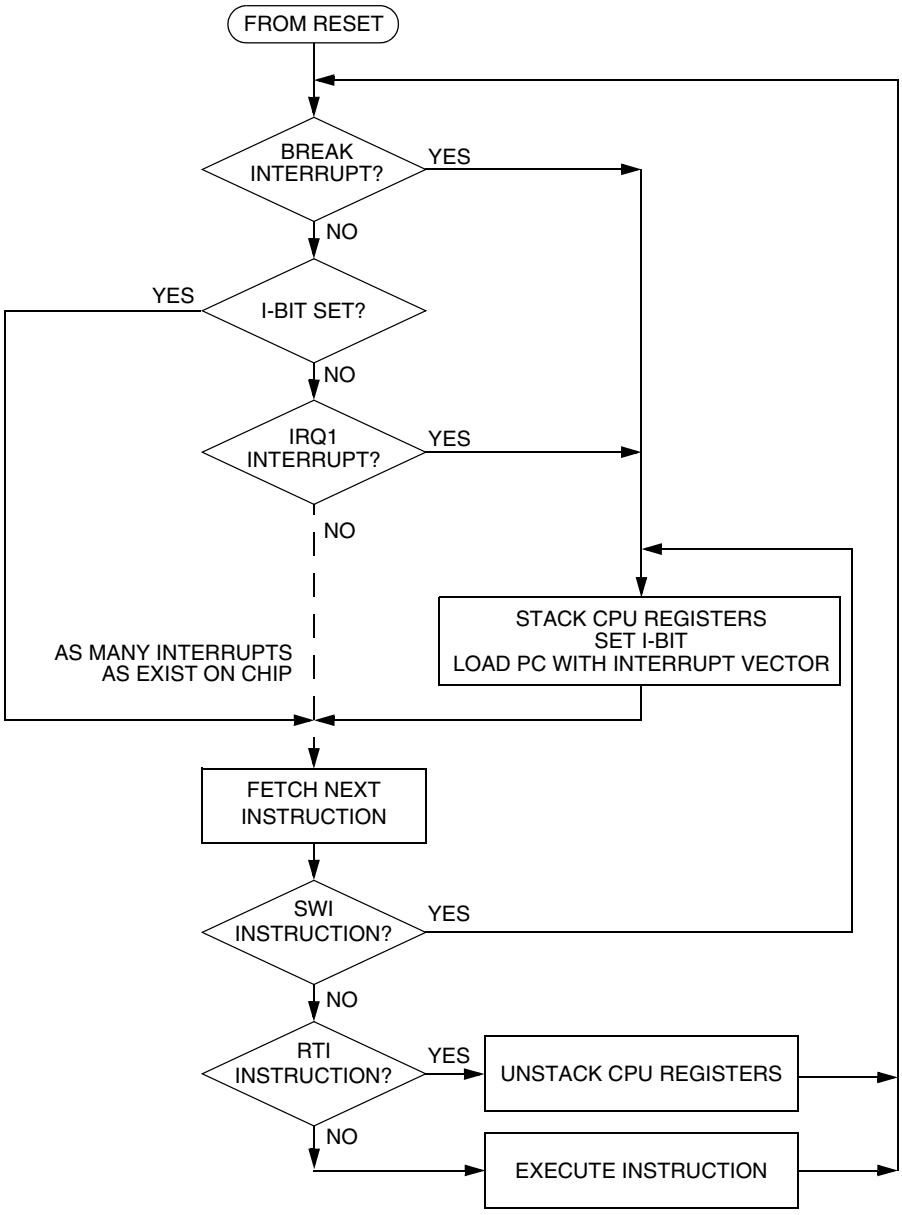


Figure 9-10. Interrupt Processing

9.6.1.1 Hardware Interrupts

A hardware interrupt does not stop the current instruction. Processing of a hardware interrupt begins after completion of the current instruction. When the current instruction is complete, the SIM checks all pending hardware interrupts. If interrupts are not masked (I bit clear in the condition code register) and if the corresponding interrupt enable bit is set, the SIM proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If more than one interrupt is pending at the end of an instruction execution, the highest priority interrupt is serviced first. **Figure 9-11** demonstrates what happens when two interrupts are pending. If an interrupt is pending upon exit from the original interrupt service routine, the pending interrupt is serviced before the LDA instruction is executed.

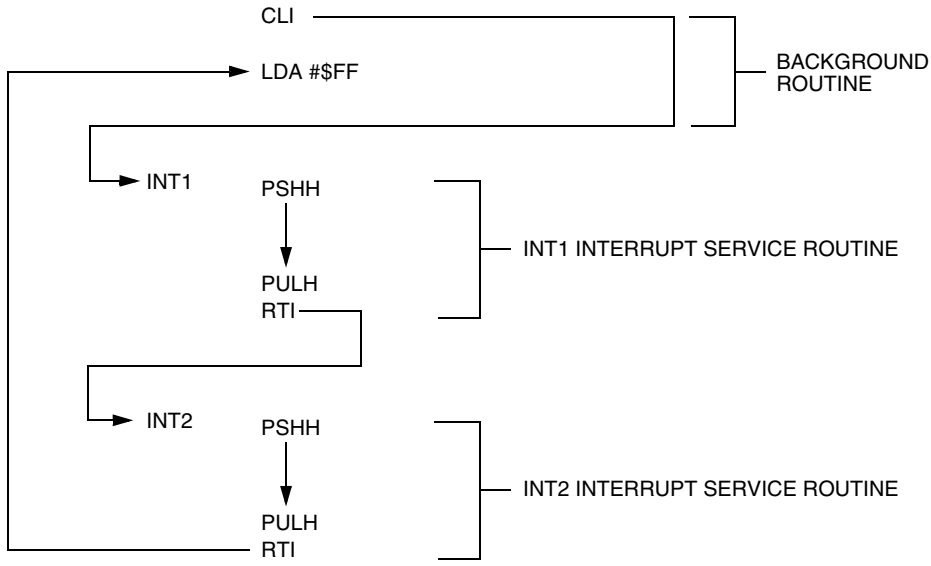


Figure 9-11. Interrupt Recognition Example

The LDA opcode is prefetched by both the INT1 and INT2 RTI instructions. However, in the case of the INT1 RTI prefetch, this is a redundant operation.

NOTE: *To maintain compatibility with the M6805 Family, the H register is not pushed on the stack during interrupt entry. If the interrupt service routine modifies the H register or uses the indexed addressing mode, software should save the H register and then restore it prior to exiting the routine.*

9.6.1.2 SWI Instruction

The SWI instruction is a non-maskable instruction that causes an interrupt regardless of the state of the interrupt mask (I bit) in the condition code register.

NOTE: *A software interrupt pushes PC onto the stack. A software interrupt does not push PC – 1, as a hardware interrupt does.*

9.6.1.3 Interrupt Status Registers

The flags in the interrupt status registers identify maskable interrupt sources. **Table 9-3** summarizes the interrupt sources and the interrupt status register flags that they set. The interrupt status registers can be useful for debugging.

9.6.1.4 Interrupt Status Register 1

Address: $FE04

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 9-12. Interrupt Status Register 1 (INT1)

IF6–IF1 — Interrupt Flags 6–1

These flags indicate the presence of interrupt requests from the sources shown in **Table 9-3**.

1 = Interrupt request present
0 = No interrupt request present

Bit 0 and Bit 1 — Always read 0

Table 9-3. Interrupt Sources

Priority	INT Flag	Vector Address	Interrupt Source
Lowest	IF17	$FFDA	Timebase Module
		$FFDB	
	IF16	$FFDC	Analog Module
		$FFDD	
	IF15	$FFDE	ADC Conversion Complete
		$FFDF	
	IF14	$FFE0	Keyboard
		$FFE1	
	IF13	$FFE2	SCI Transmit
		$FFE3	
	IF12	$FFE4	SCI Receive
		$FFE5	
	IF11	$FFE6	SCI Error
		$FFE7	
	IF10	$FFE8	MMIIC
		$FFE9	
	IF9	$FFEA	TIM2 Overflow
		$FFEB	
	IF8	$FFEC	TIM2 Channel 1
		$FFED	
	IF7	$FFEE	TIM2 Channel 0
		$FFEF	
	IF6	$FFF0	TIM1 Overflow
		$FFF1	
	IF5	$FFF2	TIM1 Channel 1
		$FFF3	
	IF4	$FFF4	TIM1 Channel 0
		$FFF5	
	IF3	$FFF6	PLL
		$FFF7	
	IF2	$FFF8	$\overline{\text{IRQ2}}$
		$FFF9	
	IF1	$FFFA	$\overline{\text{IRQ1}}$
		$FFFB	
		$FFFC	SWI
		$FFFD	
		$FFFE	Reset
Highest		$FFFF	

9.6.1.5 Interrupt Status Register 2

Address: $FE05

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 9-13. Interrupt Status Register 2 (INT2)

IF14–IF7 — Interrupt Flags 14–7

These flags indicate the presence of interrupt requests from the sources shown in **Table 9-3**.

1 = Interrupt request present
0 = No interrupt request present

9.6.1.6 Interrupt Status Register 3

Address: $FE06

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	IF17	IF16	IF15
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 9-14. Interrupt Status Register 3 (INT3)

IF17–IF15 — Interrupt Flags 17–15

These flags indicate the presence of an interrupt request from the source shown in **Table 9-3**.

1 = Interrupt request present
0 = No interrupt request present

9.6.2 Reset

All reset sources always have equal and highest priority and cannot be arbitrated.

9.6.3 Break Interrupts

The break module can stop normal program flow at a software-programmable break point by asserting its break interrupt output. (See **Section 23. Break Module (BRK)**.) The SIM puts the CPU into the break state by forcing it to the SWI vector location. Refer to the break interrupt subsection of each module to see how each module is affected by the break state.

9.6.4 Status Flag Protection in Break Mode

The SIM controls whether status flags contained in other modules can be cleared during break mode. The user can select whether flags are protected from being cleared by properly initializing the break clear flag enable bit (BCFE) in the SIM break flag control register (SBFCR).

Protecting flags in break mode ensures that set flags will not be cleared while in break mode. This protection allows registers to be freely read and written during break mode without losing status flag information.

Setting the BCFE bit enables the clearing mechanisms. Once cleared in break mode, a flag remains cleared even when break mode is exited. Status flags with a 2-step clearing mechanism — for example, a read of one register followed by the read or write of another — are protected, even when the first step is accomplished prior to entering break mode. Upon leaving break mode, execution of the second step will clear the flag as normal.

9.7 Low-Power Modes

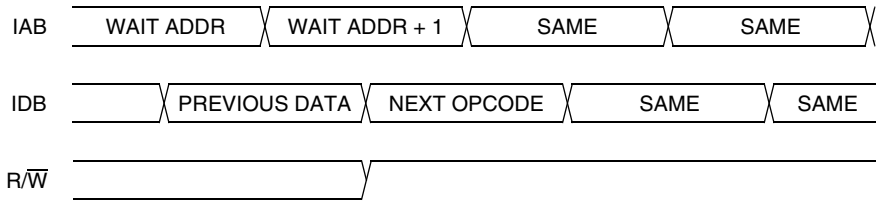
Executing the WAIT or STOP instruction puts the MCU in a low power-consumption mode for standby situations. The SIM holds the CPU in a non-clocked state. The operation of each of these modes is described in the following subsections. Both STOP and WAIT clear the interrupt mask (I) in the condition code register, allowing interrupts to occur.

9.7.1 Wait Mode

In wait mode, the CPU clocks are inactive while the peripheral clocks continue to run. **Figure 9-15** shows the timing for wait mode entry.

A module that is active during wait mode can wake up the CPU with an interrupt if the interrupt is enabled. Stacking for the interrupt begins one cycle after the WAIT instruction during which the interrupt occurred. In wait mode, the CPU clocks are inactive. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

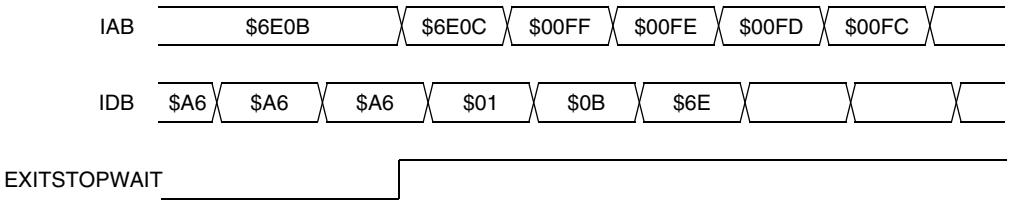
Wait mode also can be exited by a reset or break. A break interrupt during wait mode sets the SIM break stop/wait bit, SBSW, in the SIM break status register (SBSR). If the COP disable bit, COPD, in the mask option register is logic 0, then the computer operating properly module (COP) is enabled and remains active in wait mode.



NOTE: Previous data can be operand data or the WAIT opcode, depending on the last instruction.

Figure 9-15. Wait Mode Entry Timing

Figure 9-16 and **Figure 9-17** show the timing for WAIT recovery.



NOTE: EXITSTOPWAIT = $\overline{\text{RST}}$ pin OR CPU interrupt OR break interrupt

Figure 9-16. Wait Recovery from Interrupt or Break

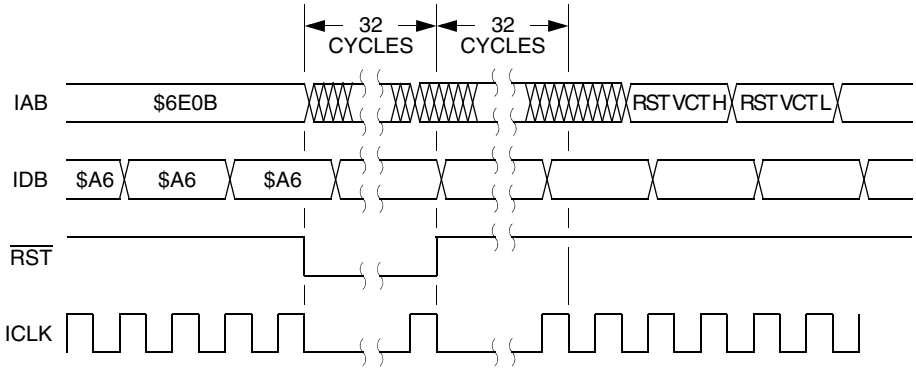


Figure 9-17. Wait Recovery from Internal Reset

9.7.2 Stop Mode

In stop mode, the SIM counter is reset and the system clocks are disabled. An interrupt request from a module can cause an exit from stop mode. Stacking for interrupts begins after the selected stop recovery time has elapsed. Reset or break also causes an exit from stop mode.

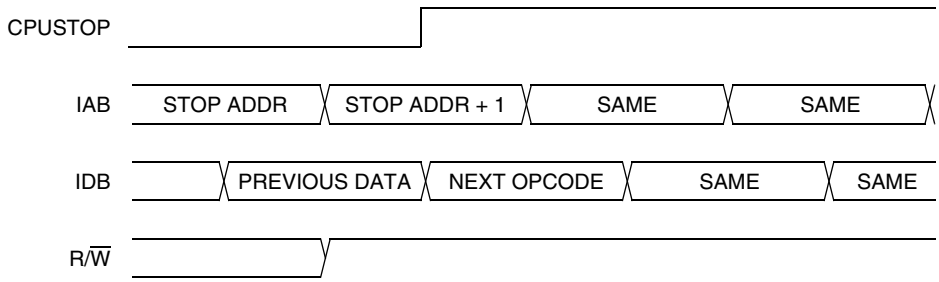
The SIM disables the clock generator module output (CGMOUT) in stop mode, stopping the CPU and peripherals. Stop recovery time is selectable using the SSREC bit in the configuration register 1 (CONFIG1). If SSREC is set, stop recovery is reduced from the normal delay of 4096 ICLK cycles down to 32. This is ideal for applications using canned oscillators that do not require long start-up times from stop mode.

NOTE: *External crystal applications should use the full stop recovery time by clearing the SSREC bit.*

A break interrupt during stop mode sets the SIM break stop/wait bit (SBSW) in the SIM break status register (SBSR).

The SIM counter is held in reset from the execution of the STOP instruction until the beginning of stop recovery. It is then used to time the recovery period. **Figure 9-18** shows stop mode entry timing.

NOTE: *To minimize stop current, all pins configured as inputs should be driven to a logic 1 or logic 0.*



NOTE: Previous data can be operand data or the STOP opcode, depending on the last instruction.

Figure 9-18. Stop Mode Entry Timing

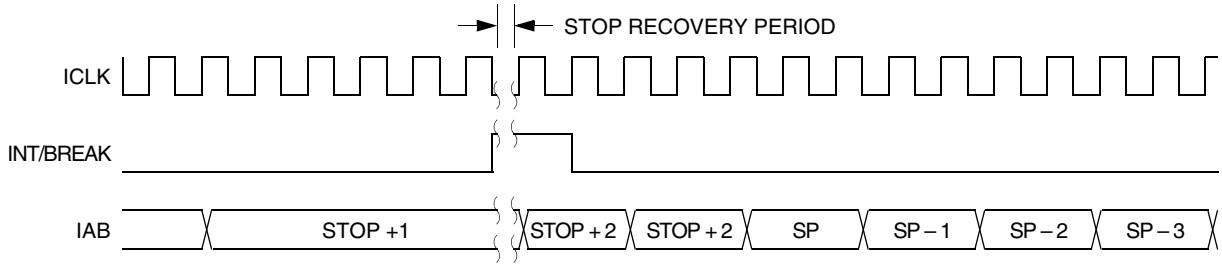


Figure 9-19. Stop Mode Recovery from Interrupt or Break

9.8 SIM Registers

The SIM has three memory-mapped registers:

- SIM Break Status Register (SBSR) — $FE00
- SIM Reset Status Register (SRSR) — $FE01
- SIM Break Flag Control Register (SBFCR) — $FE03

9.8.1 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from stop mode or wait mode.

Address: $FE00

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	R	R	R	R	R	SBSW	R
Write:							Note	
Reset:							0	

Note: Writing a logic 0 clears SBSW. R = Reserved

Figure 9-20. SIM Break Status Register (SBSR)

SBSW — Break Wait Bit

This status bit is set when a break interrupt causes an exit from wait mode or stop mode. Clear SBSW by writing a logic 0 to it. Reset clears SBSW.

1 = Stop mode or wait mode was exited by break interrupt
0 = Stop mode or wait mode was not exited by break interrupt

SBSW can be read within the break interrupt routine. The user can modify the return address on the stack by subtracting 1 from it. The following code is an example.

```
This code works if the H register has been pushed onto the stack in the break
service routine software. This code should be executed at the end of the break
service routine software.
HIBYTE   EQU
LOBYTE   EQU
         If not SBSW, do RTI
         BRCLR   SBSW,SBSR, RETURN    ;See if wait mode or stop mode was exited by
                                      ;break.
         TST     LOBYTE,SP            ;If RETURNLO is not zero,
         BNE     DOLO                 ;then just decrement low byte.
         DEC     HIBYTE,SP            ;Else deal with high byte, too.
DOLO     DEC     LOBYTE,SP            ;Point to WAIT/STOP opcode.
RETURN   PULH                         ;Restore H register.
         RTI
```

9.8.2 SIM Reset Status Register

This register contains six flags that show the source of the last reset provided all previous reset status bits have been cleared. Clear the SIM reset status register by reading it. A power-on reset sets the POR bit and clears all other bits in the register.

Address: $FE01

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	POR	PIN	COP	ILOP	ILAD	0	LVI	0
Write:								
Reset:	1	0	0	0	0	0	0	0

= Unimplemented

Figure 9-21. SIM Reset Status Register (SRSR)

POR — Power-On Reset Bit
1 = Last reset caused by POR circuit
0 = Read of SRSR

PIN — External Reset Bit
1 = Last reset caused by external reset pin ($\overline{\text{RST}}$)
0 = POR or read of SRSR

COP — Computer Operating Properly Reset Bit
1 = Last reset caused by COP counter
0 = POR or read of SRSR

ILOP — Illegal Opcode Reset Bit
1 = Last reset caused by an illegal opcode
0 = POR or read of SRSR

ILAD — Illegal Address Reset Bit (opcode fetches only)
1 = Last reset caused by an opcode fetch from an illegal address
0 = POR or read of SRSR

LVI — Low-Voltage Inhibit Reset Bit
1 = Last reset caused by the LVI circuit
0 = POR or read of SRSR

9.8.3 SIM Break Flag Control Register

The SIM break control register contains a bit that enables software to clear status bits while the MCU is in a break state.

Address: $FE03

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	BCFE	R	R	R	R	R	R	R
Reset:	0							

R = Reserved

Figure 9-22. SIM Break Flag Control Register (SBFCR)

BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

1 = Status bits clearable during break

0 = Status bits not clearable during break

Section 10. Monitor ROM (MON)

10.1 Contents

10.2 Introduction 165
10.3 Features 166
10.4 Functional Description 166
10.4.1 Entering Monitor Mode 168
10.4.2 Data Format 172
10.4.3 Break Signal 172
10.4.4 Baud Rate 173
10.4.5 Commands 173
10.5 Security 178

10.2 Introduction

This section describes the monitor ROM (MON) and the monitor mode entry methods. The monitor ROM allows complete testing of the MCU through a single-wire interface with a host computer. Monitor mode entry can be achieved without use of the higher test voltage, V_{TST}, as long as vector addresses $FFFE and $FFFF are blank, thus reducing the hardware requirements for in-circuit programming.

10.3 Features

Features of the monitor ROM include:

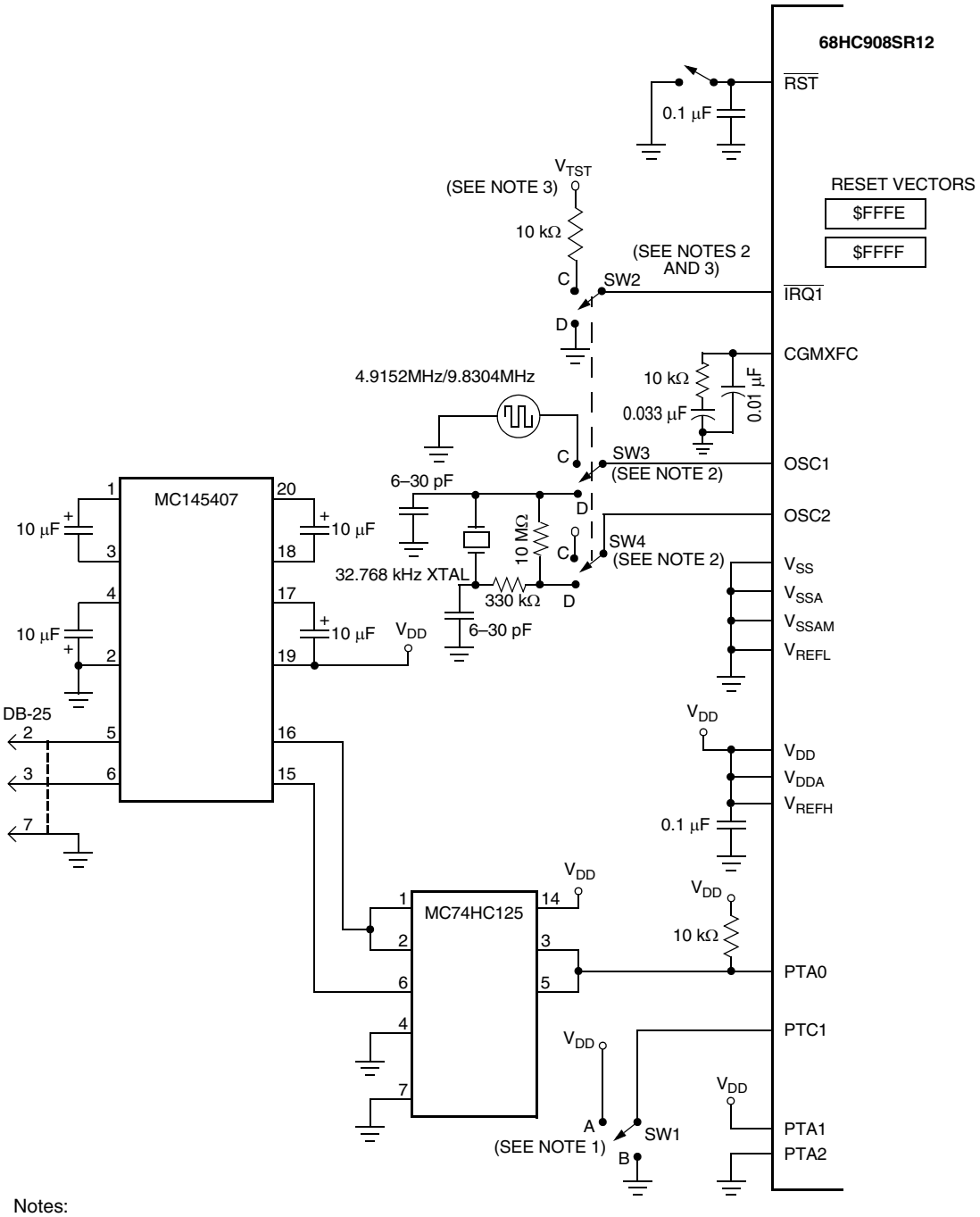
- Normal user-mode pin functionality
- One pin dedicated to serial communication between monitor ROM and host computer
- Standard mark/space non-return-to-zero (NRZ) communication with host computer
- Execution of code in RAM or FLASH
- FLASH memory security feature[1]
- FLASH memory programming interface
- Enhanced PLL (phase-locked loop) option to allow use of external 32.768-kHz crystal to generate internal frequency of 2.4576 MHz
- 368 bytes monitor ROM code size ($FE10 to $FF7F)
- Monitor mode entry without high voltage, V_{TST}, if reset vector is blank ($FFFE and $FFFF contain $FF)
- Standard monitor mode entry if high voltage, V_{TST}, is applied to $\overline{\text{IRQ1}}$

10.4 Functional Description

The monitor ROM receives and executes commands from a host computer. **Figure 10-1** shows an example circuit used to enter monitor mode and communicate with a host computer via a standard RS-232 interface.

Simple monitor commands can access any memory address. In monitor mode, the MCU can execute code downloaded into RAM by a host computer while most MCU pins retain normal operating mode functions. All communication between the host computer and the MCU is through the PTA0 pin. A level-shifting and multiplexing interface is required between PTA0 and the host computer. PTA0 is used in a wired-OR configuration and requires a pullup resistor.

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.



Notes:
1. For monitor mode entry when SW2 at position C ($\overline{IRQ1}$ = V_{TST}):
   SW1: Position A — Bus clock = CGMXCLK ÷ 4
   SW1: Position B — Bus clock = CGMXCLK ÷ 2
2. SW2, SW3, and SW4: Position C — Enter monitor mode using off-chip oscillator only.
   SW2, SW3, and SW4: Position D — Enter monitor mode using 32.768kHz XTAL and internal PLL.
3. See **Table 24-5** for $\overline{IRQ1}$ voltage level requirements.
4. See **Table 10-1** for other monitor mode entry configurations.

Figure 10-1. Monitor Mode Circuit

The monitor code allows enabling the PLL to generate the internal clock, provided the reset vector is blank ($FF), when the device is being clocked by a low-frequency crystal. This entry method, which is enabled when $\overline{IRQ1}$ is held low out of reset, is intended to support serial communication/programming at 9600 baud in monitor mode by stepping up the external frequency (assumed to be 32.768 kHz) by a fixed amount to generate the desired internal frequency (2.4576 MHz).

If the reset vector is not blank (not $FF), the frequency stepping feature is not supported, because $\overline{IRQ1}$ cannot be held low for monitor mode entry. With a non-blank reset vector, entry into monitor mode requires V_{TST} on $\overline{IRQ1}$.

10.4.1 Entering Monitor Mode

Table 10-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a POR and will allow communication at 9600 baud provided one of the following sets of conditions is met:

1. If $FFFE and $FFFF do not contain $FF (programmed state):
   - The external clock is 4.9152 MHz with PTC1 low or 9.8304 MHz with PTC1 high
   - $\overline{IRQ1}$ = V_{TST} (PLL off)
2. If $FFFE and $FFFF both contain $FF (erased state):
   - The external clock is 9.8304 MHz
   - $\overline{IRQ1}$ = V_{DD} (this can be implemented through the internal $\overline{IRQ1}$ pullup; PLL off)
3. If $FFFE and $FFFF both contain $FF (erased state):
   - The external clock is 32.768 kHz (crystal)
   - $\overline{IRQ1}$ = V_{SS} (this setting initiates the PLL to boost the external 32.768 kHz to an internal bus frequency of 2.4576 MHz)

Table 10-1. Monitor Mode Signal Requirements and Options

$\overline{IRQ1}$	$\overline{RST}$	Address $FFFE/ $FFFF	PTA2	PTA1	PTA0[1]	PTC1	External Clock	Bus Frequency	PLL	COP	Baud Rate	Comment
X	GND	X	X	X	X	X	X	0	X	Disabled	0	No operation until reset goes high
V_{TST}	V_{DD} or V_{TST}	X	0	1	1	0	4.9152 MHz[2]	2.4576 MHz	OFF	Disabled	9600	PTA1 and PTA2 voltages only required if $\overline{IRQ1}$ = V_{TST}; PTC1 determines frequency divider
V_{TST}	V_{DD} or V_{TST}	X	0	1	1	1	9.8304 MHz[2]	2.4576 MHz	OFF	Disabled	9600	PTA1 and PTA2 voltages only required if $\overline{IRQ1}$ = V_{TST}; PTC1 determines frequency divider
V_{DD}	V_{DD}	Blank "$FFFF"	X	X	1	X	9.8304 MHz[3]	2.4576 MHz	OFF	Disabled	9600	External frequency always divided by 4
GND	V_{DD}	Blank "$FFFF"	X	X	1	X	32.768 kHz[3]	2.4576 MHz	ON	Disabled	9600	PLL enabled (BCS set) in monitor code
V_{DD} or GND	V_{TST}	Blank "$FFFF"	X	X	X	X	X	—	OFF	Enabled	—	Enters user mode — will encounter an illegal address reset
V_{DD} or GND	V_{DD} or V_{TST}	Not Blank	X	X	X	X	X	—	OFF	Enabled	—	Enters user mode

Notes:
1. PTA0 = 1 if serial communication; PTA0 = 0 if parallel communication (factory use only)
2. When $\overline{IRQ1}$ = V_{TST}, external clock must be derived by a 4.9152MHz or 9.8304MHz off-chip oscillator.
3. External clock is derived by a crystal or an off-chip oscillator.

If V_{TST} is applied to $\overline{IRQ1}$ and PTC1 is low upon monitor mode entry (above condition set 1), the bus frequency is a divide-by-two of the input clock. If PTC1 is high with V_{TST} applied to $\overline{IRQ1}$ upon monitor mode entry, the bus frequency will be a divide-by-four of the input clock. Holding the PTC1 pin low when entering monitor mode causes a bypass of a divide-by-two stage at the oscillator only if V_{TST} is applied to $\overline{IRQ1}$. In this event, the CGMOUT frequency is equal to the CGMXCLK frequency, and the OSC1 input directly generates internal bus clocks. In this case, the OSC1 signal must have a 50% duty cycle at maximum bus frequency.

If entering monitor mode without high voltage on $\overline{IRQ1}$ (above condition set 2 or 3, where applied voltage is either V_{DD} or V_{SS}), then all port A pin requirements and conditions, including the PTC1 frequency divisor selection, are not in effect. This is to reduce circuit requirements when performing in-circuit programming.

NOTE: *If the reset vector is blank and monitor mode is entered, the chip will see an additional reset cycle after the initial POR reset. Once the part has been programmed, the traditional method of applying a voltage, V_{TST}, to $\overline{IRQ1}$ must be used to enter monitor mode.*

The COP module is disabled in monitor mode based on these conditions:

- If monitor mode was entered as a result of the reset vector being blank (above condition set 2 or 3), the COP is always disabled regardless of the state of $\overline{IRQ1}$ or $\overline{RST}$.
- If monitor mode was entered with V_{TST} on $\overline{IRQ1}$ (condition set 1), then the COP is disabled as long as V_{TST} is applied to either $\overline{IRQ1}$ or $\overline{RST}$.

The second condition states that as long as V_{TST} is maintained on the $\overline{IRQ1}$ pin after entering monitor mode, or if V_{TST} is applied to $\overline{RST}$ after the initial reset to get into monitor mode (when V_{TST} was applied to $\overline{IRQ1}$), then the COP will be disabled. In the latter situation, after V_{TST} is applied to the $\overline{RST}$ pin, V_{TST} can be removed from the $\overline{IRQ1}$ pin in the interest of freeing the $\overline{IRQ1}$ for normal functionality in monitor mode.

Figure 10-2 shows a simplified diagram of the monitor mode entry when the reset vector is blank and just 1 x V_{DD} voltage is applied to the $\overline{IRQ1}$ pin. An external oscillator of 9.8304 MHz is required for a baud rate of 9600, as the internal bus frequency is automatically set to the external frequency divided by four.

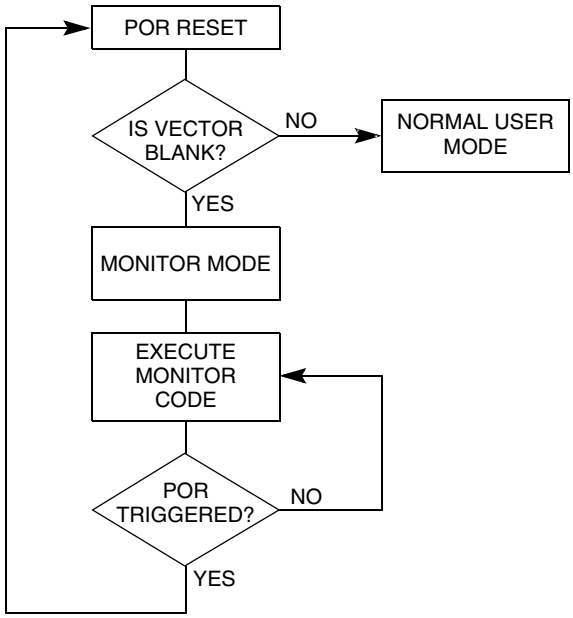


Figure 10-2. Low-Voltage Monitor Mode Entry Flowchart

Enter monitor mode with pin configuration shown in Figure 10-1 by pulling $\overline{RST}$ low and then high. The rising edge of $\overline{RST}$ latches monitor mode. Once monitor mode is latched, the values on the specified pins can change.

Once out of reset, the MCU waits for the host to send eight security bytes. (See 10.5 Security.) After the security bytes, the MCU sends a break signal (10 consecutive logic 0s) to the host, indicating that it is ready to receive a command.

In monitor mode, the MCU uses different vectors for reset, SWI (software interrupt), and break interrupt than those for user mode. The alternate vectors are in the $FE page instead of the $FF page and allow code execution from the internal monitor firmware instead of user code.

NOTE: *Exiting monitor mode after it has been initiated by having a blank reset vector requires a power-on reset (POR). Pulling $\overline{RST}$ low will not exit monitor mode in this situation.*

Table 10-2 summarizes the differences between user mode and monitor mode.

Table 10-2. Mode Differences

Modes	Functions					
	Reset Vector High	Reset Vector Low	Break Vector High	Break Vector Low	SWI Vector High	SWI Vector Low
User	$FFFE	$FFFF	$FFFC	$FFFD	$FFFC	$FFFD
Monitor	$FEFE	$FEFF	$FEFC	$FEFD	$FEFC	$FEFD

10.4.2 Data Format

Communication with the monitor ROM is in standard non-return-to-zero (NRZ) mark/space data format. Transmit and receive baud rates must be identical.

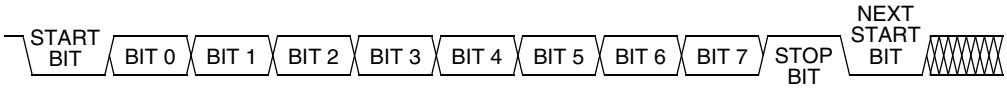


Figure 10-3. Monitor Data Format

10.4.3 Break Signal

A start bit (logic 0) followed by nine logic 0 bits is a break signal. When the monitor receives a break signal, it drives the PTA0 pin high for the duration of two bits and then echoes back the break signal.

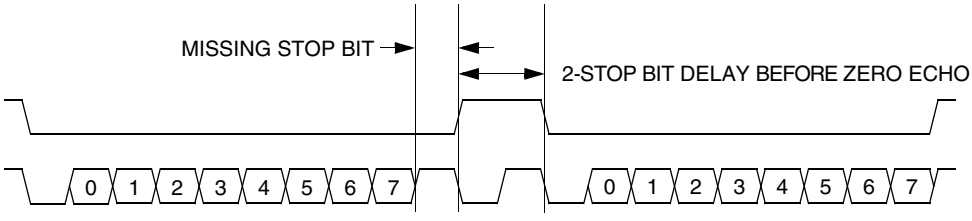


Figure 10-4. Break Transaction

10.4.4 Baud Rate

The communication baud rate is controlled by the crystal frequency and the state of the PTC1 pin (when $\overline{IRQ1}$ is set to V_{TST}) upon entry into monitor mode. When PTC1 is high, the divide by ratio is 1024. If the PTC1 pin is at logic 0 upon entry into monitor mode, the divide by ratio is 512.

If monitor mode was entered with V_{DD} on $\overline{IRQ1}$, then the divide by ratio is set at 1024, regardless of PTC1. If monitor mode was entered with V_{SS} on $\overline{IRQ}$, then the internal PLL steps up the external frequency, presumed to be 32.768 kHz, to 2.4576 MHz. These latter two conditions for monitor mode entry require that the reset vector is blank.

Table 10-3 lists external frequencies required to achieve a standard baud rate of 9600 BPS. Other standard baud rates can be accomplished using proportionally higher or lower frequency generators. If using a crystal as the clock source, be aware of the upper frequency limit that the internal clock module can handle. See **24.6 5.0V DC Electrical Characteristics** and **24.8 5.0V Control Timing** for this limit.

Table 10-3. Monitor Baud Rate Selection

External Frequency	$\overline{IRQ1}$	PTC1	Internal Frequency	Baud Rate (BPS)
4.9152 MHz	V_{TST}	0	2.4576 MHz	9600
9.8304 MHz	V_{TST}	1	2.4576 MHz	9600
9.8304 MHz	V_{DD}	X	2.4576 MHz	9600
32.768 kHz	V_{SS}	X	2.4576 MHz	9600

10.4.5 Commands

The monitor ROM firmware uses these commands:

- READ (read memory)
- WRITE (write memory)
- IREAD (indexed read)

- IWRITE (indexed write)
- READSP (read stack pointer)
- RUN (run user program)

The monitor ROM firmware echoes each received byte back to the PTA0 pin for error checking. An 11-bit delay at the end of each command allows the host to send a break character to cancel the command. A delay of two bit times occurs before each echo and before READ, IREAD, or READSP data is returned. The data returned by a read command appears after the echo of the last byte of the command.

NOTE: *Wait one bit time after each echo before sending the next byte.*

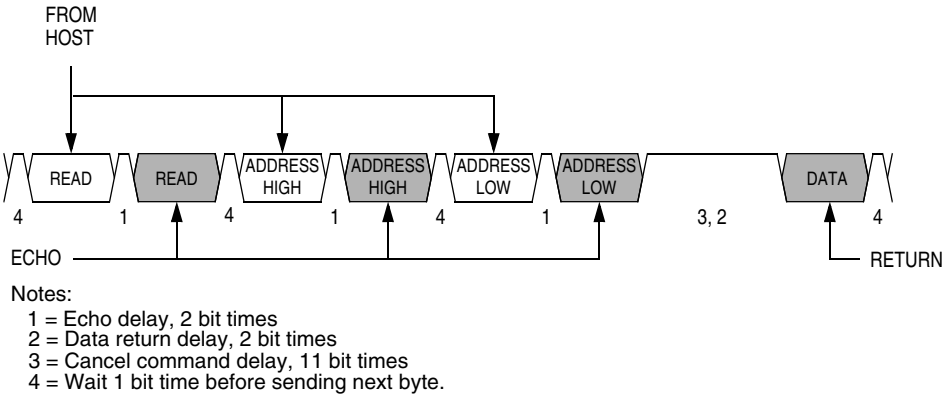


Figure 10-5. Read Transaction

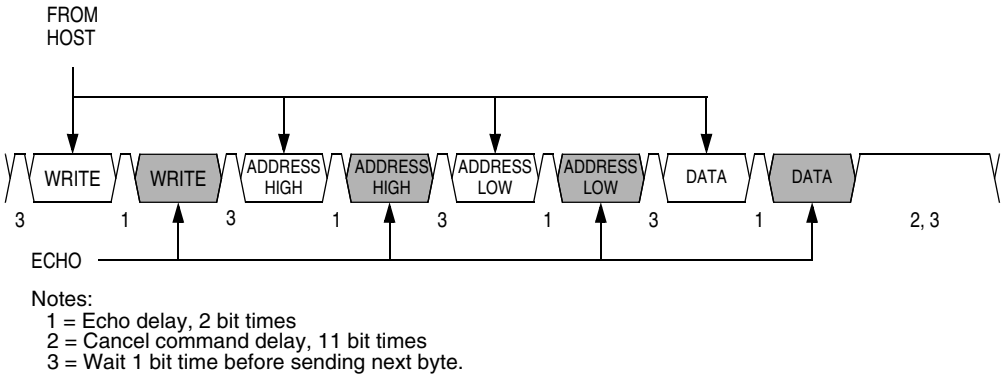


Figure 10-6. Write Transaction

A brief description of each monitor mode command is given in Table 10-4 through Table 10-9.

Table 10-4. READ (Read Memory) Command

Description	Read byte from memory
Operand	2-byte address in high-byte:low-byte order
Data Returned	Returns contents of specified address
Opcode	$4A

Command Sequence

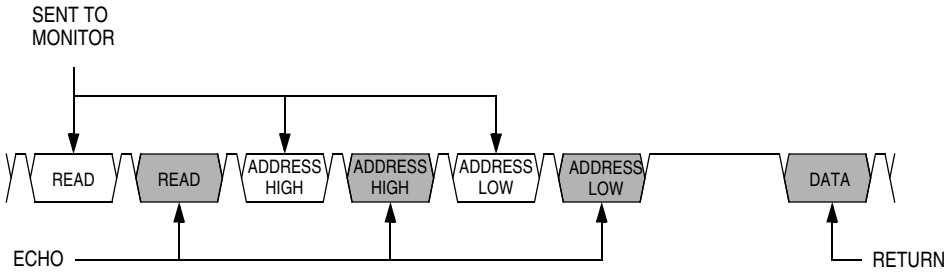


Table 10-5. WRITE (Write Memory) Command

Description	Write byte to memory
Operand	2-byte address in high-byte:low-byte order; low byte followed by data byte
Data Returned	None
Opcode	$49

Command Sequence

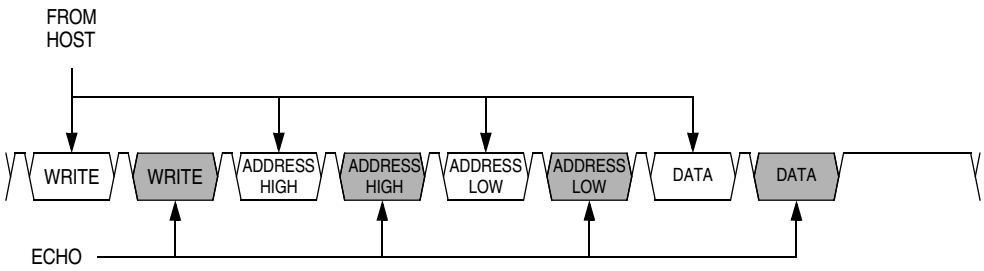


Table 10-6. IREAD (Indexed Read) Command

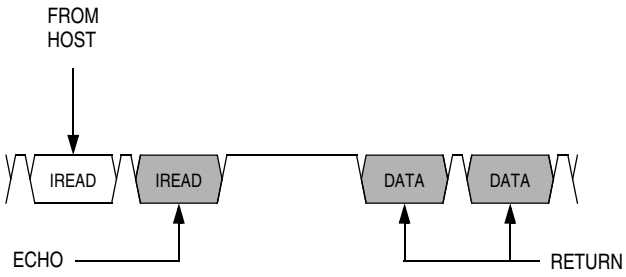
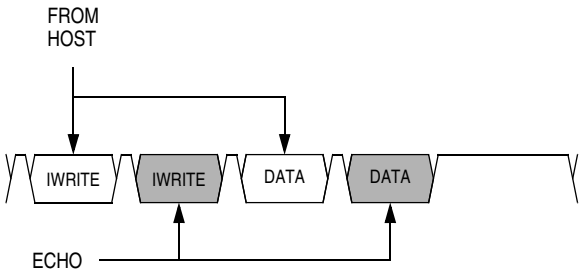
Description	Read next 2 bytes in memory from last address accessed
Operand	2-byte address in high byte:low byte order
Data Returned	Returns contents of next two addresses
Opcode	$1A
Command Sequence	

Table 10-7. IWRITE (Indexed Write) Command

Description	Write to last address accessed + 1
Operand	Single data byte
Data Returned	None
Opcode	$19
Command Sequence	

A sequence of IREAD or IWRITE commands can access a block of memory sequentially over the full 64k-byte memory map.

Table 10-8. READSP (Read Stack Pointer) Command

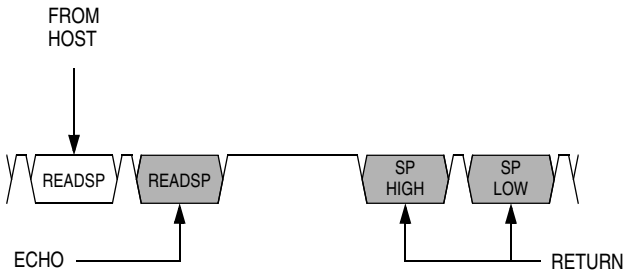
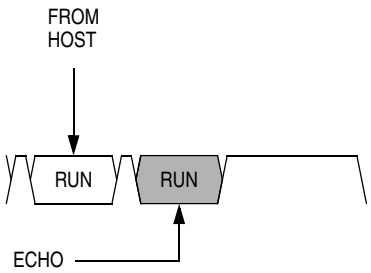
Description	Reads stack pointer
Operand	None
Data Returned	Returns incremented stack pointer value (SP + 1) in high-byte:low-byte order
Opcode	$0C
Command Sequence	

Table 10-9. RUN (Run User Program) Command

Description	Executes PULH and RTI instructions
Operand	None
Data Returned	None
Opcode	$28
Command Sequence	

The MCU executes the SWI and PSHH instructions when it enters monitor mode. The RUN command tells the MCU to execute the PULH and RTI instructions. Before sending the RUN command, the host can modify the stacked CPU registers to prepare to run the host program. The READSP command returns the incremented stack pointer value, SP + 1. The high and low bytes of the program counter are at addresses SP + 5 and SP + 6.

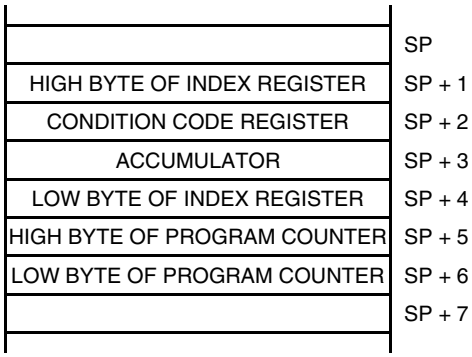


Figure 10-7. Stack Pointer at Monitor Mode Entry

10.5 Security

A security feature discourages unauthorized reading of FLASH locations while in monitor mode. The host can bypass the security feature at monitor mode entry by sending eight security bytes that match the bytes at locations $FFF6–$FFFD. Locations $FFF6–$FFFD contain user-defined data.

NOTE: *Do not leave locations $FFF6–$FFFD blank. For security reasons, program locations $FFF6–$FFFD even if they are not used for vectors.*

During monitor mode entry, the MCU waits after the power-on reset for the host to send the eight security bytes on pin PTA0. If the received bytes match those at locations $FFF6–$FFFD, the host bypasses the security feature and can read all FLASH locations and execute code from FLASH. Security remains bypassed until a power-on reset occurs. If the reset was not a power-on reset, security remains bypassed and security code entry is not required. (See **Figure 10-8**.)

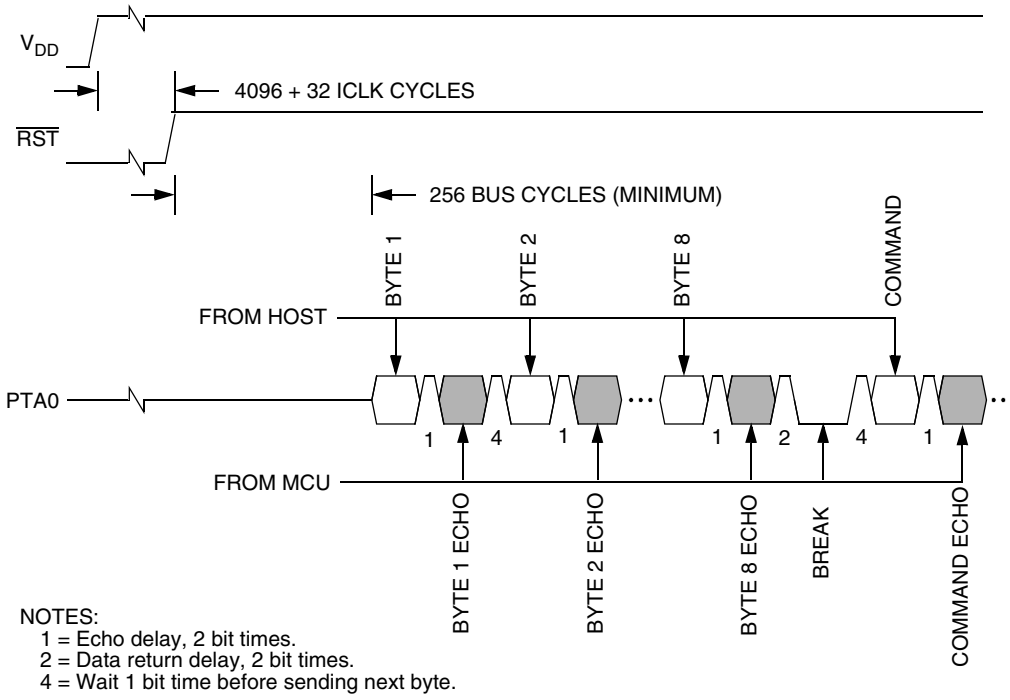


Figure 10-8. Monitor Mode Entry Timing

Upon power-on reset, if the received bytes of the security code do not match the data at locations $FFF6–$FFFD, the host fails to bypass the security feature. The MCU remains in monitor mode, but reading a FLASH location returns an invalid value and trying to execute code from FLASH causes an illegal address reset. After receiving the eight security bytes from the host, the MCU transmits a break character, signifying that it is ready to receive a command.

NOTE: *The MCU does not transmit a break character until after the host sends the eight security bits.*

To determine whether the security code entered is correct, check to see if bit 6 of RAM address $40 is set. If it is, then the correct security code has been entered and FLASH can be accessed.

If the security sequence fails, the device should be reset by a power-on reset and brought up in monitor mode to attempt another entry. After failing the security sequence, the FLASH module can also be mass erased by executing an erase routine that was downloaded into internal RAM. The mass erase operation clears the security code locations so that all eight security bytes become $FF (blank).

Section 11. Timer Interface Module (TIM)

11.1 Contents

11.2 Introduction 182

11.3 Features 182

11.4 Pin Name Conventions 183

11.5 Functional Description 183
11.5.1 TIM Counter Prescaler 187
11.5.2 Input Capture 187
11.5.3 Output Compare 188
11.5.3.1 Unbuffered Output Compare 188
11.5.3.2 Buffered Output Compare 189
11.5.4 Pulse Width Modulation (PWM) 189
11.5.4.1 Unbuffered PWM Signal Generation 190
11.5.4.2 Buffered PWM Signal Generation 191
11.5.4.3 PWM Initialization 192

11.6 Interrupts 193

11.7 Low-Power Modes 193
11.7.1 Wait Mode 194
11.7.2 Stop Mode 194

11.8 TIM During Break Interrupts 194

11.9 I/O Signals 195

11.10 I/O Registers 195
11.10.1 TIM Status and Control Register 196
11.10.2 TIM Counter Registers 198
11.10.3 TIM Counter Modulo Registers 199
11.10.4 TIM Channel Status and Control Registers 200
11.10.5 TIM Channel Registers 203

11.2 Introduction

This section describes the timer interface (TIM) module. The TIM is a two-channel timer that provides a timing reference with input capture, output compare, and pulse-width-modulation functions. **Figure 11-1** is a block diagram of the TIM.

This particular MCU has two timer interface modules which are denoted as TIM1 and TIM2.

11.3 Features

Features of the TIM include:

- Two input capture/output compare channels:
  - Rising-edge, falling-edge, or any-edge input capture trigger
  - Set, clear, or toggle output compare action
- Buffered and unbuffered pulse-width-modulation (PWM) signal generation
- Programmable TIM clock input with 7-frequency internal bus clock prescaler selection
- Free-running or modulo up-count operation
- Toggle any channel pin on overflow
- TIM counter stop and reset bits

11.4 Pin Name Conventions

The text that follows describes both timers, TIM1 and TIM2. The TIM input/output (I/O) pin names are T[1,2]CH0 (timer channel 0) and T[1,2]CH1 (timer channel 1), where “1” is used to indicate TIM1 and “2” is used to indicate TIM2. The two TIMs share four I/O pins with four I/O port pins. The full names of the TIM I/O pins are listed in Table 11-1. The generic pin names appear in the text that follows.

Table 11-1. Pin Name Conventions

TIM Generic Pin Names:		T[1,2]CH0	T[1,2]CH1
Full TIM Pin Names:	TIM1	PTA6/T1CH0	PTA7/T1CH1
	TIM2	PTB4/T2CH0	PTB5/T2CH1

NOTE: *References to either timer 1 or timer 2 may be made in the following text by omitting the timer number. For example, TCH0 may refer generically to T1CH0 and T2CH0, and TCH1 may refer to T1CH1 and T2CH1.*

11.5 Functional Description

Figure 11-1 shows the structure of the TIM. The central component of the TIM is the 16-bit TIM counter that can operate as a free-running counter or a modulo up-counter. The TIM counter provides the timing reference for the input capture and output compare functions. The TIM counter modulo registers, TMODH:TMODL, control the modulo value of the TIM counter. Software can read the TIM counter value at any time without affecting the counting sequence.

The two TIM channels (per timer) are programmable independently as input capture or output compare channels.

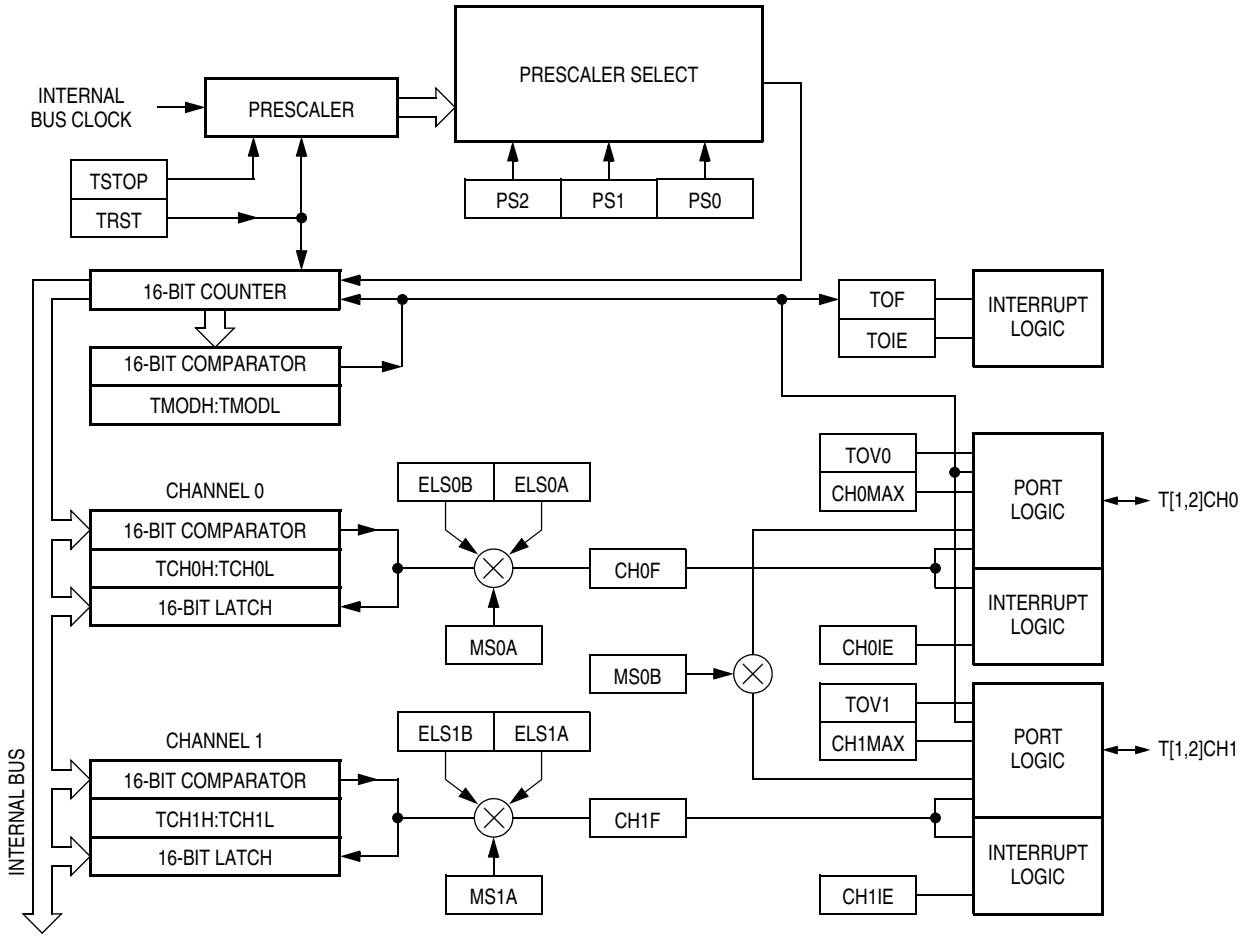


Figure 11-1. TIM Block Diagram

Figure 11-2 summarizes the timer registers.

NOTE: *References to either timer 1 or timer 2 may be made in the following text by omitting the timer number. For example, TSC may generically refer to both T1SC and T2SC.*

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0020	Timer 1 Status and Control Register (T1SC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0021	Timer 1 Counter Register High (T1CNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0022	Timer 1 Counter Register Low (T1CNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0023	Timer 1 Counter Modulo Register High (T1MODH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0024	Timer 1 Counter Modulo Register Low (T1MODL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0025	Timer 1 Channel 0 Status and Control Register (T1SC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0026	Timer 1 Channel 0 Register High (T1CH0H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0027	Timer 1 Channel 0 Register Low (T1CH0L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0028	Timer 1 Channel 1 Status and Control Register (T1SC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-2. TIM I/O Register Summary (Sheet 1 of 3)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0029	Timer 1 Channel 1 Register High (T1CH1H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							
$002A	Timer 1 Channel 1 Register Low (T1CH1L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							
$002B	Timer 2 Status and Control Register (T2SC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$002C	Timer 2 Counter Register High (T2CNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$002D	Timer 2 Counter Register Low (T2CNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$002E	Timer 2 Counter Modulo Register High (T2MODH)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
$002F	Timer 2 Counter Modulo Register Low (T2MODL)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$0030	Timer 2 Channel 0 Status and Control Register (T2SC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0031	Timer 2 Channel 0 Register High (T2CH0H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							

= Unimplemented

Figure 11-2. TIM I/O Register Summary (Sheet 2 of 3)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0032	Timer 2 Channel 0 Register Low (T2CH0L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							
$0033	Timer 2 Channel 1 Status and Control Register (T2SC1)	Read: Write:	CH1F 0	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Reset:	0	0	0	0	0	0	0	0
$0034	Timer 2 Channel 1 Register High (T2CH1H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							
$0035	Timer 2 Channel 1 Register Low (T2CH1L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							

= Unimplemented

Figure 11-2. TIM I/O Register Summary (Sheet 3 of 3)

11.5.1 TIM Counter Prescaler

The TIM clock source can be one of the seven prescaler outputs. The prescaler generates seven clock rates from the internal bus clock. The prescaler select bits, PS[2:0], in the TIM status and control register select the TIM clock source.

11.5.2 Input Capture

With the input capture function, the TIM can capture the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the TIM latches the contents of the TIM counter into the TIM channel registers, TCHxH:TCHxL. The polarity of the active edge is programmable. Input captures can generate TIM CPU interrupt requests.

11.5.3 Output Compare

With the output compare function, the TIM can generate a periodic pulse with a programmable polarity, duration, and frequency. When the counter reaches the value in the registers of an output compare channel, the TIM can set, clear, or toggle the channel pin. Output compares can generate TIM CPU interrupt requests.

11.5.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in **11.5.3 Output Compare**. The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change an output compare value could cause incorrect operation for up to two counter overflow periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that counter overflow period. Also, using a TIM overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIM may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the output compare value on channel x:

- When changing to a smaller value, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current output compare pulse. The interrupt routine has until the end of the counter overflow period to write the new value.
- When changing to a larger output compare value, enable TIM overflow interrupts and write the new value in the TIM overflow interrupt routine. The TIM overflow interrupt occurs at the end of the current counter overflow period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same counter overflow period.

11.5.3.2 Buffered Output Compare

Channels 0 and 1 can be linked to form a buffered output compare channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the output.

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The output compare value in the TIM channel 0 registers initially controls the output on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the output after the TIM overflows. At each subsequent overflow, the TIM channel registers (0 or 1) that control the output are the ones written to last. TSC0 controls and monitors the buffered output compare function, and TIM channel 1 status and control register (TSC1) is unused. While the MS0B bit is set, the channel 1 pin, TCH1, is available as a general-purpose I/O pin.

NOTE: *In buffered output compare operation, do not write new output compare values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered output compares.*

11.5.4 Pulse Width Modulation (PWM)

By using the toggle-on-overflow feature with an output compare channel, the TIM can generate a PWM signal. The value in the TIM counter modulo registers determines the period of the PWM signal. The channel pin toggles when the counter reaches the value in the TIM counter modulo registers. The time between overflows is the period of the PWM signal.

As **Figure 11-3** shows, the output compare value in the TIM channel registers determines the pulse width of the PWM signal. The time between overflow and output compare is the pulse width. Program the TIM to clear the channel pin on output compare if the state of the PWM pulse is logic 1. Program the TIM to set the pin if the state of the PWM pulse is logic 0.

The value in the TIM counter modulo registers and the selected prescaler output determines the frequency of the PWM output. The frequency of an 8-bit PWM signal is variable in 256 increments. Writing $00FF (255) to the TIM counter modulo registers produces a PWM period of 256 times the internal bus clock period if the prescaler select value is $000. See **11.10.1 TIM Status and Control Register**.

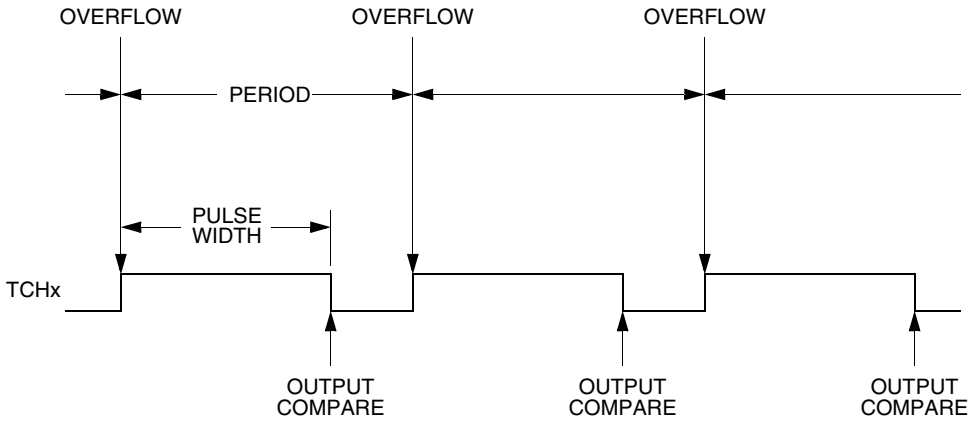


Figure 11-3. PWM Period and Pulse Width

The value in the TIM channel registers determines the pulse width of the PWM output. The pulse width of an 8-bit PWM signal is variable in 256 increments. Writing $0080 (128) to the TIM channel registers produces a duty cycle of 128/256 or 50%.

11.5.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in **11.5.4 Pulse Width Modulation (PWM)**. The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change a pulse width value could cause incorrect operation for up to two PWM periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that PWM period. Also, using a TIM overflow interrupt routine to write a new, smaller pulse width value may cause the compare to be missed. The TIM may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the PWM pulse width on channel x:

- When changing to a shorter pulse width, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current pulse. The interrupt routine has until the end of the PWM period to write the new value.
- When changing to a longer pulse width, enable TIM overflow interrupts and write the new value in the TIM overflow interrupt routine. The TIM overflow interrupt occurs at the end of the current PWM period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same PWM period.

NOTE: *In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare also can cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.*

11.5.4.2 Buffered PWM Signal Generation

Channels 0 and 1 can be linked to form a buffered PWM channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The TIM channel 0 registers initially control the pulse width on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIM channel registers (0 or 1) that control the pulse width are the ones written to last. TSC0 controls and monitors the buffered PWM function, and TIM channel 1 status and control register (TSC1) is unused. While the MS0B bit is set, the channel 1 pin, TCH1, is available as a general-purpose I/O pin.

NOTE: *In buffered PWM signal generation, do not write new pulse width values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered PWM signals.*

11.5.4.3 PWM Initialization

To ensure correct operation when generating unbuffered or buffered PWM signals, use the following initialization procedure:

1. In the TIM status and control register (TSC):
   a. Stop the TIM counter by setting the TIM stop bit, TSTOP.
   b. Reset the TIM counter and prescaler by setting the TIM reset bit, TRST.
2. In the TIM counter modulo registers (TMODH:TMODL), write the value for the required PWM period.
3. In the TIM channel x registers (TCHxH:TCHxL), write the value for the required pulse width.
4. In TIM channel x status and control register (TSCx):
   a. Write 0:1 (for unbuffered output compare or PWM signals) or 1:0 (for buffered output compare or PWM signals) to the mode select bits, MSxB:MSxA. (See **Table 11-3**.)
   b. Write 1 to the toggle-on-overflow bit, TOVx.
   c. Write 1:0 (to clear output on compare) or 1:1 (to set output on compare) to the edge/level select bits, ELSxB:ELSxA. The output action on compare must force the output to the complement of the pulse width level. (See **Table 11-3**.)

NOTE: *In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare can also cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.*

5. In the TIM status control register (TSC), clear the TIM stop bit, TSTOP.

Setting MS0B links channels 0 and 1 and configures them for buffered PWM operation. The TIM channel 0 registers (TCH0H:TCH0L) initially control the buffered PWM output. TIM status control register 0 (TSCR0) controls and monitors the PWM signal from the linked channels.

Clearing the toggle-on-overflow bit, TOVx, inhibits output toggles on TIM overflows. Subsequent output compares try to force the output to a state it is already in and have no effect. The result is a 0% duty cycle output.

Setting the channel x maximum duty cycle bit (CHxMAX) and setting the TOVx bit generates a 100% duty cycle output. (See **11.10.4 TIM Channel Status and Control Registers**.)

11.6 Interrupts

The following TIM sources can generate interrupt requests:

- TIM overflow flag (TOF) — The TOF bit is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. The TIM overflow interrupt enable bit, TOIE, enables TIM overflow CPU interrupt requests. TOF and TOIE are in the TIM status and control register.
- TIM channel flags (CH1F:CH0F) — The CHxF bit is set when an input capture or output compare occurs on channel x. Channel x TIM CPU interrupt requests are controlled by the channel x interrupt enable bit, CHxIE. Channel x TIM CPU interrupt requests are enabled when CHxIE = 1. CHxF and CHxIE are in the TIM channel x status and control register.

11.7 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

11.7.1 Wait Mode

The TIM remains active after the execution of a WAIT instruction. In wait mode, the TIM registers are not accessible by the CPU. Any enabled CPU interrupt request from the TIM can bring the MCU out of wait mode.

If TIM functions are not required during wait mode, reduce power consumption by stopping the TIM before executing the WAIT instruction.

11.7.2 Stop Mode

The TIM is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions or the state of the TIM counter. TIM operation resumes when the MCU exits stop mode after an external interrupt.

11.8 TIM During Break Interrupts

A break interrupt stops the TIM counter.

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state. (See **9.8.3 SIM Break Flag Control Register**.)

To allow software to clear status bits during a break interrupt, write a logic 1 to the BCFE bit. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect status bits during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a 2-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is at logic 0. After the break, doing the second step clears the status bit.

11.9 I/O Signals

Port A and port B each shares two of its pins with the TIM. The four TIM channel I/O pins are T1CH0, T1CH1, T2CH0, and T2CH1 as described in **11.4 Pin Name Conventions**.

Each channel I/O pin is programmable independently as an input capture pin or an output compare pin. T1CH0 and T2CH0 can be configured as buffered output compare or buffered PWM pins.

11.10 I/O Registers

NOTE: *References to either timer 1 or timer 2 may be made in the following text by omitting the timer number. For example, TSC may generically refer to both T1SC AND T2SC.*

These I/O registers control and monitor operation of the TIM:

- TIM status and control register (TSC)
- TIM counter registers (TCNTH:TCNTL)
- TIM counter modulo registers (TMODH:TMODL)
- TIM channel status and control registers (TSC0, TSC1)
- TIM channel registers (TCH0H:TCH0L, TCH1H:TCH1L)

11.10.1 TIM Status and Control Register

The TIM status and control register (TSC):

- Enables TIM overflow interrupts
- Flags TIM overflows
- Stops the TIM counter
- Resets the TIM counter
- Prescales the TIM counter clock

Address: T1SC, $0020 and T2SC, $002B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
Write:	0			TRST				
Reset:	0	0	1	0	0	0	0	0

= Unimplemented

Figure 11-4. TIM Status and Control Register (TSC)

TOF — TIM Overflow Flag Bit

This read/write flag is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. Clear TOF by reading the TIM status and control register when TOF is set and then writing a logic 0 to TOF. If another TIM overflow occurs before the clearing sequence is complete, then writing logic 0 to TOF has no effect. Therefore, a TOF interrupt request cannot be lost due to inadvertent clearing of TOF. Reset clears the TOF bit. Writing a logic 1 to TOF has no effect.

1 = TIM counter has reached modulo value
0 = TIM counter has not reached modulo value

TOIE — TIM Overflow Interrupt Enable Bit

This read/write bit enables TIM overflow interrupts when the TOF bit becomes set. Reset clears the TOIE bit.

1 = TIM overflow interrupts enabled
0 = TIM overflow interrupts disabled

TSTOP — TIM Stop Bit

This read/write bit stops the TIM counter. Counting resumes when TSTOP is cleared. Reset sets the TSTOP bit, stopping the TIM counter until software clears the TSTOP bit.

1 = TIM counter stopped
0 = TIM counter active

NOTE: *Do not set the TSTOP bit before entering wait mode if the TIM is required to exit wait mode.*

TRST — TIM Reset Bit

Setting this write-only bit resets the TIM counter and the TIM prescaler. Setting TRST has no effect on any other registers. Counting resumes from $0000. TRST is cleared automatically after the TIM counter is reset and always reads as logic 0. Reset clears the TRST bit.

1 = Prescaler and TIM counter cleared
0 = No effect

NOTE: *Setting the TSTOP and TRST bits simultaneously stops the TIM counter at a value of $0000.*

PS[2:0] — Prescaler Select Bits

These read/write bits select one of the seven prescaler outputs as the input to the TIM counter as **Table 11-2** shows. Reset clears the PS[2:0] bits.

Table 11-2. Prescaler Selection

PS2	PS1	PS0	TIM Clock Source
0	0	0	Internal bus clock ÷ 1
0	0	1	Internal bus clock ÷ 2
0	1	0	Internal bus clock ÷ 4
0	1	1	Internal bus clock ÷ 8
1	0	0	Internal bus clock ÷ 16
1	0	1	Internal bus clock ÷ 32
1	1	0	Internal bus clock ÷ 64
1	1	1	Not available

11.10.2 TIM Counter Registers

The two read-only TIM counter registers contain the high and low bytes of the value in the TIM counter. Reading the high byte (TCNTH) latches the contents of the low byte (TCNTL) into a buffer. Subsequent reads of TCNTH do not affect the latched TCNTL value until TCNTL is read. Reset clears the TIM counter registers. Setting the TIM reset bit (TRST) also clears the TIM counter registers.

NOTE: *If you read TCNTH during a break interrupt, be sure to unlatch TCNTL by reading TCNTL before exiting the break interrupt. Otherwise, TCNTL retains the value latched during the break.*

Address: T1CNTH, $0021 and T2CNTH, $002C

	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

= Unimplemented

Figure 11-5. TIM Counter Registers High (TCNTH)

Address: T1CNTL, $0022 and T2CNTL, $002D

	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

= Unimplemented

Figure 11-6. TIM Counter Registers Low (TCNTL)

11.10.3 TIM Counter Modulo Registers

The read/write TIM modulo registers contain the modulo value for the TIM counter. When the TIM counter reaches the modulo value, the overflow flag (TOF) becomes set, and the TIM counter resumes counting from $0000 at the next timer clock. Writing to the high byte (TMODH) inhibits the TOF bit and overflow interrupts until the low byte (TMODL) is written. Reset sets the TIM counter modulo registers.

Address: T1MODH, $0023 and T2MODH, $002E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Figure 11-7. TIM Counter Modulo Register High (TMODH)

Address: T1MODL, $0024 and T2MODL, $002F

	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

Figure 11-8. TIM Counter Modulo Register Low (TMODL)

NOTE: *Reset the TIM counter before writing to the TIM counter modulo registers.*

11.10.4 TIM Channel Status and Control Registers

Each of the TIM channel status and control registers:

- Flags input captures and output compares
- Enables input capture and output compare interrupts
- Selects input capture, output compare, or PWM operation
- Selects high, low, or toggling output on output compare
- Selects rising edge, falling edge, or any edge as the active input capture trigger
- Selects output toggling on TIM overflow
- Selects 0% and 100% PWM duty cycle
- Selects buffered or unbuffered output compare/PWM operation

Address: T1SC0, $0025 and T2SC0, $0030

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Figure 11-9. TIM Channel 0 Status and Control Register (TSC0)

Address: T1SC1, $0028 and T2SC1, $0033

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Figure 11-10. TIM Channel 1 Status and Control Register (TSC1)

CHxF — Channel x Flag Bit

When channel x is an input capture channel, this read/write bit is set when an active edge occurs on the channel x pin. When channel x is an output compare channel, CHxF is set when the value in the TIM counter registers matches the value in the TIM channel x registers.

When TIM CPU interrupt requests are enabled (CHxIE = 1), clear CHxF by reading TIM channel x status and control register with CHxF set and then writing a logic 0 to CHxF. If another interrupt request occurs before the clearing sequence is complete, then writing logic 0 to CHxF has no effect. Therefore, an interrupt request cannot be lost due to inadvertent clearing of CHxF.

Reset clears the CHxF bit. Writing a logic 1 to CHxF has no effect.

1 = Input capture or output compare on channel x
0 = No input capture or output compare on channel x

CHxIE — Channel x Interrupt Enable Bit

This read/write bit enables TIM CPU interrupt service requests on channel x.

Reset clears the CHxIE bit.

1 = Channel x CPU interrupt requests enabled
0 = Channel x CPU interrupt requests disabled

MSxB — Mode Select Bit B

This read/write bit selects buffered output compare/PWM operation. MSxB exists only in the TIM1 channel 0 and TIM2 channel 0 status and control registers.

Setting MS0B disables the channel 1 status and control register and reverts TCH1 to general-purpose I/O.

Reset clears the MSxB bit.

1 = Buffered output compare/PWM operation enabled
0 = Buffered output compare/PWM operation disabled

MSxA — Mode Select Bit A

When ELSxB:ELSxA ≠ 0:0, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation. See **Table 11-3**.

1 = Unbuffered output compare/PWM operation
0 = Input capture operation

When ELSxB:ELSxA = 0:0, this read/write bit selects the initial output level of the TCHx pin. See **Table 11-3**. Reset clears the MSxA bit.

1 = Initial output level low
0 = Initial output level high

NOTE: *Before changing a channel function by writing to the MSxB or MSxA bit, set the TSTOP and TRST bits in the TIM status and control register (TSC).*

ELSxB and ELSxA — Edge/Level Select Bits

When channel x is an input capture channel, these read/write bits control the active edge-sensing logic on channel x.

When channel x is an output compare channel, ELSxB and ELSxA control the channel x output behavior when an output compare occurs.

When ELSxB and ELSxA are both clear, channel x is not connected to an I/O port, and pin TCHx is available as a general-purpose I/O pin. **Table 11-3** shows how ELSxB and ELSxA work. Reset clears the ELSxB and ELSxA bits.

Table 11-3. Mode, Edge, and Level Selection

MSxB:MSxA	ELSxB:ELSxA	Mode	Configuration
X0	00	Output preset	Pin under port control; initial output level high
X1	00		Pin under port control; initial output level low
00	01	Input capture	Capture on rising edge only
00	10		Capture on falling edge only
00	11		Capture on rising or falling edge
01	01	Output compare or PWM	Toggle output on compare
01	10		Clear output on compare
01	11		Set output on compare
1X	01	Buffered output compare or buffered PWM	Toggle output on compare
1X	10		Clear output on compare
1X	11		Set output on compare

NOTE: *Before enabling a TIM channel register for input capture operation, make sure that the TCHx pin is stable for at least two bus clocks.*

TOVx — Toggle On Overflow Bit

When channel x is an output compare channel, this read/write bit controls the behavior of the channel x output when the TIM counter overflows. When channel x is an input capture channel, TOVx has no effect.

Reset clears the TOVx bit.

1 = Channel x pin toggles on TIM counter overflow

0 = Channel x pin does not toggle on TIM counter overflow

NOTE: *When TOVx is set, a TIM counter overflow takes precedence over a channel x output compare if both occur at the same time.*

CHxMAX — Channel x Maximum Duty Cycle Bit

When the TOVx bit is at logic 1, setting the CHxMAX bit forces the duty cycle of buffered and unbuffered PWM signals to 100%. As **Figure 11-11** shows, the CHxMAX bit takes effect in the cycle after it is set or cleared. The output stays at the 100% duty cycle level until the cycle after CHxMAX is cleared.

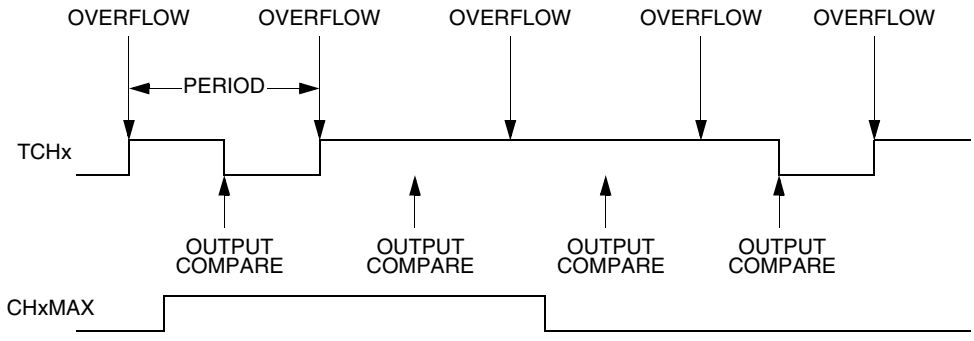


Figure 11-11. CHxMAX Latency

11.10.5 TIM Channel Registers

These read/write registers contain the captured TIM counter value of the input capture function or the output compare value of the output compare function. The state of the TIM channel registers after reset is unknown.

In input capture mode (MSxB:MSxA = 0:0), reading the high byte of the TIM channel x registers (TCHxH) inhibits input captures until the low byte (TCHxL) is read.

In output compare mode (MSxB:MSxA ≠ 0:0), writing to the high byte of the TIM channel x registers (TCHxH) inhibits output compares until the low byte (TCHxL) is written.

Address: T1CH0H, $0026 and T2CH0H, $0031

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
Reset:	Indeterminate after reset							

Figure 11-12. TIM Channel 0 Register High (TCH0H)

Address: T1CH0L, $0027 and T2CH0L $0032

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	Indeterminate after reset							

Figure 11-13. TIM Channel 0 Register Low (TCH0L)

Address: T1CH1H, $0029 and T2CH1H, $0034

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
Reset:	Indeterminate after reset							

Figure 11-14. TIM Channel 1 Register High (TCH1H)

Address: T1CH1L, $002A and T2CH1L, $0035

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	Indeterminate after reset							

Figure 11-15. TIM Channel 1 Register Low (TCH1L)

Section 12. Timebase Module (TBM)

12.1 Contents

12.2 Introduction 205
12.3 Features 205
12.4 Functional Description 206
12.5 Timebase Register Description. 207
12.6 Interrupts. 208
12.7 Low-Power Modes 209
12.7.1 Wait Mode 209
12.7.2 Stop Mode 209

12.2 Introduction

This section describes the timebase module (TBM). The TBM will generate periodic interrupts at user selectable rates using a counter clocked by the selected OSCCLK clock from the oscillator module. This TBM version uses 18 divider stages, eight of which are user selectable.

12.3 Features

Features of the TBM module include:

- Software programmable 8s, 4s, 2s, 1s, 2ms, 1ms, 0.5ms, and 0.25ms periodic interrupt using 32.768-kHz OSCCLK clock
- User selectable oscillator clock source enable during stop mode to allow periodic wake-up from stop

12.4 Functional Description

This module can generate a periodic interrupt by dividing the oscillator clock frequency, OSCCLK. The counter is initialized to all 0s when TBON bit is cleared. The counter, shown in **Figure 12-1**, starts counting when the TBON bit is set. When the counter overflows at the tap selected by TBR2:TBR0, the TBIF bit gets set. If the TBIE bit is set, an interrupt request is sent to the CPU. The TBIF flag is cleared by writing a 1 to the TACK bit. The first time the TBIF flag is set after enabling the timebase module, the interrupt is generated at approximately half of the overflow period. Subsequent events occur at the exact period.

The reference clock OSCCLK is derived from the oscillator module, see **7.3.2 TBM Reference Clock Selection**.

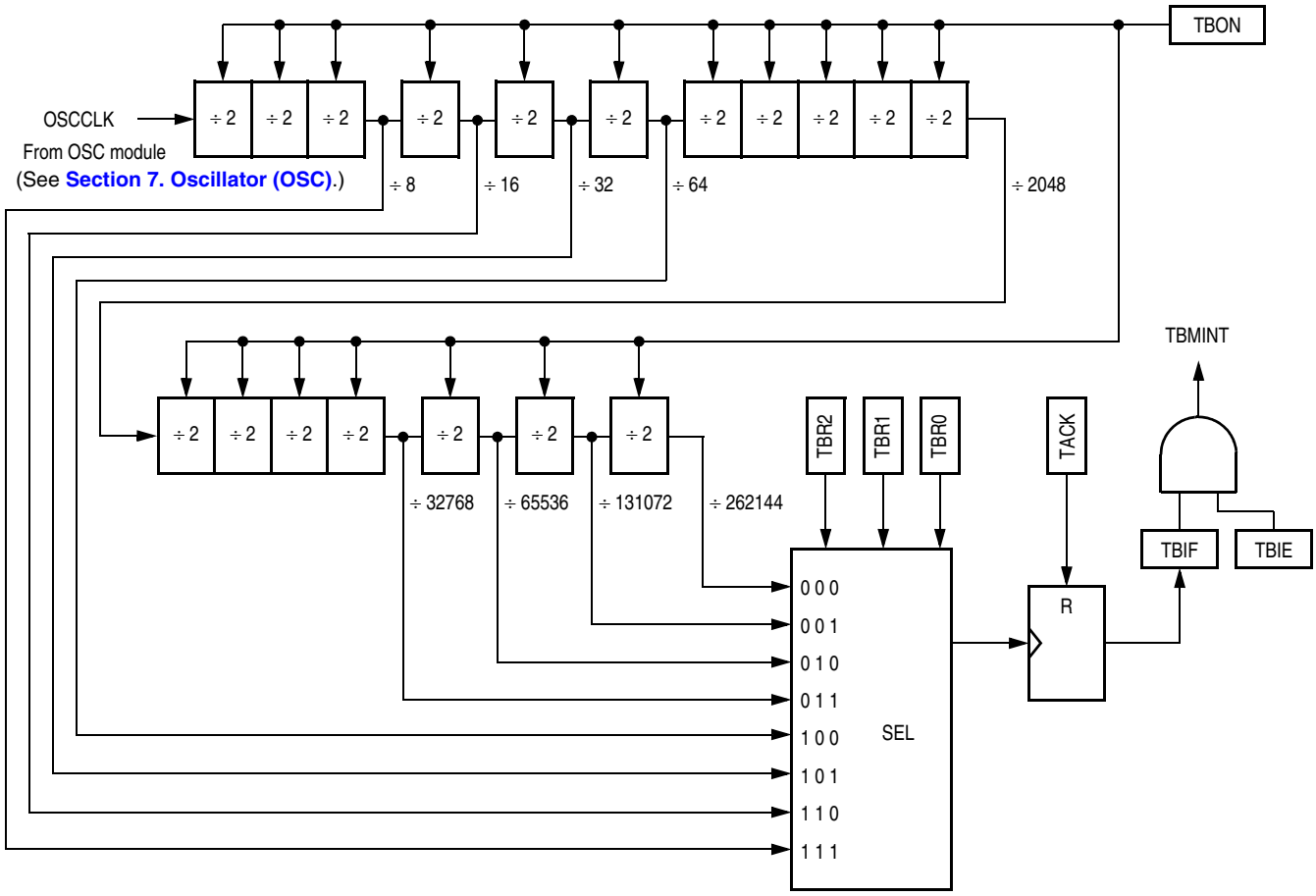


Figure 12-1. Timebase Block Diagram

12.5 Timebase Register Description

The timebase has one register, the TBCR, which is used to enable the timebase interrupts and set the rate.

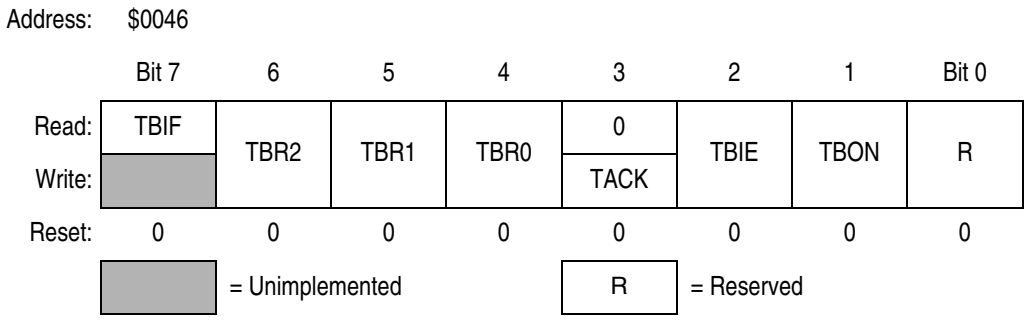


Figure 12-2. Timebase Control Register (TBCR)

TBIF — Timebase Interrupt Flag

This read-only flag bit is set when the timebase counter has rolled over.

1 = Timebase interrupt pending
0 = Timebase interrupt not pending

TBR2–TBR0 — Timebase Rate Selection

These read/write bits are used to select the rate of timebase interrupts as shown in **Table 12-1**.

Table 12-1. Timebase Rate Selection for OSCCLK = 32.768 kHz

TBR2	TBR1	TBR0	Divider	Timebase Interrupt Rate	
				Hz	ms
0	0	0	262144	0.125	8000
0	0	1	131072	0.25	4000
0	1	0	65536	0.5	2000
0	1	1	32768	1	1000
1	0	0	64	512	~2
1	0	1	32	1024	~1
1	1	0	16	2048	~0.5
1	1	1	8	4096	~0.24

NOTE: *Do not change TBR2–TBR0 bits while the timebase is enabled (TBON = 1).*

TACK — Timebase ACKnowledge

The TACK bit is a write-only bit and always reads as 0. Writing a logic 1 to this bit clears TBIF, the timebase interrupt flag bit. Writing a logic 0 to this bit has no effect.

1 = Clear timebase interrupt flag
0 = No effect

TBIE — Timebase Interrupt Enabled

This read/write bit enables the timebase interrupt when the TBIF bit becomes set. Reset clears the TBIE bit.

1 = Timebase interrupt enabled
0 = Timebase interrupt disabled

TBON — Timebase Enabled

This read/write bit enables the timebase. Timebase may be turned off to reduce power consumption when its function is not necessary. The counter can be initialized by clearing and then setting this bit. Reset clears the TBON bit.

1 = Timebase enabled
0 = Timebase disabled and the counter initialized to 0s

12.6 Interrupts

The timebase module can interrupt the CPU on a regular basis with a rate defined by TBR2–TBR0. When the timebase counter chain rolls over, the TBIF flag is set. If the TBIE bit is set, enabling the timebase interrupt, the counter chain overflow will generate a CPU interrupt request. The interrupt vector is defined in **Table 2-1 . Vector Addresses**.

Interrupts must be acknowledged by writing a logic 1 to the TACK bit.

12.7 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

12.7.1 Wait Mode

The timebase module remains active after execution of the WAIT instruction. In wait mode, the timebase register is not accessible by the CPU.

If the timebase functions are not required during wait mode, reduce the power consumption by stopping the timebase before enabling the WAIT instruction.

12.7.2 Stop Mode

The timebase module may remain active after execution of the STOP instruction if the oscillator has been enabled to operate during stop mode through the stop mode oscillator enable bit (STOP_ICLKEN, STOP_RCLKEN, or STOP_XCLKEN) for the selected oscillator in the CONFIG2 register. The timebase module can be used in this mode to generate a periodic walk-up from stop mode.

If the oscillator has not been enabled to operate in stop mode, the timebase module will not be active during STOP mode. In stop mode the timebase register is not accessible by the CPU.

If the timebase functions are not required during stop mode, reduce the power consumption by stopping the timebase before enabling the STOP instruction.

Section 13. Pulse Width Modulator (PWM)

13.1 Contents

13.2 Introduction 211
13.3 Features 212
13.4 PWM Period and Resolution. 214
13.5 PWM Automatic Phase Control 215
13.6 Low-Power Modes 216
13.7 Wait Mode. 216
13.8 Stop Mode 216
13.9 I/O Signals 217
13.10 I/O Registers. 217
13.10.1 PWM Control Register (PWMCR) 217
13.10.2 PWM Clock Control Register (PWMCCR) 218
13.10.3 PWM Data Registers (PWMDR0–PWMDR2) 219
13.10.4 PWM Phase Control Register 220

13.2 Introduction

This section describes the pulse width modulator (PWM) module. The PWM module provides three 8-bit PWM output channels, with an independent 8-bit counter for each channel. The PWM period is equal to $256 \times \frac{1}{P_{CLK}}$ seconds, where P_{CLK} is the PWM counter clock.

For a 32MHz PWM counter clock, the PWM period is 8µs (a PWM frequency of 125kHz). The automatic phase control feature allows phase delays between the channels.
Figure 13-2 shows the structure of the PWM module.

NOTE: *The CGM's PLL must be running (enabled by setting PLLON bit in the PLL control register) if the CGMVCLK is selected for the PWM module input clock. (See* ***Section 8. Clock Generator Module (CGM)****.)*

13.3 Features

Features of the PWM include the following:

- Three independent PWM channels with independent counters
- PWM input clock select
- PWM input clock prescaler
- Automatic phase control

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0051	PWM Control Register (PWMCR)	Read:	PWMEN2	PWMEN1	PWMEN0	0	0	PCH2	PCH1	PCH0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0052	PWM Clock Control Register (PWMCCR)	Read:	PCLKSEL	0	0	0	0	0	PCLK1	PCLK0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0053	PWM Data Register 0 (PWMDR0)	Read:	0PWMD7	0PWMD6	0PWMD5	0PWMD4	0PWMD3	0PWMD2	0PWMD1	0PWMD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0054	PWM Data Register 1 (PWMDR1)	Read:	1PWMD7	1PWMD6	1PWMD5	1PWMD4	1PWMD3	1PWMD2	1PWMD1	1PWMD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0055	PWM Data Register 2 (PWMDR2)	Read:	2PWMD7	2PWMD6	2PWMD5	2PWMD4	2PWMD3	2PWMD2	2PWMD1	2PWMD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0056	PWM Phase Control Register (PWMPCR)	Read:	PHEN	PHD6	PHD5	PHD4	PHD3	PHD2	PHD1	PHD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 13-1. PWM I/O Register Summary

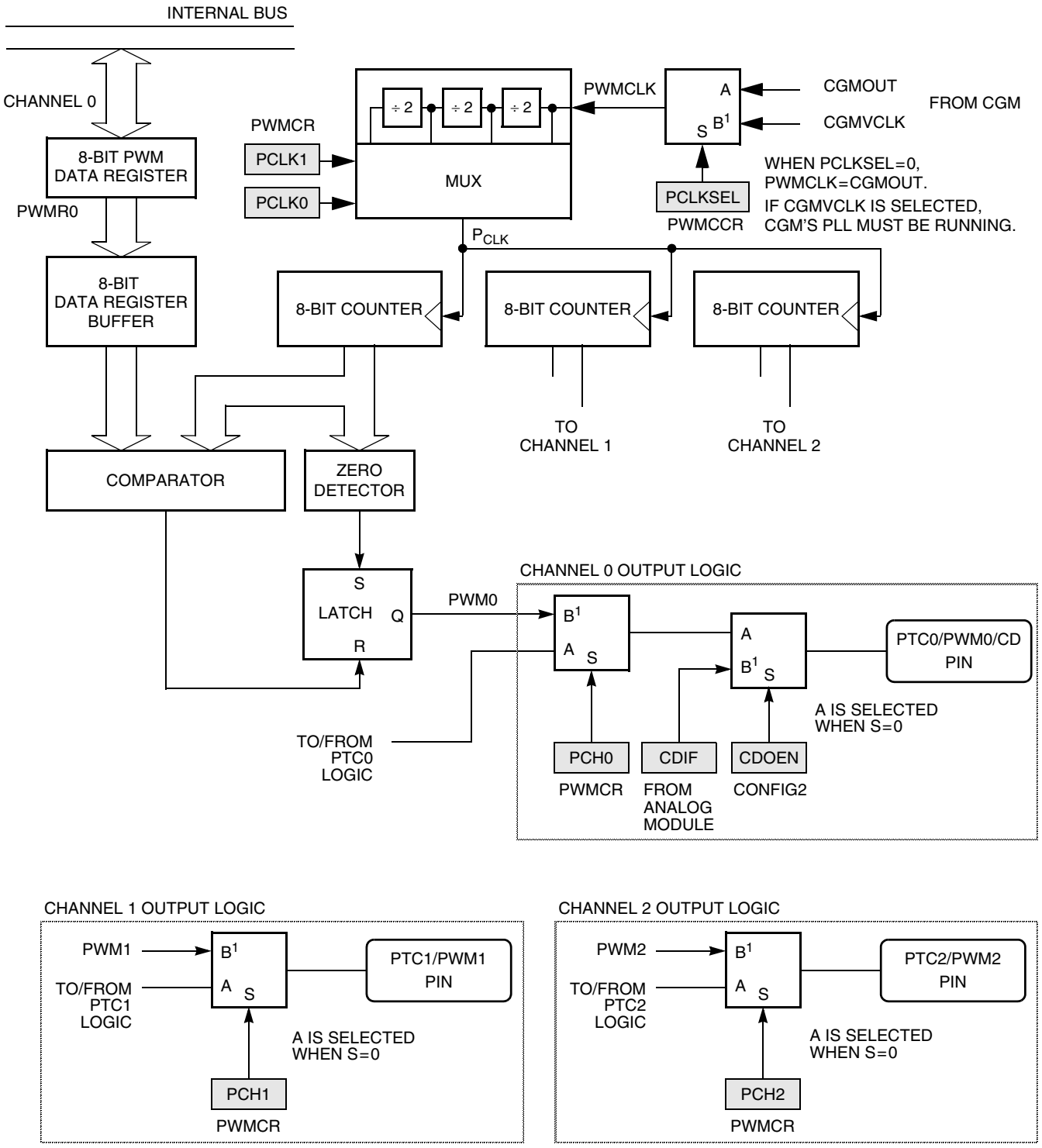


Figure 13-2. PWM Block Diagram

13.4 PWM Period and Resolution

The PWM period is equal to $256 \times \frac{1}{P_{CLK}}$, resolution is $\frac{1}{P_{CLK}}$, where P_{CLK} is the PWM counter clock. The value in the PWM data register (PWMDR) defines the period where the PWM output is high, the low period is equal to 256 minus that value. Each PWM channel has its own counter and I/O control bits so it can be turned on and off independently.

Figure 13-3 shows the PWM output waveforms for a channel with different values in the PWM data register.

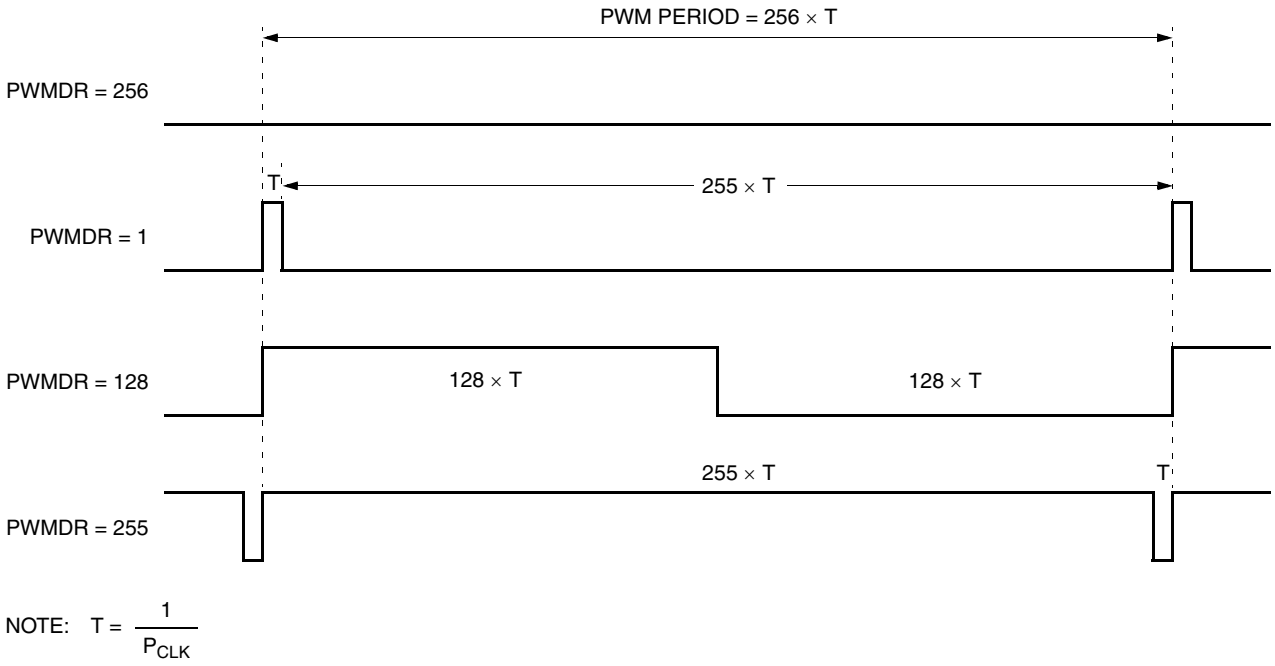


Figure 13-3. PWM Output Waveforms

13.5 PWM Automatic Phase Control

The automatic phase control function allows precise phase difference between the PWM output signals.

Figure 13-4 shows the phase delays between the PWM output signals.

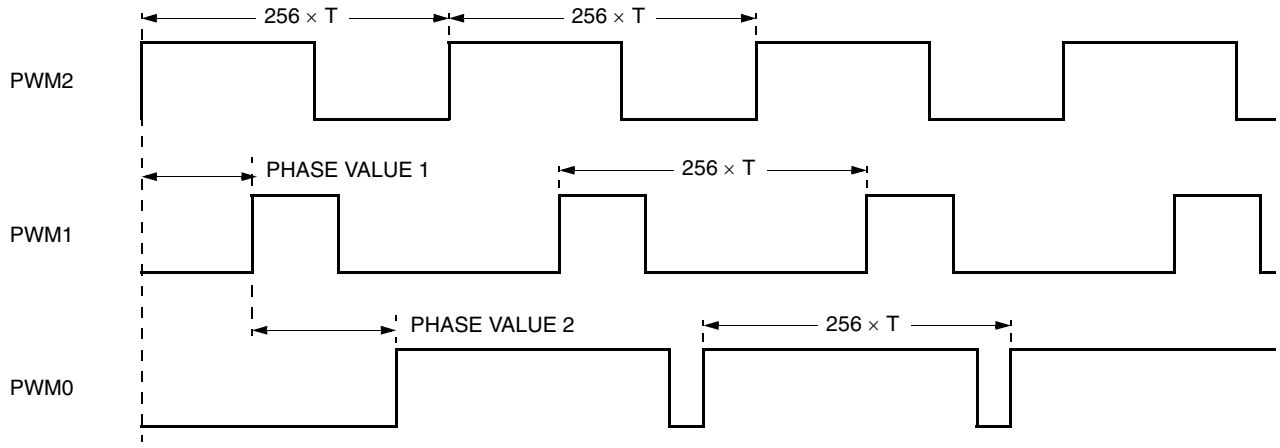


Figure 13-4. PWM Automatic Phase Control

Use the following steps to generate phase difference on PWM channels:

1. Clear PWM enable bits, PWMEN[0:2], to logic 0.
2. Write delay value in PHD[0:6].
3. Set PWM automatic phase control enable bit, PHEN, to logic 1.
4. Set the PWM channel enable bits, PCH[0:2], to logic 1.
5. Set the PWM enable bits, PWMEN[0:2], to logic 1, to enable the PWM counters.

When phase control is enabled, the PWM2 counter will start counting immediately, but the PWM1 and PWM0 counters will be held at zero. After the PWM2 counter reaches the phase value, PH[0:6], the PWM1 counter is released and starts counting. Finally, when the PMW1 counter reaches the phase value, PH[0:6], PWM0 is released and starts counting. It is possible to change the value of PH[0:6] after the PWM1 counter has started and before the start of the PWM0 counter. This way, difference phases can be set between PWM2 and PWM1; PWM1 and PWM0.

The PH[0:6] value is used once to determine the start-up time of the different PWM counters. After that, all PWM counters become free running counters and the phase between the counters will remain unchanged. Changing the value of PH[0:6] after all PWM counters are running has no effect. The counters must first be disabled by clearing the PWM enable bits, PWMEN[0:2], to logic 0, before a new phase value is effective.

Automatic phase control is only available with two (PWM2–PWM1) or three (PWM2–PWM0) PWM channels.

13.6 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

13.7 Wait Mode

The PWM module remains active after the execution of a WAIT instruction. In wait mode, the PWM registers are not accessible by the CPU.

If PWM functions are not required during wait mode, reduce power consumption by disabling the PWM before executing the WAIT instruction.

13.8 Stop Mode

The PWM is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions or the state of the PWM counters and outputs. PWM operation resumes when the MCU exits stop mode after an external interrupt.

13.9 I/O Signals

The PWM module has three output pins shared with port C: PTC0/PWM0/CD, PWM1/PTC1, and PTC2/PWM2. PTC0 is also shared with current flow detect output, CD, of the analog module, see (see **18.5 Port C**).

13.10 I/O Registers

These I/O registers control PWM operation:

- PWM control register (PWMCR)
- PWM clock control register (PWMCCR)
- PWM phase control register (PWMPCR)
- Three PWM data registers (PWMDR0–PWMDR2)

13.10.1 PWM Control Register (PWMCR)

The PWM control register (PWMCR) enables/disables the independent PWM counters and port pins used for the PWM channels.

Address: $0051

Read:	PWMEN2	PWMEN1	PWMEN0	0	0	PCH2	PCH1	PCH0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 13-5. PWM Control Register (PWMCR)

PWMEN2–PWMEN0 — PWM Enable Bits

Writing a 0 to the PWMENx bit clears the corresponding PWM counter and force the PWM channel x output to 0. Reset clears these bits.

1 = PWM channel x enabled

0 = PWM channel x is disabled; PWM counter cleared to zero and PWM channel x output forced to zero

PCH2–PCH0 — PWM Channel Enable Bits

Setting a bit will enable the corresponding port pin to be a PWM output pin. When a bit is set, the DDR bit has no effect on the port function.

1 = Port pin is enabled for PWM output
0 = Port pin is standard I/O pin

Exception for PTC0/PWM0/CD control:

Table 13-1. PTC0 Pin Configuration

Pin	CDOEN Bit ($001D)	PCH0 Bit ($0051)	Pin function
PTC0/PWM0/CD	0	0	PTC0
	0	1	PWM0
	1	X	CD

13.10.2 PWM Clock Control Register (PWMCCR)

The PWM clock control register (PWMCCR) selects and defines the clock to the PWM counter, P_{CLK}.

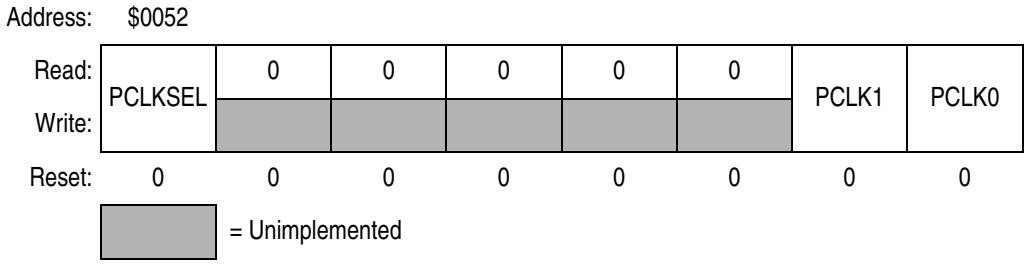


Figure 13-6. PWM Clock Control Register (PWMCCR)

PCLKSEL — PWM Input Clock Select Bit

This bit selects either the CGMOUT or CGMVCLK clock as the input clock to the PWM counters. Reset clears this bit.

1 = Select CGMVCLK as PWM input clock
0 = Select CGMOUT (CPU bus clock) as PWM input clock

PCLK1–PCLK0 — PWM Clock Prescaler Bits

These two bits select the divide ratio used to divide the PWM input clock. Table 13-2 shows the available clock divisions.

Table 13-2. PWM Counter Clock Prescaler Selection

PCLK1	PCLK0	PWM Clock, P_{CLK}
0	0	Source clock ÷ 1
0	1	Source clock ÷ 2
1	0	Source clock ÷ 4
1	1	Source clock ÷ 8

13.10.3 PWM Data Registers (PWMDR0–PWMDR2)

The three PWM data registers (PWMDR0–PWMDR2) defines the high period for corresponding PWM channels.

Address: $0053

Read:	0PWMD7	0PWMD6	0PWMD5	0PWMD4	0PWMD3	0PWMD2	0PWMD1	0PWMD0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 13-7. PWM Data Register 0 (PWMDR0)

Address: $0054

Read:	1PWMD7	1PWMD6	1PWMD5	1PWMD4	1PWMD3	1PWMD2	1PWMD1	1PWMD0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 13-8. PWM Data Register 1 (PWMDR1)

Address: $0055

Read:	2PWMD7	2PWMD6	2PWMD5	2PWMD4	2PWMD3	2PWMD2	2PWMD1	2PWMD0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 13-9. PWM Data Register 2 (PWMDR2)

The value of each PWM data register is continuously compared with the content of a PWM counter to determine the state of each PWM channel output pin.

A value of $00 loaded into these register results in a continuously low output on the corresponding PWM output pin. A value of $80 results in a 50% duty cycle output and so on. The maximum value, $FF correspond to an output which is a "1" for 255/256 of the PWM cycle.

A new value written to the PWM data register will not be effective until the end of the current PWM period. Upon the end of the current PWM period, the contain of the PWM data register is loaded into the PWM data buffer, the value of the PWM data buffer controls the PWM output.

13.10.4 PWM Phase Control Register

The PWM phase control register (PWMPCR) enables the automatic phase control and sets the phase values between the PWM channels.

Address: $0056

	PHEN	PHD6	PHD5	PHD4	PHD3	PHD2	PHD1	PHD0
Read: Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 13-10. PWM Phase Control Register (PWMPCR)

PHEN — PWM Automatic Phase Control Enable Bit

Setting this bit to 1 will enable the automatic phase control function. Reset clears this bit.

1 = Automatic phase control enabled
0 = Automatic phase control disabled

PHD6–PHD0 — PWM Phase Value Bits

This 7-bit phase value is used to determined the start-up time of the different PWM counters when PHEN bit is set. Reset clears these bits.

Section 14. Analog Module

14.1 Contents

14.2 Introduction 221
14.3 Features 222
14.4 Functional Description 223
14.4.1 On-Chip Temperature Sensor 223
14.4.2 Two-Stage Amplifier 224
14.4.3 Amplifier Response Time 224
14.4.4 Current Flow Detection Amplifier 225
14.4.5 Current Flow Detect Output 225
14.5 Interrupts 225
14.6 Low-Power Modes 225
14.6.1 Wait Mode 225
14.6.2 Stop Mode 225
14.7 Analog Module I/O Registers 226
14.7.1 Analog Module Control Register (AMCR) 226
14.7.2 Analog Module Gain Control Register (AMGCR) 227
14.7.3 Analog Module Status and Control Register (AMSCR) . . . 228

14.2 Introduction

This section describes the analog module. The analog module is designed to be use in conjunction with the analog-to-digital converter module for monitoring temperature, charge and discharge currents in smart battery applications.

NOTE: *The analog module uses clock signals from the CGM's PLL, therefore the PLL must be running — PLLON bit in the PLL control register must be set. (See* ***Section 8. Clock Generator Module (CGM)****.)*

14.3 Features

The features of the analog module include the following:

- Temperature sensor
- Current flow detection amplifier
- Two-stage amplifier

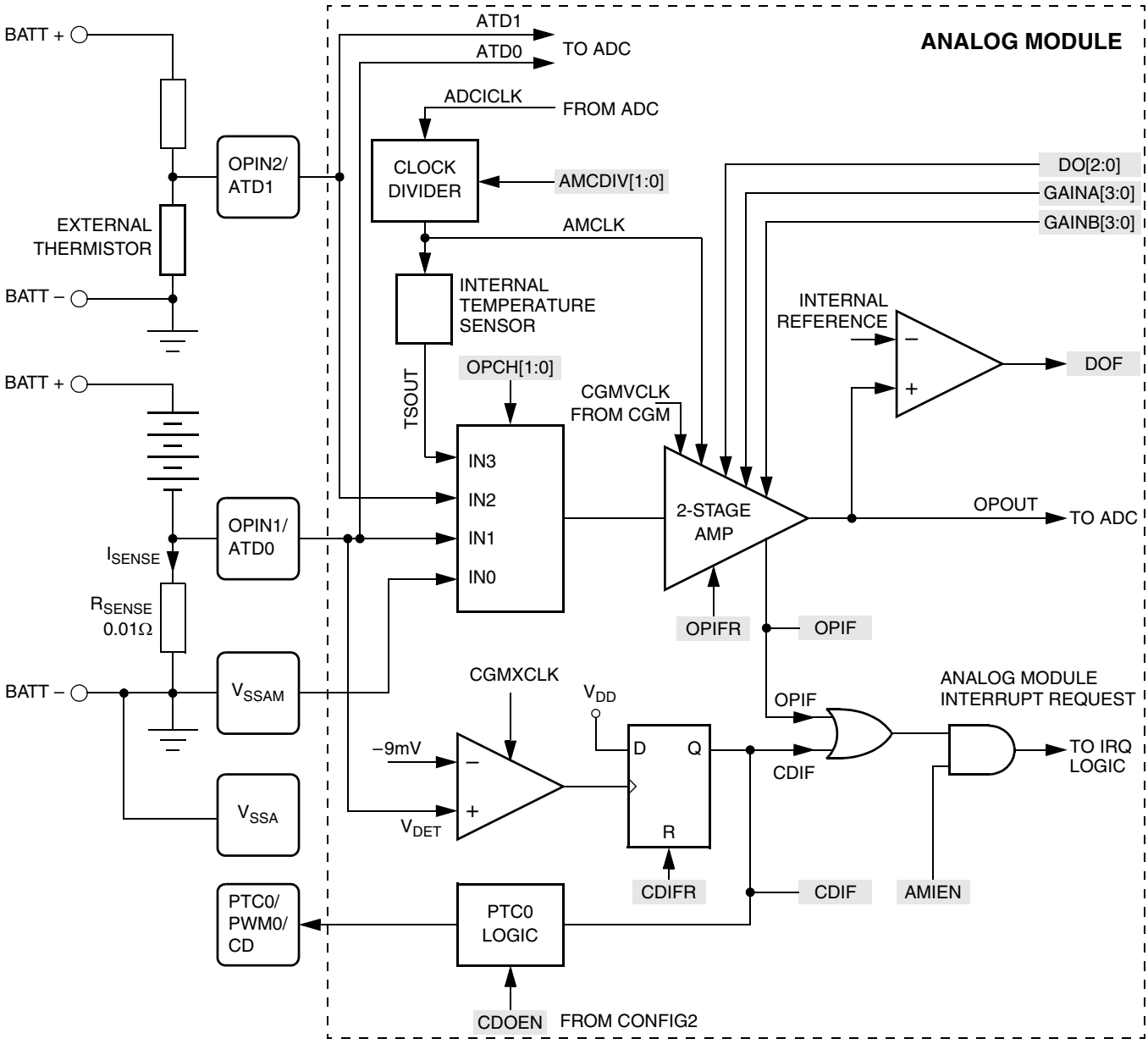


Figure 14-1. Analog Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000E	Analog Module Control Register (AMCR)	Read:	PWR1	PWR0	OPCH1	OPCH0	AMIEN	DO2	DO1	DO0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000F	Analog Module Gain Control Register (AMGCR)	Read:	GAINB3	GAINB2	GAINB1	GAINB0	GAINA3	GAINA2	GAINA1	GAINA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0010	Analog Module Status and Control Register (AMSCR)	Read:	AMCDIV1	AMCDIV0	0	OPIF	0	DOF	0	CDIF
		Write:			OPIFR				CDIFR	
		Reset:	0	0	U	0	0	0	U	0

= Unimplemented U = Unaffected

Figure 14-2. Analog Module I/O Register Summary

14.4 Functional Description

Figure 14-1 shows the block diagram of the analog module. The central component of the analog module is the two-stage gain amplifier used for amplifying the small signals on the analog input pins, OPIN1 and OPIN2. These two signals feed into a multiplexer together with the signal from the internal temperature sensor and a reference input. The selected signal is then fed into the two-stage gain amplifier before going into the analog-to-digital converter (ADC) as OPOUT. The OPIN1 and OPIN2 pins can also feed directly into the ADC as channels ATD0 and ATD1 respectively, without any amplification.

14.4.1 On-Chip Temperature Sensor

The on-chip temperature sensor is designed to measure temperatures from −20 °C to 70 °C. The output of the internal temperature sensor TSOUT is amplified by the two-stage amplifier. The amplified temperature sensor signal is routed to the analog-to-digital converter for analog-to-digital conversion (see Figure 14-1).

14.4.2 Two-Stage Amplifier

The two-stage amplifier is used to amplify small input signals from the on-chip temperature sensor or external voltage sources such as external thermistor and current sensing resistor, for temperature and current monitoring. The amplified signal, OPOUT, is fed to the ADC module for analog-to-digital conversion. The gain of the two-stage amplifier is defined by the GAINAx and GAINBx bits in the analog module gain control register (AMGCR) (see **Figure 14-1**).

14.4.3 Amplifier Response Time

The two-stage amplifier requires the input signal to be stable for sampling. This signal hold-time varies with gain setting for stage-1 of the two-stage amplifier, and is determined by the formula:

$$10 + [(\text{Gain of stage-1 amplifier} - 1) \times 2] \text{ AMCLK cycles}$$

The AMCLK clock is the analog amplifier clock, which is divided from the ADC clock, ADCICLK.

The time for the two-stage amplifier to amplify the input signal to the desired output is dependent on the gain setting in both stages of the two-stage amplifier. The amplifier response time is determined by the formula:

$$70 + (8 \times \text{Gain of stage-1}) + (6 \times \text{Gain of stage-2}) \text{ AMCLK cycles}$$

This amplifier response time should be added to the ADC conversion time to obtain the total time for the small-signal conversion.

Therefore, conversion time for OPINx signals, with amplification is:

$$\text{Amplifier response time} + \text{ADC conversion time}$$

14.4.4 Current Flow Detection Amplifier

The current flow detection amplifier is used to detect charge and discharge current flowing through an external sensing resistor, R_{SENSE}. The current flow detection flag CDIF is set when the voltage at OPIN1 exceeds –9mV (typical) (0.9 ampere when $R_{SENSE} = 0.01\Omega$). When set, CDIF can generate an interrupt request to the CPU when the analog module interrupt enable bit AMIEN is set (see **Figure 14-1**).

14.4.5 Current Flow Detect Output

The current detect flag, CDIF, can be configured for direct control to other external circuitry. When the CDOEN bit in CONFIG2 is set, the status of CDIF is reflected on the PTC0/PWM0/CD pin. (See **5.5 Configuration Register 2 (CONFIG2)** and **18.5 Port C**.)

14.5 Interrupts

When the AMIEN bit is set, the analog module is capable of generating CPU interrupt requests. The interrupt vector is defined in **Table 2-1 . Vector Addresses**.

14.6 Low-Power Modes

The STOP and WAIT instructions put the MCU in low power-consumption standby modes.

14.6.1 Wait Mode

In wait mode the analog module if enabled, continues to operate and may generate an interrupt to trigger the MCU out of wait mode.

14.6.2 Stop Mode

In stop mode, the temperature sensor and the two-stage amplifier are disabled, but the current flow detection amplifier (when enabled) continues to operate if the oscillator is enabled in stop mode. When AMIEN is set, CDIF can be used to wake-up the MCU from the stop mode.

14.7 Analog Module I/O Registers

Three registers control and monitor the operation of the analog module:

- Analog module control register (AMCR) — $000E
- Analog module gain control register (AMGCR) — $000F
- Analog module status and control register (AMSCR) — $0010

14.7.1 Analog Module Control Register (AMCR)

The analog module control register (AMCR):

- Powers on and off analog sub-modules
- Selects the input signal to the two-stage amplifier
- Enables analog module interrupt requests
- Offset adjustment for calibration

Address: $000E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PWR1	PWR0	OPCH1	OPCH0	AMIEN	DO2	DO1	DO0
Reset:	0	0	0	0	0	0	0	0

Figure 14-3. Analog Module Control Register (AMCR)

PWR1–PWR0 — Analog Module Power Control Bits

These read/write bits power on/off the different functions within the analog module. Reset clears the PWR1 and PWR0 bits.

Table 14-1. Analog Module Power Control

PWR1	PWR0	Current Detect Module	Temperature Sensor	Two-Stage Amplifier
0	0	Off	Off	Off
0	1	On	Off	Off
1	0	Off	Off	On
1	1	On	On	On

OPCH1–OPCH0 — Amplifier Channel Select Control Bits

These read/write bits select the input source to be amplified by the two-stage amplifier. Reset clears the OPCH1 and OPCH0 bits.

Table 14-2. Amplifier Channel Select Control bits

OPCH1	OPCH0	Input Source	Comments
0	0	V_{SSAM}	External negative reference
0	1	OPIN1/ATD0	External pin
1	0	OPIN2/ATD1	External pin
1	1	TSOUT (internal)	Internal temperature sensor

AMIEN — Analog Module Interrupt Enable

Setting this bit will enable the CDIF and OPIF flags to generate an CPU interrupt requests. Reset clears the AMIEN bit.

1 = Analog module CPU interrupt requests enabled
0 = Analog module CPU interrupt requests disabled

DO[2:0] — DC Offset Control Bits

Set these bits to zero for optimum analog module performance.

14.7.2 Analog Module Gain Control Register (AMGCR)

The analog module gain control register (AMGCR) selects the two gains for the two-stage amplifier.

Address: $000F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	GAINB3	GAINB2	GAINB1	GAINB0	GAINA3	GAINA2	GAINA1	GAINA0
Reset:	0	0	0	0	0	0	0	0

Figure 14-4. Analog Module Gain Control Register (AMGCR)

GAINB[3:0] — Analog Module 2nd-stage Gain Control Bits

These read/write bits define the 2nd-stage gain of the two-stage amplifier. The overall gain of the amplifier equals the 1st-stage gain multiplied by the 2nd-stage gain. Reset clears the GAINB[3:0] bits.

GAINA[3:0] — Analog Module 1st-stage Gain Control Bits

These read/write bits define the 1st-stage gain of the two-stage amplifier. The overall gain of the amplifier equals the 1st-stage gain multiplied by the 2nd-stage gain. Reset clears the GAINA[3:0] bits.

Table 14-3. Analog Module Gain Values

GAINx3	GAINx2	GAINx1	GAINx0	Amplifier Gain
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

14.7.3 Analog Module Status and Control Register (AMSCR)

The analog module status and control register (AMSCR):

- Selects input clock divider value
- Monitors and clears the amplifier ready interrupt flag
- Monitors DC offset flag
- Monitors and clears the current detect interrupt flag

Address: $0010

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AMCDIV1	AMCDIV0	0	OPIF	0	DOF	0	CDIF
Write:			OPIFR				CDIFR	
Reset:	0	0	U	0	0	0	U	0

= Unimplemented U = Unaffected

Figure 14-5. Analog Module Status and Control Register (AMSCR)

AMCDIV[1:0] — Analog Module Clock Divider Control Bits

These read/write bits select the analog module input clock divider value. The ADC clock, ADICLK, is divided by this value to obtain the AMCLK. Reset clears the AMCDIV[1:0] bits.

Table 14-4. Analog Module Clock Divider Select

AMCDIV1	AMCDIV0	Divider Value
0	0	2
0	1	4
1	0	8
1	1	16

Set AMCDIV1 and AMCDIV0 bits to zero for optimum analog module performance.

OPIFR — Amplifier Ready Interrupt Flag Reset

Writing a logic 1 to this write-only bit clears the OPIF bit. OPIFR always reads as a logic 0. Reset does not affect OPIFR.

1 = Clear OPIF bit
0 = No affect on OPIF bit

OPIF — Amplifier Ready Interrupt Flag

This read-only bit is set when the output of the two-stage amplifier is ready. A CPU interrupt request will be generated if the AMIEN bit is also set. Reset clears OPIF bit.

1 = Two-stage amplifier output is ready
0 = Two-stage amplifier output is not ready

DOF — DC Offset Flag

This is a reserved bit.

CDIFR — Current Detect Interrupt Flag Reset

Writing a logic 1 to this write-only bit clears the CDIF bit. CDIFR always reads as a logic 0. Reset does not affect CDIFR.

1 = Clear CDIF bit
0 = No affect on CDIF bit

CDIF — Current Detect Interrupt Flag

This read-only bit is set when the voltage developed across the sense resistor, R_{SENSE} is equal to or greater than V_{DET} (the current sense amplifier comparator trip voltage, typically –9 mV). CDIF generates an CPU interrupt request if AMIEN bit is also set. The CDIF bit is cleared by writing a logic 1 to the CDIFR bit. Reset clears CDIF bit.

1 = Current detect interrupt has occurred
0 = No current detect interrupt since CDIF last cleared

Section 15. Analog-to-Digital Converter (ADC)

15.1 Contents

15.2 Introduction 232

15.3 Features 232

15.4 Functional Description 234
15.4.1 ADC Port I/O Pins 234
15.4.2 Voltage Conversion 235
15.4.3 Conversion Time 236
15.4.4 Continuous Conversion 237
15.4.5 Auto-scan Mode 237
15.4.6 Result Justification 238
15.4.7 Data Register Interlocking 239
15.4.8 Monotonicity 239

15.5 Interrupts 239

15.6 Low-Power Modes 239
15.6.1 Wait Mode 239
15.6.2 Stop Mode 240

15.7 I/O Signals 240
15.7.1 ADC Voltage In (V_{ADIN}) 240
15.7.2 ADC Analog Power Pin (V_{DDA}) 240
15.7.3 ADC Analog Ground Pin (V_{SSA}) 240
15.7.4 ADC Voltage Reference High Pin (V_{REFH}) 241
15.7.5 ADC Voltage Reference Low Pin (V_{REFL}) 241

15.8 I/O Registers 241
15.8.1 ADC Status and Control Register 242
15.8.2 ADC Clock Control Register 244
15.8.3 ADC Data Register 0 (ADRH0 and ADRL0) 246
15.8.4 ADC Auto-Scan Mode Data Registers (ADRL1–ADRL3) . . 248
15.8.5 ADC Auto-Scan Control Register (ADASCR) 248

15.2 Introduction

This section describes the analog-to-digital converter (ADC). The ADC is a 14-channel 10-bit linear successive approximation ADC.

15.3 Features

Features of the ADC module include:

- Fourteen channels with multiplexed input
- High impedance buffered input
- Linear successive approximation with monotonicity
- 10-bit resolution
- Single or continuous conversion
- Auto-scan conversion on four channels
- Conversion complete flag or conversion complete interrupt
- Selectable ADC clock
- Conversion result justification
  - 8-bit truncated mode
  - Right justified mode
  - Left justified mode
  - Left justified sign mode

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0057	ADC Status and Control Register (ADSCR)	Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
		Write:								
		Reset:	0	0	0	1	1	1	1	1
$0058	ADC Clock Control Register (ADICLK)	Read:	ADIV2	ADIV1	ADIV0	ADICLK	MODE1	MODE0	0	0
		Write:								R
		Reset:	0	0	0	0	0	1	0	0
$0059	ADC Data Register High 0 (ADRH0)	Read:	ADx	ADx	ADx	ADx	ADx	ADx	ADx	ADx
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005A	ADC Data Register Low 0 (ADRL0)	Read:	ADx	ADx	ADx	ADx	ADx	ADx	ADx	ADx
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005B	ADC Data Register Low 1 (ADRL1)	Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005C	ADC Data Register Low 2 (ADRL3)	Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005D	ADC Data Register Low 3 (ADRL3)	Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005E	ADC Auto-scan Control Register (ADASCR)	Read:	0	0	0	0	0	AUTO1	AUTO0	ASCAN
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved

Figure 15-1. ADC I/O Register Summary

15.4 Functional Description

The ADC provides thirteen pins for sampling external sources at pins PTA0/ATD2–PTA5/ATD7, PTC3/ATD8–PTC7/ATD12, and OPIN1–OPIN2; one internal source from the analog module. An analog multiplexer allows the single ADC converter to select one of fourteen ADC channels as ADC voltage in (V_{ADIN}). V_{ADIN} is converted by the successive approximation register-based analog-to-digital converter. When the conversion is completed, ADC places the result in the ADC data register, high and low byte (ADRH0 and ADRL0), and sets a flag or generates an interrupt.

An additional three ADC data registers (ADRL1–ADRL3) are available to store the individual converted data for ADC channels ATD1–ATD3 when the auto-scan mode is enabled. Data from channel ATD0 is stored in ADRL0 in the auto-scan mode.

Figure 15-2 shows the structure of the ADC module.

15.4.1 ADC Port I/O Pins

PTA0–PTA5 and PTC3–PTC7 are general-purpose I/O pins that are shared with the ADC channels, OPIN1 and OPIN2 are two analog inputs that are always connected to the ADC channel select multiplexer. The channel select bits, ADCH[4:0], define which ADC channel/port pin will be used as the input signal. The ADC overrides the port I/O logic by forcing that pin as input to the ADC. The remaining ADC channels/port pins are controlled by the port I/O logic and can be used as general-purpose I/O pins. Writes to the port data register or data direction register will not have any affect on the port pin that is selected by the ADC. Read of a port pin which is in use by the ADC will return the pin condition if the corresponding DDR bit is at logic 0. If the DDR bit is at logic 1, the value in the port data latch is read.

15.4.2 Voltage Conversion

When the input voltage to the ADC equals V_{REFH}, the ADC converts the signal to $3FF (full scale). If the input voltage equals V_{REFL}, the ADC converts it to $000. Input voltages between V_{REFH} and V_{REFL} is a straight-line linear conversion. All other input voltages will result in $3FF if greater than V_{REFH} and $000 if less than V_{REFL}.

NOTE: *Input voltage should not exceed the analog supply voltages.*

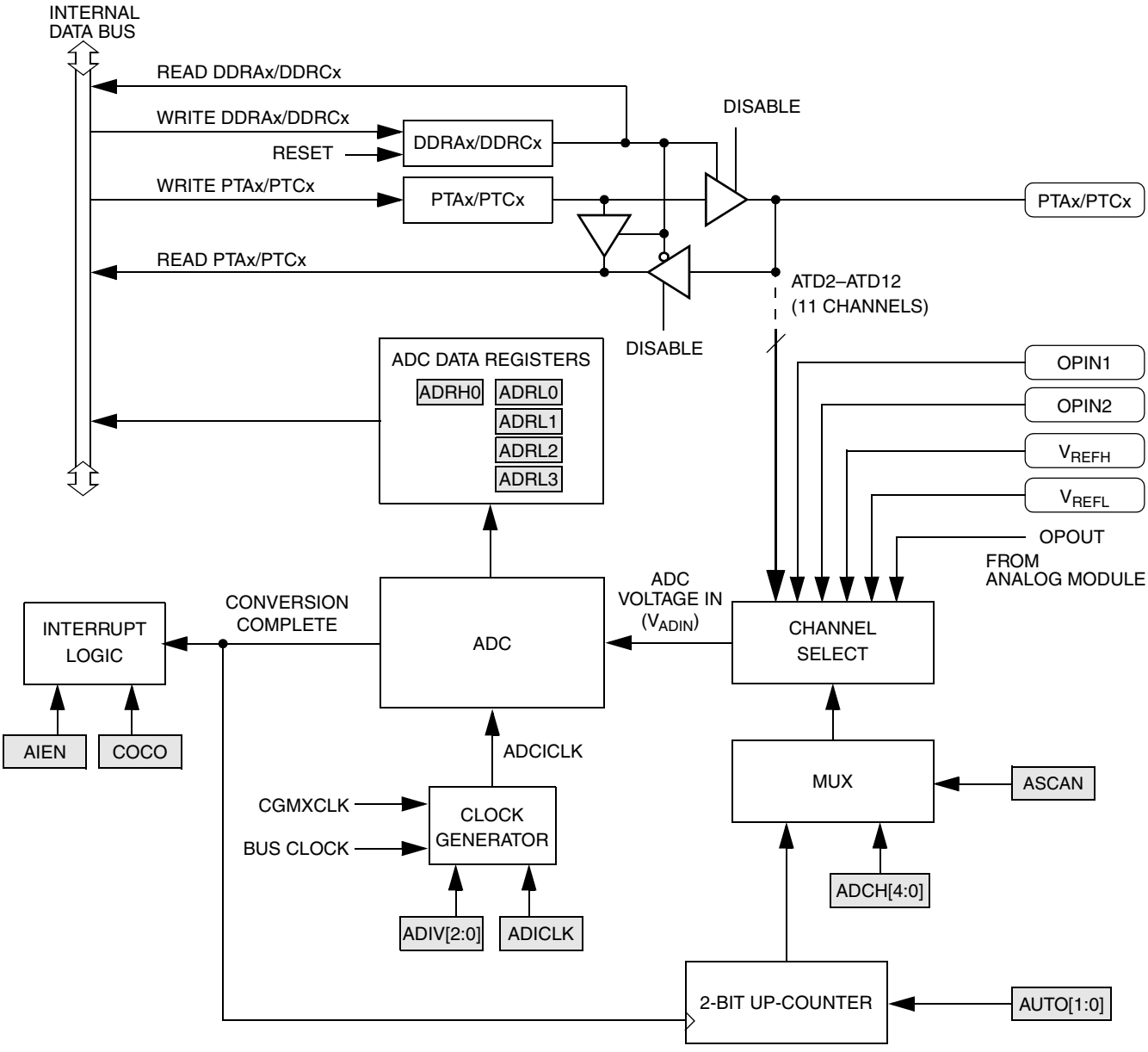


Figure 15-2. ADC Block Diagram

15.4.3 Conversion Time

Conversion starts after a write to the ADSCR. One conversion will take between 16 and 17 ADC clock cycles, therefore:

$$\text{Conversion time} = \frac{\text{16 to17 ADC cycles}}{\text{ADC frequency}}$$

$$\text{Number of bus cycles} = \text{conversion time} \times \text{bus frequency}$$

The ADC conversion time is determined by the clock source chosen and the divide ratio selected. The clock source is either the bus clock or CGMXCLK and is selectable by the ADICLK bit located in the ADC clock register. The divide ratio is selected by the ADIV[2:0] bits.

For example, if a 4MHz CGMXCLK is selected as the ADC input clock source, with a divide-by-four prescale, and the bus speed is set at 2MHz:

$$\text{Conversion time} = \frac{\text{16 to17 ADC cycles}}{\text{4MHz} \div 4} = 16 \text{ to } 17\ \mu\text{s}$$

$$\text{Number of bus cycles} = 16\ \mu\text{s} \times \text{2MHz} = 32 \text{ to } 34 \text{ cycles}$$

NOTE: *The ADC frequency must be between f_{ADIC} minimum and f_{ADIC} maximum to meet ADC specifications. (See **24.12 5.0V ADC Electrical Characteristics**.)*

Since an ADC cycle may comprised of several bus cycles (two in the previous example) and the start of a conversion is initiated by a bus cycle write to the ADSCR, from zero to two additional bus cycles may occur before the start of the initial ADC cycle. This results in a fractional ADC cycle and is represented as the 17th cycle.

NOTE: *When OPOUT is selected as the ADC input, V_{ADIN}, the conversion time is the accumulation of the op-amp settling time and the normal ADC conversion time. After writing to the ADSCR to initiate a conversion cycle, the ADC module sends a signal to the analog module for a OPOUT output. A signal will be sent back to the ADC by the analog module to indicate that OPOUT signal is ready for sampling. Upon receiving this signal, the ADC module starts its normal conversion cycle. (See **24.12 5.0V ADC Electrical Characteristics**.)*

15.4.4 Continuous Conversion

In the continuous conversion mode, the ADC continuously converts the selected channel, filling the ADC data register with new data after each conversion. Data from the previous conversion will be overwritten whether that data has been read or not. Conversions will continue until the ADCO bit is cleared. The COCO bit is set after each conversion and can be cleared by writing to the ADC status and control register or reading of the ADRL0 data register.

15.4.5 Auto-scan Mode

In auto-scan mode, the ADC input channel is selected by the value of the 2-bit up-counter, instead of the channel select bits, ADCH[4:0]. The value of the counter also defines the data register ADRLx to be used to store the conversion result. When ASCAN bit is set, a write to ADC status and control register (ADSCR) will reset the auto-scan up-counter and ADC conversion will start on the channel 0 up to the channel number defined by the integer value of AUTO[1:0]. After a channel conversion is completed, data is stored in ADRLx and the COCO-bit will be set. The counter value will be incremented by 1 and a new conversion will start. This process will continue until the counter value reaches the value of AUTO[1:0]. When this happens, it indicates that the current channel is the last channel to be converted. Upon the completion on the last channel, the counter value will not be incremented and no further conversion will be performed. To start another auto-scan cycle, a write to ADSCR must be performed.

NOTE: *The system only provides 8-bit data storage in auto-scan code, user must clear MODE[1:0] bits to select 8-bit truncation mode before entering auto-scan mode.*

It is recommended that user should disable the auto-scan function before switching channel and also before entering STOP mode.

15.4.6 Result Justification

The conversion result may be formatted in four different ways.

- Left justified
- Right justified
- Left justified sign data mode
- 8-bit truncation

All four of these modes are controlled using MODE0 and MODE1 bits located in the ADC clock control register (ADICLK).

Left justification will place the eight most significant bits (MSB) in the corresponding ADC data register high (ADRH). This may be useful if the result is to be treated as an 8-bit result where the least significant two bits, located in the ADC data register low (ADRL) can be ignored. However, you must read ADRL after ADRH or else the interlocking will prevent all new conversions from being stored.

Right justification will place only the two MSBs in the corresponding ADC data register high (ADRH) and the eight LSB bits in ADC data register low (ADRL). This mode of operation typically is used when a 10-bit unsigned result is desired.

Left justified sign data mode is similar to left justified mode with one exception. The MSB of the 10-bit result, AD9 located in ADRH is complemented. This mode of operation is useful when a result, represented as a signed magnitude from mid-scale, is needed.

Finally, 8-bit truncation mode will place the eight MSBs in ADC data register low (ADRL). The two LSBs are dropped. This mode of operation is used when compatibility with 8-bit ADC designs are required. No interlocking between ADRH and ADRL is present.

15.4.7 Data Register Interlocking

Reading ADRH in any 10-bit mode latches the contents of ADRL until ADRL is read. Until ADRL is read all subsequent ADC results will be lost. This register interlocking can also be reset by a write to the ADC status and control register, or ADC clock control register. A power-on reset or reset will also clear the interlocking. Note that an external conversion request will not reset the lock.

15.4.8 Monotonicity

The conversion process is monotonic and has no missing codes.

15.5 Interrupts

When the AIEN bit is set, the ADC module is capable of generating a CPU interrupt after each ADC conversion or after an auto-scan conversion cycle. A CPU interrupt is generated if the COCO bit is at logic 0. The COCO bit is not used as a conversion complete flag when interrupts are enabled. The interrupt vector is defined in **Table 2-1 . Vector Addresses**.

15.6 Low-Power Modes

The STOP and WAIT instructions put the MCU in low power-consumption standby modes.

15.6.1 Wait Mode

The ADC continues normal operation in wait mode. Any enabled CPU interrupt request from the ADC can bring the MCU out of wait mode. If the ADC is not required to bring the MCU out of wait mode, power down the ADC by setting the ADCH[4:0] bits to logic 1’s before executing the WAIT instruction.

15.6.2 Stop Mode

The ADC module is inactive after the execution of a STOP instruction. Any pending conversion is aborted. ADC conversions resume when the MCU exits stop mode. Allow one conversion cycle to stabilize the analog circuitry before attempting a new ADC conversion after exiting stop mode.

15.7 I/O Signals

The ADC module has fourteen channels, eleven channels are shared with port A and port C I/O pins; two channels are analog pins, OPIN1 and OPIN2, that are shared with the analog module; and one channel, OPOUT, from the analog module.

15.7.1 ADC Voltage In (V_{ADIN})

V_{ADIN} is the input voltage signal from one of the fourteen channels to the ADC module.

15.7.2 ADC Analog Power Pin (V_{DDA})

The ADC analog portion uses V_{DDA} as its power pin. Connect the V_{DDA} pin to the same voltage potential as V_{DD}. External filtering may be necessary to ensure clean V_{DDA} for good results.

NOTE: *Route V_{DDA} carefully for maximum noise immunity and place bypass capacitors as close as possible to the package.*

15.7.3 ADC Analog Ground Pin (V_{SSA})

The ADC analog portion uses V_{SSA} as its ground pin. Connect the V_{SSA} pin to the same voltage potential as V_{SS}.

15.7.4 ADC Voltage Reference High Pin (V_{REFH})

V_{REFH} is the power supply for setting the reference voltage V_{REFH}. Connect the V_{REFH} pin to the same voltage potential as V_{DDA}. There will be a finite current associated with V_{REFH} (see **Section 24. Electrical Specifications**).

NOTE: *Route V_{REFH} carefully for maximum noise immunity and place bypass capacitors as close as possible to the package.*

15.7.5 ADC Voltage Reference Low Pin (V_{REFL})

V_{REFL} is the lower reference supply for the ADC. Connect the V_{REFL} pin to the same voltage potential as V_{SSA}. There will be a finite current associated with V_{REFL} (see **Section 24. Electrical Specifications**).

15.8 I/O Registers

These I/O registers control and monitor ADC operation:

- ADC status and control register (ADSCR) — $0057
- ADC clock control register (ADICLK) — $0058
- ADC data register high 0 (ADRH0) — $0059
- ADC data register low 0 (ADRL0) — $005A
- ADC data register low 1 (ADRL1) — $005B
- ADC data register low 2 (ADRL2) — $005C
- ADC data register low 3 (ADRL3) — $005D
- ADC auto-scan control register (ADASCR) — $005E

15.8.1 ADC Status and Control Register

Function of the ADC status and control register is described here.

Address:	$0057							
Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
Write:								
Reset:	0	0	0	1	1	1	1	1

= Unimplemented

Figure 15-3. ADC Status and Control Register (ADSCR)

COCO — Conversions Complete Bit

When the AIEN bit is a logic 0, the COCO is a read-only bit which is set each time a conversion is completed. This bit is cleared whenever the ADSCR is written, or whenever the ADC clock control register is written, or whenever the ADC data register low, ADRLx, is read.

If the AIEN bit is logic 1, the COCO bit always read as logic 0. ADC interrupt will be generated at the end if an ADC conversion. Reset clears the COCO bit.

1 = Conversion completed (AIEN = 0)
0 = Conversion not completed (AIEN = 0)/CPU interrupt (AIEN=1)

AIEN — ADC Interrupt Enable Bit

When this bit is set, an interrupt is generated at the end of an ADC conversion. The interrupt signal is cleared when the data register, ADR0, is read or the ADSCR is written. Reset clears the AIEN bit.

1 = ADC interrupt enabled
0 = ADC interrupt disabled

ADCO — ADC Continuous Conversion Bit

When set, the ADC will convert samples continuously and update the ADC data register at the end of each conversion. Only one conversion is allowed when this bit is cleared. Reset clears the ADCO bit.

1 = Continuous ADC conversion
0 = One ADC conversion

This bit should not be set when auto-scan mode is enabled; i.e. when ASCAN=1.

ADCH[4:0] — ADC Channel Select Bits

ADCH[4:0] form a 5-bit field which is used to select one of the ADC channels when not in auto-scan mode. The five channel select bits are detailed in **Table 15-1**.

NOTE: Care should be taken when using a port pin as both an analog and a digital input simultaneously to prevent switching noise from corrupting the analog signal.

NOTE: Recovery from the disabled state requires one conversion cycle to stabilize.

Table 15-1. MUX Channel Select

ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	ADC Channel	Input Select
0	0	0	0	0	ATD0	OPIN1
0	0	0	0	1	ATD1	OPIN2
0	0	0	1	0	ATD2	PTA0
0	0	0	1	1	ATD3	PTA1
0	0	1	0	0	ATD4	PTA2
0	0	1	0	1	ATD5	PTA3
0	0	1	1	0	ATD6	PTA4
0	0	1	1	1	ATD7	PTA5
0	1	0	0	0	ATD8	PTC3
0	1	0	0	1	ATD9	PTC4
0	1	0	1	0	ATD10	PTC5
0	1	0	1	1	ATD11	PTC6
0	1	1	0	0	ATD12	PTC7
0	1	1	0	1	ATD13	OPOUT
0 ↓ 1	1 ↓ 1	0 ↓ 1	0 ↓ 0	0 ↓ 0	ATD14 ↓ ATD28	Reserved
1	1	1	0	1	ATD29	V_{REFH} (see Note 2)
1	1	1	1	0	ATD30	V_{REFL} (see Note 2)
1	1	1	1	1	ADC powered-off	—

NOTES:
1. If any unused channels are selected, the resulting ADC conversion will be unknown.
2. The voltage levels supplied from internal reference nodes as specified in the table are used to verify the operation of the ADC converter both in production test and for user applications.

15.8.2 ADC Clock Control Register

The ADC clock control register (ADICLK) selects the clock frequency for the ADC.

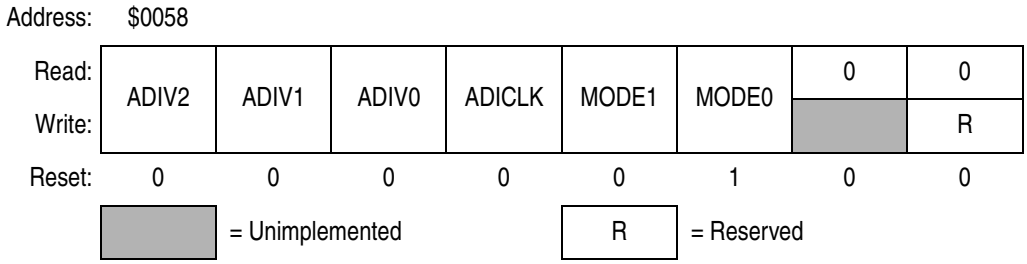


Figure 15-4. ADC Clock Control Register (ADICLK)

ADIV[2:0] — ADC Clock Prescaler Bits

ADIV2, ADIV1, and ADIV0 form a 3-bit field which selects the divide ratio used by the ADC to generate the internal ADC clock. Table 15-2 shows the available clock configurations. The ADC clock should be set to between 500kHz and 2MHz.

Table 15-2. ADC Clock Divide Ratio

ADIV2	ADIV1	ADIV0	ADC Clock Rate
0	0	0	ADC input clock ÷ 1
0	0	1	ADC input clock ÷ 2
0	1	0	ADC input clock ÷ 4
0	1	1	ADC input clock ÷ 8
1	X	X	ADC input clock ÷ 16

X = don't care

ADICLK — ADC Input Clock Select Bit

ADICLK selects either bus clock or CGMXCLK as the input clock source to generate the internal ADC clock. Reset selects CGMXCLK as the ADC clock source.

If the external clock (CGMXCLK) is equal to or greater than 1 MHz, CGMXCLK can be used as the clock source for the ADC. If CGMXCLK is less than 1 MHz, use the PLL-generated bus clock as the clock source. As long as the internal ADC clock is at f_{ADIC}, correct operation can be guaranteed.

1 = Internal bus clock
0 = External clock, CGMXCLK

$$f_{ADIC} = \frac{\text{CGMXCLK or bus frequency}}{\text{ADIV[2:0]}}$$

MODE1 and MODE0 — Modes of Result Justification

MODE1 and MODE0 selects between four modes of operation. The manner in which the ADC conversion results will be placed in the ADC data registers is controlled by these modes of operation. Reset returns right-justified mode.

Table 15-3. ADC Mode Select

MODE1	MODE0	Justification Mode
0	0	8-bit truncated mode
0	1	Right justified mode
1	0	Left justified mode
1	1	Left justified sign data mode

15.8.3 ADC Data Register 0 (ADRH0 and ADRL0)

The ADC data register 0 consist of a pair of 8-bit registers: high byte (ADRH0), and low byte (ADRL0). This pair form a 16-bit register to store the 10-bit ADC result for the selected ADC result justification mode.

In 8-bit truncated mode, the ADRL0 holds the eight most significant bits (MSBs) of the 10-bit result. The ADRL0 is updated each time an ADC conversion completes. In 8-bit truncated mode, ADRL0 contains no interlocking with ADRH0. (See **Figure 15-5 . ADRH0 and ADRL0 in 8-Bit Truncated Mode**.)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0059	ADC Data Register High 0 (ADRH0)	Read:	0	0	0	0	0	0	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005A	ADC Data Register Low 0 (ADRL0)	Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0

Figure 15-5. ADRH0 and ADRL0 in 8-Bit Truncated Mode

In right justified mode the ADRH0 holds the two MSBs, and the ADRL0 holds the eight least significant bits (LSBs), of the 10-bit result. ADRH0 and ADRL0 are updated each time a single channel ADC conversion completes. Reading ADRH0 latches the contents of ADRL0. Until ADRL0 is read all subsequent ADC results will be lost.
(See **Figure 15-6 . ADRH0 and ADRL0 in Right Justified Mode**.)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0059	ADC Data Register High 0 (ADRH0)	Read:	0	0	0	0	0	0	AD9	AD8
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005A	ADC Data Register Low 0 (ADRL0)	Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0

Figure 15-6. ADRH0 and ADRL0 in Right Justified Mode

In left justified mode the ADRH0 holds the eight most significant bits (MSBs), and the ADRL0 holds the two least significant bits (LSBs), of the 10-bit result. The ADRH0 and ADRL0 are updated each time a single channel ADC conversion completes. Reading ADRH0 latches the contents of ADRL0. Until ADRL0 is read all subsequent ADC results will be lost. (See **Figure 15-7 . ADRH0 and ADRL0 in Left Justified Mode**.)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0059	ADC Data Register High 0 (ADRH0)	Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005A	ADC Data Register Low 0 (ADRL0)	Read:	AD1	AD0	0	0	0	0	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0

Figure 15-7. ADRH0 and ADRL0 in Left Justified Mode

In left justified sign mode the ADRH0 holds the eight MSBs with the MSB complemented, and the ADRL0 holds the two least significant bits (LSBs), of the 10-bit result. The ADRH0 and ADRL0 are updated each time a single channel ADC conversion completes. Reading ADRH0 latches the contents of ADRL0. Until ADRL0 is read all subsequent ADC results will be lost. (See **Figure 15-8 . ADRH0 and ADRL0 in Left Justified Sign Data Mode**.)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0059	ADC Data Register High 0 (ADRH0)	Read:	$\overline{AD9}$	AD8	AD7	AD6	AD5	AD4	AD3	AD2
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005A	ADC Data Register Low 0 (ADRL0)	Read:	AD1	AD0	0	0	0	0	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0

Figure 15-8. ADRH0 and ADRL0 in Left Justified Sign Data Mode

15.8.4 ADC Auto-Scan Mode Data Registers (ADRL1–ADRL3)

The ADC data registers 1 to 3 (ADRL1–ADRL3), are 8-bit registers for conversion results in 8-bit truncated mode, for channels ATD1 to ATD3, when the ADC is operating in auto-scan mode (MODE[1:0] = 00).

Address: ADRL1, $005B; ADRL2, $005C; and ADRL3, $005D

Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 15-9. ADC Data Register Low 1 to 3 (ADRL1–ADRL3)

15.8.5 ADC Auto-Scan Control Register (ADASCR)

The ADC auto-scan control register (ADASCR) enables and controls the ADC auto-scan function.

Address: $005E

Read:	0	0	0	0	0	AUTO1	AUTO0	ASCAN
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved

Figure 15-10. ADC Scan Control Register (ADASCR)

AUTO[1:0] — Auto-scan Mode Channel Select Bits

AUTO1 and AUTO0 form a 2-bit field which is used to define the number of auto-scan channels used when in auto-scan mode. Reset clears these bits.

Table 15-4. Auto-scan Mode Channel Select

AUTO1	AUTO0	Auto-Scan Channels
0	0	ATD0 only
0	1	ATD0 to ATD1
1	0	ATD0 to ATD2
1	1	ATD0 to ATD3

ASCAN — Auto-scan Mode Enable Bit

This bit enable/disable the Auto-scan mode. Reset clears this bit.

1 = Auto-scan mode is enabled
0 = Auto-scan mode is disabled

Auto-scan mode should not be enabled when ADC continuous conversion is enabled; i.e. when ADCO=1.

Section 16. Serial Communications Interface (SCI)

16.1 Contents

16.2 Introduction 252
16.3 Features 252
16.4 Pin Name Conventions 254
16.5 Functional Description 254
16.5.1 Data Format 257
16.5.2 Transmitter 257
16.5.2.1 Character Length 259
16.5.2.2 Character Transmission 259
16.5.2.3 Break Characters 260
16.5.2.4 Idle Characters 260
16.5.2.5 Inversion of Transmitted Output 261
16.5.2.6 Transmitter Interrupts 261
16.5.3 Receiver 262
16.5.3.1 Character Length 262
16.5.3.2 Character Reception 262
16.5.3.3 Data Sampling 264
16.5.3.4 Framing Errors 266
16.5.3.5 Baud Rate Tolerance 266
16.5.3.6 Receiver Wakeup 269
16.5.3.7 Receiver Interrupts 270
16.5.3.8 Error Interrupts 270
16.6 Low-Power Modes 271
16.6.1 Wait Mode 271
16.6.2 Stop Mode 271
16.7 SCI During Break Module Interrupts 272
16.8 I/O Signals 272
16.8.1 TxD (Transmit Data) 272

16.8.2 RxD (Receive Data) 273

16.9 I/O Registers 273
16.9.1 SCI Control Register 1 273
16.9.2 SCI Control Register 2 276
16.9.3 SCI Control Register 3 279
16.9.4 SCI Status Register 1 282
16.9.5 SCI Status Register 2 286
16.9.6 SCI Data Register 287
16.9.7 SCI Baud Rate Register 288

16.2 Introduction

This section describes the serial communications interface (SCI) module, which allows high-speed asynchronous communications with peripheral devices and other MCUs.

NOTE: *When the SCI is enabled, the TxD pin is an open-drain output and requires a pullup resistor to be connected for proper SCI operation.*

NOTE: *References to DMA (direct-memory access) and associated functions are only valid if the MCU has a DMA module. This MCU does not have the DMA function. Any DMA-related register bits should be left in their reset state for normal MCU operation.*

16.3 Features

Features of the SCI module include the following:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- 32 programmable baud rates
- Programmable 8-bit or 9-bit character length
- Separately enabled transmitter and receiver
- Separate receiver and transmitter CPU interrupt requests
- Programmable transmitter output polarity

- Two receiver wakeup methods:
  - Idle line wakeup
  - Address mark wakeup
- Interrupt-driven operation with eight interrupt 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
- 1/16 bit-time noise detection
- Configuration register bit, SCIBDSRC, to allow selection of baud rate clock source

16.4 Pin Name Conventions

The generic names of the SCI I/O pins are:

- RxD (receive data)
- TxD (transmit data)

SCI I/O (input/output) lines are implemented by sharing parallel I/O port pins. The full name of an SCI input or output reflects the name of the shared port pin. **Table 16-1** shows the full names and the generic names of the SCI I/O pins. The generic pin names appear in the text of this section.

Table 16-1. Pin Name Conventions

Generic Pin Names:	RxD	TxD
Full Pin Names:	PTB3/SCL1/RxD	PTB2/SDA1/TxD

NOTE: *When the SCI is enabled, the TxD pin is an open-drain output and requires a pullup resistor to be connected for proper SCI operation.*

16.5 Functional Description

Figure 16-1 shows the structure of the SCI module. The SCI allows full-duplex, asynchronous, NRZ serial communication among the MCU and remote devices, including other MCUs. The transmitter and receiver of the SCI operate independently, although they use the same baud rate generator. During normal operation, the CPU monitors the status of the SCI, writes the data to be transmitted, and processes received data.

The baud rate clock source for the SCI can be selected via the configuration bit, SCIBDSRC, of the CONFIG2 register ($001D). Source selection values are shown in **Figure 16-1**.

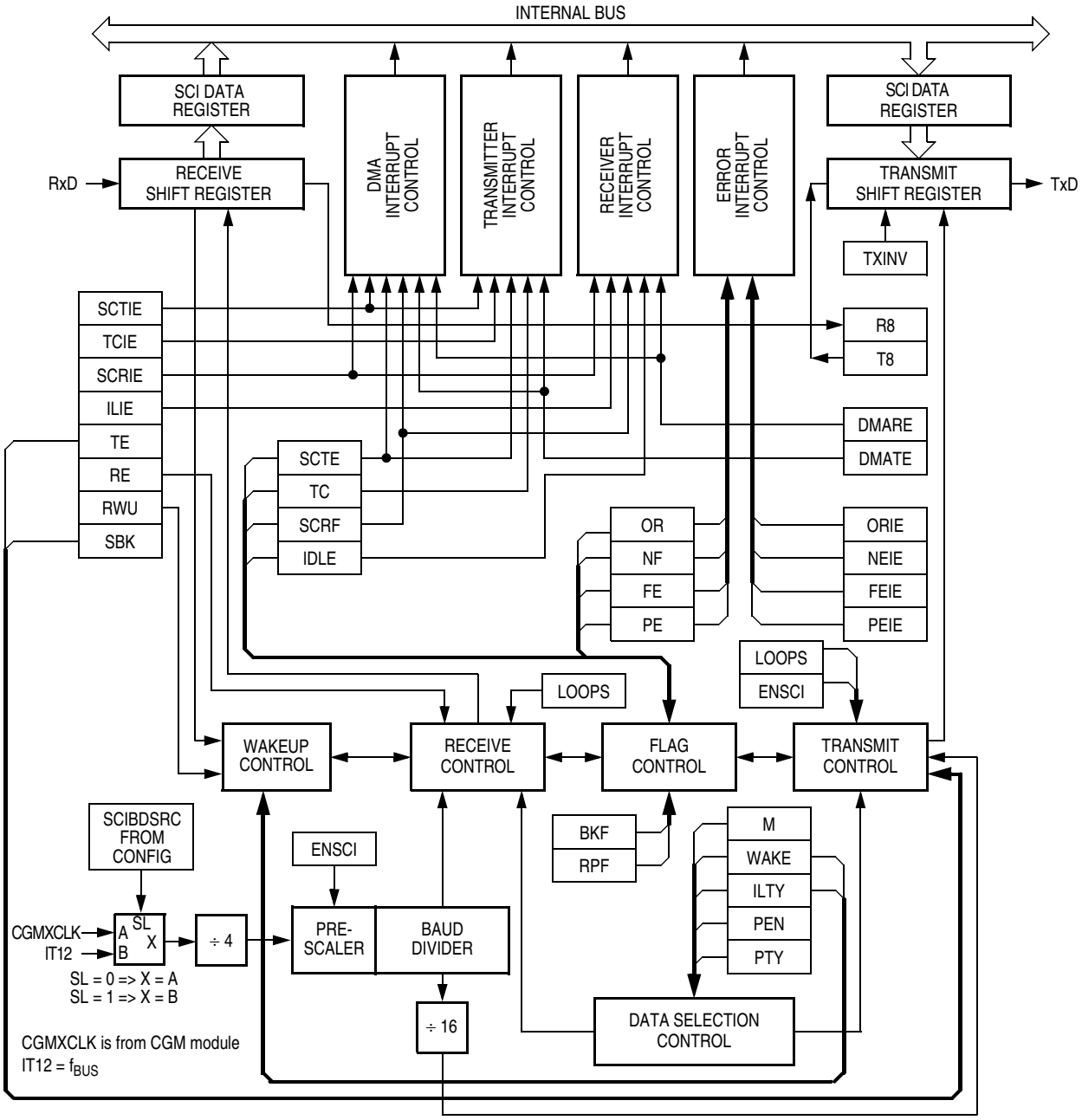


Figure 16-1. SCI Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0013	SCI Control Register 1 (SCC1)	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0014	SCI Control Register 2 (SCC2)	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0015	SCI Control Register 3 (SCC3)	Read:	R8	T8	DMARE	DMATE	ORIE	NEIE	FEIE	PEIE
		Write:								
		Reset:	U	U	0	0	0	0	0	0
$0016	SCI Status Register 1 (SCS1)	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:								
		Reset:	1	1	0	0	0	0	0	0
$0017	SCI Status Register 2 (SCS2)	Read:							BKF	RPF
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0018	SCI Data Register (SCDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							
$0019	SCI Baud Rate Register (SCBR)	Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved U = Unaffected

Figure 16-2. SCI I/O Register Summary

16.5.1 Data Format

The SCI uses the standard non-return-to-zero mark/space data format illustrated in **Figure 16-3**.

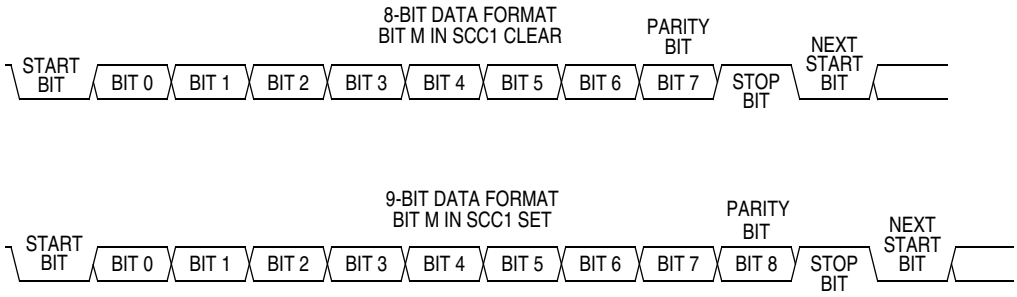


Figure 16-3. SCI Data Formats

16.5.2 Transmitter

Figure 16-4 shows the structure of the SCI transmitter.

The baud rate clock source for the SCI can be selected via the configuration bit, SCIBDSRC. Source selection values are shown in **Figure 16-4**.

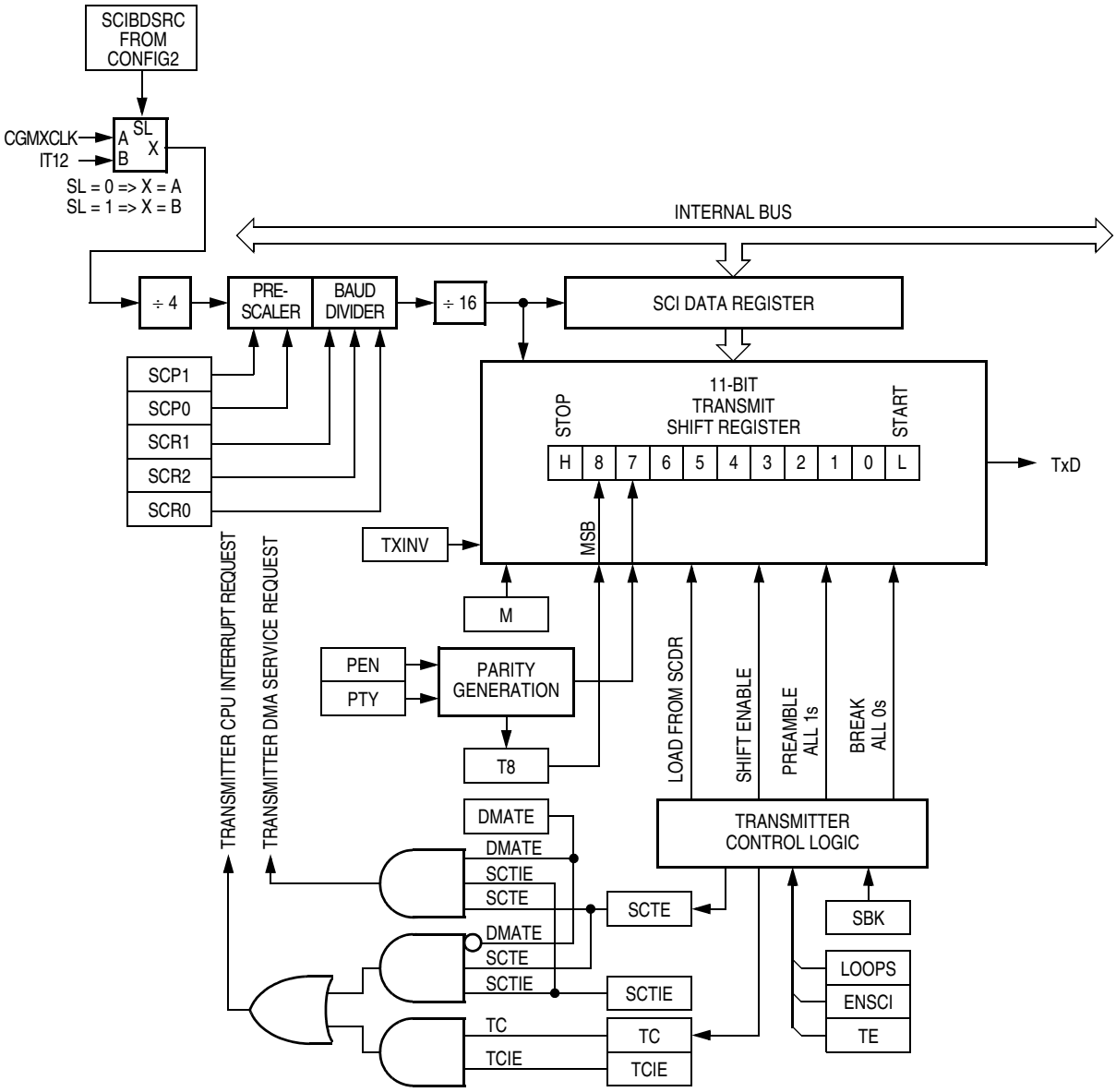


Figure 16-4. SCI Transmitter

16.5.2.1 Character Length

The transmitter can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCC1) determines character length. When transmitting 9-bit data, bit T8 in SCI control register 3 (SCC3) is the ninth bit (bit 8).

16.5.2.2 Character Transmission

During an SCI transmission, the transmit shift register shifts a character out to the TxD pin. The SCI data register (SCDR) is the write-only buffer between the internal data bus and the transmit shift register. To initiate an SCI transmission:

1. Enable the SCI by writing a logic 1 to the enable SCI bit (ENSCI) in SCI control register 1 (SCC1).
2. Enable the transmitter by writing a logic 1 to the transmitter enable bit (TE) in SCI control register 2 (SCC2).
3. Clear the SCI transmitter empty bit by first reading SCI status register 1 (SCS1) and then writing to the SCDR.
4. Repeat step 3 for each subsequent transmission.

At the start of a transmission, transmitter control logic automatically loads the transmit shift register with a preamble of logic 1s. After the preamble shifts out, control logic transfers the SCDR data 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.

The SCI transmitter empty bit, SCTE, in SCS1 becomes set when the SCDR transfers a byte to the transmit shift register. The SCTE bit indicates that the SCDR can accept new data from the internal data bus. If the SCI transmit interrupt enable bit, SCTIE, in SCC2 is also set, the SCTE bit generates a transmitter CPU interrupt request.

When the transmit shift register is not transmitting a character, the TxD pin goes to the idle condition, logic 1. If at any time software clears the ENSCI bit in SCI control register 1 (SCC1), the transmitter and receiver relinquish control of the port pin.

16.5.2.3 Break Characters

Writing a logic 1 to the send break bit, SBK, in SCC2 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 SCC1. 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 character.

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 bit (FE) in SCS1
- Sets the SCI receiver full bit (SCRF) in SCS1
- Clears the SCI data register (SCDR)
- Clears the R8 bit in SCC3
- Sets the break flag bit (BKF) in SCS2
- May set the overrun (OR), noise flag (NF), parity error (PE), or reception in progress flag (RPF) bits

16.5.2.4 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 SCC1. The preamble is a synchronizing idle character that begins every transmission.

If the TE bit is cleared during a transmission, the TxD pin 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 character currently being transmitted.

NOTE: *When queueing an idle character, return the TE bit to logic 1 before the stop bit of the current character shifts out to the TxD pin. Setting TE after the stop bit appears on TxD causes data previously written to the SCDR to be lost.*

Toggle the TE bit for a queued idle character when the SCTE bit becomes set and just before writing the next byte to the SCDR.

16.5.2.5 Inversion of Transmitted Output

The transmit inversion bit (TXINV) in SCI control register 1 (SCC1) reverses the polarity of transmitted data. All transmitted values, including idle, break, start, and stop bits, are inverted when TXINV is at logic 1. (See **16.9.1 SCI Control Register 1**.)

16.5.2.6 Transmitter Interrupts

These conditions can generate CPU interrupt requests from the SCI transmitter:

- SCI transmitter empty (SCTE) — The SCTE bit in SCS1 indicates that the SCDR has transferred a character to the transmit shift register. SCTE can generate a transmitter CPU interrupt request. Setting the SCI transmit interrupt enable bit, SCTIE, in SCC2 enables the SCTE bit to generate transmitter CPU interrupt requests.
- Transmission complete (TC) — The TC bit in SCS1 indicates that the transmit shift register and the SCDR are empty and that no break or idle character has been generated. The transmission complete interrupt enable bit, TCIE, in SCC2 enables the TC bit to generate transmitter CPU interrupt requests.

16.5.3 Receiver

Figure 16-5 shows the structure of the SCI receiver.

16.5.3.1 Character Length

The receiver can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCC1) determines character length. When receiving 9-bit data, bit R8 in SCI control register 2 (SCC2) is the ninth bit (bit 8). When receiving 8-bit data, bit R8 is a copy of the eighth bit (bit 7).

16.5.3.2 Character Reception

During an SCI reception, the receive shift register shifts characters in from the RxD pin. The SCI data register (SCDR) is the read-only buffer between the internal data bus and the receive shift register.

After a complete character shifts into the receive shift register, the data portion of the character transfers to the SCDR. The SCI receiver full bit, SCRF, in SCI status register 1 (SCS1) becomes set, indicating that the received byte can be read. If the SCI receive interrupt enable bit, SCRIE, in SCC2 is also set, the SCRF bit generates a receiver CPU interrupt request.

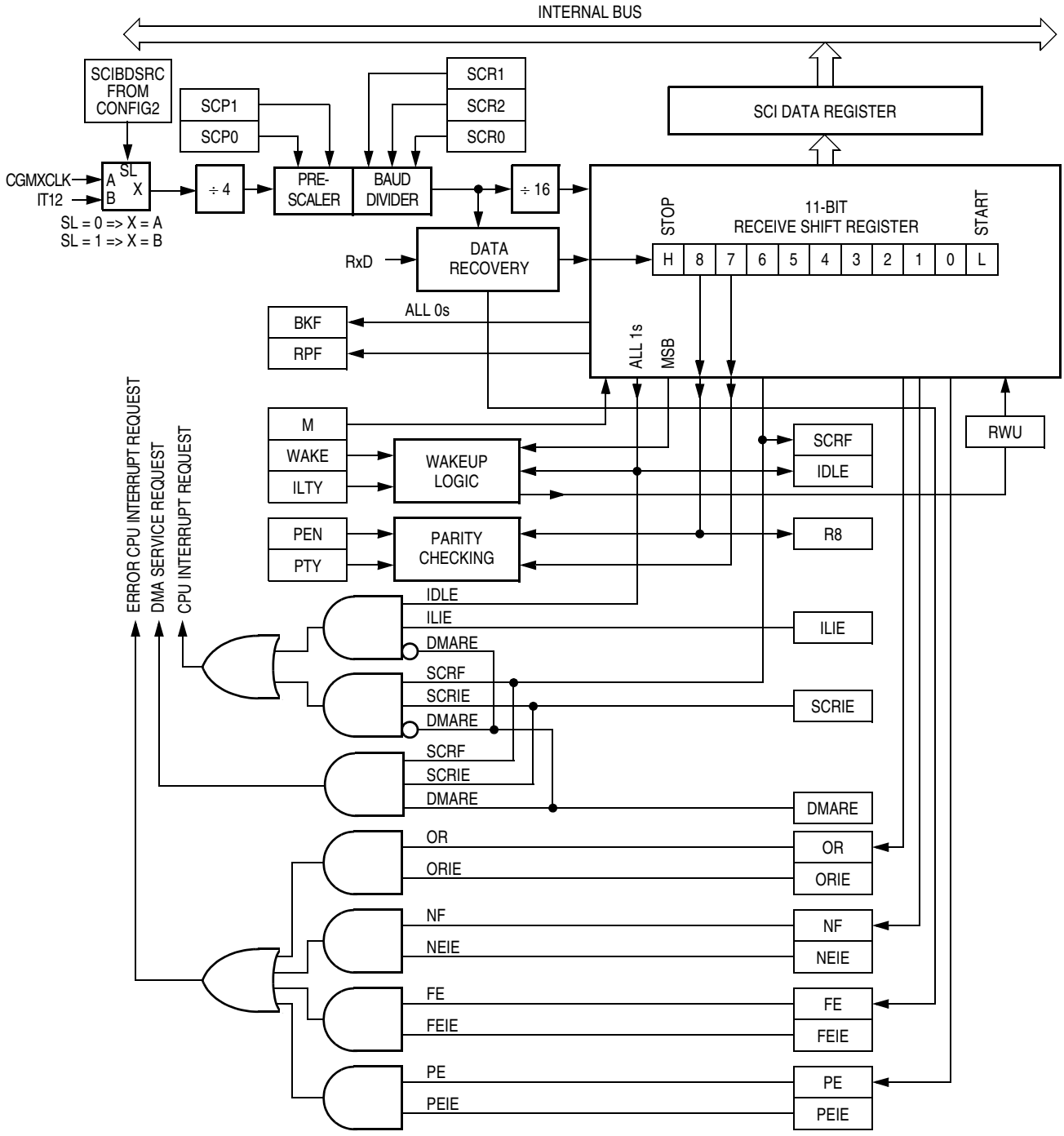


Figure 16-5. SCI Receiver Block Diagram

16.5.3.3 Data Sampling

The receiver samples the RxD pin 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 is resynchronized at the following times (see **Figure 16-6**):

- 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.

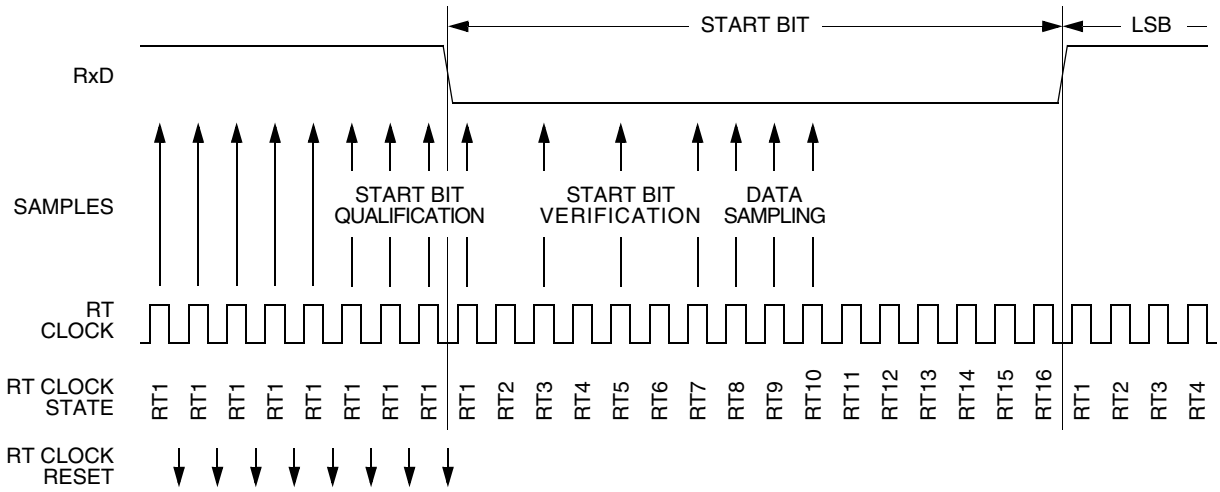


Figure 16-6. Receiver Data Sampling

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7. Table 16-2 summarizes the results of the start bit verification samples.

Table 16-2. 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

Start bit verification is not successful if any two of the three verification samples are logic 1s. 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 16-3 summarizes the results of the data bit samples.

Table 16-3. 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.*

To verify a stop bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. **Table 16-4** summarizes the results of the stop bit samples.

Table 16-4. 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

16.5.3.4 Framing Errors

If the data recovery logic does not detect a logic 1 where the stop bit should be in an incoming character, it sets the framing error bit, FE, in SCS1. A break character also sets the FE bit because a break character has no stop bit. The FE bit is set at the same time that the SCRF bit is set.

16.5.3.5 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 to fall outside the actual stop bit. Then a noise error occurs. If more than one of the samples is outside the stop bit, a framing error occurs. In most applications, the baud rate

tolerance is much more than the degree of misalignment that is likely to occur.

As the receiver samples an incoming character, it resynchronizes the RT clock on any valid falling edge within the character. Resynchronization within characters corrects misalignments between transmitter bit times and receiver bit times.

Slow Data Tolerance

Figure 16-7 shows how much a slow received character 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.

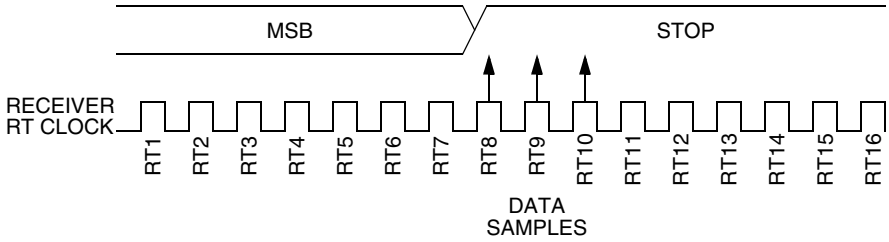


Figure 16-7. Slow Data

For an 8-bit character, data sampling of the stop bit takes the receiver 9 bit times × 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in Figure 16-7, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 9 bit times × 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 character with no errors is

$$\left|\frac{154 - 147}{154}\right| \times 100 = 4.54\%$$

For a 9-bit character, data sampling of the stop bit takes the receiver 10 bit times × 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in **Figure 16-7**, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 10 bit times × 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

$$\left|\frac{170 - 163}{170}\right| \times 100 = 4.12\%$$

Fast Data Tolerance

Figure 16-8 shows how much a fast received character can be misaligned without causing a noise error or a framing error. The fast stop bit ends at RT10 instead of RT16 but is still there for the stop bit data samples at RT8, RT9, and RT10.

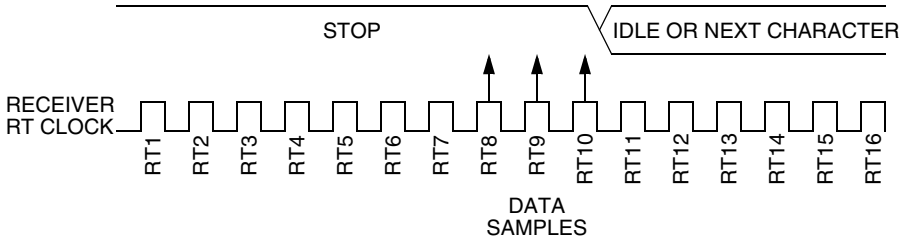


Figure 16-8. Fast Data

For an 8-bit character, data sampling of the stop bit takes the receiver 9 bit times × 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in **Figure 16-8**, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 10 bit times × 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

$$\left|\frac{154 - 160}{154}\right| \times 100 = 3.90\%$$

For a 9-bit character, data sampling of the stop bit takes the receiver 10 bit times × 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in **Figure 16-8**, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 11 bit times × 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

$$\left|\frac{170 - 176}{170}\right| \times 100 = 3.53\%$$

16.5.3.6 Receiver Wakeup

So that the MCU can 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 SCC2 puts the receiver into a standby state during which receiver interrupts are disabled.

Depending on the state of the WAKE bit in SCC1, either of two conditions on the RxD pin can bring the receiver out of the standby state:

- Address mark — An address mark is a logic 1 in the most significant bit position of a received character. When the WAKE bit is set, an address mark wakes the receiver from the standby state by clearing the RWU bit. The address mark also sets the SCI receiver full bit, SCRF. Software can then compare the character containing the address mark to the user-defined address of the receiver. If they are the same, the receiver remains awake and processes the characters that follow. If they are not the same, software can set the RWU bit and put the receiver back into the standby state.
- Idle input line condition — When the WAKE bit is clear, an idle character on the RxD pin wakes the receiver from the standby state by clearing the RWU bit. The idle character that wakes the receiver does not set the receiver idle bit, IDLE, or the SCI receiver

full bit, SCRF. The idle line type bit, ILTY, determines whether the receiver begins counting logic 1s as idle character bits after the start bit or after the stop bit.

NOTE: *With the WAKE bit clear, setting the RWU bit after the RxD pin has been idle may cause the receiver to wake up immediately.*

16.5.3.7 Receiver Interrupts

The following sources can generate CPU interrupt requests from the SCI receiver:

- SCI receiver full (SCRF) — The SCRF bit in SCS1 indicates that the receive shift register has transferred a character to the SCDR. SCRF can generate a receiver CPU interrupt request. Setting the SCI receive interrupt enable bit, SCRIE, in SCC2 enables the SCRF bit to generate receiver CPU interrupts.
- Idle input (IDLE) — The IDLE bit in SCS1 indicates that 10 or 11 consecutive logic 1s shifted in from the RxD pin. The idle line interrupt enable bit, ILIE, in SCC2 enables the IDLE bit to generate CPU interrupt requests.

16.5.3.8 Error Interrupts

The following receiver error flags in SCS1 can generate CPU interrupt requests:

- Receiver overrun (OR) — The OR bit indicates that the receive shift register shifted in a new character before the previous character was read from the SCDR. The previous character remains in the SCDR, and the new character is lost. The overrun interrupt enable bit, ORIE, in SCC3 enables OR to generate SCI error CPU interrupt requests.
- Noise flag (NF) — The NF bit is set when the SCI detects noise on incoming data or break characters, including start, data, and stop bits. The noise error interrupt enable bit, NEIE, in SCC3 enables NF to generate SCI error CPU interrupt requests.

- Framing error (FE) — The FE bit in SCS1 is set when a logic 0 occurs where the receiver expects a stop bit. The framing error interrupt enable bit, FEIE, in SCC3 enables FE to generate SCI error CPU interrupt requests.
- Parity error (PE) — The PE bit in SCS1 is set when the SCI detects a parity error in incoming data. The parity error interrupt enable bit, PEIE, in SCC3 enables PE to generate SCI error CPU interrupt requests.

16.6 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

16.6.1 Wait Mode

The SCI module remains active after the execution of a WAIT instruction. In wait mode, the SCI module registers are not accessible by the CPU. Any enabled CPU interrupt request from the SCI module can bring the MCU out of wait mode.

If SCI module functions are not required during wait mode, reduce power consumption by disabling the module before executing the WAIT instruction.

Refer to **9.7 Low-Power Modes** for information on exiting wait mode.

16.6.2 Stop Mode

The SCI module is inactive after the execution of a STOP instruction. The STOP instruction does not affect SCI register states. SCI module operation resumes after an external interrupt.

Because the internal clock is inactive during stop mode, entering stop mode during an SCI transmission or reception results in invalid data.

Refer to **9.7 Low-Power Modes** for information on exiting stop mode.

16.7 SCI During Break Module Interrupts

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state.

To allow software to clear status bits during a break interrupt, write a logic 1 to the BCFE bit. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect status bits during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a 2-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is at logic 0. After the break, doing the second step clears the status bit.

16.8 I/O Signals

Port B shares two of its pins with the SCI module.
The two SCI I/O pins are:

- PTB2/SDA1/TxD — Transmit data
- PTB3/SCL1/RxD — Receive data

16.8.1 TxD (Transmit Data)

When the SCI is enabled (ENSCI=1), the PTB2/SDA1/TxD pin becomes the serial data output, TxD, from the SCI transmitter regardless of the state of the DDRB2 bit in data direction register B (DDRB). The TxD pin is an open-drain output and requires a pullup resistor to be connected for proper SCI operation.

NOTE: *The PTB2/SDA1/TxD pin is an open-drain pin when configured as an output. Therefore, when configured as a general purpose output pin (PTB2), a pullup resistor must be connected to this pin.*

16.8.2 RxD (Receive Data)

When the SCI is enabled (ENSCI=1), the PTB3/SCL1/RxD pin becomes the serial data input, RxD, to the SCI receiver regardless of the state of the DDRB3 bit in data direction register B (DDRB).

NOTE: *The PTB3/SCL1/RxD pin is an open-drain pin when configured as an output. Therefore, when configured as a general purpose output pin (PTB3), a pullup resistor must be connected to this pin.*

16.9 I/O Registers

These I/O registers control and monitor SCI operation:

- SCI control register 1 (SCC1)
- SCI control register 2 (SCC2)
- SCI control register 3 (SCC3)
- SCI status register 1 (SCS1)
- SCI status register 2 (SCS2)
- SCI data register (SCDR)
- SCI baud rate register (SCBR)

16.9.1 SCI Control Register 1

SCI control register 1:

- Enables loop mode operation
- Enables the SCI
- Controls output polarity
- Controls character length
- Controls SCI wakeup method
- Controls idle character detection
- Enables parity function
- Controls parity type

Address: $0013

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
Reset:	0	0	0	0	0	0	0	0

Figure 16-9. SCI Control Register 1 (SCC1)

LOOPS — Loop Mode Select Bit

This read/write bit enables loop mode operation. In loop mode the RxD pin is disconnected from the SCI, and the transmitter output goes into the receiver input. Both the transmitter and the receiver must be enabled to use loop mode. Reset clears the LOOPS bit.

1 = Loop mode enabled
0 = Normal operation enabled

ENSCI — Enable SCI Bit

This read/write bit enables the SCI and the SCI baud rate generator. Clearing ENSCI sets the SCTE and TC bits in SCI status register 1 and disables transmitter interrupts. Reset clears the ENSCI bit.

1 = SCI enabled
0 = SCI disabled

TXINV — Transmit Inversion Bit

This read/write bit reverses the polarity of transmitted data. Reset clears the TXINV bit.

1 = Transmitter output inverted
0 = Transmitter output not inverted

NOTE: *Setting the TXINV bit inverts all transmitted values, including idle, break, start, and stop bits.*

M — Mode (Character Length) Bit

This read/write bit determines whether SCI characters are eight or nine bits long. (See Table 16-5.) The ninth bit can serve as an extra stop bit, as a receiver wakeup signal, or as a parity bit. Reset clears the M bit.

1 = 9-bit SCI characters
0 = 8-bit SCI characters

WAKE — Wakeup Condition Bit

This read/write bit determines which condition wakes up the SCI: a logic 1 (address mark) in the most significant bit position of a received character or an idle condition on the RxD pin. Reset clears the WAKE bit.

1 = Address mark wakeup
0 = Idle line wakeup

ILTY — Idle Line Type Bit

This read/write bit determines when the SCI 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. Reset clears the ILTY bit.

1 = Idle character bit count begins after stop bit
0 = Idle character bit count begins after start bit

PEN — Parity Enable Bit

This read/write bit enables the SCI parity function. (See Table 16-5.) When enabled, the parity function inserts a parity bit in the most significant bit position. (See Figure 16-3.) Reset clears the PEN bit.

1 = Parity function enabled
0 = Parity function disabled

PTY — Parity Bit

This read/write bit determines whether the SCI generates and checks for odd parity or even parity. (See Table 16-5.) Reset clears the PTY bit.

1 = Odd parity
0 = Even parity

NOTE: *Changing the PTY bit in the middle of a transmission or reception can generate a parity error.*

Table 16-5. Character Format Selection

Control Bits		Character Format				
M	**PEN and PTY**	**Start Bits**	**Data Bits**	**Parity**	**Stop Bits**	**Character Length**
0	0X	1	8	None	1	10 bits
1	0X	1	9	None	1	11 bits
0	10	1	7	Even	1	10 bits
0	11	1	7	Odd	1	10 bits
1	10	1	8	Even	1	11 bits
1	11	1	8	Odd	1	11 bits

16.9.2 SCI Control Register 2

SCI control register 2:

- Enables the following CPU interrupt requests:
  - Enables the SCTE bit to generate transmitter CPU interrupt requests
  - Enables the TC bit to generate transmitter CPU interrupt requests
  - Enables the SCRF bit to generate receiver CPU interrupt requests
  - Enables the IDLE bit to generate receiver CPU interrupt requests

- Enables the transmitter
- Enables the receiver
- Enables SCI wakeup
- Transmits SCI break characters

Address: $0014

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0

Figure 16-10. SCI Control Register 2 (SCC2)

SCTIE — SCI Transmit Interrupt Enable Bit

This read/write bit enables the SCTE bit to generate SCI transmitter CPU interrupt requests. Reset clears the SCTIE bit.

1 = SCTE enabled to generate CPU interrupt
0 = SCTE not enabled to generate CPU interrupt

TCIE — Transmission Complete Interrupt Enable Bit

This read/write bit enables the TC bit to generate SCI transmitter CPU interrupt requests. Reset clears the TCIE bit.

1 = TC enabled to generate CPU interrupt requests
0 = TC not enabled to generate CPU interrupt requests

SCRIE — SCI Receive Interrupt Enable Bit

This read/write bit enables the SCRF bit to generate SCI receiver CPU interrupt requests. Reset clears the SCRIE bit.

1 = SCRF enabled to generate CPU interrupt
0 = SCRF not enabled to generate CPU interrupt

ILIE — Idle Line Interrupt Enable Bit

This read/write bit enables the IDLE bit to generate SCI receiver CPU interrupt requests. Reset clears the ILIE bit.

1 = IDLE enabled to generate CPU interrupt requests
0 = IDLE not enabled to generate CPU interrupt requests

TE — Transmitter Enable Bit

Setting this read/write bit begins the transmission by sending a preamble of 10 or 11 logic 1s from the transmit shift register to the TxD pin. If software clears the TE bit, the transmitter completes any transmission in progress before the TxD returns to the idle condition (logic 1). Clearing and then setting TE during a transmission queues an idle character to be sent after the character currently being transmitted. Reset clears the TE bit.

1 = Transmitter enabled
0 = Transmitter disabled

NOTE: *Writing to the TE bit is not allowed when the enable SCI bit (ENSCI) is clear. ENSCI is in SCI control register 1.*

RE — Receiver Enable Bit

Setting this read/write bit enables the receiver. Clearing the RE bit disables the receiver but does not affect receiver interrupt flag bits. Reset clears the RE bit.

1 = Receiver enabled
0 = Receiver disabled

NOTE: *Writing to the RE bit is not allowed when the enable SCI bit (ENSCI) is clear. ENSCI is in SCI control register 1.*

RWU — Receiver Wakeup Bit

This read/write bit puts the receiver in a standby state during which receiver interrupts are disabled. The WAKE bit in SCC1 determines whether an idle input or an address mark brings the receiver out of the standby state and clears the RWU bit. Reset clears the RWU bit.

1 = Standby state
0 = Normal operation

SBK — Send Break Bit

Setting and then clearing this read/write bit transmits a break character followed by a logic 1. The logic 1 after the break character guarantees recognition of a valid start bit. If SBK remains set, the transmitter continuously transmits break characters with no logic 1s between them. Reset clears the SBK bit.

1 = Transmit break characters
0 = No break characters being transmitted

NOTE: *Do not toggle the SBK bit immediately after setting the SCTE bit. Toggling SBK before the preamble begins causes the SCI to send a break character instead of a preamble.*

16.9.3 SCI Control Register 3

SCI control register 3:

- Stores the ninth SCI data bit received and the ninth SCI data bit to be transmitted
- Enables these interrupts:
  - Receiver overrun interrupts
  - Noise error interrupts
  - Framing error interrupts
- Parity error interrupts

Address: $0015

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R8	T8	DMARE	DMATE	ORIE	NEIE	FEIE	PEIE
Write:								
Reset:	U	U	0	0	0	0	0	0

= Unimplemented U = Unaffected

Figure 16-11. SCI Control Register 3 (SCC3)

R8 — Received Bit 8

When the SCI is receiving 9-bit characters, R8 is the read-only ninth bit (bit 8) of the received character. R8 is received at the same time that the SCDR receives the other 8 bits.
When the SCI is receiving 8-bit characters, R8 is a copy of the eighth bit (bit 7). Reset has no effect on the R8 bit.

T8 — Transmitted Bit 8

When the SCI is transmitting 9-bit characters, T8 is the read/write ninth bit (bit 8) of the transmitted character. T8 is loaded into the transmit shift register at the same time that the SCDR is loaded into the transmit shift register. Reset has no effect on the T8 bit.

DMARE — DMA Receive Enable Bit

CAUTION: *The DMA module is not included on this MCU. Writing a logic 1 to DMARE or DMATE may adversely affect MCU performance.*

1 = DMA not enabled to service SCI receiver DMA service requests generated by the SCRF bit (SCI receiver CPU interrupt requests enabled)
0 = DMA not enabled to service SCI receiver DMA service requests generated by the SCRF bit (SCI receiver CPU interrupt requests enabled)

DMATE — DMA Transfer Enable Bit

CAUTION: *The DMA module is not included on this MCU. Writing a logic 1 to DMARE or DMATE may adversely affect MCU performance.*

1 = SCTE DMA service requests enabled; SCTE CPU interrupt requests disabled
0 = SCTE DMA service requests disabled; SCTE CPU interrupt requests enabled

ORIE — Receiver Overrun Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the receiver overrun bit, OR.

1 = SCI error CPU interrupt requests from OR bit enabled
0 = SCI error CPU interrupt requests from OR bit disabled

NEIE — Receiver Noise Error Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the noise error bit, NE. Reset clears NEIE.

1 = SCI error CPU interrupt requests from NE bit enabled
0 = SCI error CPU interrupt requests from NE bit disabled

FEIE — Receiver Framing Error Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the framing error bit, FE. Reset clears FEIE.

1 = SCI error CPU interrupt requests from FE bit enabled
0 = SCI error CPU interrupt requests from FE bit disabled

PEIE — Receiver Parity Error Interrupt Enable Bit

This read/write bit enables SCI receiver CPU interrupt requests generated by the parity error bit, PE. (See **16.9.4 SCI Status Register 1**.) Reset clears PEIE.

1 = SCI error CPU interrupt requests from PE bit enabled
0 = SCI error CPU interrupt requests from PE bit disabled

16.9.4 SCI Status Register 1

SCI status register 1 (SCS1) contains flags to signal these conditions:

- Transfer of SCDR data to transmit shift register complete
- Transmission complete
- Transfer of receive shift register data to SCDR complete
- Receiver input idle
- Receiver overrun
- Noisy data
- Framing error
- Parity error

Address: $0016

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
Write:								
Reset:	1	1	0	0	0	0	0	0

= Unimplemented

Figure 16-12. SCI Status Register 1 (SCS1)

SCTE — SCI Transmitter Empty Bit

This clearable, read-only bit is set when the SCDR transfers a character to the transmit shift register. SCTE can generate an SCI transmitter CPU interrupt request. When the SCTIE bit in SCC2 is set, SCTE generates an SCI transmitter CPU interrupt request. In normal operation, clear the SCTE bit by reading SCS1 with SCTE set and then writing to SCDR. Reset sets the SCTE bit.

1 = SCDR data transferred to transmit shift register
0 = SCDR data not transferred to transmit shift register

TC — Transmission Complete Bit

This read-only bit is set when the SCTE bit is set, and no data, preamble, or break character is being transmitted. TC generates an SCI transmitter CPU interrupt request if the TCIE bit in SCC2 is also set. TC is automatically cleared when data, preamble or break is queued and ready to be sent. There may be up to 1.5 transmitter clocks of latency between queueing data, preamble, and break and the transmission actually starting. Reset sets the TC bit.

1 = No transmission in progress
0 = Transmission in progress

SCRF — SCI Receiver Full Bit

This clearable, read-only bit is set when the data in the receive shift register transfers to the SCI data register. SCRF can generate an SCI receiver CPU interrupt request. When the SCRIE bit in SCC2 is set, SCRF generates a CPU interrupt request. In normal operation, clear the SCRF bit by reading SCS1 with SCRF set and then reading the SCDR. Reset clears SCRF.

1 = Received data available in SCDR
0 = Data not available in SCDR

IDLE — Receiver Idle Bit

This clearable, read-only bit is set when 10 or 11 consecutive logic 1s appear on the receiver input. IDLE generates an SCI error CPU interrupt request if the ILIE bit in SCC2 is also set. Clear the IDLE bit by reading SCS1 with IDLE set and then reading the SCDR. After the receiver is enabled, it must receive a valid character that sets the SCRF bit before an idle condition can set the IDLE bit. Also, after the IDLE bit has been cleared, a valid character must again set the SCRF bit before an idle condition can set the IDLE bit. Reset clears the IDLE bit.

1 = Receiver input idle
0 = Receiver input active (or idle since the IDLE bit was cleared)

OR — Receiver Overrun Bit

This clearable, read-only bit is set when software fails to read the SCDR before the receive shift register receives the next character. The OR bit generates an SCI error CPU interrupt request if the ORIE

bit in SCC3 is also set. The data in the shift register is lost, but the data already in the SCDR is not affected. Clear the OR bit by reading SCS1 with OR set and then reading the SCDR. Reset clears the OR bit.

1 = Receive shift register full and SCRF = 1
0 = No receiver overrun

Software latency may allow an overrun to occur between reads of SCS1 and SCDR in the flag-clearing sequence. **Figure 16-13** shows the normal flag-clearing sequence and an example of an overrun caused by a delayed flag-clearing sequence. The delayed read of SCDR does not clear the OR bit because OR was not set when SCS1 was read. Byte 2 caused the overrun and is lost. The next flag-clearing sequence reads byte 3 in the SCDR instead of byte 2.

In applications that are subject to software latency or in which it is important to know which byte is lost due to an overrun, the flag-clearing routine can check the OR bit in a second read of SCS1 after reading the data register.

NF — Receiver Noise Flag Bit

This clearable, read-only bit is set when the SCI detects noise on the RxD pin. NF generates an NF CPU interrupt request if the NEIE bit in SCC3 is also set. Clear the NF bit by reading SCS1 and then reading the SCDR. Reset clears the NF bit.

1 = Noise detected
0 = No noise detected

FE — Receiver Framing Error Bit

This clearable, read-only bit is set when a logic 0 is accepted as the stop bit. FE generates an SCI error CPU interrupt request if the FEIE bit in SCC3 also is set. Clear the FE bit by reading SCS1 with FE set and then reading the SCDR. Reset clears the FE bit.

1 = Framing error detected
0 = No framing error detected

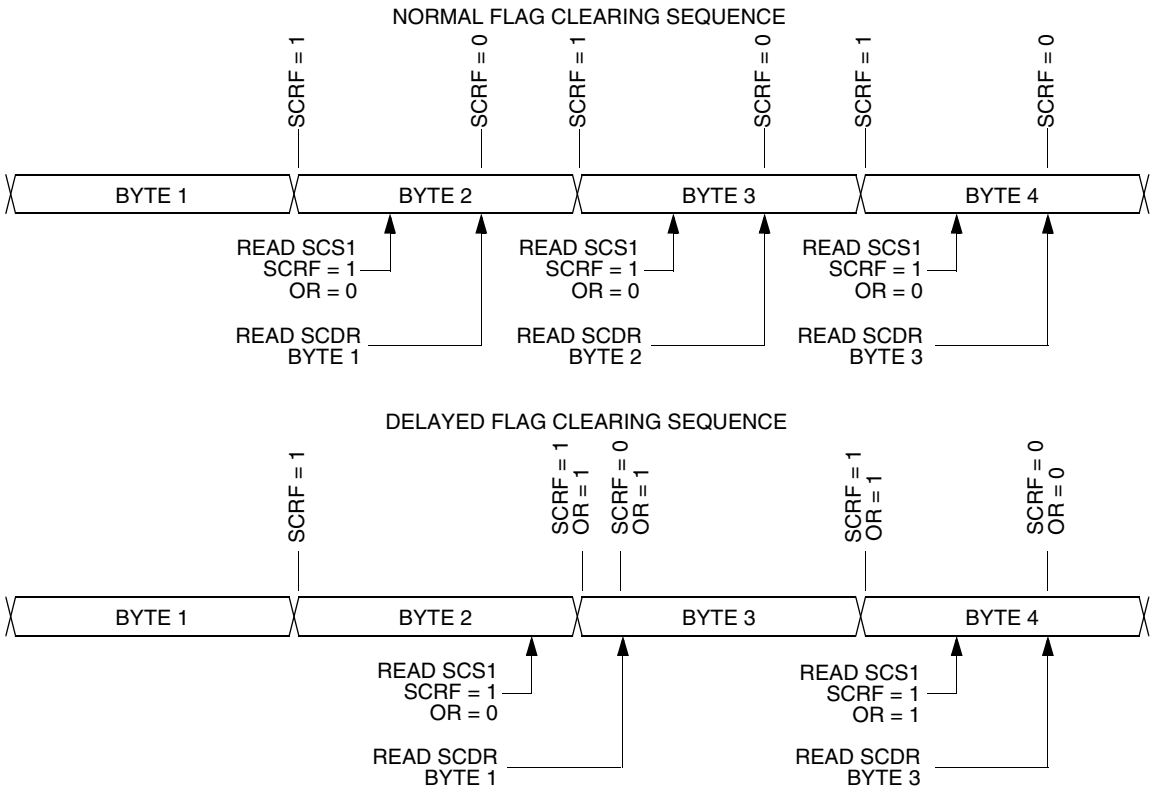


Figure 16-13. Flag Clearing Sequence

PE — Receiver Parity Error Bit

This clearable, read-only bit is set when the SCI detects a parity error in incoming data. PE generates a PE CPU interrupt request if the PEIE bit in SCC3 is also set. Clear the PE bit by reading SCS1 with PE set and then reading the SCDR. Reset clears the PE bit.

1 = Parity error detected

0 = No parity error detected

16.9.5 SCI Status Register 2

SCI status register 2 contains flags to signal the following conditions:

- Break character detected
- Incoming data

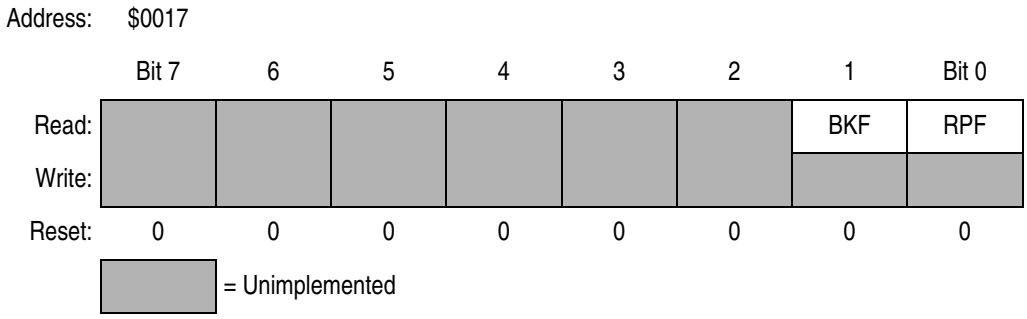


Figure 16-14. SCI Status Register 2 (SCS2)

BKF — Break Flag Bit

This clearable, read-only bit is set when the SCI detects a break character on the RxD pin. In SCS1, the FE and SCRF bits are also set. In 9-bit character transmissions, the R8 bit in SCC3 is cleared. BKF does not generate a CPU interrupt request. Clear BKF by reading SCS2 with BKF set and then reading the SCDR. Once cleared, BKF can become set again only after logic 1s again appear on the RxD pin followed by another break character. Reset clears the BKF bit.

1 = Break character detected
0 = No break character detected

RPF — Reception in Progress Flag Bit

This read-only bit is set when the receiver detects a logic 0 during the RT1 time period of the start bit search. RPF does not generate an interrupt request. RPF is reset after the receiver detects false start bits (usually from noise or a baud rate mismatch) or when the receiver detects an idle character. Polling RPF before disabling the SCI module or entering stop mode can show whether a reception is in progress.

1 = Reception in progress
0 = No reception in progress

16.9.6 SCI Data Register

The SCI data register (SCDR) is the buffer between the internal data bus and the receive and transmit shift registers. Reset has no effect on data in the SCI data register.

Address: $0018

	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:	Unaffected by reset							

Figure 16-15. SCI Data Register (SCDR)

R7/T7–R0/T0 — Receive/Transmit Data Bits

Reading the SCI data register (SCDR) accesses the read-only received data bits, R7:R0. Writing to the SCDR writes the data to be transmitted, T7:T0. Reset has no effect on the SCDR.

NOTE: *Do not use read/modify/write instructions on the SCI data register.*

16.9.7 SCI Baud Rate Register

The baud rate register (SCBR) selects the baud rate for both the receiver and the transmitter.

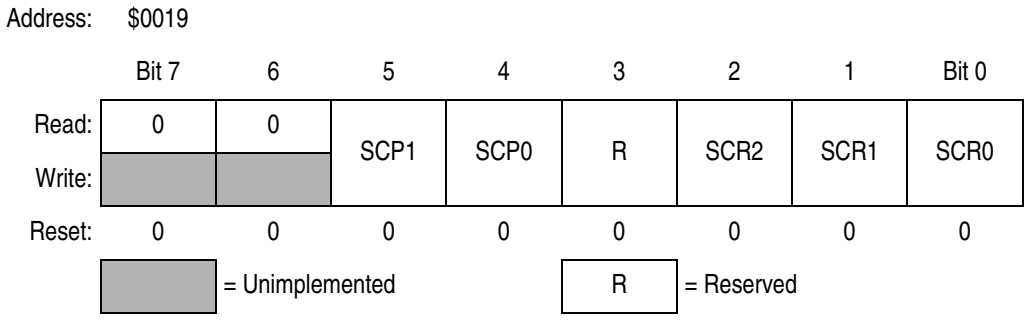


Figure 16-16. SCI Baud Rate Register (SCBR)

SCP1 and SCP0 — SCI Baud Rate Prescaler Bits

These read/write bits select the baud rate prescaler divisor as shown in **Table 16-6**. Reset clears SCP1 and SCP0.

Table 16-6. SCI Baud Rate Prescaling

SCP1 and SCP0	Prescaler Divisor (PD)
00	1
01	3
10	4
11	13

SCR2–SCR0 — SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate divisor as shown in **Table 16-7**. Reset clears SCR2–SCR0.

Table 16-7. SCI Baud Rate Selection

SCR2, SCR1, and SCR0	Baud Rate Divisor (BD)
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

Use this formula to calculate the SCI baud rate:

$$\text{baud rate} = \frac{\text{SCI clock source}}{64 \times \text{PD} \times \text{BD}}$$

where:

SCI clock source = f_{BUS} or CGMXCLK
(selected by SCIBDSRC bit in CONFIG2 register)
PD = prescaler divisor
BD = baud rate divisor

Table 16-8 shows the SCI baud rates that can be generated with a 4.9152-MHz bus clock when f_{BUS} is selected as SCI clock source.

Table 16-8. SCI Baud Rate Selection Examples

SCP1 and SCP0	Prescaler Divisor (PD)	SCR2, SCR1, and SCR0	Baud Rate Divisor (BD)	Baud Rate (f_{BUS} = 4.9152 MHz)
00	1	000	1	76,800
00	1	001	2	38,400
00	1	010	4	19,200
00	1	011	8	9600
00	1	100	16	4800
00	1	101	32	2400
00	1	110	64	1200
00	1	111	128	600
01	3	000	1	25,600
01	3	001	2	12,800
01	3	010	4	6400
01	3	011	8	3200
01	3	100	16	1600
01	3	101	32	800
01	3	110	64	400
01	3	111	128	200
10	4	000	1	19,200
10	4	001	2	9600
10	4	010	4	4800
10	4	011	8	2400
10	4	100	16	1200
10	4	101	32	600
10	4	110	64	300
10	4	111	128	150
11	13	000	1	5908
11	13	001	2	2954
11	13	010	4	1477
11	13	011	8	739
11	13	100	16	369
11	13	101	32	185
11	13	110	64	92
11	13	111	128	46

Section 17. Multi-Master IIC Interface (MMIIC)

17.1 Contents

17.2 Introduction 292
17.3 Features 293
17.4 I/O Pins 293
17.5 Multi-Master IIC System Configuration 295
17.6 Multi-Master IIC Bus Protocol 295
17.6.1 START Signal 296
17.6.2 Slave Address Transmission 296
17.6.3 Data Transfer 296
17.6.4 Repeated START Signal 297
17.6.5 STOP Signal 297
17.6.6 Arbitration Procedure 297
17.6.7 Clock Synchronization 298
17.6.8 Handshaking 298
17.6.9 Packet Error Code 299
17.7 MMIIC I/O Registers 299
17.7.1 MMIIC Address Register (MMADR) 299
17.7.2 MMIIC Control Register 1 (MMCR1) 301
17.7.3 MMIIC Control Register 2 (MMCR2) 303
17.7.4 MMIIC Status Register (MMSR) 305
17.7.5 MMIIC Data Transmit Register (MMDTR) 307
17.7.6 MMIIC Data Receive Register (MMDRR) 308
17.7.7 MMIIC CRC Data Register (MMCRCDR) 309
17.7.8 MMIIC Frequency Divider Register (MMFDR) 310
17.8 Program Algorithm 311
17.8.1 Data Sequence 312
17.9 SMBus Protocols with PEC and without PEC 313

17.9.1 Quick Command 313
17.9.2 Send Byte 313
17.9.3 Receive Byte 313
17.9.4 Write Byte/Word 314
17.9.5 Read Byte/Word 314
17.9.6 Process Call 315
17.9.7 Block Read/Write 315

17.10 SMBus Protocol Implementation 316

17.2 Introduction

The multi-master IIC (MMIIC) interface is a two wire, bidirectional serial bus which provides a simple, efficient way for data exchange between devices. The interface is designed for internal serial communication between the MCU and other IIC devices. It has hardware generated START and STOP signals; and byte by byte interrupt driven software algorithm.

This bus is suitable for applications which need frequent communications over a short distance between a number of devices. It also provides a flexibility that allows additional devices to be connected to the bus. The maximum data rate is 100k-bps, and the maximum communication distance and number of devices that can be connected is limited by a maximum bus capacitance of 400pF.

This MMIIC interface is also SMBus (System Management Bus) version 1.0 and 1.1 compatible, with hardware cyclic redundancy code (CRC) generation, making it suitable for smart battery applications.

For connection flexibility, two channels are available:

- Channel 0 — SDA0 and SCL0
- Channel 1 — SDA1 and SCL1

The two channels are multiplexed; only one channel is active at any one time.

17.3 Features

Features of the MMIC module include:

- Full SMBus version 1.0/1.1 compliance
- Multi-master IIC bus standard
- Software programmable for one of eight different serial clock frequencies
- Software controllable acknowledge bit generation
- Interrupt driven byte by byte data transfer
- Calling address identification interrupt
- Arbitration loss detection and no-ACK awareness in master mode and automatic mode switching from master to slave
- Auto detection of R/W bit and switching of transmit or receive mode accordingly
- Detection of START, repeated START, and STOP signals
- Auto generation of START and STOP condition in master mode
- Repeated start generation
- Master clock generator with eight selectable baud rates
- Automatic recognition of the received acknowledge bit
- Busy detection
- Software enabled 8-bit CRC generation/decoding

17.4 I/O Pins

The MMIIC module uses four I/O pins, shared with standard port I/O pins. The full name of the MMIIC I/O pins are listed in **Table 17-1**. The generic pin name appear in the text that follows.

The SDA0/SCL0 and SDA1/SCL1 pins are open-drain. When configured as general purpose output pins (PTB0–PTB3), pullup resistors must be connected to these pins.

Table 17-1. Pin Name Conventions

MMIIC Generic Pin Names:	Full MCU Pin Names:	Pin Selected for MMIIC Function By:
SDA0	PTB0/SDA0	MMEN and SDASCL1 bits in MMCR1 ($0049)
SCL0	PTB1/SCL0	
SDA1	PTB2/SDA1/TxD	ENSCI bit in SCC1 ($0013); MMEN and SDASCL1 bits in MMCR1 ($0049)
SCL1	PTB3/SCL1/RxD	

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0048	MMIIC Address Register (MMADR)	Read:	MMAD7	MMAD6	MMAD5	MMAD4	MMAD3	MMAD2	MMAD1	MMEXTAD
		Write:								
		Reset:	1	0	1	0	0	0	0	0
$0049	MMIIC Control Register 1 (MMCR1)	Read:	MMEN	MMIEN	0	0	MMTXAK	REPSEN	MMCRCBYTE	SDASCL1
		Write:			MMCLRBB					
		Reset:	0	0	0	0	0	0	0	0
$004A	MMIIC Control Register 2 (MMCR2)	Read:	MMALIF	MMNAKIF	MMBB	MMAST	MMRW	0	0	MMCRCEF
		Write:	0	0						
		Reset:	0	0	0	0	0	0	0	Unaffected
$004B	MMIIC Status Register (MMSR)	Read:	MMRXIF	MMTXIF	MMATCH	MMSRW	MMRXAK	MMCRCBF	MMTXBE	MMRXBF
		Write:	0	0						
		Reset:	0	0	0	0	1	0	1	0
$004C	MMIIC Data Transmit Register (MMDTR)	Read:	MMTD7	MMTD6	MMTD5	MMTD4	MMTD3	MMTD2	MMTD1	MMTD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$004D	MMIIC Data Receive Register (MDDRR)	Read:	MMRD7	MMRD6	MMRD5	MMRD4	MMRD3	MMRD2	MMRD1	MMRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$004E	MMIIC CRC Data Register (MMCRDR)	Read:	MMCRCD7	MMCRCD6	MMCRCD5	MMCRCD4	MMCRCD3	MMCRCD2	MMCRCD1	MMCRCD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$004F	MMIIC Frequency Divider Register (MMFDR)	Read:	0	0	0	0	0	MMBR2	MMBR1	MMBR0
		Write:								
		Reset:	0	0	0	0	0	1	0	0

= Unimplemented

Figure 17-1. MMIIC I/O Register Summary

17.5 Multi-Master IIC System Configuration

The multi-master IIC system uses a serial data line SDA and a serial clock line SCL for data transfer. All devices connected to it must have open collector (drain) outputs and the logical-AND function is performed on both lines by two pull-up resistors.

17.6 Multi-Master IIC Bus Protocol

Normally a standard communication is composed of four parts:

1. START signal,
2. slave address transmission,
3. data transfer, and
4. STOP signal.

These are described briefly in the following sections and illustrated in **Figure 17-2**.

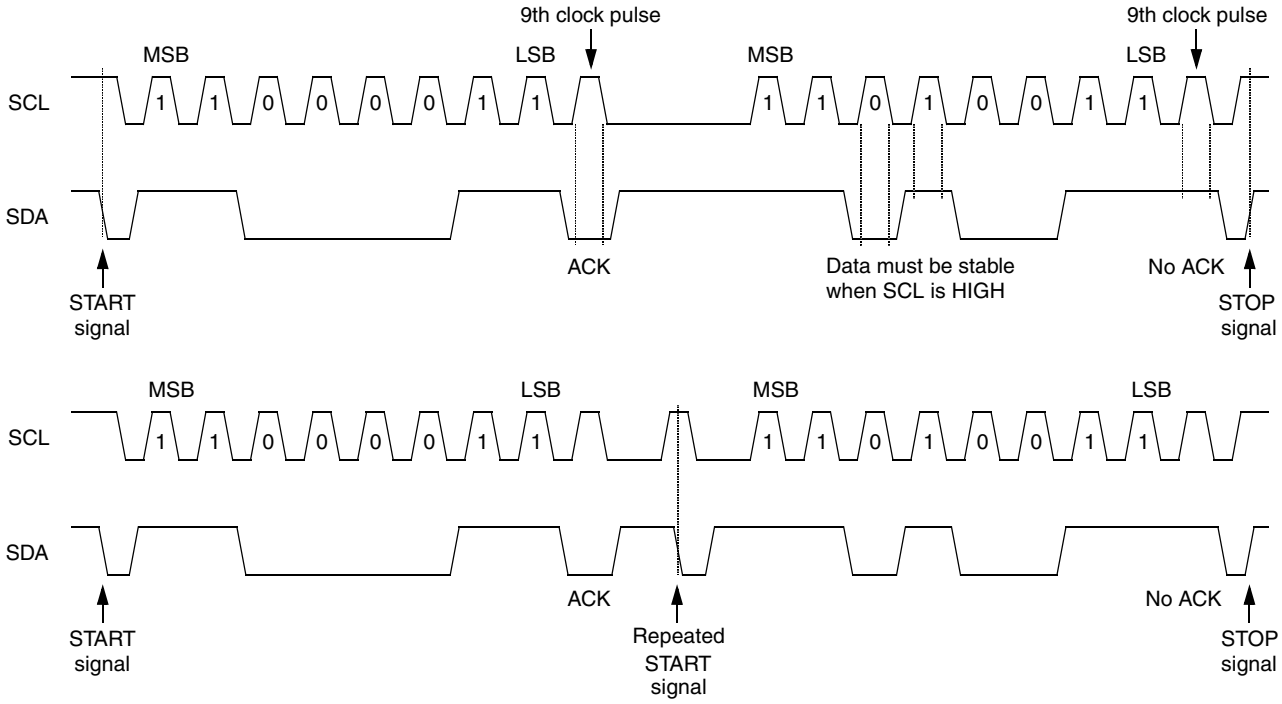


Figure 17-2. Multi-Master IIC Bus Transmission Signal Diagram

17.6.1 START Signal

When the bus is free, (i.e. no master device is engaging the bus — both SCL and SDA lines are at logic high) a master may initiate communication by sending a START signal. As shown in **Figure 17-2**, a START signal is defined as a high to low transition of SDA while SCL is high. This signal denotes the beginning of a new data transfer (each data transfer may contain several bytes of data) and wakes up all slaves.

17.6.2 Slave Address Transmission

The first byte transferred immediately after the START signal is the slave address transmitted by the master. This is a 7-bit calling address followed by a R/W-bit. The R/W-bit dictates to the slave the desired direction of the data transfer. A logic 0 indicates that the master wishes to transmit data to the slave; a logic 1 indicates that the master wishes to receive data from the slave.

Only the slave with a matched address will respond by sending back an acknowledge bit by pulling SDA low on the 9th clock cycle.
(See **Figure 17-2**.)

17.6.3 Data Transfer

Once a successful slave addressing is achieved, the data transfer can proceed byte by byte in the direction specified by the R/W-bit sent by the calling master.

Each data byte is 8 bits. Data can be changed only when SCL is low and must be held stable when SCL is high as shown in **Figure 17-2**. The MSB is transmitted first and each byte has to be followed by an acknowledge bit. This is signalled by the receiving device by pulling the SDA low on the 9th clock cycle. Therefore, one complete data byte transfer requires 9 clock cycles.

If the slave receiver does not acknowledge the master, the SDA line should be left high by the slave. The master can then generate a STOP signal to abort the data transfer or a START signal (repeated START) to commence a new transfer.

If the master receiver does not acknowledge the slave transmitter after a byte has been transmitted, it means an “end of data” to the slave. The slave should release the SDA line for the master to generate a STOP or START signal.

17.6.4 Repeated START Signal

As shown in **Figure 17-2**, a repeated START signal is used to generate START signal without first generating a STOP to terminate the communication. This is used by the master to communicate with another slave or with the same slave in a different mode (transmit/receive mode) without releasing the bus.

17.6.5 STOP Signal

The master can terminate the communication by generating a STOP signal to free the bus. However, the master may generate a START signal followed by a calling command without first generating a STOP signal. This is called repeat START. A STOP signal is defined as a low to high transition of SDA while SCL is at logic high (see **Figure 17-2**).

17.6.6 Arbitration Procedure

The interface circuit is a multi-master system which allows more than one master to be connected. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock. The clock low period is equal to the longest clock low period and the clock high period is equal to the shortest one among the masters. A data arbitration procedure determines the priority. A master will lose arbitration if it transmits a logic 1 while another transmits a logic 0. The losing master will immediately switch over to slave receive mode and stops its data and clock outputs. The transition from master to slave will not generate a STOP condition. Meanwhile a software bit will be set by hardware to indicates loss of arbitration.

17.6.7 Clock Synchronization

Since wired-AND logic is performed on SCL line, a high to low transition on the SCL line will affect the devices connected to the bus. The devices start counting their low period once a device's clock has gone low, it will hold the SCL line low until the clock high state is reached. However, the change of low to high in this device clock may not change the state of the SCL line if another device clock is still in its low period. Therefore the synchronized clock SCL will be held low by the device which last releases SCL to logic high. Devices with shorter low periods enter a high wait state during this time. When all devices concerned have counted off their low period, the synchronized SCL line will be released and go high, and all devices will start counting their high periods. The first device to complete its high period will again pull the SCL line low. **Figure 17-3** illustrates the clock synchronization waveforms.

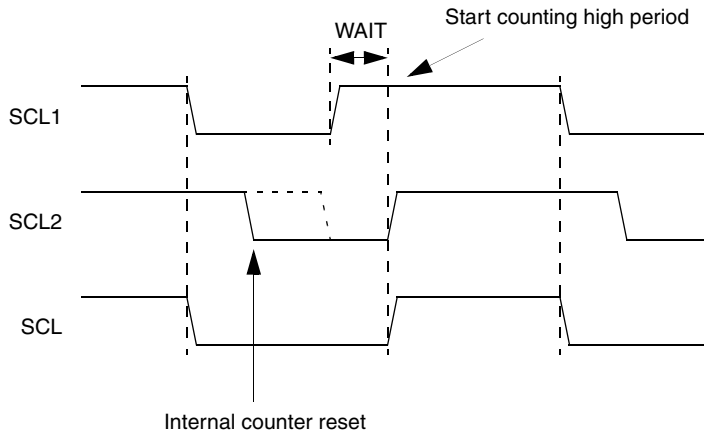


Figure 17-3. Clock Synchronization

17.6.8 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfer. A slave device may hold the SCL low after completion of one byte data transfer and will halt the bus clock, forcing the master clock into a wait state until the slave releases the SCL line.

17.6.9 Packet Error Code

The packet error code (PEC) for the MMIIC interface is in the form a cyclic redundancy code (CRC). The PEC is generated by hardware for every transmitted and received byte of data. The transmission of the generated PEC is controlled by user software.

The CRC data register, MMCRCDR, contains the generated PEC byte, with three other bits in the MMIIC control registers and status register monitoring and controlling the PEC byte.

17.7 MMIIC I/O Registers

These I/O registers control and monitor MMIIC operation:

- MMIIC address register (MMADR) — $0048
- MMIIC control register 1 (MMCR1) — $0049
- MMIIC control register 2 (MMCR2) — $004A
- MMIIC status register (MMSR) — $004B
- MMIIC data transmit register (MMDTR) — $004C
- MMIIC data receive register (MMDRR) — $004D
- MMIIC CRC data register (MMCRCDR) — $004E
- MMIIC frequency divide register (MMFDR) — $004F

17.7.1 MMIIC Address Register (MMADR)

Address: $0048

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	MMAD7	MMAD6	MMAD5	MMAD4	MMAD3	MMAD2	MMAD1	MMEXTAD
Reset:	1	0	1	0	0	0	0	0

Figure 17-4. MMIIC Address Register (MMADR)

MMAD[7:1] — Multi-Master Address

These seven bits represent the MMIIC interface's own specific slave address when in slave mode, and the calling address when in master mode. Software must update MMAD[7:1] as the calling address while entering master mode and restore its own slave address after master mode is relinquished. This register is cleared as $A0 upon reset.

MMEXTAD — Multi-Master Expanded Address

This bit is set to expand the address of the MMIIC in slave mode. When set, the MMIIC will acknowledge the following addresses from a calling master: $MMAD[7:1], 0000000, and 0001100.
Reset clears this bit.

1 = MMIIC responds to the following calling addresses: $MMAD[7:1], 0000000, and 0001100.
0 = MMIIC responds to address $MMAD[7:1]

For example, when MMADR is configured as:

MMAD7	MMAD6	MMAD5	MMAD4	MMAD3	MMAD2	MMAD1	MMEXTAD
1	1	0	1	0	1	0	1

The MMIIC module will respond to the calling address:

Bit 7	6	5	4	3	2	Bit 1
1	1	0	1	0	1	0

or the general calling address:

0	0	0	0	0	0	0

or the calling address:

Bit 7	6	5	4	3	2	Bit 1
0	0	0	1	1	0	0

Note that bit-0 of the 8-bit calling address is the MMRW bit from the calling master.

17.7.2 MMIIC Control Register 1 (MMCR1)

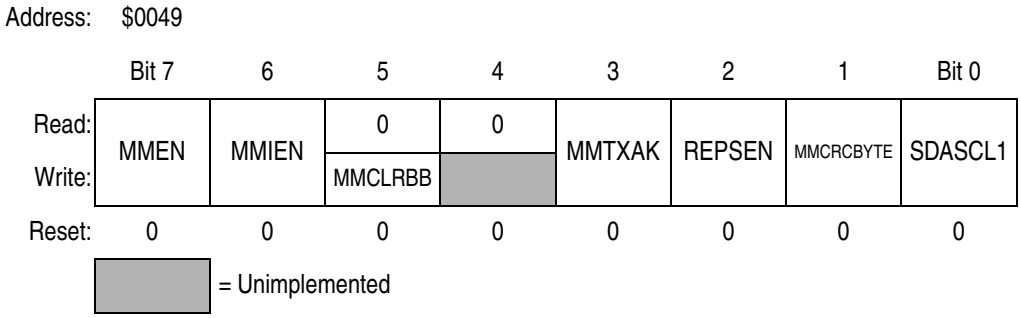


Figure 17-5. MMIIC Control Register 1 (MMCR1)

MMEN — MMIIC Enable

This bit is set to enable the Multi-master IIC module. When MMEN = 0, module is disabled and all flags will restore to its power-on default states. Reset clears this bit.

1 = MMIIC module enabled
0 = MMIIC module disabled

MMIEN — MMIIC Interrupt Enable

When this bit is set, the MMTXIF, MMRXIF, MMALIF, and MMNAKIF flags are enabled to generate an interrupt request to the CPU. When MMIEN is cleared, the these flags are prevented from generating an interrupt request. Reset clears this bit.

1 = MMTXIF, MMRXIF, MMALIF, and/or MMNAKIF bit set will generate interrupt request to CPU
0 = MMTXIF, MMRXIF, MMALIF, and/or MMNAKIF bit set will not generate interrupt request to CPU

MMCLRBB — MMIIC Clear Busy Flag

Writing a logic 1 to this write-only bit clears the MMBB flag. MMCLRBB always reads as a logic 0. Reset clears this bit.

1 = Clear MMBB flag
0 = No affect on MMBB flag

MMTXAK — MMIIC Transmit Acknowledge Enable

This bit is set to disable the MMIIC from sending out an acknowledge signal to the bus at the 9th clock bit after receiving 8 data bits. When MMTXAK is cleared, an acknowledge signal will be sent at the 9th clock bit. Reset clears this bit.

1 = MMIIC does not send acknowledge signals at 9th clock bit
0 = MMIIC sends acknowledge signal at 9th clock bit

REPSEN — Repeated Start Enable

This bit is set to enable repeated START signal to be generated when in master mode transfer (MMAST = 1). The REPSEN bit is cleared by hardware after the completion of repeated START signal or when the MMAST bit is cleared. Reset clears this bit.

1 = Repeated START signal will be generated if MMAST bit is set
0 = No repeated START signal will be generated

MMCRCBYTE — MMIIC CRC Byte

In receive mode, this bit is set by software to indicate that the next receiving byte will be the packet error checking (PEC) data.

In master receive mode, after completion of CRC generation on the received PEC data, an acknowledge signal is sent if MMTXAK = 0; no acknowledge is sent If MMTXAK = 1.

In slave receive mode, no acknowledge signal is sent if a CRC error is detected on the received PEC data. If no CRC error is detected, an acknowledge signal is sent if MMTXAK = 0; no acknowledge is sent If MMTXAK = 1.

Under normal operation, the user software should clear MMTXAK bit before setting MMCRCBYTE bit to ensure that an acknowledge signal is sent when no CRC error is detected.

The MMCRCBYTE bit should not be set in transmit mode. This bit is cleared by the next START signal. Reset also clears this bit.

1 = Next receiving byte is the packet error checking (PEC) data
0 = Next receiving byte is not PEC data

SDASCL1 — SDA and SCL I/O Pin Select

This bit selects either SDA0 and SCL0, or SDA1 and SCL1, for MMIIC I/O pins when MMIIC module is enabled (MMEN = 1). If the SCI module is enabled (ENSCI = 0), the SDA1 and SCL1 pins are not available for MMIIC.
Reset clears SDASCL1 bit.

1 = MMIIC module uses SDA1 and SCL1 I/O pins
0 = MMIIC module uses SDA0 and SCL0 I/O pins

17.7.3 MMIIC Control Register 2 (MMCR2)

Address: $004A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MMALIF	MMNAKIF	MMBB	MMAST	MMRW	0	0	MMCRCEF
Write:	0	0						
Reset:	0	0	0	0	0	0	0	Unaffected

= Unimplemented

Figure 17-6. MMIIC Control Register 2 (MMCR2)

MMALIF — Arbitration Loss Interrupt Flag

This flag is set when software attempt to set MMAST but the MMBB has been set by detecting the start condition on the lines or when the MMIIC is transmitting a "1" to SDA line but detected a "0" from SDA line in master mode — an arbitration loss. This bit generates an interrupt request to the CPU if the MMIEN bit in MMCR1 is set. This bit is cleared by writing "0" to it or by reset.

1 = Lost arbitration in master mode
0 = No arbitration lost

MMNAKIF — No AcKnowledge Interrupt Flag (Master Mode)

This flag is only set in master mode (MMAST = 1) when there is no acknowledge bit detected after one data byte or calling address is transferred. This flag also clears MMAST. MMNAKIF generates an interrupt request to CPU if the MMIEN bit in MMCR1 is set. This bit is cleared by writing "0" to it or by reset.

1 = No acknowledge bit detected
0 = Acknowledge bit detected

MMBB — MMIIC Bus Busy Flag

This flag is set after a start condition is detected (bus busy), and is cleared when a stop condition (bus idle) is detected or the MMIIC is disabled. Reset clears this bit.

1 = Start condition detected
0 = Stop condition detected or MMIIC is disabled

MMAST — MMIIC Master Control

This bit is set to initiate a master mode transfer. In master mode, the module generates a start condition to the SDA and SCL lines, followed by sending the calling address stored in MMADR.
When the MMAST bit is cleared by MMNAKIF set (no acknowledge) or by software, the module generates the stop condition to the lines after the current byte is transmitted.
If an arbitration loss occurs (MMALIF = 1), the module reverts to slave mode by clearing MMAST, and releasing SDA and SCL lines immediately.
This bit is cleared by writing "0" to it or by reset.

1 = Master mode operation
0 = Slave mode operation

MMRW — MMIIC Master Read/Write

This bit is transmitted out as bit 0 of the calling address when the module sets the MMAST bit to enter master mode. The MMRW bit determines the transfer direction of the data bytes that follows. When it is "1", the module is in master receive mode. When it is "0", the module is in master transmit mode. Reset clears this bit.

1 = Master mode receive
0 = Master mode transmit

MMCRCEF — MMIIC CRC Error Flag

This flag is set when a CRC error is detected, and cleared when no CRC error is detected. The MMCRCEF is only meaningful after receiving a PEC data. This flag is unaffected by reset.

1 = CRC error detected on PEC byte
0 = No CRC error detected on PEC byte

17.7.4 MMIIC Status Register (MMSR)

Address: $004B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MMRXIF	MMTXIF	MMATCH	MMSRW	MMRXAK	MMCRCBF	MMTXBE	MMRXBF
Write:	0	0						
Reset:	0	0	0	0	1	0	1	0

= Unimplemented

Figure 17-7. MMIIC Status Register (MMSR)

MMRXIF — MMIIC Receive Interrupt Flag

This flag is set after the data receive register (MMDRR) is loaded with a new received data. Once the MMDRR is loaded with received data, no more received data can be loaded to the MMDRR register until the CPU reads the data from the MMDRR to clear MMRXBF flag. MMRXIF generates an interrupt request to CPU if the MMIEN bit in MMCR is also set. This bit is cleared by writing "0" to it or by reset; or when the MMEN = 0.

1 = New data in data receive register (MMDRR)
0 = No data received

MMTXIF — MMIIC Transmit Interrupt Flag

This flag is set when data in the data transmit register (MMDTR) is downloaded to the output circuit, and that new data can be written to the MMDTR. MMTXIF generates an interrupt request to CPU if the MMIEN bit in MMCR is also set. This bit is cleared by writing "0" to it or when the MMEN = 0.

1 = Data transfer completed
0 = Data transfer in progress

MMATCH — MMIIC Address Match Flag

This flag is set when the received data in the data receive register (MMDRR) is a calling address which matches with the address or its extended addresses (MMEXTAD = 1) specified in the address register (MMADR). The MMATCH flag is set at the 9th clock of the calling address and will be cleared on the 9th clock of the next receiving data. Note: slave transmits do not clear MMATCH.

1 = Received address matches MMADR
0 = Received address does not match

MMSRW — MMIIC Slave Read/Write Select

This bit indicates the data direction when the module is in slave mode. It is updated after the calling address is received from a master device. MMSRW = 1 when the calling master is reading data from the module (slave transmit mode). MMSRW = 0 when the master is writing data to the module (receive mode).

1 = Slave mode transmit
0 = Slave mode receive

MMRXAK — MMIIC Receive Acknowledge

When this bit is cleared, it indicates an acknowledge signal has been received after the completion of eight data bits transmission on the bus. When MMRXAK is set, it indicates no acknowledge signal has been detected at the 9th clock; the module will release the SDA line for the master to generate STOP or repeated START condition. Reset sets this bit.

1 = No acknowledge signal received at 9th clock
0 = Acknowledge signal received at 9th clock

MMCRCBF — CRC Data Buffer Full Flag

This flag is set when the CRC data register (MMCRCDR) is loaded with a CRC byte for the current received or transmitted data.

In transmit mode, after a byte of data has been sent (MMTXIF = 1), the MMCRCBF will be set when the CRC byte has been generated and ready in the MMCRCDR. The content of the MMCRCDR should be copied to the MMDTR for transmission.

In receive mode, the MMCRCBF is set when the CRC byte has been generated and ready in MMCRCDR, for the current byte of received data.

The MMCRCBF bit is cleared when the CRC data register is read. Reset also clears this bit.

1 = Data ready in CRC data register (MMCRCDR)
0 = Data not ready in CRC data register (MMCRCDR)

MMTXBE — MMIIC Transmit Buffer Empty

This flag indicates the status of the data transmit register (MMDTR). When the CPU writes the data to the MMDTR, the MMTXBE flag will be cleared. MMTXBE is set when MMDTR is emptied by a transfer of its data to the output circuit. Reset sets this bit.

1 = Data transmit register empty
0 = Data transmit register full

MMRXBF — MMIIC Receive Buffer Full

This flag indicates the status of the data receive register (MMDRR). When the CPU reads the data from the MMDRR, the MMRXBF flag will be cleared. MMRXBF is set when MMDRR is full by a transfer of data from the input circuit to the MMDRR. Reset clears this bit.

1 = Data receive register full
0 = Data receive register empty

17.7.5 MMIIC Data Transmit Register (MMDTR)

Address: $004C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	MMTD7	MMTD6	MMTD5	MMTD4	MMTD3	MMTD2	MMTD1	MMTD0
Reset:	0	0	0	0	0	0	0	0

Figure 17-8. MMIIC Data Transmit Register (MMDTR)

When the MMIIC module is enabled, MMEN = 1, data written into this register depends on whether module is in master or slave mode.

In slave mode, the data in MMDTR will be transferred to the output circuit when:

- the module detects a matched calling address (MMATCH = 1), with the calling master requesting data (MMSRW = 1); or
- the previous data in the output circuit has be transmitted and the receiving master returns an acknowledge bit, indicated by a received acknowledge bit (MMRXAK = 0).

If the calling master does not return an acknowledge bit (MMRXAK = 1), the module will release the SDA line for master to generate a STOP or repeated START condition. The data in the MMDTR will not be transferred to the output circuit until the next calling from a master. The transmit buffer empty flag remains cleared (MMTXBE = 0).

In master mode, the data in MMDTR will be transferred to the output circuit when:

- the module receives an acknowledge bit (MMRXAK = 0), after setting master transmit mode (MMRW = 0), and the calling address has been transmitted; or
- the previous data in the output circuit has be transmitted and the receiving slave returns an acknowledge bit, indicated by a received acknowledge bit (MMRXAK = 0).

If the slave does not return an acknowledge bit (MMRXAK = 1), the master will generate a STOP or repeated START condition. The data in the MMDTR will not be transferred to the output circuit. The transmit buffer empty flag remains cleared (MMTXBE = 0).

The sequence of events for slave transmit and master transmit are illustrated in **Figure 17-12**.

17.7.6 MMIIC Data Receive Register (MMDRR)

Address: $004D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MMRD7	MMRD6	MMRD5	MMRD4	MMRD3	MMRD2	MMRD1	MMRD0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 17-9. MMIIC Data Receive Register (MMDRR)

When the MMIIC module is enabled, MMEN = 1, data in this read-only register depends on whether module is in master or slave mode.

In slave mode, the data in MMDRR is:

- the calling address from the master when the address match flag is set (MMATCH = 1); or
- the last data received when MMATCH = 0.

In master mode, the data in the MMDRR is:

- the last data received.

When the MMDRR is read by the CPU, the receive buffer full flag is cleared (MMRXBF = 0), and the next received data is loaded to the MMDRR. Each time when new data is loaded to the MMDRR, the MMRXIF interrupt flag is set, indicating that new data is available in MMDRR.

The sequence of events for slave receive and master receive are illustrated in **Figure 17-12**.

17.7.7 MMIIC CRC Data Register (MMCRCDR)

Address: $004E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MMCRCD7	MMCRCD6	MMCRCD5	MMCRCD4	MMCRCD3	MMCRCD2	MMCRCD1	MMCRCD0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 17-10. MMIIC CRC Data Register (MMCRCDR)

When the MMIIC module is enabled, MMEN = 1, and the CRC buffer full flag is set (MMCRCBF = 1), data in this read-only register contains the generated CRC byte for the last byte of received or transmitted data.

A CRC byte is generated for each received and transmitted data byte and loaded to the CRC data register. The MMCRCBF bit will be set to indicate the CRC byte is ready in the CRC data register.

Reading the CRC data register clears the MMCRCBF bit. If the CRC data register is not read, the MMCRCBF bit will be cleared by hardware before the next CRC byte is loaded.

17.7.8 MMIIC Frequency Divider Register (MMFDR)

Address: $004F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	MMBR2	MMBR1	MMBR0
Write:								
Reset:	0	0	0	0	0	1	0	0

= Unimplemented

Figure 17-11. MMIIC Frequency Divider Register (MMFDR)

The three bits in the frequency divider register (MMFDR) selects the divider to divide the bus clock to the desired baud rate for the MMIIC data transfer.

Table 17-2 shows the divider values for MMBR[2:0].

Table 17-2. MMIIC Baud Rate Selection

MMBR2	MMBR1	MMBR0	Divider	MMIIC Baud Rates for Bus Clocks:			
				8MHz	4MHz	2MHz	1MHz
0	0	0	20	400kHz	200kHz	100kHz	50kHz
0	0	1	40	200kHz	100kHz	50kHz	25kHz
0	1	0	80	100kHz	50kHz	25kHz	12.5kHz
0	1	1	160	50kHz	25kHz	12.5kHz	6.25kHz
1	0	0	320	25kHz	12.5kHz	6.25kHz	3.125kHz
1	0	1	640	12.5kHz	6.25kHz	3.125kHz	1.5625kHz
1	1	0	1280	6.25kHz	3.125kHz	1.5625kHz	0.78125kHz
1	1	1	2560	3.125kHz	1.5625kHz	0.78125kHz	0.3906kHz

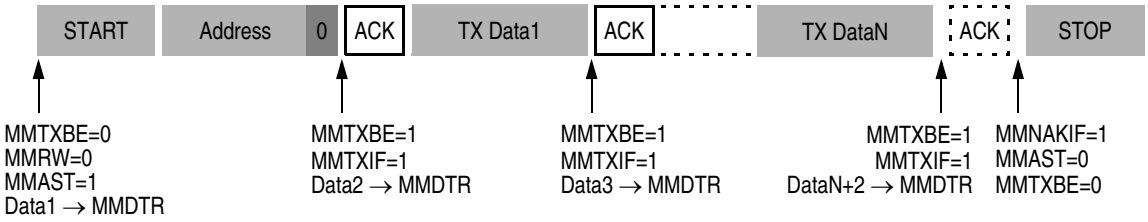
NOTE: *The frequency of the MMIIC baud rate is only guaranteed for 100kHz to 10kHz. The divider is available for the flexibility on bus frequency selection.*

17.8 Program Algorithm

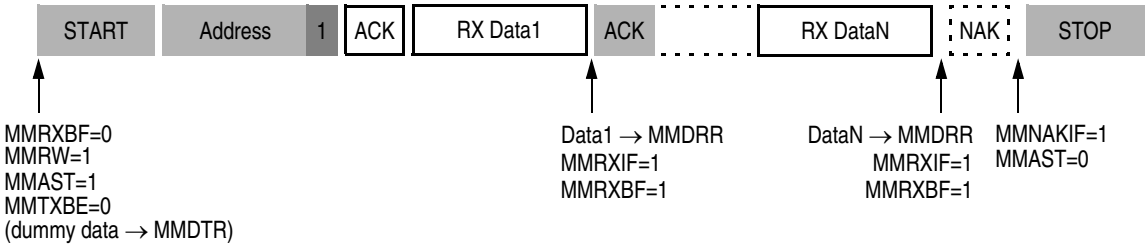
When the MMIIC module detects an arbitration loss in master mode, it releases both SDA and SCL lines immediately. But if there are no further STOP conditions detected, the module will hang up. Therefore, it is recommended to have time-out software to recover from this condition. The software can start the time-out counter by looking at the MMBB (bus busy) flag and reset the counter on the completion of one byte transmission. If a time-out has occurred, software can clear the MMEN bit (disable MMIIC module) to release the bus, and hence clear the MMBB flag. This is the only way to clear the MMBB flag by software if the module hangs up due to a no STOP condition received. The MMIIC can resume operation again by setting the MMEN bit.

17.8.1 Data Sequence

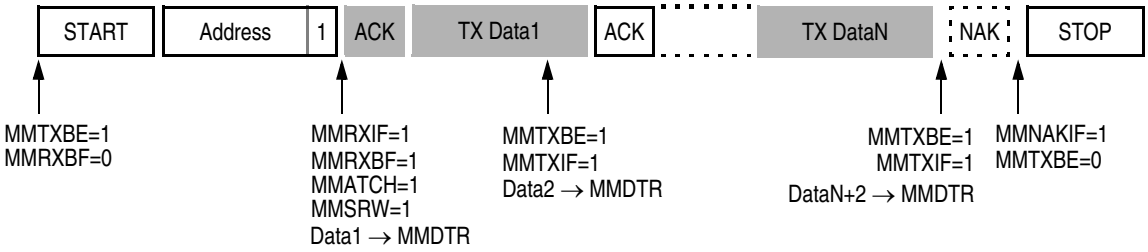
(a) Master Transmit Mode



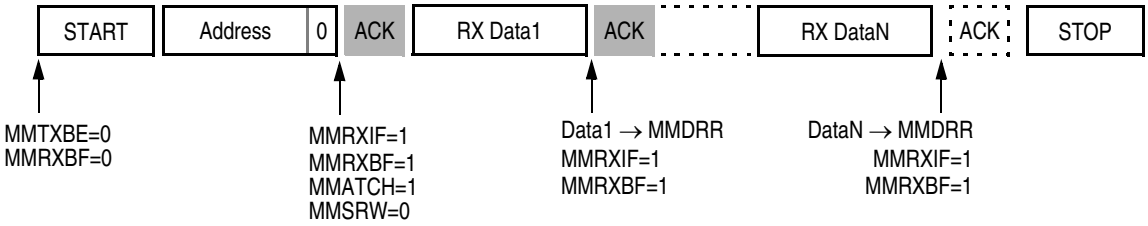
(b) Master Receive Mode



(c) Slave Transmit Mode



(d) Slave Receive Mode



Shaded data packets indicate transmissions by the MCU

Figure 17-12. Data Transfer Sequences for Master/Slave Transmit/Receive Modes

17.9 SMBus Protocols with PEC and without PEC

Following is a description of the various MMIIC bus protocols with and without a packet error code (PEC).

17.9.1 Quick Command

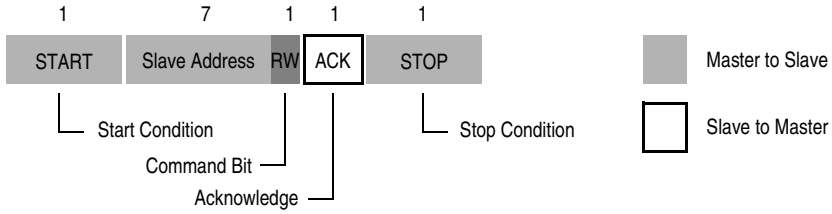


Figure 17-13. Quick Command

17.9.2 Send Byte

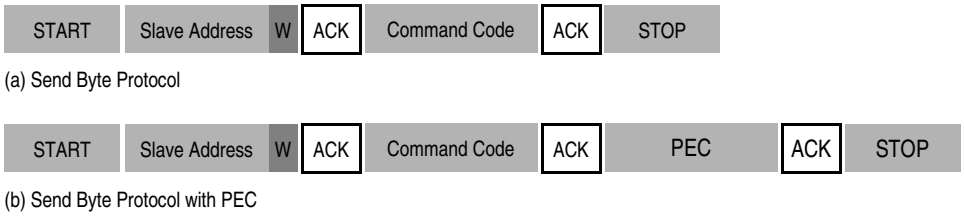


Figure 17-14. Send Byte

17.9.3 Receive Byte

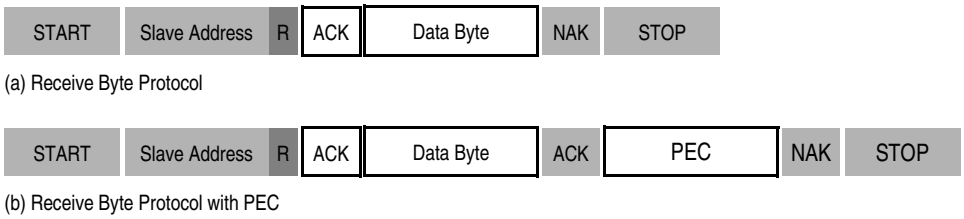


Figure 17-15. Receive Byte

17.9.4 Write Byte/Word

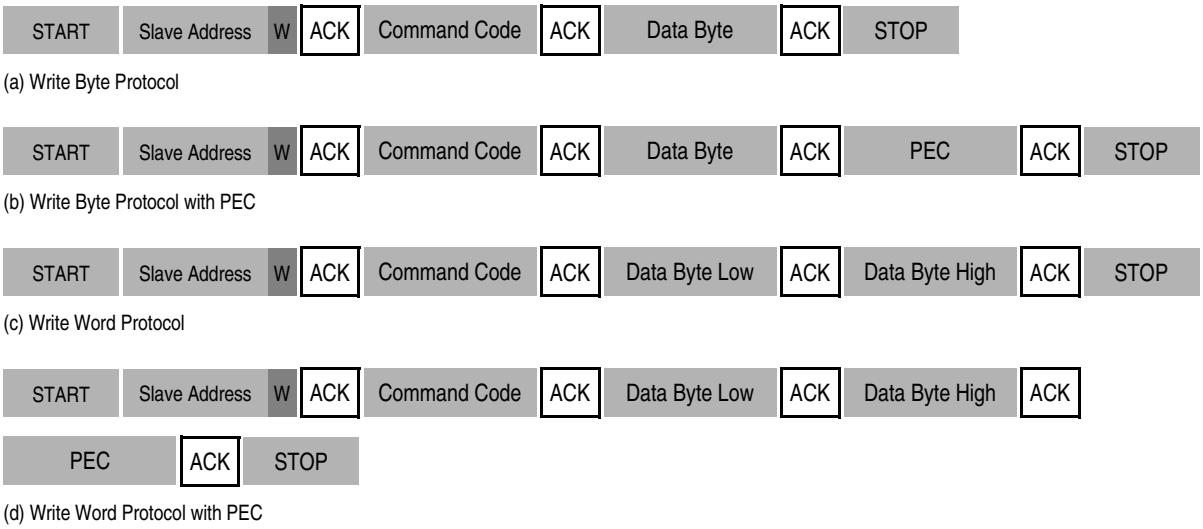


Figure 17-16. Write Byte/Word

17.9.5 Read Byte/Word

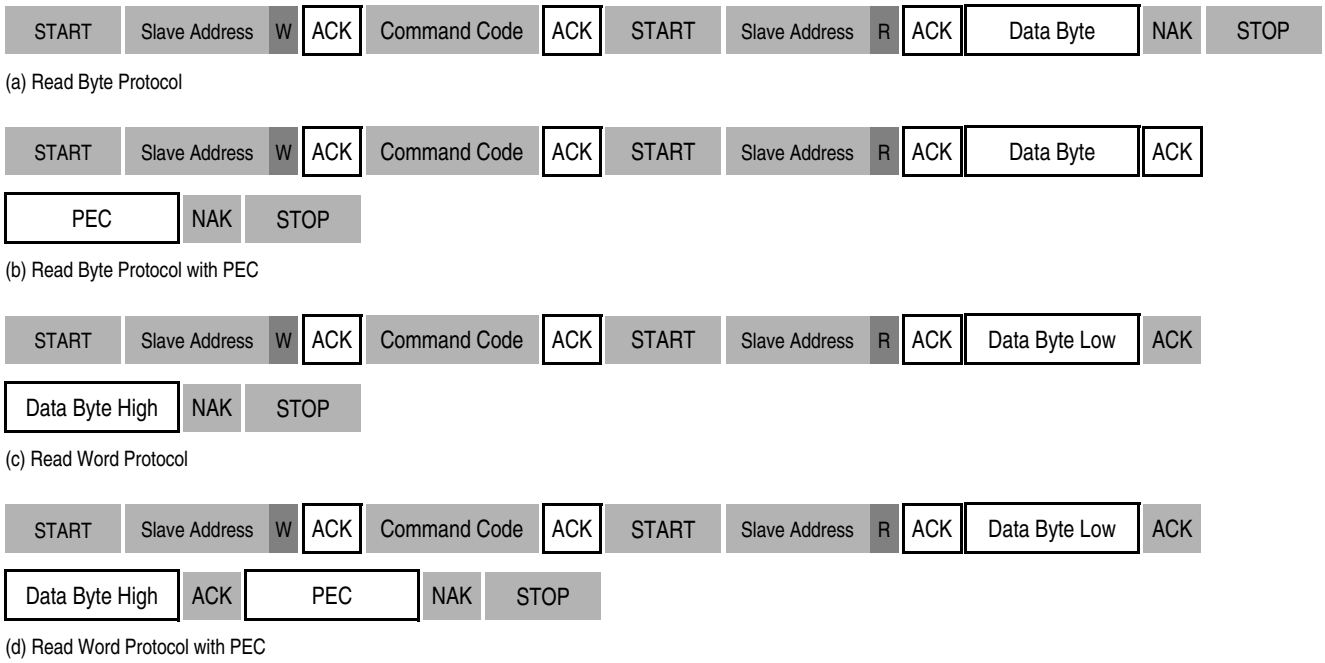


Figure 17-17. Read Byte/Word

17.9.6 Process Call

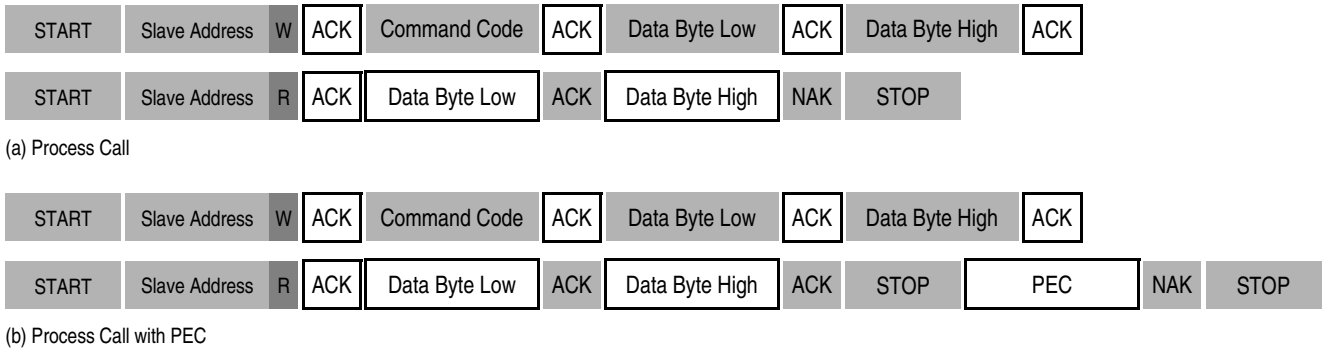


Figure 17-18. Process Call

17.9.7 Block Read/Write

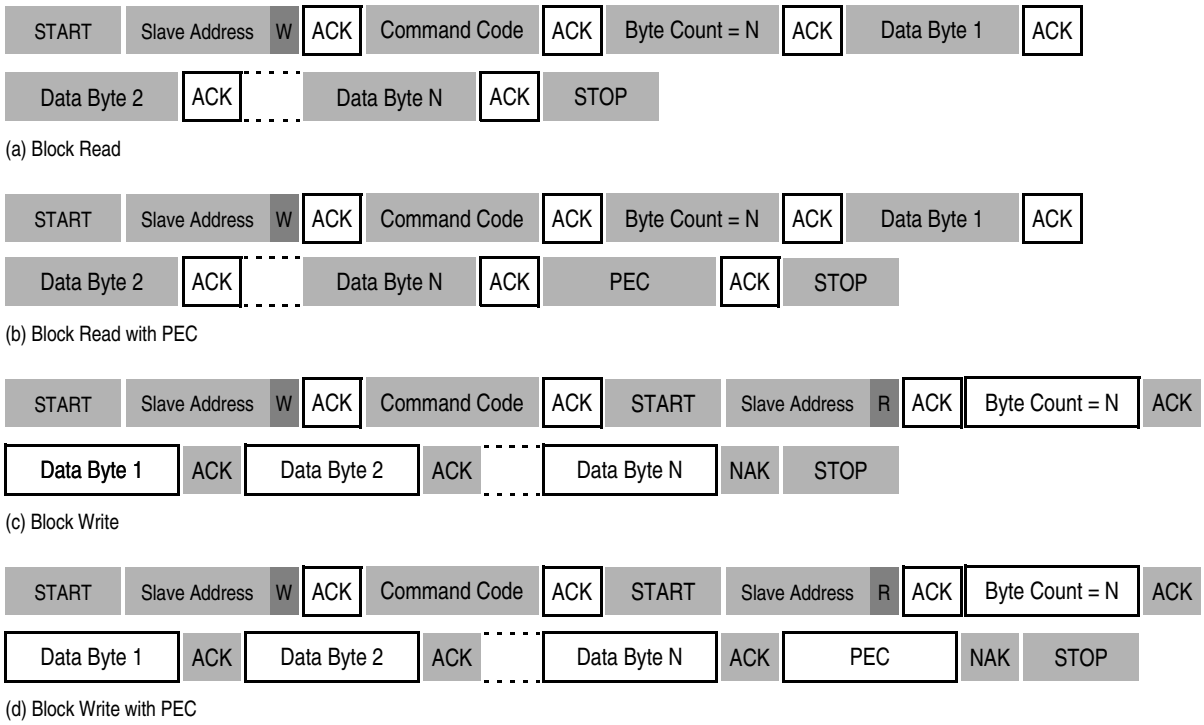


Figure 17-19. Block Read/Write

17.10 SMBus Protocol Implementation

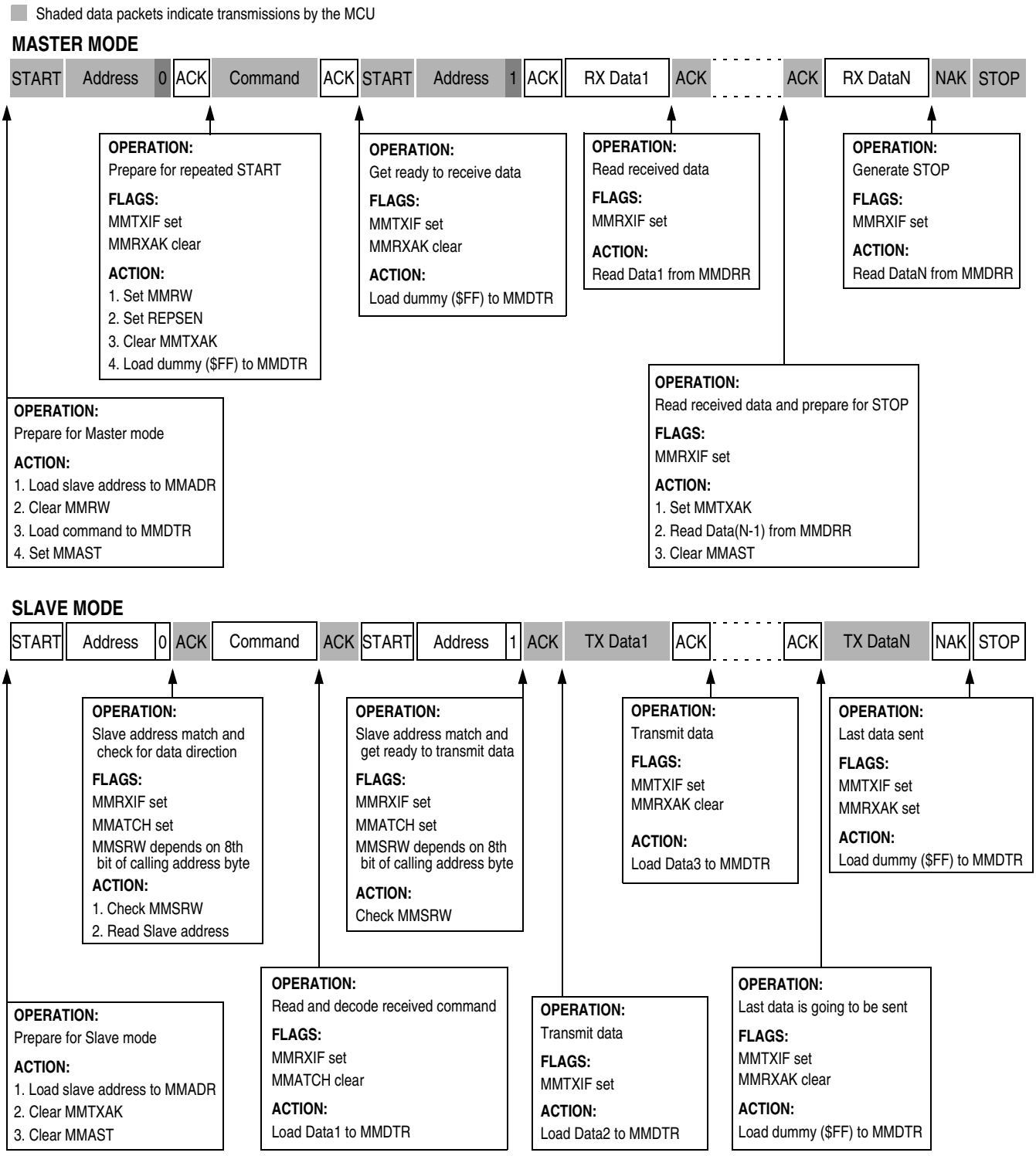


Figure 17-20. SMBus Protocol Implementation

Section 18. Input/Output (I/O) Ports

18.1 Contents

18.2 Introduction 317

18.3 Port A 320
18.3.1 Port A Data Register (PTA) 320
18.3.2 Data Direction Register A (DDRA) 321
18.3.3 Port A LED Control Register (LEDA) 323

18.4 Port B 323
18.4.1 Port B Data Register (PTB) 324
18.4.2 Data Direction Register B (DDRB) 325

18.5 Port C 327
18.5.1 Port C Data Register (PTC) 327
18.5.2 Data Direction Register C (DDRC) 329
18.5.3 Port C LED Control Register (LEDC) 330

18.6 Port D 331
18.6.1 Port D Data Register (PTD) 331
18.6.2 Data Direction Register D (DDRD) 332

18.2 Introduction

Thirty-one (31) bidirectional input-output (I/O) pins form four parallel ports. All I/O pins are programmable as inputs or outputs.

NOTE: *Connect any unused I/O pins to an appropriate logic level, either V_{DD} or V_{SS}. Although the I/O ports do not require termination for proper operation, termination reduces excess current consumption and the possibility of electrostatic damage.*

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA)	Read:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PTB)	Read:	0	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PTC)	Read:	PTC7	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
		Write:								
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PTD)	Read:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Write:								
		Reset:	Unaffected by reset							
$0004	Data Direction Register A (DDRA)	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB)	Read:	0	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0006	Data Direction Register C (DDRC)	Read:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007	Data Direction Register D (DDRD)	Read:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000C	Port-A LED Control Register (LEDA)	Read:	0	0	LEDA5	LEDA4	LEDA3	LEDA2	LEDA1	LEDA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000D	Port-C LED Control Register (LEDC)	Read:	LEDC7	LEDC6	LEDC5	LEDC4	LEDC3	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

Figure 18-1. I/O Port Register Summary

Table 18-1. Port Control Register Bits Summary

Port	Bit	DDR	Module Control			Pin
			Module	Register	Control Bit	
A	0	DDRA0	ADC	ADSCR ($0057)	ADCH[4:0]	PTA0/ATD2
	1	DDRA1				PTA1/ATD3
	2	DDRA2				PTA2/ATD4
	3	DDRA3				PTA3/ATD5
	4	DDRA4				PTA4/ATD6
	5	DDRA5				PTA5/ATD7
	6	DDRA6	TIM1	T1SC0 ($0025)	ELS0B:ELS0A	PTA6/T1CH0
	7	DDRA7		T1SC1 ($0028)	ELS1B:ELS1A	PTA7/T1CH1
B	0	DDRB0	MBUS	MMCR1 ($0049)	MMEN SDASCL1	PTB0/SDA0(1)
	1	DDRB1				PTB1/SCL0(1)
	2	DDRB2	SCI	SCC1 ($0013)(2)	ENSCI	PTB2/SDA1/TxD(1)
	3	DDRB3	MBUS	MMCR1 ($0049)	MMEN SDASCL1	PTB3/SCL1/RxD(1)
	4	DDRB4	TIM2	T2SC0 ($0030)	ELS0B:ELS0A	PTB4/T2CH0
	5	DDRB5		T2SC1 ($0033)	ELS1B:ELS1A	PTB5/T2CH1
	6	DDRB6	IRQ	—	—	PTB6/$\overline{\text{IRQ2}}$
C	0	DDRC0	ANALOG	CONFIG2 ($001D)(1)	CDOEN	PTC0/PWM0/CD
			PWM	PWMCR ($0051)	PCH0	
	1	DDRC1			PCH1	PTC1/PWM1
	2	DDRC2			PCH2	PTC2/PWM2
	3	DDRC3	ADC	ADSCR ($0057)	ADCH[4:0]	PTC3/ATD8
	4	DDRC4				PTC4/ATD9
	5	DDRC5				PTC5/ATD10
	6	DDRC6				PTC6/ATD11
	7	DDRC7				PTC7/ATD12
D	0	DDRD0	KBI	KBIER ($001B)	KBIE0	PTD0/KBI0
	1	DDRD1			KBIE1	PTD1/KBI1
	2	DDRD2			KBIE2	PTD2/KBI2
	3	DDRD3			KBIE3	PTD3/KBI3
	4	DDRD4			KBIE4	PTD4/KBI4
	5	DDRD5			KBIE5	PTD5/KBI5
	6	DDRD6			KBIE6	PTD6/KBI6
	7	DDRD7			KBIE7	PTD7/KBI7

Notes:
1. Pins are open-drain when configured as outputs. Pullup resistors must be connected when configured as outputs.
2. Register has the highest priority control on port pin.

18.3 Port A

Port A is an 8-bit special function port that shares six of its port pins with the analog-to-digital converter (ADC) module and two of its port pins with the timer interface module 1 (TIM1). See **Section 15. Analog-to-Digital Converter (ADC)** and **Section 11. Timer Interface Module (TIM)**.

PTA5–PTA0 pins can be configured for direct LED drive.

18.3.1 Port A Data Register (PTA)

The port A data register contains a data latch for each of the eight port A pins.

Address: $0000

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Reset:	Unaffected by Reset							
Alternative Function:	T1CH1	T1CH0	ATD7	ATD6	ATD5	ATD4	ATD3	ATD2
Additional Function:			LED drive	LED drive	LED drive	LED drive	LED drive	LED drive

Figure 18-2. Port A Data Register (PTA)

PTA[7:0] — Port A Data Bits

These read/write bits are software programmable. Data direction of each port A pin is under the control of the corresponding bit in data direction register A. Reset has no effect on port A data.

ATD[7:2] — ADC channels 2 to 7

ATD[7:2] are pins used for the input channels to the analog-to-digital converter module. The channel select bits, ADCH[4:0], in the ADC status and control register define which port pin will be used as an ADC input and overrides any control from the port I/O logic. See **Section 14. Analog Module**.

T1CH[1:0] — Timer 1 Channel I/O Bits

The T1CH1 and T1CH0 pins are the TIM1 input capture/output compare pins. The edge/level select bits, ELSxB:ELSxA, determine whether the PTA7/T1CH1 and PTA6/T1CH0 pins are timer channel I/O pins or general-purpose I/O pins. See **Section 11. Timer Interface Module (TIM)**.

NOTE: *Care must be taken when reading port A while applying analog voltages to ATD[7:2] pins. If the appropriate ADC channel is not enabled, excessive current drain may occur if analog voltages are applied to the PTAx/ATDx pin, while PTA is read as a digital input. Those ports not selected as analog input channels are considered digital I/O ports.*

LED drive — Direct LED drive Pins

PTA5–PTA0 pins can be configured for direct LED drive. See **18.3.3 Port A LED Control Register (LEDA)**.

18.3.2 Data Direction Register A (DDRA)

Data direction register A determines whether each port A pin is an input or an output. Writing a logic 1 to a DDRA bit enables the output buffer for the corresponding port A pin; a logic 0 disables the output buffer.

Address: $0004

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset:	0	0	0	0	0	0	0	0

Figure 18-3. Data Direction Register A (DDRA)

DDRA[7:0] — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA[7:0], configuring all port A pins as inputs.

1 = Corresponding port A pin configured as output
0 = Corresponding port A pin configured as input

NOTE: *Avoid glitches on port A pins by writing to the port A data register before changing data direction register A bits from 0 to 1.* ***Figure 18-4*** *shows the port A I/O logic.*

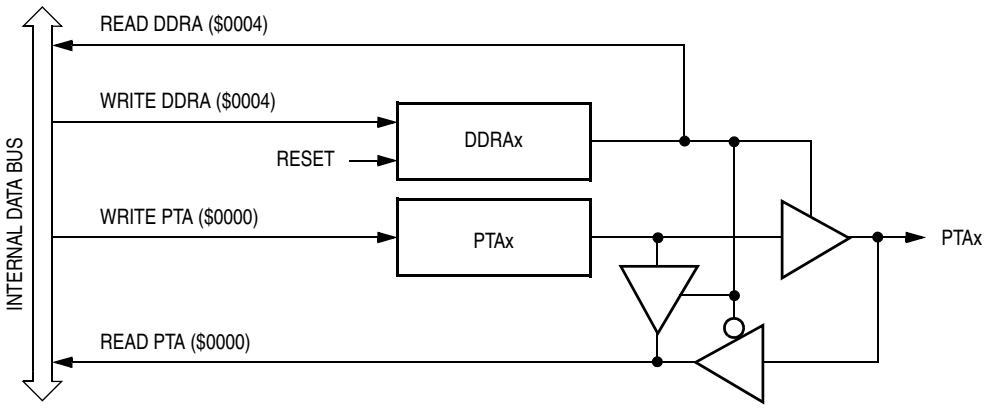


Figure 18-4. Port A I/O Circuit

When DDRAx is a logic 1, reading address $0000 reads the PTAx data latch. When DDRAx is a logic 0, reading address $0000 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit.

Table 18-2 summarizes the operation of the port A pins.

Table 18-2. Port A Pin Functions

DDRA Bit	PTA Bit	I/O Pin Mode	Accesses to DDRA	Accesses to PTA	
			Read/Write	Read	Write
0	X(1)	Input, Hi-Z(2)	DDRA[7:0]	Pin	PTA[7:0](3)
1	X	Output	DDRA[7:0]	PTA[7:0]	PTA[7:0]

Notes:
1. X = don't care.
2. Hi-Z = high impedance.
3. Writing affects data register, but does not affect input.

18.3.3 Port A LED Control Register (LEDA)

The port-A LED control register (LEDA) controls the direct LED drive capability on PTA5–PTA0 pins. Each bit is individually configurable and requires that the data direction register, DDRA, bit be configured as an output.

Address: $000C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	LEDA5	LEDA4	LEDA3	LEDA2	LEDA1	LEDA0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 18-5. Port A LED Control Register (LEDA)

LEDA[5:0] — Port A LED Drive Enable Bits

These read/write bits are software programmable to enable the direct LED drive on an output port pin.

1 = Corresponding port A pin configured for direct LED drive
0 = Corresponding port A pin configured for standard drive

18.4 Port B

Port B is a 7-bit special function port that shares four of its port pins with the multi-master IIC (MMIIC) interface module, two of its port pins with the serial communications interface (SCI) module, two of its port pins with the timer interface module 2 (TIM2), and one of its port pins with the IRQ module. See **Section 17. Multi-Master IIC Interface (MMIIC)**, **Section 16. Serial Communications Interface (SCI)**, **Section 11. Timer Interface Module (TIM)**, and **Section 19. External Interrupt (IRQ)**.

NOTE: *PTB3–PTB0 are open-drain pins when configured as outputs regardless whether the pins are used as general purpose I/O pins, MMIIC pins, or SCI pins. Therefore, when configured as general purpose output pins, MMIIC pins, or SCI pins (the TxD pin), pullup resistors must be connected to these pins.*

18.4.1 Port B Data Register (PTB)

The port B data register contains a data latch for each of the eight port B pins.

Address: $0001

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Write:								
Reset:	Unaffected by reset							
Alternative Functions:		IRQ2	T2CH1	T2CH0	RxD / SCL1	TxD / SDA1	SCL0	SDA0
					These four pins are open-drain when configured as output pins. Pullup resistors must be connected when configured as outputs.			

Figure 18-6. Port B Data Register (PTB)

PTB[6:0] — Port B Data Bits

These read/write bits are software programmable. Data direction of each port B pin is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.

SDA0, SCL0, SDA1, SCL1 — MMIIC Channels 1 and 2

SDAx and SCLx are the data and clock lines for the MMIIC module. The multi-master enable bit, MMEN, and the MMIIC channel select bit, SDASCL1, in the MMIIC control register 1 determine whether the PTB0/SDA0, PTB1/SCL0, PTB2/SDA1/TxD, and PTB3/SCL1/RxD pins are MMIIC I/O pins or general purpose I/O pins. See **Section 17. Multi-Master IIC Interface (MMIIC)**.

TxD, RxD — SCI Data I/O Pins

The TxD and RxD pins are the transmit data output and receive data input for the SCI module. The enable SCI bit, ENSCI, in the SCI control register 1 enables the PTB2/SDA1/TxD and PTB3/SCL1/RxD pins as SCI TxD and RxD pins and overrides any control from the port I/O or MMIIC logic. See **Section 16. Serial Communications Interface (SCI)**.

Table 18-3. PTB2 and PTB3 Pin Configurations

Pin	ENSCI Bit ($0013)	MMEN Bit ($0049)	SDASCL1 Bit ($0049)	Pin function
PTB2/SDA1/TxD PTB3/SCL1/RxD	0	0	X	PTB2, PTB3
	0	1	0	PTB2, PTB3
	0	1	1	SDA1, SCL1
	1	X	X	TxD, RxD

T2CH[1:0] — Timer 2 Channel I/O Bits

The T2CH1 and T2CH0 pins are the TIM2 input capture/output compare pins. The edge/level select bits, ELSxB:ELSxA, determine whether the PTB5/T2CH1 and PTB4/T2CH0 pins are timer channel I/O pins or general-purpose I/O pins.See **Section 11. Timer Interface Module (TIM)**.

IRQ2 — External Interrupt Pin 2

$\overline{\text{IRQ2}}$ pin is the second external interrupt input to the IRQ module. When PTB6/$\overline{\text{IRQ2}}$ is configured as an input by the data direction bit bit, DDRB6, the pin is both a standard port input pin and an external interrupt pin. See **Section 19. External Interrupt (IRQ)**.

18.4.2 Data Direction Register B (DDRB)

Data direction register B determines whether each port B pin is an input or an output. Writing a logic 1 to a DDRB bit enables the output buffer for the corresponding port B pin; a logic 0 disables the output buffer.

Address: $0005

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 18-7. Data Direction Register B (DDRB)

DDRB[6:0] — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB[6:0], configuring all port B pins as inputs.

1 = Corresponding port B pin configured as output
0 = Corresponding port B pin configured as input

NOTE: *Avoid glitches on port B pins by writing to the port B data register before changing data direction register B bits from 0 to 1.* ***Figure 18-8*** *shows the port B I/O logic.*

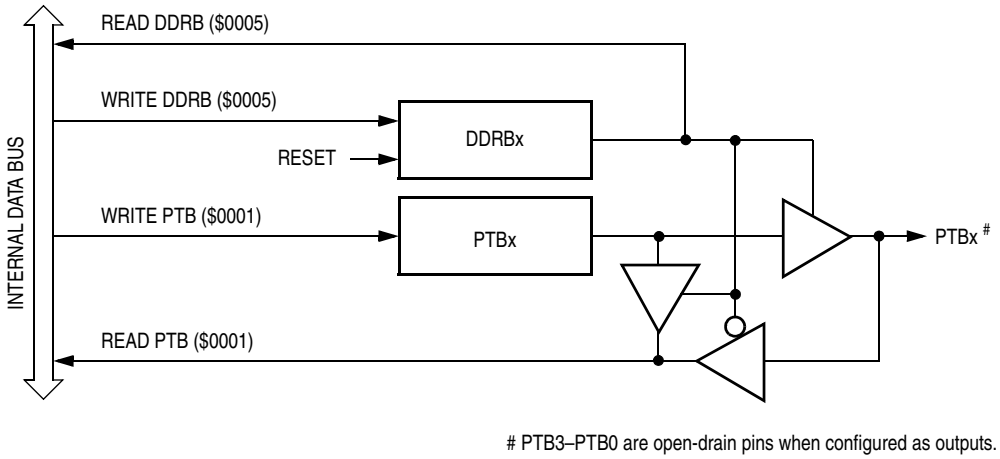


Figure 18-8. Port B I/O Circuit

When DDRBx is a logic 1, reading address $0001 reads the PTBx data latch. When DDRBx is a logic 0, reading address $0001 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit.

Table 18-4 summarizes the operation of the port B pins.

Table 18-4. Port B Pin Functions

DDRB Bit	PTB Bit	I/O Pin Mode	Accesses to DDRB	Accesses to PTB	
			Read/Write	Read	Write
0	X(1)	Input, Hi-Z(2)	DDRB[6:0]	Pin	PTB[6:0](3)
1	X	Output	DDRB[6:0]	PTB[6:0]	PTB[6:0]

Notes:
1. X = don't care.
2. Hi-Z = high impedance.
3. Writing affects data register, but does not affect input.

18.5 Port C

Port C is an 8-bit special function port that shares three of its port pins with the pulse width modulator module, five of its port pins with the analog-to-digital converter module, and one of its pins with the analog module. See **Section 13. Pulse Width Modulator (PWM)**, **Section 15. Analog-to-Digital Converter (ADC)**, and **Section 14. Analog Module**.

PTC7–PTC3 pins can be configured for direct LED drive.

18.5.1 Port C Data Register (PTC)

The port C data register contains a data latch for each of the six port C pins.

NOTE: *Bit 7 and bit 6 of PTC are not available in a 42-pin shrink dual in-line package.*

Address: $0002

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTC7	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
Reset:	Unaffected by reset							
Alternative Function:	ATD12	ATD11	ATD10	ATD9	ATD8	PWM2	PWM1	PWM0
								CD
Additional Function:	LED drive	LED drive	LED drive	LED drive	LED drive			

Figure 18-9. Port C Data Register (PTC)

PTC[7:0] — Port C Data Bits

These read/write bits are software programmable. Data direction of each port C pin is under the control of the corresponding bit in data direction register C. Reset has no effect on port C data.

PWM[2:0] — PWM Channels 0 to 2

PWM[2:0] are pins used for the output channels from the pulse width modulator (PWM) module. The PWM enables bit, PCH[2:0], in the PWM control register define which port pin will be used as a PWM output and overrides any control from the port I/O logic. The PWM0 function on PCT0/PWM0/CD pin can be overrided by the CD output function. See **Section 13. Pulse Width Modulator (PWM)**.

ATD[12:8] — ADC channels 8 to 12

ATD[12:8] are pins used for the input channels to the analog-to-digital converter module. The channel select bits, ADCH[4:0], in the ADC status and control register define which port pin will be used as an ADC input and overrides any control from the port I/O logic. See **Section 15. Analog-to-Digital Converter (ADC)**.

NOTE: *Care must be taken when reading port C while applying analog voltages to ATD[12:8] pins. If the appropriate ADC channel is not enabled, excessive current drain may occur if analog voltages are applied to the PTCx/ATDx pin, while PTC is read as a digital input. Those ports not selected as analog input channels are considered digital I/O ports.*

CD — Current Detect Output Pin

The CD pin is used for the current detect output from the analog module. The pin reflects the status of the current detect interrupt flag. The current detect output enable bit, CDOEN, in the configuration register 2 enables the PTC0/PWM0/CD pin as the CD output pin and overrides any control from the port I/O or PWM logic. See **Section 14. Analog Module**.

Table 18-5. PTC0 Pin Configuration

Pin	CDOEN Bit ($001D)	PCH0 Bit ($0051)	Pin function
PTC0/PWM0/CD	0	0	PTC0
	0	1	PWM0
	1	X	CD

LED drive — Direct LED drive Pins

PTC7–PTC3 pins can be configured for direct LED drive. See **18.5.3 Port C LED Control Register (LEDC)**.

18.5.2 Data Direction Register C (DDRC)

Data direction register C determines whether each port C pin is an input or an output. Writing a logic 1 to a DDRC bit enables the output buffer for the corresponding port C pin; a logic 0 disables the output buffer.

Address: $0006

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
Reset:	0	0	0	0	0	0	0	0

Figure 18-10. Data Direction Register B (DDRB)

DDRC[7:0] — Data Direction Register C Bits

These read/write bits control port C data direction. Reset clears DDRC[7:0], configuring all port C pins as inputs.

1 = Corresponding port C pin configured as output
0 = Corresponding port C pin configured as input

NOTE: *Avoid glitches on port C pins by writing to the port C data register before changing data direction register C bits from 0 to 1. **Figure 18-11** shows the port C I/O logic.*

NOTE: *For those devices packaged in a 42-pin shrink dual in-line package, PTC6 and PTC7 are not connected. DDRC6 and DDRC7 should be set to a 1 to configure PTC6 and PTC7 as outputs.*

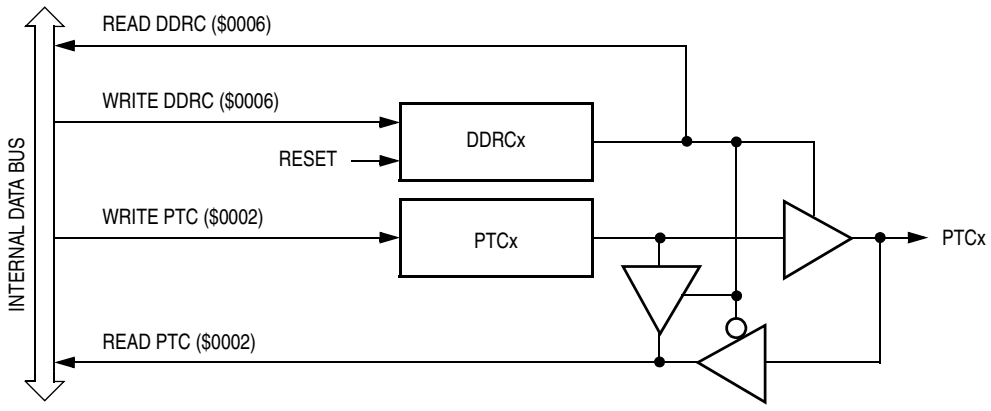


Figure 18-11. Port C I/O Circuit

When DDRCx is a logic 1, reading address $0002 reads the PTCx data latch. When DDRCx is a logic 0, reading address $0002 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit.

Table 18-6 summarizes the operation of the port C pins.

Table 18-6. Port C Pin Functions

DDRC Bit	PTC Bit	I/O Pin Mode	Accesses to DDRC	Accesses to PTC	
			Read/Write	Read	Write
0	X(1)	Input, Hi-Z(2)	DDRC[7:0]	Pin	PTC[7:0](3)
1	X	Output	DDRC[7:0]	PTC[7:0]	PTC[7:0]

Notes:
1. X = don't care.
2. Hi-Z = high impedance.
3. Writing affects data register, but does not affect input.

18.5.3 Port C LED Control Register (LEDC)

The port-C LED control register (LEDC) controls the direct LED drive capability on PTC7–PTC3 pins. Each bit is individually configurable and requires that the data direction register, DDRD, bit be configured as an output.

Address: $000D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	LEDC7	LEDC6	LEDC5	LEDC4	LEDC3	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 18-12. Port A LED Control Register (LEDA)

LEDC[7:3] — Port C LED Drive Enable Bits

These read/write bits are software programmable to enable the direct LED drive on an output port pin.

1 = Corresponding port C pin configured for direct LED drive
0 = Corresponding port C pin configured for standard drive

18.6 Port D

Port D is an 8-bit special function port that shares all of its pins with the keyboard interrupt module. See **Section 20. Keyboard Interrupt Module (KBI)**.

18.6.1 Port D Data Register (PTD)

The port D data register contains a data latch for each of the eight port D pins.

Address: $0003

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
Reset:	Unaffected by reset							
Alternative Function:	KBI7	KBI6	KBI5	KBI4	KBI3	KBI2	KBI1	KBI0

Figure 18-13. Port D Data Register (PTD)

PTD[7:0] — Port D Data Bits

These read/write bits are software programmable. Data direction of each port D pin is under the control of the corresponding bit in data direction register D. Reset has no effect on port D data.

KBI[7:0] — Keyboard Interrupt Pins

The keyboard interrupt enable bits, KBIE[7:0], in the keyboard interrupt enable register (KBIER), enable the port D pins as external interrupt pins. See **Section 20. Keyboard Interrupt Module (KBI)**.

18.6.2 Data Direction Register D (DDRD)

Data direction register D determines whether each port D pin is an input or an output. Writing a logic 1 to a DDRD bit enables the output buffer for the corresponding port D pin; a logic 0 disables the output buffer.

Address: $0007

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Reset:	0	0	0	0	0	0	0	0

Figure 18-14. Data Direction Register D (DDRD)

DDRD[7:0] — Data Direction Register D Bits

These read/write bits control port D data direction. Reset clears DDRD[7:0], configuring all port D pins as inputs.

1 = Corresponding port D pin configured as output
0 = Corresponding port D pin configured as input

NOTE: *Avoid glitches on port D pins by writing to the port D data register before changing data direction register D bits from 0 to 1.* ***Figure 18-15*** *shows the port D I/O logic.*

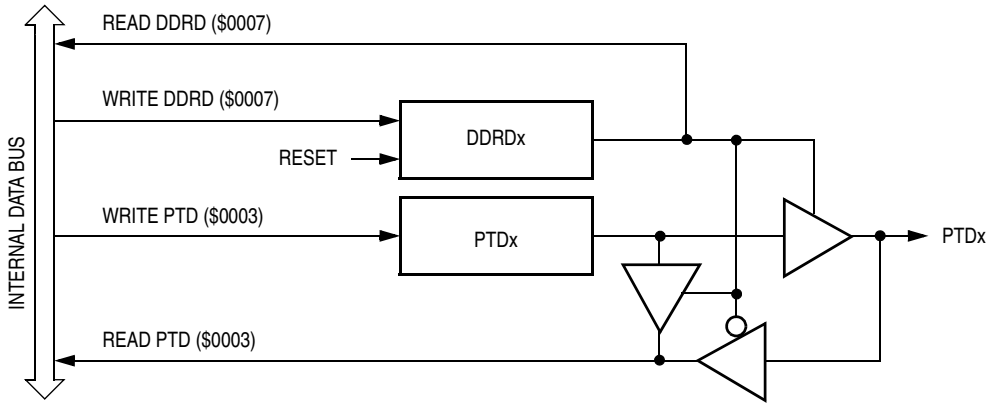


Figure 18-15. Port D I/O Circuit

When DDRDx is a logic 1, reading address $0003 reads the PTDx data latch. When DDRDx is a logic 0, reading address $0003 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit.

Table 18-7 summarizes the operation of the port D pins.

Table 18-7. Port D Pin Functions

DDRD Bit	PTD Bit	I/O Pin Mode	Accesses to DDRD	Accesses to PTD	
			Read/Write	Read	Write
0	X(1)	Input, Hi-Z(2)	DDRD[7:0]	Pin	PTD[7:0](3)
1	X	Output	DDRD[7:0]	PTD[7:0]	PTD[7:0]

Notes:
1. X = don't care.
2. Hi-Z = high impedance.
3. Writing affects data register, but does not affect the input.

Section 19. External Interrupt (IRQ)

19.1 Contents

19.2 Introduction 335
19.3 Features 335
19.4 Functional Description 336
19.5 $\overline{IRQ1}$ and $\overline{IRQ2}$ Pins 338
19.6 IRQ Module During Break Interrupts 339
19.7 IRQ Registers 340
19.7.1 IRQ1 Status and Control Register 340
19.7.2 IRQ2 Status and Control Register 341

19.2 Introduction

The external interrupt (IRQ) module provides two maskable interrupt inputs: $\overline{IRQ1}$ and $\overline{IRQ2}$.

19.3 Features

Features of the IRQ module include:

- A dedicated external interrupt pin, $\overline{IRQ1}$
- An external interrupt pin shared with a port pin, $\overline{IRQ2}$/PTB6
- Separate IRQ interrupt control bits for $\overline{IRQ1}$ and $\overline{IRQ2}$
- Hysteresis buffers
- Programmable edge-only or edge and level interrupt sensitivity
- Automatic interrupt acknowledge
- Internal pullup resistor, with disable option on $\overline{IRQ2}$

NOTE: *References to either IRQ1 or IRQ2 may be made in the following text by omitting the IRQ number. For example, IRQF may refer generically to IRQ1F and IRQ2F, and IMASK may refer to IMASK1 and IMASK2.*

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001C	IRQ2 Status and Control Register (INTSCR2)	Read:	0	PTBPUE6	0	0	IRQ2F	0	IMASK2	MODE2
		Write:						ACK2		
		Reset:	0	0	0	0	0	0	0	0
$001E	IRQ1 Status and Control Register (INTSCR1)	Read:	0	0	0	0	IRQ1F	0	IMASK1	MODE1
		Write:						ACK1		
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 19-1. External Interrupt I/O Register Summary

19.4 Functional Description

A logic 0 applied to the external interrupt pin can latch a CPU interrupt request. **Figure 19-2** and **Figure 19-3** shows the structure of the IRQ module.

Interrupt signals on the $\overline{\text{IRQ}}$ pin are latched into the IRQ latch. An interrupt latch remains set until one of the following actions occurs:

- Vector fetch — A vector fetch automatically generates an interrupt acknowledge signal that clears the latch that caused the vector fetch.
- Software clear — Software can clear an interrupt latch by writing to the appropriate acknowledge bit in the interrupt status and control register (INTSCR). Writing a logic 1 to the ACK bit clears the IRQ latch.
- Reset — A reset automatically clears the interrupt latch.

The external interrupt pin is falling-edge-triggered and is software-configurable to be either falling-edge or falling-edge and low-level-triggered. The MODE bit in the INTSCR controls the triggering sensitivity of the $\overline{\text{IRQ}}$ pin.

When an interrupt pin is edge-triggered only, the interrupt remains set until a vector fetch, software clear, or reset occurs.

When an interrupt pin is both falling-edge and low-level-triggered, the interrupt remains set until both of the following occur:

- Vector fetch or software clear
- Return of the interrupt pin to logic 1

The vector fetch or software clear may occur before or after the interrupt pin returns to logic 1. As long as the pin is low, the interrupt request remains pending. A reset will clear the latch and the MODE1 control bit, thereby clearing the interrupt even if the pin stays low.

When set, the IMASK bit in the INTSCR mask all external interrupt requests. A latched interrupt request is not presented to the interrupt priority logic unless the IMASK bit is clear.

NOTE: *The interrupt mask (I) in the condition code register (CCR) masks all interrupt requests, including external interrupt requests.*

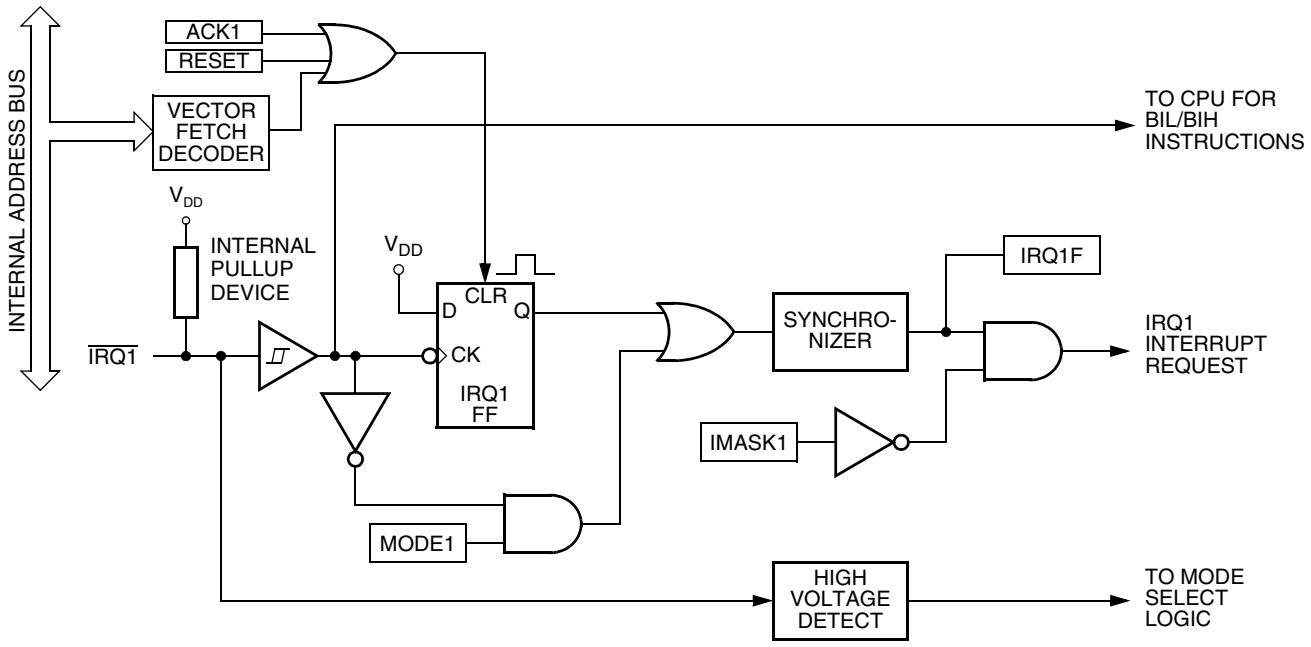


Figure 19-2. IRQ1 Block Diagram

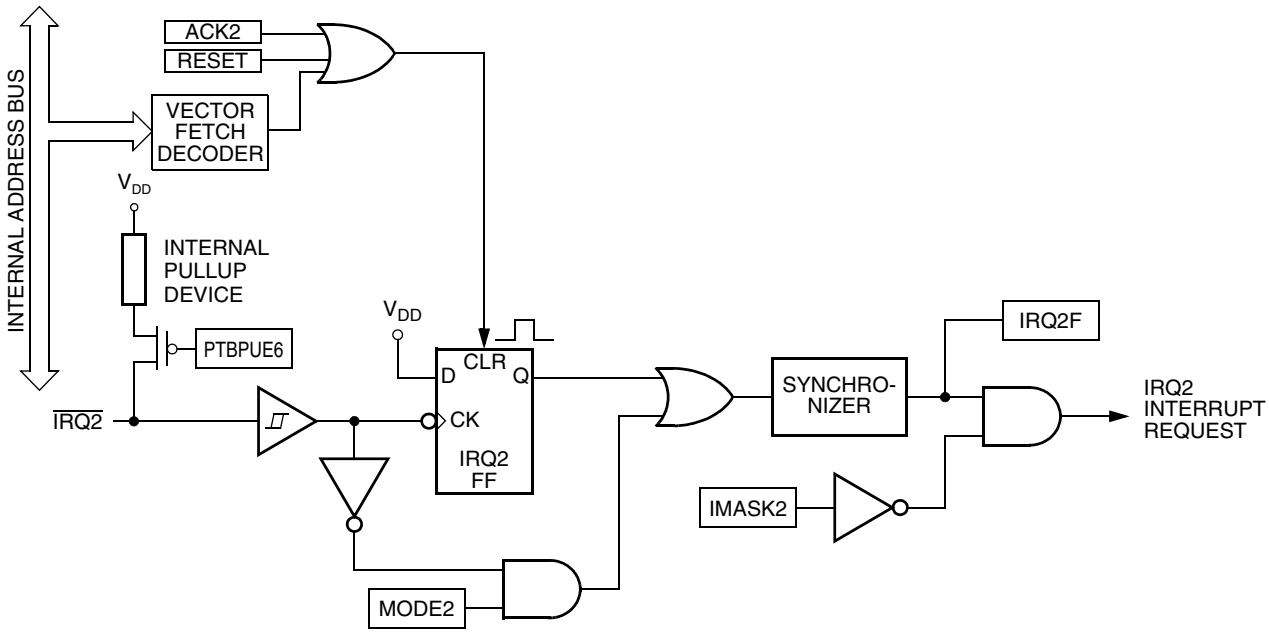


Figure 19-3. IRQ2 Block Diagram

19.5 $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$ Pins

A logic 0 on the $\overline{\text{IRQ}}$ pin can latch an interrupt request into the IRQ latch. A vector fetch, software clear, or reset clears the IRQ latch.

If the MODE bit is set, the $\overline{\text{IRQ}}$ pin is both falling-edge-sensitive and low-level-sensitive. With MODE set, both of the following actions must occur to clear IRQ:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the latch. Software may generate the interrupt acknowledge signal by writing a logic 1 to the ACK bit in the interrupt status and control register (INTSCR). The ACK bit is useful in applications that poll the $\overline{\text{IRQ}}$ pin and require software to clear the IRQ latch. Writing to the ACK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACK does not affect subsequent transitions on the $\overline{\text{IRQ}}$ pin. A falling edge that occurs after writing to the ACK bit another interrupt request. If the IRQ mask bit, IMASK, is clear, the CPU loads the program counter with the vector address at location defined in **Table 2-1 . Vector Addresses**.

- Return of the $\overline{\text{IRQ}}$ pin to logic 1 — As long as the $\overline{\text{IRQ}}$ pin is at logic 0, IRQ remains active.

The vector fetch or software clear and the return of the $\overline{\text{IRQ}}$ pin to logic 1 may occur in any order. The interrupt request remains pending as long as the $\overline{\text{IRQ}}$ pin is at logic 0. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

If the MODE bit is clear, the $\overline{\text{IRQ}}$ pin is falling-edge-sensitive only. With MODE clear, a vector fetch or software clear immediately clears the IRQ latch.

The IRQF bit in the INTSCR register can be used to check for pending interrupts. The IRQF bit is not affected by the IMASK bit, which makes it useful in applications where polling is preferred.

Use the BIH or BIL instruction to read the logic level on the $\overline{\text{IRQ1}}$ pin.

NOTE: *The BIH and BIL instructions do not read the logic level on the $\overline{\text{IRQ2}}$ pin.*

NOTE: *When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine.*

The $\overline{\text{IRQ1}}$ pin has a permanent internal pullup device connected, while the $\overline{\text{IRQ2}}$ pin has an optional pullup device that can be enabled or disabled by the PTBPUE6 bit in the INTSCR2 register.

19.6 IRQ Module During Break Interrupts

The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear the latch during the break state. (See **Section 23. Break Module (BRK)**.)

To allow software to clear the IRQ latch during a break interrupt, write a logic 1 to the BCFE bit. If a latch is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect CPU interrupt flags during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), writing to the ACK bit in the IRQ status and control register during the break state has no effect on the IRQ interrupt flags.

19.7 IRQ Registers

Each IRQ is controlled and monitored by an status and control register.

- IRQ1 Status and Control Register — $001E
- IRQ2 Status and Control Register — $001C

19.7.1 IRQ1 Status and Control Register

The IRQ1 status and control register (INTSCR1) controls and monitors operation of IRQ1. The INTSCR1 has the following functions:

- Shows the state of the IRQ1 flag
- Clears the IRQ1 latch
- Masks IRQ1 interrupt request
- Controls triggering sensitivity of the $\overline{\text{IRQ1}}$ interrupt pin

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	IRQ1F	0	IMASK1	MODE1
Write:						ACK1		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 19-4. IRQ1 Status and Control Register (INTSCR1)

IRQ1F — IRQ1 Flag Bit

This read-only status bit is high when the IRQ1 interrupt is pending.
1 = $\overline{\text{IRQ1}}$ interrupt pending
0 = $\overline{\text{IRQ1}}$ interrupt not pending

ACK1 — IRQ1 Interrupt Request Acknowledge Bit

Writing a logic 1 to this write-only bit clears the IRQ1 latch. ACK1 always reads as logic 0. Reset clears ACK1.

IMASK1 — IRQ1 Interrupt Mask Bit

Writing a logic 1 to this read/write bit disables IRQ1 interrupt requests. Reset clears IMASK1.

1 = $\overline{\text{IRQ1}}$ interrupt requests disabled
0 = $\overline{\text{IRQ1}}$ interrupt requests enabled

MODE1 — IRQ1 Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the $\overline{\text{IRQ1}}$ pin. Reset clears MODE1.

1 = $\overline{\text{IRQ1}}$ interrupt requests on falling edges and low levels
0 = $\overline{\text{IRQ1}}$ interrupt requests on falling edges only

19.7.2 IRQ2 Status and Control Register

The IRQ2 status and control register (INTSCR2) controls and monitors operation of IRQ2. The INTSCR2 has the following functions:

- Enables/disables the internal pullup device on $\overline{\text{IRQ2}}$ pin
- Shows the state of the IRQ2 flag
- Clears the IRQ2 latch
- Masks IRQ2 interrupt request
- Controls triggering sensitivity of the $\overline{\text{IRQ2}}$ interrupt pin

Address: $001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PTBPUE6	0	0	IRQ2F	0	IMASK2	MODE2
Write:						ACK2		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 19-5. IRQ2 Status and Control Register (INTSCR2)

PTBPUE6 — $\overline{\text{IRQ2}}$ Pin Pullup Enable Bit.

Setting this bit to logic 1 disables the pullup on PTB6/$\overline{\text{IRQ2}}$ pin. Reset clears this bit.

1 = $\overline{\text{IRQ2}}$ pin internal pull-up is disabled
0 = $\overline{\text{IRQ2}}$ pin internal pull-up is enabled

IRQ2F — IRQ2 Flag Bit

This read-only status bit is high when the IRQ2 interrupt is pending.

1 = $\overline{\text{IRQ2}}$ interrupt pending
0 = $\overline{\text{IRQ2}}$ interrupt not pending

ACK2 — IRQ2 Interrupt Request Acknowledge Bit

Writing a logic 1 to this write-only bit clears the IRQ2 latch. ACK2 always reads as logic 0. Reset clears ACK2.

IMASK2 — IRQ2 Interrupt Mask Bit

Writing a logic 1 to this read/write bit disables IRQ2 interrupt requests. Reset clears IMASK2.

1 = IRQ2 interrupt requests disabled
0 = IRQ2 interrupt requests enabled

MODE2 — IRQ2 Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the $\overline{\text{IRQ2}}$ pin. Reset clears MODE2.

1 = $\overline{\text{IRQ2}}$ interrupt requests on falling edges and low levels
0 = $\overline{\text{IRQ2}}$ interrupt requests on falling edges only

Section 20. Keyboard Interrupt Module (KBI)

20.1 Contents

20.2 Introduction 343
20.3 Features 344
20.4 I/O Pins 344
20.5 Functional Description 345
20.5.1 Keyboard Initialization 347
20.6 Keyboard Interrupt Registers 347
20.6.1 Keyboard Status and Control Register 348
20.6.2 Keyboard Interrupt Enable Register 349
20.7 Low-Power Modes 349
20.7.1 Wait Mode 349
20.7.2 Stop Mode 349
20.10 Keyboard Module During Break Interrupts 350

20.2 Introduction

The keyboard interrupt module (KBI) provides eight independently maskable external interrupts which are accessible via PTD0–PTD7. When a port pin is enabled for keyboard interrupt function, an internal 30 kΩ pullup device is also enabled on the pin.

20.3 Features

Features of the keyboard interrupt module include the following:

- Eight keyboard interrupt pins with pullup devices
- Separate keyboard interrupt enable bits and one keyboard interrupt mask
- Programmable edge-only or edge- and level- interrupt sensitivity
- Exit from low-lower modes

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001A	Keyboard Status and Control Register (KBSCR)	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$001B	Keyboard Interrupt Enable Register (KBIER)	Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 20-1. KBI I/O Register Summary

20.4 I/O Pins

The eight keyboard interrupt pins are shared with standard port I/O pins. The full name of the KBI pins are listed in **Table 20-1**. The generic pin name appear in the text that follows.

Table 20-1. Pin Name Conventions

KBI Generic Pin Name	Full MCU Pin Name	Pin Selected for KBI Function by KBIEx Bit in KBIER
KBI0–KBI7	PTD0/KBI0–PTD7/KBI7	KBIE0–KBIE7

20.5 Functional Description

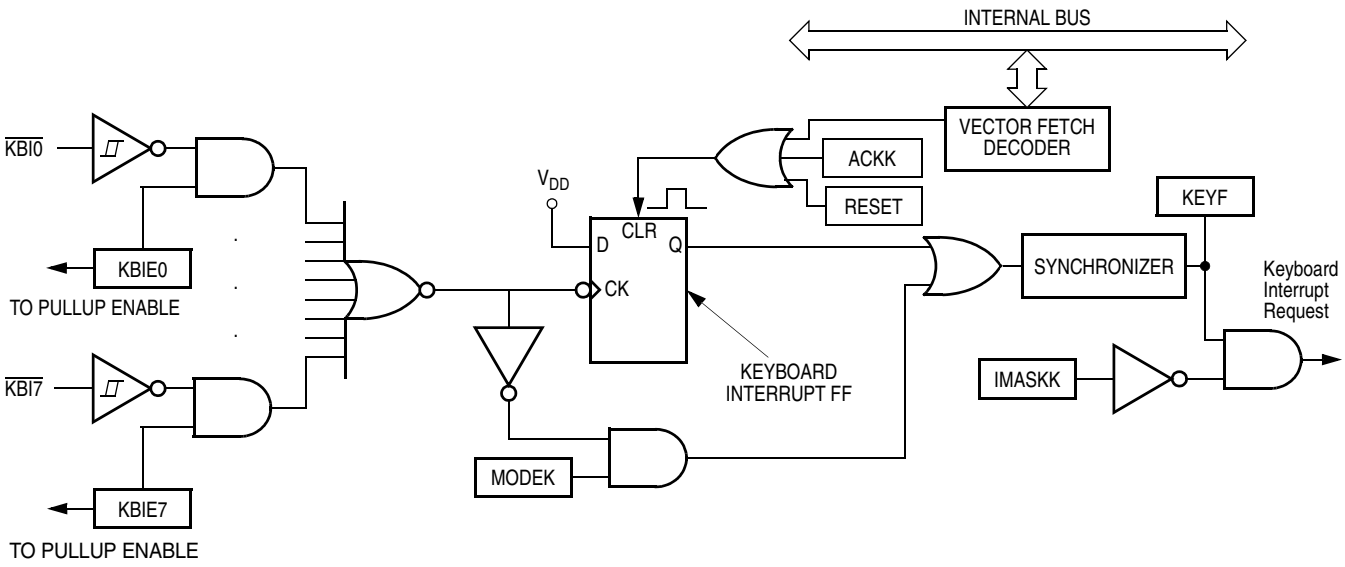


Figure 20-2. Keyboard Interrupt Block Diagram

Writing to the KBIE7–KBIE0 bits in the keyboard interrupt enable register independently enables or disables each port D pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin in port D also enables its internal pull-up device. A logic 0 applied to an enabled keyboard interrupt pin latches a keyboard interrupt request.

A keyboard interrupt is latched when one or more keyboard pins goes low after all were high. The MODEK bit in the keyboard status and control register controls the triggering mode of the keyboard interrupt.

- If the keyboard interrupt is edge-sensitive only, a falling edge on a keyboard pin does not latch an interrupt request if another keyboard pin is already low. To prevent losing an interrupt request on one pin because another pin is still low, software can disable the latter pin while it is low.
- If the keyboard interrupt is falling edge- and low level-sensitive, an interrupt request is present as long as any keyboard pin is low.

If the MODEK bit is set, the keyboard interrupt pins are both falling edge- and low level-sensitive, and both of the following actions must occur to clear a keyboard interrupt request:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the interrupt request. Software may generate the interrupt acknowledge signal by writing a logic 1 to the ACKK bit in the keyboard status and control register KBSCR. The ACKK bit is useful in applications that poll the keyboard interrupt pins and require software to clear the keyboard interrupt request. Writing to the ACKK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACKK does not affect subsequent transitions on the keyboard interrupt pins. A falling edge that occurs after writing to the ACKK bit latches another interrupt request. If the keyboard interrupt mask bit, IMASKK, is clear, the CPU loads the program counter with the vector address at locations $FFE0 and $FFE1.
- Return of all enabled keyboard interrupt pins to logic 1 — As long as any enabled keyboard interrupt pin is at logic 0, the keyboard interrupt remains set.

The vector fetch or software clear and the return of all enabled keyboard interrupt pins to logic 1 may occur in any order.

If the MODEK bit is clear, the keyboard interrupt pin is falling-edge-sensitive only. With MODEK clear, a vector fetch or software clear immediately clears the keyboard interrupt request.

Reset clears the keyboard interrupt request and the MODEK bit, clearing the interrupt request even if a keyboard interrupt pin stays at logic 0.

The keyboard flag bit (KEYF) in the keyboard status and control register can be used to see if a pending interrupt exists. The KEYF bit is not affected by the keyboard interrupt mask bit (IMASKK) which makes it useful in applications where polling is preferred.

To determine the logic level on a keyboard interrupt pin, use the data direction register to configure the pin as an input and read the data register.

NOTE: *Setting a keyboard interrupt enable bit (KBIEx) forces the corresponding keyboard interrupt pin to be an input, overriding the data direction register. However, the data direction register bit must be a logic 0 for software to read the pin.*

20.5.1 Keyboard Initialization

When a keyboard interrupt pin is enabled, it takes time for the internal pull-up to reach a logic 1. Therefore a false interrupt can occur as soon as the pin is enabled.

To prevent a false interrupt on keyboard initialization:

1. Mask keyboard interrupts by setting the IMASKK bit in the keyboard status and control register.
2. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.
3. Write to the ACKK bit in the keyboard status and control register to clear any false interrupts.
4. Clear the IMASKK bit.

An interrupt signal on an edge-triggered pin can be acknowledged immediately after enabling the pin. An interrupt signal on an edge- and level-triggered interrupt pin must be acknowledged after a delay that depends on the external load.

Another way to avoid a false interrupt:

1. Configure the keyboard pins as outputs by setting the appropriate DDR bits in data direction register.
2. Write logic 1s to the appropriate data register bits.
3. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.

20.6 Keyboard Interrupt Registers

Two registers control the operation of the keyboard interrupt module:

- Keyboard Status and Control Register — $001A
- Keyboard Interrupt Enable Register — $001B

20.6.1 Keyboard Status and Control Register

- Flags keyboard interrupt requests
- Acknowledges keyboard interrupt requests
- Masks keyboard interrupt requests
- Controls keyboard interrupt triggering sensitivity

Address: $001A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
Write:						ACKK		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 20-3. Keyboard Status and Control Register (KBSCR)

KEYF — Keyboard Flag Bit

This read-only bit is set when a keyboard interrupt is pending. Reset clears the KEYF bit.

1 = Keyboard interrupt pending
0 = No keyboard interrupt pending

ACKK — Keyboard Acknowledge Bit

Writing a logic 1 to this write-only bit clears the keyboard interrupt request. ACKK always reads as logic 0. Reset clears ACKK.

IMASKK — Keyboard Interrupt Mask Bit

Writing a logic 1 to this read/write bit prevents the output of the keyboard interrupt mask from generating interrupt requests. Reset clears the IMASKK bit.

1 = Keyboard interrupt requests masked
0 = Keyboard interrupt requests not masked

MODEK — Keyboard Triggering Sensitivity Bit

This read/write bit controls the triggering sensitivity of the keyboard interrupt pins. Reset clears MODEK.

1 = Keyboard interrupt requests on falling edges and low levels
0 = Keyboard interrupt requests on falling edges only

20.6.2 Keyboard Interrupt Enable Register

The port-D keyboard interrupt enable register enables or disables each port-D pin to operate as a keyboard interrupt pin.

Address: $001B

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Reset:	0	0	0	0	0	0	0	0

Figure 20-4. Keyboard Interrupt Enable Register (KBIER)

KBIE7–KBIE0 — Keyboard Interrupt Enable Bits

Each of these read/write bits enables the corresponding keyboard interrupt pin to latch interrupt requests. Reset clears the keyboard interrupt enable register.

1 = KBIx pin enabled as keyboard interrupt pin
0 = KBIx pin not enabled as keyboard interrupt pin

20.7 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

20.7.1 Wait Mode

The keyboard interrupt module remains active in wait mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of wait mode.

20.7.2 Stop Mode

The keyboard interrupt module remains active in stop mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of stop mode.

20.8 Keyboard Module During Break Interrupts

The system integration module (SIM) controls whether the keyboard interrupt latch can be cleared during the break state. The BCFE bit in the SIM break flag control register (BFCR) enables software to clear status bits during the break state.

To allow software to clear the keyboard interrupt latch during a break interrupt, write a logic 1 to the BCFE bit. If a latch is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect the latch during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), writing to the keyboard acknowledge bit (ACKK) in the keyboard status and control register during the break state has no effect.

Section 21. Computer Operating Properly (COP)

21.1 Contents

21.2 Introduction 351
21.3 Functional Description 352
21.4 I/O Signals 353
21.4.1 ICLK 353
21.4.2 STOP Instruction 353
21.4.3 COPCTL Write 353
21.4.4 Power-On Reset 353
21.4.5 Internal Reset 354
21.4.6 Reset Vector Fetch 354
21.4.7 COPD (COP Disable) 354
21.4.8 COPRS (COP Rate Select) 354
21.5 COP Control Register 355
21.6 Interrupts 355
21.7 Monitor Mode 355
21.8 Low-Power Modes 355
21.8.1 Wait Mode 356
21.8.2 Stop Mode 356
21.9 COP Module During Break Mode 356

21.2 Introduction

The computer operating properly (COP) module contains a free-running counter that generates a reset if allowed to overflow. The COP module helps software recover from runaway code. Prevent a COP reset by clearing the COP counter periodically. The COP module can be disabled through the COPD bit in the configuration register 1 (CONFIG1).

21.3 Functional Description

Figure 21-1 shows the structure of the COP module.

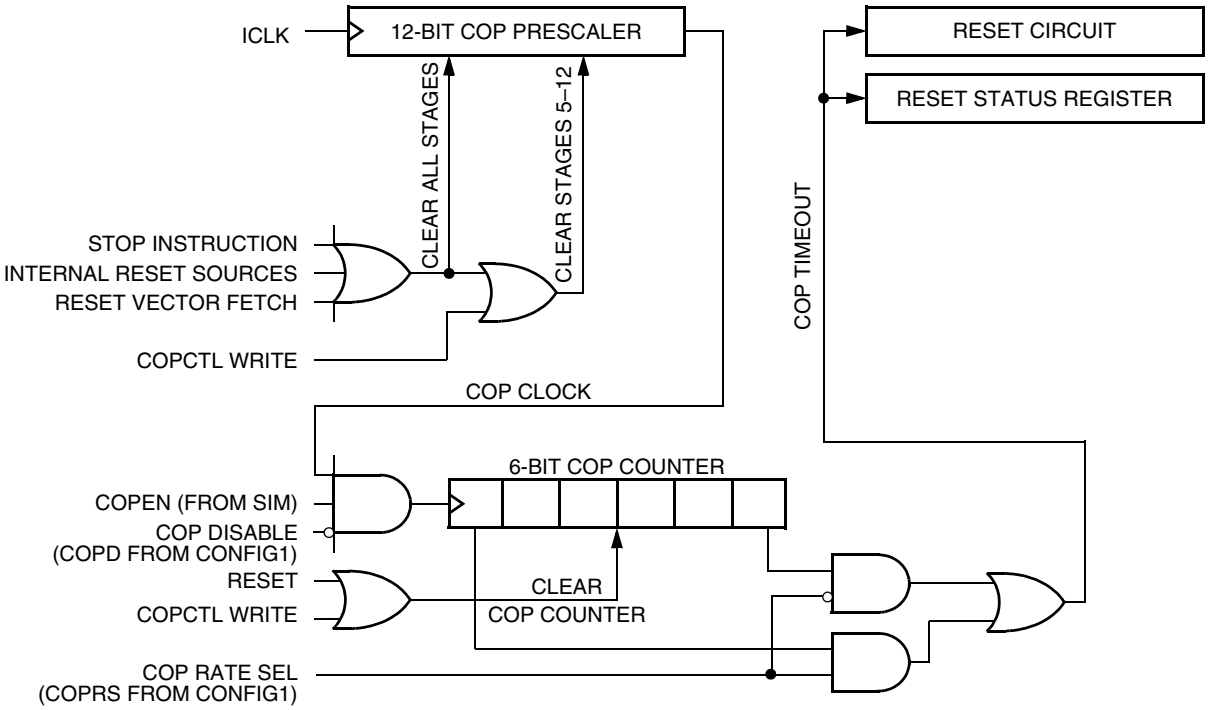


Figure 21-1. COP Block Diagram

The COP counter is a free-running 6-bit counter preceded by a 12-bit prescaler counter. If not cleared by software, the COP counter overflows and generates an asynchronous reset after $2^{18} - 2^4$ or $2^{13} - 2^4$ ICLK cycles, depending on the state of the COP rate select bit, COPRS, in the CONFIG1 register. With a $2^{13} - 2^4$ ICLK cycle overflow option, a 24-kHz ICLK gives a COP timeout period of 341 ms. Writing any value to location $FFFF before an overflow occurs prevents a COP reset by clearing the COP counter and stages 12 through 5 of the prescaler.

NOTE: *Service the COP immediately after reset and before entering or after exiting STOP Mode to guarantee the maximum time before the first COP counter overflow.*

A COP reset pulls the $\overline{\text{RST}}$ pin low for 32 ICLK cycles and sets the COP bit in the SIM reset status register (SRSR).

In monitor mode, the COP is disabled if the $\overline{\text{RST}}$ pin or the $\overline{\text{IRQ1}}$ is held at V_{TST}. During the break state, V_{TST} on the $\overline{\text{RST}}$ pin disables the COP.

NOTE: *Place COP clearing instructions in the main program and not in an interrupt subroutine. Such an interrupt subroutine could keep the COP from generating a reset even while the main program is not working properly.*

21.4 I/O Signals

The following paragraphs describe the signals shown in **Figure 21-1**.

21.4.1 ICLK

ICLK is the internal oscillator output signal. ICLK frequency is approximately equal to 24kHz. See **Section 24. Electrical Specifications** for ICLK parameters.

21.4.2 STOP Instruction

The STOP instruction clears the COP prescaler.

21.4.3 COPCTL Write

Writing any value to the COP control register (COPCTL) (see **21.5 COP Control Register**) clears the COP counter and clears bits 12 through 5 of the prescaler. Reading the COP control register returns the low byte of the reset vector.

21.4.4 Power-On Reset

The power-on reset (POR) circuit clears the COP prescaler 4096 ICLK cycles after power-up.

21.4.5 Internal Reset

An internal reset clears the COP prescaler and the COP counter.

21.4.6 Reset Vector Fetch

A reset vector fetch occurs when the vector address appears on the data bus. A reset vector fetch clears the COP prescaler.

21.4.7 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the CONFIG1 register. (See **Figure 21-2** and **Section 5. Configuration and Mask Option Registers (CONFIG & MOR)**.)

21.4.8 COPRS (COP Rate Select)

The COPRS signal reflects the state of the COP rate select bit (COPRS) in the CONFIG1 register.

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3	SSREC	STOP	COPD
Reset:	0	0	0	0	0*	0	0	0

* Reset by POR only.

Figure 21-2. Configuration Register 1 (CONFIG1)

COPRS — COP Rate Select

COPRS selects the COP time-out period. Reset clears COPRS.

1 = COP time out period = $2^{13} – 2^4$ ICLK cycles
0 = COP time out period = $2^{18} – 2^4$ ICLK cycles

COPD — COP Disable Bit

COPD disables the COP module.

1 = COP module disabled
0 = COP module enabled

21.5 COP Control Register

The COP control register is located at address $FFFF and overlaps the reset vector. Writing any value to $FFFF clears the COP counter and starts a new timeout period. Reading location $FFFF returns the low byte of the reset vector.

Address: $FFFF

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Low byte of reset vector							
Write:	Clear COP counter							
Reset:	Unaffected by reset							

Figure 21-3. COP Control Register (COPCTL)

21.6 Interrupts

The COP does not generate CPU interrupt requests.

21.7 Monitor Mode

When monitor mode is entered with V_{TST} on the $\overline{IRQ1}$ pin, the COP is disabled as long as V_{TST} remains on the $\overline{IRQ1}$ pin or the $\overline{RST}$ pin. When monitor mode is entered by having blank reset vectors and not having V_{TST} on the $\overline{IRQ1}$ pin, the COP is automatically disabled until a POR occurs.

21.8 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

21.8.1 Wait Mode

The COP remains active during wait mode. To prevent a COP reset during wait mode, periodically clear the COP counter in a CPU interrupt routine.

21.8.2 Stop Mode

Stop mode turns off the ICLK input to the COP and clears the COP prescaler. Service the COP immediately before entering or after exiting stop mode to ensure a full COP timeout period after entering or exiting stop mode.

To prevent inadvertently turning off the COP with a STOP instruction, a configuration option is available that disables the STOP instruction. When the STOP bit in the configuration register has the STOP instruction is disabled, execution of a STOP instruction results in an illegal opcode reset.

21.9 COP Module During Break Mode

The COP is disabled during a break interrupt when V_{TST} is present on the $\overline{RST}$ pin.

Section 22. Low-Voltage Inhibit (LVI)

22.1 Contents

22.2 Introduction 357
22.3 Features 357
22.4 Functional Description 358
22.4.1 Polled LVI Operation 359
22.4.2 Forced Reset Operation 360
22.4.3 Voltage Hysteresis Protection 360
22.4.4 LVI Trip Selection 360
22.5 LVI Status Register 361
22.6 LVI Interrupts 361
22.7 Low-Power Modes 361
22.7.1 Wait Mode 362
22.7.2 Stop Mode 362

22.2 Introduction

This section describes the low-voltage inhibit (LVI) module, which monitors the voltage on the V_{DD} pin and can force a reset when the V_{DD} voltage falls below the LVI trip falling voltage, V_{TRIPF}.

22.3 Features

Features of the LVI module include:

- Programmable LVI reset
- Selectable LVI trip voltage
- Programmable stop mode operation

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE0F	Low-Voltage Inhibit Status Register (LVISR)	Read:	LVIOUT	0	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 22-1. LVI I/O Register Summary

22.4 Functional Description

Figure 22-2 shows the structure of the LVI module. The LVI is enabled out of reset. The LVI module contains a bandgap reference circuit and comparator. Clearing the LVI power disable bit, LVIPWRD, enables the LVI to monitor V_{DD} voltage. Clearing the LVI reset disable bit, LVIRSTD, enables the LVI module to generate a reset when V_{DD} falls below a voltage, V_{TRIPF}. Setting the LVI enable in stop mode bit, LVISTOP, enables the LVI to operate in stop mode. Setting the LVI 5V or 3V trip point bit, LVI5OR3, enables the trip point voltage, V_{TRIPF}, to be configured for 5V operation. Clearing the LVI5OR3 bit enables the trip point voltage, V_{TRIPF}, to be configured for 3V operation. The actual trip points are shown in Section 24. Electrical Specifications.

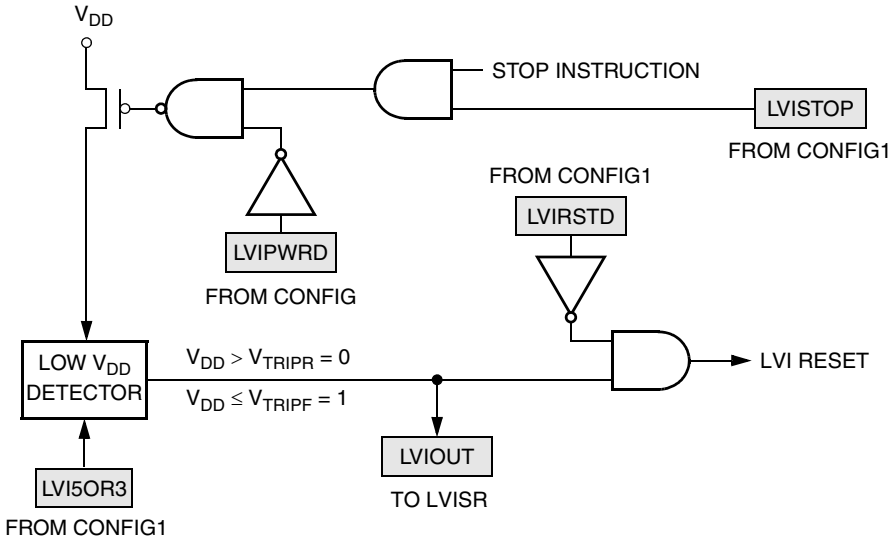


Figure 22-2. LVI Module Block Diagram

NOTE: *After a power-on reset (POR) the LVI's default mode of operation is 3V. If a 5V system is used, the user must set the LVI5OR3 bit to raise the trip point to 5V operation. Note that this must be done after every power-on reset since the default will revert back to 3V mode after each power-on reset. If the V_{DD} supply is below the 5V mode trip voltage but above the 3V mode trip voltage when POR is released, the MCU will operate because V_{TRIPF} defaults to 3V mode after a POR. So, in a 5V system care must be taken to ensure that V_{DD} is above the 5V mode trip voltage after POR is released.*

NOTE: *If the user requires 5V mode and sets the LVI5OR3 bit after a power-on reset while the V_{DD} supply is not above the V_{TRIPF} for 5V mode, the MCU will immediately go into reset. The LVI in this case will hold the MCU in reset until either V_{DD} goes above the rising 5V trip point, V_{TRIPR}, which will release reset or V_{DD} decreases to approximately 0V which will re-trigger the power-on reset and reset the trip point to 3V operation.*

LVISTOP, LVIPWRD, LVI5OR3, and LVIRSTD are in the configuration register 1 (CONFIG1). See **Section 5. Configuration and Mask Option Registers (CONFIG & MOR)** for details of the LVI's configuration bits. Once an LVI reset occurs, the MCU remains in reset until V_{DD} rises above a voltage, V_{TRIPR}, which causes the MCU to exit reset. See **9.4.2.5 Low-Voltage Inhibit (LVI) Reset** for details of the interaction between the SIM and the LVI. The output of the comparator controls the state of the LVIOUT flag in the LVI status register (LVISR).

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

22.4.1 Polled LVI Operation

In applications that can operate at V_{DD} levels below the V_{TRIPF} level, software can monitor V_{DD} by polling the LVIOUT bit. In the configuration register 1 (CONFIG1), the LVIPWRD bit must be at logic 0 to enable the LVI module, and the LVIRSTD bit must be at logic 1 to disable LVI resets.

22.4.2 Forced Reset Operation

In applications that require V_{DD} to remain above the V_{TRIPF} level, enabling LVI resets allows the LVI module to reset the MCU when V_{DD} falls below the V_{TRIPF} level. In the configuration register 1 (CONFIG1), the LVIPWRD and LVIRSTD bits must be at logic 0 to enable the LVI module and to enable LVI resets.

22.4.3 Voltage Hysteresis Protection

Once the LVI has triggered (by having V_{DD} fall below V_{TRIPF}), the LVI will maintain a reset condition until V_{DD} rises above the rising trip point voltage, V_{TRIPR}. This prevents a condition in which the MCU is continually entering and exiting reset if V_{DD} is approximately equal to V_{TRIPF}. V_{TRIPR} is greater than V_{TRIPF} by the hysteresis voltage, V_{HYS}.

22.4.4 LVI Trip Selection

The LVI5OR3 bit in the CONFIG1 register selects whether the LVI is configured for 5V or 3V protection.

NOTE: *The MCU is guaranteed to operate at a minimum supply voltage. The trip point (V_{TRIPF} [5 V] or V_{TRIPF} [3 V]) may be lower than this. (See **Section 24. Electrical Specifications** for the actual trip point voltages.)*

22.5 LVI Status Register

The LVI status register (LVISR) indicates if the V_{DD} voltage was detected below the V_{TRIPF} level.

Address: $FE0F

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

= Unimplemented

Figure 22-3. LVI Status Register

LVIOUT — LVI Output Bit

This read-only flag becomes set when the V_{DD} voltage falls below the V_{TRIPF} trip voltage (see Table 22-1). Reset clears the LVIOUT bit.

Table 22-1. LVIOUT Bit Indication

V_{DD}	LVIOUT
$V_{DD} > V_{TRIPR}$	0
$V_{DD} < V_{TRIPF}$	1
$V_{TRIPF} < V_{DD} < V_{TRIPR}$	Previous value

22.6 LVI Interrupts

The LVI module does not generate interrupt requests.

22.7 Low-Power Modes

The STOP and WAIT instructions put the MCU in low power-consumption standby modes.

22.7.1 Wait Mode

If enabled, the LVI module remains active in wait mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of wait mode.

22.7.2 Stop Mode

If enabled in stop mode (LVISTOP = 1), the LVI module remains active in stop mode. If enabled to generate resets (LVIRSTD = 0), the LVI module can generate a reset and bring the MCU out of stop mode.

Section 23. Break Module (BRK)

23.1 Contents

23.2 Introduction 363
23.3 Features 364
23.4 Functional Description 364
23.4.1 Flag Protection During Break Interrupts 366
23.4.2 CPU During Break Interrupts 366
23.4.3 TIM1 and TIM2 During Break Interrupts 366
23.4.4 COP During Break Interrupts 366
23.5 Low-Power Modes 366
23.5.1 Wait Mode 366
23.5.2 Stop Mode 367
23.6 Break Module Registers 367
23.6.1 Break Status and Control Register 367
23.6.2 Break Address Registers 368
23.6.3 SIM Break Status Register 368
23.6.4 SIM Break Flag Control Register 370

23.2 Introduction

This section describes the break module. The break module can generate a break interrupt that stops normal program flow at a defined address to enter a background program.

23.3 Features

Features of the break module include:

- Accessible input/output (I/O) registers during the break interrupt
- CPU-generated break interrupts
- Software-generated break interrupts
- COP disabling during break interrupts

23.4 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal to the CPU. The CPU then loads the instruction register with a software interrupt instruction (SWI) after completion of the current CPU instruction. The program counter vectors to $FFFC and $FFFD ($FEFC and $FEFD in monitor mode).

The following events can cause a break interrupt to occur:

- A CPU-generated address (the address in the program counter) matches the contents of the break address registers.
- Software writes a logic 1 to the BRKA bit in the break status and control register.

When a CPU-generated address matches the contents of the break address registers, the break interrupt begins after the CPU completes its current instruction. A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation. **Figure 23-1** shows the structure of the break module.

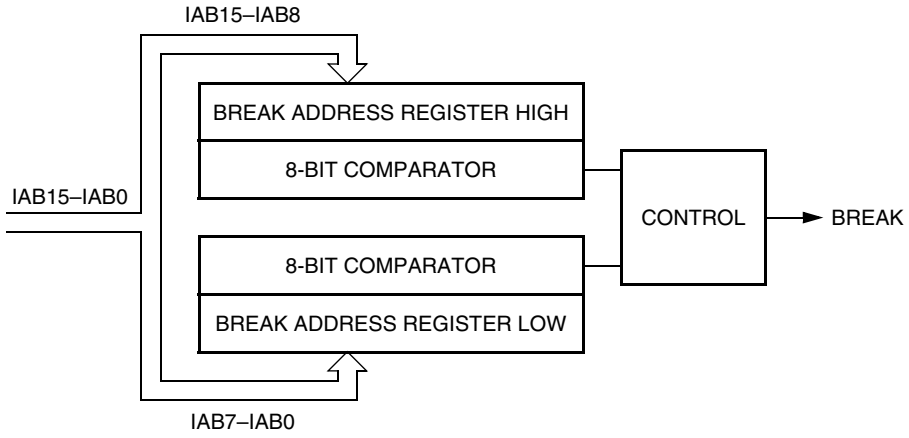


Figure 23-1. Break Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE00	SIM Break Status Register (SBSR)	Read:	R	R	R	R	R	R	SBSW	R
		Write:							Note	
		Reset:							0	
$FE03	SIM Break Flag Control Register (SBFCR)	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE0C	Break Address Register High (BRKH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0D	Break Address Register Low (BRKL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0E	Break Status and Control Register (BRKSCR)	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

Note: Writing a logic 0 clears SBSW.

= Unimplemented R = Reserved

Figure 23-2. Break Module I/O Register Summary

23.4.1 Flag Protection During Break Interrupts

The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state.

23.4.2 CPU During Break Interrupts

The CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with $FFFC and $FFFD ($FEFC and $FEFD in monitor mode)

The break interrupt begins after completion of the CPU instruction in progress. If the break address register match occurs on the last cycle of a CPU instruction, the break interrupt begins immediately.

23.4.3 TIM1 and TIM2 During Break Interrupts

A break interrupt stops the timer counters.

23.4.4 COP During Break Interrupts

The COP is disabled during a break interrupt when V_{TST} is present on the $\overline{RST}$ pin.

23.5 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

23.5.1 Wait Mode

If enabled, the break module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if SBSW is set (see **Section 9. System Integration Module (SIM)**). Clear the SBSW bit by writing logic 0 to it.

23.5.2 Stop Mode

A break interrupt causes exit from stop mode and sets the SBSW bit in the break status register.

23.6 Break Module Registers

These registers control and monitor operation of the break module:

- Break status and control register (BRKSCR)
- Break address register high (BRKH)
- Break address register low (BRKL)
- SIM break status register (SBSR)
- SIM break flag control register (SBFCR)

23.6.1 Break Status and Control Register

The break status and control register (BRKSCR) contains break module enable and status bits.

Address: $FE0E

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

= Unimplemented

Figure 23-3. Break Status and Control Register (BRKSCR)

BRKE — Break Enable Bit

This read/write bit enables breaks on break address register matches. Clear BRKE by writing a logic 0 to bit 7. Reset clears the BRKE bit.

1 = Breaks enabled on 16-bit address match
0 = Breaks disabled on 16-bit address match

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a logic 1 to BRKA generates a break interrupt. Clear BRKA by writing a logic 0 to it before exiting the break routine. Reset clears the BRKA bit.

1 = (When read) Break address match
0 = (When read) No break address match

23.6.2 Break Address Registers

The break address registers (BRKH and BRKL) contain the high and low bytes of the desired breakpoint address. Reset clears the break address registers.

Address: $FE0C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
Reset:	0	0	0	0	0	0	0	0

Figure 23-4. Break Address Register High (BRKH)

Address: $FE0D

	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

Figure 23-5. Break Address Register Low (BRKL)

23.6.3 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from wait mode. The flag is useful in applications requiring a return to wait mode after exiting from a break interrupt.

Address: $FE00

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	R	R	R	R	R	SBSW	R
Write:							Note	
Reset:							0	

Note: Writing a logic 0 clears SBSW. R = Reserved

Figure 23-6. SIM Break Status Register (SBSR)

SBSW — Break Wait Bit

This status bit is set when a break interrupt causes an exit from wait mode or stop mode. Clear SBSW by writing a logic 0 to it. Reset clears SBSW.

1 = Stop mode or wait mode was exited by break interrupt
0 = Stop mode or wait mode was not exited by break interrupt

SBSW can be read within the break interrupt routine. The user can modify the return address on the stack by subtracting 1 from it. The following code is an example.

```
;This code works if the H register has been pushed onto the stack in the break
;service routine software. This code should be executed at the end of the break
;service routine software.

 HIBYTE   EQU     5

 LOBYTE   EQU     6

;         If not SBSW, do RTI

          BRCLR   SBSW,SBSR, RETURN    ;See if wait mode or stop mode was exited by
                                       ;break.

          TST     LOBYTE,SP            ;If RETURNLO is not zero,

          BNE     DOLO                 ;then just decrement low byte.

          DEC     HIBYTE,SP            ;Else deal with high byte, too.

 DOLO     DEC     LOBYTE,SP            ;Point to WAIT/STOP opcode.

 RETURN   PULH                         ;Restore H register.
          RTI
```

23.6.4 SIM Break Flag Control Register

The SIM break flag control register (SBFCR) contains a bit that enables software to clear status bits while the MCU is in a break state.

Address: $FE03

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

R = Reserved

Figure 23-7. SIM Break Flag Control Register (SBFCR)

BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

1 = Status bits clearable during break

0 = Status bits not clearable during break

Section 24. Electrical Specifications

24.1 Contents

24.2 Introduction 372
24.3 Absolute Maximum Ratings 372
24.4 Functional Operating Range 373
24.5 Thermal Characteristics 373
24.6 5.0V DC Electrical Characteristics 374
24.7 3.0V DC Electrical Characteristics 376
24.8 5.0V Control Timing 377
24.9 3.0V Control Timing 377
24.10 5.0V Oscillator Characteristics 378
24.11 3.0V Oscillator Characteristics 379
24.12 5.0V ADC Electrical Characteristics 380
24.13 3.0V ADC Electrical Characteristics 381
24.14 Analog Module Electrical Characteristics 382
24.14.1 Temperature Sensor Electrical Characteristics 382
24.14.2 Current Detection Electrical Characteristics 382
24.14.3 Two-Stage Amplifier Electrical Characteristics 382
24.15 Timer Interface Module Characteristics 383
24.16 MMIIC Electrical Characteristics 383
24.17 CGM Electrical Specifications 385
24.18 FLASH Memory Characteristics 386

24.2 Introduction

This section contains electrical and timing specifications.

24.3 Absolute Maximum Ratings

Maximum ratings are the extreme limits to which the MCU can be exposed without permanently damaging it.

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to* ***24.6 5.0V DC Electrical Characteristics*** *for guaranteed operating conditions.*

Table 24-1. Absolute Maximum Ratings(1)

Characteristic	Symbol	Value	Unit
Supply voltage	V_{DD}	−0.3 to +6.0	V
Input voltage All pins (except $\overline{IRQ1}$) $\overline{IRQ1}$ pin	V_{IN}	V_{SS}−0.3 to V_{DD} +0.3 V_{SS}−0.3 to 8.5	V
Maximum current per pin excluding V_{DD} and V_{SS}	I	±25	mA
Maximum current out of V_{SS}	I_{MVSS}	100	mA
Maximum current into V_{DD}	I_{MVDD}	100	mA
Storage temperature	T_{STG}	−55 to +150	°C

Notes:
1. Voltages referenced to V_{SS}.

NOTE: *This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{IN} and V_{OUT} be constrained to the range $V_{SS} \le (V_{IN}$ or $V_{OUT}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD}.)*

24.4 Functional Operating Range

Table 24-2. Operating Range

Characteristic	Symbol	Value		Unit
Operating temperature range	T_A	−40 to +125	−40 to +85	°C
Operating voltage range	V_{DD}	— 5V ± 10%	3V ± 10%(1) 5V ± 10%	V

Notes:
1. A minimum operating V_{DD} of 3V is required to achieve the ADC module specifications stated in **Table 24-11 . 3V ADC Electrical Characteristics**.

24.5 Thermal Characteristics

Table 24-3. Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance 42-pin SDIP 48-pin LQFP	θ_{JA}	 60 80	 °C/W °C/W
I/O pin power dissipation	$P_{I/O}$	User determined	W
Power dissipation(1)	P_D	$P_D = (I_{DD} \times V_{DD}) + P_{I/O} = K/(T_J + 273\ °C)$	W
Constant(2)	K	$P_D \times (T_A + 273\ °C) + P_D^2 \times \theta_{JA}$	W/°C
Average junction temperature	T_J	$T_A + (P_D \times \theta_{JA})$	°C

Notes:
1. Power dissipation is a function of temperature.
2. K constant unique to the device. K can be determined for a known T_A and measured P_D. With this value of K, P_D and T_J can be determined for any value of T_A.

24.6 5.0V DC Electrical Characteristics

Table 24-4. 5V DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage (I_{LOAD} = −2.0 mA) PTA[0:7], PTB[4:6], PTC[0:7], PTD[0:7]	V_{OH}	V_{DD}−0.8	—	—	V
Output low voltage (I_{LOAD} = 1.6mA) PTA[0:7], PTB[0:6], PTC[0:7], PTD[0:7]	V_{OL}	—	—	0.4	V
LED sink current (V_{Drain} = 4.0V) PTA[0:5], PTC[3:7]	I_{OL}	—	−15	—	mA
Input high voltage PTA[0:7], PTB[0:6], PTC[0:7], PTD[0:7], $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, OSC1.	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage PTA[0:7], PTB[0:6], PTC[0:7], PTD[0:7], $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, OSC1	V_{IL}	V_{SS}	—	0.3 × V_{DD}	V
V_{DD} supply current					
Run[3], f_{OP} = 8.0 MHz					
with ADC on		—	24	40	mA
with ADC off		—	18	30	mA
Wait[4], f_{OP} = 8.0 MHz		—	7.5	15	mA
Stop[5]					
25°C (with OSC, TBM, current sense, LVI)		—	50	150	µA
25°C (with OSC, TBM, current sense)		—	12	40	µA
25°C (with OSC, TBM)		—	9	30	µA
25°C		—	1	10	µA
−40°C to 85°C (with OSC, TBM, current sense, LVI)		—	—	180	µA
−40°C to 85°C (with OSC, TBM, current sense)		—	—	50	µA
−40°C to 85°C (with OSC, TBM)		—	—	40	µA
−40°C to 85°C		—	—	15	µA
−40°C to 125°C (with OSC, TBM, current sense, LVI)		—	—	200	µA
−40°C to 125°C (with OSC, TBM, current sense)		—	—	60	µA
−40°C to 125°C (with OSC, TBM)		—	—	50	µA
−40°C to 125°C		—	—	25	µA
Digital I/O ports Hi-Z leakage current	I_{IL}	—	—	± 10	µA
Input current	I_{IN}	—	—	± 1	µA
Capacitance Ports (as input or output)	C_{OUT} C_{IN}	— —	— —	12 8	pF
POR re-arm voltage[6]	V_{POR}	0	—	100	mV

Table 24-4. 5V DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
POR rise-time ramp rate[7]	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{TST}	$1.4 \times V_{DD}$	—	8	V
Pullup resistors[8] PTD[0:7] configured as KBI[0:7] $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$	 R_{PU1} R_{PU2}	 24 24	 35 35	 42 42	 kΩ kΩ
Low-voltage inhibit, trip falling voltage	V_{LVII5}	3.80	4.15	4.45	V
Low-voltage inhibit, trip rising voltage	V_{LVII5}	3.95	4.30	4.60	V
Schmitt trigger input low level trip voltage $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, KBI[0:7]	V_{SCMTL}	—	1.21	—	V
Schmitt trigger input high level trip voltage $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, KBI[0:7]	V_{SCMTH}	—	1.65	—	V

Notes:
1. V_{DD} = 4.5 to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
3. Run (operating) I_{DD} measured using external square wave clock source. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD}. Measured with all modules enabled.
4. Wait I_{DD} measured using external square wave clock source. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects wait I_{DD}.
5. STOP I_{DD} measured with OSC1 grounded, no port pins sourcing current.
6. Maximum is highest voltage that POR is guaranteed.
7. If minimum V_{DD} is not reached before the internal POR reset is released, $\overline{RST}$ must be driven low externally until minimum V_{DD} is reached.
8. R_{PU1} and R_{PU2} are measured at V_{DD} = 5.0V

24.7 3.0V DC Electrical Characteristics

Table 24-5. 3V DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage (I_{LOAD} = −1.0mA) PTA[0:7], PTB[4:6], PTC[0:7], PTD[0:7]	V_{OH}	V_{DD}−0.4	—	—	V
Output low voltage (I_{LOAD} = 0.8mA) PTA[0:7], PTB[0:6], PTC[0:7], PTD[0:7]	V_{OL}	—	—	0.4	V
LED sink current (V_{Drain} = 2.0V) PTA[0:5], PTC[3:7]	V_{OL}	—	−5	—	mA
Input high voltage PTA[0:7], PTB[0:6], PTC[0:7], PTD[0:7], $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage PTA[0:7], PTB[0:6], PTC[0:7], PTD[0:7], $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, OSC1	V_{IL}	V_{SS}	—	0.3 × V_{DD}	V
V_{DD} supply current	I_{DD}				
Run[3], f_{OP} = 4.0 MHz					
with ADC on		—	7	16	mA
with ADC off		—	5	12	mA
Wait[4], f_{OP} = 4.0 MHz		—	2.5	6	mA
Stop[5]					
25°C (with OSC, TBM, LVI)		—	27	80	µA
25°C (with OSC, TBM)		—	5	12	µA
25°C		—	0.5	5	µA
−40°C to 85°C (with OSC, TBM, LVI)		—	—	100	µA
−40°C to 85°C (with OSC, TBM)		—	—	15	µA
−40°C to 85°C		—	—	5	µA
Digital I/O ports Hi-Z leakage current	I_{IL}	—	—	± 10	µA
Input current	I_{IN}	—	—	± 1	µA
Capacitance Ports (as input or output)	C_{OUT} C_{IN}	— —	— —	12 8	pF
POR re-arm voltage[6]	V_{POR}	0	—	100	mV
POR rise-time ramp rate[7]	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{HI}	1.4 × V_{DD}	—	2.0 × V_{DD}	V
Pullup resistors[8] PTD[0:7] configured as KBI[0:7] $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$	 R_{PU1} R_{PU2}	 24 24	 35 35	 42 42	 kΩ kΩ

Table 24-5. 3V DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Low-voltage inhibit, trip voltage (No hysteresis implemented for 3V LVI)	V_{LVI3}	2.32	2.49	2.68	V
Schmitt trigger input low level trip voltage $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, KBI[0:7]	V_{SCMTL}	—	0.8	—	V
Schmitt trigger input high level trip voltage $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, KBI[0:7]	V_{SCMTH}	—	1.2	—	V

Notes:
1. V_{DD} = 2.7 to 3.3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
3. Run (operating) I_{DD} measured using external square wave clock source. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD}. Measured with all modules enabled.
4. Wait I_{DD} measured using external square wave clock source. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects wait I_{DD}.
5. STOP I_{DD} measured with OSC1 grounded, no port pins sourcing current.
6. Maximum is highest voltage that POR is guaranteed.
7. If minimum V_{DD} is not reached before the internal POR reset is released, $\overline{RST}$ must be driven low externally until minimum V_{DD} is reached.
8. R_{PU1} and R_{PU2} are measured at V_{DD} = 5.0V.

24.8 5.0V Control Timing

Table 24-6. 5V Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency[2]	f_{OP}	—	8.0	MHz
$\overline{RST}$ input pulse width low[3]	t_{IRL}	750	—	ns

Notes:
1. V_{SS} = 0 Vdc; timing shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Some modules may require a minimum frequency greater than dc for proper operation; see appropriate table for this information.
3. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.

24.9 3.0V Control Timing

Table 24-7. 3V Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency[2]	f_{OP}	—	4.0	MHz
$\overline{RST}$ input pulse width low[3]	t_{IRL}	1.5	—	µs

Notes:
1. V_{SS} = 0 Vdc; timing shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Some modules may require a minimum frequency greater than dc for proper operation; see appropriate table for this information.
3. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.

24.10 5.0V Oscillator Characteristics

Table 24-8. 5V Oscillator Specifications

Characteristic	Symbol	Min	Typ	Max	Unit
Internal oscillator clock frequency	f_{ICLK}	19.2k	24k	28.8k	Hz
External reference clock to OSC1(1)	f_{OSC}	dc	—	20M	Hz
Crystal reference frequency(2)	f_{XCLK}		32.768k	4.9152M	Hz
Crystal load capacitance(3)	C_L	—	—	—	
Crystal fixed capacitance	C_1	—	$2 \times C_L$ (25p)	—	F
Crystal tuning capacitance	C_2	—	$2 \times C_L$ (25p)	—	F
Feedback bias resistor	R_B	—	10M	—	Ω
Series resistor(4)	R_S	—	100k	—	Ω
External RC clock frequency	f_{RCCLK}	2M	—	18M	Hz
External resistor	R_{EXT}	See Figure 24-1			Ω
External capacitor	C_{EXT}	—	10	—	pF

Notes:
1. No more than 10% duty cycle deviation from 50%.
2. Fundamental mode crystals only.
3. Consult crystal manufacturer's data.
4. Not Required for high frequency crystals.

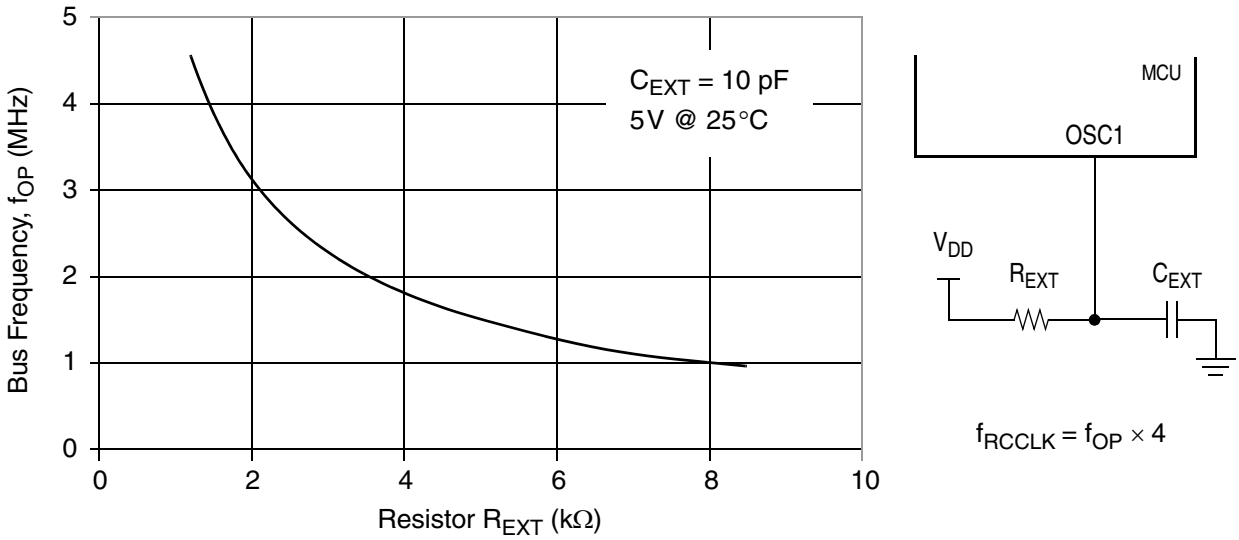


Figure 24-1. RC vs. Bus Frequency (5V @25°C)

24.11 3.0V Oscillator Characteristics

Table 24-9. 3V Oscillator Specifications

Characteristic	Symbol	Min	Typ	Max	Unit
Internal oscillator clock frequency	f_{ICLK}	13.8k	17.2k	20.6k	Hz
External reference clock to OSC1(1)	f_{OSC}	dc	—	16M	Hz
Crystal reference frequency(2)	f_{XCLK}		32.768k	4.9152M	Hz
Crystal load capacitance(3)	C_L	—	—	—	
Crystal fixed capacitance	C_1	—	$2 \times C_L$ (25p)	—	F
Crystal tuning capacitance	C_2	—	$2 \times C_L$ (25p)	—	F
Feedback bias resistor	R_B	—	10M	—	Ω
Series resistor(4)	R_S	—	100k	—	Ω
External RC clock frequency	f_{RCCLK}	2M	—	10M	Hz
External resistor	R_{EXT}	See Figure 24-2			Ω
External capacitor	C_{EXT}	—	10	—	pF

Notes:
1. No more than 10% duty cycle deviation from 50%.
2. Fundamental mode crystals only.
3. Consult crystal manufacturer's data.
4. Not Required for high frequency crystals.

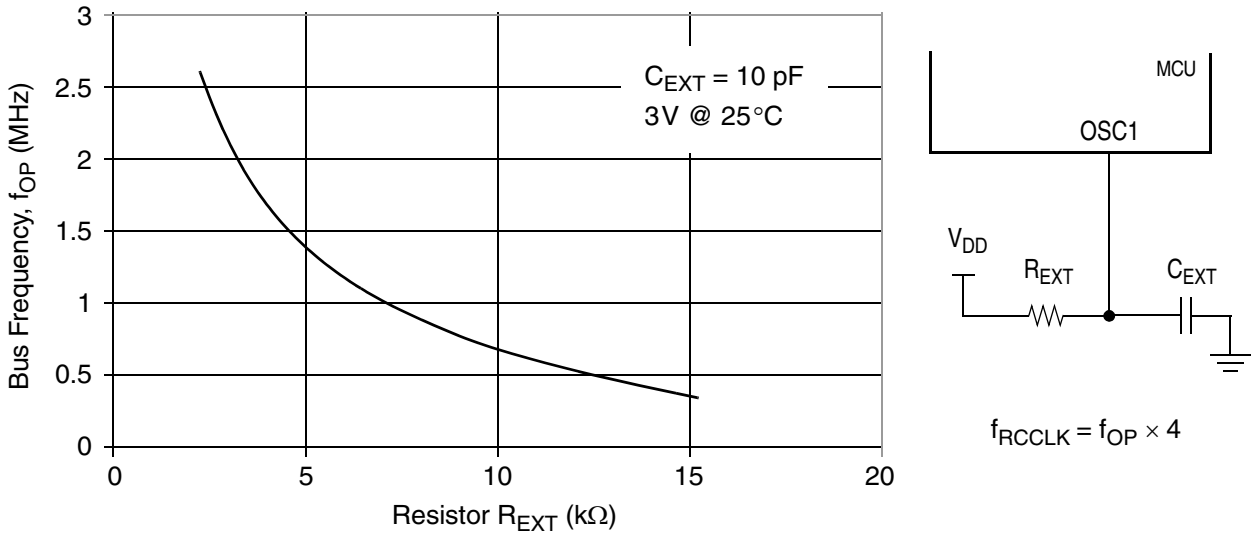


Figure 24-2. RC vs. Bus Frequency (3V @25°C)

24.12 5.0V ADC Electrical Characteristics

Table 24-10. 5V ADC Electrical Characteristics

Characteristic	Symbol	Min	Max	Unit	Notes
Supply voltage	V_{DDA}	4.5	5.5	V	V_{DDA} is an dedicated pin and should be tied to V_{DD} on the PCB with proper decoupling.
Input range	V_{ADIN}	0	V_{DDA}	V	$V_{ADIN} \leq V_{DDA}$
Resolution	B_{AD}	10	10	bits	
Absolute accuracy	A_{AD}	—	± 1.5	LSB	Includes quantization. ±0.5 LSB = ±1 ADC step.
ADC internal clock	f_{ADIC}	500k	2M	Hz	$t_{ADIC} = 1/f_{ADIC}$
Conversion range	R_{AD}	V_{REFL}	V_{REFH}	V	
ADC voltage reference high	V_{REFH}	—	V_{DDA} + 0.1	V	
ADC voltage reference low	V_{REFL}	V_{SSA} – 0.1	—	V	
Conversion time	t_{ADC}	16	17	t_{ADIC} cycles	
Sample time	t_{ADS}	5	—	t_{ADIC} cycles	
Monotonicity	M_{AD}	Guaranteed			
Zero input reading	Z_{ADI}	000	001	HEX	$V_{ADIN} = V_{REFL}$
Full-scale reading	F_{ADI}	3FD	3FF	HEX	$V_{ADIN} = V_{REFH}$
Input capacitance	C_{ADI}	—	20	pF	Not tested.
Input impedance	R_{ADI}	20M	—	Ω	
V_{REFH}/V_{REFL}	I_{VREF}	—	1.6	mA	Not tested.

24.13 3.0V ADC Electrical Characteristics

Table 24-11. 3V ADC Electrical Characteristics

Characteristic	Symbol	Min	Max	Unit	Notes
Supply voltage	V_{DDA}	3.0	3.3	V	V_{DDA} is an dedicated pin and should be tied to V_{DD} on the PCB with proper decoupling.
Input range	V_{ADIN}	0	V_{DDA}	V	$V_{ADIN} \leq V_{DDA}$
Resolution	B_{AD}	10	10	bits	
Absolute accuracy	A_{AD}	—	± 1.5	LSB	Includes quantization. ±0.5 LSB = ±1 ADC step.
ADC internal clock	f_{ADIC}	500 k	2 M	Hz	$t_{ADIC} = 1/f_{ADIC}$
Conversion range	R_{AD}	V_{REFL}	V_{REFH}	V	
ADC voltage reference high	V_{REFH}	—	V_{DDA} + 0.1	V	
ADC voltage reference low	V_{REFL}	V_{SSA} – 0.1	—	V	
Conversion time	t_{ADC}	16	17	t_{ADIC} cycles	
Sample time	t_{ADS}	5	—	t_{ADIC} cycles	
Monotonicity	M_{AD}	Guaranteed			
Zero input reading	Z_{ADI}	000	001	HEX	$V_{ADIN} = V_{REFL}$
Full-scale reading	F_{ADI}	3FD	3FF	HEX	$V_{ADIN} = V_{REFH}$
Input capacitance	C_{ADI}	—	20	pF	Not tested.
Input impedance	R_{ADI}	20M	—	Ω	Measured at 5V
V_{REFH}/V_{REFL}	I_{VREF}	—	1.6	mA	Not tested.

24.14 Analog Module Electrical Characteristics

24.14.1 Temperature Sensor Electrical Characteristics

Table 24-12. Temperature Sensor Electrical Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Temperature range		−20	—	70	°C
Temperature slope V_{DD}=5V ± 10%, GAINA=2, GAINB=6 V_{DD}=3V ± 10%, GAINA=2, GAINB=4		 1.275 1.048	 1.338 1.089	 1.372 1.146	ADC steps/°C

24.14.2 Current Detection Electrical Characteristics

Table 24-13. Current Detection Electrical Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Trip point(1)	V_{DET}	−6	—	+12	mV

Notes:
1. The current detect comparator is designed for V_{DD}=5V ± 10% only.

24.14.3 Two-Stage Amplifier Electrical Characteristics

Table 24-14. Two-Stage Amplifier Electrical Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Amplifier input signal hold time	t_{AMH}	10 + [(GAINA – 1) × 2]			t_{AM} cycles(1)
Amplifier response time	t_{AMR}	70 + (8×GAINA) + (6×GAINB)			t_{AM} cycles
Amplifier gain tolerance V_{DD}=5V ± 10%, GAINA=4, GAINB=16 V_{IN} = 10mV to 30mV V_{IN} = 30mV to 65mV V_{DD}=5V ± 10%, GAINA=6, GAINB=16 V_{IN} = 10mV to 30mV V_{IN} = 30mV to 44mV V_{DD}=3V ± 10%, GAINA=4, GAINB=16 V_{IN} = 10mV to 38mV V_{DD}=3V ± 10%, GAINA=6, GAINB=16 V_{IN} = 10mV to 24mV		 −3.5 −1.5 −3.5 −1.5 −3.5 −3.5	 — — — — — —	 +3.5 +1.5 +3.5 +1.5 +3.5 +3.5	%

Notes:
1. t_{AM} is the AMCLK.

24.15 Timer Interface Module Characteristics

Characteristic	Symbol	Min	Max	Unit
Input capture pulse width	t_{TIH}, t_{TIL}	1	—	t_{cyc}

24.16 MMIIC Electrical Characteristics

Table 24-15. MMIIC DC Electrical Characteristics

Characteristic(1)	Symbol	Min	Typ	Max	Unit	Comments
Input low	V_{IL}	–0.5		0.8	V	Data, clock input low.
Input high	V_{IH}	2.1		5.5	V	Data, clock input high.
Output low	V_{OL}			0.4	V	Data, clock output low; @$I_{PULLUP,MAX}$
Input leakage	I_{LEAK}			± 5	μA	Input leakage current
Pullup current	I_{PULLUP}	100		350	μA	Current through pull-up resistor or current source. See note.(2)

Notes:
1. V_{DD} = 2.7 to 5.5Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. The I_{PULLUP} (max) specification is determined primarily by the need to accommodate a maximum of 1.1kΩ equivalent series resistor of removable SMBus devices, such as the smart battery, while maintaining the V_{OL} (max) of the bus.

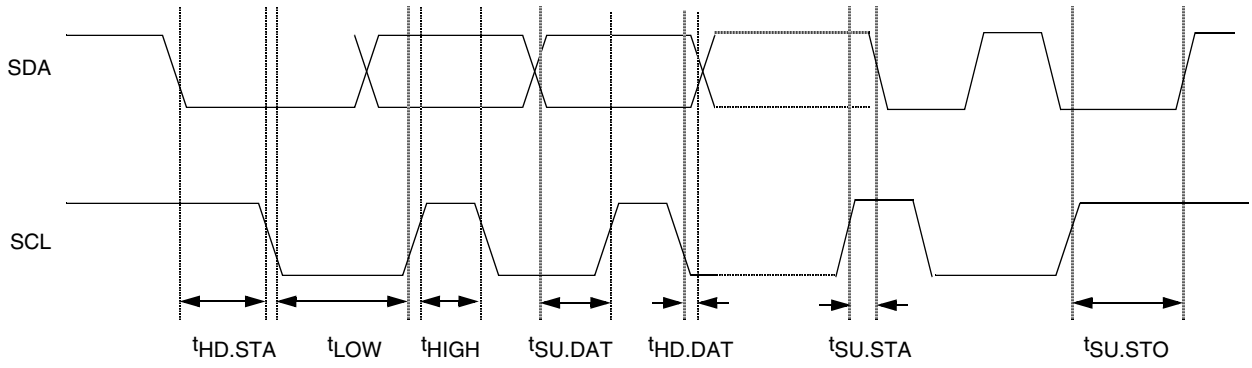


Figure 24-3. MMIIC Signal Timings

See **Table 24-16** for MMIIC timing parameters.

Table 24-16. MMIIC Interface Input/Output Signal Timing

Characteristic	Symbol	Min	Typ	Max	Unit	Comments
Operating frequency	f_{SMB}	10		100	kHz	MMIIC operating frequency
Bus free time	t_{BUF}	4.7			μs	Bus free time between STOP and START condition
Repeated start hold time.	$t_{HD.STA}$	4.0			μs	Hold time after (repeated) START condition. After this period, the first clock is generated.
Repeated start setup time.	$t_{SU.STA}$	4.7			μs	Repeated START condition setup time.
Stop setup time	$t_{SU.STO}$	4.0			μs	Stop condition setup time.
Hold time	$t_{HD.DAT}$	300			ns	Data hold time.
Setup time	$t_{SU.DAT}$	250			ns	Data setup time.
Clock low time-out	$t_{TIMEOUT}$	25		35	ms	Clock low time-out.(1)
Clock low	t_{LOW}	4.7			μs	Clock low period
Clock high	t_{HIGH}	4.0			μs	Clock high period.(2)
Slave clock low extend time	$t_{LOW.SEXT}$			25	ms	Cumulative clock low extend time (slave device)(3)
Master clock low extend time	$t_{LOW.MEXT}$			10	ms	Cumulative clock low extend time (master device) (4)
Fall time	t_F			300	ns	Clock/Data Fall Time(5)
Rise time	t_R			1000	ns	Clock/Data Rise Time(5)

Notes:

1. Devices participating in a transfer will timeout when any clock low exceeds the value of $T_{TIMEOUT}$ min. of 25ms. Devices that have detected a timeout condition must reset the communication no later than $T_{TIMEOUT}$ max of 35ms. The maximum value specified must be adhered to by both a master and a slave as it incorporates the cumulative limit for both a master (10 ms) and a slave (25 ms).
   Software should turn-off the MMIIC module to release the SDA and SCL lines.
2. $T_{HIGH\ MAX}$ provides a simple guaranteed method for devices to detect the idle conditions.
3. $T_{LOW.SEXT}$ is the cumulative time a slave device is allowed to extend the clock cycles in one message from the initial start to the stop. If a slave device exceeds this time, it is expected to release both its clock and data lines and reset itself.
4. $T_{LOW.MEXT}$ is the cumulative time a master device is allowed to extend its clock cycles within each byte of a message as defined from start-to-ack, ack-to-ack, or ack-to-stop.
5. Rise and fall time is defined as follows: $T_R = (V_{ILMAX} - 0.15)$ to $(V_{IHMIN} + 0.15)$, $T_F = 0.9 \times V_{DD}$ to $(V_{ILMAX} - 0.15)$.

24.17 CGM Electrical Specifications

Characteristic	Symbol	Min	Typ	Max	Unit
Reference frequency	f_{RDV}	30	32.768	100	kHz
Range nominal multiplies	f_{NOM}	—	38.4	—	kHz
VCO center-of-range frequency	f_{VRS}	38.4k	—	40.0M	Hz
VCO range linear range multiplier	L	1	—	255	
VCO power-of-two-range multiplier	2^E	1	—	4	
VCO multiply factor	N	1	—	4095	
VCO prescale multiplier	2^P	1	—	8	
Reference divider factor	R	1	1	15	
VCO operating frequency	f_{VCLK}	38.4k	—	40.0M	Hz
Manual acquisition time	t_{LOCK}	—	—	50	ms
Automatic lock time	t_{LOCK}	—	—	50	ms
Automatic lock time Wake up from stop with OSC enabled(1)	t_{LOCK}	—	10	15	ms
PLL jitter(2)	f_J	0	—	$f_{RCLK} \times 0.025\% \times 2^P N/4$	Hz

Notes:
1. Test condition: V_{DD} = 5.0Vdc / 3.0Vdc, V_{SS} = 0 Vdc. Reference frequency = 32.768kHz, locking to 4MHz bus frequency.
2. Deviation of average bus frequency over 2ms. N = VCO multiplier.

24.18 FLASH Memory Characteristics

Table 24-17. FLASH Memory Electrical Characteristics

Characteristic	Symbol	Min.	Max.	Unit
Data retention voltage	V_{RDR}	1.3	—	V
Number of rows per page		2		Rows
Number of bytes per page		128		Bytes
Read bus clock frequency	f_{Read} (1)	32k	8M	Hz
Page erase time	t_{Erase} (2)	1	—	ms
Mass erase time	t_{MErase} (3)	4	—	ms
PGM/ERASE to HVEN setup time	t_{nvs}	10	—	µs
High-voltage hold time	t_{nvh}	5	—	µs
High-voltage hold time (mass erase)	t_{nvhl}	100	—	µs
Program hold time	t_{pgs}	5	—	µs
Program time	t_{Prog}	30	40	µs
Address/data setup time	t_{ads}	—	30	ns
Address/data hold time	t_{adh}	—	30	ns
Recovery time	t_{rcv} (4)	1	—	µs
Cumulative HV period	t_{hv} (5)	—	25	ms
Row erase endurance (6)	—	10k	—	Cycles
Row program endurance (7)	—	10k	—	Cycles
Data retention time (8)	—	10	—	Years

Notes:
1. f_{Read} is defined as the frequency range for which the FLASH memory can be read.
2. If the page erase time is longer than t_{Erase} (Min.), there is no erase-disturb, but it reduces the endurance of the FLASH memory.
3. If the mass erase time is longer than t_{MErase} (Min.), there is no erase-disturb, but is reduces the endurance of the FLASH memory.
4. It is defined as the time it needs before the FLASH can be read after turning off the high voltage charge pump, by clearing HVEN to logic 0.
5. t_{hv} is the cumulative high voltage programming time to the same row before next erase, and the same address can not be programmed twice before next erase.
6. The minimum row endurance value specifies each row of the FLASH memory is guaranteed to work for at least this many erase/program cycles.
7. The minimum row endurance value specifies each row of the FLASH memory is guaranteed to work for at least this many erase/program cycle.
8. The FLASH is guaranteed to retain data over the entire operating temperature range for at least the minimum time specified.

Section 25. Mechanical Specifications

25.1 Contents

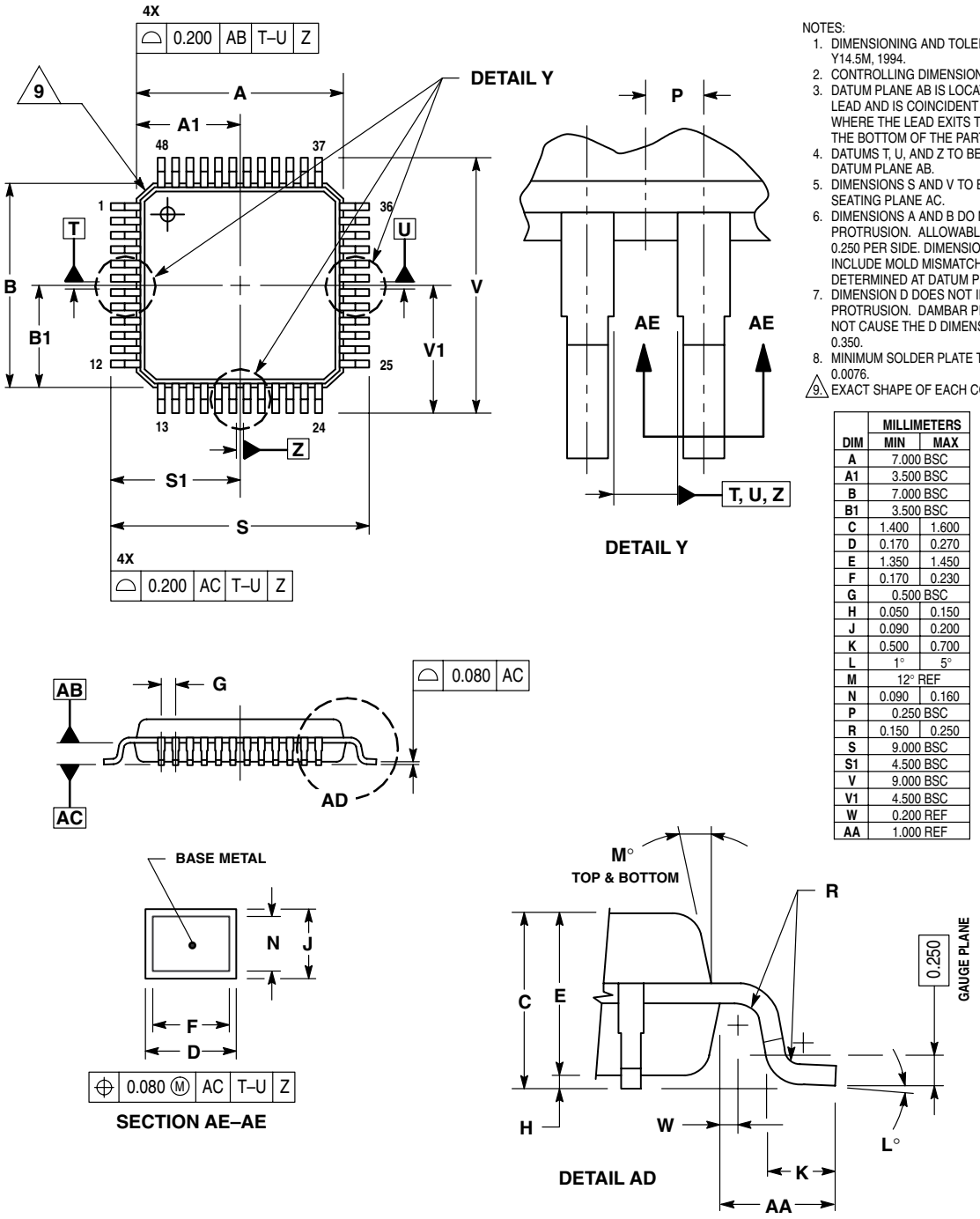
25.2 Introduction 387
25.3 48-Pin Plastic Low Quad Flat Pack (LQFP) 388
25.4 42-Pin Shrink Dual In-Line Package (SDIP) 389

25.2 Introduction

This section gives the dimensions for:

- 48-pin plastic low quad flat pack (case #932-02)
- 42-pin shrink dual in-line package (case #858-01)

25.3 48-Pin Plastic Low Quad Flat Pack (LQFP)

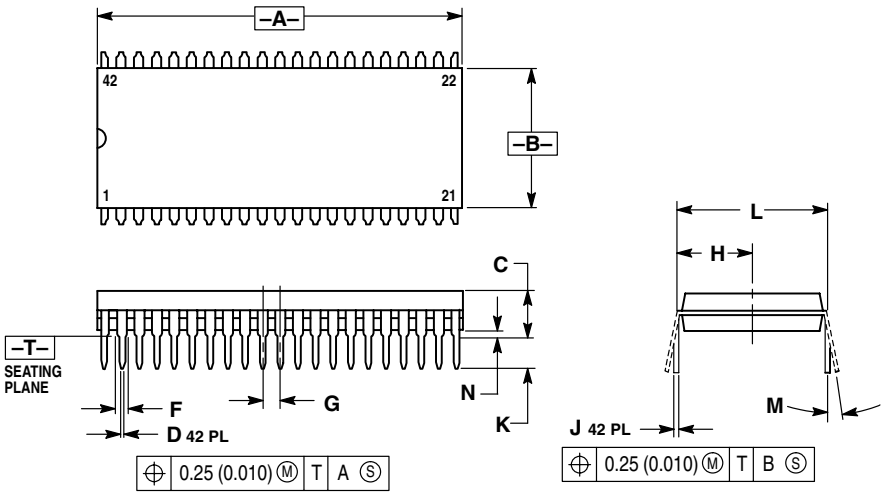


NOTES:
1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1994.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE AB IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS T, U, AND Z TO BE DETERMINED AT DATUM PLANE AB.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE AC.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.250 PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE AB.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE D DIMENSION TO EXCEED 0.350.
8. MINIMUM SOLDER PLATE THICKNESS SHALL BE 0.0076.
9. EXACT SHAPE OF EACH CORNER IS OPTIONAL.

	MILLIMETERS	
DIM	MIN	MAX
A	7.000 BSC	
A1	3.500 BSC	
B	7.000 BSC	
B1	3.500 BSC	
C	1.400	1.600
D	0.170	0.270
E	1.350	1.450
F	0.170	0.230
G	0.500 BSC	
H	0.050	0.150
J	0.090	0.200
K	0.500	0.700
L	1°	5°
M	12° REF	
N	0.090	0.160
P	0.250 BSC	
R	0.150	0.250
S	9.000 BSC	
S1	4.500 BSC	
V	9.000 BSC	
V1	4.500 BSC	
W	0.200 REF	
AA	1.000 REF	

Figure 25-1. 48-Pin LQFP (Case #932-02)

25.4 42-Pin Shrink Dual In-Line Package (SDIP)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSIONS A AND B DO NOT INCLUDE MOLD FLASH. MAXIMUM MOLD FLASH 0.25 (0.010).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	1.435	1.465	36.45	37.21
B	0.540	0.560	13.72	14.22
C	0.155	0.200	3.94	5.08
D	0.014	0.022	0.36	0.56
F	0.032	0.046	0.81	1.17
G	0.070 BSC		1.778 BSC	
H	0.300 BSC		7.62 BSC	
J	0.008	0.015	0.20	0.38
K	0.115	0.135	2.92	3.43
L	0.600 BSC		15.24 BSC	
M	0°	15°	0°	15°
N	0.020	0.040	0.51	1.02

Figure 25-2. 42-Pin SDIP (Case #858-01)

Section 26. Ordering Information

26.1 Contents

26.2 Introduction 391
26.3 MC Order Numbers 391

26.2 Introduction

This section contains ordering numbers for the MC68HC908SR12.

26.3 MC Order Numbers

Table 26-1. MC Order Numbers

MC order number	Operating temperature range	Package
MC68HC908SR12CB	–40 °C to +85 °C	42-Pin SDIP(1)
MC68HC908SR12MB(2)	–40 °C to +125 °C	
MC68HC908SR12CFA	–40 °C to +85 °C	48-pin LQFP(3)
MC68HC908SR12MFA(2)	–40 °C to +125 °C	

Notes:
1. SDIP = Shrink Dual In-Line Package.
2. Temperature grade "M" is available for 5V operating voltage only.
3. LQFP = Low Quad Flat Pack.

Ordering Information

Appendix A. MC68HC08SR12

A.1 Contents

A.2 Introduction 394
A.3 MCU Block Diagram 394
A.4 Memory Map 394
A.5 Mask Option Register 397
A.6 Reserved Registers 397
A.7 Monitor ROM 397
A.8 Electrical Specifications 397
A.8.1 5.0V DC Electrical Characteristics 398
A.8.2 3.0V DC Electrical Characteristics 399
A.8.3 Memory Characteristics 401
A.9 ROM Order Numbers 401

A.2 Introduction

This section introduces the MC68HC08SR12, the ROM part equivalent to the MC68HC908SR12. The entire data book apply to this ROM device, with exceptions outlined in this appendix.

Table A-1. Summary of MC68HC08SR12 and MC68HC908SR12 Differences

	MC68HC08SR12	MC68HC908SR12
Memory ($C000–$EFFF)	12,288 bytes ROM	12,288 bytes FLASH
User vectors ($FFDA–$FFFF)	38 bytes ROM	38 bytes FLASH
Oscillator selection (Register at $FF80)	Mask option register; defined by mask; read only. $FF80 — MOR	Mask option register; defined by programming FLASH location $FF80. $FF80 — MOR
Registers at $FE08 and $FF09	Not used; locations are reserved.	FLASH related registers. $FE08 — FLCR $FF09 — FLBPR
Monitor ROM ($FE10–$FF7F)	Used for testing purposes only.	Used for testing and FLASH programming/erasing.

A.3 MCU Block Diagram

Figure A-1 shows the block diagram of the MC68HC08SR12.

A.4 Memory Map

The MC68HC08SR12 has 12,288 bytes of user ROM from $C000 to $EFFF, and 38 bytes of user ROM vectors from $FFDA to $FFFF. On the MC68HC908SR12, these memory locations are FLASH memory.

Figure A-2 shows the memory map of the MC68HC08SR12

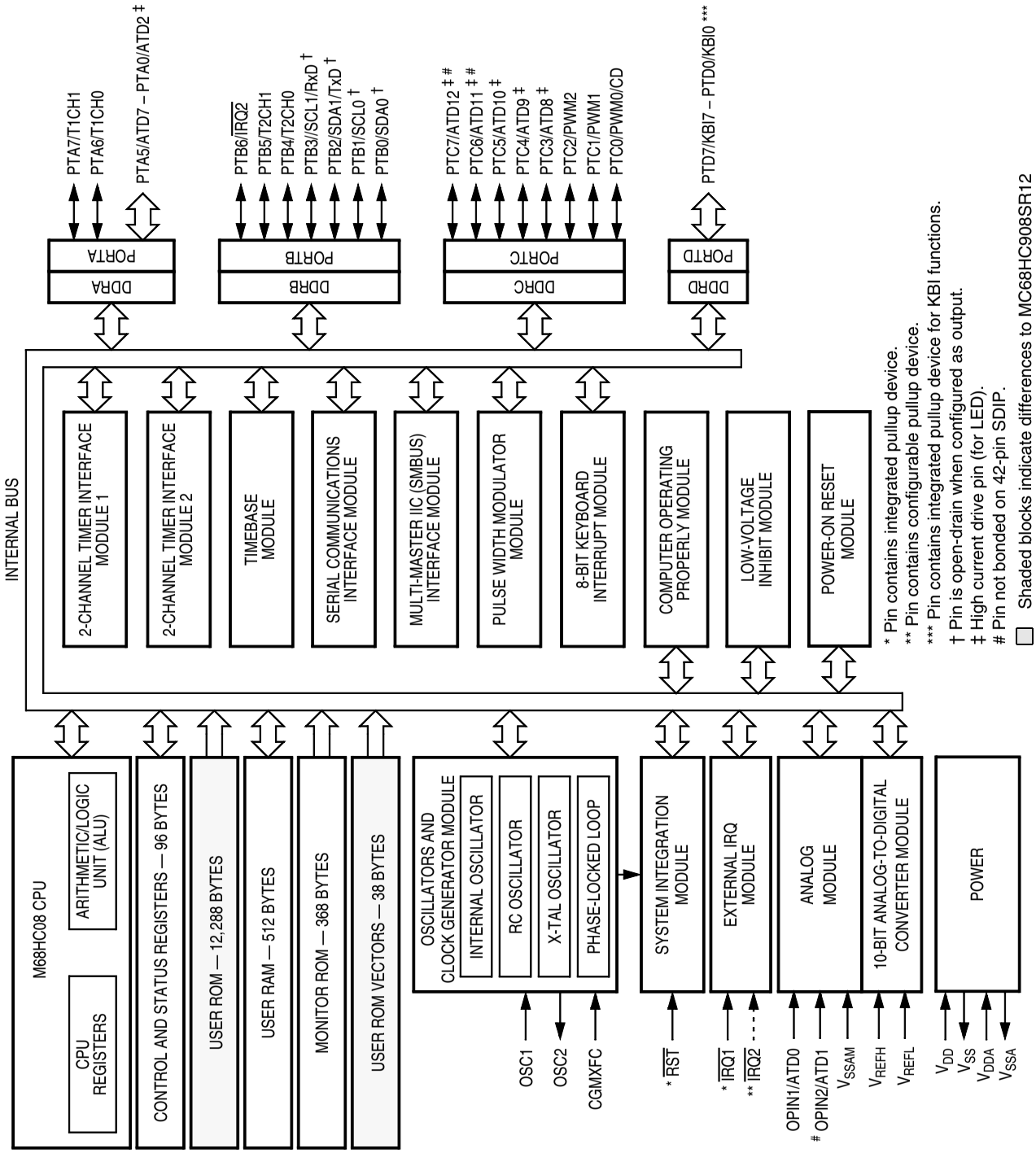


Figure A-1. MC68HC08SR12 Block Diagram

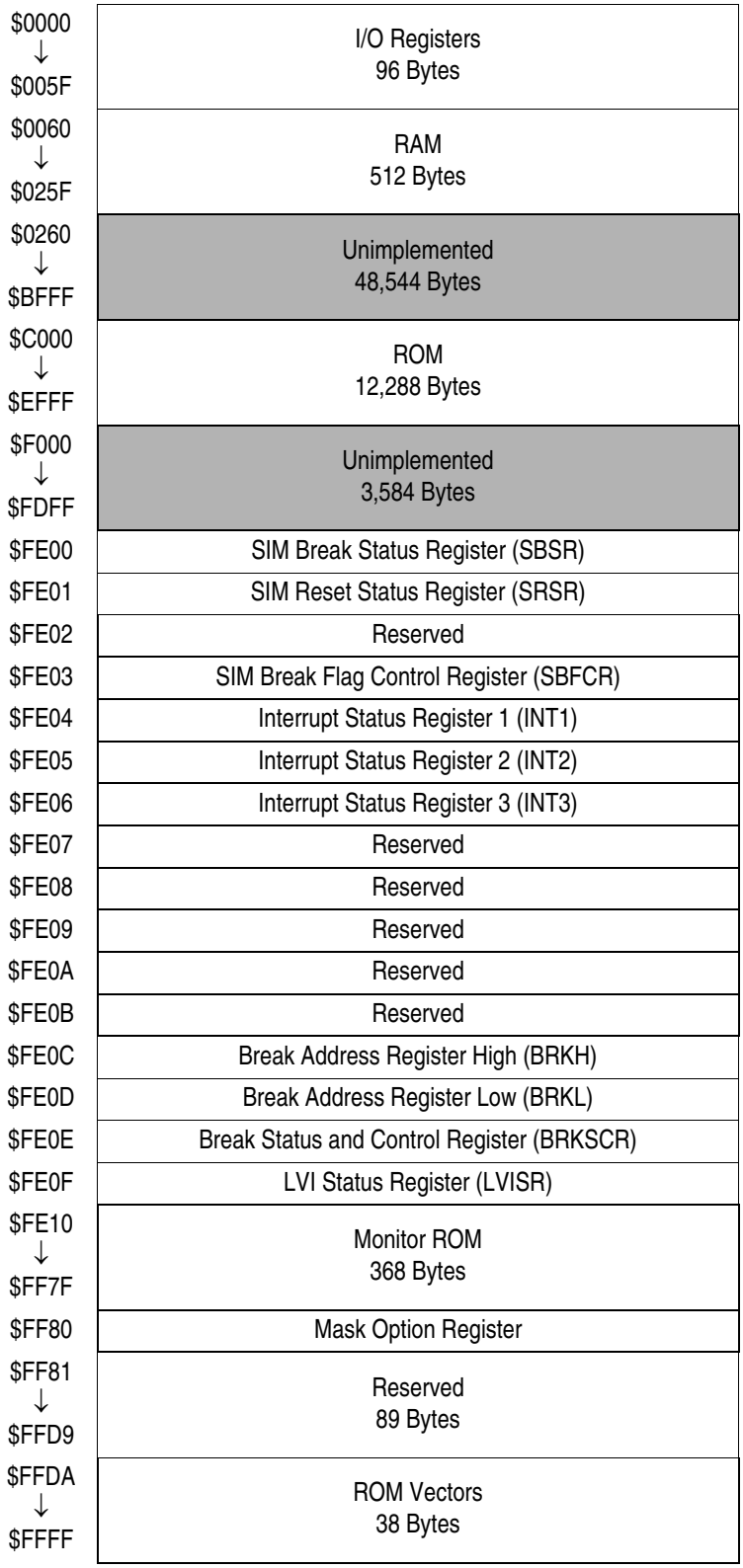


Figure A-2. MC68HC08SR12 Memory Map

A.5 Mask Option Register

The mask option register (MOR) is used for selecting one of the three clock options for the MCU. The MOR at $FF80 is a read-only register on the MC68HC08SR12. It is defined by a mask option (hard-wired connection) specified at the same time as the ROM code submission.

On the MC68HC908SR12, the MOR is a byte located in FLASH memory, and is written by a FLASH programming routine.

A.6 Reserved Registers

The two registers at $FE08 and $FF09 are reserved locations on the MC68HC08SR12.

On the MC68HC908SR12, these two locations are the FLASH control register and the FLASH block protect register respectively.

A.7 Monitor ROM

The monitor program (monitor ROM, $FE10–$FF7F) on the MC68HC08SR12 is for device testing only.

A.8 Electrical Specifications

Electrical specifications for the MC68HC908SR12 apply to the MC68HC08SR12, except for the parameters indicated below.

A.8.1 5.0V DC Electrical Characteristics

Table A-2. 5V DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage (I_{LOAD} = –2.0 mA) PTA[0:7], PTB[4:6], PTC[0:7], PTD[0:7]	V_{OH}	V_{DD}–0.8	—	—	V
Output low voltage (I_{LOAD} = 1.6mA) PTA[0:7], PTB[0:6], PTC[0:7], PTD[0:7]	V_{OL}	—	—	0.4	V
LED sink current (V_{Drain} = 4.0V) PTA[0:5], PTC[3:7]	I_{OL}	—	–15	—	mA
Input high voltage PTA[0:7], PTB[0:6], PTC[0:7], PTD[0:7], $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, OSC1.	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage PTA[0:7], PTB[0:6], PTC[0:7], PTD[0:7], $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, OSC1	V_{IL}	V_{SS}	—	0.3 × V_{DD}	V
V_{DD} supply current Run[3], f_{OP} = 8.0 MHz with ADC on with ADC off Wait[4], f_{OP} = 8.0 MHz Stop[5] 25°C (with OSC, TBM, current sense, LVI) 25°C (with OSC, TBM, current sense) 25°C (with OSC, TBM) 25°C –40°C to 85°C (with OSC, TBM, current sense, LVI) –40°C to 85°C (with OSC, TBM, current sense) –40°C to 85°C (with OSC, TBM) –40°C to 85°C	I_{DD}	 — — — — — — — — — — —	 24 18 7.5 50 12 9 2 — — — —	 40 30 15 150 40 30 10 180 50 40 15	 mA mA mA µA µA µA µA µA µA µA µA
Digital I/O ports Hi-Z leakage current	I_{IL}	—	—	± 10	µA
Input current	I_{IN}	—	—	± 1	µA
Capacitance Ports (as input or output)	C_{OUT} C_{IN}	— —	— —	12 8	pF
POR re-arm voltage[6]	V_{POR}	0	—	100	mV
POR rise-time ramp rate[7]	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{TST}	1.4 × V_{DD}	—	8	V

Table A-2. 5V DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Pullup resistors[8] PTD[0:7] configured as KBI[0:7] $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$	 R_{PU1} R_{PU2}	 24 24	 35 35	 42 42	 kΩ kΩ
Low-voltage inhibit, trip falling voltage	V_{LVI5}	3.80	4.15	4.45	V
Low-voltage inhibit, trip rising voltage	V_{LVI5}	3.95	4.30	4.60	V
Schmitt trigger input low level trip voltage $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, KBI[0:7]	V_{SCMTL}	—	1.21	—	V
Schmitt trigger input high level trip voltage $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, KBI[0:7]	V_{SCMTH}	—	1.65	—	V

Notes:
1. V_{DD} = 4.5 to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
3. Run (operating) I_{DD} measured using external square wave clock source. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD}. Measured with all modules enabled.
4. Wait I_{DD} measured using external square wave clock source. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects wait I_{DD}.
5. STOP I_{DD} measured with OSC1 grounded, no port pins sourcing current.
6. Maximum is highest voltage that POR is guaranteed.
7. If minimum V_{DD} is not reached before the internal POR reset is released, $\overline{RST}$ must be driven low externally until minimum V_{DD} is reached.
8. R_{PU1} and R_{PU2} are measured at V_{DD} = 5.0V

A.8.2 3.0V DC Electrical Characteristics

Table A-3. 3V DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage (I_{LOAD} = −1.0mA) PTA[0:7], PTB[4:6], PTC[0:7], PTD[0:7]	V_{OH}	V_{DD}−0.4	—	—	V
Output low voltage (I_{LOAD} = 0.8mA) PTA[0:7], PTB[0:6], PTC[0:7], PTD[0:7]	V_{OL}	—	—	0.4	V
LED sink current (V_{Drain} = 2.0V) PTA[0:5], PTC[3:7]	V_{OL}	—	−5	—	mA
Input high voltage PTA[0:7], PTB[0:6], PTC[0:7], PTD[0:7], $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage PTA[0:7], PTB[0:6], PTC[0:7], PTD[0:7], $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, OSC1	V_{IL}	V_{SS}	—	0.3 × V_{DD}	V

Table A-3. 3V DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
V_{DD} supply current					
Run[3], f_{OP} = 4.0 MHz					
with ADC on	I_{DD}	—	7	16	mA
with ADC off		—	5	12	mA
Wait[4], f_{OP} = 4.0 MHz		—	2.5	6	mA
Stop[5]					
25°C (with OSC, TBM, LVI)		—	27	80	µA
25°C (with OSC, TBM)		—	5	12	µA
25°C		—	1.5	5	µA
−40°C to 85°C (with OSC, TBM, LVI)		—	—	100	µA
−40°C to 85°C (with OSC, TBM)		—	—	15	µA
−40°C to 85°C		—	—	5	µA
Digital I/O ports Hi-Z leakage current	I_{IL}	—	—	± 10	µA
Input current	I_{IN}	—	—	± 1	µA
Capacitance	C_{OUT}	—	—	12	pF
Ports (as input or output)	C_{IN}	—	—	8	
POR re-arm voltage[6]	V_{POR}	0	—	100	mV
POR rise-time ramp rate[7]	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{HI}	1.4 × V_{DD}	—	2.0 × V_{DD}	V
Pullup resistors[8]					
PTD[0:7] configured as KBI[0:7]	R_{PU1}	24	35	42	kΩ
$\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$	R_{PU2}	24	35	42	kΩ
Low-voltage inhibit, trip voltage (No hysteresis implemented for 3V LVI)	V_{LVI3}	2.32	2.49	2.68	V
Schmitt trigger input low level trip voltage $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, KBI[0:7]	V_{SCMTL}	—	0.8	—	V
Schmitt trigger input high level trip voltage $\overline{RST}$, $\overline{IRQ1}$, $\overline{IRQ2}$, KBI[0:7]	V_{SCMTH}	—	1.2	—	V

Notes:
1. V_{DD} = 2.7 to 3.3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
3. Run (operating) I_{DD} measured using external square wave clock source. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD}. Measured with all modules enabled.
4. Wait I_{DD} measured using external square wave clock source. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects wait I_{DD}.
5. STOP I_{DD} measured with OSC1 grounded, no port pins sourcing current.
6. Maximum is highest voltage that POR is guaranteed.
7. If minimum V_{DD} is not reached before the internal POR reset is released, $\overline{RST}$ must be driven low externally until minimum V_{DD} is reached.
8. R_{PU1} and R_{PU2} are measured at V_{DD} = 5.0V.

A.8.3 Memory Characteristics

Characteristic	Symbol	Min	Max	Unit
RAM data retention voltage	V_{RDR}	1.3	—	V

Notes:
Since MC68HC08SR12 is a ROM device, FLASH memory electrical characteristics do not apply.

A.9 ROM Order Numbers

These part numbers are generic numbers only. To place an order, ROM code must be submitted to the ROM Processing Center (RPC).

Table A-4. MC68HC08SR12 Order Numbers

MC order number	Operating temperature range	Package
MC68HC08SR12CB	–40 to +85 °C	42-Pin SDIP(1)
MC68HC08SR12CFA	–40 to +85 °C	48-pin LQFP(2)

Notes:
1. SDIP = Shrink Dual In-Line Package.
2. LQFP = Low Quad Flat Pack.

How to Reach Us:

USA/Europe/Locations Not Listed:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

Japan:
Freescale Semiconductor Japan Ltd.
Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-8573, Japan
81-3-3440-3569

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
852-26668334

Home Page:
www.freescale.com

RoHS-compliant and/or Pb- free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb- free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale.s Environmental Products program, go to http://www.freescale.com/epp.



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

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

Learn More: For more information about Freescale products, please visit www.freescale.com

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

© Freescale Semiconductor, Inc. 2004

MC68HC908SR12
Rev. 5
07/2004