

MC68HC908BD48

Data Sheet

M68HC08
Microcontrollers

Rev. 2.1
MC68HC908BD48/D
August 1, 2005

freescale.com



MC68HC908BD48

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://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, Inc., 2003

Revision History

| Date | Revision Level | Description | Page Number(s) |
|----------|----------------|---|----------------|
| 9/2003 | 2 | Incorporated addendum to rev. 1.0 data sheet. | |
| | | Removed 28-pin PDIP references. | |
| | | Section 10. Timer Interface Module (TIM) — Timer discrepancies corrected throughout this section. | 125 |
| | | Section 22. Mechanical Specifications — Replaced incorrect 44-pin QFP drawing, case 824E to case 824A. | 283 |
| | | Section 9. Monitor ROM (MON) — Updated Figure 9-1 . Monitor Mode Circuit . | 117 |
| 8/1/2005 | 2.1 | Updated to meet Freescale identity guidelines. | |

List of Sections

| | |
|--|-----|
| Section 1. General Description | 25 |
| Section 2. Memory Map | 35 |
| Section 3. Random-Access Memory (RAM) | 53 |
| Section 4. FLASH Memory | 55 |
| Section 5. Configuration Register (CONFIG) | 63 |
| Section 6. Central Processor Unit (CPU) | 67 |
| Section 7. System Integration Module (SIM) | 87 |
| Section 8. Oscillator (OSC) | 111 |
| Section 9. Monitor ROM (MON) | 115 |
| Section 10. Timer Interface Module (TIM) | 125 |
| Section 11. Pulse Width Modulator (PWM) | 147 |
| Section 12. Analog-to-Digital Converter (ADC) | 153 |
| Section 13. Universal Serial Bus Module (USB) | 163 |
| Section 14. Multi-Master IIC Interface (MMIIC) | 179 |
| Section 15. DDC12AB Interface | 193 |
| Section 16. Sync Processor | 209 |
| Section 17. Input/Output (I/O) Ports | 229 |
| Section 18. External Interrupt (IRQ) | 251 |
| Section 19. Computer Operating Properly (COP) | 257 |
| Section 20. Break Module (BRK) | 263 |
| Section 21. Electrical Specifications | 271 |
| Section 22. Mechanical Specifications | 283 |
| Section 23. Ordering Information | 285 |



Table of Contents

Section 1. General Description

| | | |
|-----|-----------------------------|----|
| 1.1 | Contents | 25 |
| 1.2 | Introduction | 25 |
| 1.3 | Features | 25 |
| 1.4 | MCU Block Diagram | 27 |
| 1.5 | Pin Assignments | 29 |
| 1.6 | Pin Functions | 31 |

Section 2. Memory Map

| | | |
|-----|--|----|
| 2.1 | Contents | 35 |
| 2.2 | Introduction | 35 |
| 2.3 | Unimplemented Memory Locations | 35 |
| 2.4 | Reserved Memory Locations | 36 |
| 2.5 | Input/Output (I/O) Section | 36 |

Section 3. Random-Access Memory (RAM)

| | | |
|-----|----------------------------------|----|
| 3.1 | Contents | 53 |
| 3.2 | Introduction | 53 |
| 3.3 | Functional Description | 53 |

Section 4. FLASH Memory

| | | |
|-----|------------------------|----|
| 4.1 | Contents | 55 |
| 4.2 | Introduction | 55 |

| | | |
|-----|--|----|
| 4.3 | Functional Description | 56 |
| 4.4 | FLASH Control Register (FLCR) | 56 |
| 4.5 | FLASH Block Erase Operation | 57 |
| 4.6 | FLASH Mass Erase Operation | 58 |
| 4.7 | FLASH Program Operation | 59 |
| 4.8 | FLASH Block Protection | 61 |
| 4.9 | FLASH Block Protect Register (FLBPR) | 61 |

Section 5. Configuration Register (CONFIG)

| | | |
|-------|------------------------------------|----|
| 5.1 | Contents | 63 |
| 5.2 | Introduction | 63 |
| 5.3 | Functional Description | 63 |
| 5.3.1 | Configuration Register 0 | 64 |
| 5.3.2 | Configuration Register 1 | 65 |

Section 6. Central Processor Unit (CPU)

| | | |
|-------|---------------------------------------|----|
| 6.1 | Contents | 67 |
| 6.2 | Introduction | 67 |
| 6.3 | Features | 68 |
| 6.4 | CPU Registers | 68 |
| 6.4.1 | Accumulator | 69 |
| 6.4.2 | Index Register | 70 |
| 6.4.3 | Stack Pointer | 70 |
| 6.4.4 | Program Counter | 71 |
| 6.4.5 | Condition Code Register | 71 |
| 6.5 | Arithmetic/Logic Unit (ALU) | 74 |
| 6.6 | Low-Power Modes | 74 |
| 6.6.1 | Wait Mode | 74 |
| 6.6.2 | Stop Mode | 75 |
| 6.7 | CPU During Break Interrupts | 75 |

| | | |
|---|---|-----|
| 6.8 | Instruction Set Summary | 75 |
| 6.9 | Opcode Map | 75 |
| Section 7. System Integration Module (SIM) | | |
| 7.1 | Contents | 87 |
| 7.2 | Introduction | 88 |
| 7.3 | SIM Bus Clock Control and Generation | 91 |
| 7.3.1 | Bus Timing | 91 |
| 7.3.2 | Clock Start-Up from POR | 91 |
| 7.3.3 | Clocks in Stop Mode and Wait Mode | 91 |
| 7.4 | Reset and System Initialization | 92 |
| 7.4.1 | External Pin Reset | 92 |
| 7.4.2 | Active Resets from Internal Sources | 93 |
| 7.4.2.1 | Power-On Reset | 94 |
| 7.4.2.2 | Computer Operating Properly (COP) Reset | 95 |
| 7.4.2.3 | Illegal Opcode Reset | 96 |
| 7.4.2.4 | Illegal Address Reset | 96 |
| 7.5 | SIM Counter | 96 |
| 7.5.1 | SIM Counter During Power-On Reset | 96 |
| 7.5.2 | SIM Counter During Stop Mode Recovery | 97 |
| 7.5.3 | SIM Counter and Reset States | 97 |
| 7.6 | Exception Control | 97 |
| 7.6.1 | Interrupts | 98 |
| 7.6.1.1 | Hardware Interrupts | 100 |
| 7.6.1.2 | SWI Instruction | 101 |
| 7.6.2 | Interrupt Status Registers | 101 |
| 7.6.2.1 | Interrupt Status Register 1 | 103 |
| 7.6.2.2 | Interrupt Status Register 2 | 103 |
| 7.6.3 | Reset | 104 |
| 7.6.4 | Break Interrupts | 104 |
| 7.6.5 | Status Flag Protection in Break Mode | 104 |
| 7.7 | Low-Power Modes | 105 |
| 7.7.1 | Wait Mode | 105 |
| 7.7.2 | Stop Mode | 106 |

| | | |
|-------|---|-----|
| 7.8 | SIM Registers | 108 |
| 7.8.1 | SIM Break Status Register (SBSR) | 108 |
| 7.8.2 | SIM Reset Status Register (SRSR) | 109 |
| 7.8.3 | SIM Break Flag Control Register (SBFCR) | 110 |

Section 8. Oscillator (OSC)

| | | |
|-------|---|-----|
| 8.1 | Contents | 111 |
| 8.2 | Introduction | 111 |
| 8.3 | Oscillator External Connections | 112 |
| 8.4 | I/O Signals | 113 |
| 8.4.1 | Crystal Amplifier Input Pin (OSC1) | 113 |
| 8.4.2 | Crystal Amplifier Output Pin (OSC2) | 113 |
| 8.4.3 | Oscillator Enable Signal (SIMOSCEN) | 113 |
| 8.4.4 | External Clock Source (OSCXCLK) | 113 |
| 8.4.5 | Oscillator Out (OSCOUT) | 113 |
| 8.5 | Low Power Modes | 114 |
| 8.5.1 | Wait Mode | 114 |
| 8.5.2 | Stop Mode | 114 |
| 8.6 | Oscillator During Break Mode | 114 |

Section 9. Monitor ROM (MON)

| | | |
|-------|----------------------------------|-----|
| 9.1 | Contents | 115 |
| 9.2 | Introduction | 115 |
| 9.3 | Features | 116 |
| 9.4 | Functional Description | 116 |
| 9.4.1 | Entering Monitor Mode | 118 |
| 9.4.2 | Data Format | 119 |
| 9.4.3 | Echoing | 120 |
| 9.4.4 | Break Signal | 120 |
| 9.4.5 | Commands | 121 |
| 9.4.6 | Baud Rate | 124 |

Section 10. Timer Interface Module (TIM)

| | | |
|----------|--|-----|
| 10.1 | Contents | 125 |
| 10.2 | Introduction | 126 |
| 10.3 | Features | 126 |
| 10.4 | Pin Name Conventions | 126 |
| 10.5 | Functional Description | 127 |
| 10.5.1 | TIM Counter Prescaler | 129 |
| 10.5.2 | Input Capture | 129 |
| 10.5.3 | Output Compare | 129 |
| 10.5.3.1 | Unbuffered Output Compare | 130 |
| 10.5.3.2 | Buffered Output Compare | 131 |
| 10.5.4 | Pulse Width Modulation (PWM) | 131 |
| 10.5.4.1 | Unbuffered PWM Signal Generation | 132 |
| 10.5.4.2 | Buffered PWM Signal Generation | 133 |
| 10.5.4.3 | PWM Initialization | 134 |
| 10.6 | Interrupts | 135 |
| 10.7 | Wait Mode | 135 |
| 10.8 | TIM During Break Interrupts | 136 |
| 10.9 | I/O Signals | 136 |
| 10.10 | I/O Registers | 137 |
| 10.10.1 | TIM Status and Control Register (TSC) | 137 |
| 10.10.2 | TIM Counter Registers (TCNTH:TCNTL) | 139 |
| 10.10.3 | TIM Counter Modulo Registers (TMODH:TMODL) | 140 |
| 10.10.4 | TIM Channel Status and Control Registers (TSC0:TSC1) | 141 |
| 10.10.5 | TIM Channel Registers (TCH0H/L:TCH1H/L) | 145 |

Section 11. Pulse Width Modulator (PWM)

| | | |
|------|------------------------------|-----|
| 11.1 | Contents | 147 |
| 11.2 | Introduction | 147 |
| 11.3 | Functional Description | 147 |
| 11.4 | PWM Registers | 149 |

11.4.1 PWM Data Registers 0 to 15 (0PWM–15PWM) 150
 11.4.2 PWM Control Registers 1 and 2 (PWMCR1:PWMCR2) . . 151

Section 12. Analog-to-Digital Converter (ADC)

12.1 Contents 153
 12.2 Introduction 153
 12.3 Features 154
 12.4 Functional Description 154
 12.4.1 ADC Port I/O Pins 155
 12.4.2 Voltage Conversion 156
 12.4.3 Conversion Time 156
 12.4.4 Continuous Conversion 156
 12.4.5 Accuracy and Precision 157
 12.5 Interrupts 157
 12.6 Low-Power Modes 157
 12.6.1 Wait Mode 157
 12.6.2 Stop Mode 157
 12.7 I/O Signals 157
 12.7.1 ADC Voltage In (ADCVIN) 158
 12.8 I/O Registers 158
 12.8.1 ADC Status and Control Register 158
 12.8.2 ADC Data Register 160
 12.8.3 ADC Input Clock Register 161

Section 13. Universal Serial Bus Module (USB)

13.1 Contents 163
 13.2 Introduction 163
 13.3 Features 164
 13.4 I/O Pins 164
 13.5 Registers 165
 13.5.1 USB Address Register (UADR) 166

| | | |
|--------|---|-----|
| 13.5.2 | USB Interrupt Register (UINTR) | 166 |
| 13.5.3 | USB Interrupt Register 1 (UIR1) | 169 |
| 13.5.4 | USB Control Register 0 (UCR0) | 171 |
| 13.5.5 | USB Control Register 1 (UCR1) | 172 |
| 13.5.6 | USB Control Register 2 (UCR2) | 174 |
| 13.5.7 | USB Status Register (USR) | 175 |
| 13.5.8 | USB Endpoint 0 Data Registers 0 to 7 (UD0R0–UD0R7) | 177 |
| 13.5.9 | USB Endpoint 1/2 Data Registers 0 to 7 (UD1R0–UD1R7) | 177 |

Section 14. Multi-Master IIC Interface (MMIIC)

| | | |
|--------|--|-----|
| 14.1 | Contents | 179 |
| 14.2 | Introduction | 179 |
| 14.3 | Features | 180 |
| 14.4 | I/O Pins | 180 |
| 14.5 | Registers | 181 |
| 14.5.1 | Multi-Master IIC Address Register (MMADR) | 182 |
| 14.5.2 | Multi-Master IIC Control Register (MMCR) | 183 |
| 14.5.3 | Multi-Master IIC Master Control Register (MIMCR) | 184 |
| 14.5.4 | Multi-Master IIC Status Register (MMSR) | 186 |
| 14.5.5 | Multi-Master IIC Data Transmit Register (MMDTR) | 188 |
| 14.5.6 | Multi-Master IIC Data Receive Register (MMDRR) | 189 |
| 14.6 | Programming Considerations | 190 |

Section 15. DDC12AB Interface

| | | |
|------|-------------------------|-----|
| 15.1 | Contents | 193 |
| 15.2 | Introduction | 193 |
| 15.3 | Features | 194 |
| 15.4 | I/O Pins | 194 |
| 15.5 | DDC Protocols | 196 |
| 15.6 | Registers | 196 |

| | | |
|--------|--|-----|
| 15.6.1 | DDC Address Register (DADR) | 196 |
| 15.6.2 | DDC2 Address Register (D2ADR) | 197 |
| 15.6.3 | DDC Control Register (DCR) | 198 |
| 15.6.4 | DDC Master Control Register (DMCR) | 199 |
| 15.6.5 | DDC Status Register (DSR) | 202 |
| 15.6.6 | DDC Data Transmit Register (DDTR) | 204 |
| 15.6.7 | DDC Data Receive Register (DDRR) | 205 |
| 15.7 | Programming Considerations | 206 |

Section 16. Sync Processor

| | | |
|----------|---|-----|
| 16.1 | Contents | 209 |
| 16.2 | Introduction | 209 |
| 16.3 | Features | 211 |
| 16.4 | I/O Pins | 211 |
| 16.5 | Functional Blocks | 213 |
| 16.5.1 | Polarity Detection | 214 |
| 16.5.1.1 | Hsync Polarity Detection | 214 |
| 16.5.1.2 | Vsync Polarity Detection | 214 |
| 16.5.1.3 | Composite Sync Polarity Detection | 214 |
| 16.5.2 | Sync Signal Counters | 215 |
| 16.5.3 | Polarity Controlled HSYNCO and VSYNCO Outputs | 215 |
| 16.5.4 | Clamp Pulse Output | 216 |
| 16.5.5 | Low Vertical Frequency Detect | 217 |
| 16.6 | Registers | 217 |
| 16.6.1 | Sync Processor Control & Status Register (SPCSR) | 217 |
| 16.6.2 | Sync Processor Input/Output Control Register (SPIOCR) | 219 |
| 16.6.3 | Vertical Frequency Registers (VFRs) | 221 |
| 16.6.4 | Hsync Frequency Registers (HFRs) | 223 |
| 16.6.5 | Sync Processor Control Register 1 (SPCR1) | 225 |
| 16.6.6 | H&V Sync Output Control Register (HVOCR) | 226 |
| 16.7 | System Operation | 227 |

Section 17. Input/Output (I/O) Ports

| | | |
|--------|---------------------------------|-----|
| 17.1 | Contents | 229 |
| 17.2 | Introduction | 229 |
| 17.3 | Port A | 233 |
| 17.3.1 | Port A Data Register | 233 |
| 17.3.2 | Data Direction Register A | 234 |
| 17.3.3 | Port A Options | 235 |
| 17.4 | Port B | 236 |
| 17.4.1 | Port B Data Register | 236 |
| 17.4.2 | Data Direction Register B | 237 |
| 17.4.3 | Port B Options | 238 |
| 17.5 | Port C | 239 |
| 17.5.1 | Port C Data Register | 239 |
| 17.5.2 | Data Direction Register C | 240 |
| 17.5.3 | Port C Options | 241 |
| 17.6 | Port D | 242 |
| 17.6.1 | Port D Data Register | 242 |
| 17.6.2 | Data Direction Register D | 243 |
| 17.6.3 | Port D Options | 245 |
| 17.7 | Port E | 247 |
| 17.7.1 | Port E Data Register | 247 |
| 17.7.2 | Data Direction Register E | 248 |
| 17.7.3 | Port E Options | 250 |

Section 18. External Interrupt (IRQ)

| | | |
|------|--|-----|
| 18.1 | Contents | 251 |
| 18.2 | Introduction | 251 |
| 18.3 | Features | 251 |
| 18.4 | Functional Description | 252 |
| 18.5 | $\overline{\text{IRQ}}$ Pin | 254 |
| 18.6 | IRQ Module During Break Interrupts | 255 |
| 18.7 | IRQ Status and Control Register | 255 |

Section 19. Computer Operating Properly (COP)

19.1 Contents257

19.2 Introduction257

19.3 Functional Description258

19.4 I/O Signals259

19.4.1 OSCXCLK259

19.4.2 STOP Instruction259

19.4.3 COPCTL Write259

19.4.4 Power-On Reset259

19.4.5 Internal Reset260

19.4.6 Reset Vector Fetch260

19.4.7 COPD (COP Disable)260

19.4.8 COPRS (COP Rate Select)260

19.5 COP Control Register261

19.6 Interrupts261

19.7 Monitor Mode261

19.8 Low-Power Modes261

19.8.1 Wait Mode262

19.8.2 Stop Mode262

19.9 COP Module During Break Mode262

Section 20. Break Module (BRK)

20.1 Contents263

20.2 Introduction263

20.3 Features264

20.4 Functional Description264

20.4.1 Flag Protection During Break Interrupts266

20.4.2 CPU During Break Interrupts266

20.4.3 TIM During Break Interrupts266

20.4.4 COP During Break Interrupts266

20.5 Low-Power Modes266

| | | |
|--------|---|-----|
| 20.5.1 | Wait Mode | 266 |
| 20.5.2 | Stop Mode | 267 |
| 20.6 | Break Module Registers | 267 |
| 20.6.1 | Break Status and Control Register | 267 |
| 20.6.2 | Break Address Registers | 268 |
| 20.6.3 | SIM Break Status Register | 268 |
| 20.6.4 | SIM Break Flag Control Register | 270 |

Section 21. Electrical Specifications

| | | |
|---------|---|-----|
| 21.1 | Contents | 271 |
| 21.2 | Introduction | 271 |
| 21.3 | Absolute Maximum Ratings | 272 |
| 21.4 | Functional Operating Range | 273 |
| 21.5 | Thermal Characteristics | 273 |
| 21.6 | DC Electrical Characteristics | 274 |
| 21.7 | Control Timing | 275 |
| 21.8 | Oscillator Characteristics | 275 |
| 21.9 | ADC Characteristics | 276 |
| 21.10 | USB DC Electrical Characteristics | 277 |
| 21.11 | USB Low Speed Source Electrical Characteristics | 278 |
| 21.12 | Timer Interface Module Characteristics | 278 |
| 21.13 | DDC12AB/MMIIC Timing | 279 |
| 21.13.1 | DDC12AB/MMIIC Interface Input Signal Timing | 279 |
| 21.13.2 | DDC12AB/MMIIC Interface Output Signal Timing | 279 |
| 21.14 | Sync Processor Timing | 280 |
| 21.15 | Memory Characteristics | 281 |

Section 22. Mechanical Specifications

22.1 Contents283

22.2 Introduction283

22.4 42-Pin Shrink Dual in-Line Package (SDIP)283

22.5 44-Pin Plastic Quad Flat Pack (QFP)284

Section 23. Ordering Information

23.1 Contents285

23.2 Introduction285

23.3 MC Order Numbers285

List of Figures

| Figure | Title | Page |
|--------|--|------|
| 1-1 | MCU Block Diagram | 28 |
| 1-2 | 44-Pin QFP Pin Assignments | 29 |
| 1-3 | 42-Pin SDIP Pin Assignments | 30 |
| 2-1 | Memory Map | 37 |
| 2-2 | Control, Status, and Data Registers | 39 |
| 4-1 | FLASH Control Register (FLCR) | 56 |
| 4-2 | FLASH Programming Flowchart | 60 |
| 4-3 | FLASH Block Protect Register (FLBPR) | 61 |
| 4-4 | FLASH Block Protect Start Address | 62 |
| 5-1 | Configuration Register 0 (CONFIG0) | 64 |
| 5-2 | Configuration Register 1 (CONFIG1) | 65 |
| 6-1 | CPU Registers | 69 |
| 6-2 | Accumulator (A) | 69 |
| 6-3 | Index Register (H:X) | 70 |
| 6-4 | Stack Pointer (SP) | 71 |
| 6-5 | Program Counter (PC) | 71 |
| 6-6 | Condition Code Register (CCR) | 72 |
| 7-1 | SIM Block Diagram | 89 |
| 7-2 | OSC Clock Signals | 91 |
| 7-3 | External Reset Timing | 93 |
| 7-4 | Internal Reset Timing | 93 |
| 7-5 | Sources of Internal Reset | 94 |
| 7-6 | POR Recovery | 95 |
| 7-7 | Interrupt Entry | 98 |
| 7-8 | Interrupt Recovery | 98 |
| 7-9 | Interrupt Processing | 99 |
| 7-10 | Interrupt Recognition Example | 100 |

| Figure | Title | Page |
|--------|--|------|
| 7-11 | Interrupt Status Register 1 (INT1) | 103 |
| 7-12 | Interrupt Status Register 2 (INT2) | 103 |
| 7-13 | Wait Mode Entry Timing | 105 |
| 7-14 | Wait Recovery from Interrupt or Break | 106 |
| 7-15 | Wait Recovery from Internal Reset | 106 |
| 7-16 | Stop Mode Entry Timing | 107 |
| 7-17 | Stop Mode Recovery from Interrupt or Break | 107 |
| 7-18 | SIM Break Status Register (SBSR) | 108 |
| 7-19 | SIM Reset Status Register (SRSR) | 109 |
| 7-20 | SIM Break Flag Control Register (SBFCR) | 110 |
| | | |
| 8-1 | Oscillator External Connections | 112 |
| | | |
| 9-1 | Monitor Mode Circuit | 117 |
| 9-2 | Monitor Data Format | 119 |
| 9-3 | Sample Monitor Waveforms | 119 |
| 9-4 | Read Transaction | 120 |
| 9-5 | Break Transaction | 120 |
| | | |
| 10-1 | TIM Block Diagram | 127 |
| 10-2 | PWM Period and Pulse Width | 132 |
| 10-3 | TIM Status and Control Register (TSC) | 137 |
| 10-4 | TIM Counter Registers (TCNTH:TCNTL) | 140 |
| 10-5 | TIM Counter Modulo Registers (TMODH:TMODL) | 141 |
| 10-6 | TIM Channel Status and Control Registers (TSC0:TSC1) | 142 |
| 10-7 | CHxMAX Latency | 145 |
| 10-8 | TIM Channel Registers (TCH0H/L:TCH1H/L) | 146 |
| | | |
| 11-1 | PWM Data Registers 0 to 15 (0PWM–15PWM) | 150 |
| 11-2 | PWM Control Register 1 and 2 (PWMCR1:PWMCR2) | 151 |
| 11-3 | 8-Bit PWM Output Waveforms | 152 |
| | | |
| 12-1 | ADC Block Diagram | 155 |
| 12-2 | ADC Status and Control Register (ADSCR) | 158 |
| 12-3 | ADC Data Register (ADR) | 160 |
| 12-4 | ADC Input Clock Register (ADICLK) | 161 |
| | | |
| 13-1 | USB Address Register (UADR) | 166 |
| 13-2 | USB Interrupt Register (UINTR) | 166 |

| Figure | Title | Page |
|--------|--|------|
| 13-3 | USB Interrupt Register 1 (UIR1) | 169 |
| 13-4 | USB Control Register 0 (UCR0) | 171 |
| 13-5 | USB Control Register 1 (UCR1) | 172 |
| 13-6 | USB Control Register 2 (UCR2) | 174 |
| 13-7 | USB Status Register (USR) | 175 |
| 13-8 | USB Endpoint 0 Data Registers 0 to 7 (UD0R0–UD0R7) . . . | 177 |
| 13-9 | USB Endpoint 1 Data Registers 0 to 7 (UD1R0–UD1R7) . . . | 177 |
| | | |
| 14-1 | Multi-Master IIC Address Register (MMADR) | 182 |
| 14-2 | Multi-Master IIC Control Register (MMCR) | 183 |
| 14-3 | Multi-Master IIC Master Control Register (MIMCR) | 184 |
| 14-4 | Multi-Master IIC Status Register (MMSR) | 186 |
| 14-5 | Multi-Master IIC Data Transmit Register (MMDTR) | 188 |
| 14-6 | Multi-Master IIC Data Receive Register (MMDRR) | 189 |
| 14-7 | Data Transfer Sequences for Master/Slave Transmit/Receive Modes | 191 |
| | | |
| 15-1 | DDC Address Register (DADR) | 196 |
| 15-2 | DDC2 Address Register (D2ADR) | 197 |
| 15-3 | DDC Control Register (DCR) | 198 |
| 15-4 | DDC Master Control Register (DMCR) | 199 |
| 15-5 | DDC Status Register (DSR) | 202 |
| 15-6 | DDC Data Transmit Register (DDTR) | 204 |
| 15-7 | DDC Data Receive Register (DDRR) | 205 |
| 15-8 | Data Transfer Sequences for Master/Slave Transmit/Receive Modes | 207 |
| | | |
| 16-1 | Sync Processor Block Diagram | 213 |
| 16-2 | Clamp Pulse Output Timing | 216 |
| 16-3 | Sync Processor Control & Status Register (SPCSR) | 217 |
| 16-4 | Sync Processor Input/Output Control Register (SPIOCR) . . . | 219 |
| 16-5 | Vertical Frequency High Register | 221 |
| 16-6 | Vertical Frequency Low Register | 221 |
| 16-7 | Hsync Frequency High Register | 223 |
| 16-8 | Hsync Frequency Low Register | 223 |
| 16-9 | Sync Processor Control Register 1 (SPCR1) | 225 |
| 16-10 | H&V Sync Output Control Register (HVOCR) | 226 |

| Figure | Title | Page |
|--------|--|------|
| 17-1 | Port A Data Register (PTA) | 233 |
| 17-2 | Data Direction Register A (DDRA) | 234 |
| 17-3 | Port A I/O Circuit. | 234 |
| 17-4 | PWM Control Register 1 (PWMCR1) | 235 |
| 17-5 | Port B Data Register (PTB) | 236 |
| 17-6 | Data Direction Register B (DDRB) | 237 |
| 17-7 | Port B I/O Circuit. | 237 |
| 17-8 | PWM Control Register 1 (PWMCR1) | 238 |
| 17-9 | Port C Data Register (PTC) | 239 |
| 17-10 | Data Direction Register C (DDRC) | 240 |
| 17-11 | Port C I/O Circuit. | 241 |
| 17-12 | Port D Data Register (PTD) | 242 |
| 17-13 | Data Direction Register D (DDRD) | 243 |
| 17-14 | Port D I/O Circuit. | 244 |
| 17-15 | Port D Configuration Register (PDCR) | 245 |
| 17-16 | Port E Data Register (PTE) | 247 |
| 17-17 | Data Direction Register E (DDRE) | 248 |
| 17-18 | Port E I/O Circuit. | 249 |
| 17-19 | Configuration Register 0 (CONFIG0) | 250 |
| 18-1 | IRQ Module Block Diagram | 253 |
| 18-2 | IRQ Status and Control Register (INTSCR) | 256 |
| 19-1 | COP Block Diagram | 258 |
| 19-2 | Configuration Register 1 (CONFIG1) | 260 |
| 19-3 | COP Control Register (COPCTL) | 261 |
| 20-1 | Break Module Block Diagram | 265 |
| 20-2 | Break Status and Control Register (BRKSCR) | 267 |
| 20-3 | Break Address Register High (BRKH) | 268 |
| 20-4 | Break Address Register Low (BRKL) | 268 |
| 20-5 | SIM Break Status Register (SBSR) | 269 |
| 20-6 | SIM Break Flag Control Register (SBFCR) | 270 |
| 21-1 | ADC Input Voltage vs. Step Readings | 277 |
| 22-2 | 42-Pin SDIP (Case #858) | 283 |
| 22-3 | 44-Pin QFP (Case #824A) | 284 |



List of Tables

| Table | Title | Page |
|-------|---|------|
| 1-1 | Pin Functions | 31 |
| 2-1 | Vector Addresses | 51 |
| 6-1 | Instruction Set Summary | 76 |
| 6-2 | Opcode Map | 85 |
| 7-1 | SIM I/O Register Summary | 90 |
| 7-2 | Signal Name Conventions | 90 |
| 7-3 | PIN Bit Set Timing | 92 |
| 7-4 | Interrupt Sources | 102 |
| 7-5 | SIM Registers Summary | 108 |
| 9-1 | Mode Selection | 118 |
| 9-2 | Mode Differences | 119 |
| 9-3 | READ (Read Memory) Command | 121 |
| 9-4 | WRITE (Write Memory) Command | 122 |
| 9-5 | IREAD (Indexed Read) Command | 122 |
| 9-6 | IWRITE (Indexed Write) Command | 123 |
| 9-7 | READSP (Read Stack Pointer) Command | 123 |
| 9-8 | RUN (Run User Program) Command | 124 |
| 9-9 | Monitor Baud Rate Selection | 124 |
| 10-1 | Pin Name Conventions | 126 |
| 10-2 | TIM I/O Register Summary | 128 |
| 10-3 | Prescaler Selection | 139 |
| 10-4 | Mode, Edge, and Level Selection | 144 |
| 11-1 | PWM I/O Register Summary | 148 |
| 11-2 | PWM Channels and Port I/O pins | 151 |

| Table | Title | Page |
|-------|--|------|
| 12-1 | ADC Register Summary | 154 |
| 12-2 | MUX Channel Select | 160 |
| 12-3 | ADC Clock Divide Ratio | 161 |
| 13-1 | Pin Name Conventions | 164 |
| 13-2 | USB I/O Register Summary | 165 |
| 14-1 | Pin Name Conventions | 180 |
| 14-2 | MMIIC I/O Register Summary | 181 |
| 14-3 | Baud Rate Select | 186 |
| 15-1 | Pin Name Conventions | 194 |
| 15-2 | DDC I/O Register Summary | 195 |
| 15-3 | Baud Rate Select | 201 |
| 16-1 | Pin Name Conventions | 211 |
| 16-2 | Sync Processor I/O Register Summary | 212 |
| 16-3 | Sync Output Control | 215 |
| 16-4 | Sync Output Polarity | 216 |
| 16-5 | ATPOL, VINVO, and HINVO setting | 219 |
| 16-6 | Sample Vertical Frame Frequencies | 222 |
| 16-7 | Clamp Pulse Width | 223 |
| 16-8 | HSYNC Polarity Detection Pulse Width | 225 |
| 16-9 | ATPOL, VINVO, and HINVO setting | 226 |
| 16-10 | Free-Running HSYNC and VSYNC Options | 227 |
| 17-1 | I/O Port Register Summary | 230 |
| 17-2 | Port Control Register Bits Summary | 232 |
| 17-3 | Port A Pin Functions | 235 |
| 17-4 | Port B Pin Functions | 238 |
| 17-5 | Port C Pin Functions | 241 |
| 17-6 | Port D Pin Functions | 244 |
| 17-7 | Port E Pin Functions | 249 |
| 18-1 | IRQ I/O Register Summary | 253 |
| 20-1 | Break Module I/O Register Summary | 265 |
| 23-1 | MC Order Numbers | 285 |

Section 1. General Description

1.1 Contents

| | | |
|-----|-----------------------------|----|
| 1.2 | Introduction | 25 |
| 1.3 | Features | 25 |
| 1.4 | MCU Block Diagram | 27 |
| 1.5 | Pin Assignments | 29 |
| 1.6 | Pin Functions | 31 |

1.2 Introduction

The MC68HC908BD48 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.

With special modules such as the sync processor, analog-to-digital converter, pulse modulator module, DDC12AB interface, multi-master IIC interface, and universal serial bus interface, the MC68HC908BD48 is designed specifically for use in digital monitor systems.

1.3 Features

Features of the MC68HC908BD48 MCU include the following:

- High-performance M68HC08 architecture
- Fully upward-compatible object code with M6805, M146805, and M68HC05 families

- Low-power design; fully static with stop and wait modes
- 5V operating voltage
- 6MHz internal bus frequency; with 24MHz external crystal
- 48,128 bytes of on-chip FLASH memory
- 1,024 bytes of on-chip random access memory (RAM)
- Sync signal processor with the following features:
 - Horizontal and vertical frequency counters
 - Low vertical frequency indicator (40.7Hz)
 - Polarity controlled Hsync and Vsync outputs from separate sync or composite sync inputs
 - Internal generated free-running Hsync and Vsync pulses
 - CLAMP pulse output to the external pre-amp chip
- 6-channel, 8-bit analog-to-digital converter (ADC)
- 16-channel, 8-bit pulse width modulator (PWM)
- Full universal serial bus (USB) specification 1.0 compliant low-speed bus with 3 endpoints:
 - 1 Control endpoint (two 8-byte buffer)
 - 2 Interrupt endpoints (one 8-byte buffer shared)
- On-chip 3.3V regulator for USB pull-up resistor
- DDC12AB¹ module with the following:
 - DDC1 hardware
 - Multi-master IIC² hardware for DDC2AB; with dual address
- Additional multi-master IIC module
- 16-bit, 2-channel timer interface modules (TIM) with selectable input capture, output compare, and PWM capability on one channel
- 32 general purpose input/output (I/O) pins, including:
 - 4 open-drain pins

1. DDC is a VESA bus standard.

2. IIC is a proprietary Philips interface bus.

- System protection features:
 - Optional computer operating properly (COP) reset
 - Illegal opcode detection with reset
 - Illegal address detection with reset
- FLASH memory security¹
- Master reset pin with internal pull-up and power-on reset
- $\overline{\text{IRQ}}$ with programmable pull-up and schmitt-trigger input
- 42-pin SDIP and 44-pin QFP packages

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
- Third party C language support

1.4 MCU Block Diagram

Figure 1-1 shows the structure of the MC68HC908BD48.

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

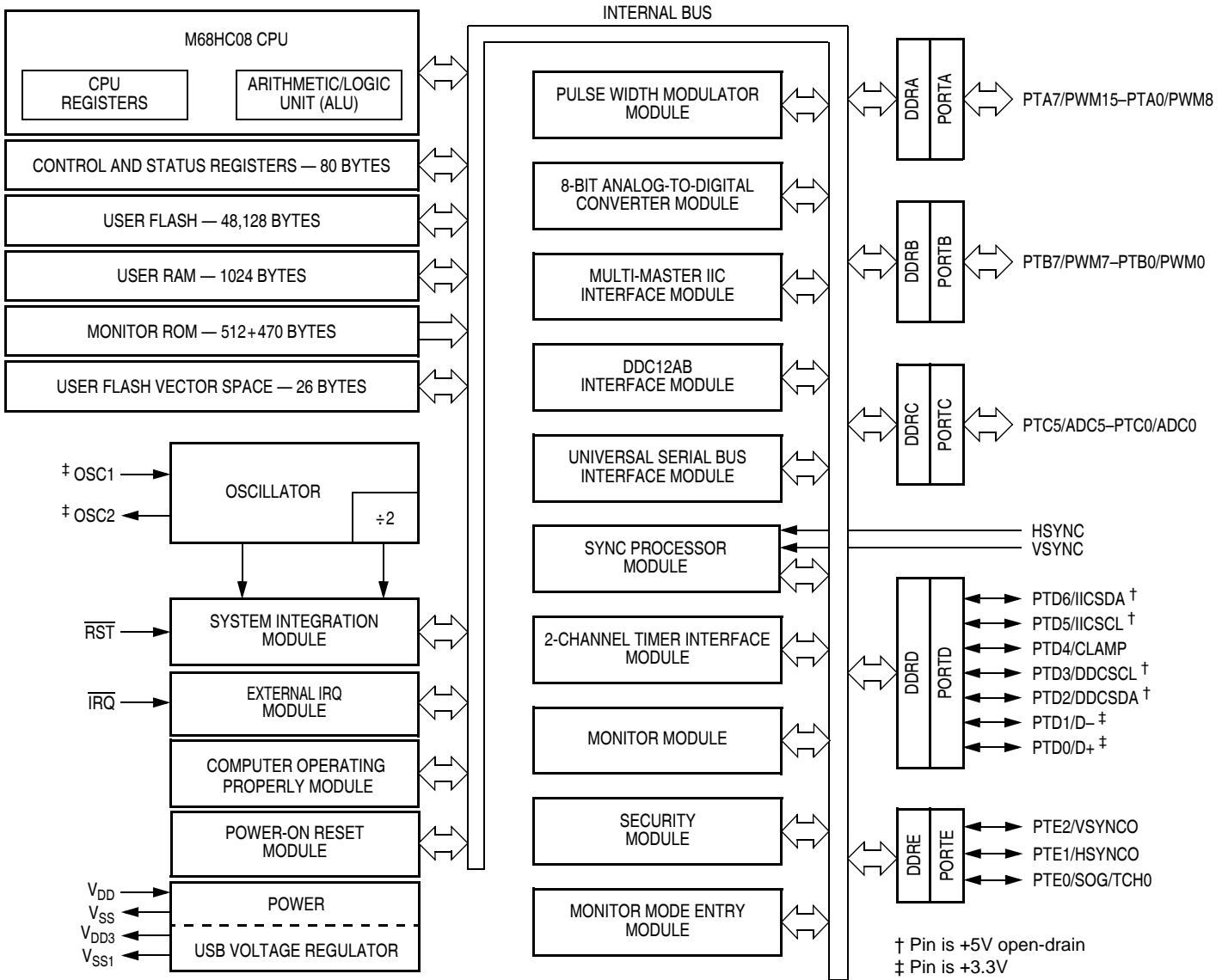


Figure 1-1. MCU Block Diagram



1.5 Pin Assignments

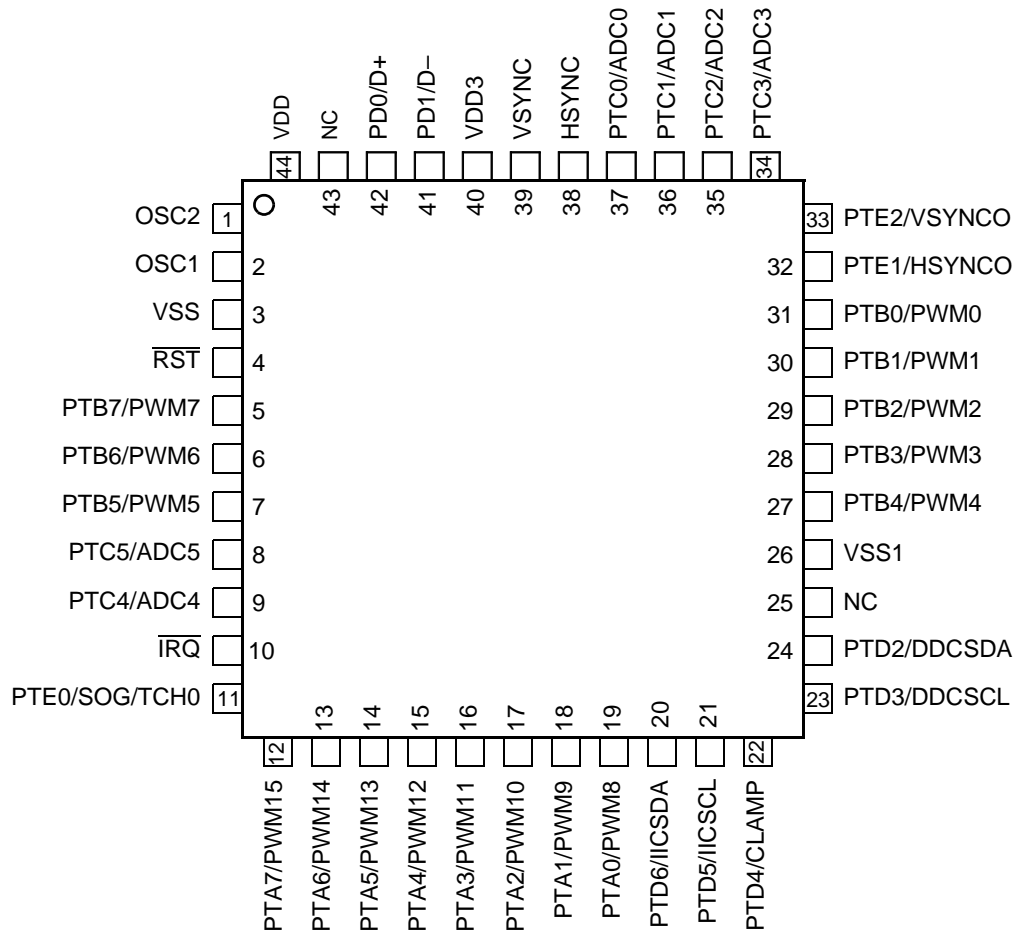


Figure 1-2. 44-Pin QFP Pin Assignments

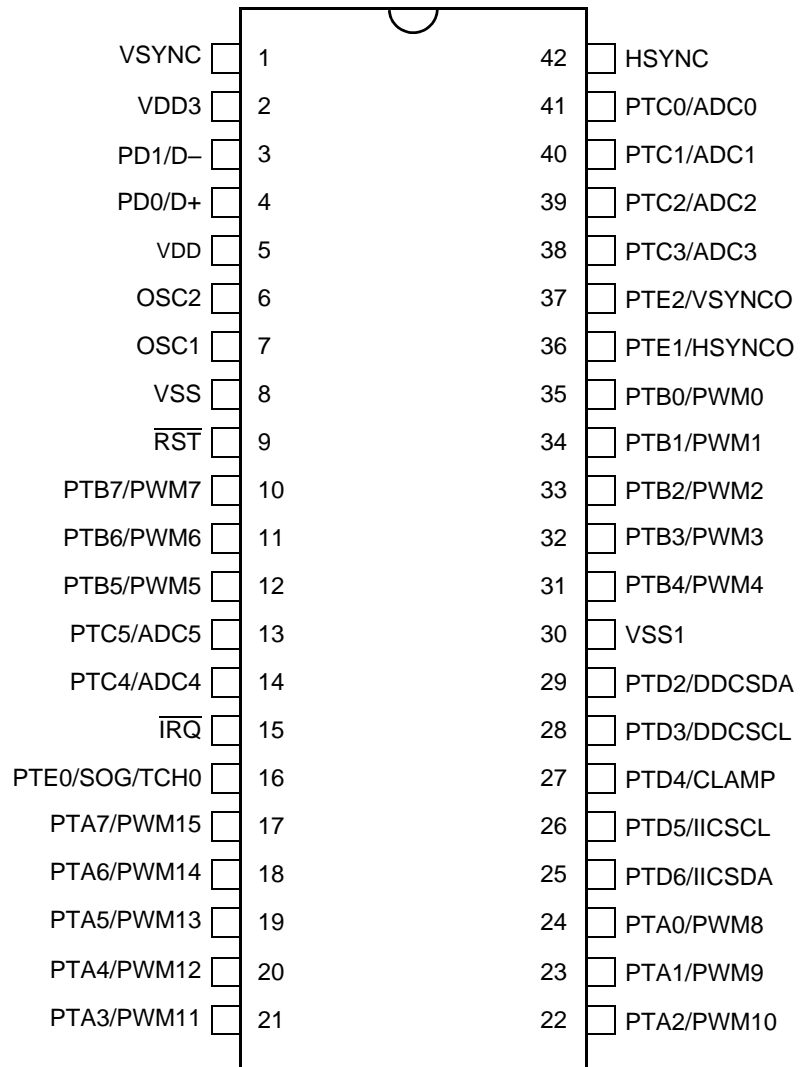


Figure 1-3. 42-Pin SDIP Pin Assignments

1.6 Pin Functions

Description of the pin functions are provided in [Table 1-1](#).

Table 1-1. Pin Functions

| PIN NAME | PIN DESCRIPTION |
|-------------------------|---|
| VDD | Power supply input to the MCU. |
| VSS | Power supply ground. |
| VDD3 | 3.3V regulated output from the MCU. |
| VSS1 | Power supply ground. |
| OSC1 OSC2 | Connections to the on-chip oscillator. An external clock can be connected directly to OSC1; with OSC2 floating. These are 3.3V pins. See Section 8. Oscillator (OSC) . |
| $\overline{\text{RST}}$ | A logic 0 on the $\overline{\text{RST}}$ pin forces the MCU to a known startup 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 7. System Integration Module (SIM) . |
| $\overline{\text{IRQ}}$ | External IRQ pin; with software programmable internal pull-up and schmitt trigger input. This pin is also used for mode entry selection. See Section 7. System Integration Module (SIM) . |
| VSYNC | Vsync input to the sync processor. See Section 16. Sync Processor . |
| HSYNC | Hsync input to the sync processor. See Section 16. Sync Processor . |
| PTA7/PWM15–PTA0/PWM8 | These are shared-function pins. Each pin can be configured as a standard I/O pin or a PWM output channel. See Section 17. Input/Output (I/O) Ports and Section 11. Pulse Width Modulator (PWM) . |
| PTB7/PWM7–PTB0/PWM0 | These are shared-function pins. Each pin can be configured as a standard I/O pin or a PWM output channel. See Section 17. Input/Output (I/O) Ports and Section 11. Pulse Width Modulator (PWM) . |

Table 1-1. Pin Functions

| PIN NAME | PIN DESCRIPTION |
|---------------------|--|
| PTC5/ADC5–PTC0/ADC0 | <p>These are shared-function pins. Each pin can be configured as a standard I/O pin or an ADC input channel.</p> <p>See Section 17. Input/Output (I/O) Ports and Section 12. Analog-to-Digital Converter (ADC).</p> |
| PTD6/IICSDA | <p>This is a shared function pin. It can be configured as a standard I/O pin or the data line of the multi-master IIC module. This pin is open-drain when configured as output.</p> <p>See Section 17. Input/Output (I/O) Ports and Section 14. Multi-Master IIC Interface (MMIIC).</p> |
| PTD5/IICSCL | <p>This is a shared function pin. It can be configured as a standard I/O pin or the clock line of the multi-master IIC module. This pin is open-drain when configured as output.</p> <p>See Section 17. Input/Output (I/O) Ports and Section 14. Multi-Master IIC Interface (MMIIC).</p> |
| PTD4/CLAMP | <p>This is a shared function pin. It can be configured as a standard I/O pin or the clamp output from the sync processor.</p> <p>See Section 17. Input/Output (I/O) Ports and Section 16. Sync Processor.</p> |
| PTD3/DDCSCL | <p>This is a shared function pin. It can be configured as a standard I/O pin or as the clock line of the DDC12AB module. This pin is open-drain when configured as output.</p> <p>See Section 17. Input/Output (I/O) Ports and Section 15. DDC12AB Interface.</p> |
| PTD2/DDCSDA | <p>This is a shared function pin. It can be configured as a standard I/O pin or the data line of the DDC12AB module. This pin is open-drain when configured as output.</p> <p>See Section 17. Input/Output (I/O) Ports and Section 15. DDC12AB Interface.</p> |
| PTD1/D– PTD0/D+ | <p>These are 3.3V, shared function pins. The pins can be configured as standard I/O pins or USB interface differential data lines.</p> <p>See Section 17. Input/Output (I/O) Ports and Section 13. Universal Serial Bus Module (USB).</p> |

Table 1-1. Pin Functions

| PIN NAME | PIN DESCRIPTION |
|---------------|---|
| PTE2/VSYNCO | This is a shared function pin. It can be configured as a standard I/O pin or the Hsync output from the sync processor. See Section 17. Input/Output (I/O) Ports and Section 16. Sync Processor . |
| PTE1/HSYNCO | This is a shared function pin. It can be configured as a standard I/O pin or the Vsync output from the sync processor. See Section 17. Input/Output (I/O) Ports and Section 16. Sync Processor . |
| PTE0/SOG/TCH0 | This is a shared function pin. It can be configured as a standard I/O pin, the SOG input to the sync processor, or the timer channel 0 I/O pin. See Section 17. Input/Output (I/O) Ports , Section 16. Sync Processor , and Section 10. Timer Interface Module (TIM) . |

NOTE: Any unused inputs and I/O ports should be tied to an appropriate logic level (either V_{DD} or V_{SS}). Although the I/O ports of the MC68HC908BD48 do not require termination, termination is recommended to reduce the possibility of static damage.



Section 2. Memory Map

2.1 Contents

| | | |
|-----|--|----|
| 2.2 | Introduction | 35 |
| 2.3 | Unimplemented Memory Locations | 35 |
| 2.4 | Reserved Memory Locations | 36 |
| 2.5 | Input/Output (I/O) Section. | 36 |

2.2 Introduction

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

- 48,128 bytes of FLASH memory
- 1,024 bytes of random-access memory (RAM)
- 26 bytes of user-defined vectors
- 512 + 470 bytes of monitor ROM

2.3 Unimplemented Memory Locations

Accessing an unimplemented location can cause an illegal address reset if illegal address resets are enabled. 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 area of \$0000–\$005F. Additional I/O registers have these addresses:

- \$FE00; SIM Break Status Register, SBSR
- \$FE01; SIM Reset Status Register, SRSR
- \$FE02; reserved
- \$FE03; SIM Break Flag Control Register, SBFCR
- \$FE04; Interrupt Status Register 1, INT1
- \$FE05; Interrupt Status Register 2, INT2
- \$FE06; reserved
- \$FE07; FLASH Control Register, FLCR
- \$FE08; FLASH block protect register, FLBPR
- \$FE09; reserved
- \$FE0A; reserved
- \$FE0B; reserved
- \$FE0C; Break Address Register High, BRKH
- \$FE0D; Break Address Register Low, BRKL
- \$FE0E; Break Status and Control Register, BRKSCR

Data registers are shown in [Figure 2-2](#). [Table 2-1](#) is a list of vector locations.

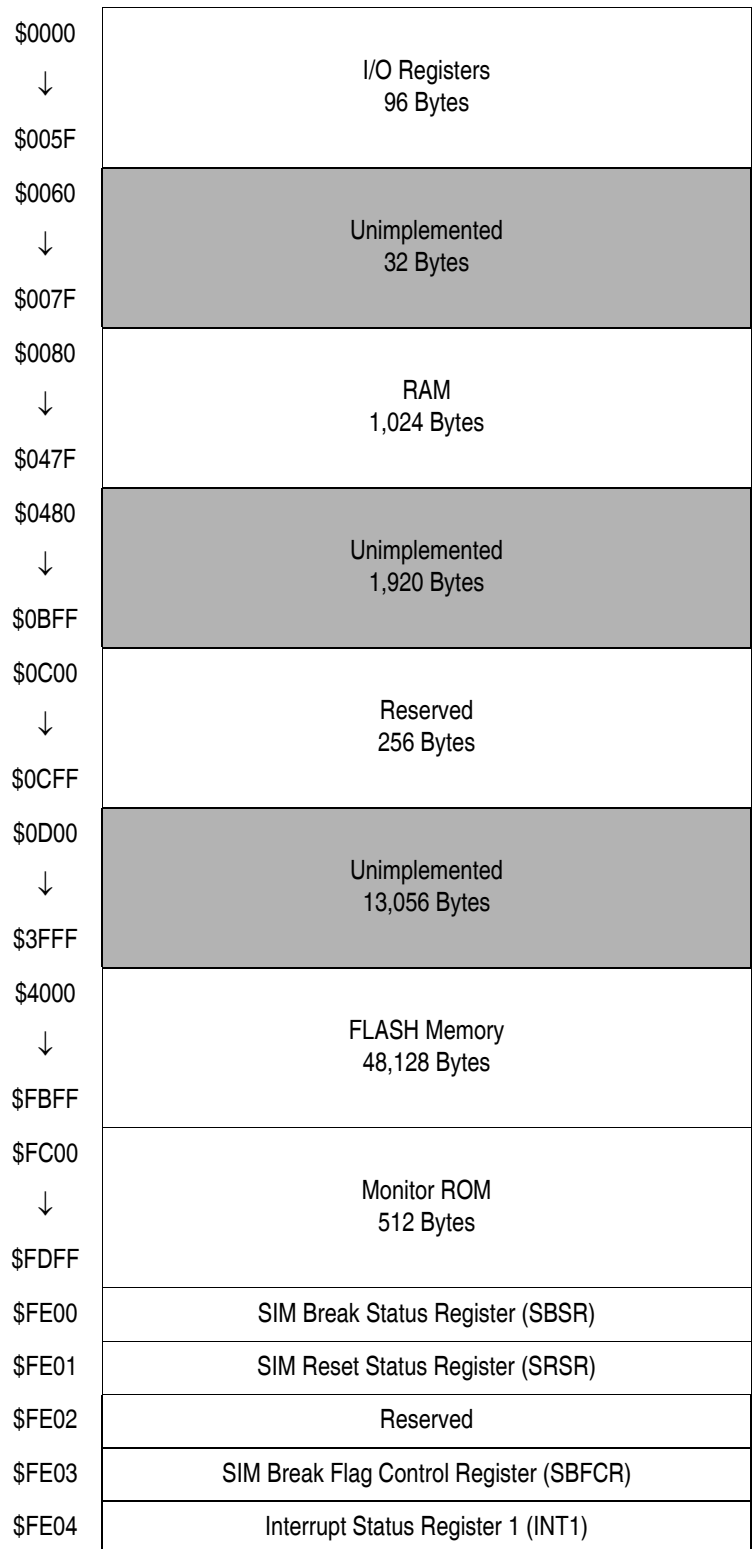


Figure 2-1. Memory Map

| | |
|--------|--|
| \$FE05 | Interrupt Status Register 2 (INT2) |
| \$FE06 | Reserved |
| \$FE07 | FLASH Control Register (FLCR) |
| \$FE08 | FLASH Block Protect Register (FLBPR) |
| \$FE09 | Reserved |
| \$FE0A | Reserved |
| \$FE0B | Reserved |
| \$FE0C | Break Address Register High (BRKH) |
| \$FE0D | Break Address Register Low (BRKL) |
| \$FE0E | Break Status and Control Register (BRKSCR) |
| \$FE0F | Reserved |
| \$FE10 | Monitor ROM 470 Bytes |
| ↓ | |
| \$FFE5 | FLASH Vectors 26 Bytes |
| \$FFE6 | |
| ↓ | |
| \$FFFF | |

Figure 2-1. Memory Map (Continued)

| 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: | PTB7 | PTB6 | PTB5 | PTB4 | PTB3 | PTB2 | PTB1 | PTB0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0002 | Port C Data Register (PTC) | Read: | 0 | 0 | PTC5 | PTC4 | PTC3 | PTC2 | PTC1 | PTC0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0003 | Port D Data Register (PTD) | Read: | 0 | 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: | DDRB7 | DDRB6 | DDRB5 | DDRB4 | DDRB3 | DDRB2 | DDRB1 | DDRB0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0006 | Data Direction Register C (DDRC) | Read: | 0 | 0 | DDRC5 | DDRC4 | DDRC3 | DDRC2 | DDRC1 | DDRC0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0007 | Data Direction Register D (DDRD) | Read: | 0 | DDRD6 | DDRD5 | DDRD4 | DDRD3 | DDRD2 | DDRD1 | DDRD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0008 | Port E Data Register (PTE) | Read: | 0 | 0 | 0 | 0 | 0 | PTE2 | PTE1 | PTE0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0009 | Data Direction Register E (DDRE) | Read: | 0 | 0 | 0 | 0 | 0 | DDRE2 | DDRE1 | DDRE0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented
 R = Reserved

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

Memory Map

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--------|--|--------|---------------------------|-------|-------|-------|-------|-------|-------|--------|
| \$000A | TIM Status and Control Register (TSC) | Read: | TOF | TOIE | TSTOP | 0 | 0 | PS2 | PS1 | PS0 |
| | | Write: | 0 | | | TRST | | | | |
| | | Reset: | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| \$000B | Unimplemented | Read: | | | | | | | | |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$000C | TIM Counter Register High (TCNTH) | Read: | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$000D | TIM Counter Register Low (TCNTL) | Read: | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$000E | TIM Counter Modulo Register High (TMODH) | Read: | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$000F | TIM Counter Modulo Register Low (TMDL) | Read: | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$0010 | TIM Channel 0 Status and Control Register (TSC0) | Read: | CH0F | CH0IE | MS0B | MS0A | ELS0B | ELS0A | TOV0 | CH0MAX |
| | | Write: | 0 | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0011 | TIM Channel 0 Register High (TCH0H) | Read: | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
| | | Write: | | | | | | | | |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0012 | TIM Channel 0 Register Low (TCH0L) | Read: | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| | | Write: | | | | | | | | |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0013 | TIM Channel 1 Status and Control Register (TSC1) | Read: | CH1F | CH1IE | 0 | MS1A | ELS1B | ELS1A | TOV1 | CH1MAX |
| | | Write: | 0 | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented
 = 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 | TIM Channel 1 Register High (TCH1H) | Read: | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
| | | Write: | | | | | | | | |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0015 | TIM Channel 1 Register Low (TCH1L) | Read: | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| | | Write: | | | | | | | | |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0016 | DDC Master Control Register (DMCR) | Read: | ALIF | NAKIF | BB | MAST | MRW | BR2 | BR1 | BR0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0017 | DDC Address Register (DADR) | Read: | DAD7 | DAD6 | DAD5 | DAD4 | DAD3 | DAD2 | DAD1 | EXTAD |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| \$0018 | DDC Control Register (DCR) | Read: | DEN | DIEN | 0 | 0 | TXAK | SCLIEN | DDC1EN | 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0019 | DDC Status Register (DSR) | Read: | RXIF | TXIF | MATCH | SRW | RXAK | SCLIF | TXBE | RXBF |
| | | Write: | 0 | 0 | | | | 0 | | |
| | | Reset: | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| \$001A | DDC Data Transmit Register (DDTR) | Read: | DTD7 | DTD6 | DTD5 | DTD4 | DTD3 | DTD2 | DTD1 | DTD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$001B | DDC Data Receive Register (DDRR) | Read: | DRD7 | DRD6 | DRD5 | DRD4 | DRD3 | DRD2 | DRD1 | DRD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$001C | DDC2 Address Register (D2ADR) | Read: | D2AD7 | D2AD6 | D2AD5 | D2AD4 | D2AD3 | D2AD2 | D2AD1 | 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$001D | Configuration Register 0 (CONFIG0) | Read: | HSYNCOE | VSYNCOE | SOGE | 0 | 0 | 0 | 0 | 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented
 R = Reserved

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

Memory Map

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--|---|--------|-----------------|-------|-------|-------|--------------|-------|-------|-------|
| \$001E | IRQ Status and Control Register (INTSCR) | Read: | 0 | 0 | 0 | 0 | IRQF | 0 | IMASK | MODE |
| | | Write: | | | | | | ACK | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$001F | Configuration Register 1 (CONFIG1) [†] | Read: | 0 | 0 | 0 | 0 | SSREC | COPRS | STOP | COPD |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| † One-time writable register after each reset. | | | | | | | | | | |
| \$0020 | PWM0 Data Register (0PWM) | Read: | 0PWM4 | 0PWM3 | 0PWM2 | 0PWM1 | 0PWM0 | 0BRM2 | 0BRM1 | 0BRM0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0021 | PWM1 Data Register (1PWM) | Read: | 1PWM4 | 1PWM3 | 1PWM2 | 1PWM1 | 1PWM0 | 1BRM2 | 1BRM1 | 1BRM0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0022 | PWM2 Data Register (2PWM) | Read: | 2PWM4 | 2PWM3 | 2PWM2 | 2PWM1 | 2PWM0 | 2BRM2 | 2BRM1 | 2BRM0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0023 | PWM3 Data Register (3PWM) | Read: | 3PWM4 | 3PWM3 | 3PWM2 | 3PWM1 | 3PWM0 | 3BRM2 | 3BRM1 | 3BRM0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0024 | PWM4 Data Register (4PWM) | Read: | 4PWM4 | 4PWM3 | 4PWM2 | 4PWM1 | 4PWM0 | 4BRM2 | 4BRM1 | 4BRM0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0025 | PWM5 Data Register (5PWM) | Read: | 5PWM4 | 5PWM3 | 5PWM2 | 5PWM1 | 5PWM0 | 5BRM2 | 5BRM1 | 5BRM0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0026 | PWM6 Data Register (6PWM) | Read: | 6PWM4 | 6PWM3 | 6PWM2 | 6PWM1 | 6PWM0 | 6BRM2 | 6BRM1 | 6BRM0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0027 | PWM7 Data Register (7PWM) | Read: | 7PWM4 | 7PWM3 | 7PWM2 | 7PWM1 | 7PWM0 | 7BRM2 | 7BRM1 | 7BRM0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | = 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 | |
|--------|--|--------|---------------------------|---------|---------|---------|---------|---------|---------|---------|
| \$0028 | PWM Control Register 1 (PVMCR1) | Read: | PWM7E | PWM6E | PWM5E | PWM4E | PWM3E | PWM2E | PWM1E | PWM0E |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0029 | USB Address Register (UADR) | Read: | USBEN | UADD6 | UADD5 | UADD4 | UADD3 | UADD2 | UADD1 | UADD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002A | USB Interrupt Register (UINTR) | Read: | TBEF | RBF | EOPIF | RSTIF | TBIE | RBIE | EOPIE | RSTIE |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002B | USB Control Register 0 (UCR0) | Read: | T0SEQ | STALL0 | TX0E | RX0E | TP0SIZ3 | TP0SIZ2 | TP0SIZ1 | TP0SIZ0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002C | USB Status Register (USR) | Read: | RSEQ | SETUP | TX1ST | 0 | RPSIZ3 | RPSIZ2 | RPSIZ1 | RPSIZ0 |
| | | Write: | | | | | | | | |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$002D | USB Control Register 2 (UCR2) | Read: | 0 | 0 | PULLEN | SUSPND | ENABLE2 | ENABLE1 | STALL2 | STALL1 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002E | USB Interrupt Register 1 (UIR1) | Read: | TXD1F | TXD1IE | RESUMF | 0 | 0 | 0 | 0 | 0 |
| | | Write: | | | | RESUMFR | TBEFR | RBFFR | TXD1FR | EOPFR |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002F | USB Control Register 1 (UCR1) | Read: | T1SEQ | ENDADD | TX1E | FRESUM | TP1SIZ3 | TP1SIZ2 | TP1SIZ1 | TP1SIZ0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0030 | USB Endpoint 0 Data Register 0 (UDOR0) | Read: | UE0RD07 | UE0RD06 | UE0RD05 | UE0RD04 | UE0RD03 | UE0RD02 | UE0RD01 | UE0RD00 |
| | | Write: | UE0TD07 | UE0TD06 | UE0TD05 | UE0TD04 | UE0TD03 | UE0TD02 | UE0TD01 | UE0TD00 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0031 | USB Endpoint 0 Data Register 1 (UDOR1) | Read: | UE0RD17 | UE0RD16 | UE0RD15 | UE0RD14 | UE0RD13 | UE0RD12 | UE0RD11 | UE0RD10 |
| | | Write: | UE0TD17 | UE0TD16 | UE0TD15 | UE0TD14 | UE0TD13 | UE0TD12 | UE0TD11 | UE0TD10 |
| | | Reset: | Indeterminate after reset | | | | | | | |

= Unimplemented
 = 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 | |
|--------|--|--------|---------------------------|---------|---------|---------|---------|---------|---------|---------|
| \$0032 | USB Endpoint 0 Data Register 2 (UD0R2) | Read: | UE0RD27 | UE0RD26 | UE0RD25 | UE0RD24 | UE0RD23 | UE0RD22 | UE0RD21 | UE0RD20 |
| | | Write: | UE0TD27 | UE0TD26 | UE0TD25 | UE0TD24 | UE0TD23 | UE0TD22 | UE0TD21 | UE0TD20 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0033 | USB Endpoint 0 Data Register 3 (UD0R3) | Read: | UE0RD37 | UE0RD36 | UE0RD35 | UE0RD34 | UE0RD33 | UE0RD32 | UE0RD31 | UE0RD30 |
| | | Write: | UE0TD37 | UE0TD36 | UE0TD35 | UE0TD34 | UE0TD33 | UE0TD32 | UE0TD31 | UE0TD30 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0034 | USB Endpoint 0 Data Register 4 (UD0R4) | Read: | UE0RD47 | UE0RD46 | UE0RD45 | UE0RD44 | UE0RD43 | UE0RD42 | UE0RD41 | UE0RD40 |
| | | Write: | UE0TD47 | UE0TD46 | UE0TD45 | UE0TD44 | UE0TD43 | UE0TD42 | UE0TD41 | UE0TD40 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0035 | USB Endpoint 0 Data Register 5 (UD0R5) | Read: | UE0RD57 | UE0RD56 | UE0RD55 | UE0RD54 | UE0RD53 | UE0RD52 | UE0RD51 | UE0RD50 |
| | | Write: | UE0TD57 | UE0TD56 | UE0TD55 | UE0TD54 | UE0TD53 | UE0TD52 | UE0TD51 | UE0TD50 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0036 | USB Endpoint 0 Data Register 6 (UD0R6) | Read: | UE0RD67 | UE0RD66 | UE0RD65 | UE0RD64 | UE0RD63 | UE0RD62 | UE0RD61 | UE0RD60 |
| | | Write: | UE0TD67 | UE0TD66 | UE0TD65 | UE0TD64 | UE0TD63 | UE0TD62 | UE0TD61 | UE0TD60 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0037 | USB Endpoint 0 Data Register 7 (UD0R7) | Read: | UE0RD77 | UE0RD76 | UE0RD75 | UE0RD74 | UE0RD73 | UE0RD72 | UE0RD71 | UE0RD70 |
| | | Write: | UE0TD77 | UE0TD76 | UE0TD75 | UE0TD74 | UE0TD73 | UE0TD72 | UE0TD71 | UE0TD70 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0038 | USB Endpoint 1/2 Data Register 0 (UD1R0) | Read: | | | | | | | | |
| | | Write: | UE1TD07 | UE1TD06 | UE1TD05 | UE1TD04 | UE1TD03 | UE1TD02 | UE1TD01 | UE1TD00 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0039 | USB Endpoint 1/2 Data Register 1 (UD1R1) | Read: | | | | | | | | |
| | | Write: | UE1TD17 | UE1TD16 | UE1TD15 | UE1TD14 | UE1TD13 | UE1TD12 | UE1TD11 | UE1TD10 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$003A | USB Endpoint 1/2 Data Register 2 (UD1R2) | Read: | | | | | | | | |
| | | Write: | UE1TD27 | UE1TD26 | UE1TD25 | UE1TD24 | UE1TD23 | UE1TD22 | UE1TD21 | UE1TD20 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$003B | USB Endpoint 1/2 Data Register 3 (UD1R3) | Read: | | | | | | | | |
| | | Write: | UE1TD37 | UE1TD36 | UE1TD35 | UE1TD34 | UE1TD33 | UE1TD32 | UE1TD31 | UE1TD30 |
| | | Reset: | Indeterminate after reset | | | | | | | |

= 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 | |
|--------|--|--------|---------------------------|---------|---------|---------|---------|---------|---------|---------|
| \$003C | USB Endpoint 1/2 Data Register 4 (UD1R4) | Read: | | | | | | | | |
| | | Write: | UE1TD47 | UE1TD46 | UE1TD45 | UE1TD44 | UE1TD43 | UE1TD42 | UE1TD41 | UE1TD40 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$003D | USB Endpoint 1/2 Data Register 5 (UD1R5) | Read: | | | | | | | | |
| | | Write: | UE1TD57 | UE1TD56 | UE1TD55 | UE1TD54 | UE1TD53 | UE1TD52 | UE1TD51 | UE1TD50 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$003E | USB Endpoint 1/2 Data Register 6 (UD1R6) | Read: | | | | | | | | |
| | | Write: | UE1TD67 | UE1TD66 | UE1TD65 | UE1TD64 | UE1TD63 | UE1TD62 | UE1TD61 | UE1TD60 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$003F | USB Endpoint 1/2 Data Register 7 (UD1R7) | Read: | | | | | | | | |
| | | Write: | UE1TD77 | UE1TD76 | UE1TD75 | UE1TD74 | UE1TD73 | UE1TD72 | UE1TD71 | UE1TD70 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0040 | Sync Processor Control and Status Register (SPCSR) | Read: | VSIE | VEDGE | VSIF | COMP | VINVO | HINVO | VPOL | HPOL |
| | | Write: | | | 0 | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0041 | Vertical Frequency High Register (VFHR) | Read: | VOF | 0 | 0 | VF12 | VF11 | VF10 | VF9 | VF8 |
| | | Write: | | CPW1 | CPW0 | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0042 | Vertical Frequency Low Register (VFLR) | Read: | VF7 | VF6 | VF5 | VF4 | VF3 | VF2 | VF1 | VF0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0043 | Hsync Frequency High Register (HFHR) | Read: | HFH7 | HFH6 | HFH5 | HFH4 | HFH3 | HFH2 | HFH1 | HFH0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0044 | Hsync Frequency Low Register (HFLR) | Read: | HOVER | 0 | 0 | HFL4 | HFL3 | HFL2 | HFL1 | HFL0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0045 | Sync Processor I/O Control Register (SPIOCR) | Read: | VSYNCNS | HSYNCNS | COINV | R | SOGSEL | CLAMPOE | BPOR | SOUT |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented
 R = Reserved


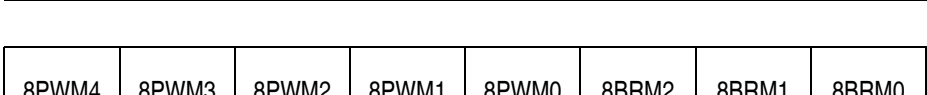
Figure 2-2. Control, Status, and Data Registers (Sheet 7 of 12)

Memory Map

| Addr. | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|--|--------|--------|---------|---------|--------|---------|---------|--------|---------|
| \$0046 | Sync Processor Control Register 1 (SPCR1) | Read: | LVSIE | LVSIF | HPS1 | HPS0 | R | R | ATPOL | FSHF |
| | | Write: | | 0 | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0047 | H&V Sync Output Control Register (HVOCR) | Read: | R | 0 | 0 | 0 | 0 | HVOCR2 | HVOCR1 | HVOCR0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0048 | Unimplemented | Read: | | | | | | | | |
| | | Write: | | | | | | | | |
| | | Reset: | | | | | | | | |
| \$0049 | Port D Configuration Register (PDCR) | Read: | 0 | IICDATE | IICSCLE | CLAMPE | DDCSCLE | DDCDATE | USBD-E | USBD+E |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$004A | Multi-Master IIC Master Control Register (MIMCR) | Read: | MMALIF | MMNAKIF | MMBB | MMAST | MMRW | MMBR2 | MMBR1 | MMBR0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$004B | Multi-Master IIC Address Register (MMADR) | Read: | MMAD7 | MMAD6 | MMAD5 | MMAD4 | MMAD3 | MMAD2 | MMAD1 | MMEXTAD |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| \$004C | Multi-Master IIC Control Register (MMCR) | Read: | MMEN | MMIEN | 0 | 0 | MMTXAK | 0 | 0 | 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$004D | Multi-Master IIC Status Register (MMSR) | Read: | MMRXIF | MMTXIF | MMATCH | MMSRW | MMRXAK | 0 | MMTXBE | MMRXBF |
| | | Write: | 0 | 0 | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| \$004E | Multi-Master IIC Data Transmit Register (MMDTR) | Read: | MMTD7 | MMTD6 | MMTD5 | MMTD4 | MMTD3 | MMTD2 | MMTD1 | MMTD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$004F | Multi-Master IIC Data Receive Register (MMDRR) | Read: | MMRD7 | MMRD6 | MMRD5 | MMRD4 | MMRD3 | MMRD2 | MMRD1 | MMRD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= 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 |
|--------|--------------------------------|---|--------|--------|--------|--------|--------|--------|--------|
| \$0050 | Unimplemented | Read:  Write:  Reset: | | | | | | | |
| \$0051 | PWM8 Data Register (8PWM) | 8PWM4 | 8PWM3 | 8PWM2 | 8PWM1 | 8PWM0 | 8BRM2 | 8BRM1 | 8BRM0 |
| \$0052 | PWM9 Data Register (9PWM) | 9PWM4 | 9PWM3 | 9PWM2 | 9PWM1 | 9PWM0 | 9BRM2 | 9BRM1 | 9BRM0 |
| \$0053 | PWM10 Data Register (10PWM) | 10PWM4 | 10PWM3 | 10PWM2 | 10PWM1 | 10PWM0 | 10BRM2 | 10BRM1 | 10BRM0 |
| \$0054 | PWM11 Data Register (11PWM) | 11PWM4 | 11PWM3 | 11PWM2 | 11PWM1 | 11PWM0 | 11BRM2 | 11BRM1 | 11BRM0 |
| \$0055 | PWM12 Data Register (12PWM) | 12PWM4 | 12PWM3 | 12PWM2 | 12PWM1 | 12PWM0 | 12BRM2 | 12BRM1 | 12BRM0 |
| \$0056 | PWM13 Data Register (13PWM) | 13PWM4 | 13PWM3 | 13PWM2 | 13PWM1 | 13PWM0 | 13BRM2 | 13BRM1 | 13BRM0 |
| \$0057 | PWM14 Data Register (14PWM) | 14PWM4 | PWM3 | 14PWM2 | 14PWM1 | 14PWM0 | 14BRM2 | 14BRM1 | 14BRM0 |
| \$0058 | PWM15 Data Register (15PWM) | 15PWM4 | 15PWM3 | 15PWM2 | 15PWM1 | 15PWM0 | 15BRM2 | 15BRM1 | 15BRM0 |
| \$0059 | PWM Control Register 2 (PWMC2) | PWM15E | PWM14E | PWM13E | PWM12E | PWM11E | PWM10E | PWM9E | PWM8E |

= Unimplemented
 R = Reserved

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

Memory Map

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--------------------------------------|---|--------|------------------------|-------|-------|-------|-------|-------|-------|-------|
| \$005A | Unimplemented | Read: | | | | | | | | |
| | | Write: | | | | | | | | |
| | | Reset: | | | | | | | | |
| \$005B | Unimplemented | Read: | | | | | | | | |
| | | Write: | | | | | | | | |
| | | Reset: | | | | | | | | |
| \$005C | Unimplemented | Read: | | | | | | | | |
| | | Write: | | | | | | | | |
| | | Reset: | | | | | | | | |
| \$005D | 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 |
| \$005E | ADC Data Register (ADR) | Read: | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | AD1 | AD0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected after Reset | | | | | | | |
| \$005F | ADC Input Clock Register (ADICLK) | Read: | ADIV2 | ADIV1 | ADIV0 | 0 | 0 | 0 | 0 | 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE00 | SIM Break Status Register (SBSR) | Read: | R | R | R | R | R | SBSW | R | |
| | | Write: | | | | | | Note | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Note: Writing a logic 0 clears SBSW. | | | | | | | | | | |
| \$FE01 | SIM Reset Status Register (SRSR) | Read: | POR | PIN | COP | ILOP | ILAD | 0 | 0 | 0 |
| | | Write: | | | | | | | | |
| | | POR: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE02 | Reserved | Read: | R | R | R | R | R | R | R | R |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= 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 |
|--------|---|--------|-------|-------|-------|-------|-------|-------|-------|
| \$FE03 | SIM Break Flag Control Register (SBFCR) | Read: | BCFE | R | R | R | R | R | R |
| | | Write: | | | | | | | |
| | | Reset: | 0 | | | | | | |
| \$FE04 | Interrupt Status Register 1 (INT1) | Read: | IF6 | IF5 | IF4 | IF3 | IF2 | IF1 | 0 |
| | | Write: | R | R | R | R | R | R | R |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE05 | Interrupt Status Register 2 (INT2) | Read: | 0 | 0 | 0 | 0 | IF10 | IF9 | IF8 |
| | | Write: | R | R | R | R | R | R | R |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE06 | Reserved | Read: | R | R | R | R | R | R | R |
| | | Write: | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE07 | FLASH Control Register (FLCR) | Read: | 0 | 0 | 0 | 0 | HVEN | MASS | ERASE |
| | | Write: | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE08 | FLASH Block Protect Register (FLBPR) | Read: | BPR7 | BPR6 | BPR5 | BPR4 | BPR3 | BPR2 | BPR1 |
| | | Write: | | | | | | | 0 |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE09 | Reserved | Read: | R | R | R | R | R | R | R |
| | | Write: | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE0A | Reserved | Read: | R | R | R | R | R | R | R |
| | | Write: | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE0B | Reserved | Read: | R | R | R | R | R | R | R |
| | | Write: | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE0C | Break Address High Register (BRKH) | Read: | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 |
| | | Write: | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented
 = Reserved

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


Memory Map

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--------|--|--------|--|------|------|------|------|------|-------|------|
| \$FE0D | Break Address low Register (BRKL) | Read: | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| | | 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 |
| \$FFFF | COP Control Register (COPCTL) | Read: | Low byte of reset vector | | | | | | | |
| | | Write: | Writing clears COP counter (any value) | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |

= Unimplemented
 R = Reserved

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

Table 2-1. Vector Addresses

| Vector Priority | INT Flag | Address | Vector |
|---|----------|---------------------------------------|------------------------------|
| Lowest  Highest | — | \$FFE6 | Reserved |
| | | \$FFE7 | Reserved |
| | IF10 | \$FFE8 | ADC Interrupt Vector (High) |
| | | \$FFE9 | ADC Interrupt Vector (Low) |
| | IF9 | \$FFEA | MMIIC Vector (High) |
| | | \$FFEB | MMIIC Vector (Low) |
| | IF8 | \$FFEC | Sync Processor Vector (High) |
| | | \$FFED | Sync Processor Vector (Low) |
| | IF7 | \$FFEE | TIM Overflow Vector (High) |
| | | \$FFEF | TIM Overflow Vector (Low) |
| | IF6 | \$FFF0 | TIM Channel 1 Vector (High) |
| | | \$FFF1 | TIM Channel 1 Vector (Low) |
| | IF5 | \$FFF2 | TIM Channel 0 Vector (High) |
| | | \$FFF3 | TIM Channel 0 Vector (Low) |
| | IF4 | \$FFF4 | Reserved |
| | | \$FFF5 | Reserved |
| | IF3 | \$FFF6 | DDC12AB Vector (High) |
| | | \$FFF7 | DDC12AB Vector (Low) |
| | IF2 | \$FFF8 | USB Vector (High) |
| | | \$FFF9 | USB Vector (Low) |
| IF1 | \$FFFA | $\overline{\text{IRQ}}$ Vector (High) | |
| | \$FFFB | $\overline{\text{IRQ}}$ Vector (Low) | |
| — | \$FFFC | SWI Vector (High) | |
| | \$FFFD | SWI Vector (Low) | |
| — | \$FFFE | Reset Vector (High) | |
| | \$FFFF | Reset Vector (Low) | |



Section 3. Random-Access Memory (RAM)

3.1 Contents

| | | |
|-----|----------------------------------|----|
| 3.2 | Introduction | 53 |
| 3.3 | Functional Description | 53 |

3.2 Introduction

This section describes the 1,024 bytes of RAM (random-access memory).

3.3 Functional Description

Addresses \$0080 through \$047F are RAM locations. The location of the stack RAM is programmable. The 16-bit stack pointer allows the stack to be anywhere in the 64-Kbyte memory space.

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

Within page zero are 128 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 | 55 |
| 4.3 | Functional Description | 56 |
| 4.4 | FLASH Control Register (FLCR) | 56 |
| 4.5 | FLASH Block Erase Operation | 57 |
| 4.6 | FLASH Mass Erase Operation | 58 |
| 4.7 | FLASH Program Operation. | 59 |
| 4.8 | FLASH Block Protection | 61 |
| 4.9 | FLASH Block Protect Register (FLBPR). | 61 |

4.2 Introduction

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

4.3 Functional Description

The FLASH memory is an array of 48,128 bytes with an additional 26 bytes of user vectors. *An erased bit reads as logic 1 and a programmed bit reads as a logic 0.* Program and erase operations are facilitated through control bits in a memory mapped FLASH control register (FLCR). The address ranges for the user memory and vectors are:

- \$4000–\$FBFF; user memory
- \$FE07; FLASH control register
- \$FE08; FLASH block protect register
- \$FFE6–\$FFFF; these locations are reserved for user-defined interrupt and reset vectors

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

4.4 FLASH Control Register (FLCR)

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

Address: \$FE07

| | 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 |

= Unimplemented

Figure 4-1. 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

Setting this read/write bit configures the 48,128 bytes FLASH array for mass erase operation.

- 1 = Mass Erase operation selected
- 0 = Mass Erase operation not 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 Block Erase Operation

Use the following procedure to erase a block (512 bytes) of FLASH memory:

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 block address range desired.
3. Wait for a time, t_{nvs} (min. 5 μ s)

4. Set the HVEN bit.
5. Wait for a time, t_{Erase} (min. 2ms)
6. Clear the ERASE bit.
7. Wait for a time, t_{nvh} (min. 5 μ s)
8. Clear the HVEN bit.
9. After a time, t_{rcv} (min. 1 μ s), the memory can be accessed again in read mode.

NOTE: *Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. 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 entire FLASH memory:

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} (5 μ s).
4. Set the HVEN bit.
5. Wait for a time, t_{MErase} (4ms).
6. Clear the ERASE bit.
7. Wait for a time, t_{nvhl} (100 μ s).
8. Clear the HVEN bit.
9. After time, t_{rcv} (1 μ s), the memory can be accessed again in read mode.

NOTE: *Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. 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, and \$XXC0. Use this step-by-step procedure to program a row of FLASH memory ([Figure 4-2](#) is a flowchart representation):

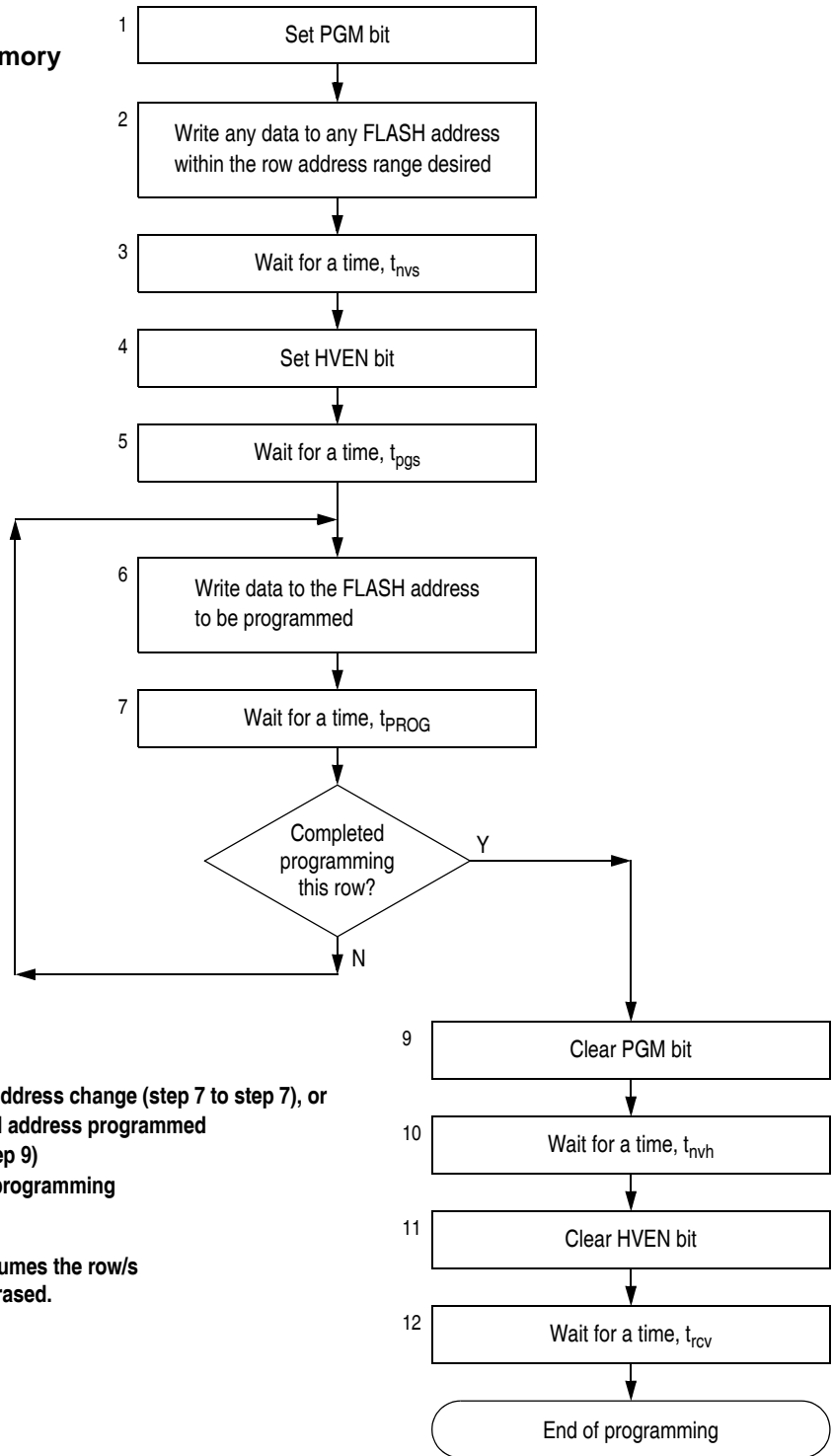
NOTE: *In order to avoid program disturbs, the row must be erased before any byte on that row is programmed.*

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} (min. 5 μ s).
4. Set the HVEN bit.
5. Wait for a time, t_{pgs} (min. 10 μ s).
6. Write data to the FLASH address to be programmed.
7. Wait for time, t_{PROG} (min. 20 μ s).
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} (min. 5 μ s).
11. Clear the HVEN bit.
12. After time, t_{rcv} (min 1 μ s), the memory can be accessed in read mode again.

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

NOTE: *Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order shown, other unrelated operations may occur between the steps. Do not exceed t_{PROG} maximum. See [21.15 Memory Characteristics](#).*

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



NOTE:

The time between each FLASH address change (step 7 to step 7), 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 4-2. FLASH Programming Flowchart

4.8 FLASH Block 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 for protecting blocks 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 at the bottom of the FLASH memory (\$FFFF). When the memory is protected, the HVEN bit cannot be set in either ERASE or PROGRAM operations.

4.9 FLASH Block Protect Register (FLBPR)

The FLASH block protect register is implemented as an 7-bit I/O register. The BPR bit content of the register determines the starting location of the protected range within the FLASH memory.

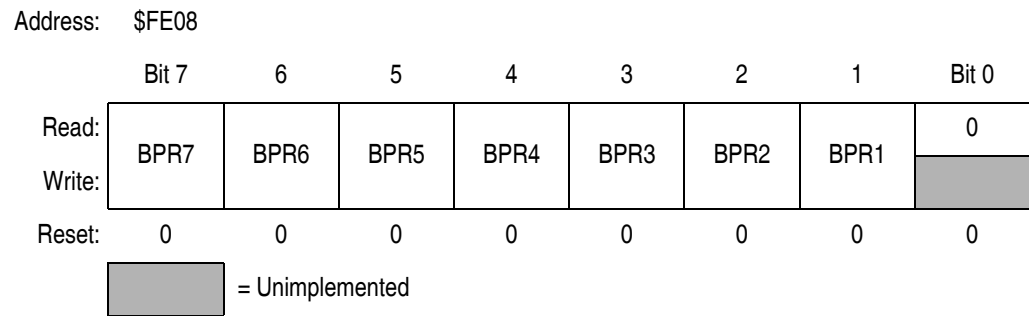


Figure 4-3. FLASH Block Protect Register (FLBPR)

BPR[7:1] — FLASH Block Protect Bits

These seven bits represent bits [15:9] of a 16-bit memory address. Bits [8:0] are logic 0s.

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.



Figure 4-4. FLASH Block Protect Start Address

Examples of protect FLASH memory:

| BPR[7:0] | FLASH Memory Protected Range |
|---|---|
| \$00–3E | The entire FLASH memory is not protected . |
| \$40 | The entire FLASH memory is protected. |
| \$42 (0100 0010) | \$4200 (0100 0010 0000 0000) to \$FFFF |
| \$44 (0100 0100) | \$4400 (0100 0100 0000 0000) to \$FFFF |
| and so on... | |
| \$F8 (1111 1000) | \$F800 (1111 1000 0000 0000) to \$FFFF |
| \$FA (1111 1010) | \$FA00 (1111 1010 0000 0000) to \$FFFF |
| Note: The user FLASH vectors from \$FFE6 to \$FFFF are always protected, and can only be erased by a FLASH mass erase operation. | |

Section 5. Configuration Register (CONFIG)

5.1 Contents

| | | |
|-------|-----------------------------------|----|
| 5.2 | Introduction | 63 |
| 5.3 | Functional Description | 63 |
| 5.3.1 | Configuration Register 0. | 64 |
| 5.3.2 | Configuration Register 1. | 65 |

5.2 Introduction

This section describes the configuration registers, CONFIG0 and CONFIG1. The configuration registers enable or disable these options:

- Sync Processor HSYNCO output pin
- Sync Processor VSYNCO output pin
- Sync Processor SOG input pin
- Stop mode recovery time (32 OSCXCLK cycles or 4096 OSCXCLK cycles)
- COP timeout period ($2^{18} - 2^4$ or $2^{13} - 2^4$ OSCXCLK cycles)
- STOP instruction
- Computer operating properly module (COP)

5.3 Functional Description

The configuration registers are used in the initialization of various options. 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 configuration register may be read at anytime.

5.3.1 Configuration Register 0

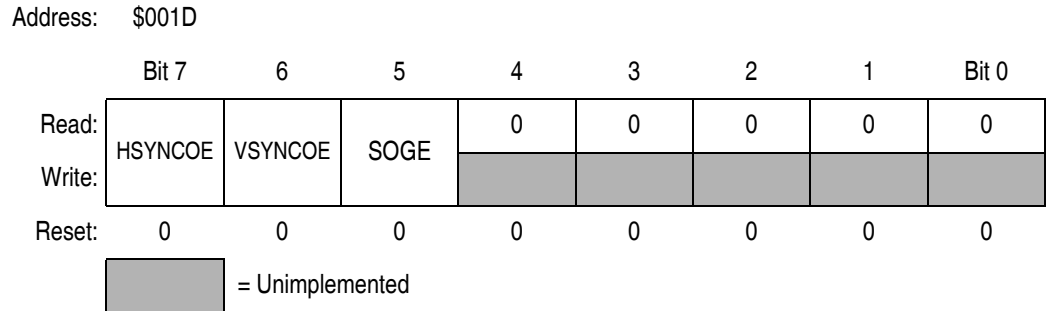


Figure 5-1. Configuration Register 0 (CONFIG0)

HSYNCOE — VSYNCO Enable

This bit is set to configure the PTE1/HSYNCO pin for HSYNCO output function. Reset clears this bit.

- 1 = PTE1/HSYNCO pin configured as HSYNCO pin
- 0 = PTE1/HSYNCO pin configured as standard I/O pin

VSYNCOE — VSYNCO Enable

This bit is set to configure the PTE2/VSYNCO pin for VSYNCO output function. Reset clears this bit.

- 1 = PTE2/VSYNCO pin configured as VSYNCO pin
- 0 = PTE2/VSYNCO pin configured as standard I/O pin

SOGE — SOG Enable

This bit is set to configure the PTE0/SOG/TCH0 pin for SOG output function. Reset clears this bit.

- 1 = PTE0/SOG/TCH0 pin configured as SOG pin
- 0 = PTE0/SOG/TCH0 pin configured as standard I/O or TCH0 pin.
TCH0 function is configured by ELS0B and ELS0A bits in TSC0 (bits 3 and 2 in \$0010). (See [10.10.4 TIM Channel Status and Control Registers \(TSC0:TSC1\)](#).)

5.3.2 Configuration Register 1

Address: \$001F

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|---|---|---|-------|-------|------|-------|
| Read: | 0 | 0 | 0 | 0 | SSREC | COPRS | STOP | COPD |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

Figure 5-2. Configuration Register 1 (CONFIG1)

SSREC — Short Stop Recovery Bit

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

- 1 = Stop mode recovery after 32 OSCXCLK cycles
- 0 = Stop mode recovery after 4096 OSCXCLK cycles

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

If using an external crystal oscillator, do not set the SSREC bit.

COPRS — COP Rate Select Bit

COPRS selects the COP timeout period. Reset clears COPRS. (See [Section 19. Computer Operating Properly \(COP\)](#).)

- 1 = COP timeout period = $2^{13} - 2^4$ CGMXCLK cycles
- 0 = COP timeout period = $2^{18} - 2^4$ CGMXCLK cycles

STOP — STOP Instruction Enable Bit

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 19. Computer Operating Properly \(COP\)](#).)

- 1 = COP module disabled
- 0 = COP module enabled



Configuration Register (CONFIG)

Section 6. Central Processor Unit (CPU)

6.1 Contents

| | | |
|-------|---------------------------------------|----|
| 6.2 | Introduction | 67 |
| 6.3 | Features | 68 |
| 6.4 | CPU Registers | 68 |
| 6.4.1 | Accumulator | 69 |
| 6.4.2 | Index Register | 70 |
| 6.4.3 | Stack Pointer | 70 |
| 6.4.4 | Program Counter | 71 |
| 6.4.5 | Condition Code Register | 71 |
| 6.5 | Arithmetic/Logic Unit (ALU) | 74 |
| 6.6 | Low-Power Modes | 74 |
| 6.6.1 | Wait Mode | 74 |
| 6.6.2 | Stop Mode | 75 |
| 6.7 | CPU During Break Interrupts | 75 |
| 6.8 | Instruction Set Summary | 75 |
| 6.9 | Opcode Map | 75 |

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
- 64-Kbyte 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 64 Kbytes
- 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.

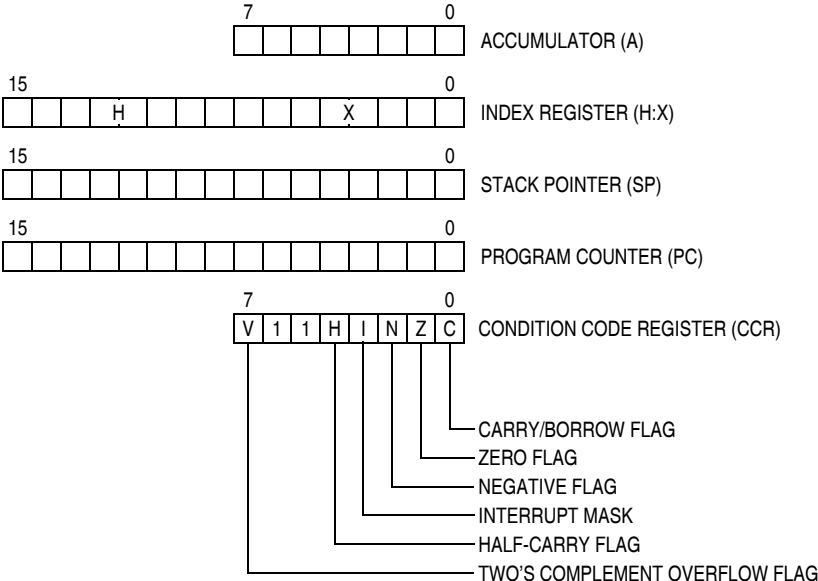


Figure 6-1. CPU Registers

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 64-Kbyte 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.

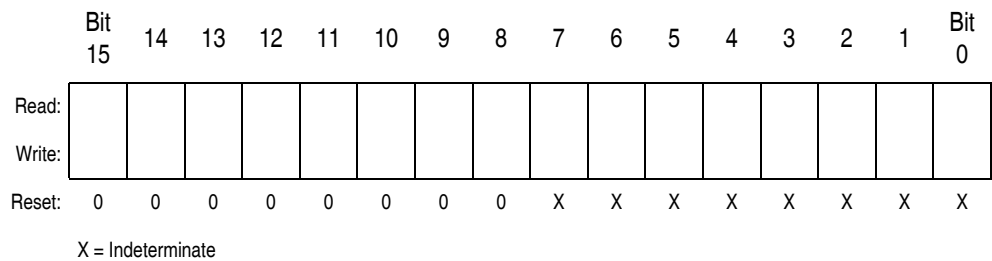


Figure 6-3. Index Register (H:X)

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.

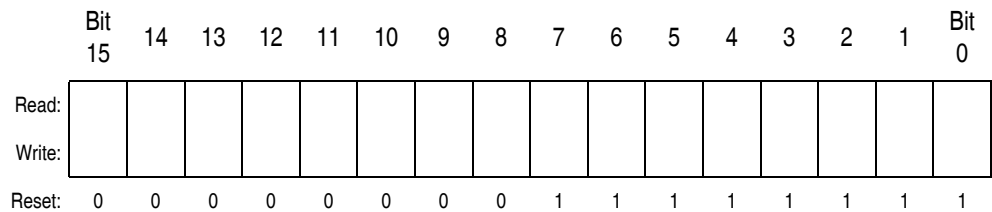


Figure 6-4. Stack Pointer (SP)

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.

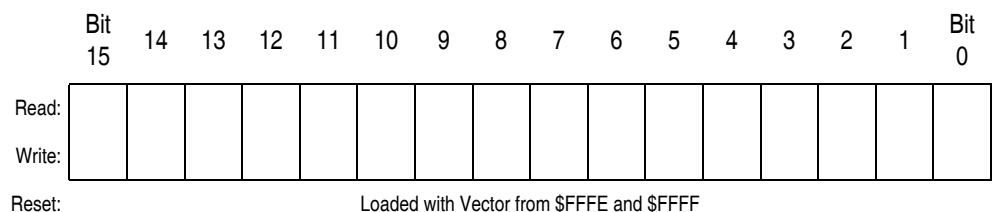


Figure 6-5. Program Counter (PC)

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: | V | 1 | 1 | H | I | N | Z | C |
| Write: | | | | | | | | |
| 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

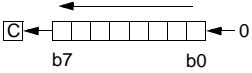
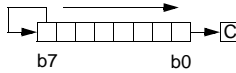
| Source Form | Operation | Description | Effect on CCR | | | | | | Address Mode | Opcode | Operand | Cycles |
|--|--|---|---------------|---|---|---|---|---|--|--|---|--------------------------------------|
| | | | V | H | I | N | Z | C | | | | |
| ADC #opr ADC opr ADC opr ADC opr,X ADC opr,X ADC ,X ADC opr,SP ADC opr,SP | Add with Carry | $A \leftarrow (A) + (M) + (C)$ | ↕ | ↕ | – | ↕ | ↕ | ↕ | IMM DIR EXT IX2 IX1 IX SP1 SP2 | A9 B9 C9 D9 E9 F9 9EE9 9ED9 | ii dd hh ll ee ff ff ff 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 | $A \leftarrow (A) + (M)$ | ↕ | ↕ | – | ↕ | ↕ | ↕ | IMM DIR EXT IX2 IX1 IX SP1 SP2 | AB BB CB DB EB FB 9EEB 9EDB | ii dd hh ll ee ff ff ff ff ee ff | 2 3 4 4 3 2 4 5 |
| AIS #opr | Add Immediate Value (Signed) to SP | $SP \leftarrow (SP) + (16 \ll M)$ | – | – | – | – | – | – | IMM | A7 | ii | 2 |
| AIX #opr | Add Immediate Value (Signed) to H:X | $H:X \leftarrow (H:X) + (16 \ll M)$ | – | – | – | – | – | – | 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 | $A \leftarrow (A) \& (M)$ | 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 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 | $PC \leftarrow (PC) + 2 + rel \text{ ? } (C) = 0$ | – | – | – | – | – | – | REL | 24 | rr | 3 |
| BCLR n, opr | Clear Bit n in M | $M_n \leftarrow 0$ | – | – | – | – | – | – | 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 <i>rel</i> | Branch if Carry Bit Set (Same as BLO) | $PC \leftarrow (PC) + 2 + rel ? (C) = 1$ | - | - | - | - | - | - | REL | 25 | rr | 3 |
| BEQ <i>rel</i> | Branch if Equal | $PC \leftarrow (PC) + 2 + rel ? (Z) = 1$ | - | - | - | - | - | - | REL | 27 | rr | 3 |
| BGE <i>opr</i> | Branch if Greater Than or Equal To (Signed Operands) | $PC \leftarrow (PC) + 2 + rel ? (N \oplus V) = 0$ | - | - | - | - | - | - | REL | 90 | rr | 3 |
| BGT <i>opr</i> | Branch if Greater Than (Signed Operands) | $PC \leftarrow (PC) + 2 + rel ? (Z) (N \oplus V) = 0$ | - | - | - | - | - | - | REL | 92 | rr | 3 |
| BHCC <i>rel</i> | Branch if Half Carry Bit Clear | $PC \leftarrow (PC) + 2 + rel ? (H) = 0$ | - | - | - | - | - | - | REL | 28 | rr | 3 |
| BHCS <i>rel</i> | Branch if Half Carry Bit Set | $PC \leftarrow (PC) + 2 + rel ? (H) = 1$ | - | - | - | - | - | - | REL | 29 | rr | 3 |
| BHI <i>rel</i> | Branch if Higher | $PC \leftarrow (PC) + 2 + rel ? (C) (Z) = 0$ | - | - | - | - | - | - | REL | 22 | rr | 3 |
| BHS <i>rel</i> | Branch if Higher or Same (Same as BCC) | $PC \leftarrow (PC) + 2 + rel ? (C) = 0$ | - | - | - | - | - | - | REL | 24 | rr | 3 |
| BIH <i>rel</i> | Branch if \overline{IRQ} Pin High | $PC \leftarrow (PC) + 2 + rel ? \overline{IRQ} = 1$ | - | - | - | - | - | - | REL | 2F | rr | 3 |
| BIL <i>rel</i> | Branch if \overline{IRQ} Pin Low | $PC \leftarrow (PC) + 2 + rel ? \overline{IRQ} = 0$ | - | - | - | - | - | - | REL | 2E | rr | 3 |
| BIT # <i>opr</i> BIT <i>opr</i> BIT <i>opr</i> BIT <i>opr</i> ,X BIT <i>opr</i> ,X BIT ,X BIT <i>opr</i> ,SP BIT <i>opr</i> ,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 <i>opr</i> | Branch if Less Than or Equal To (Signed Operands) | $PC \leftarrow (PC) + 2 + rel ? (Z) (N \oplus V) = 1$ | - | - | - | - | - | - | REL | 93 | rr | 3 |
| BLO <i>rel</i> | Branch if Lower (Same as BCS) | $PC \leftarrow (PC) + 2 + rel ? (C) = 1$ | - | - | - | - | - | - | REL | 25 | rr | 3 |
| BLS <i>rel</i> | Branch if Lower or Same | $PC \leftarrow (PC) + 2 + rel ? (C) (Z) = 1$ | - | - | - | - | - | - | REL | 23 | rr | 3 |
| BLT <i>opr</i> | Branch if Less Than (Signed Operands) | $PC \leftarrow (PC) + 2 + rel ? (N \oplus V) = 1$ | - | - | - | - | - | - | REL | 91 | rr | 3 |
| BMC <i>rel</i> | Branch if Interrupt Mask Clear | $PC \leftarrow (PC) + 2 + rel ? (I) = 0$ | - | - | - | - | - | - | REL | 2C | rr | 3 |
| BMI <i>rel</i> | Branch if Minus | $PC \leftarrow (PC) + 2 + rel ? (N) = 1$ | - | - | - | - | - | - | REL | 2B | rr | 3 |
| BMS <i>rel</i> | Branch if Interrupt Mask Set | $PC \leftarrow (PC) + 2 + rel ? (I) = 1$ | - | - | - | - | - | - | REL | 2D | rr | 3 |
| BNE <i>rel</i> | Branch if Not Equal | $PC \leftarrow (PC) + 2 + rel ? (Z) = 0$ | - | - | - | - | - | - | REL | 26 | rr | 3 |
| BPL <i>rel</i> | Branch if Plus | $PC \leftarrow (PC) + 2 + rel ? (N) = 0$ | - | - | - | - | - | - | REL | 2A | rr | 3 |
| BRA <i>rel</i> | Branch Always | $PC \leftarrow (PC) + 2 + rel$ | - | - | - | - | - | - | 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 <i>n,opr,rel</i> | Branch if Bit <i>n</i> in M Clear | $PC \leftarrow (PC) + 3 + rel ? (Mn) = 0$ | | | | | | ↓ | DIR (b0) | 01 | dd rr | 5 | | | |
| | | | | | | | | | | DIR (b1) | 03 | dd rr | 5 | | |
| | | | | | | | | | | | DIR (b2) | 05 | dd rr | 5 | |
| | | | | | | | | | | | DIR (b3) | 07 | dd rr | 5 | |
| | | | | | | | | | | | DIR (b4) | 09 | dd rr | 5 | |
| | | | | | | | | | | | DIR (b5) | 0B | dd rr | 5 | |
| | | | | | | | | | | | DIR (b6) | 0D | dd rr | 5 | |
| | | | | | | | | | | | DIR (b7) | 0F | dd rr | 5 | |
| BRN <i>rel</i> | Branch Never | $PC \leftarrow (PC) + 2$ | - | - | - | - | - | - | REL | 21 | rr | 3 | | | |
| BRSET <i>n,opr,rel</i> | Branch if Bit <i>n</i> in M Set | $PC \leftarrow (PC) + 3 + rel ? (Mn) = 1$ | | | | | | ↓ | DIR (b0) | 00 | dd rr | 5 | | | |
| | | | | | | | | | | DIR (b1) | 02 | dd rr | 5 | | |
| | | | | | | | | | | | DIR (b2) | 04 | dd rr | 5 | |
| | | | | | | | | | | | DIR (b3) | 06 | dd rr | 5 | |
| | | | | | | | | | | | DIR (b4) | 08 | dd rr | 5 | |
| | | | | | | | | | | | DIR (b5) | 0A | dd rr | 5 | |
| | | | | | | | | | | | DIR (b6) | 0C | dd rr | 5 | |
| | | | | | | | | | | | DIR (b7) | 0E | dd rr | 5 | |
| BSET <i>n,opr</i> | Set Bit <i>n</i> in M | $Mn \leftarrow 1$ | | | | | | | DIR (b0) | 10 | dd | 4 | | | |
| | | | | | | | | | | DIR (b1) | 12 | dd | 4 | | |
| | | | | | | | | | | | DIR (b2) | 14 | dd | 4 | |
| | | | | | | | | | | | DIR (b3) | 16 | dd | 4 | |
| | | | | | | | | | | | DIR (b4) | 18 | dd | 4 | |
| | | | | | | | | | | | DIR (b5) | 1A | dd | 4 | |
| | | | | | | | | | | | DIR (b6) | 1C | dd | 4 | |
| | | | | | | | | | | | DIR (b7) | 1E | dd | 4 | |
| BSR <i>rel</i> | Branch to Subroutine | $PC \leftarrow (PC) + 2$; push (PCL) $SP \leftarrow (SP) - 1$; push (PCH) $SP \leftarrow (SP) - 1$ $PC \leftarrow (PC) + rel$ | - | - | - | - | - | - | REL | AD | rr | 4 | | | |
| CBEQ <i>opr,rel</i> CBEQA # <i>opr,rel</i> CBEQX # <i>opr,rel</i> CBEQ <i>opr,X+,rel</i> CBEQ <i>X+,rel</i> CBEQ <i>opr,SP,rel</i> | Compare and Branch if Equal | $PC \leftarrow (PC) + 3 + rel ? (A) - (M) = \00 $PC \leftarrow (PC) + 3 + rel ? (A) - (M) = \00 $PC \leftarrow (PC) + 3 + rel ? (X) - (M) = \00 $PC \leftarrow (PC) + 3 + rel ? (A) - (M) = \00 $PC \leftarrow (PC) + 2 + rel ? (A) - (M) = \00 $PC \leftarrow (PC) + 4 + rel ? (A) - (M) = \00 | | | | | | | DIR | 31 | dd rr | 5 | | | |
| | | | | | | | | | | IMM | 41 | ii rr | 4 | | |
| | | | | | | | | | | | IMM | 51 | ii rr | 4 | |
| | | | | | | | | | | | IX1+ | 61 | ff rr | 5 | |
| | | | | | | | | | | | IX+ | 71 | rr | 4 | |
| | | | | | | | | | | | SP1 | 9E61 | ff rr | 6 | |
| CLC | Clear Carry Bit | $C \leftarrow 0$ | - | - | - | - | 0 | - | INH | 98 | | 1 | | | |
| CLI | Clear Interrupt Mask | $I \leftarrow 0$ | - | - | 0 | - | - | - | INH | 9A | | 2 | | | |
| CLR <i>opr</i> CLRA CLR X CLR H CLR <i>opr,X</i> CLR ,X CLR <i>opr,SP</i> | Clear | $M \leftarrow \$00$ $A \leftarrow \$00$ $X \leftarrow \$00$ $H \leftarrow \$00$ $M \leftarrow \$00$ $M \leftarrow \$00$ $M \leftarrow \$00$ | | | | | | | DIR | 3F | dd | 3 | | | |
| | | | | | | | | | | INH | 4F | | 1 | | |
| | | | | | | | | | | | INH | 5F | | 1 | |
| | | | | | | | | | | | INH | 8C | | 1 | |
| | | | | | | 0 | - | - | 0 | 1 | - | INH | 6F | ff | 3 |
| | | | | | | | | | | | IX | 7F | | 2 | |
| | | | | | | | | | | | SP1 | 9E6F | ff | 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 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) | $M \leftarrow (\overline{M}) = \$FF - (M)$ $A \leftarrow (\overline{A}) = \$FF - (M)$ $X \leftarrow (\overline{X}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$ | 0 | – | – | ‡ | ‡ | 1 | DIR INH INH IX1 IX SP1 | 33 43 53 63 73 9E63 | dd ff 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 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 | $A \leftarrow (A) - 1$ or $M \leftarrow (M) - 1$ or $X \leftarrow (X) - 1$ $PC \leftarrow (PC) + 3 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 2 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 2 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 3 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 2 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 2 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 4 + rel ? (result) \neq 0$ | – | – | – | – | – | – | 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 | $M \leftarrow (M) - 1$ $A \leftarrow (A) - 1$ $X \leftarrow (X) - 1$ $M \leftarrow (M) - 1$ $M \leftarrow (M) - 1$ $M \leftarrow (M) - 1$ | ‡ | – | – | ‡ | ‡ | – | DIR INH INH IX1 IX SP1 | 3A 4A 5A 6A 7A 9E6A | dd ff ff ff | 4 1 1 4 3 5 |
| DIV | Divide | $A \leftarrow (H:A)/(X)$ 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 | $A \leftarrow (A \oplus M)$ | 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 ff ee ff | 2 3 4 4 3 2 4 5 |
| INC opr INCA INCA INC opr,X INC ,X INC opr,SP INC opr,SP | Increment | $M \leftarrow (M) + 1$ $A \leftarrow (A) + 1$ $X \leftarrow (X) + 1$ $M \leftarrow (M) + 1$ $M \leftarrow (M) + 1$ $M \leftarrow (M) + 1$ | ↕ | - | - | ↕ | ↕ | - | DIR INH INH IX1 IX SP1 | 3C 4C 5C 6C 7C 9E6C | dd dd ff ff ff | 4 1 1 4 3 5 |
| JMP opr JMP opr JMP opr,X JMP opr,X JMP ,X | Jump | $PC \leftarrow \text{Jump Address}$ | - | - | - | - | - | - | 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 | $PC \leftarrow (PC) + n$ ($n = 1, 2, \text{ or } 3$) Push (PCL); $SP \leftarrow (SP) - 1$ Push (PCH); $SP \leftarrow (SP) - 1$ $PC \leftarrow \text{Unconditional Address}$ | - | - | - | - | - | - | 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 | $A \leftarrow (M)$ | 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 ff ee ff | 2 3 4 4 3 2 4 5 |
| LDHX #opr LDHX opr | Load H:X from M | $H:X \leftarrow (M:M + 1)$ | 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 | $X \leftarrow (M)$ | 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 ff ee ff | 2 3 4 4 3 2 4 5 |
| LSL opr LSLA LSLX LSL opr,X LSL ,X LSL opr,SP LSL opr,SP | Logical Shift Left (Same as ASL) | | ↕ | - | - | ↕ | ↕ | ↕ | DIR INH INH IX1 IX SP1 | 38 48 58 68 78 9E68 | dd dd ff 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 <i>opr</i> LSRA LSRX LSR <i>opr</i> ,X LSR ,X LSR <i>opr</i> ,SP | Logical Shift Right | | ↓ | - | - | 0 | ↓ | ↓ | DIR INH IX1 IX SP1 | 34 44 54 64 74 9E64 | dd ff ff | 4 1 1 4 3 5 |
| MOV <i>opr</i> , <i>opr</i> MOV <i>opr</i> ,X+ MOV # <i>opr</i> , <i>opr</i> MOV X+, <i>opr</i> | Move | $(M)_{\text{Destination}} \leftarrow (M)_{\text{Source}}$ $H:X \leftarrow (H:X) + 1 \text{ (IX+D, DIX+)}$ | 0 | - | - | ↓ | ↓ | - | DD DIX+ IMD IX+D | 4E 5E 6E 7E | dd dd dd ii dd dd | 5 4 4 4 |
| MUL | Unsigned multiply | $X:A \leftarrow (X) \times (A)$ | - | 0 | - | - | - | 0 | INH | 42 | | 5 |
| NEG <i>opr</i> NEGA NEGX NEG <i>opr</i> ,X NEG ,X NEG <i>opr</i> ,SP | Negate (Two's Complement) | $M \leftarrow -(M) = \$00 - (M)$ $A \leftarrow -(A) = \$00 - (A)$ $X \leftarrow -(X) = \$00 - (X)$ $M \leftarrow -(M) = \$00 - (M)$ $M \leftarrow -(M) = \$00 - (M)$ | ↓ | - | - | ↓ | ↓ | ↓ | 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 | $A \leftarrow (A[3:0]:A[7:4])$ | - | - | - | - | - | - | INH | 62 | | 3 |
| ORA # <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ,X ORA <i>opr</i> ,X ORA ,X ORA <i>opr</i> ,SP ORA <i>opr</i> ,SP | Inclusive OR A and M | $A \leftarrow (A) (M)$ | 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 ff ee ff | 2 3 4 4 3 2 4 5 |
| PSHA | Push A onto Stack | Push (A); $SP \leftarrow (SP) - 1$ | - | - | - | - | - | - | INH | 87 | | 2 |
| PSHH | Push H onto Stack | Push (H); $SP \leftarrow (SP) - 1$ | - | - | - | - | - | - | INH | 8B | | 2 |
| PSHX | Push X onto Stack | Push (X); $SP \leftarrow (SP) - 1$ | - | - | - | - | - | - | INH | 89 | | 2 |
| PULA | Pull A from Stack | $SP \leftarrow (SP + 1)$; Pull (A) | - | - | - | - | - | - | INH | 86 | | 2 |
| PULH | Pull H from Stack | $SP \leftarrow (SP + 1)$; Pull (H) | - | - | - | - | - | - | INH | 8A | | 2 |
| PULX | Pull X from Stack | $SP \leftarrow (SP + 1)$; Pull (X) | - | - | - | - | - | - | INH | 88 | | 2 |
| ROL <i>opr</i> ROLA ROLX ROL <i>opr</i> ,X ROL ,X ROL <i>opr</i> ,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 <i>opr</i> RORA RORX ROR <i>opr,X</i> ROR , <i>X</i> ROR <i>opr,SP</i> | 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 # <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> SBC <i>opr,X</i> SBC <i>opr,X</i> SBC , <i>X</i> SBC <i>opr,SP</i> SBC <i>opr,SP</i> | 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 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 <i>opr</i> STA <i>opr</i> STA <i>opr,X</i> STA <i>opr,X</i> STA , <i>X</i> STA <i>opr,SP</i> STA <i>opr,SP</i> | 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 ff ee ff | 3 4 4 3 2 4 5 |
| STHX <i>opr</i> | 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 <i>opr</i> STX <i>opr</i> STX <i>opr,X</i> STX <i>opr,X</i> STX , <i>X</i> STX <i>opr,SP</i> STX <i>opr,SP</i> | 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 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 | $A \leftarrow (A) - (M)$ | ‡ | - | - | ‡ | ‡ | ‡ | IMM DIR EXT IX2 IX1 IX SP1 SP2 | A0 B0 C0 D0 E0 F0 9EE0 9ED0 | ii dd hh ll ee ff ff ff ff ee ff | 2 3 4 4 3 2 4 5 |
| SWI | Software Interrupt | $PC \leftarrow (PC) + 1$; Push (PCL) $SP \leftarrow (SP) - 1$; Push (PCH) $SP \leftarrow (SP) - 1$; Push (X) $SP \leftarrow (SP) - 1$; Push (A) $SP \leftarrow (SP) - 1$; Push (CCR) $SP \leftarrow (SP) - 1$; $I \leftarrow 1$ PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte | - | - | 1 | - | - | - | INH | 83 | | 9 |
| TAP | Transfer A to CCR | $CCR \leftarrow (A)$ | ‡ | ‡ | ‡ | ‡ | ‡ | ‡ | INH | 84 | | 2 |
| TAX | Transfer A to X | $X \leftarrow (A)$ | - | - | - | - | - | - | INH | 97 | | 1 |
| TPA | Transfer CCR to A | $A \leftarrow (CCR)$ | - | - | - | - | - | - | 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 | $H:X \leftarrow (SP) + 1$ | - | - | - | - | - | - | INH | 95 | | 2 |
| TXA | Transfer X to A | $A \leftarrow (X)$ | - | - | - | - | - | - | INH | 9F | | 1 |
| TXS | Transfer H:X to SP | $(SP) \leftarrow (H:X) - 1$ | - | - | - | - | - | - | 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 | <i>n</i> | | | | | | | | | | |
| C | Carry/borrow bit | <i>opr</i> | | | | | | | | | | |
| CCR | Condition code register | PC | | | | | | | | | | |
| dd | Direct address of operand | PCH | | | | | | | | | | |
| dd rr | Direct address of operand and relative offset of branch instruction | PCL | | | | | | | | | | |
| DD | Direct to direct addressing mode | REL | | | | | | | | | | |
| DIR | Direct addressing mode | <i>rel</i> | | | | | | | | | | |
| DIX+ | Direct to indexed with post increment addressing mode | <i>rr</i> | | | | | | | | | | |
| ee ff | High and low bytes of offset in indexed, 16-bit offset addressing | SP1 | | | | | | | | | | |
| EXT | Extended addressing mode | SP2 | | | | | | | | | | |
| ff | Offset byte in indexed, 8-bit offset addressing | SP | | | | | | | | | | |
| H | Half-carry bit | U | | | | | | | | | | |
| H | Index register high byte | V | | | | | | | | | | |
| hh ll | High and low bytes of operand address in extended addressing | X | | | | | | | | | | |
| I | Interrupt mask | Z | | | | | | | | | | |
| ii | Immediate operand byte | & | | | | | | | | | | |
| IMD | Immediate source to direct destination addressing mode | | | | | | | | | | | |
| IMM | Immediate addressing mode | ⊕ | | | | | | | | | | |
| INH | Inherent addressing mode | () | | | | | | | | | | |
| IX | Indexed, no offset addressing mode | -() | | | | | | | | | | |
| IX+ | Indexed, no offset, post increment addressing mode | # | | | | | | | | | | |
| IX+D | Indexed with post increment to direct addressing mode | « | | | | | | | | | | |
| IX1 | Indexed, 8-bit offset addressing mode | ← | | | | | | | | | | |
| IX1+ | Indexed, 8-bit offset, post increment addressing mode | ? | | | | | | | | | | |
| IX2 | Indexed, 16-bit offset addressing mode | : | | | | | | | | | | |
| M | Memory location | ‡ | | | | | | | | | | |
| N | Negative bit | — | | | | | | | | | | |

Table 6-2. Opcode Map

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

| | | |
|-----|----------------------|--|
| MSB | 0 | High Byte of Opcode in Hexadecimal |
| LSB | 5 BRSET0 3 DIR | Cycles Opcode Mnemonic Number of Bytes / Addressing Mode |



Section 7. System Integration Module (SIM)

7.1 Contents

| | | |
|---------|---|-----|
| 7.2 | Introduction | 88 |
| 7.3 | SIM Bus Clock Control and Generation | 91 |
| 7.3.1 | Bus Timing | 91 |
| 7.3.2 | Clock Start-Up from POR | 91 |
| 7.3.3 | Clocks in Stop Mode and Wait Mode | 91 |
| 7.4 | Reset and System Initialization | 92 |
| 7.4.1 | External Pin Reset | 92 |
| 7.4.2 | Active Resets from Internal Sources | 93 |
| 7.4.2.1 | Power-On Reset | 94 |
| 7.4.2.2 | Computer Operating Properly (COP) Reset | 95 |
| 7.4.2.3 | Illegal Opcode Reset | 96 |
| 7.4.2.4 | Illegal Address Reset | 96 |
| 7.5 | SIM Counter | 96 |
| 7.5.1 | SIM Counter During Power-On Reset | 96 |
| 7.5.2 | SIM Counter During Stop Mode Recovery | 97 |
| 7.5.3 | SIM Counter and Reset States | 97 |
| 7.6 | Exception Control | 97 |
| 7.6.1 | Interrupts | 98 |
| 7.6.1.1 | Hardware Interrupts | 100 |
| 7.6.1.2 | SWI Instruction | 101 |
| 7.6.2 | Interrupt Status Registers | 101 |
| 7.6.2.1 | Interrupt Status Register 1 | 103 |
| 7.6.2.2 | Interrupt Status Register 2 | 103 |
| 7.6.3 | Reset | 104 |
| 7.6.4 | Break Interrupts | 104 |
| 7.6.5 | Status Flag Protection in Break Mode | 104 |
| 7.7 | Low-Power Modes | 105 |

| | | |
|-------|---|-----|
| 7.7.1 | Wait Mode | 105 |
| 7.7.2 | Stop Mode | 106 |
| 7.8 | SIM Registers | 108 |
| 7.8.1 | SIM Break Status Register (SBSR) | 108 |
| 7.8.2 | SIM Reset Status Register (SRSR) | 109 |
| 7.8.3 | SIM Break Flag Control Register (SBFCR) | 110 |

7.2 Introduction

This section describes the system integration module, which supports up to 16 external and/or internal interrupts. Together with the CPU, the SIM controls all MCU activities. A block diagram of the SIM is shown in [Figure 7-1](#). [Table 7-1](#) shows a summary of the SIM 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

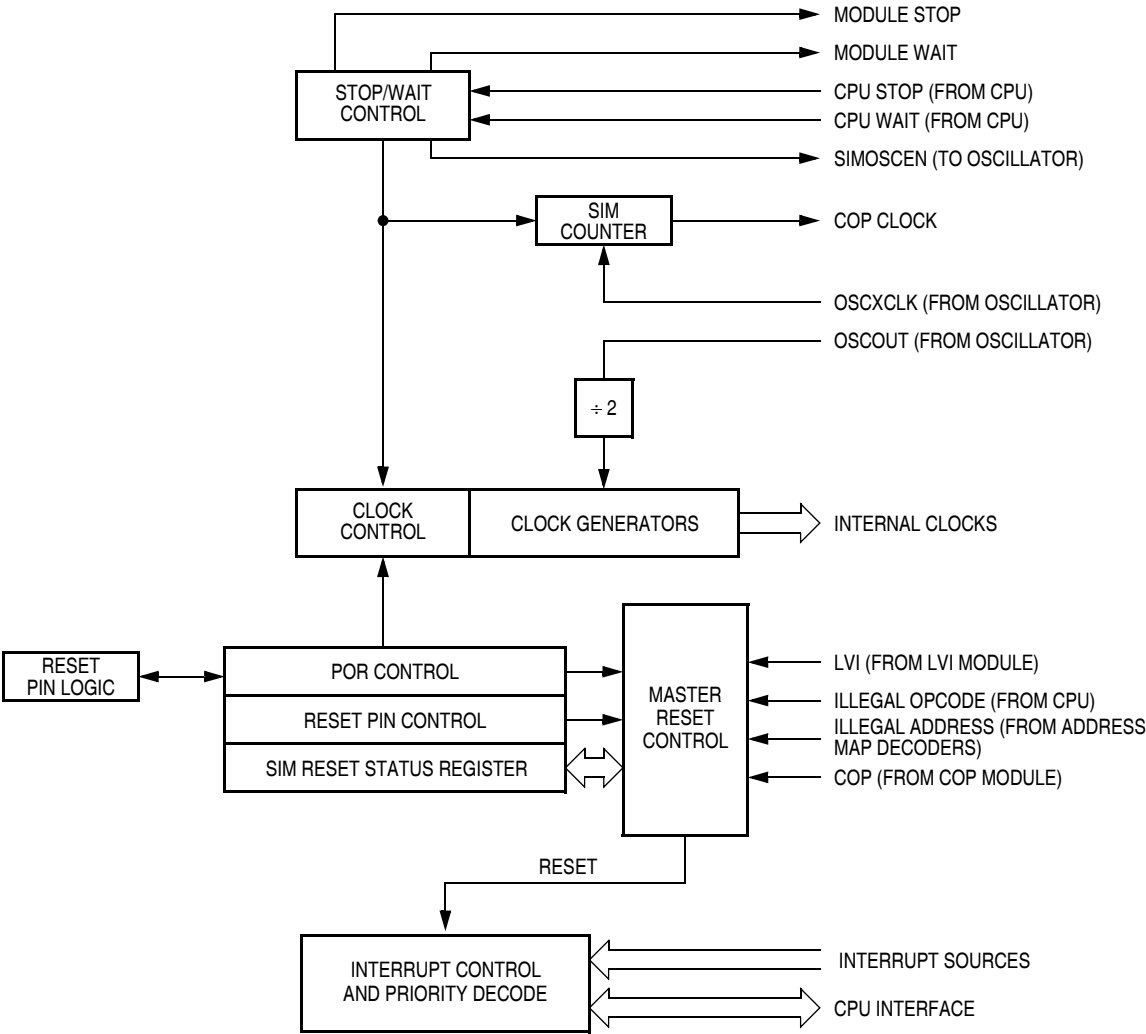


Figure 7-1. SIM Block Diagram

Table 7-1. SIM I/O Register Summary

| 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 | SBSW | R |
| | | Write: | | | | | | Note | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE01 | SIM Reset Status Register (SRSR) | Read: | POR | PIN | COP | ILOP | ILAD | 0 | 0 |
| | | Write: | | | | | | | |
| | | POR: | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE03 | SIM Break Flag Control Register (SBFCR) | Read: | BCFE | R | R | R | R | R | R |
| | | Write: | | | | | | | |
| | | Reset: | 0 | | | | | | |
| \$FE04 | Interrupt Status Register 1 (INT1) | Read: | IF6 | IF5 | IF4 | IF3 | IF2 | IF1 | 0 |
| | | Write: | R | R | R | R | R | R | R |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$FE05 | Interrupt Status Register 2 (INT2) | Read: | 0 | 0 | 0 | 0 | IF10 | IF9 | IF8 |
| | | Write: | R | R | R | R | R | R | R |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: Writing a logic 0 clears SBSW.

= Unimplemented R = Reserved

Table 7-2 shows the internal signal names used in this section.

Table 7-2. Signal Name Conventions

| Signal Name | Description |
|-------------------|---|
| OSCXCLK | Buffered version of OSC1 from the oscillator |
| OSCOUT | The OSCXCLK frequency divided by two. This signal is again divided by two in the SIM to generate the internal bus clocks. (Bus clock = OSCXCLK divided by four) |
| 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 |

7.3 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, OSCOUT, as shown in [Figure 7-2](#).

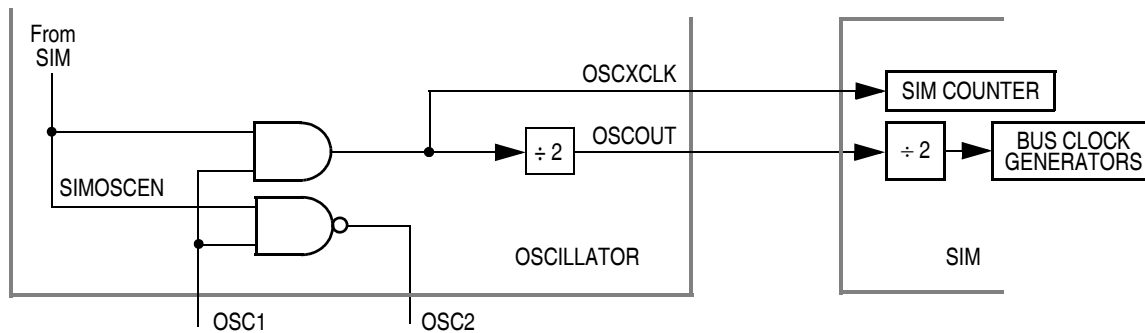


Figure 7-2. OSC Clock Signals

7.3.1 Bus Timing

In user mode, the internal bus frequency is the oscillator frequency (OSCXCLK) divided by four.

7.3.2 Clock Start-Up from POR

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

7.3.3 Clocks in Stop Mode and Wait Mode

Upon exit from stop mode (by an interrupt, break, or reset), the SIM allows OSCXCLK 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 OSCXCLK cycles. (See [7.7.2 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.

7.4 Reset and System Initialization

The MCU has the following reset sources:

- Power-on reset module (POR)
- External reset pin (\overline{RST})
- Computer operating properly module (COP)
- 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 [7.5 SIM Counter](#)), but an external reset does not. Each of the resets sets a corresponding bit in the SIM reset status register (SRSR) (see [7.8 SIM Registers](#)).

7.4.1 External Pin Reset

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 OSCXCLK cycles, assuming that the POR was the source of the reset (see [Table 7-3. PIN Bit Set Timing](#)). [Figure 7-3](#) shows the relative timing.

Table 7-3. PIN Bit Set Timing

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

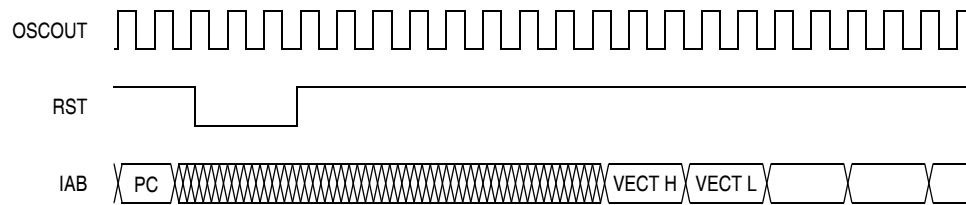


Figure 7-3. External Reset Timing

7.4.2 Active Resets from Internal Sources

SIM module in HC08 has the capability to drive the $\overline{\text{RST}}$ pin low when internal reset events occur.

All internal reset sources actively pull the $\overline{\text{RST}}$ pin low for 32 OSCXCLK cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles (see [Figure 7-4. Internal Reset Timing](#)). An internal reset can be caused by an illegal address, illegal opcode, COP timeout, or POR (see [Figure 7-5. Sources of Internal Reset](#)). Note that for POR resets, the SIM cycles through 4096 OSCXCLK cycles during which the SIM forces the $\overline{\text{RST}}$ pin low. The internal reset signal then follows the sequence from the falling edge of $\overline{\text{RST}}$ shown in [Figure 7-4](#).

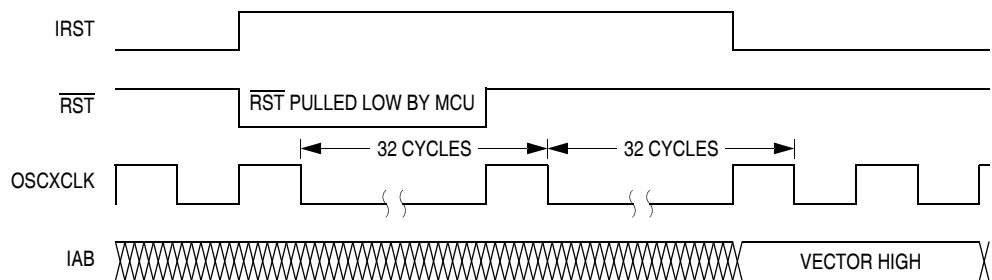


Figure 7-4. Internal Reset Timing

The COP reset is asynchronous to the bus clock.

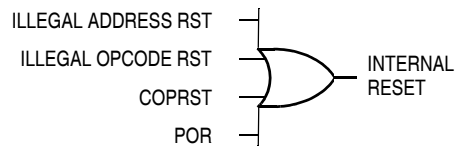


Figure 7-5. Sources of Internal Reset

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

7.4.2.1 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 ($\overline{\text{RST}}$) is held low while the SIM counter counts out 4096 OSCXCLK cycles. Sixty-four OSCXCLK cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur.

At power-on, the following events occur:

- A POR pulse is generated.
- The internal reset signal is asserted.
- The SIM enables the oscillator to drive OSCXCLK.
- Internal clocks to the CPU and modules are held inactive for 4096 OSCXCLK cycles to allow stabilization of the oscillator.
- The $\overline{\text{RST}}$ 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.

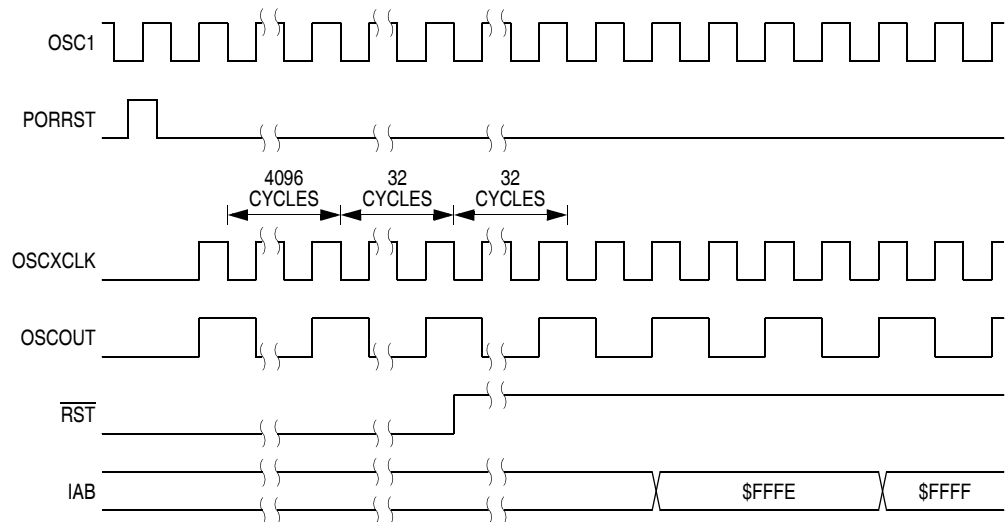


Figure 7-6. POR Recovery

7.4.2.2 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 $2^{12} - 2^4$ OSCXCLK 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 $\overline{\text{IRQ}}$ 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 $\overline{\text{RST}}$ pin or the $\overline{\text{IRQ}}$ 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.

7.4.2.3 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 configure register 1 (CONFIG1) is logic zero, 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.

7.4.2.4 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.

7.5 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 OSCXCLK.

7.5.1 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 oscillator to drive the bus clock state machine.

7.5.2 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 configure register 1 (CONFIG1). If the SSREC bit is a logic one, then the stop recovery is reduced from the normal delay of 4096 OSCXCLK cycles down to 32 OSCXCLK 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.

7.5.3 SIM Counter and Reset States

External reset has no effect on the SIM counter ([see 7.7.2 Stop Mode](#)). The SIM counter is free-running after all reset states ([see 7.4.2 Active Resets from Internal Sources](#) for counter control and internal reset recovery sequences).

7.6 Exception Control

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

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

7.6.1 Interrupts

An interrupt temporarily changes the sequence of program execution to respond to a particular event. **Figure 7-9** flow charts the handling of system interrupts.

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).

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 7-7** shows interrupt entry timing. **Figure 7-8** shows interrupt recovery timing.

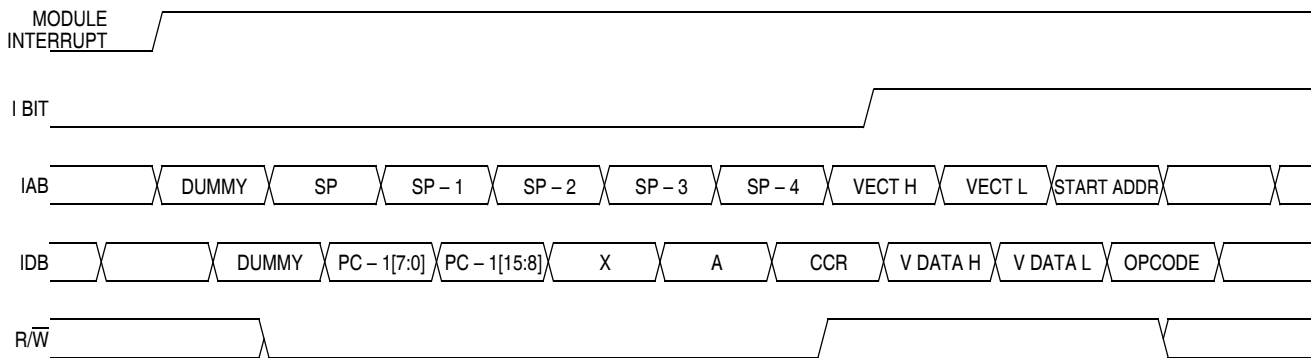


Figure 7-7. Interrupt Entry

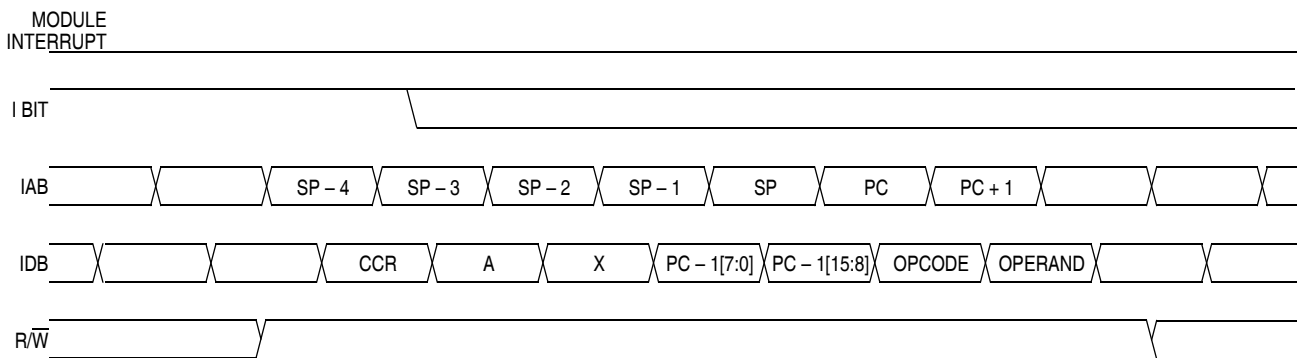


Figure 7-8. Interrupt Recovery

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 may take precedence, regardless of priority, until the latched interrupt is serviced (or the I bit is cleared). (See [Figure 7-9. Interrupt Processing.](#))

7.6.1.1 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 7-10](#) 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.

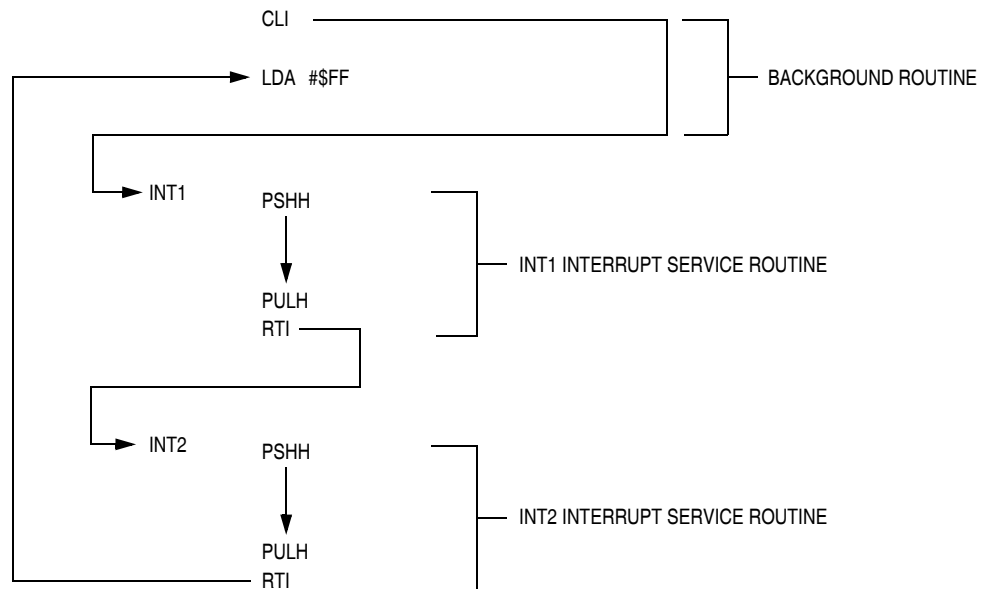


Figure 7-10. Interrupt Recognition Example

The LDA opcode is pre-fetched by both the INT1 and INT2 RTI instructions. However, in the case of the INT1 RTI pre-fetch, 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.*

7.6.1.2 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.*

7.6.2 Interrupt Status Registers

The flags in the interrupt status registers identify maskable interrupt sources. [Table 7-4](#) summarizes the interrupt sources and the interrupt status register flags that they set. The interrupt status registers can be useful for debugging.

Table 7-4. Interrupt Sources

| Source | Flag | Mask ¹ | INT Register Flag | Priority ² | Vector Address |
|-------------------------|---------|-------------------|-------------------|-----------------------|----------------|
| Reset | None | None | None | 0 | \$FFFE-\$FFFF |
| SWI Instruction | None | None | None | 0 | \$FFFC-\$FFFD |
| IRQ pin | IRQF | IMASK | IF1 | 1 | \$FFFA-\$FFFB |
| USB | TBEF | TBIE | IF2 | 2 | \$FFF8-\$FFF9 |
| | RBFF | RBIE | | | |
| | EOPIF | EOPIE | | | |
| | RSTIF | RSTIE | | | |
| | TXD1F | TXD1IE | | | |
| DDC12AB | ALIF | DIEN | IF3 | 3 | \$FFF6-\$FFF7 |
| | NAKIF | | | | |
| | RXIF | | | | |
| | TXIF | | | | |
| | SCLIF | SCLIEN | | | |
| Reserved | — | — | — | — | \$FFF4-\$FFF5 |
| TIM channel 0 | CH0F | CH0IE | IF5 | 5 | \$FFF2-\$FFF3 |
| TIM channel 1 | CH1F | CH1IE | IF6 | 6 | \$FFF0-\$FFF1 |
| TIM overflow | TOF | TOIE | IF7 | 7 | \$FFEE-\$FFEF |
| Sync processor | VSIF | VSIE | IF8 | 8 | \$FFEC-\$FFED |
| | LVSIF | LVSIE | | | |
| Multi-master IIC | MMALIF | MMIEN | IF9 | 9 | \$FFEA-\$FFEB |
| | MMNAKIF | | | | |
| | MMRXIF | | | | |
| | MMTXIF | | | | |
| ADC conversion complete | COCO | AIEN | IF10 | 10 | \$FFE8-\$FFE9 |
| Reserved | — | — | — | — | \$FFE6-\$FFE7 |

1. The I bit in the condition code register is a global mask for all interrupts sources except the SWI instruction.
2. 0 = highest priority

7.6.2.1 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 7-11. 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 7-4](#).

1 = Interrupt request present

0 = No interrupt request present

Bit 1 and Bit 0 — Always read 0

7.6.2.2 Interrupt Status Register 2

Address: \$FE05

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|---|---|---|------|-----|-----|-------|
| Read: | 0 | 0 | 0 | 0 | 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 7-12. Interrupt Status Register 2 (INT2)

IF10–IF7 — Interrupt Flags 6–1

These flags indicate the presence of interrupt requests from the sources shown in [Table 7-4](#).

1 = Interrupt request present

0 = No interrupt request present

Bit 7 and Bit 4 — Always read 0

7.6.3 Reset

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

7.6.4 Break Interrupts

The break module can stop normal program flow at a software-programmable break point by asserting its break interrupt output (see [Section 20. 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.

7.6.5 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 two-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.

7.7 Low-Power Modes

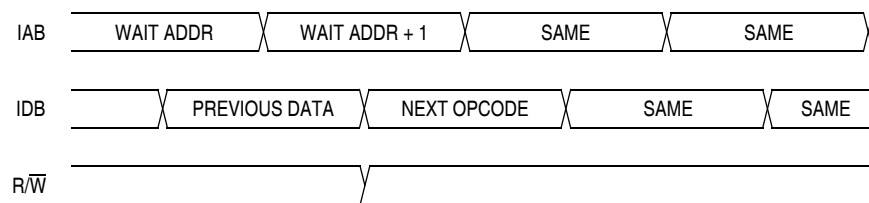
Executing the WAIT or STOP instruction puts the MCU in a low-power-consumption mode for standby situations. The SIM holds the CPU in a non-clocked state. The operation of each of these modes is described below. Both STOP and WAIT clear the interrupt mask (I) in the condition code register, allowing interrupts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

7.7.1 Wait Mode

In wait mode, the CPU clocks are inactive while the peripheral clocks continue to run. [Figure 7-13](#) 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.

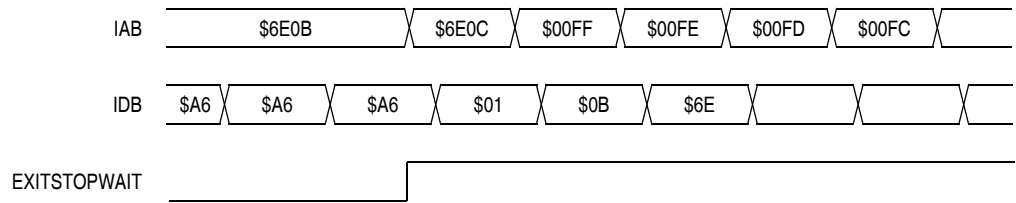
Wait mode can also be exited by a reset or break. A break interrupt during wait mode sets the SIM break stop/wait bit, SBSW, in the SIM break status register (SBSR). If the COP disable bit, COPD, in configuration register 1 (CONFIG1) is logic zero, then the computer operating properly module (COP) is enabled and remains active in wait mode.



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

Figure 7-13. Wait Mode Entry Timing

[Figure 7-14](#) and [Figure 7-15](#) show the timing for WAIT recovery.



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

Figure 7-14. Wait Recovery from Interrupt or Break

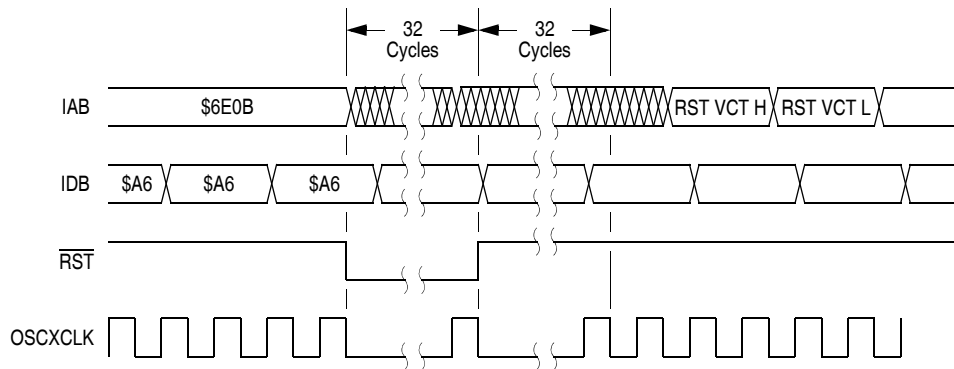


Figure 7-15. Wait Recovery from Internal Reset

7.7.2 Stop Mode

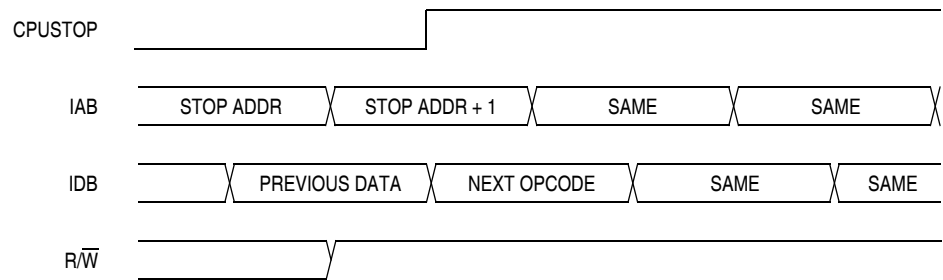
In stop mode, the SIM counter is reset and the system clocks are disabled. An interrupt request from a module can cause an exit from stop mode. Stacking for interrupts begins after the selected stop recovery time has elapsed. Reset or break also causes an exit from stop mode.

The SIM disables the oscillator signals (OSCOOUT and OSCXCLK) in stop mode, stopping the CPU and peripherals. Stop recovery time is selectable using the SSREC bit in configuration register 1 (CONFIG1). If SSREC is set, stop recovery is reduced from the normal delay of 4096 OSCXCLK 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 7-16** shows stop mode entry timing.



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

Figure 7-16. Stop Mode Entry Timing

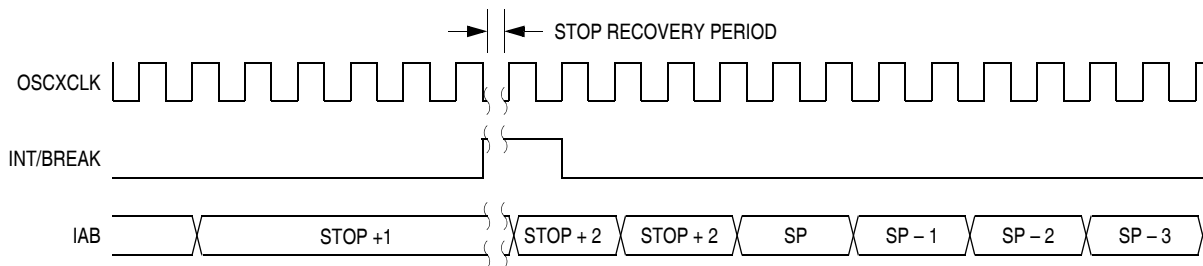


Figure 7-17. Stop Mode Recovery from Interrupt or Break

7.8 SIM Registers

The SIM has three memory mapped registers. [Table 7-5](#) shows the mapping of these registers.

Table 7-5. SIM Registers Summary

| Address | Register | Access Mode |
|---------|----------|-------------|
| \$FE00 | SBSR | User |
| \$FE01 | SRSR | User |
| \$FE03 | SBFCR | User |

7.8.1 SIM Break Status Register (SBSR)

The SIM break status register contains a flag to indicate that a break caused an exit from stop 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: | R | R | R | R | R | R | Note | R |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

Figure 7-18. SIM Break Status Register (SBSR)

SBSW — SIM Break Stop/Wait Bit

This status bit is useful in applications requiring a return to wait or stop mode after exiting from a break interrupt. 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 one 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
    
```

7.8.2 SIM Reset Status Register (SRSR)

This register contains six flags that show the source of the last reset. 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 | 0 | 0 |
| Write: | | | | | | | | |
| POR: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |


 = Unimplemented

Figure 7-19. 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{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

7.8.3 SIM Break Flag Control Register (SBFCR)

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

Address: \$FE03

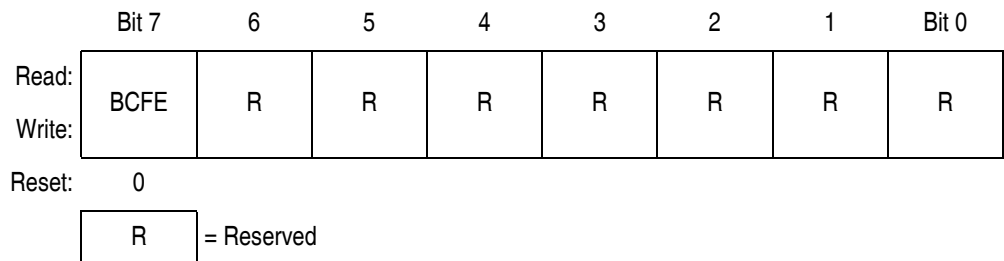


Figure 7-20. SIM Break Flag Control Register (SBFCR)

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 8. Oscillator (OSC)

8.1 Contents

| | | |
|-------|---|-----|
| 8.2 | Introduction | 111 |
| 8.3 | Oscillator External Connections | 112 |
| 8.4 | I/O Signals | 113 |
| 8.4.1 | Crystal Amplifier Input Pin (OSC1) | 113 |
| 8.4.2 | Crystal Amplifier Output Pin (OSC2) | 113 |
| 8.4.3 | Oscillator Enable Signal (SIMOSCEN) | 113 |
| 8.4.4 | External Clock Source (OSCXCLK) | 113 |
| 8.4.5 | Oscillator Out (OSCOUT) | 113 |
| 8.5 | Low Power Modes | 114 |
| 8.5.1 | Wait Mode | 114 |
| 8.5.2 | Stop Mode | 114 |
| 8.6 | Oscillator During Break Mode | 114 |

8.2 Introduction

The oscillator circuit is designed for use with crystals or ceramic resonators. The oscillator circuit generates the crystal clock signal, OSCXCLK, at the frequency of the crystal. This signal is divided by two before being passed on to the SIM for bus clock generation. **Figure 8-1** shows the structure of the oscillator. The oscillator requires various external components.

8.3 Oscillator External Connections

In its typical configuration, the oscillator requires five external components. The crystal oscillator is normally connected in a Pierce oscillator configuration, as shown in **Figure 8-1**. 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)

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.

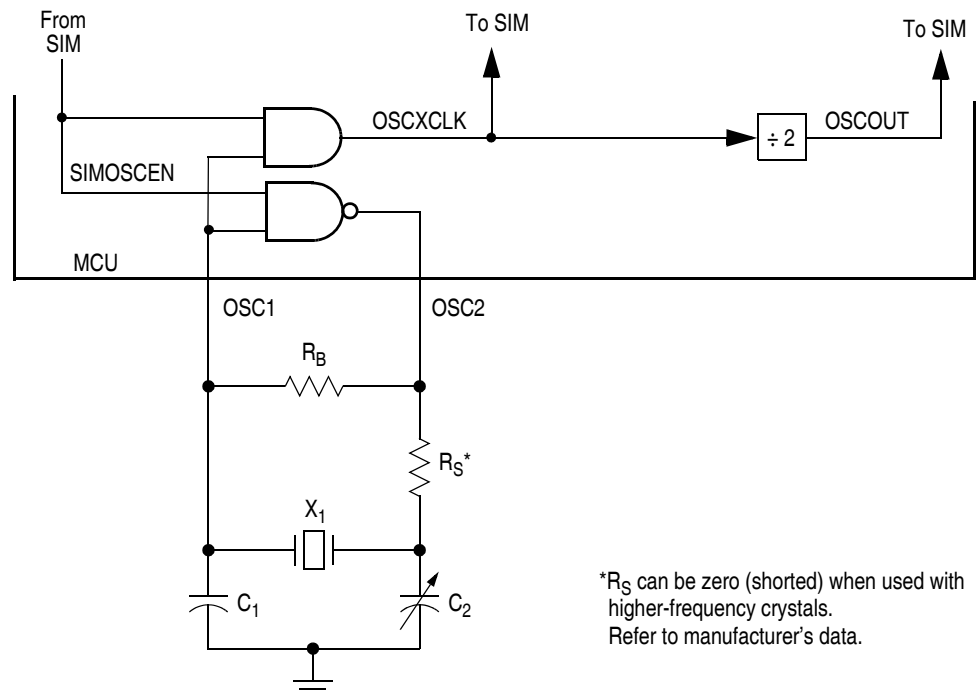


Figure 8-1. Oscillator External Connections

8.4 I/O Signals

The following paragraphs describe the oscillator I/O signals.

8.4.1 Crystal Amplifier Input Pin (OSC1)

The OSC1 pin is an input to the crystal oscillator amplifier.

An externally generated clock also can feed the OSC1 pin of the crystal oscillator circuit. Connect the external clock to the OSC1 pin and let the OSC2 pin float.

8.4.2 Crystal Amplifier Output Pin (OSC2)

The OSC2 pin is the output of the crystal oscillator inverting amplifier.

8.4.3 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal comes from the SIM and enables the oscillator.

8.4.4 External Clock Source (OSCXCLK)

OSCXCLK is the crystal oscillator output signal. It runs at the full speed of the crystal (f_{XCLK}) and comes directly from the crystal oscillator circuit. **Figure 8-1** shows only the logical relation of OSCXCLK to OSC1 and OSC2 and may not represent the actual circuitry. The duty cycle of OSCXCLK is unknown and may depend on the crystal and other external factors. Also, the frequency and amplitude of OSCXCLK can be unstable at start-up.

8.4.5 Oscillator Out (OSCOUT)

The clock driven to the SIM is the crystal frequency divided by two. This signal is driven to the SIM for generation of the bus clocks used by the CPU and other modules on the MCU. OSCOUT will be divided again in the SIM and results in the internal bus frequency being one fourth of the OSCXCLK frequency.

8.5 Low Power Modes

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

8.5.1 Wait Mode

The WAIT instruction has no effect on the oscillator logic. OSCXCLK continues to drive to the SIM module.

8.5.2 Stop Mode

The STOP instruction disables the OSCXCLK output.

8.6 Oscillator During Break Mode

The oscillator continues drive OSCXCLK when the chip enters the break state.

Section 9. Monitor ROM (MON)

9.1 Contents

| | | |
|-------|----------------------------------|-----|
| 9.2 | Introduction | 115 |
| 9.3 | Features | 116 |
| 9.4 | Functional Description | 116 |
| 9.4.1 | Entering Monitor Mode | 118 |
| 9.4.2 | Data Format | 119 |
| 9.4.3 | Echoing | 120 |
| 9.4.4 | Break Signal | 120 |
| 9.4.5 | Commands | 121 |
| 9.4.6 | Baud Rate | 124 |

9.2 Introduction

This section describes the monitor ROM. The monitor ROM allows complete testing of the MCU through a single-wire interface with a host computer.

9.3 Features

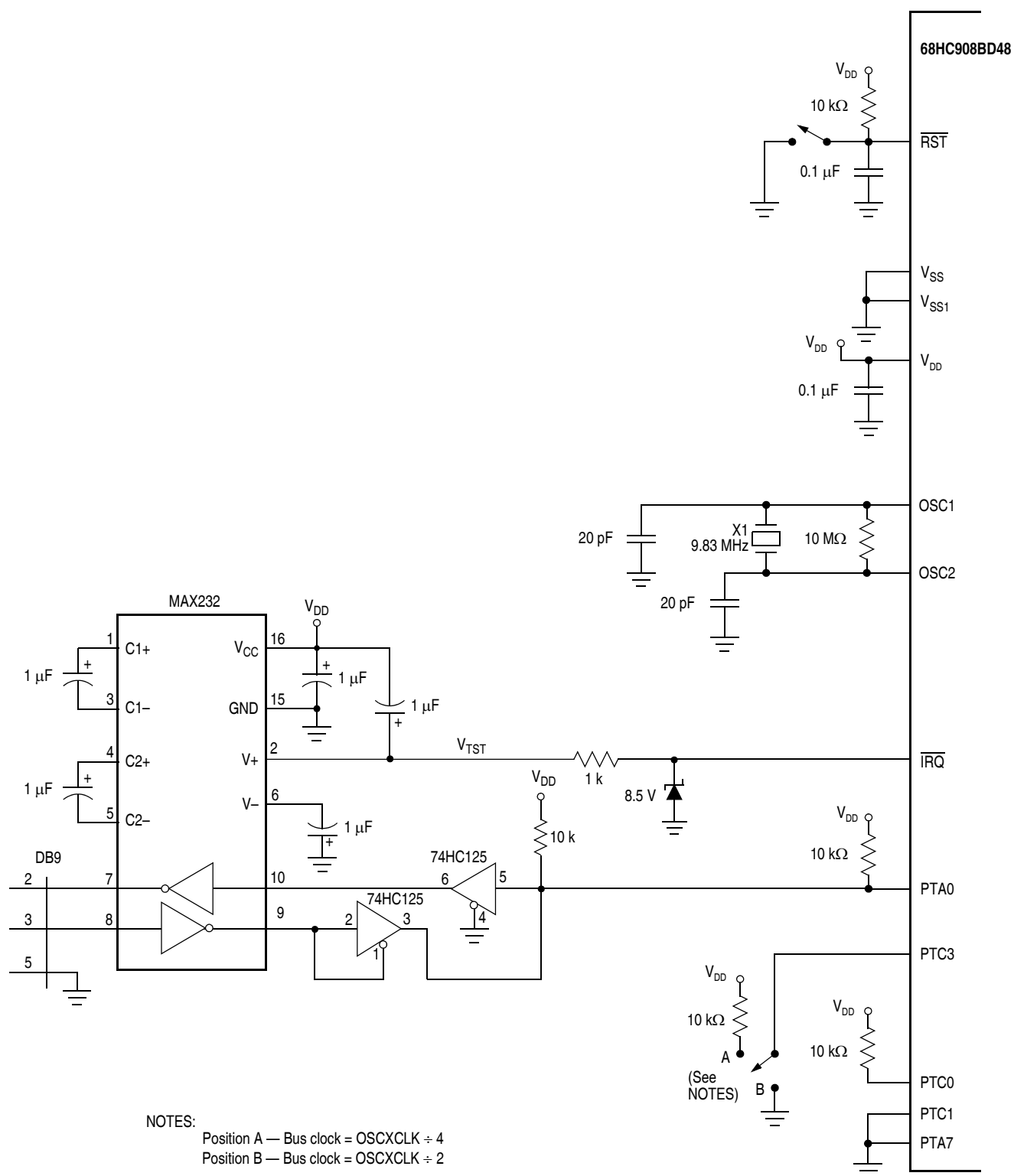
Features of the monitor ROM include:

- 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
- 9600 Baud communication with host computer
- Execution of code in RAM or FLASH
- FLASH memory programming

9.4 Functional Description

The monitor ROM receives and executes commands from a host computer. [Figure 9-1](#) shows a sample 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 host-computer code in RAM while all 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 pull-up resistor.


Figure 9-1. Monitor Mode Circuit

9.4.1 Entering Monitor Mode

Table 9-1 shows the pin conditions for entering monitor mode.

Table 9-1. Mode Selection

| IRQ Pin | PTC0 Pin | PTC1 Pin | PTA7 Pin | PTA0 Pin | PTC3 Pin | Mode | OSCOUT | Bus Frequency |
|------------------|----------|----------|----------|----------|----------|---------|--------------------|--------------------|
| V _{TST} | 1 | 0 | 0 | 1 | 1 | Monitor | $\frac{OSCCLK}{2}$ | $\frac{OSCCLK}{4}$ |
| V _{TST} | 1 | 0 | 0 | 1 | 0 | Monitor | OSCCLK | $\frac{OSCCLK}{2}$ |

NOTE: Holding the PTC3 pin low when entering monitor mode causes a bypass of a divide-by-two stage at the oscillator. The OSCOUT frequency is equal to the OSCXCLK 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.

Enter monitor mode with the pin configuration shown above 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 monitor mode firmware then sends a break signal (10 consecutive logic zeros) to the host computer, indicating that it is ready to receive a command. The break signal also provides a timing reference to allow the host to determine the necessary baud rate.

Monitor mode uses different vectors for reset and SWI. 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.

When the host computer has completed downloading code into the MCU RAM, This code can be executed by driving PTA0 low while asserting \overline{RST} low and then high. The internal monitor ROM firmware will interpret the low on PTA0 as an indication to jump to RAM, and execution control will then continue from RAM. Execution of an SWI from the downloaded code will return program control to the internal monitor ROM firmware.

Alternatively, the host can send a RUN command, which executes an RTI, and this can be used to send control to the address on the stack pointer.

The COP module is disabled in monitor mode as long as V_{TST} is applied to the \overline{IRQ} or the \overline{RST} pin. (See [Section 7. System Integration Module \(SIM\)](#) for more information on modes of operation.)

Table 9-2 is a summary of the differences between user mode and monitor mode.

Table 9-2. Mode Differences

| Modes | Functions | | | | |
|---------|-------------------------|-------------------|------------------|-----------------|----------------|
| | COP | Reset Vector High | Reset Vector Low | SWI Vector High | SWI Vector Low |
| User | Enabled | \$FFFE | \$FFFF | \$FFFC | \$FFFD |
| Monitor | Disabled ⁽¹⁾ | \$FEFE | \$FEFF | \$FEFC | \$FEFD |

Notes:

1. If the high voltage (V_{TST}) is removed from the \overline{IRQ} pin, the SIM asserts its COP enable output. The COP is a mask option enabled or disabled by the COPD bit in the configuration register.

9.4.2 Data Format

Communication with the monitor ROM is in standard non-return-to-zero (NRZ) mark/space data format. (See [Figure 9-2](#) and [Figure 9-3](#).)

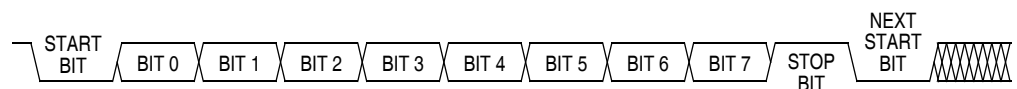


Figure 9-2. Monitor Data Format

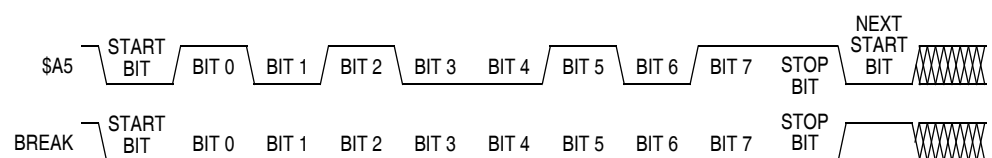


Figure 9-3. Sample Monitor Waveforms

The data transmit and receive rate can be anywhere from 4800 baud to 28.8 kbaud. Transmit and receive baud rates must be identical.

9.4.3 Echoing

As shown in [Figure 9-4](#), the monitor ROM immediately echoes each received byte back to the PTA0 pin for error checking.

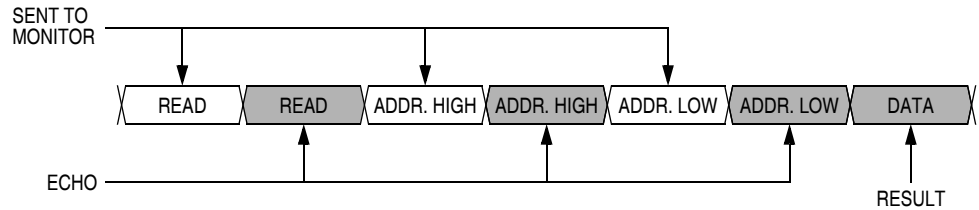


Figure 9-4. Read Transaction

Any result of a command appears after the echo of the last byte of the command.

9.4.4 Break Signal

A start bit followed by nine low bits is a break signal (see [Figure 9-5](#)). When the monitor receives a break signal, it drives the PTA0 pin high for the duration of two bits before echoing the break signal.

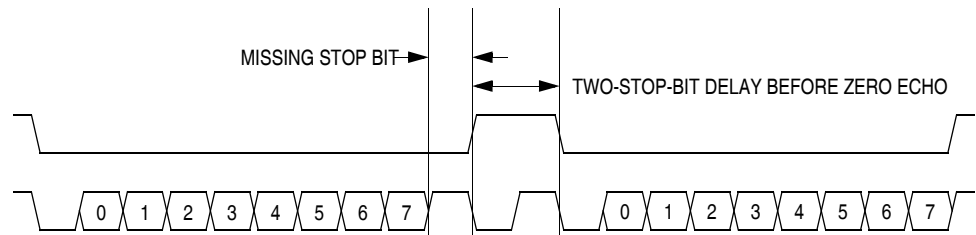


Figure 9-5. Break Transaction

9.4.5 Commands

The monitor ROM uses the following commands:

- READ (read memory)
- WRITE (write memory)
- IREAD (indexed read)
- IWRITE (indexed write)
- READSP (read stack pointer)
- RUN (run user program)

Table 9-3. READ (Read Memory) Command

| | |
|---|--|
| Description | Read byte from memory |
| Operand | Specifies 2-byte address in high byte:low byte order |
| Data Returned | Returns contents of specified address |
| Opcode | \$4A |
| Command Sequence | |
| <p>The diagram illustrates the timing sequence for the READ command. It starts with a 'SENT TO MONITOR' signal pointing to the first 'READ' pulse. This is followed by a second 'READ' pulse, which is labeled 'ECHO'. Then, the address is provided in two parts: 'ADDRESS HIGH' and 'ADDRESS LOW'. Finally, the 'DATA' returned from memory is shown as a pulse labeled 'RETURN'.</p> | |

Table 9-4. WRITE (Write Memory) Command

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

Table 9-5. IREAD (Indexed Read) Command

| | |
|-------------------------|--|
| Description | Read Next 2 Bytes in Memory from Last Address Accessed |
| Operand | Specifies 2-byte address in high byte:low byte order |
| Data Returned | Returns contents of next two addresses |
| Opcode | \$1A |
| Command Sequence | |
| | |

Table 9-6. IWRITE (Indexed Write) Command

| | |
|-------------------------|------------------------------------|
| Description | Write to last address accessed + 1 |
| Operand | Specifies single data byte |
| Data Returned | None |
| Opcode | \$19 |
| Command Sequence | |
| | |

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

Table 9-7. READSP (Read Stack Pointer) Command

| | |
|-------------------------|---|
| Description | Reads stack pointer |
| Operand | None |
| Data Returned | Returns stack pointer in high byte:low byte order |
| Opcode | \$0C |
| Command Sequence | |
| | |

Table 9-8. RUN (Run User Program) Command

| | |
|--------------------------------|--------------------------|
| Description | Executes RTI instruction |
| Operand | None |
| Data Returned | None |
| Opcode | \$28 |
| <p>Command Sequence</p> | |

9.4.6 Baud Rate

The communication baud rate is controlled by crystal frequency and the state of the PTC3 pin upon entry into monitor mode. When PTC3 is high, the divide by ratio is 1024. If the PTC3 pin is at logic zero upon entry into monitor mode, the divide by ratio is 512.

Table 9-9. Monitor Baud Rate Selection

| Crystal Frequency | PTC3 Pin | Baud Rate |
|-------------------|----------|-----------|
| 19.66 MHz | 0 | 19200 bps |
| 9.83 MHz | 0 | 9600 bps |
| 9.83 MHz | 1 | 4800 bps |

Section 10. Timer Interface Module (TIM)

10.1 Contents

| | | |
|----------|--|-----|
| 10.2 | Introduction | 126 |
| 10.3 | Features | 126 |
| 10.4 | Pin Name Conventions | 126 |
| 10.5 | Functional Description | 127 |
| 10.5.1 | TIM Counter Prescaler | 129 |
| 10.5.2 | Input Capture | 129 |
| 10.5.3 | Output Compare | 129 |
| 10.5.3.1 | Unbuffered Output Compare | 130 |
| 10.5.3.2 | Buffered Output Compare | 131 |
| 10.5.4 | Pulse Width Modulation (PWM) | 131 |
| 10.5.4.1 | Unbuffered PWM Signal Generation | 132 |
| 10.5.4.2 | Buffered PWM Signal Generation | 133 |
| 10.5.4.3 | PWM Initialization | 134 |
| 10.6 | Interrupts | 135 |
| 10.7 | Wait Mode | 135 |
| 10.8 | TIM During Break Interrupts | 136 |
| 10.9 | I/O Signals | 136 |
| 10.10 | I/O Registers | 137 |
| 10.10.1 | TIM Status and Control Register (TSC) | 137 |
| 10.10.2 | TIM Counter Registers (TCNTH:TCNTL) | 139 |
| 10.10.3 | TIM Counter Modulo Registers (TMODH:TMODL) | 140 |
| 10.10.4 | TIM Channel Status and Control Registers (TSC0:TSC1) | 141 |
| 10.10.5 | TIM Channel Registers (TCH0H/L:TCH1H/L) | 145 |

10.2 Introduction

This section describes the timer interface module (TIM2, version B). The TIM is a two-channel timer that provides a timing reference with input capture, output compare, and pulse-width-modulation functions. [Figure 10-1](#) is a block diagram of the TIM.

10.3 Features

Features of the TIM include the following:

- 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

NOTE: *TCH1 (timer channel 1) is not bonded to an external pin on this MCU. Therefore, any references to the timer TCH1 pin in the following text should be interpreted as not available — but the internal status and control registers are still available.*

10.4 Pin Name Conventions

The TIM share one I/O pin with one port E I/O pin. The full name of the TIM I/O pin is listed in [Table 10-1](#). The generic pin name appear in the text that follows.

Table 10-1. Pin Name Conventions

| TIM Generic Pin Names: | TCH0 | TCH1 |
|------------------------|---------------|---------------|
| Full TIM Pin Names: | PTE0/SOG/TCH0 | Not Available |

10.5 Functional Description

Figure 10-1 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 are programmable independently as input capture or output compare channels.

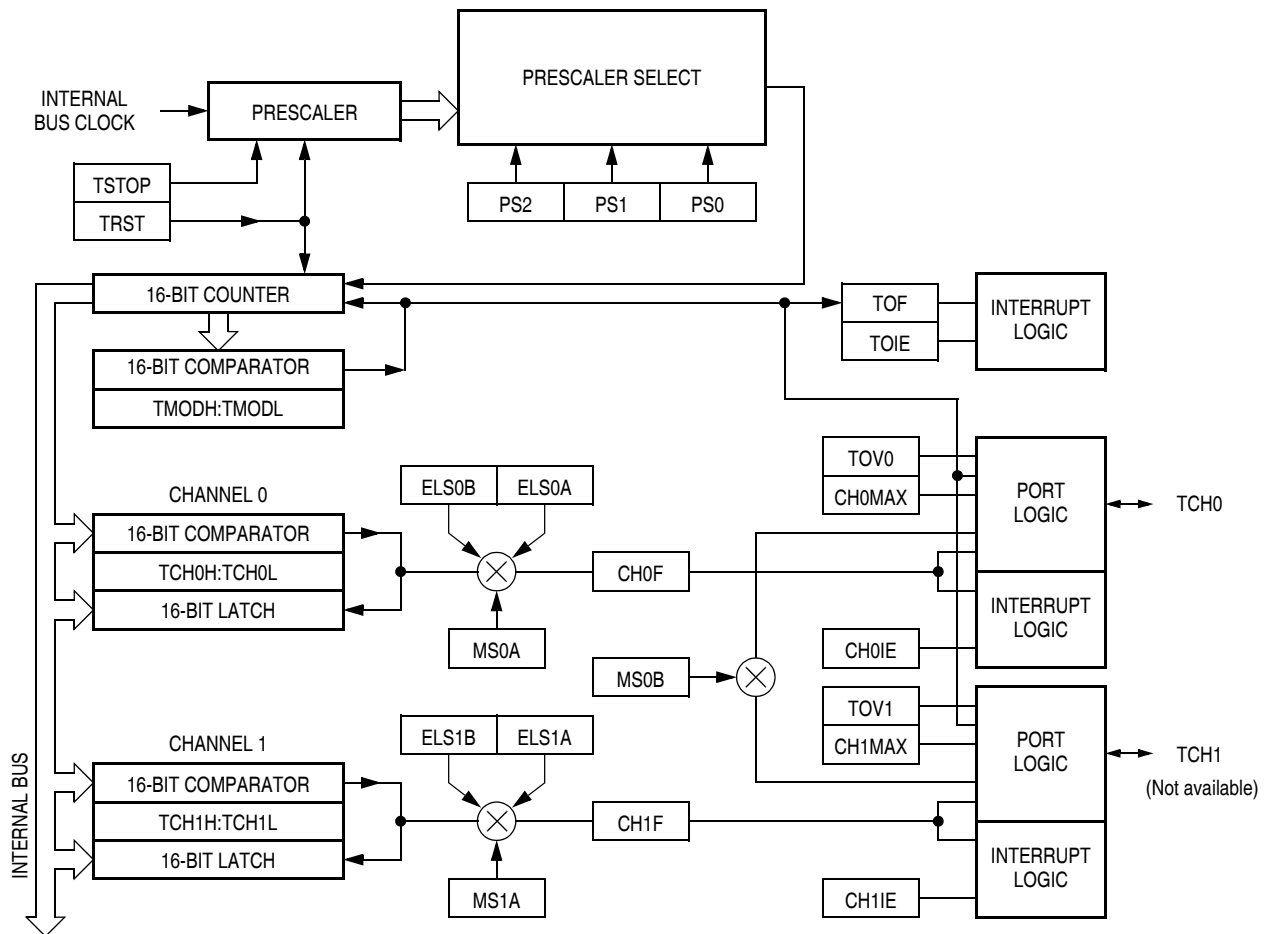
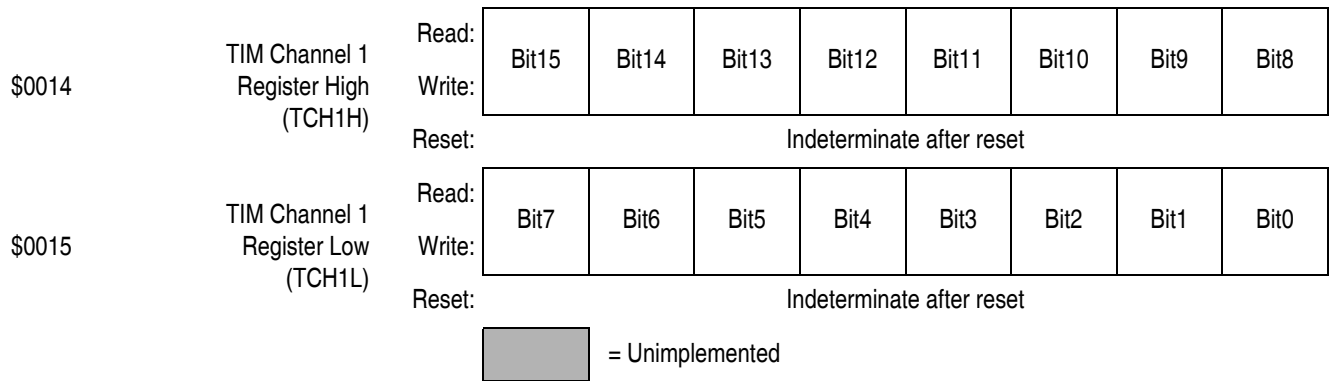


Figure 10-1. TIM Block Diagram

Table 10-2. TIM I/O Register Summary

| Addr. | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|--|--------|---------------------------|-------|-------|-------|-------|-------|------|--------|
| \$000A | TIM Status and Control Register (TSC) | Read: | TOF | TOIE | TSTOP | 0 | 0 | PS2 | PS1 | PS0 |
| | | Write: | 0 | | | TRST | | | | |
| | | Reset: | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| \$000C | TIM Counter Register High (TCNTH) | Read: | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$000D | TIM Counter Register Low (TCNTL) | Read: | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$000E | TIM Counter Modulo Register High (TMODH) | Read: | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$000F | TIM Counter Modulo Register Low (TMODL) | Read: | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$0010 | TIM Channel 0 Status/Control Register (TSC0) | Read: | CH0F | CH0IE | MS0B | MS0A | ELS0B | ELS0A | TOV0 | CH0MAX |
| | | Write: | 0 | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0011 | TIM Channel 0 Register High (TCH0H) | Read: | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
| | | Write: | | | | | | | | |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0012 | TIM Channel 0 Register Low (TCH0L) | Read: | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| | | Write: | | | | | | | | |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0013 | TIM Channel 1 Status/Control Register (TSC1) | Read: | CH1F | CH1IE | 0 | MS1A | ELS1B | ELS1A | TOV1 | CH1MAX |
| | | Write: | 0 | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



10.5.1 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 (TSC) select the TIM clock source.

10.5.2 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.

10.5.3 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.

10.5.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in [10.5.3 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.

10.5.3.2 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.*

10.5.4 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 10-2](#) 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 one. Program the TIM to set the pin if the state of the PWM pulse is logic zero.

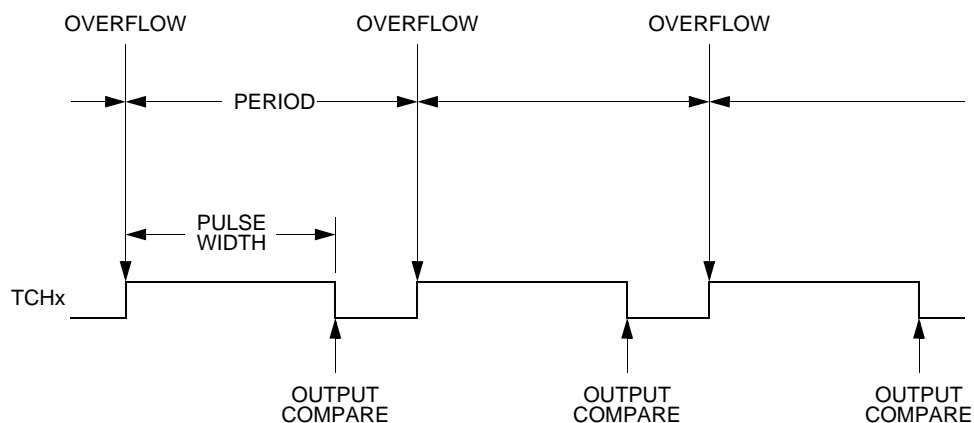


Figure 10-2. PWM Period and Pulse Width

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 [10.10.1 TIM Status and Control Register \(TSC\)](#)).

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%.

10.5.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in [10.5.4 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.*

10.5.4.2 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.*

10.5.4.3 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 10-4](#).)
 - 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 10-4](#).)

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 channel 0 status and control register (TSC0) 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 [10.10.4 TIM Channel Status and Control Registers \(TSC0:TSC1\)](#).

10.6 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.

10.7 Wait Mode

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

10.8 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 break flag control register (BFCR) enables software to clear status bits during the break state. (See [7.8.3 SIM Break Flag Control Register \(SBFCR\)](#).)

To allow software to clear status bits during a break interrupt, write a logic one 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 zero to the BCFE bit. With BCFE at logic zero (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a two-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 zero. After the break, doing the second step clears the status bit.

10.9 I/O Signals

Port E shares one of its pins with the TIM. The TIM channel I/O pin is PTE0/SOG/TCH0.

TCH0 pin is programmable independently as an input capture pin or an output compare pin. It also can be configured as a buffered output compare or buffered PWM pin.

10.10 I/O Registers

The following 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 and TSC1)
- TIM channel registers (TCH0H:TCH0L and TCH1H:TCH1L)

10.10.1 TIM Status and Control Register (TSC)

The TIM status and control register does the following:

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

Address: \$000A

| | 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 10-3. 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 zero to TOF. If another TIM overflow occurs before the

clearing sequence is complete, then writing logic zero 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 one 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 zero. 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 either the TCLK pin or one of the seven prescaler outputs as the input to the TIM counter as [Table 10-3](#) shows. Reset clears the PS[2:0] bits.

Table 10-3. 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 |

10.10.2 TIM Counter Registers (TCNTH:TCNTL)

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.*

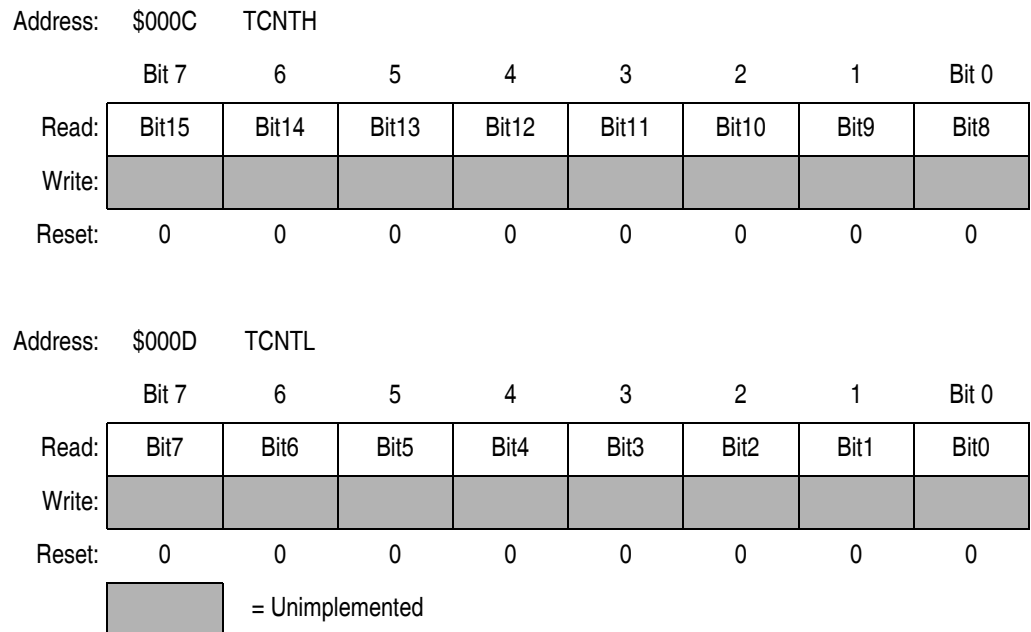


Figure 10-4. TIM Counter Registers (TCNTH:TCNTL)

10.10.3 TIM Counter Modulo Registers (TMODH:TMODL)

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.

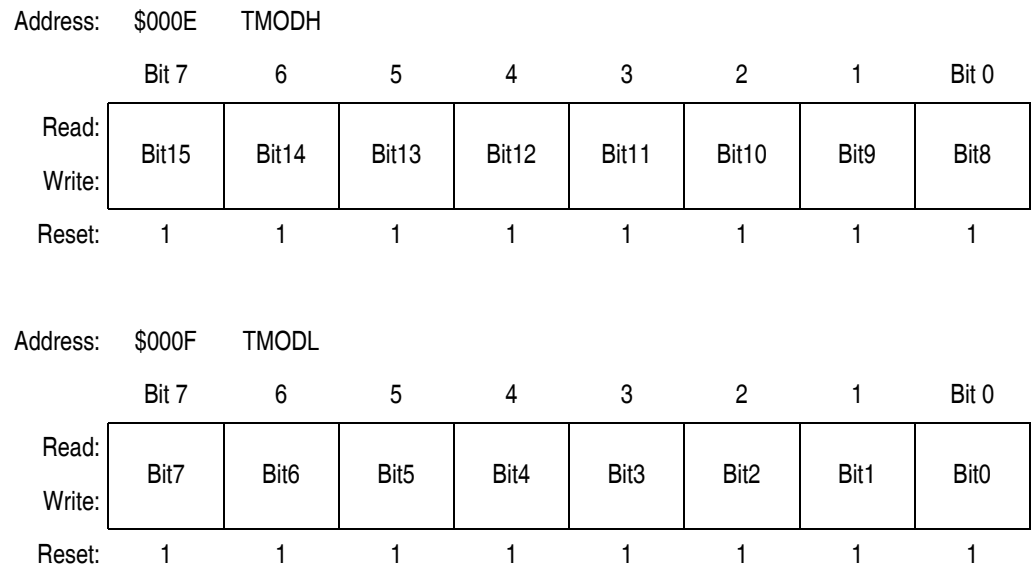


Figure 10-5. TIM Counter Modulo Registers (TMODH:TMODL)

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

10.10.4 TIM Channel Status and Control Registers (TSC0:TSC1)

Each of the TIM channel status and control registers does the following:

- 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

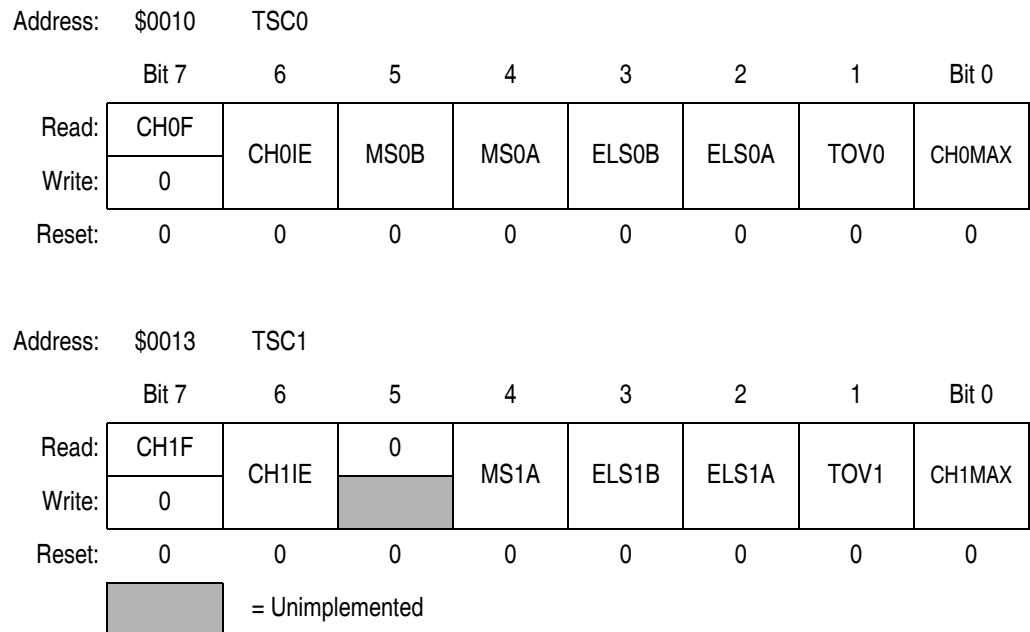


Figure 10-6. TIM Channel Status and Control Registers (TSC0:TSC1)

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 the TIM channel x status and control register with CHxF set and then writing a logic zero to CHxF. If another interrupt request occurs before the clearing sequence is complete, then writing logic zero 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 one 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 TIM channel 0 status and control register.

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:A \neq 00, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation.

See [Table 10-4](#).

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

When ELSxB:A = 00, this read/write bit selects the initial output level of the TCHx pin. (See [Table 10-4](#).) 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 port I/O pin. [Table 10-4](#) shows how ELSxB and ELSxA work. Reset clears the ELSxB and ELSxA bits.

Table 10-4. Mode, Edge, and Level Selection

| MSxB | MSxA | ELSxB | ELSxA | Mode | Configuration |
|------|------|-------|-------|---|---|
| X | 0 | 0 | 0 | Output Preset | Pin under Port Control; Initial Output Level High |
| X | 1 | 0 | 0 | | Pin under Port Control; Initial Output Level Low |
| 0 | 0 | 0 | 1 | Input Capture | Capture on Rising Edge Only |
| 0 | 0 | 1 | 0 | | Capture on Falling Edge Only |
| 0 | 0 | 1 | 1 | | Capture on Rising or Falling Edge |
| 0 | 1 | 0 | 1 | Output Compare or PWM | Toggle Output on Compare |
| 0 | 1 | 1 | 0 | | Clear Output on Compare |
| 0 | 1 | 1 | 1 | | Set Output on Compare |
| 1 | X | 0 | 1 | Buffered Output Compare or Buffered PWM | Toggle Output on Compare |
| 1 | X | 1 | 0 | | Clear Output on Compare |
| 1 | X | 1 | 1 | | 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 one, setting the CHxMAX bit forces the duty cycle of buffered and unbuffered PWM signals to 100%. As [Figure 10-7](#) 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.

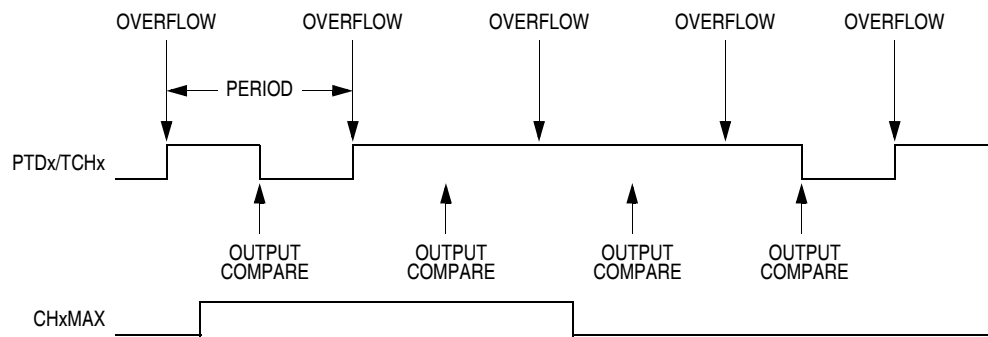


Figure 10-7. CHxMAX Latency

10.10.5 TIM Channel Registers (TCH0H/L:TCH1H/L)

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 ($MSxB:MSxA \neq 0:0$), writing to the high byte of the TIM channel x registers (TCHxH) inhibits output compares until the low byte (TCHxL) is written.

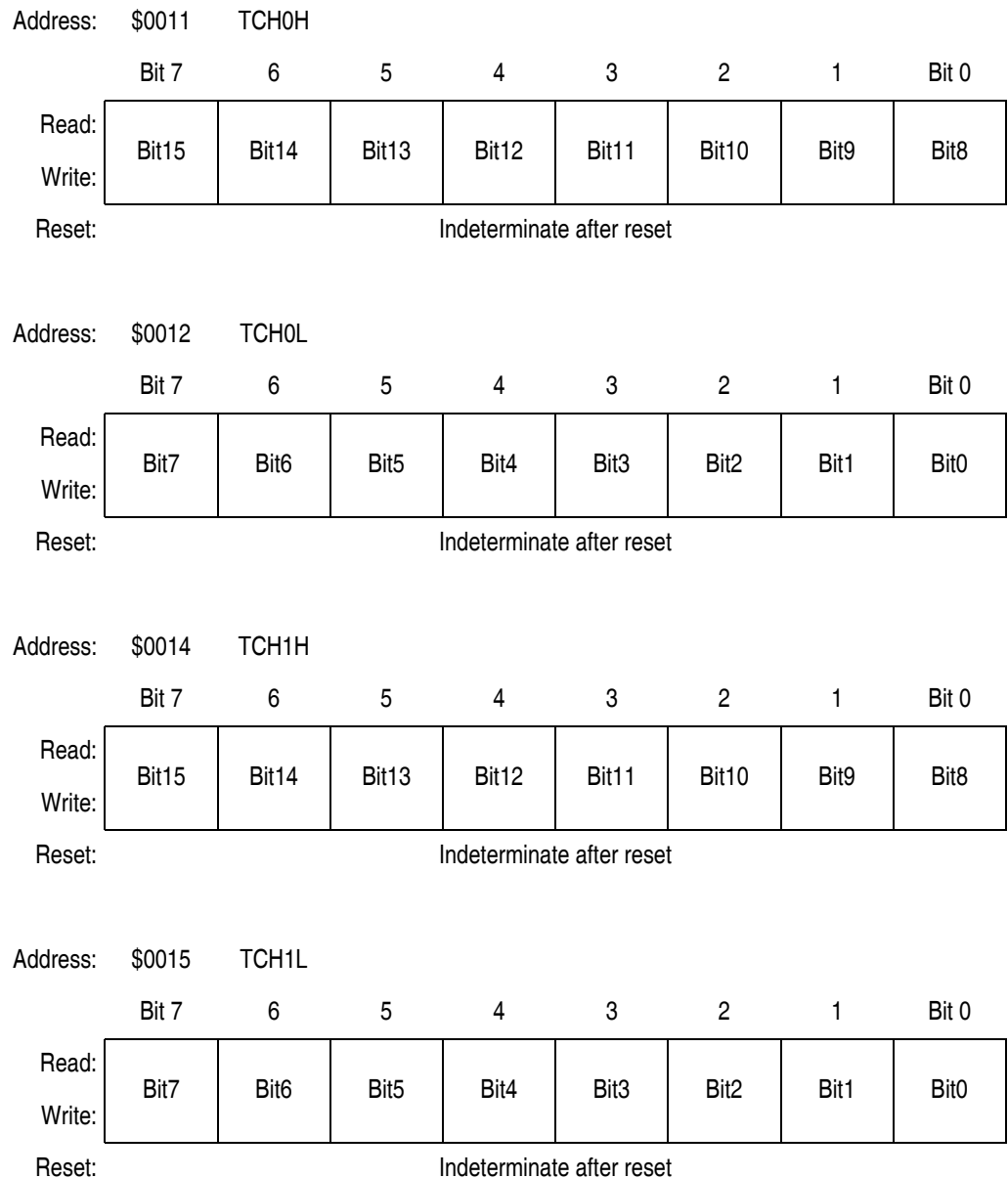


Figure 10-8. TIM Channel Registers (TCH0H/L:TCH1H/L)

Section 11. Pulse Width Modulator (PWM)

11.1 Contents

| | | |
|--------|---|-----|
| 11.2 | Introduction | 147 |
| 11.3 | Functional Description | 147 |
| 11.4 | PWM Registers. | 149 |
| 11.4.1 | PWM Data Registers 0 to 15 (0PWM–15PWM). | 150 |
| 11.4.2 | PWM Control Registers 1 and 2 (PWMCR1:PWMCR2) . . | 151 |

11.2 Introduction

Sixteen 8-bits PWM channels are available on the MC68HC908BD48. Channels 0 to 7 are shared with port-B I/O pins under the control of the PWM control register 1. Channels 8 to 15 are shared with port-A I/O pins under the control of the PWM control register 2.

11.3 Functional Description

Each 8-bit PWM channel is composed of an 8-bit register which contains a 5-bit PWM in MSB portion and a 3-bit binary rate multiplier (BRM) in LSB portion. There are 16 PWM data registers as shown in [Table 11-1](#). The value programmed in the 5-bit PWM portion will determine the pulse length of the output. The clock to the 5-bit PWM portion is the system clock, the repetition rate of the output is hence 187.5KHz at 6MHz clock.

The 3-bit BRM will generate a number of narrow pulses which are equally distributed among an 8-PWM-cycle frame. The number of pulses generated is equal to the number programmed in the 3-bit BRM portion. Examples of the waveforms are shown in [Figure 11-3](#).

Combining the 5-bit PWM together with the 3-bit BRM, the average duty cycle at the output will be $(M+N/8)/32$, where M is the content of the 5-bit PWM portion, and N is the content of the 3-bit BRM portion. Using this mechanism, a true 8-bit resolution PWM type DAC with reasonably high repetition rate can be obtained.

The value of each PWM Data Register is continuously compared with the content of an internal counter to determine the state of each PWM channel output pin. Double buffering is not used in this PWM design.

Table 11-1. PWM I/O Register Summary

| Addr. | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|---------------------------------|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| \$0020 | PWM0 Data Register (0PWM) | Read: | 0PWM4 | 0PWM3 | 0PWM2 | 0PWM1 | 0PWM0 | 0BRM2 | 0BRM1 | 0BRM0 |
| | | Write: | | | | | | | | |
| \$0021 | PWM1 Data Register (1PWM) | Read: | 1PWM4 | 1PWM3 | 1PWM2 | 1PWM1 | 1PWM0 | 1BRM2 | 1BRM1 | 1BRM0 |
| | | Write: | | | | | | | | |
| \$0022 | PWM2 Data Register (2PWM) | Read: | 2PWM4 | 2PWM3 | 2PWM2 | 2PWM1 | 2PWM0 | 2BRM2 | 2BRM1 | 2BRM0 |
| | | Write: | | | | | | | | |
| \$0023 | PWM3 Data Register (3PWM) | Read: | 3PWM4 | 3PWM3 | 3PWM2 | 3PWM1 | 3PWM0 | 3BRM2 | 3BRM1 | 3BRM0 |
| | | Write: | | | | | | | | |
| \$0024 | PWM4 Data Register (4PWM) | Read: | 4PWM4 | 4PWM3 | 4PWM2 | 4PWM1 | 4PWM0 | 4BRM2 | 4BRM1 | 4BRM0 |
| | | Write: | | | | | | | | |
| \$0025 | PWM5 Data Register (5PWM) | Read: | 5PWM4 | 5PWM3 | 5PWM2 | 5PWM1 | 5PWM0 | 5BRM2 | 5BRM1 | 5BRM0 |
| | | Write: | | | | | | | | |
| \$0026 | PWM6 Data Register (6PWM) | Read: | 6PWM4 | 6PWM3 | 6PWM2 | 6PWM1 | 6PWM0 | 6BRM2 | 6BRM1 | 6BRM0 |
| | | Write: | | | | | | | | |
| \$0027 | PWM7 Data Register (7PWM) | Read: | 7PWM4 | 7PWM3 | 7PWM2 | 7PWM1 | 7PWM0 | 7BRM2 | 7BRM1 | 7BRM0 |
| | | Write: | | | | | | | | |
| \$0028 | PWM Control Register 1 (PWMCR1) | Read: | PWM7E | PWM6E | PWM5E | PWM4E | PWM3E | PWM2E | PWM1E | PWM0E |
| | | Write: | | | | | | | | |
| Reset: | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 11-1. PWM I/O Register Summary

| | | | | | | | | | | |
|--------|---------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| \$0051 | PWM8 Data Register (8PWM) | Read: | 8PWM4 | 8PWM3 | 8PWM2 | 8PWM1 | 8PWM0 | 8BRM2 | 8BRM1 | 8BRM0 |
| | | Write: | | | | | | | | |
| \$0052 | PWM9 Data Register (9PWM) | Read: | 9PWM4 | 9PWM3 | 9PWM2 | 9PWM1 | 9PWM0 | 9BRM2 | 9BRM1 | 9BRM0 |
| | | Write: | | | | | | | | |
| \$0053 | PWM10 Data Register (10PWM) | Read: | 10PWM4 | 10PWM3 | 10PWM2 | 10PWM1 | 10PWM0 | 10BRM2 | 10BRM1 | 10BRM0 |
| | | Write: | | | | | | | | |
| \$0054 | PWM11 Data Register (11PWM) | Read: | 11PWM4 | 11PWM3 | 11PWM2 | 11PWM1 | 11PWM0 | 11BRM2 | 11BRM1 | 11BRM0 |
| | | Write: | | | | | | | | |
| \$0055 | PWM12 Data Register (12PWM) | Read: | 12PWM4 | 12PWM3 | 12PWM2 | 12PWM1 | 12PWM0 | 12BRM2 | 12BRM1 | 12BRM0 |
| | | Write: | | | | | | | | |
| \$0056 | PWM13 Data Register (13PWM) | Read: | 13PWM4 | 13PWM3 | 13PWM2 | 13PWM1 | 13PWM0 | 13BRM2 | 13BRM1 | 13BRM0 |
| | | Write: | | | | | | | | |
| \$0057 | PWM14 Data Register (14PWM) | Read: | 14PWM4 | PWM3 | 14PWM2 | 14PWM1 | 14PWM0 | 14BRM2 | 14BRM1 | 14BRM0 |
| | | Write: | | | | | | | | |
| \$0058 | PWM15 Data Register (15PWM) | Read: | 15PWM4 | 15PWM3 | 15PWM2 | 15PWM1 | 15PWM0 | 15BRM2 | 15BRM1 | 15BRM0 |
| | | Write: | | | | | | | | |
| \$0059 | PWM Control Register 2 (PWMCR2) | Read: | PWM15E | PWM14E | PWM13E | PWM12E | PWM11E | PWM10E | PWM9E | PWM8E |
| | | Write: | | | | | | | | |
| Reset: | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

11.4 PWM Registers

The PWM module uses of 18 registers for data and control functions.

- 16 PWM data registers (\$0020–\$0027 and \$0051–\$0058)
- 2 PWM control registers (\$0028 and \$0059)

11.4.1 PWM Data Registers 0 to 15 (0PWM–15PWM)

Address: \$0020–\$0027 and \$0051–\$0058

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Read: | xPWM4 | xPWM3 | xPWM2 | xPWM1 | xPWM0 | xBRM2 | xBRM1 | xBRM0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 11-1. PWM Data Registers 0 to 15 (0PWM–15PWM)

The output waveform of the 16 PWM channels are each configured by an 8-bit register, which contains a 5-bit PWM in MSB portion and a 3-bit binary rate multiplier (BRM) in LSB portion

xPWM4–xPWM0 — PWM Bits

The value programmed in the 5-bit PWM portion will determine the pulse length of the output. The clock to the 5-bit PWM portion is the system clock (CPU clock), the repetition rate of the output is hence $f_{OP} \div 32$. Examples of PWM output waveforms are shown in [Figure 11-3](#).

xBRM2–xBRM0 — Binary Rate Multiplier Bits

The 3-bit BRM will generate a number of narrow pulses which are equally distributed among an 8-PWM-cycle frame. The number of pulses generated is equal to the number programmed in the 3-bit BRM portion. Examples of PWM output waveforms are shown in [Figure 11-3](#).

11.4.2 PWM Control Registers 1 and 2 (PWMCR1:PWMCR2)

| | | | | | | | | | | |
|--------|---------------------------------|--------|--------|--------|--------|--------|--------|--------|-------|-------|
| \$0028 | PWM Control Register 1 (PWMCR1) | Read: | PWM7E | PWM6E | PWM5E | PWM4E | PWM3E | PWM2E | PWM1E | PWM0E |
| | | Write: | | | | | | | | |
| \$0059 | PWM Control Register 2 (PWMCR2) | Read: | PWM15E | PWM14E | PWM13E | PWM12E | PWM11E | PWM10E | PWM9E | PWM8E |
| | | Write: | | | | | | | | |
| Reset: | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 11-2. PWM Control Register 1 and 2 (PWMCR1:PWMCR2)

PWM15E–PWM0E — PWM Output Enable

Setting a bit to 1 will enable the corresponding PWM channel to use as PWM output. A zero configures the corresponding PWM pin as a standard I/O port pin. Reset clears these bits.

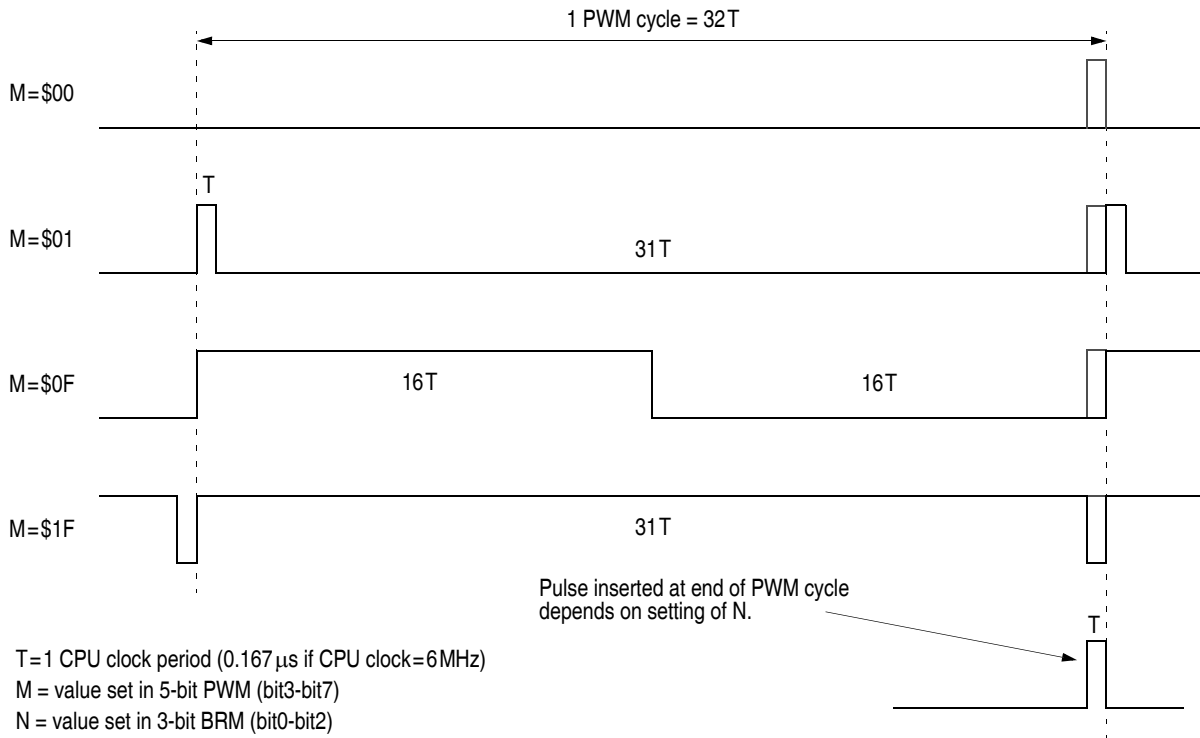
1 = Port pin configured as PWM output

0 = Port pin configured as standard I/O port pin.

Table 11-2. PWM Channels and Port I/O pins

| Port Pin | PWM Channel | Control Bit | Port Pin | PWM Channel | Control Bit |
|----------|-------------|-------------|----------|-------------|-------------|
| PTB0 | PWM0 | PWM0E | PTA0 | PWM8 | PWM8E |
| PTB1 | PWM1 | PWM1E | PTA1 | PWM9 | PWM9E |
| PTB2 | PWM2 | PWM2E | PTA2 | PWM10 | PWM10E |
| PTB3 | PWM3 | PWM3E | PTA3 | PWM11 | PWM11E |
| PTB4 | PWM4 | PWM4E | PTA4 | PWM12 | PWM12E |
| PTB5 | PWM5 | PWM5E | PTA5 | PWM13 | PWM13E |
| PTB6 | PWM6 | PWM6E | PTA6 | PWM14 | PWM14E |
| PTB7 | PWM7 | PWM7E | PTA7 | PWM15 | PWM15E |

Pulse Width Modulator (PWM)



| N | PWM cycles where pulses are inserted in a 8-cycle frame | Number of inserted pulses in a 8-cycle frame |
|-----|---|--|
| xx1 | 4 | 1 |
| x1x | 2, 6 | 2 |
| 1xx | 1, 3, 5, 7 | 4 |

Figure 11-3. 8-Bit PWM Output Waveforms

Section 12. Analog-to-Digital Converter (ADC)

12.1 Contents

| | | |
|--------|---------------------------------|-----|
| 12.2 | Introduction | 153 |
| 12.3 | Features | 154 |
| 12.4 | Functional Description | 154 |
| 12.4.1 | ADC Port I/O Pins | 155 |
| 12.4.2 | Voltage Conversion | 156 |
| 12.4.3 | Conversion Time | 156 |
| 12.4.4 | Continuous Conversion | 156 |
| 12.4.5 | Accuracy and Precision | 157 |
| 12.5 | Interrupts | 157 |
| 12.6 | Low-Power Modes | 157 |
| 12.6.1 | Wait Mode | 157 |
| 12.6.2 | Stop Mode | 157 |
| 12.7 | I/O Signals | 157 |
| 12.7.1 | ADC Voltage In (ADCVIN) | 158 |
| 12.8 | I/O Registers | 158 |
| 12.8.1 | ADC Status and Control Register | 158 |
| 12.8.2 | ADC Data Register | 160 |
| 12.8.3 | ADC Input Clock Register | 161 |

12.2 Introduction

This section describes the analog-to-digital converter (ADC). The ADC is an 8-bit 6-channels analog-to-digital converter.

12.3 Features

Features of the ADC module include:

- 6 Channels ADC with Multiplexed Input
- Linear Successive Approximation
- 8-Bit Resolution
- Single or Continuous Conversion
- Conversion Complete Flag or Conversion Complete Interrupt
- Selectable ADC Clock

Table 12-1. ADC Register Summary

| Addr. | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|---|--------|---------------------------|-------|-------|-------|-------|-------|-------|-------|
| \$005D | 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 |
| \$005E | ADC Data Register (ADR) | Read: | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | AD1 | AD0 |
| | | Write: | | | | | | | | |
| | | Reset: | Indeterminate after Reset | | | | | | | |
| \$005F | ADC Input Clock Register (ADICLK) | Read: | | | | 0 | 0 | 0 | 0 | 0 |
| | | Write: | ADIV2 | ADIV1 | ADIV0 | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

12.4 Functional Description

Six ADC channels are available for sampling external sources at pins PTC5–PTC0. An analog multiplexer allows the single ADC converter to select one of the 6 ADC channels as ADC voltage input (ADCVIN). ADCVIN is converted by the successive approximation register-based counters. The ADC resolution is 8 bits. When the conversion is completed, ADC puts the result in the ADC data register and sets a flag or generates an interrupt. [Figure 12-1](#) shows a block diagram of the ADC.

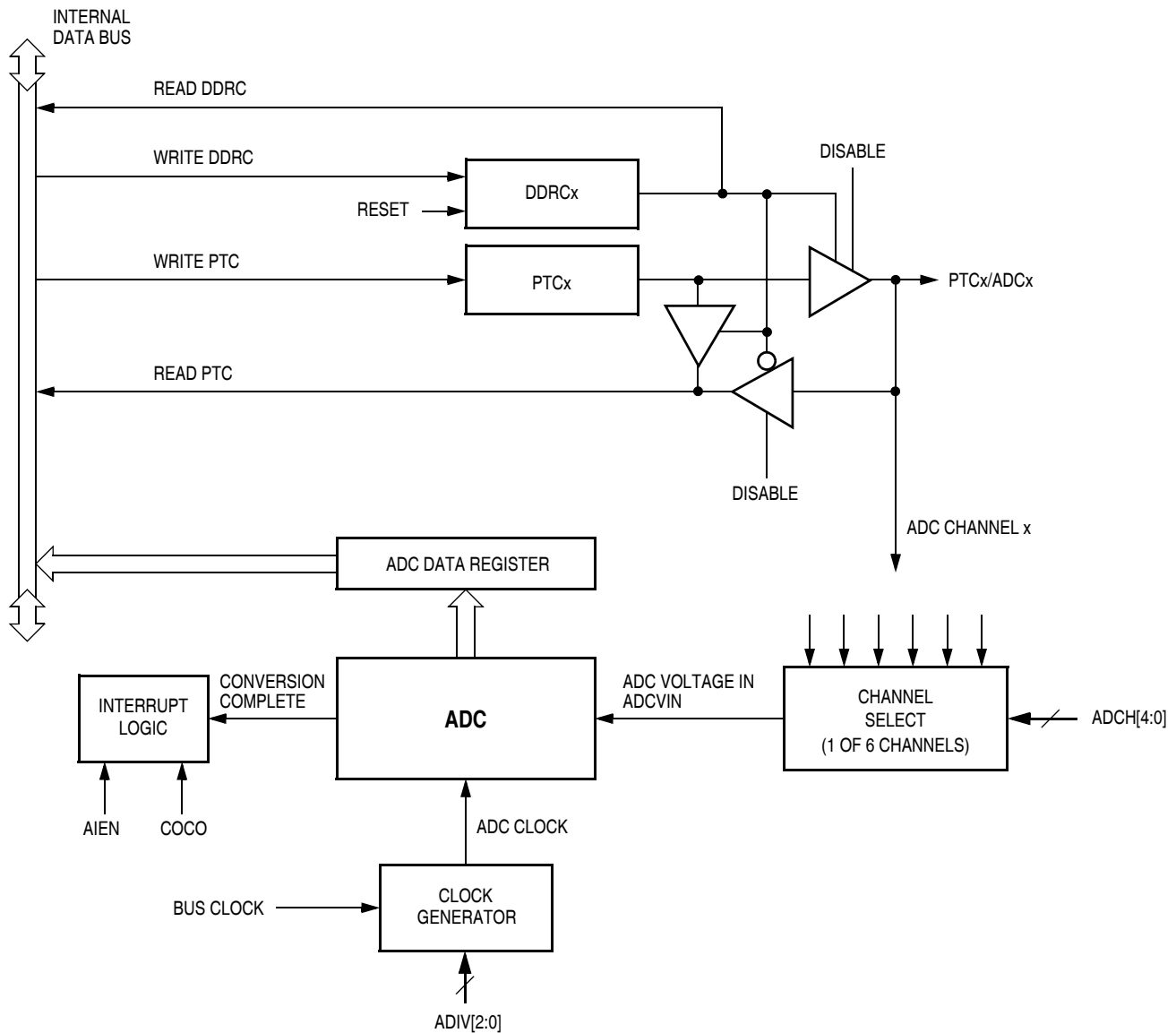


Figure 12-1. ADC Block Diagram

12.4.1 ADC Port I/O Pins

PTC5–PTC0 are general-purpose I/O pins that are shared with the ADC channels. The channel select bits (ADC status control register, \$005D), 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. Writes to the port register

or DDR 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 an unknown state 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.

12.4.2 Voltage Conversion

When the input voltage to the ADC equals $\frac{2}{3} V_{DD}$, the ADC converts the signal to \$FF (full scale). If the input voltage equals V_{SS} , the ADC converts it to \$00. Input voltage between $\frac{2}{3} V_{DD}$ and V_{SS} are a straight-line linear conversion. All other input voltages will result in \$FF if greater than $\frac{2}{3} V_{DD}$ and \$00 if less than V_{SS} .

NOTE: *Input voltage should not exceed the analog supply voltages.*

12.4.3 Conversion Time

Twelve ADC internal clocks are required to perform one conversion. The ADC starts a conversion on the first rising edge of the ADC internal clock immediately following a write to the ADSCR. If the ADC internal clock is selected to run at 1 MHz, then one conversion will take 12 μ s to complete. With a 1 MHz ADC internal clock the maximum sample rate is 83.3kHz.

$$\text{Conversion Time} = \frac{12 \text{ ADC Clock Cycles}}{\text{ADC Clock Frequency}}$$

$$\text{Number of Bus Cycles} = \text{Conversion Time} \times \text{Bus Frequency}$$

12.4.4 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 (ADC status control register, \$005D) is set after each conversion and can be cleared by writing the ADC status and control register or reading of the ADC data register.

12.4.5 Accuracy and Precision

The conversion process is monotonic and has no missing codes.

12.5 Interrupts

When the AIEN bit is set, the ADC module is capable of generating a CPU interrupt after each ADC conversion. 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.

12.6 Low-Power Modes

The following subsections describe the low-power modes.

12.6.1 Wait Mode

The ADC continues normal operation during 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 in the ADC status and control register to logic 1's before executing the WAIT instruction.

12.6.2 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.

12.7 I/O Signals

The ADC module has 6 channels that are shared with I/O port C.

12.7.1 ADC Voltage In (ADCVIN)

ADCVIN is the input voltage signal from one of the 6 ADC channels to the ADC module.

12.8 I/O Registers

Three I/O registers control and monitor ADC operation:

- ADC status and control register (ADSCR, \$005D)
- ADC data register (ADR, \$005E)
- ADC clock register (ADICLK, \$005F)

12.8.1 ADC Status and Control Register

The following paragraphs describe the function of the ADC status and control register.

Address: \$005D

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|------|-------|-------|-------|-------|-------|
| Read: | COCO | AIEN | ADCO | ADCH4 | ADCH3 | ADCH2 | ADCH1 | ADCH0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

= Unimplemented

Figure 12-2. 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 ADC status and control register is written or whenever the ADC data register is read. Reset clears this bit.

1 = conversion completed (AIEN = 0)

0 = conversion not completed (AIEN = 0)

When the AIEN bit is a logic 1 (CPU interrupt enabled), the COCO is a read-only bit, and will always be logic 0 when read.

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 is read or the status/control register 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 ADR 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

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. The five channel select bits are detailed in the following table. 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. (See [Table 12-2](#).)

The ADC subsystem is turned off when the channel select bits are all set to one. This feature allows for reduced power consumption for the MCU when the ADC is not used. Reset sets all of these bits to a logic 1.

NOTE: *Recovery from the disabled state requires one conversion cycle to stabilize.*

Table 12-2. MUX Channel Select

| ADCH4 | ADCH3 | ADCH2 | ADCH1 | ADCH0 | ADC Channel | Input Select |
|-------|-------|-------|-------|-------|-------------|-------------------------------|
| 0 | 0 | 0 | 0 | 0 | ADC0 | PTC0 |
| 0 | 0 | 0 | 0 | 1 | ADC1 | PTC1 |
| 0 | 0 | 0 | 1 | 0 | ADC2 | PTC2 |
| 0 | 0 | 0 | 1 | 1 | ADC3 | PTC3 |
| 0 | 0 | 1 | 0 | 0 | ADC4 | PTC4 |
| 0 | 0 | 1 | 0 | 1 | ADC5 | PTC5 |
| 0 | 0 | 1 | 1 | 0 | — | Unused (see Note 1) |
| : | : | : | : | : | | |
| 1 | 1 | 0 | 1 | 0 | | |
| 1 | 1 | 0 | 1 | 1 | — | Reserved |
| 1 | 1 | 1 | 0 | 0 | — | Unused |
| 1 | 1 | 1 | 0 | 1 | | V _{DDA} (see Note 2) |
| 1 | 1 | 1 | 1 | 0 | | V _{SSA} (see Note 2) |
| 1 | 1 | 1 | 1 | 1 | | ADC power 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.

12.8.2 ADC Data Register

One 8-bit result register is provided. This register is updated each time an ADC conversion completes.

Address: \$005E

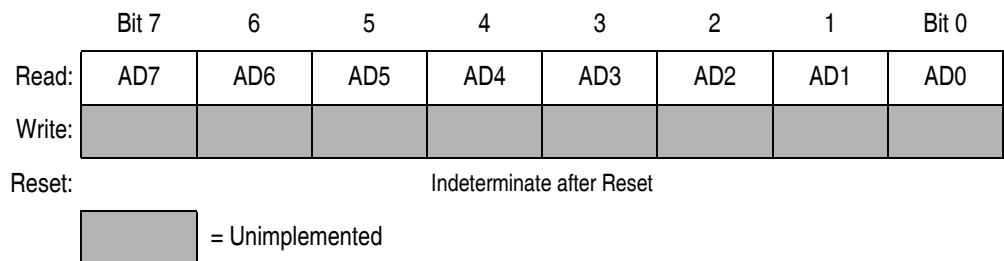


Figure 12-3. ADC Data Register (ADR)

12.8.3 ADC Input Clock Register

This register selects the clock frequency for the ADC.

Address: \$005F

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-------|-------|---|---|---|---|-------|
| Read: | ADIV2 | ADIV1 | ADIV0 | 0 | 0 | 0 | 0 | 0 |
| Write: | ADIV2 | ADIV1 | ADIV0 | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

Figure 12-4. ADC Input Clock Register (ADICLK)

ADIV2:ADIV0 — 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 12-3](#) shows the available clock configurations. The ADC clock should be set to approximately 1 MHz. With an internal bus frequency of 6MHz, set ADIV[2:0] = 010, for a divide by four ADC clock rate.

Table 12-3. ADC Clock Divide Ratio

| ADIV2 | ADIV1 | ADIV0 | ADC Clock Rate |
|-------|-------|-------|-------------------------|
| 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 | X | X | Internal bus clock ÷ 16 |

X = don't care



Section 13. Universal Serial Bus Module (USB)

13.1 Contents

| | | |
|--------|---|-----|
| 13.2 | Introduction | 163 |
| 13.3 | Features | 164 |
| 13.4 | I/O Pins | 164 |
| 13.5 | Registers | 165 |
| 13.5.1 | USB Address Register (UADR) | 166 |
| 13.5.2 | USB Interrupt Register (UINTR) | 166 |
| 13.5.3 | USB Interrupt Register 1 (UIR1) | 169 |
| 13.5.4 | USB Control Register 0 (UCR0) | 171 |
| 13.5.5 | USB Control Register 1 (UCR1) | 172 |
| 13.5.6 | USB Control Register 2 (UCR2) | 174 |
| 13.5.7 | USB Status Register (USR) | 175 |
| 13.5.8 | USB Endpoint 0 Data Registers 0 to 7 (UD0R0–UD0R7) | 177 |
| 13.5.9 | USB Endpoint 1/2 Data Registers 0 to 7 (UD1R0–UD1R7) | 177 |

13.2 Introduction

This USB module is designed to serve as a low-speed (LS) USB device per the *Universal Serial Bus Specification Rev 1.0*.

Three types of USB data transfers are supported: control, interrupt, and bulk (transmit only). Endpoint 0 functions as a receive/transmit control endpoint. Endpoints 1 and 2 can function as interrupt or bulk, but only in transmit direction.

13.3 Features

Features of the USB (universal serial bus) module include the following:

- Integrated 3.3-volt regulator with 3.3-volt output pin
- Integrated USB transceiver supporting low-speed functions
- USB data control logic
 - Packet decoding/generation
 - CRC (cyclic redundancy check) generation and checking
 - NRZI (non-return-to zero inserted) encoding/decoding
 - Bit-stuffing
- USB reset support
- Control endpoint 0 and interrupt endpoints 1 and 2
- Two 8-byte transmit buffers
- One 8-byte receive buffer
- Suspend and resume operations
 - Remote wakeup support
- USB generated interrupts
 - Transaction interrupt driven
 - Resume interrupt
 - End-of-pack (EOP) interrupt
- STALL, NAK, and ACK handshake generation

13.4 I/O Pins

The USB module uses two I/O pins, shared with standard port I/O pins. The full name of the USB I/O pins are listed in [Table 13-1](#). The generic pin name appear in the text that follows.

Table 13-1. Pin Name Conventions

| USB Generic Pin Names: | Full MCU Pin Names: | Pin Selected for USB Function By: |
|---------------------------|---------------------|--------------------------------------|
| D+ | PTD0/D+ | USBD+E bit in PDCR (\$0049) |
| D- | PTD1/D- | USBD-E bit in PDCR (\$0049) |

Table 13-2. USB I/O Register Summary

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|------------------|--|--------|---------------------------|---------|---------|---------|---------|---------|---------|---------|
| \$0029 | USB Address Register (UADR) | Read: | USBEN | UADD6 | UADD5 | UADD4 | UADD3 | UADD2 | UADD1 | UADD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002A | USB Interrupt Register (UINTR) | Read: | TBEF | RBF | EOPIF | RSTIF | TBIE | RBIE | EOPIE | RSTIE |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002B | USB Control Register 0 (UCR0) | Read: | T0SEQ | STALL0 | TX0E | RX0E | TP0SIZ3 | TP0SIZ2 | TP0SIZ1 | TP0SIZ0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002C | USB Status Register (USR) | Read: | RSEQ | SETUP | TX1ST | 0 | RPSIZ3 | RPSIZ2 | RPSIZ1 | RPSIZ0 |
| | | Write: | | | | | | | | |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$002D | USB Control Register 2 (UCR2) | Read: | 0 | 0 | PULLEN | SUSPND | ENABLE2 | ENABLE1 | STALL2 | STALL1 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002E | USB Interrupt Register 1 (UIR1) | Read: | TXD1F | TXD1IE | RESUMF | 0 | 0 | 0 | 0 | 0 |
| | | Write: | | | RESUMFR | TBEFR | RBFFR | TXD1FR | EOPFR | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002F | USB Control Register 1 (UCR1) | Read: | T1SEQ | ENDADD | TX1E | FRESUM | TP1SIZ3 | TP1SIZ2 | TP1SIZ1 | TP1SIZ0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0030 to \$0037 | USB Endpoint 0 Data Register 0–7 (UD0R0–UD0R7) | Read: | UE0RDx7 | UE0RDx6 | UE0RDx5 | UE0RDx4 | UE0RDx3 | UE0RDx2 | UE0RDx1 | UE0RDx0 |
| | | Write: | UE0TDx7 | UE0TDx6 | UE0TDx5 | UE0TDx4 | UE0TDx3 | UE0TDx2 | UE0TDx1 | UE0TDx0 |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$0038 to \$003F | USB Endpoint 1/2 Data Register 0–7 (UD1R0–UD1R7) | Read: | | | | | | | | |
| | | Write: | UE1TDx7 | UE1TDx6 | UE1TDx5 | UE1TDx4 | UE1TDx3 | UE1TDx2 | UE1TDx1 | UE1TDx0 |
| | | Reset: | Indeterminate after reset | | | | | | | |

13.5 Registers

There are seven control/status registers and 24 data buffers in the USB module. These registers are discussed in the following paragraphs.

13.5.1 USB Address Register (UADR)

Address: \$0029

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Read: | USBEN | UADD6 | UADD5 | UADD4 | UADD3 | UADD2 | UADD1 | UADD0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 13-1. USB Address Register (UADR)

USBEN — USB Enable

This read/write bit enables/disables the USB module. When USBEN bit is clear, USB will not respond to any tokens, but still can detect USB reset signal, or EOP signals, and non-IDLE state if in Suspend Mode. Reset clears this bit. After USB reset, software will set this bit to enable USB.

- 1 = USB module enabled
- 0 = USB module disabled

UADD[6:0] — USB Address

These bits specify the USB address of the device. Reset clears these bits to \$00, the default address.

13.5.2 USB Interrupt Register (UINTR)

Address: \$002A

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-----|-------|-------|------|------|-------|-------|
| Read: | TBEF | RBF | EOPIF | RSTIF | TBIE | RBIE | EORIE | RSTIE |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

Figure 13-2. USB Interrupt Register (UINTR)

TBEF — Transmit Buffer Empty Flag

This flag is set after the data stored in endpoint 0 transmit buffer has been sent and ACK handshake packet is received. Software must clear this flag by writing a logic "1" to TBEFR bit after the data is ready in the transmit buffer. This enables the next data packet transmission when endpoint 0 transmit is enabled (TX0E = 1). TBEF generates an interrupt request to the CPU if the TBIE bit is also set. Reset clears this bit.

- 1 = Transmit on endpoint 0 has occurred
- 0 = Transmit on endpoint 0 has not occurred

RBFF — Receive Buffer Full Flag

This flag is set when the module has received one data packet and replied with ACK handshake packet. Software must clear this flag by writing "1" to RBFFR bit after all the received data have been read to enable the next data packet reception. RBFF generates an interrupt request to the CPU if the RBIE bit is also set. Reset clears this bit.

- 1 = Receive on endpoint 0 has occurred
- 0 = Receive on endpoint 0 has not occurred

EOPIF — End Of Packet Interrupt Flag

This flag is set when a valid EOP signal transition is detected on the D+ and D– lines. This flag can be cleared by writing "1" to EOPIFR bit. EOPIF generates an interrupt request to the CPU if the EOPIE bit is also set. Reset clears this bit.

- 1 = End-of-packet sequence has been detected
- 0 = End-of-packet sequence has not been detected

RSTIF — Reset Interrupt Flag

The flag is set when a valid reset signal state is detected on the D+ and D– lines. This flag can be cleared by writing "1" to RSTIFR bit. RSTIF generates an interrupt request to the CPU if the RSTIE bit is also set. Reset clears this bit.

- 1 = USB reset condition has been detected
- 0 = USB reset condition has not been detected

TBIE — Transmit Buffer Interrupt Enable

When this bit is set, the TBEF flag is enabled to generate an interrupt request to the CPU. When TBIE is cleared, the TBEF flag is prevented from generating an interrupt request to the CPU. Reset clears this bit.

1 = TBEF bit set will generate interrupt request to CPU

0 = TBEF bit set does not generate interrupt request to CPU

RBIE — Receive Buffer Interrupt Enable

When this bit is set, the RBFF flag is enabled to generate an interrupt request to the CPU. When RBIE is cleared, the RBFF flag is prevented from generating an interrupt request to the CPU. Reset clears this bit.

1 = RBFF bit set will generate interrupt request to CPU

0 = RBFF bit set does not generate interrupt request to CPU

EOPIE — End Of Packet Interrupt Enable

When this bit is set, the EOPIF flag is enabled to generate an interrupt request to the CPU. When EOPIE is cleared, the EOPIF flag is prevented from generating an interrupt request to the CPU. Reset clears this bit.

1 = EOPIF bit set will generate interrupt request to CPU

0 = EOPIF bit set does not generate interrupt request to CPU

RSTIE — Reset Interrupt Enable

When this bit is set, the RSTIF flag is enabled to generate an interrupt request to the CPU. When RSTIE is cleared, the RSTIF flag is prevented from generating an interrupt request to the CPU. Reset clears this bit.

1 = RSTIF bit set will generate interrupt request to CPU

0 = RSTIF bit set does not generate interrupt request to CPU

NOTE: *Since there are more than one interrupt flags in the register, it is possible that program use Read-Modify-Write instruction to clear one flag, will occasionally clear the other flags which was just set after Read cycle of Read-Modify-Write operation.*

13.5.3 USB Interrupt Register 1 (UIR1)

Address: \$002E

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|--------|--------|---------|-------|-------|--------|-------|
| Read: | TXD1F | TXD1IE | RESUMF | 0 | 0 | 0 | 0 | 0 |
| Write: | 0 | | 0 | RESUMFR | TBEFR | RBFFR | TXD1FR | EOPFR |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 13-3. USB Interrupt Register 1 (UIR1)

TXD1F — Endpoint 1/2 Data Buffer Transmit Flag

The flag is set after the data stored in endpoint 1/2 transmit buffer has been sent and ACK handshake packet from host is received.

Software must clear this flag by writing a logic 1 to TXD1FR bit after the data is ready in the transmit buffer to enable the next data packet transmission when TX1E is enabled. Reset clears this bit.

1 = Transmit on endpoint 1/2 has occurred

0 = Transmit on endpoint 1/2 has not occurred

TXD1IE — Transmit Buffer Interrupt Enable

When this bit is set, the TXD1F flag is enabled to generate an interrupt request to the CPU. When TXD1IE is cleared, the TXD1F flag is prevented from generating an interrupt request to the CPU. Reset clears this bit.

1 = TXD1F bit set will generate interrupt request to CPU

0 = TXD1F bit set does not generate interrupt request to CPU

RESUMF — Resume Flag

This flag is set if the transaction from idle state to non-idle state is detected while in suspend mode (SUSPND = 1). An interrupt will be generated to wake up CPU to indicate a resume signalling from host and software will clear SUSPND bit and exit from suspend mode.

USB reset signals cannot be detected while in suspend mode until SUSPND bit is cleared. The RESUMF interrupt service routine is generated by SE0 to wake up the USB module. This bit can be cleared by writing "1" to RESUMFR bit. Reset clears this bit.

1 = USB bus activity detected while in suspend mode

0 = If in suspend mode, no USB bus activity has been detected

RESUMFR — Resume Flag Clear

Writing a logic "1" to this bit clears the RESUMF flag. Writing a "0" has no effect. Reset clears this bit.

1 = Writing 1 clears RESUMFR

0 = No effect

TBEFR — Transmit Buffer Empty Flag Clear

Writing a logic "1" to this bit clears the TBEF flag. Writing a "0" has no effect. Reset clears this bit.

1 = Writing 1 clears TBEF

0 = No effect

RBFFR — Receive Buffer Full Flag Clear

Writing a logic "1" to this bit clears the RBFF flag. Writing a "0" has no effect. Reset clears this bit.

1 = Writing 1 clears RBFF

0 = No effect

TXD1FR — Endpoint 1 and 2 Data Buffer Transmit Flag Clear

Writing a logic "1" to this bit clears the TXD1F flag. Writing a "0" has no effect. Reset clears this bit.

1 = Writing 1 clears TXD1F

0 = No effect

EOPIFR — End Of Packet Interrupt Flag Clear

Writing a logic "1" to this bit clears the EOPIF flag. Writing a "0" has no effect. Reset clears this bit.

1 = Writing 1 clears EOPIF

0 = No effect

13.5.4 USB Control Register 0 (UCR0)

Address: \$002B

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|--------|------|------|---------|---------|---------|---------|
| Read: | T0SEQ | STALL0 | TX0E | RX0E | TP0SIZ3 | TP0SIZ2 | TP0SIZ1 | TP0SIZ0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 13-4. USB Control Register 0 (UCR0)

T0SEQ — Endpoint 0 Data Packet PID Select

This bit will determine the transmit data packet PID of endpoint 0. When it is "0", the DATA0 PID (b0011) will be sent. When it is "1", the DATA1 PID (b1011) will be sent. Reset clears this bit.

- 1 = DATA1 PID will be sent on endpoint 0
- 0 = DATA0 PID will be sent on endpoint 0

STALL0 — Endpoint 0 STALL Handshake

This bit is set to generate the STALL handshake packet as next IN or OUT transactions handshake packet from endpoint 0. The USB hardware clears the STALL0 bit when a SETUP packet is received. Reset clears this bit.

- 1 = Send STALL handshake on endpoint 0
- 0 = Do not send STALL handshake on endpoint 0

TX0E — Endpoint 0 Transmit Enable

This bit is set to enable data packet transmission from endpoint 0. Software should set this bit when data is ready for data packet transmission. It must be cleared when no more data needs to be transmitted. If TX0E is "0" or TXD0F is "1", a NAK handshake will be returned for the next IN token. Reset clears this bit.

- 1 = Data is ready to be sent on endpoint 0
- 0 = Data is not ready; respond with ACK

RX0E — Endpoint 0 Receive Enable

This bit is set to enable data packet reception from endpoint 0. Software should set this bit when it is ready for data packet reception. It must be cleared when data cannot be received. If RX0E "0" or RBFF is "1", a NAK handshake will be returned for the next OUT token. Reset clears this bit.

- 1 = Data is read to be received on endpoint 0
- 0 = Not ready for data; respond with NAK

TP0SIZ[3:0] — Endpoint 0 Transmit Data Size

The TP0SIZ[3:0] is used to store the number of transmit data bytes from endpoint 0. The default size of transmit data is "0" after reset.

13.5.5 USB Control Register 1 (UCR1)

Address: \$002F

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|--------|------|--------|---------|---------|---------|---------|
| Read: | T1SEQ | ENDADD | TX1E | FRESUM | TP1SIZ3 | TP1SIZ2 | TP1SIZ1 | TP1SIZ0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 13-5. USB Control Register 1 (UCR1)

T1SEQ — Endpoint 1/2 Data Packet PID Select

This bit will determine the transmit data packet PID of endpoint 1/2. When it is "0", the DATA0 PID (b0011) will be sent. When it is "1", the DATA1 PID (b1011) will be sent. Reset clears this bit.

- 1 = DATA1 PID will be sent on endpoint 1/2
- 0 = DATA0 PID will be sent on endpoint 1/2

ENDADD — Endpoint Address Select

This bit specifies the which endpoint (1 or 2) uses the eight data registers, UD1R0–UD1R7, for data transmit buffers. Clearing this bit selects endpoint 1 to use the data registers. Setting this bit selects endpoint 2 to use the data registers.

If all the conditions for a successful endpoint 2 USB response to a host's IN token are satisfied (TXD1F = 0, TX1E = 1, STALL2 = 0, ENABLE2 = 1), but endpoint 1 is selected (ENDADD = 0), the module will respond with an NAK handshake packet. Reset clears this bit.

1 = The transmit buffer is used for endpoint 2

0 = The transmit buffer is used for endpoint 1

TX1E — Endpoint 1/2 Transmit Enable

This bit is set to enable data packet transmission from endpoint 1/2. Software should set this bit when data is ready for data packet transmission. It must be cleared when no more data needs to be transmitted. If TX1E is "0" or TXD1F is "1", a NAK handshake will be returned for the next IN token. Reset clears this bit.

1 = Data is ready to be sent on endpoint 1/2

0 = Data is not ready; respond with ACK

FRESUM — Force Resume

This bit is set to force a resume state on USB bus lines until software clears this bit. Before setting the FRESUM bit, the SUSPND bit must be cleared in order to drive the USB bus lines. Setting the FRESUM bit will not cause RESUMF to be set. Reset clears this bit.

1 = Resume state forced on USB bus; only if SUSPND is cleared

0 = No effect

TP1SIZ[3:0] — Endpoint 1/2 Transmit Data Size

The TP1SIZ[3:0] is used to store the number of transmit data bytes from endpoint 1/2. The default size of transmit data is "0" after reset.

13.5.6 USB Control Register 2 (UCR2)

Address: \$002D

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|--------|--------|--------|--------|---------|---------|--------|--------|
| Read: | 0 | 0 | PULLEN | SUSPND | ENABLE2 | ENABLE1 | STALL2 | STALL1 |
| Write: | RSTIFR | TX1STR | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 13-6. USB Control Register 2 (UCR2)

RSTIFR — Reset Interrupt Flag Clear

Writing a logic "1" to this bit clears the RSTIFR flag. Writing a "0" has no effect. Reset clears this bit.

- 1 = Writing 1 clears RSTIFR
- 0 = No effect

TX1STR — TX1ST Clear

Writing a logic "1" to this bit clears the TX1STR flag. Writing a "0" has no effect. Reset clears this bit.

- 1 = Writing 1 clears TX1STR
- 0 = No effect

PULLEN — Pullup Enable on D-

This bit is set to enable the internal 1.5kΩ pullup resistor connected between the USB D- line and 3.3V. Reset clears this bit.

- 1 = Enable 1.5kΩ pullup resistor between D- and 3.3V
- 0 = Disable 1.5kΩ pullup resistor between D- and 3.3V

SUSPND — USB Suspend

If the 3ms constant idle state is detected on the USB bus, user software should set this bit to allow the USB module to enter suspend mode. In the suspend mode, the clock to USB module will be stopped, and other unnecessary analog circuitry will be powered down. When a resume is detected (RESUMF = 1), user software must clear SUSPND in the interrupt service routine.

- 1 = Enable USB suspend mode
- 0 = Disable USB suspend mode

ENABLE2 — Endpoint 2 Enable

This bit is set to enable endpoint 2 to respond to addressed IN packet tokens. Reset clears this bit.

- 1 = Endpoint 2 enabled; and responds to IN tokens
- 0 = Endpoint 2 disabled

ENABLE1 — Endpoint 1 Enable

This bit is set to enable endpoint 1 to respond to addressed IN packet tokens. Reset clears this bit.

- 1 = Endpoint 1 enabled; and responds to IN tokens
- 0 = Endpoint 1 disabled

STALL2 — Endpoint 2 STALL Handshake

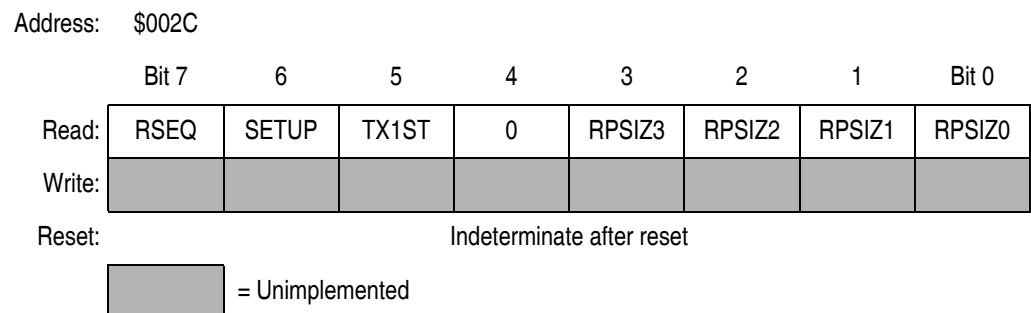
This bit is set to generate the STALL handshake packet as next IN transaction handshake packet from endpoint 2. Reset clears this bit.

- 1 = Send STALL handshake on endpoint 2
- 0 = Do not send STALL handshake on endpoint 2

STALL1 — Endpoint 1 STALL Handshake

This bit is set to generate the STALL handshake packet as next IN transaction handshake packet from endpoint 1. Reset clears this bit.

- 1 = Send STALL handshake on endpoint 1
- 0 = Do not send STALL handshake on endpoint 1

13.5.7 USB Status Register (USR)

Figure 13-7. USB Status Register (USR)

RSEQ — Received Data Sequence

This bit indicates the type of data packet of the last received data packet on endpoint 0. RSEQ = 0 indicates the last received data packet is type DATA0. RSEQ = 1 indicates the last received data packet is type DATA1.

- 1 = Last token received on endpoint 0 is a DATA1 token
- 0 = Last token received on endpoint 0 is a DATA0 token

SETUP — SETUP Token

This bit is set when the received token packet for endpoint 0 is a SETUP token (PID = b1101).

- 1 = Last token received on endpoint 0 is a SETUP token
- 0 = Last token received on endpoint 0 is not a SETUP token

TX1ST — Transmit First Flag

This bit is set if the endpoint 0 transmit buffer empty flag (TBEF) is set when the control logic is setting the endpoint 0 receive buffer full flag (RBFF). This happens when TBEF is still set at the end of an endpoint 0 reception.

- 1 = IN transaction occurred before SETUP or OUT
- 0 = IN transaction occurred after SETUP or OUT

RPSIZ[3:0] — Received Data Size

The RPSIZ[3:0] indicates the number of received data bytes in a data packet. Reset will not affect these bits

13.5.8 USB Endpoint 0 Data Registers 0 to 7 (UD0R0–UD0R7)

Address: \$0030–\$0037

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|---------------------------|---------|---------|---------|---------|---------|---------|---------|
| Read: | UE0RDx7 | UE0RDx6 | UE0RDx5 | UE0RDx4 | UE0RDx3 | UE0RDx2 | UE0RDx1 | UE0RDx0 |
| Write: | UE0TDx7 | UE0TDx6 | UE0TDx5 | UE0TDx4 | UE0TDx3 | UE0TDx2 | UE0TDx1 | UE0TDx0 |
| Reset: | Indeterminate after reset | | | | | | | |

Figure 13-8. USB Endpoint 0 Data Registers 0 to 7 (UD0R0–UD0R7)

UE0RDx[7:0] — Endpoint 0 Receive Data Buffers

These eight 8-bit buffers are serially loaded with OUT token or SETUP token data received over the USB's D+ and D– lines. These buffers are for endpoint 0.

UE0TDx[7:0] — Endpoint 0 Transmit Data Buffers

These eight 8-bit buffers are loaded by user software with data to be sent on the USB bus on the next IN token directed at endpoint 0.

13.5.9 USB Endpoint 1/2 Data Registers 0 to 7 (UD1R0–UD1R7)

Address: \$0038–\$003F

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|---------------------------|---------|---------|---------|---------|---------|---------|---------|
| Read: | | | | | | | | |
| Write: | UE1TDx7 | UE1TDx6 | UE1TDx5 | UE1TDx4 | UE1TDx3 | UE1TDx2 | UE1TDx1 | UE1TDx0 |
| Reset: | Indeterminate after reset | | | | | | | |

Figure 13-9. USB Endpoint 1 Data Registers 0 to 7 (UD1R0–UD1R7)

UE1TDx[7:0] — Endpoint 1/2 Transmit Data Buffers

These eight 8-bit buffers are loaded by user software with data to be sent on the USB bus on the next IN token directed at endpoint 1 or endpoint 2. The ENDADD bit in the USB control register 1 determines either endpoint 1 or endpoint 2 uses these buffers.



Section 14. Multi-Master IIC Interface (MMIIC)

14.1 Contents

| | | |
|--------|--|-----|
| 14.2 | Introduction | 179 |
| 14.3 | Features | 180 |
| 14.4 | I/O Pins | 180 |
| 14.5 | Registers | 181 |
| 14.5.1 | Multi-Master IIC Address Register (MMADR) | 182 |
| 14.5.2 | Multi-Master IIC Control Register (MMCR) | 183 |
| 14.5.3 | Multi-Master IIC Master Control Register (MIMCR) | 184 |
| 14.5.4 | Multi-Master IIC Status Register (MMSR) | 186 |
| 14.5.5 | Multi-Master IIC Data Transmit Register (MMDTR) | 188 |
| 14.5.6 | Multi-Master IIC Data Receive Register (MMDRR) | 189 |
| 14.6 | Programming Considerations | 190 |

14.2 Introduction

This Multi-master IIC (MMIIC) Interface is designed for internal serial communication between the MCU and other IIC devices. A hardware circuit generates "start" and "stop" signal, while byte by byte data transfer is interrupt driven by the software algorithm. Therefore, it can greatly help the software in dealing with other devices to have higher system efficiency in a typical digital monitor system.

This module not only can be applied in internal communications, but can also be used as a typical command reception serial bus for factory setup and alignment purposes. It also provides the flexibility of hooking additional devices to an existing system for future expansion without adding extra hardware.

This Multi-master IIC module uses the IIC_SCL clock line and the IIC_SDA data line to communicate with external DDC host or IIC interface. These two pins are shared with port pins PTD5 and PTD6 respectively. The outputs of IIC_SDA and IIC_SCL pins are open-drain type — no clamping diode is connected between the pin and internal V_{DD} . The maximum data rate typically is 750k-bps. The maximum communication length and the number of devices that can be connected are limited by a maximum bus capacitance of 400pF.

14.3 Features

- Compatibility with multi-master IIC bus standard
- Software controllable acknowledge bit generation
- Interrupt driven byte by byte data transfer
- Calling address identification interrupt
- Auto detection of R/W bit and switching of transmit or receive mode
- Detection of START, repeated START, and STOP signals
- Auto generation of START and STOP condition in master mode
- Arbitration loss detection and No-ACK awareness in master mode
- 8 selectable baud rate master clocks
- Automatic recognition of the received acknowledge bit

14.4 I/O Pins

The MMIIC module uses two I/O pins, shared with standard port I/O pins. The full name of the MMIIC I/O pins are listed in [Table 14-1](#). The generic pin name appear in the text that follows.

Table 14-1. Pin Name Conventions

| MMIIC Generic Pin Names: | Full MCU Pin Names: | Pin Selected for IIC Function By: |
|-----------------------------|---------------------|--------------------------------------|
| SDA | PTD6/IIC_SDA | IICDATE bit in PDCR (\$0049) |
| SCL | PTD5/IIC_SCL | IICSCLE bit in PDCR (\$0049) |

Table 14-2. MMIIC I/O Register Summary

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--------|--|--------|--------|---------|--------|-------|--------|-------|--------|---------|
| \$004A | Multi-Master IIC Master Control Register (MIMCR) | Read: | MMALIF | MMNAKIF | MMBB | MMAST | MMRW | MMBR2 | MMBR1 | MMBR0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$004B | Multi-Master IIC Address Register (MMADR) | Read: | MMAD7 | MMAD6 | MMAD5 | MMAD4 | MMAD3 | MMAD2 | MMAD1 | MMEXTAD |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| \$004C | Multi-Master IIC Control Register (MMCR) | Read: | MMEN | MMIEN | 0 | 0 | MMTXAK | 0 | 0 | 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$004D | Multi-Master IIC Status Register (MMSR) | Read: | MMRXIF | MMTXIF | MMATCH | MMSRW | MMRXAK | 0 | MMTXBE | MMRXBF |
| | | Write: | 0 | 0 | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| \$004E | Multi-Master IIC Data Transmit Register (MMDTR) | Read: | MMTD7 | MMTD6 | MMTD5 | MMTD4 | MMTD3 | MMTD2 | MMTD1 | MMTD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$004F | Multi-Master IIC Data Receive Register (MMDRR) | Read: | MMRD7 | MMRD6 | MMRD5 | MMRD4 | MMRD3 | MMRD2 | MMRD1 | MMRD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

14.5 Registers

Six registers are associated with the Multi-master IIC module, they are outlined in the following sections.

14.5.1 Multi-Master IIC Address Register (MMADR)

Address: \$004B

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|---------|
| Read: | MMAD7 | MMAD6 | MMAD5 | MMAD4 | MMAD3 | MMAD2 | MMAD1 | MMEXTAD |
| Write: | | | | | | | | |
| Reset: | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

Figure 14-1. Multi-Master IIC Address Register (MMADR)

MMAD[7:1] — Multi-Master Address

These 7 bits can be the MMIIC interface’s own specific slave address in slave mode or the calling address when in master mode. Reset sets a default value of \$A0.

MMEXTAD — Multi-Master Expanded Address

This bit is set to expand the calling address of the MMIIC in slave mode. When set, the MMIIC will acknowledge the general call address \$00 and the matched 4-bit MSB address, MMAD[7:4]. For example, when MMAD[7:1] = \$A1 and MMEXTAD = 1, the MMIIC calling address is \$A0, and it will acknowledge calling addresses \$00 and \$A0 to \$AF.

Reset clears this bit.

- 1 = MMIIC calling address is \$MMAD[7:4]0
MMIIC respond address is \$00, and \$MMAD[7:4]0 to \$MMAD[7:4]F
- 0 = MMIIC address is \$MMAD[7:1]

14.5.2 Multi-Master IIC Control Register (MMCR)

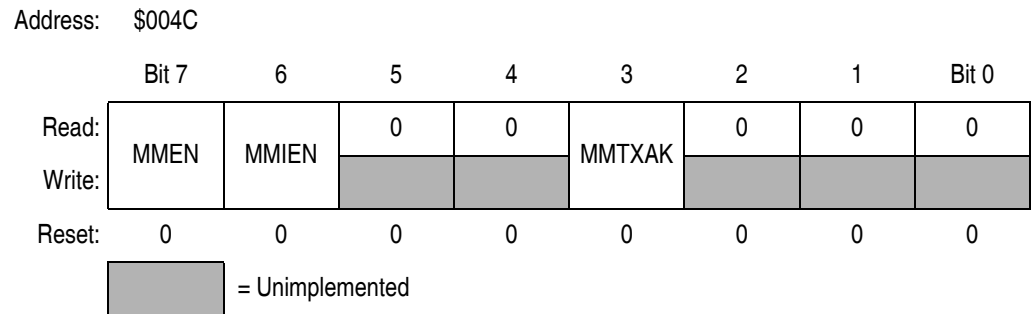


Figure 14-2. Multi-Master IIC Control Register (MMCR)

MMEN — Multi-Master IIC 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 — Multi-Master IIC 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

MMTXAK — 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

14.5.3 Multi-Master IIC Master Control Register (MIMCR)

Address: \$004A

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|--------|---------|------|-------|------|-------|-------|-------|
| Read: | MMALIF | MMNAKIF | MMBB | MMAST | MMRW | MMBR2 | MMBR1 | MMBR0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 14-3. Multi-Master IIC Master Control Register (MIMCR)

MMALIF — Multi-Master Arbitration Lost 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 MMCR is also 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

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 MMCR is also set. This bit is cleared by writing "0" to it or by reset.

- 1 = No acknowledge bit detected
- 0 = Acknowledge bit detected

MMBB — 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 — Master Control Bit

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 — Master Read/Write

This bit will be 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

MMBR2–MMBR0 — Baud Rate Select

These three bits select one of eight clock rates as the master clock when the module is in master mode.

Since this master clock is derived the CPU bus clock, the user program should not execute the WAIT instruction when the MMIIC module in master mode. This will cause the SDA and SCL lines to hang, as the WAIT instruction places the MCU in WAIT mode, with CPU clock is halted. These bits are cleared upon reset. (See [Table 14-3 . Baud Rate Select.](#))

Table 14-3. Baud Rate Select

| MMBR2 | MMBR1 | MMBR0 | Baud Rate |
|-------|-------|-------|-----------|
| 0 | 0 | 0 | 750k |
| 0 | 0 | 1 | 375k |
| 0 | 1 | 0 | 187.5k |
| 0 | 1 | 1 | 93.75k |
| 1 | 0 | 0 | 46.875k |
| 1 | 0 | 1 | 23.437k |
| 1 | 1 | 0 | 11.719k |
| 1 | 1 | 1 | 5.859k |

NOTE:
CPU bus clock is external clock ÷ 4 = 6MHz

14.5.4 Multi-Master IIC Status Register (MMSR)

Address: \$004D

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|--------|--------|--------|-------|--------|---|--------|--------|
| Read: | MMRXIF | MMTXIF | MMATCH | MMSRW | MMRXAK | 0 | MMTXBE | MMRXBF |
| Write: | 0 | 0 | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

= Unimplemented

Figure 14-4. Multi-Master IIC Status Register (MMSR)

MMRXIF — Multi-Master IIC 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 — Multi-Master 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 — Multi-Master Address Match

This flag is set when the received data in the data receive register (MMDRR) is an calling address which matches with the address or its extended addresses (MMEXTAD=1) specified in the MMADR register.

- 1 = Received address matches MMADR
- 0 = Received address does not match

MMSRW — Multi-Master Slave Read/Write

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 — Multi-Master Receive Acknowledge

When this bit is cleared, it indicates an acknowledge signal has been received after the completion of 8 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 bit
- 0 = Acknowledge signal received at 9th clock bit

MMTXBE — Multi-Master 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 — Multi-Master 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

14.5.5 Multi-Master IIC Data Transmit Register (MMDTR)

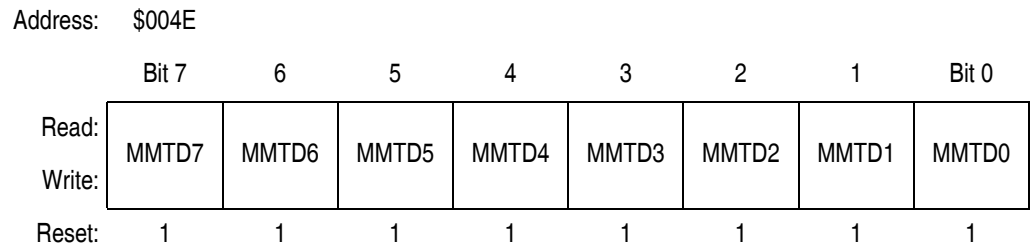


Figure 14-5. Multi-Master IIC Data Transmit Register (MMDTR)

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 been 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 been 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 14-7](#).

14.5.6 Multi-Master IIC Data Receive Register (MMDRR)

Address: \$004F

| | 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 14-6. Multi-Master IIC 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.

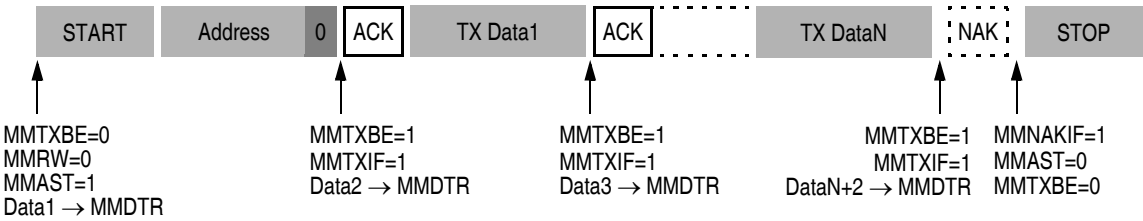
When the MMDRR is read by the CPU, the receive buffer full flag is cleared (MMRXBF = 0), and the next received data is loaded to the MMDRR. Each time when new data is loaded to the MMDRR, the MMRXIF interrupt flag is set, indicating that new data is available in MMDRR.

The sequence of events for slave receive and master receive are illustrated in [Figure 14-7](#).

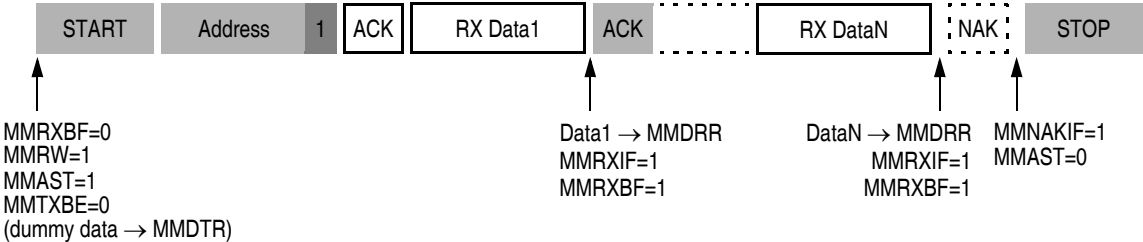
14.6 Programming Considerations

When the MMIIC module detects an arbitration loss in master mode, it will release 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 such ill condition. The software can start the time-out counter by looking at the MMBB (Bus Busy) flag in the MIMCR and reset the counter on the completion of one byte transmission. If a time-out occur, software can clear the MMEN bit (disable MMIIC module) to release the bus, and hence clearing 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.

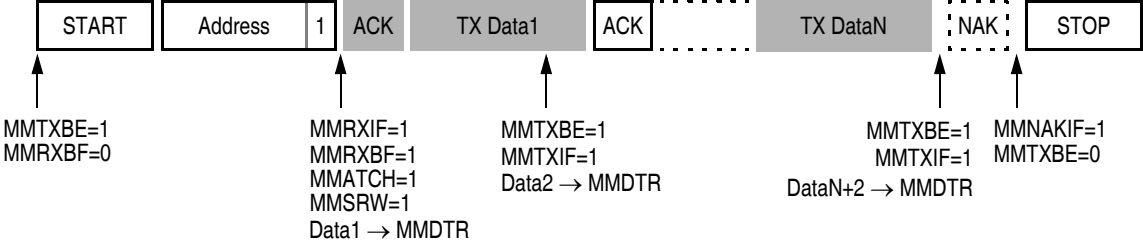
(a) Master Transmit Mode



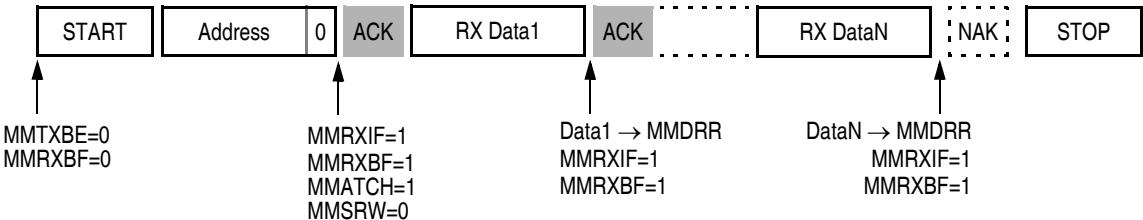
(b) Master Receive Mode



(c) Slave Transmit Mode



(d) Slave Receive Mode



KEY: shaded data packets indicate a transmit by the MCU's MMIIC module

Figure 14-7. Data Transfer Sequences for Master/Slave Transmit/Receive Modes



Section 15. DDC12AB Interface

15.1 Contents

| | | |
|--------|--|-----|
| 15.2 | Introduction | 193 |
| 15.3 | Features | 194 |
| 15.4 | I/O Pins | 194 |
| 15.5 | DDC Protocols | 196 |
| 15.6 | Registers | 196 |
| 15.6.1 | DDC Address Register (DADR) | 196 |
| 15.6.2 | DDC2 Address Register (D2ADR) | 197 |
| 15.6.3 | DDC Control Register (DCR) | 198 |
| 15.6.4 | DDC Master Control Register (DMCR) | 199 |
| 15.6.5 | DDC Status Register (DSR) | 202 |
| 15.6.6 | DDC Data Transmit Register (DDTR) | 204 |
| 15.6.7 | DDC Data Receive Register (DDRR) | 205 |
| 15.7 | Programming Considerations | 206 |

15.2 Introduction

This DDC12AB Interface module is used by the digital monitor to show its identification information to the video controller. It contains DDC1 hardware and a two-wire, bidirectional serial bus which is fully compatible with multi-master IIC bus protocol to support DDC2AB interface.

This module not only can be applied in internal communications, but can also be used as a typical command reception serial bus for factory setup and alignment purposes. It also provides the flexibility of hooking additional devices to an existing system for future expansion without adding extra hardware.

This DDC12AB module uses the DDCSCL clock line and the DDCSDA data line to communicate with external DDC host or IIC interface. These two pins are shared with port pins PTD3 and PTD2 respectively. The outputs of DDCSDA and DDCSCL pins are open-drain type — no clamping diode is connected between the pin and internal V_{DD} . The maximum data rate typically is 100k-bps. The maximum communication length and the number of devices that can be connected are limited by a maximum bus capacitance of 400pF.

15.3 Features

- DDC1 hardware
- Compatibility with multi-master IIC bus standard
- Software controllable acknowledge bit generation
- Interrupt driven byte by byte data transfer
- Calling address identification interrupt
- Auto detection of R/W bit and switching of transmit or receive mode
- Detection of START, repeated START, and STOP signals
- Auto generation of START and STOP condition in master mode
- Arbitration loss detection and No-ACK awareness in master mode
- 8 selectable baud rate master clocks
- Automatic recognition of the received acknowledge bit

15.4 I/O Pins

The DDC12AB module uses two I/O pins, shared with standard port I/O pins. The full name of the DDC12AB I/O pins are listed in [Table 15-1](#). The generic pin name appear in the text that follows.

Table 15-1. Pin Name Conventions

| DDC12AB Generic Pin Names: | Full MCU Pin Names: | Pin Selected for DDC Function By: |
|-------------------------------|---------------------|--------------------------------------|
| SDA | PTD2/DDCSDA | DDCDATE bit in PDCR (\$0049) |
| SCL | PTD3/DDCSCL | DDCSCLE bit in PDCR (\$0049) |

Table 15-2. DDC I/O Register Summary

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--------|------------------------------------|--------|-------|-------|-------|-------|-------|--------|--------|-------|
| \$0016 | DDC Master Control Register (DMCR) | Read: | | | | | | | | |
| | | Write: | ALIF | NAKIF | BB | MAST | MRW | BR2 | BR1 | BR0 |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0017 | DDC Address Register (DADR) | Read: | | | | | | | | |
| | | Write: | DAD7 | DAD6 | DAD5 | DAD4 | DAD3 | DAD2 | DAD1 | EXTAD |
| | | Reset: | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| \$0018 | DDC Control Register (DCR) | Read: | | | 0 | 0 | | | | 0 |
| | | Write: | DEN | DIEN | | | TXAK | SCLIEN | DDC1EN | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0019 | DDC Status Register (DSR) | Read: | RXIF | TXIF | MATCH | SRW | RXAK | SCLIF | TXBE | RXBF |
| | | Write: | 0 | 0 | | | | 0 | | |
| | | Reset: | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| \$001A | DDC Data Transmit Register (DDTR) | Read: | | | | | | | | |
| | | Write: | DTD7 | DTD6 | DTD5 | DTD4 | DTD3 | DTD2 | DTD1 | DTD0 |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$001B | DDC Data Receive Register (DDRR) | Read: | DRD7 | DRD6 | DRD5 | DRD4 | DRD3 | DRD2 | DRD1 | DRD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$001C | DDC2 Address Register (D2ADR) | Read: | | | | | | | | 0 |
| | | Write: | D2AD7 | D2AD6 | D2AD5 | D2AD4 | D2AD3 | D2AD2 | D2AD1 | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

15.5 DDC Protocols

In DDC1 protocol communication, the module is in transmit mode. The data written to the transmit register is continuously clocked out to the SDA line by the rising edge of the Vsync input signal. During DDC1 communication, a falling transition on the SCL line can be detected to generate an interrupt to the CPU for mode switching.

In DDC2AB protocol communication, the module can be either in transmit mode or in receive mode, controlled by the calling master.

In DDC2 protocol communication, the module will act as a standard IIC module, able to act as a master or a slave device.

15.6 Registers

Seven registers are associated with the DDC module, they outlined in the following sections.

15.6.1 DDC Address Register (DADR)

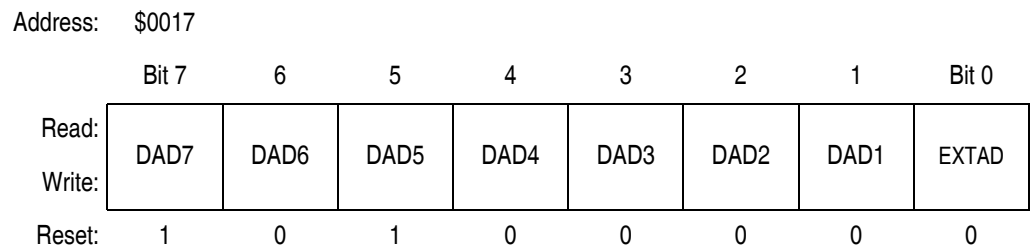


Figure 15-1. DDC Address Register (DADR)

DAD[7:1] — DDC Address

These 7 bits can be the DDC2 interface's own specific slave address in slave mode or the calling address when in master mode. Reset sets a default value of \$A0.

EXTAD — DDC Expanded Address

This bit is set to expand the calling address of the DDC in slave mode. When set, the DDC will acknowledge the general call address \$00 and the matched 4-bit MSB address, DAD[7:4].

For example, when DAD[7:1] = \$A1 and EXTAD = 1, the DDC calling address is \$A0, and it will acknowledge calling addresses \$00 and \$A0 to \$AF.

Reset clears this bit.

1 = DDC calling address is \$DAD[7:4]0

DDC respond address is \$00, and \$DAD[7:4]0 to \$DAD[7:4]F

0 = DDC address id \$DAD[7:1]

15.6.2 DDC2 Address Register (D2ADR)

Address: \$001C

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Read: | D2AD7 | D2AD6 | D2AD5 | D2AD4 | D2AD3 | D2AD2 | D2AD1 | 0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 15-2. DDC2 Address Register (D2ADR)

D2AD[7:1] — DDC2 Address

These 7 bits represent the second slave address for the DDC2BI protocol. D2AD[7:1] should be set to the same value as DAD[7:1] in DADR if user application do not use DDC2BI. Reset clears all bits this register.

15.6.3 DDC Control Register (DCR)

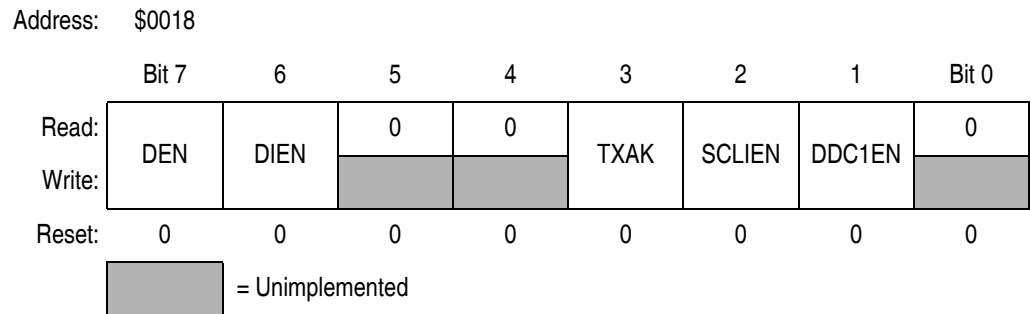


Figure 15-3. DDC Control Register (DCR)

DEN — DDC Enable

This bit is set to enable the DDC module. When DEN = 0, module is disabled and all flags will restore to its power-on default states. Reset clears this bit.

- 1 = DDC module enabled
- 0 = DDC module disabled

DIEN — DDC Interrupt Enable

When this bit is set, the TXIF, RXIF, ALIF, and NAKIF flags are enabled to generate an interrupt request to the CPU. When DIEN is cleared, the these flags are prevented from generating an interrupt request. Reset clears this bit.

- 1 = TXIF, RXIF, ALIF, and/or NAKIF bit set will generate interrupt request to CPU
- 0 = TXIF, RXIF, ALIF, and/or NAKIF bit set will not generate interrupt request to CPU

TXAK — Transmit Acknowledge Enable

This bit is set to disable the DDC from sending out an acknowledge signal to the bus at the 9th clock bit after receiving 8 data bits. When TXAK is cleared, an acknowledge signal will be sent at the 9th clock bit. Reset clears this bit.

- 1 = DDC does not send acknowledge signals at 9th clock bit
- 0 = DDC sends acknowledge signal at 9th clock bit

SCLIEN — SCL Interrupt Enable

When this bit is set, the SCLIF flag is enabled to generate an interrupt request to the CPU. When SCLIEN is cleared, SCLIF is prevented from generating an interrupt request. Reset clears this bit.

- 1 = SCLIF bit set will generate interrupt request to CPU
- 0 = SCLIF bit set will not generate interrupt request to CPU

DDC1EN — DDC1 Protocol Enable

This bit is set to enable DDC1 protocol. The DDC1 protocol will use the Vsync input (from sync processor) as the master clock input to the DDC module. Vsync rising-edge will continuously clock out the data to the output circuit. No calling address comparison is performed. The SRW bit in DDC status register (DSR) will always read as "1". Reset clears this bit.

- 1 = DDC1 protocol enabled
- 0 = DDC1 protocol disabled

15.6.4 DDC Master Control Register (DMCR)

Address: \$0016

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-------|----|------|-----|-----|-----|-------|
| Read: | ALIF | NAKIF | BB | MAST | MRW | BR2 | BR1 | BR0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 15-4. DDC Master Control Register (DMCR)

ALIF — DDC Arbitration Lost Interrupt Flag

This flag is set when software attempt to set MAST but the BB has been set by detecting the start condition on the lines or when the DDC 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 DIEN bit in DCR is also set. This bit is cleared by writing "0" to it or by reset.

- 1 = Lost arbitration in master mode
- 0 = No arbitration lost

NAKIF — No Acknowledge Interrupt Flag

This flag is only set in master mode (MAST = 1) when there is no acknowledge bit detected after one data byte or calling address is transferred. This flag also clears MAST. NAKIF generates an interrupt request to CPU if the DIEN bit in DCR is also set. This bit is cleared by writing "0" to it or by reset.

- 1 = No acknowledge bit detected
- 0 = Acknowledge bit detected

BB — 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 DDC is disabled. Reset clears this bit.

- 1 = Start condition detected
- 0 = Stop condition detected or DDC is disabled

MAST — Master Control Bit

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 DADR.

When the MAST bit is cleared by NAKIF 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 (ALIF = 1), the module reverts to slave mode by clearing MAST, 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

MRW — Master Read/Write

This bit will be transmitted out as bit 0 of the calling address when the module sets the MAST bit to enter master mode. The MRW 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

BR2–BR0 — Baud Rate Select

These three bits select one of eight clock rates as the master clock when the module is in master mode.

Since this master clock is derived the CPU bus clock, the user program should not execute the WAIT instruction when the DDC module in master mode. This will cause the SDA and SCL lines to hang, as the WAIT instruction places the MCU in WAIT mode, with CPU clock is halted. These bits are cleared upon reset. (See [Table 15-3 . Baud Rate Select.](#))

Table 15-3. Baud Rate Select

| BR2 | BR1 | BR0 | Baud Rate |
|---|-----|-----|-----------|
| 0 | 0 | 0 | 100k |
| 0 | 0 | 1 | 50k |
| 0 | 1 | 0 | 25k |
| 0 | 1 | 1 | 12.5k |
| 1 | 0 | 0 | 6.25k |
| 1 | 0 | 1 | 3.125k |
| 1 | 1 | 0 | 1.56k |
| 1 | 1 | 1 | 0.78k |
| NOTE: CPU bus clock is external clock ÷ 4 = 6MHz | | | |

15.6.5 DDC Status Register (DSR)

Address: \$0019

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|-------|-----|------|-------|------|-------|
| Read: | RXIF | TXIF | MATCH | SRW | RXAK | SCLIF | TXBE | RXBF |
| Write: | 0 | 0 | | | | 0 | | |
| Reset: | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

= Unimplemented

Figure 15-5. DDC Status Register (DSR)

RXIF — DDC Receive Interrupt Flag

This flag is set after the data receive register (DDRR) is loaded with a new received data. Once the DDRR is loaded with received data, no more received data can be loaded to the DDRR register until the CPU reads the data from the DDRR to clear RXBF flag. RXIF generates an interrupt request to CPU if the DIEN bit in DCR is also set. This bit is cleared by writing "0" to it or by reset; or when the DEN = 0.

- 1 = New data in data receive register (DDRR)
- 0 = No data received

TXIF — DDC Transmit Interrupt Flag

This flag is set when data in the data transmit register (DDTR) is downloaded to the output circuit, and that new data can be written to the DDTR. TXIF generates an interrupt request to CPU if the DIEN bit in DCR is also set. This bit is cleared by writing "0" to it or when the DEN = 0.

- 1 = Data transfer completed
- 0 = Data transfer in progress

MATCH — DDC Address Match

This flag is set when the received data in the data receive register (DDRR) is an calling address which matches with the address or its extended addresses (EXTAD=1) specified in the DADR register.

- 1 = Received address matches DADR
- 0 = Received address does not match

SRW — DDC Slave Read/Write

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. SRW = 1 when the calling master is reading data from the module (slave transmit mode). SRW = 0 when the master is writing data to the module (receive mode).

- 1 = Slave mode transmit
- 0 = Slave mode receive

RXAK — DDC Receive Acknowledge

When this bit is cleared, it indicates an acknowledge signal has been received after the completion of 8 data bits transmission on the bus. When RXAK 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 bit
- 0 = Acknowledge signal received at 9th clock bit

SCLIF — SCL Interrupt Flag

This flag is set when a falling edge is detected on the SCL line, only if DDC1EN bit is set. SCLIF generates an interrupt request to CPU if the SCLIFEN bit in DCR is also set. SCLIF is cleared by writing "0" to it or when the DDC1EN = 0, or DEN = 0. Reset clears this bit.

- 1 = Falling edge detected on SCL line
- 0 = No falling edge detected on SCL line

TXBE — DDC Transmit Buffer Empty

This flag indicates the status of the data transmit register (DDTR). When the CPU writes the data to the DDTR, the TXBE flag will be cleared. TXBE is set when DDTR 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

RXBF — DDC Receive Buffer Full

This flag indicates the status of the data receive register (DDRR). When the CPU reads the data from the DDRR, the RXBF flag will be cleared. RXBF is set when DDRR is full by a transfer of data from the input circuit to the DDRR. Reset clears this bit.

1 = Data receive register full

0 = Data receive register empty

15.6.6 DDC Data Transmit Register (DDTR)

Address: \$001A

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|------|------|------|------|------|-------|
| Read: | DTD7 | DTD6 | DTD5 | DTD4 | DTD3 | DTD2 | DTD1 | DTD0 |
| Write: | | | | | | | | |
| Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Figure 15-6. DDC Data Transmit Register (DDTR)

When the DDC module is enabled, DEN = 1, data written into this register depends on whether module is in master or slave mode.

In slave mode, the data in DDTR will be transferred to the output circuit when:

- the module detects a matched calling address (MATCH = 1), with the calling master requesting data (SRW = 1); or
- the previous data in the output circuit has been transmitted and the receiving master returns an acknowledge bit, indicated by a received acknowledge bit (RXAK = 0).

If the calling master does not return an acknowledge bit (RXAK = 1), the module will release the SDA line for master to generate a "stop" or "repeated start" condition. The data in the DDTR will not be transferred to the output circuit until the next calling from a master. The transmit buffer empty flag remains cleared (TXBE = 0).

In master mode, the data in DDTR will be transferred to the output circuit when:

- the module receives an acknowledge bit (RXAK = 0), after setting master transmit mode (MRW = 0), and the calling address has been transmitted; or
- the previous data in the output circuit has been transmitted and the receiving slave returns an acknowledge bit, indicated by a received acknowledge bit (RXAK = 0).

If the slave does not return an acknowledge bit (RXAK = 1), the master will generate a "stop" or "repeated start" condition. The data in the DDTR will not be transferred to the output circuit. The transmit buffer empty flag remains cleared (TXBE = 0).

The sequence of events for slave transmit and master transmit are illustrated in [Figure 15-8](#).

15.6.7 DDC Data Receive Register (DDRR)

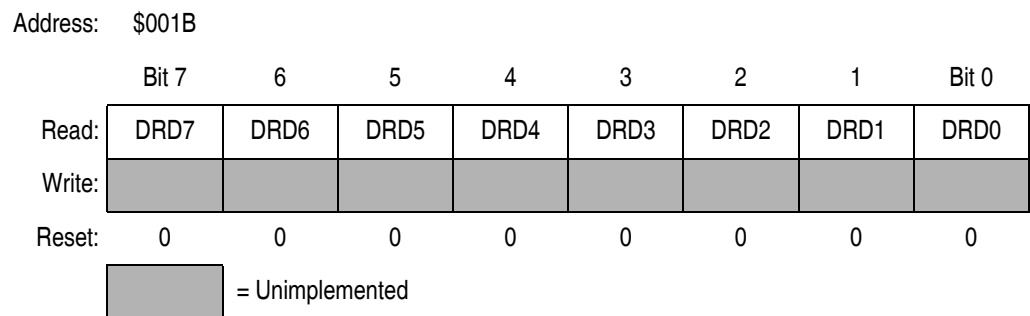


Figure 15-7. DDC Data Receive Register (DDRR)

When the DDC module is enabled, DEN = 1, data in this read-only register depends on whether module is in master or slave mode.

In slave mode, the data in DDRR is:

- the calling address from the master when the address match flag is set (MATCH = 1); or
- the last data received when MATCH = 0.

In master mode, the data in the DDRR is:

- the last data received.

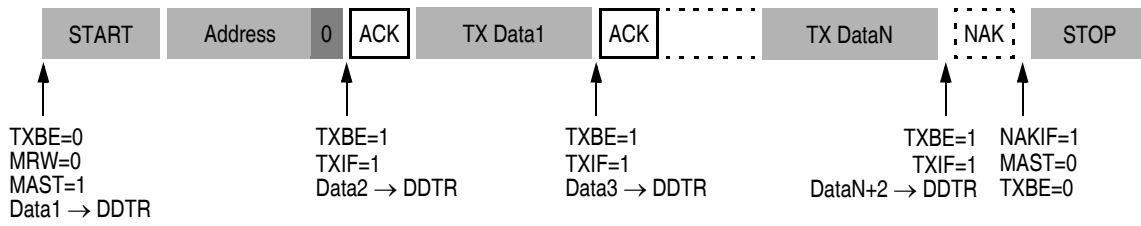
When the DRR is read by the CPU, the receive buffer full flag is cleared (RXBF = 0), and the next received data is loaded to the DRR. Each time when new data is loaded to the DRR, the RXIF interrupt flag is set, indicating that new data is available in DRR.

The sequence of events for slave receive and master receive are illustrated in [Figure 15-8](#).

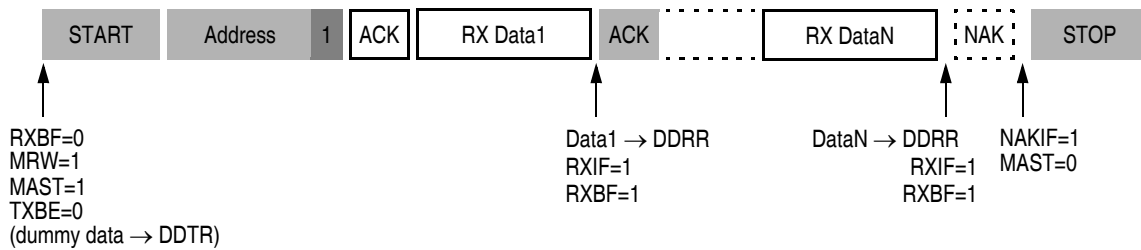
15.7 Programming Considerations

When the DDC module detects an arbitration loss in master mode, it will release 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 such ill condition. The software can start the time-out counter by looking at the BB (Bus Busy) flag in the DMCR and reset the counter on the completion of one byte transmission. If a time-out occur, software can clear the DEN bit (disable DDC module) to release the bus, and hence clearing the BB flag. This is the only way to clear the BB flag by software if the module hangs up due to a no STOP condition received. The DDC can resume operation again by setting the DEN bit.

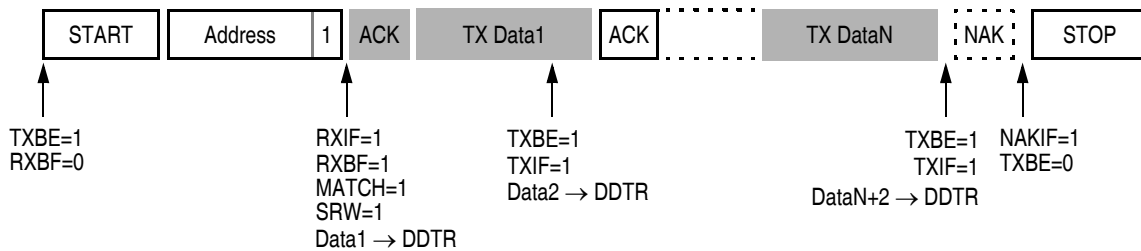
(a) Master Transmit Mode



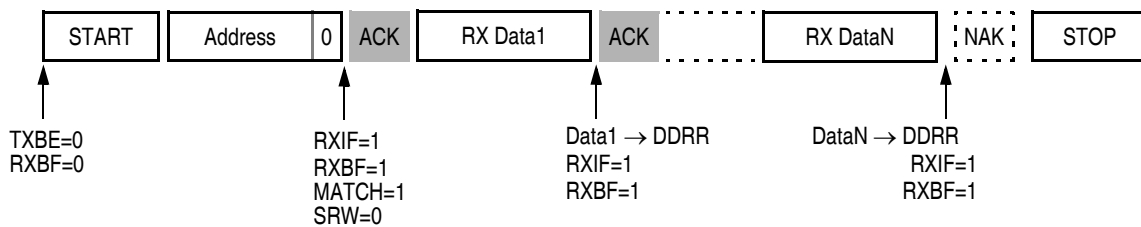
(b) Master Receive Mode



(c) Slave Transmit Mode



(d) Slave Receive Mode



KEY: shaded data packets indicate a transmit by the MCU's DDC module

Figure 15-8. Data Transfer Sequences for Master/Slave Transmit/Receive Modes



Section 16. Sync Processor

16.1 Contents

| | | |
|----------|---|-----|
| 16.2 | Introduction | 209 |
| 16.3 | Features | 211 |
| 16.4 | I/O Pins | 211 |
| 16.5 | Functional Blocks | 213 |
| 16.5.1 | Polarity Detection | 214 |
| 16.5.1.1 | Hsync Polarity Detection | 214 |
| 16.5.1.2 | Vsync Polarity Detection | 214 |
| 16.5.1.3 | Composite Sync Polarity Detection | 214 |
| 16.5.2 | Sync Signal Counters | 215 |
| 16.5.3 | Polarity Controlled HSYNCO and VSYNCO Outputs | 215 |
| 16.5.4 | Clamp Pulse Output | 216 |
| 16.5.5 | Low Vertical Frequency Detect | 217 |
| 16.6 | Registers | 217 |
| 16.6.1 | Sync Processor Control & Status Register (SPCSR) | 217 |
| 16.6.2 | Sync Processor Input/Output Control Register (SPIOCR) | 219 |
| 16.6.3 | Vertical Frequency Registers (VFRs) | 221 |
| 16.6.4 | Hsync Frequency Registers (HFRs) | 223 |
| 16.6.5 | Sync Processor Control Register 1 (SPCR1) | 225 |
| 16.6.6 | H&V Sync Output Control Register (HVOCR) | 226 |
| 16.7 | System Operation | 227 |

16.2 Introduction

The Sync Processor is designed to detect and process sync signals inside a digital monitor system — from separated Hsync and Vsync inputs, or from composite sync inputs such as Sync-On-Green (SOG).

After detection and the necessary polarity correction and/or sync separation, the corrected sync signals are sent out. The MCU can also send commands to other monitor circuitry, such as for the geometry correction and OSD, using the DDC12AB and/or the IIC communication channels.

The block diagram of the Sync Processor is shown in [Figure 16-1](#).

NOTE: *All quoted timings in this section assume an internal bus frequency of 6MHz.*

16.3 Features

Features of the Sync Processor include the following:

- Polarity detector
- Horizontal frequency counter
- Vertical frequency counter
- Low vertical frequency indicator (40.7Hz)
- Polarity controlled HSYNCO and VSYNCO outputs:
 - From separate Hsync and Vsync
 - From composite sync on HSYNC or SOG input pin
 - From internal selectable free running Hsync and Vsync pulses
- CLAMP pulse output to the external pre-amp chip
- Internal schmitt trigger on HSYNC, VSYNC, and SOG input pins to improve noise immunity

16.4 I/O Pins

The Sync Processor uses six I/O pins, with four pins shared with standard port I/O pins. The full name of the Sync Processor I/O pins are listed in [Table 16-1](#). The generic pin name appear in the text that follows.

Table 16-1. Pin Name Conventions

| Sync Processor Generic Pin Names: | Full MCU Pin Names: | Pin Selected for Sync Processor Function By: |
|-----------------------------------|---------------------|--|
| HSYNC | HSYNC | — |
| VSYNC | VSYNC | — |
| SOG | PTE0/SOG/TCH0 | SOGE bit in CONFIG1 (\$001D) |
| HSYNCO | PTE1/HSYNCO | HSYNCOE bit in CONFIG 1 (\$001D) |
| VSYNCO | PTE2/VSYNCO | VSYNCOE bit in CONFIG 1 (\$001D) |
| CLAMP | PTD4/CLAMP | CLAMPE bit in PDCR (\$0049) |

Table 16-2. Sync Processor I/O Register Summary

| Addr. | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|--|--------|--------|--------|-------|------|--------|---------|--------|--------|
| \$0040 | Sync Processor Control and Status Register (SPCSR) | Read: | VSIE | VEDGE | VSIF | COMP | VINVO | HINVO | VPOL | HPOL |
| | | Write: | | | 0 | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0041 | Vertical Frequency High Register (VFHR) | Read: | VOF | 0 | 0 | VF12 | VF11 | VF10 | VF9 | VF8 |
| | | Write: | | CPW1 | CPW0 | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0042 | Vertical Frequency Low Register (VFLR) | Read: | VF7 | VF6 | VF5 | VF4 | VF3 | VF2 | VF1 | VF0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0043 | Hsync Frequency High Register (HFHR) | Read: | HFH7 | HFH6 | HFH5 | HFH4 | HFH3 | HFH2 | HFH1 | HFH0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0044 | Hsync Frequency Low Register (HFLR) | Read: | HOVER | 0 | 0 | HFL4 | HFL3 | HFL2 | HFL1 | HFL0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0045 | Sync Processor I/O Control Register (SPIOCR) | Read: | VSYNCS | HSYNCS | COINV | R | SOGSEL | CLAMPOE | BPOR | SOUT |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0046 | Sync Processor Control Register 1 (SPCR1) | Read: | LVSIE | LVSIF | HPS1 | HPS0 | R | R | ATPOL | FSHF |
| | | Write: | | 0 | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0047 | H&V Sync Output Control Register (HVOCR) | Read: | R | 0 | 0 | 0 | 0 | HVOCR2 | HVOCR1 | HVOCR0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented
 R = Reserved

16.5 Functional Blocks

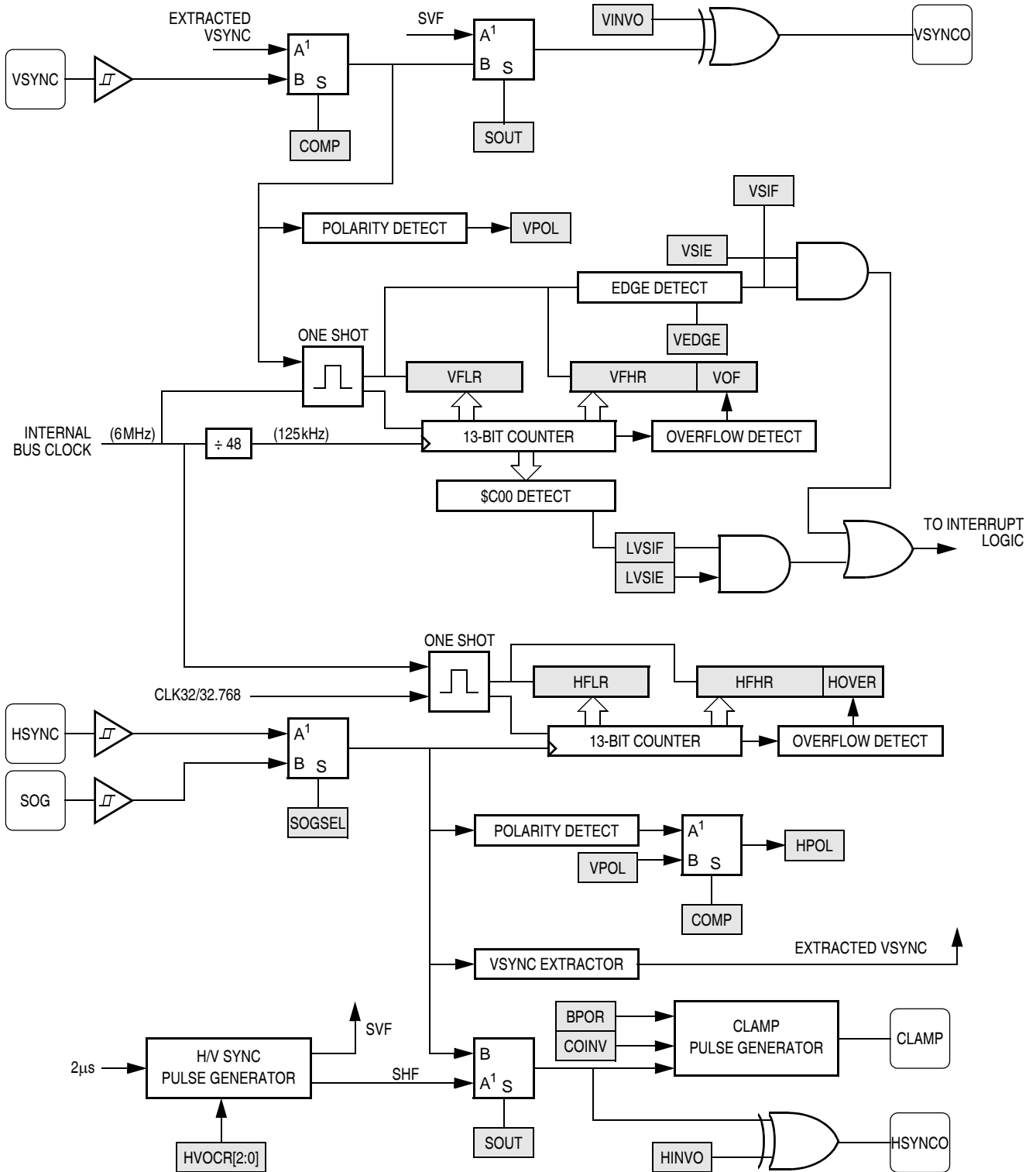


Figure 16-1. Sync Processor Block Diagram

16.5.1 Polarity Detection

16.5.1.1 Hsync Polarity Detection

The Hsync polarity detection circuit measures the length of high and low period of the HSYNC input. If the length of high is longer than L and the length of low is shorter than S , the HPOL bit will be "0", indicating a negative polarity HSYNC input. If the length of low is longer than L and the length of high is shorter than S , the HPOL bit will be "1", indicating a positive polarity HSYNC input. The table below shows three possible cases for HSYNC polarity detection — the conditions are selected by the HPS[1:0] bits in the Sync Processor Control Register 1 (SPCR1).

| Polarity Detection Pulse Width | | SPCR1 (\$0046) | |
|--------------------------------|----------------------------|----------------|------|
| Long is greater than (L) | Short is less than (S) | HPS1 | HPS0 |
| 7 μ s | 6 μ s | 0 | 0 |
| 3.5 μ s | 3 μ s | 1 | X |
| 14 μ s | 12 μ s | 0 | 1 |

16.5.1.2 Vsync Polarity Detection

The Vsync polarity detection circuit performs a similar function as for Hsync. If the length of high is longer than 4ms and the length of low is shorter than 2ms, the VPOL bit will be "0", indicating a negative polarity VSYNC input. If the length of low is longer than 4ms and the length of high is shorter than 2ms, the VPOL bit will be "1", indicating a positive polarity VSYNC input.

16.5.1.3 Composite Sync Polarity Detection

When a composite sync signal is the input (COMP = 1 for composite sync processing), the HPOL bit = VPOL bit, and the polarity is detected using the VSYNC polarity detection criteria described in section [16.5.1.2](#).

16.5.2 Sync Signal Counters

There are two counters: a 13-bit horizontal frequency counter to count the number of horizontal sync pulses within a 32ms or 8ms period; and a 13-bit vertical frequency counter to count the number of system clock cycles between two vertical sync pulses. These two data can be read by the CPU to check the signal frequencies and to determine the video mode.

The 13-bit vertical frequency register encompasses vertical frequency range from approximately 15Hz to 128kHz. Due to the asynchronous timing between the incoming VSYNC signal and internal system clock, there will be ± 1 count error on reading the **Vertical Frequency Registers (VFRs)** for the same vertical frequency.

The horizontal counter counts the pulses on HSYNC pin input, and is uploaded to the **Hsync Frequency Registers (HFRs)** every 32.768ms or 8.192ms.

16.5.3 Polarity Controlled HSYNCO and VSYNCO Outputs

The processed sync signals are output on HSYNCO and VSYNCO when the corresponding bits in Configuration Register 0 (\$001D) are set. The signal to these output pins depend on SOUT and COMP bits (see **Table 16-3**), with polarity controlled by ATPOL, HINVO, and VINVO bits as shown in **Table 16-4**.

Table 16-3. Sync Output Control

| SOUT | COMP | Sync Outputs: VSYNCO and HSYNCO |
|------|------|---|
| 1 | X | Free-running pulse with negative polarity |
| 0 | 0 | Sync outputs follow sync inputs VSYNC and HSYNC respectively, with polarity correction shown in Table 16-4 . |
| 0 | 1 | HSYNCO follows the composite sync input and VSYNCO is the extracted Vsync (3 to 14 μ s delay to composite input), with polarity correction shown in Table 16-4 . |

Table 16-4. Sync Output Polarity

| ATPOL | SOUT | VINVO or HINVO | Sync Outputs: VSYNCO/HSYNCO |
|-------|------|----------------|---|
| X | 1 | X | Free-running pulse with negative polarity |
| 0 | 0 | 0 | Same polarity as sync input |
| 0 | 0 | 1 | Inverted polarity of sync input |
| 1 | 0 | 0 | Negative polarity sync output |
| 1 | 0 | 1 | Positive polarity sync output |

When the SOUT bit is set, the HSYNCO output is a free-running pulse with 2 μ s width. Both HSYNCO and VSYNCO outputs are negative polarity, with frequencies selected by the H & V Sync Output Control Register (HVOCR).

16.5.4 Clamp Pulse Output

When the CLAMPOE bit in SPIOCR is set to "1", a clamp signal is output on the CLAMP pin. This clamp pulse is triggered either on the leading edge or the trailing edge of HSYNC, controlled by BPOR bit, with the polarity controlled by the COINV bit. See [Figure 16-2 . Clamp Pulse Output Timing](#).

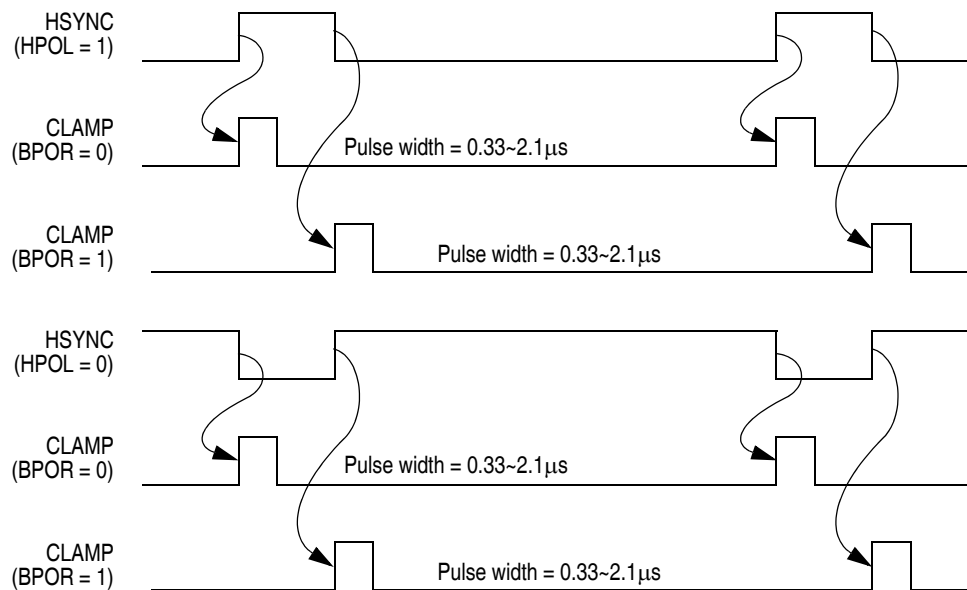


Figure 16-2. Clamp Pulse Output Timing

16.5.5 Low Vertical Frequency Detect

Logic monitors the value of the Vsync Frequency Register (VFR), and sets the low vertical frequency flag (LVSIF) when the value of VFR is higher than \$C00 (frequency below 40.7Hz). LVSIF bit can generate an interrupt request to the CPU when the LVSIE bit is set and I-bit in the Condition Code Register is "0". The LVSIF bit can help the system to detect video off mode fast.

16.6 Registers

Eight registers are associated with the Sync Processor, they outlined in the following sections.

16.6.1 Sync Processor Control & Status Register (SPCSR)

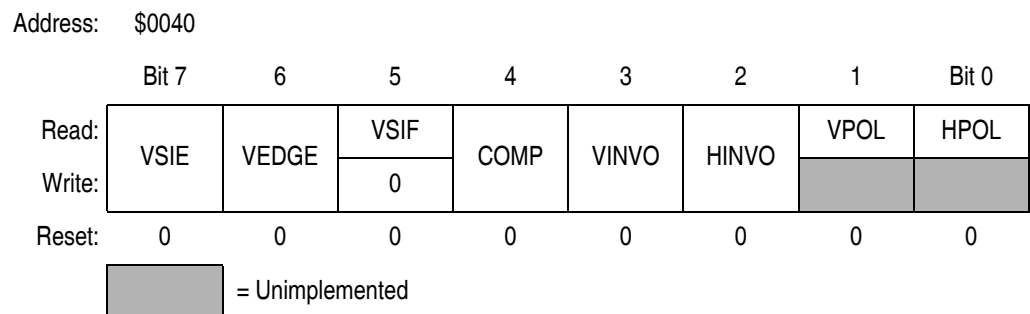


Figure 16-3. Sync Processor Control & Status Register (SPCSR)

VSIE — VSync Interrupt Enable

When this bit is set, the VSIF flag is enabled to generate an interrupt request to the CPU. When VSIE is cleared, the VSIF flag is prevented from generating an interrupt request to the CPU. Reset clears this bit.

1 = VSIF bit set will generate interrupt request to CPU

0 = VSIF bit set does not generate interrupt request to CPU

VEDGE — VSync Interrupt Edge Select

This bit specifies the triggering edge of Vsync interrupt. When it is "0", the rising edge of internal Vsync signal which is either from the VSYNC pin or extracted from the composite input signal will set VSIF flag. When it is "1", the falling edge of internal Vsync signal will set VSIF flag. Reset clears this bit.

1 = VSIF bit will be set by rising edge of Vsync

0 = VSIF bit will be set by falling edge of Vsync

VSIF — VSync Interrupt Flag

This flag is only set by the specified edge of the internal Vsync signal, which is either from the VSYNC input pin or extracted from the composite sync input signal. The triggering edge is specified by the VEDGE bit. VSIF generates an interrupt request to the CPU if the VSIE bit is also set. This bit is cleared by writing a "0" to it or by a reset.

1 = A valid edge is detected on the Vsync

0 = No valid Vsync is detected

COMP — Composite Sync Input Enable

This bit is set to enable the separator circuit which extracts the Vsync pulse from the composite sync input on HSYNC or SOG pin (select by SOGSEL bit). The extracted Vsync signal is used as it were from the VSYNC input. Reset clears this bit.

1 = Composite Sync Input Enabled

0 = Composite Sync Input Disabled

VINVO — pVSYNCO Signal Polarity

This bit, together with the ATPOL bit in SPCR1 controls the output polarity of the VSYNCO signal (see [Table 16-5](#)).

HINVO — HSYNCO Signal Polarity

This bit, together with the ATPOL bit in SPCR1 controls the output polarity of the HSYNCO signal (see [Table 16-5](#)).

Table 16-5. ATPOL, VINVO, and HINVO setting

| ATPOL | VINVO / HINVO | Sync Outputs: VSYNCO/HSYNCO |
|-------|---------------|---------------------------------|
| 0 | 0 | Same polarity as sync input |
| 0 | 1 | Inverted polarity of sync input |
| 1 | 0 | Negative polarity sync output |
| 1 | 1 | Positive polarity sync output |

VPOL — Vsync Input Polarity

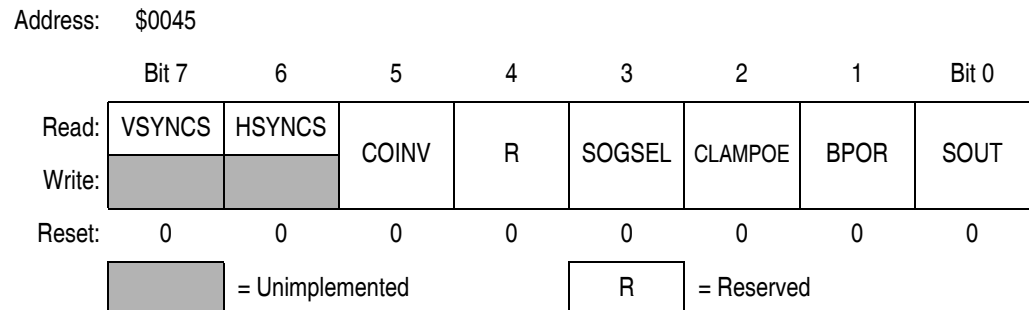
This bit indicates the polarity of the VSYNC input, or the extracted Vsync from a composite sync input (COMP=1). Reset clears this bit.

- 1 = Vsync is positive polarity
- 0 = Vsync is negative polarity

HPOL — pHsync Input Polarity

This bit indicates the polarity of the HSYNC input. This bit equals the VPOL bit when the COMP bit is set. Reset clears this bit.

- 1 = Hsync is positive polarity
- 0 = Hsync is negative polarity

16.6.2 Sync Processor Input/Output Control Register (SPIOCR)

Figure 16-4. Sync Processor Input/Output Control Register (SPIOCR)
VSYNCS — VSYNC Input State

This read-only bit reflects the logical state of the VSYNC input.

HSYNCS — HSYNC Input State

This read-only bit reflects the logical state of the HSYNC input.

COINV — Clamp Output Invert

This bit is set to invert the clamp pulse output to negative. Reset clears this bit.

- 1 = clamp output is set for negative pulses
- 0 = clamp output is set for positive pulses

SOGSEL —p SOG Select

This bit selects either the HSYNC pin or SOG pin as the composite sync signal input pin. Reset clears this bit.

- 1 = SOG pin is used as the composite sync input
- 0 = HSYNC pin is used as the composite sync input

CLAMPOE —pClamp Output Enable

This bit is set to enable the clamp pulse output circuitry. Reset clears this bit.

- 1 = Clamp pulse circuit enabled
- 0 = Clamp pulse circuit disabled

BPOR — Back Porch

This bit defines the triggering edge of the clamp pulse output relative to the HSYNC input. Reset clears this bit.

- 1 = Clamp pulse is generated on the trailing edge of HSYNC
- 0 = Clamp pulse is generated on the leading edge of HSYNC

SOUT — Sync Output Enable

This bit will select the output signals for the VSYNCO and HSYNCO pins. Reset clears this bit.

- 1 = VSYNCO and HSYNCO outputs are internally generated free-running sync pulses with frequencies determined by HVCOR[2:0] bits in HVCOR.
- 0 = VSYNCO and HSYNCO outputs are processed VSYNC and HSYNC inputs respectively

16.6.3 Vertical Frequency Registers (VFRs)

This register pair contains the 13-bit vertical frequency count value, an overflow bit, and the clamp pulse width selection bits.

Address: \$0041

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|------|------|------|------|-----|-------|
| Read: | VOF | 0 | 0 | VF12 | VF11 | VF10 | VF9 | VF8 |
| Write: | | CPW1 | CPW0 | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 16-5. Vertical Frequency High Register

Address: \$0042

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-----|-----|-----|-----|-----|-----|-------|
| Read: | VF7 | VF6 | VF5 | VF4 | VF3 | VF2 | VF1 | VF0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |


 = Unimplemented

Figure 16-6. Vertical Frequency Low Register

VF[12:0] — Vertical Frame Frequency

This read-only 13-bit contains information of the vertical frame frequency. An internal 13-bit counter counts the number of $8\mu\text{s}$ periods between two Vsync pulses. The most significant 5 bits of the counted value is transferred to the high byte register, and the least significant 8 bits is transferred to an intermediate buffer. When the high byte register is read, the 8-bit counted value stored in the intermediate buffer will be uploaded to the low byte register. Therefore, user program must read the high byte register first, then low byte register in order to get the complete counted value of one vertical frame. If the counter overflows, the overflow flag, VOF, will be set, indicating the counter value stored in the VFRs is meaningless. The data corresponds to the period of one vertical frame. This register can be read to determine if the frame frequency is valid, and to determine the video mode.

The frame frequency is calculated by:

$$\text{Vertical Frame Frequency} = \frac{1}{\text{VFR} \pm 1 \times 48 \times t_{\text{CYC}}}$$

$$= \frac{1}{\text{VFR} \pm 1 \times 8\mu\text{s}}$$

for internal bus clock of 6 MHz

Table 16-6 shows examples for the Vertical Frequency Register, all VFR numbers are in hexadecimal.

Table 16-6. Sample Vertical Frame Frequencies

| VFR | Max Freq. | Min Freq. | VFR | Max Freq. | Min Freq. |
|--------|-----------|-----------|--------|-----------|-----------|
| \$02A0 | 186.20 Hz | 185.70 Hz | \$0780 | 65.10 Hz | 65.00 Hz |
| \$03C0 | 130.34 Hz | 130.07 Hz | \$0823 | 60.04 Hz | 59.98 Hz |
| \$03C1 | 130.21 Hz | 129.94 Hz | \$0824 | 60.01 Hz | 59.95 Hz |
| \$03C2 | 130.07 Hz | 129.80 Hz | \$0825 | 59.98 Hz | 59.92 Hz |
| \$04E2 | 100.08 Hz | 99.92 Hz | \$09C4 | 50.02 Hz | 49.98 Hz |
| \$04E3 | 100.00 Hz | 99.84 Hz | \$09C5 | 50.00 Hz | 49.96 Hz |
| \$04E4 | 99.92 Hz | 99.76 Hz | \$09C6 | 49.98 Hz | 49.94 Hz |
| \$06F9 | 70.07 Hz | 69.99 Hz | \$1FFD | 15.266 Hz | 15.262 Hz |
| \$06FA | 70.03 Hz | 69.95 Hz | \$1FFE | 15.264 Hz | 15.260 Hz |
| \$06FB | 69.99 Hz | 69.91 Hz | \$1FFF | 15.262 Hz | 15.258 Hz |

VOF — Vertical Frequency Counter Overflow

This read-only bit is set when an overflow has occurred on the 13-bit vertical frequency counter. Reset clears this bit, and will be updated every vertical frame.

An overflow occurs when the period of Vsync frame exceeds 64.768ms (a vertical frame frequency lower than 15.258Hz).

1 = A vertical frequency counter overflow has occurred

0 = No vertical frequency counter overflow has occurred

CPW[1:0] — Clamp Pulse Width

The CPW1 and CPW0 bits are used to select the output clamp pulse width. Reset clears these bits, selecting a default clamp pulse width between 0.33 μ s and 0.375 μ s. These bits always read as Zeros.

Table 16-7. Clamp Pulse Width

| CPW1 | CPW0 | Clamp Pulse Width |
|------|------|-------------------------------|
| 0 | 0 | 0.33 μ s to 0.375 μ s |
| 0 | 1 | 0.5 μ s to 0.542 μ s |
| 1 | 0 | 0.75 μ s to 0.792 μ s |
| 1 | 1 | 2 μ s to 2.042 μ s |

16.6.4 Hsync Frequency Registers (HFRs)

This register pair contains the 13-bit Hsync frequency count value and an overflow bit.

Address: \$0043

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|------|------|------|------|------|-------|
| Read: | HFH7 | HFH6 | HFH5 | HFH4 | HFH3 | HFH2 | HFH1 | HFH0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 16-7. Hsync Frequency High Register

Address: \$0044

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|---|---|------|------|------|------|-------|
| Read: | HOVER | 0 | 0 | HFL4 | HFL3 | HFL2 | HFL1 | HFL0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |


 = Unimplemented

Figure 16-8. Hsync Frequency Low Register

HFH[7:0], HFL[4:0] — Horizontal Line Frequency

This read-only 13-bit contains the number of horizontal lines in a 32ms window. An internal 13-bit counter counts the Hsync pulses within a 32ms window in every 32.768ms period. If the FSHF bit in SPCR1 is set, only the most 11-bits (HFH[7:0] & HFL[4:2]) will be updated by the counter. Thus, providing a Hsync pulse count in a 8ms window in every 8.192ms.

The most significant 8 bits of counted value is transferred to the high byte register, and the least significant 5 bits is transferred to an intermediate buffer. When the high byte register is read, the 5-bit counted value stored in the intermediate buffer will be uploaded to the low byte register. Therefore, user the program must read the high byte register first then low byte register in order to get the complete counted value of Hsync pulses. If the counter overflows, the overflow flag, HOVER, will be set, indicating the number of Hsync pulses in 32ms are more than 8191 ($2^{13}-1$), i.e. a Hsync frequency greater than 256kHz.

For the 32ms window, the HFHR and HFLR are such that the frequency step unit in the 5-bit of HFLR is 0.03125kHz, and the step unit in the 8-bit HFHR is 1kHz. Therefore, the Hsync frequency can be easily calculated by:

$$\text{Hsync Frequency} = [\text{HFH} + (\text{HFL} \times 0.03125)]\text{kHz}$$

where: *HFH* is the value of HFH[7:0]

HFL is the value of HFL[4:0]

HOVER — Hsync Frequency Counter Overflow

This read-only bit is set when an overflow has occurred on the 13-bit Hsync frequency counter. Reset clears this bit, and will be updated every count period.

An overflow occurs when the number Hsync pulses exceed 8191, a Hsync frequency greater than 256kHz.

1 = A Hsync frequency counter overflow has occurred

0 = No Hsync frequency counter overflow has occurred

16.6.5 Sync Processor Control Register 1 (SPCR1)

Address: \$0046

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-------|------|------|---|---|-------|-------|
| Read: | LVSIE | LVSIF | HPS1 | HPS0 | R | R | ATPOL | FSHF |
| Write: | | 0 | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented
 R = Reserved

Figure 16-9. Sync Processor Control Register 1 (SPCR1)

LVSIE — Low VSync Interrupt Enable

When this bit is set, the LVSIF flag is enabled to generate an interrupt request to the CPU. When LVSIE is cleared, the LVSIF flag is prevented from generating an interrupt request to the CPU. Reset clears this bit.

- 1 = Low Vsync interrupt enabled
- 0 = Low Vsync interrupt disabled

LVSIF — Low VSync Interrupt Flag

This read-only bit is set when the value of VFR is higher than \$C00 (vertical frame frequency below 40.7Hz). LVSIF generates an interrupt request to the CPU if the LVSIE is also set. This bit is cleared by writing a "0" to it or reset.

- 1 = Vertical frequency is below 40.7Hz
- 0 = Vertical frequency is higher than 40.7Hz

HPS[1:0] — HSYNC input Detection Pulse Width

These two bits control the detection pulse width of HSYNC input. Reset clears these two bits, setting a default middle frequency of HSYNC input.

Table 16-8. HSYNC Polarity Detection Pulse Width

| HPS1 | HPS0 | Polarity Detection Pulse Width |
|------|------|--|
| 0 | 0 | Long > 7 μ s and Short < 6 μ s |
| 1 | X | Long > 3.5 μ s and Short < 3 μ s |
| 0 | 1 | Long > 14 μ s and Short < 12 μ s |

ATPOL — Auto Polarity

This bit, together with the VINVO or HINVO bits in SPCSR controls the output polarity of the VSYNCO or HSYNCO signals respectively. Reset clears this bit (see [Table 16-9](#)).

Table 16-9. ATPOL, VINVO, and HINVO setting

| ATPOL | VINVO / HINVO | Sync Outputs: VSYNCO/HSYNCO |
|-------|---------------|---------------------------------|
| 0 | 0 | Same polarity as sync input |
| 0 | 1 | Inverted polarity of sync input |
| 1 | 0 | Negative polarity sync output |
| 1 | 1 | Positive polarity sync output |

FSHF — Fast Horizontal Frequency Count

This bit is set to shorten the measurement cycle of the horizontal frequency. If it is set, only HFH[7:0] and HFL[4:2] will be updated by the Hsync counter, providing a count in a 8ms window in every 8.192ms, with HFL[1:0] reading as zeros. Therefore, user can determine the horizontal frequency change within 8.192ms to protect critical circuitry. Reset clears this bit.

- 1 = Number of Hsync pulses is counted in an 8ms window
- 0 = Number of Hsync pulses is counted in a 32ms window

16.6.6 H&V Sync Output Control Register (HVOCR)

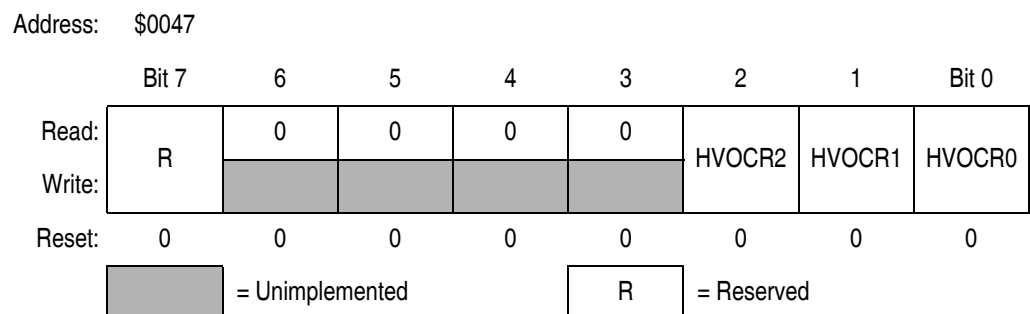


Figure 16-10. H&V Sync Output Control Register (HVOCR)

HVOCR[2:0] — H&V Output Select Bits

These three bits select the frequencies of the internal generated free-running sync pulses for output to HSYNCO and VSYNCO pins, when the SOUT bit is set in the SPIOCR. Reset clears these bits, setting a default horizontal frequency of 31.25kHz and a vertical frequency of 60Hz, a video mode of 640×480.

Table 16-10. Free-Running HSYNC and VSYNC Options

| HVOCR | HSYNCO | | VSYNCO | | Video Mode |
|-------|--------------------|------------|----------------------|-----------|------------|
| | Pulse width | Frequency | Pulse width | Frequency | |
| 000 | Negative 2 μ s | 31.25kHz | Negative 192 μ s | 59.98 Hz | 640×480 |
| 001 | Negative 2 μ s | 43.48kHz | Negative 138 μ s | 84.92 Hz | 640×480 |
| 010 | Negative 2 μ s | 48.78kHz | Negative 123 μ s | 60.00 Hz | 1024×768 |
| 011 | Negative 2 μ s | 54.05kHz | Negative 111 μ s | 84.98 Hz | 800×600 |
| 100 | Negative 2 μ s | 60.61 kHz | Negative 99 μ s | 75.01 Hz | 1024×768 |
| 101 | Negative 2 μ s | 80.00kHz | Negative 75 μ s | 74.98 Hz | 1280×1024 |
| 110 | Negative 2 μ s | 90.91 kHz | Negative 66 μ s | 84.96 Hz | 1280×1024 |
| 111 | Negative 2 μ s | 105.26 kHz | Negative 57 μ s | 85.02 Hz | 1600×1200 |

16.7 System Operation

This Sync Processor is designed to assist in determining the video mode of incoming HSYNC and VSYNC of various frequencies and polarities, and DPMS modes. In the DPMS standard, a no sync pulses definition can be detected when the value of the Hsync Frequency Register (the number of Hsync pulses) is less than one or when the VOF bit is set. Since the Hsync Frequency Register is updated repeatedly in every 32.768ms, and a valid Vsync must have a frequency greater than 40.7Hz, a valid Vsync pulse will arrive within the 32.768ms window. Therefore, the user should read the Hsync Frequency Register every 32.768ms to determine the presence of Hsync and/or Vsync pulses.



Section 17. Input/Output (I/O) Ports

17.1 Contents

| | | |
|--------|---------------------------|-----|
| 17.2 | Introduction | 229 |
| 17.3 | Port A | 233 |
| 17.3.1 | Port A Data Register | 233 |
| 17.3.2 | Data Direction Register A | 234 |
| 17.3.3 | Port A Options | 235 |
| 17.4 | Port B | 236 |
| 17.4.1 | Port B Data Register | 236 |
| 17.4.2 | Data Direction Register B | 237 |
| 17.4.3 | Port B Options | 238 |
| 17.5 | Port C | 239 |
| 17.5.1 | Port C Data Register | 239 |
| 17.5.2 | Data Direction Register C | 240 |
| 17.5.3 | Port C Options | 241 |
| 17.6 | Port D | 242 |
| 17.6.1 | Port D Data Register | 242 |
| 17.6.2 | Data Direction Register D | 243 |
| 17.6.3 | Port D Options | 245 |
| 17.7 | Port E | 247 |
| 17.7.1 | Port E Data Register | 247 |
| 17.7.2 | Data Direction Register E | 248 |
| 17.7.3 | Port E Options | 250 |

17.2 Introduction

Thirty-two (32) 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.

Table 17-1. I/O Port Register Summary

| 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: | PTB7 | PTB6 | PTB5 | PTB4 | PTB3 | PTB2 | PTB1 | PTB0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0002 | Port C Data Register (PTC) | Read: | 0 | 0 | PTC5 | PTC4 | PTC3 | PTC2 | PTC1 | PTC0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0003 | Port D Data Register (PTD) | Read: | 0 | 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: | DDRB7 | DDRB6 | DDRB5 | DDRB4 | DDRB3 | DDRB2 | DDRB1 | DDRB0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0006 | Data Direction Register C (DDRC) | Read: | 0 | 0 | DDRC5 | DDRC4 | DDRC3 | DDRC2 | DDRC1 | DDRC0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

Table 17-1. I/O Port Register Summary (Continued)

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--------|--------------------------------------|--------|---------------------|---------|---------|--------|---------|---------|--------|--------|
| \$0007 | Data Direction Register D (DDRD) | Read: | 0 | DDRD6 | DDRD5 | DDRD4 | DDRD3 | DDRD2 | DDRD1 | DDRD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0008 | Port E Data Register (PTE) | Read: | 0 | 0 | 0 | 0 | 0 | PTE2 | PTE1 | PTE0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0009 | Data Direction Register E (DDRE) | Read: | 0 | 0 | 0 | 0 | 0 | DDRE2 | DDRE1 | DDRE0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$001D | Configuration Register 0 (CONFIG0) | Read: | HSYNCOE | VSYNCOE | SOGE | 0 | 0 | 0 | 0 | 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0028 | PWM Control Register 1 (PWMCR1) | Read: | PWM7E | PWM6E | PWM5E | PWM4E | PWM3E | PWM2E | PWM1E | PWM0E |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0049 | Port D Configuration Register (PDCR) | Read: | 0 | IICDATE | IICSCLE | CLAMPE | DDCSCLE | DDCDATE | USBD-E | USBD+E |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0059 | PWM Control Register 2 (PWMCR2) | Read: | PWM15E | PWM14E | PWM13E | PWM12E | PWM11E | PWM10E | PWM9E | PWM8E |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

Table 17-2. Port Control Register Bits Summary

| Port | Bit | DDR | Module Control | | | Pin |
|------|-----|-------|----------------|-------------------|-------------|---------------|
| | | | Module | Register | Control Bit | |
| A | 0 | DDRA0 | PWM | PWMCR2 \$0059 | PWM8E | PTA0/PWM8 |
| | 1 | DDRA1 | | | PWM9E | PTA1/PWM9 |
| | 2 | DDRA2 | | | PWM10E | PTA2/PWM10 |
| | 3 | DDRA3 | | | PWM11E | PTA3/PWM11 |
| | 4 | DDRA4 | | | PWM12E | PTA4/PWM12 |
| | 5 | DDRA5 | | | PWM13E | PTA5/PWM13 |
| | 6 | DDRA6 | | | PWM14E | PTA6/PWM14 |
| | 7 | DDRA7 | | | PWM15E | PTA7/PWM15 |
| B | 0 | DDRB0 | PWM | PWMCR1 \$0028 | PWM0E | PTB0/PWM0 |
| | 1 | DDRB1 | | | PWM1E | PTB1/PWM1 |
| | 2 | DDRB2 | | | PWM2E | PTB2/PWM2 |
| | 3 | DDRB3 | | | PWM3E | PTB3/PWM3 |
| | 4 | DDRB4 | | | PWM4E | PTB4/PWM4 |
| | 5 | DDRB5 | | | PWM5E | PTB5/PWM5 |
| | 6 | DDRB6 | | | PWM6E | PTB6/PWM6 |
| | 7 | DDRB7 | | | PWM7E | PTB7/PWM7 |
| C | 0 | DDRC0 | ADC | ADSCR \$005D | ADCH[4:0] | PTC0/ADC0 |
| | 1 | DDRC1 | | | | PTC1/ADC1 |
| | 2 | DDRC2 | | | | PTC2/ADC2 |
| | 3 | DDRC3 | | | | PTC3/ADC3/ |
| | 4 | DDRC4 | | | | PTC4/ADC4 |
| | 5 | DDRC5 | | | | PTC5/ADC5 |
| D | 0 | DDRD0 | USB | PDCR \$0049 | D+E | PTD0/D+ |
| | 1 | DDRD1 | DDC12AB | | D-E | PTD1/D- |
| | 2 | DDRD2 | | | DDCDATE | PTD2/DDCSDA |
| | 3 | DDRD3 | DDCSCLE | | PTD3/DDCSCL | |
| | 4 | DDRD4 | SYNC | | CLAMPE | PTD4/CLAMP |
| | 5 | DDRD5 | MMIIC | | IICSCLE | PTD5/IICSCSCL |
| | 6 | DDRD6 | | | IICDATE | PTD6/IICSDA |
| E | 0 | DDRE0 | SYNC/TIM | CONFIG0 \$001D | SOGE | PTE0/SOG/TCH0 |
| | 1 | DDRE1 | SYNC | | HSYNCOE | PTE1/HSYNCO |
| | 2 | DDRE2 | | | VSYNCOE | PTE2/VSYNCO |

17.3 Port A

Port A is an 8-bit special-function port that shares all eight of its pins with the pulse width modulator (PWM).

17.3.1 Port A Data Register

The port A data register (PTA) 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: | PTA7 | PTA6 | PTA5 | PTA4 | PTA3 | PTA2 | PTA1 | PTA0 |
| Write: | | | | | | | | |
| Reset: | Unaffected by reset | | | | | | | |
| Alternate Function: | PWM15 | PWM14 | PWM13 | PWM12 | PWM11 | PWM10 | PWM9 | PWM8 |

Figure 17-1. Port A Data Register (PTA)

PTA7–PTA0 — 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.

PWM15–PWM8 — PWM Outputs 15–8

The PWM output enable bits PWM15E–PWM8E, in PWM control register 2 (PWMCR2) enable port A pins as PWM output pins. (See [17.3.3 Port A Options](#).)

17.3.2 Data Direction Register A

Data direction register A (DDRA) 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.

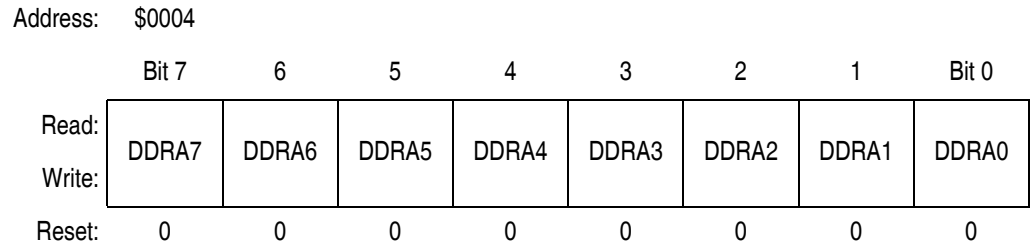


Figure 17-2. Data Direction Register A (DDRA)

DDRA7–DDRA0 — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA7–DDRA0, 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 17-3 shows the port A I/O logic.

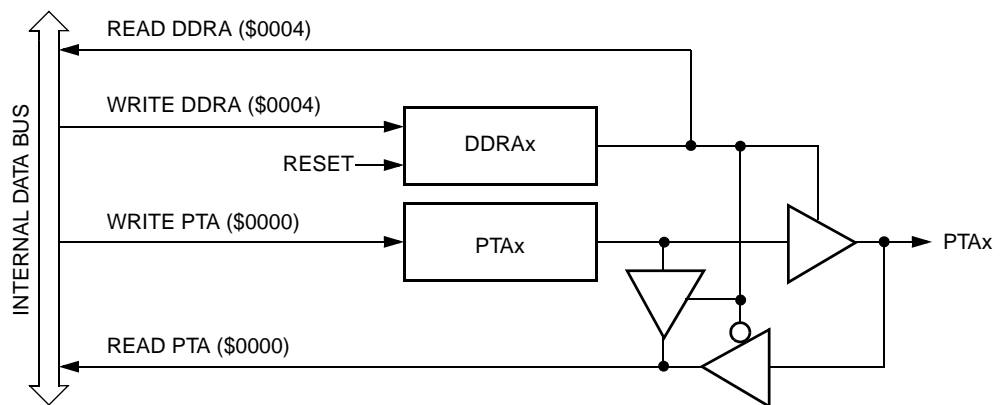


Figure 17-3. Port A I/O Circuit

When bit DDRAx is a logic 1, reading address \$0000 reads the PTAx data latch. When bit 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 17-3](#) summarizes the operation of the port A pins.

Table 17-3. Port A Pin Functions

| PTAPUE Bit | DDRA Bit | PTA Bit | I/O Pin Mode | Accesses to DDRA | Accesses to PTA | |
|------------|----------|------------------|----------------------------|------------------|-----------------|--------------------------|
| | | | | Read/Write | Read | Write |
| 0 | 0 | X ⁽¹⁾ | Input, Hi-Z ⁽²⁾ | DDRA7–DDRA0 | Pin | PTA7–PTA0 ⁽³⁾ |
| X | 1 | X | Output | DDRA7–DDRA0 | PTA7–PTA0 | PTA7–PTA0 |

NOTES:

1. X = Don't care
2. Hi-Z = High impedance
3. Writing affects data register, but does not affect input.

17.3.3 Port A Options

The PWM control register 2 (PWMCR2) selects the port A pins for PWM function or as standard I/O function. See [11.4.2 PWM Control Registers 1 and 2 \(PWMCR1:PWMCR2\)](#).

Address: \$0059

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|--------|--------|--------|--------|--------|--------|-------|-------|
| Read: | PWM15E | PWM14E | PWM13E | PWM12E | PWM11E | PWM10E | PWM9E | PWM8E |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 17-4. PWM Control Register 1 (PWMCR1)

PWM15E–PWM8E — PWM Output Enable 15–8

Setting a bit to "1" will configure the corresponding PTAx/PWMx pin for PWM output function. Reset clears these bits.

- 1 = PTAx/PWMx pin configured as PWMx output pin
- 0 = PTAx/PWMx pin configured as standard I/O pin

17.4 Port B

Port B is an 8-bit special-function port that shares all eight of its pins with the pulse width modulator (PWM).

17.4.1 Port B Data Register

The port B data register (PTB) contains a data latch for each of the eight port pins.

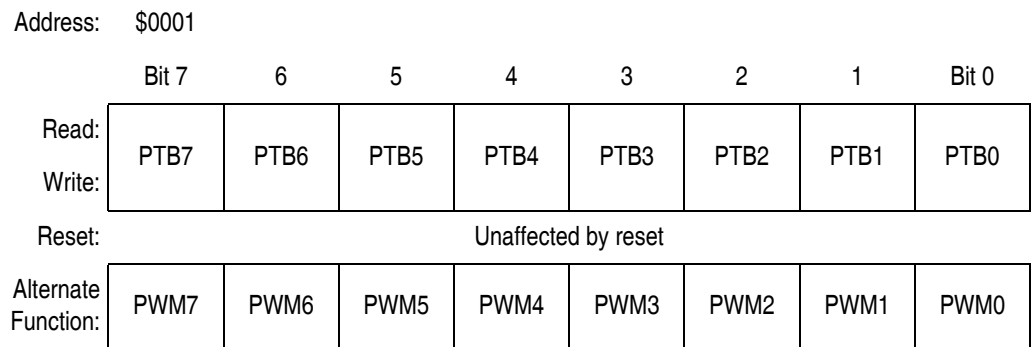


Figure 17-5. Port B Data Register (PTB)

PTB7–PTB0 — 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.

PWM7–PWM0 — PWM Outputs 7–0

The PWM output enable bits PWM7E–PWM0E, in PWM control register 1 (PWMCR1) enable port B pins as PWM output pins. (See [17.4.3 Port B Options](#).)

17.4.2 Data Direction Register B

Data direction register B (DDRB) 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: | DDRB7 | DDRB6 | DDRB5 | DDRB4 | DDRB3 | DDRB2 | DDRB1 | DDRB0 |
| Write: | DDRB7 | DDRB6 | DDRB5 | DDRB4 | DDRB3 | DDRB2 | DDRB1 | DDRB0 |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 17-6. Data Direction Register B (DDRB)

DDRB7–DDRB0 — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB7–DDRB0, 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 17-7 shows the port B I/O logic.

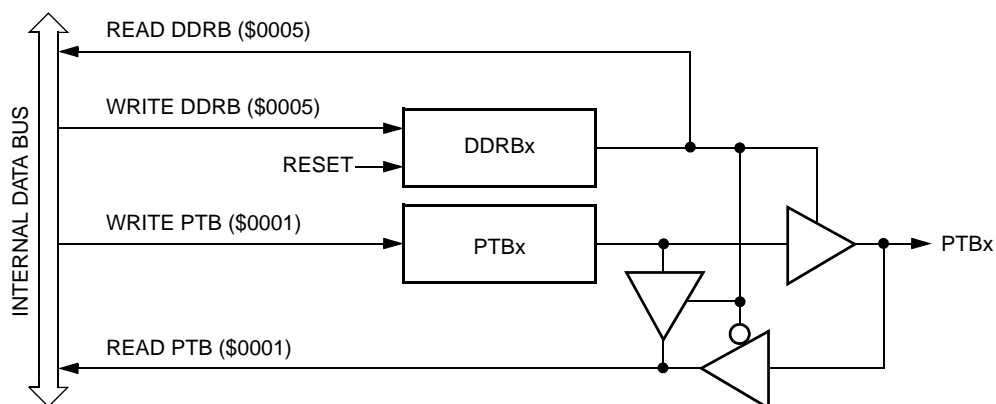


Figure 17-7. Port B I/O Circuit

When bit DDRBx is a logic 1, reading address \$0001 reads the PTBx data latch. When bit 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 17-4](#) summarizes the operation of the port B pins.

Table 17-4. 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⁾ | DDRB7–DDRB0 | Pin | PTB7–PTB0 ⁽³⁾ |
| 1 | X | Output | DDRB7–DDRB0 | PTB7–PTB0 | PTB7–PTB0 |

Notes:

1. X = Don't care
2. Hi-Z = High impedance
3. Writing affects data register, but does not affect input.

17.4.3 Port B Options

The PWM control register 1 (PWMCR1) selects the port B pins for PWM function or as standard I/O function. See [11.4.2 PWM Control Registers 1 and 2 \(PWMCR1:PWMCR2\)](#).

Address: \$0028

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Read: | PWM7E | PWM6E | PWM5E | PWM4E | PWM3E | PWM2E | PWM1E | PWM0E |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 17-8. PWM Control Register 1 (PWMCR1)

PWM7E–PWM0E — PWM Output Enable 7–0

Setting a bit to "1" will configure the corresponding PTBx/PWMx pin for PWM output function. Reset clears these bits.

- 1 = PTBx/PWMx pin configured as PWMx output pin
- 0 = PTBx/PWMx pin configured as standard I/O pin

17.5 Port C

Port C is a 6-bit special-function port that shares all six of its pins with the analog-to-digital converter (ADC) module.

17.5.1 Port C Data Register

The port C data register (PTC) contains a data latch for each of the seven port C pins.

Address: \$0002

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---------------------|---------------------|---|------|------|------|------|------|-------|
| Read: | 0 | 0 | PTC5 | PTC4 | PTC3 | PTC2 | PTC1 | PTC0 |
| Write: | | | | | | | | |
| Reset: | Unaffected by reset | | | | | | | |
| Alternate Function: | | | ADC5 | ADC4 | ADC3 | ADC2 | ADC1 | ADC0 |


 = Unimplemented

Figure 17-9. Port C Data Register (PTC)

PTC5–PTC0 — 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.

ADC5–ADC0 — Analog-to-Digital Input Bits

ADC5–ADC0 are pins used for the input channels to the analog-to-digital converter module. The channel select bits in the [ADC Status and Control Register](#) define which port C pin will be used as an ADC input and overrides any control from the port I/O logic by forcing that pin as the input to the analog circuitry.

NOTE: Care must be taken when reading port C while applying analog voltages to ADC5–ADC0 pins. If the appropriate ADC channel is not enabled, excessive current drain may occur if analog voltages are applied to the PTCx/ADCx pin, while PTC is read as a digital input. Those ports not selected as analog input channels are considered digital I/O ports.

17.5.2 Data Direction Register C

Data direction register C (DDRC) 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: | 0 | 0 | DDRC5 | DDRC4 | DDRC3 | DDRC2 | DDRC1 | DDRC0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

Figure 17-10. Data Direction Register C (DDRC)

DDRC5–DDRC0 — Data Direction Register C Bits

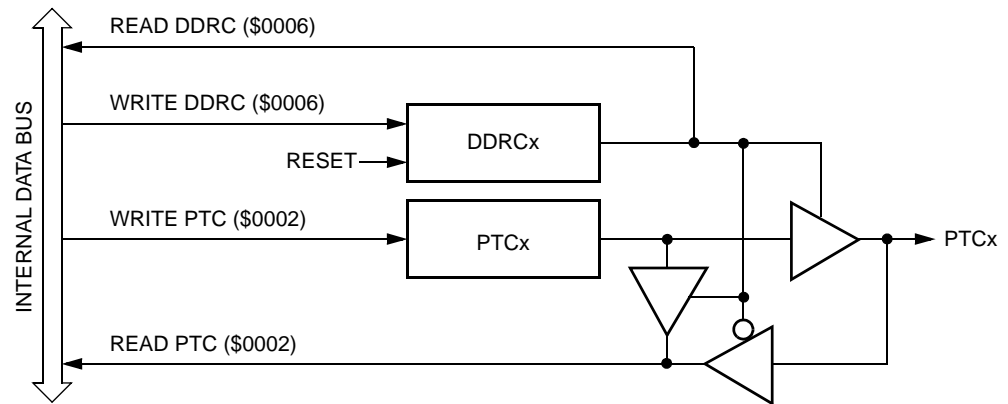
These read/write bits control port C data direction. Reset clears DDRC5–DDRC0, configuring all port C pins as inputs.

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 17-11 shows the port C I/O logic.


Figure 17-11. Port C I/O Circuit

When bit DDRCx is a logic 1, reading address \$0002 reads the PTCx data latch. When bit 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 17-5](#) summarizes the operation of the port C pins.

Table 17-5. Port C Pin Functions

| PTCPUE Bit | DDRC Bit | PTC Bit | I/O Pin Mode | Accesses to DDRC | | Accesses to PTC | |
|------------|----------|---------|----------------------------|------------------|--|-----------------|--------------------------|
| | | | | Read/Write | | Read | Write |
| 0 | 0 | X | Input, Hi-Z ⁽²⁾ | DDRC5–DDRC0 | | Pin | PTC5–PTC0 ⁽³⁾ |
| X | 1 | X | Output | DDRC5–DDRC0 | | PTC5–PTC0 | PTC5–PTC0 |

Notes:

1. X = Don't care
2. Hi-Z = High impedance
3. Writing affects data register, but does not affect input.

17.5.3 Port C Options

The ADCH4–ADCH0 bits in the [ADC Status and Control Register](#) (ADSCR) defines which PTCx/ADCx pin is used as an ADC input and overrides any control from the port I/O logic by forcing that pin as the input to the analog circuitry. See [12.8.1 ADC Status and Control Register](#).

17.6 Port D

Port D is an 7-bit special-function port that shares two of its pins with the multi-master IIC (MMIIC) module, one of its pins with the sync processor, two of its pins with the DDC12AB module, and two of its pins with the USB module.

NOTE: *PTD1–PTD0 are 3.3V pins.*

17.6.1 Port D Data Register

The port D data register (PTD) contains a data latch for each of the eight port D pins.

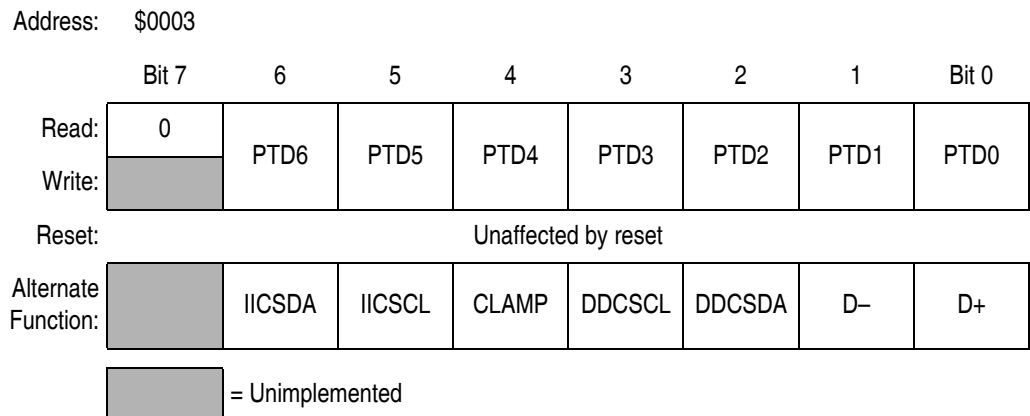


Figure 17-12. Port D Data Register (PTD)

PTD6–PTD0 — 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.

IICSDA, IICSCSCL — Multi-master IIC Data and Clock pins

The PTD6/IICSDA and PTD5/IICSCSCL pins are multi-master IIC data and clock pins. When the IICDATE and IICSCLE bits in the port D configuration register (PDCR) is clear, the PTD6/IICSDA and PTD5/IICSCSCL pins are available for general-purpose I/O. See [17.6.3 Port D Options](#).

CLAMP — Sync Processor Clamp pulse output pin

The PTD4/CLAMP pin is the sync processor clamp pulse output pin. When the CLAMPE bit in the port D configuration register (PDCR) is clear, the PTD4/CLAMP pin is available for general-purpose I/O. See [17.6.3 Port D Options](#).

DDCSCL, DDCSDA — DDC12AB Data and Clock pins

The PTD3/DDCSCL and PTD2/DDCSDA pins are DDC12AB clock and data pins respectively. When the DDCSCLE and DDCDATE bits in the port D configuration register (PDCR) is clear, the PTD3/DDCSCL and PTD2/DDCSDA pins are available for general-purpose I/O. See [17.6.3 Port D Options](#).

D–, D+ — USB I/O pins

The PTD1/D– and PTD0/D+ pins are the USB port pins. When the USBD–E and USBD+E bits in the port D configuration register (PDCR) is clear, the PTD1/D– and PTD0/D+ pins are available for general-purpose I/O. See [17.6.3 Port D Options](#).

17.6.2 Data Direction Register D

Data direction register D (DDRD) 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: | 0 | DDRD6 | DDRD5 | DDRD4 | DDRD3 | DDRD2 | DDRD1 | DDRD0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 17-13. Data Direction Register D (DDRD)

DDRD6–DDRD0 — Data Direction Register D Bits

These read/write bits control port D data direction. Reset clears DDRD6–DDRD0, 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 17-14 shows the port D I/O logic.

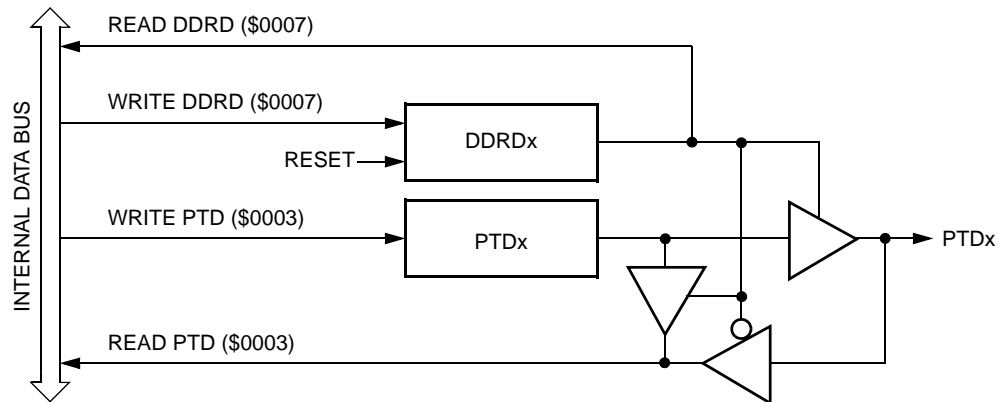


Figure 17-14. Port D I/O Circuit

When bit DDRDx is a logic 1, reading address \$0003 reads the PTDx data latch. When bit 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 17-6 summarizes the operation of the port D pins.

Table 17-6. Port D Pin Functions

| PTDPUE Bit | DDRD Bit | PTD Bit | I/O Pin Mode | Accesses to DDRD | | Accesses to PTD | |
|------------|----------|---------|----------------------------|------------------|--|-----------------|--------------------------|
| | | | | Read/Write | | Read | Write |
| 0 | 0 | X | Input, Hi-Z ⁽²⁾ | DDRD6–DDRD0 | | Pin | PTD6–PTD0 ⁽³⁾ |
| X | 1 | X | Output | DDRD6–DDRD0 | | PTD6–PTD0 | PTD6–PTD0 |

Notes:

1. X = Don't care
2. Hi-Z = High impedance
3. Writing affects data register, but does not affect input.

17.6.3 Port D Options

The port D configuration register (PDCR) selects the port D pins for module function or as standard I/O function.

Address: \$0049

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|---------|---------|--------|---------|----------|-------|-------|
| Read: | 0 | IICDATE | IICSCLE | CLAMPE | DDCSCLE | DDCCDATE | USBDE | USBDE |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |


 = Unimplemented

Figure 17-15. Port D Configuration Register (PDCR)

IICDATE — MMIIC Data Pin Enable

This bit is set to configure the PTD6/IICSDA pin for IICSDA function. Reset clears this bit.

- 1 = PTD6/IICSDA pin configured as IICSDA pin
- 0 = PTD6/IICSDA pin configured as standard I/O pin

IICSCLE — MMIIC Clock Pin Enable

This bit is set to configure the PTD5/IICSCSCL pin for IICSCSCL function. Reset clears this bit.

- 1 = PTD5/IICSCSCL pin configured as IICSCSCL pin
- 0 = PTD5/IICSCSCL pin configured as standard I/O pin

CLAMP — CLAMP Pin Enable

This bit is set to configure the PTD4/CLAMP pin for sync processor clamp pulse output. Reset clears this bit.

- 1 = PTD4/CLAMP pin configured as CLAMP pin
- 0 = PTD4/CLAMP pin configured as standard I/O pin

DDCSCLE — DDC Clock Pin Enable

This bit is set to configure the PTD3/DDCSCL pin for DDCSCL function. Reset clears this bit.

- 1 = PTD3/DDCSCL pin configured as DDCSCL pin
- 0 = PTD3/DDCSCL pin configured as standard I/O port

DDCDATE — DDC Data Pin Enable

This bit is set to configure the PTD2/DDCSDA pin for DDCSDA function. Reset clears this bit.

- 1 = PTD2/DDCSDA pin configured as DDCSDA pin
- 0 = PTD2/DDCSDA pin configured as standard I/O port

USBD-E — USB D- Pin Enable

This bit is set to configure the PTD1/D- pin for D- function. Reset clears this bit.

- 1 = PTD1/D- pin configured as D- pin
- 0 = PTD1/D- pin configured as standard I/O port

USBD+E — USB D+ Pin Enable

This bit is set to configure the PTD0/D+ pin for D+ function. Reset clears this bit.

- 1 = PTD0/D+ pin configured as D+ pin
- 0 = PTD0/D+ pin configured as standard I/O port

17.7 Port E

Port E is a 3-bit special-function port that shares all of its pins with the sync processor.

17.7.1 Port E Data Register

The port E data register contains a data latch for each of the three port E pins.

Address: \$0008

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---------------------|---------------------|---|---|---|---|--------|--------|-------------|
| Read: | 0 | 0 | 0 | 0 | 0 | PTE2 | PTE1 | PTE0 |
| Write: | | | | | | | | |
| Reset: | Unaffected by reset | | | | | | | |
| Alternate Function: | | | | | | VSYNCO | HSYNCO | SOG or TCH0 |


 = Unimplemented

Figure 17-16. Port E Data Register (PTE)

PTE2–PTE0 — Port E Data Bits

PTE2–PTE0 are read/write, software programmable bits. Data direction of each port E pin is under the control of the corresponding bit in data direction register E.

VSYNCO — Vsync Output

The PTE2/VSYNCO pin is the Vsync output from the sync processor. When the VSYNCOE is clear, the PTE2/VSYNCO pin is available for general-purpose I/O. See [17.7.3 Port E Options](#).

HSYNC — Hsync Output

The PTE1/HSYNCO pin is the Hsync output from the sync processor. When the HSYNCOE is clear, the PTE1/HSYNCO pin is available for general-purpose I/O. See [17.7.3 Port E Options](#).

SOG/TCH0 — SOG Output or TCH0 Input

The PTE0/SOG/TCH0 pin is the SOG input for the sync processor or the input capture of the TIM channel 0. See [17.7.3 Port E Options](#).

17.7.2 Data Direction Register E

Data direction register E (DDRE) determines whether each port E pin is an input or an output. Writing a logic 1 to a DDRE bit enables the output buffer for the corresponding port E pin; a logic 0 disables the output buffer.

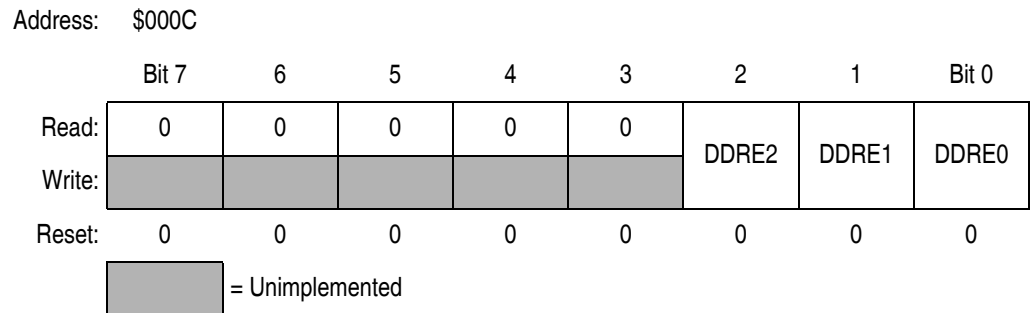


Figure 17-17. Data Direction Register E (DDRE)

DDRE2–DDRE0 — Data Direction Register E Bits

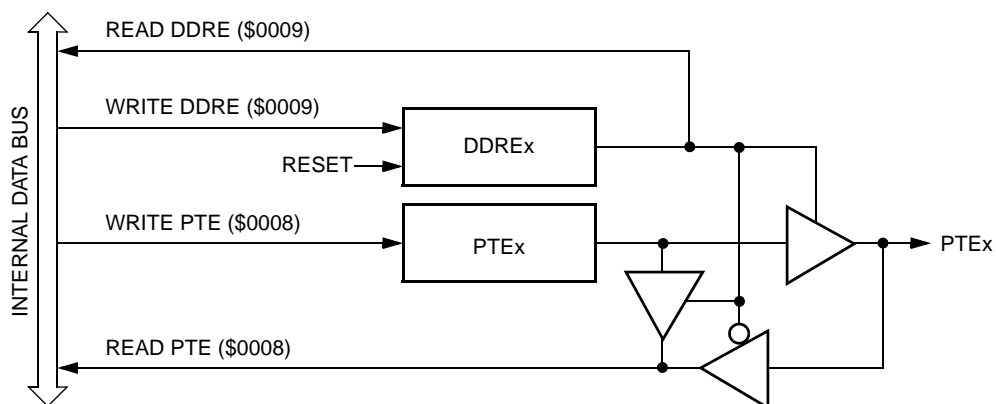
These read/write bits control port E data direction. Reset clears DDRE2–DDRE0, configuring all port E pins as inputs.

1 = Corresponding port E pin configured as output

0 = Corresponding port E pin configured as input

NOTE: *Avoid glitches on port E pins by writing to the port E data register before changing data direction register E bits from 0 to 1.*

[Figure 17-18](#) shows the port E I/O logic.


Figure 17-18. Port E I/O Circuit

When bit DDREx is a logic 1, reading address \$0008 reads the PTE_x data latch. When bit DDREx is a logic 0, reading address \$0008 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. [Table 17-7](#) summarizes the operation of the port E pins.

Table 17-7. Port E Pin Functions

| DDRE Bit | PTE Bit | I/O Pin Mode | Accesses to DDRE | Accesses to PTE | |
|----------|------------------|----------------------------|------------------|-----------------|--------------------------|
| | | | Read/Write | Read | Write |
| 0 | X ⁽¹⁾ | Input, Hi-Z ⁽²⁾ | DDRE2–DDRE0 | Pin | PTE2–PTE0 ⁽³⁾ |
| 1 | X | Output | DDRE2–DDRE0] | PTE2–PTE0 | PTE2–PTE0 |

Notes:

1. X = Don't care
2. Hi-Z = High impedance
3. Writing affects data register, but does not affect input.

17.7.3 Port E Options

The configuration register 0 (CONFIG0) selects the port E pins for module function or as standard I/O function.

Address: \$001D

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|---------|---------|------|---|---|---|---|-------|
| Read: | HSYNCOE | VSYNCOE | SOGE | 0 | 0 | 0 | 0 | 0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

Figure 17-19. Configuration Register 0 (CONFIG0)

HSYNCOE — VSYNCO Enable

This bit is set to configure the PTE1/HSYNCO pin for HSYNCO output function. Reset clears this bit.

- 1 = PTE1/HSYNCO pin configured as HSYNCO pin
- 0 = PTE1/HSYNCO pin configured as standard I/O pin

VSYNCOE — VSYNCO Enable

This bit is set to configure the PTE2/VSYNCO pin for VSYNCO output function. Reset clears this bit.

- 1 = PTE2/VSYNCO pin configured as VSYNCO pin
- 0 = PTE2/VSYNCO pin configured as standard I/O pin

SOGE — SOG Enable

This bit is set to configure the PTE0/SOG/TCH0 pin for SOG output function. Reset clears this bit.

- 1 = PTE0/SOG/TCH0 pin configured as SOG pin
- 0 = PTE0/SOG/TCH0 pin configured as standard I/O or TCH0 pin.
TCH0 function is configured by ELS0B and ELS0A bits in TSC0 (bits 3 and 2 in \$0010).

Section 18. External Interrupt (IRQ)

18.1 Contents

| | | |
|------|--|-----|
| 18.2 | Introduction | 251 |
| 18.3 | Features | 251 |
| 18.4 | Functional Description | 252 |
| 18.5 | $\overline{\text{IRQ}}$ Pin | 254 |
| 18.6 | IRQ Module During Break Interrupts | 255 |
| 18.7 | IRQ Status and Control Register | 255 |

18.2 Introduction

The IRQ (external interrupt) module provides a maskable interrupt input.

18.3 Features

Features of the IRQ module include:

- A dedicated external interrupt pin ($\overline{\text{IRQ}}$)
- IRQ interrupt control bits
- Hysteresis buffer
- Programmable edge-only or edge and level interrupt sensitivity
- Automatic interrupt acknowledge
- Internal pullup resistor

18.4 Functional Description

A logic 0 applied to the external interrupt pin can latch a CPU interrupt request. [Figure 18-1](#) 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 MODE 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.

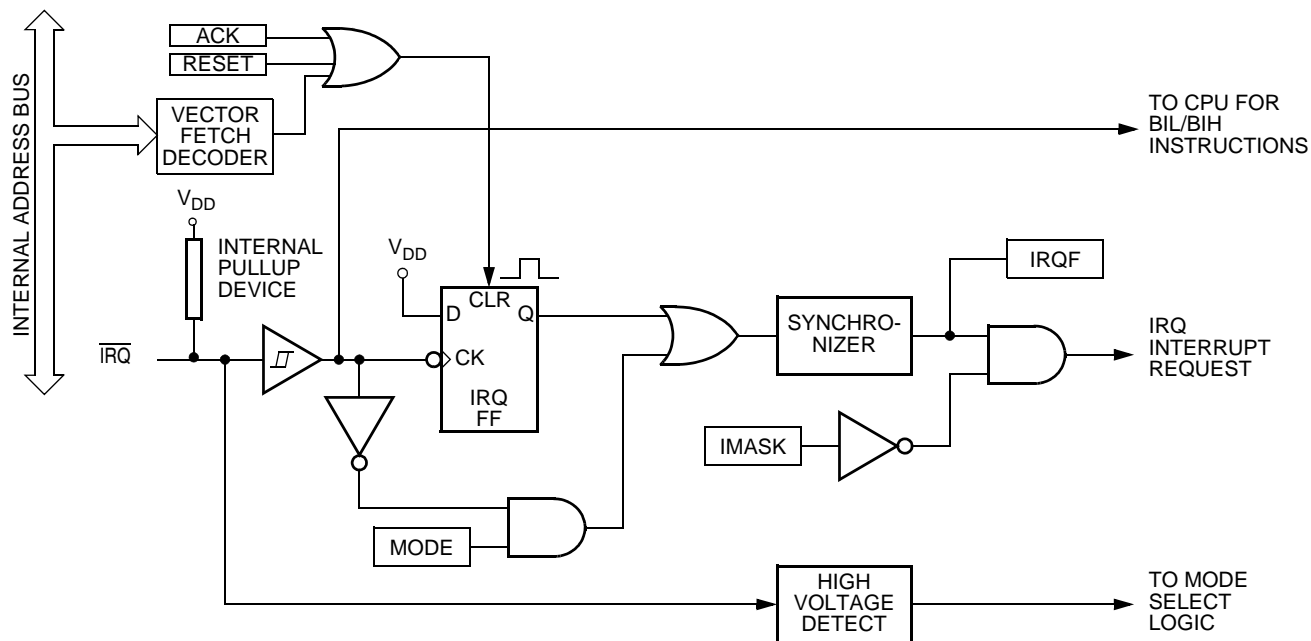


Figure 18-1. IRQ Module Block Diagram

Table 18-1. IRQ I/O Register Summary

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--------|--|--------|-----------------|---|---|---|------|-----|-------|------|
| \$001E | IRQ Status and Control Register (INTSCR) | Read: | 0 | 0 | 0 | 0 | IRQF | 0 | IMASK | MODE |
| | | Write: | [Unimplemented] | | | | | ACK | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

[Unimplemented] = Unimplemented

18.5 $\overline{\text{IRQ}}$ Pin

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 locations \$FFFA and \$FFFB.
- 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 $\overline{\text{IRQ}}$ pin.

NOTE: *When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine.*

18.6 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 20. 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.

18.7 IRQ Status and Control Register

The IRQ status and control register (INTSCR) controls and monitors operation of the IRQ module. The INTSCR:

- Shows the state of the IRQ flag
- Clears the IRQ latch
- Masks IRQ interrupt request
- Controls triggering sensitivity of the $\overline{\text{IRQ}}$ interrupt pin

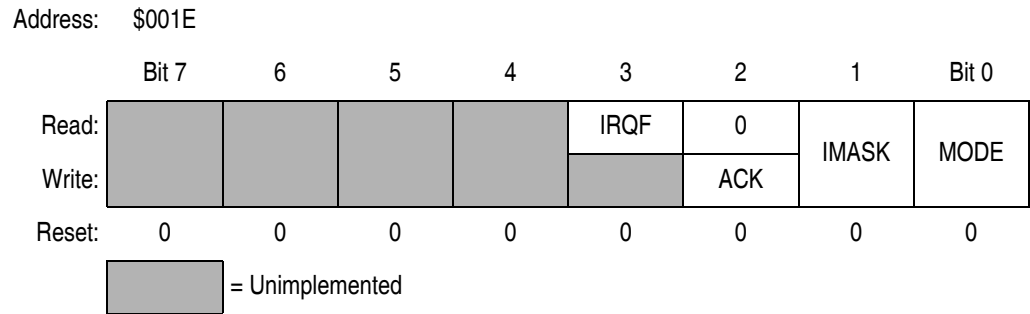


Figure 18-2. IRQ Status and Control Register (INTSCR)

IRQF — IRQ Flag Bit

This read-only status bit is high when the IRQ interrupt is pending.

1 = $\overline{\text{IRQ}}$ interrupt pending

0 = $\overline{\text{IRQ}}$ interrupt not pending

ACK — IRQ Interrupt Request Acknowledge Bit

Writing a logic 1 to this write-only bit clears the IRQ latch. ACK always reads as logic 0. Reset clears ACK.

IMASK — IRQ Interrupt Mask Bit

Writing a logic 1 to this read/write bit disables IRQ interrupt requests. Reset clears IMASK.

1 = IRQ interrupt requests disabled

0 = IRQ interrupt requests enabled

MODE — IRQ Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the $\overline{\text{IRQ}}$ pin. Reset clears MODE.

1 = $\overline{\text{IRQ}}$ interrupt requests on falling edges and low levels

0 = $\overline{\text{IRQ}}$ interrupt requests on falling edges only

Section 19. Computer Operating Properly (COP)

19.1 Contents

| | | |
|--------|------------------------------|-----|
| 19.2 | Introduction | 257 |
| 19.3 | Functional Description | 258 |
| 19.4 | I/O Signals | 259 |
| 19.4.1 | OSCCLK | 259 |
| 19.4.2 | STOP Instruction | 259 |
| 19.4.3 | COPCTL Write | 259 |
| 19.4.4 | Power-On Reset | 259 |
| 19.4.5 | Internal Reset | 260 |
| 19.4.6 | Reset Vector Fetch | 260 |
| 19.4.7 | COPD (COP Disable) | 260 |
| 19.4.8 | COPRS (COP Rate Select) | 260 |
| 19.5 | COP Control Register | 261 |
| 19.6 | Interrupts | 261 |
| 19.7 | Monitor Mode | 261 |
| 19.8 | Low-Power Modes | 261 |
| 19.8.1 | Wait Mode | 262 |
| 19.8.2 | Stop Mode | 262 |
| 19.9 | COP Module During Break Mode | 262 |

19.2 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 CONFIG register.

19.3 Functional Description

Figure 19-1 shows the structure of the COP module.

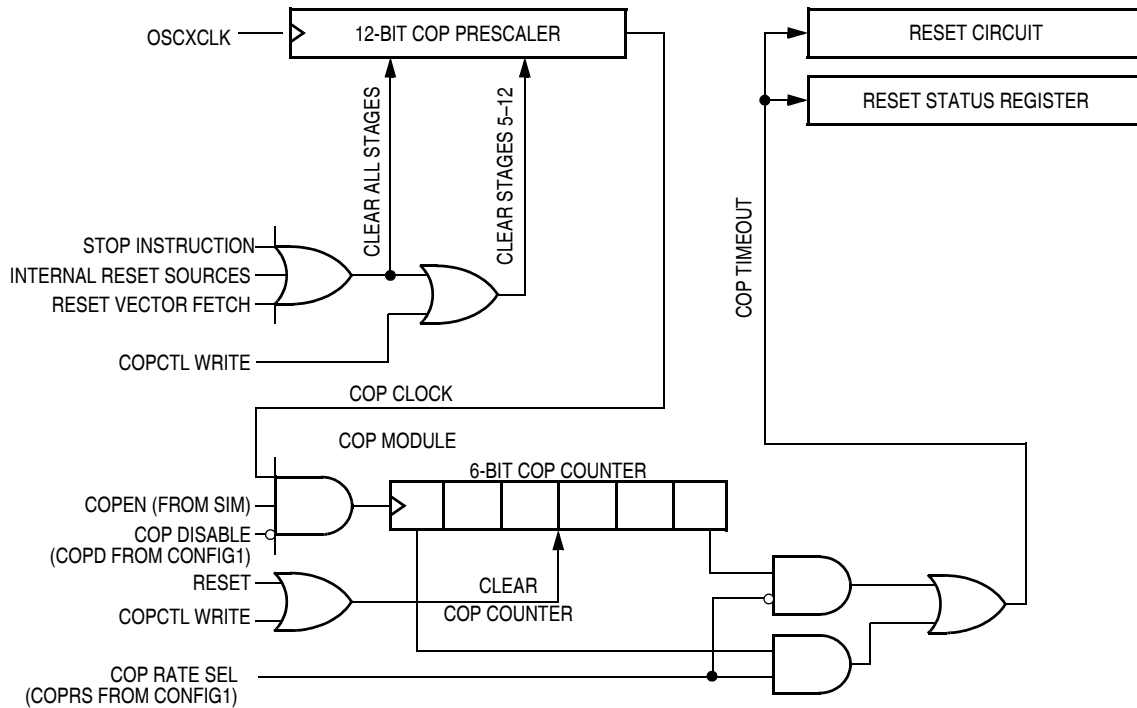


Figure 19-1. COP Block Diagram

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 $2^{13} - 2^4$ OSCXCLK cycles, depending on the state of the COP rate select bit, COPRS, in configuration register 1. With a $2^{18} - 2^4$ OSCXCLK cycle overflow option, a 24MHz crystal gives a COP timeout period of 10.922ms. 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 OSCXCLK 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.*

19.4 I/O Signals

The following paragraphs describe the signals shown in [Figure 19-1](#).

19.4.1 OSCXCLK

OSCXCLK is the crystal oscillator output signal. OSCXCLK frequency is equal to the crystal frequency.

19.4.2 STOP Instruction

The STOP instruction clears the COP prescaler.

19.4.3 COPCTL Write

Writing any value to the COP control register (COPCTL) ([see 19.5 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.

19.4.4 Power-On Reset

The power-on reset (POR) circuit clears the COP prescaler 4096 OSCXCLK cycles after power-up.

19.4.5 Internal Reset

An internal reset clears the COP prescaler and the COP counter.

19.4.6 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.

19.4.7 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the configuration register 1 (see [Figure 19-2](#)).

19.4.8 COPRS (COP Rate Select)

The COPRS signal reflects the state of the COP rate select bit (COPRS) in the configuration register 1 (see [Figure 19-2](#)).

Address: \$001F

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-----------------|---|---|---|-------|-------|------|-------|
| Read: | 0 | 0 | 0 | 0 | SSREC | COPRS | STOP | COPD |
| Write: | [Unimplemented] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

[Unimplemented] = Unimplemented

Figure 19-2. Configuration Register 1 (CONFIG1)

COPRS — COP Rate Select Bit

COPRS selects the COP timeout period. Reset clears COPRS.

1 = COP timeout period = $2^{13} - 2^4$ OSCXCLK cycles

0 = COP timeout period = $2^{18} - 2^4$ OSCXCLK cycles

COPD — COP Disable Bit

COPD disables the COP module.

1 = COP module disabled

0 = COP module enabled

19.5 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 19-3. COP Control Register (COPCTL)

19.6 Interrupts

The COP does not generate CPU interrupt requests.

19.7 Monitor Mode

When monitor mode is entered with V_{TST} on the \overline{IRQ} pin, the COP is disabled as long as V_{TST} remains on the \overline{IRQ} pin or the \overline{RST} pin. When monitor mode is entered by having blank reset vectors and not having V_{TST} on the \overline{IRQ} pin, the COP is automatically disabled until a POR occurs.

19.8 Low-Power Modes

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

19.8.1 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.

19.8.2 Stop Mode

Stop mode turns off the OSCXCLK 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.

19.9 COP Module During Break Mode

The COP is disabled during a break interrupt when V_{TST} is present on the \overline{RST} pin.

Section 20. Break Module (BRK)

20.1 Contents

| | | |
|--------|---|-----|
| 20.2 | Introduction | 263 |
| 20.3 | Features | 264 |
| 20.4 | Functional Description | 264 |
| 20.4.1 | Flag Protection During Break Interrupts | 266 |
| 20.4.2 | CPU During Break Interrupts | 266 |
| 20.4.3 | TIM During Break Interrupts | 266 |
| 20.4.4 | COP During Break Interrupts | 266 |
| 20.5 | Low-Power Modes | 266 |
| 20.5.1 | Wait Mode | 266 |
| 20.5.2 | Stop Mode | 267 |
| 20.6 | Break Module Registers | 267 |
| 20.6.1 | Break Status and Control Register | 267 |
| 20.6.2 | Break Address Registers | 268 |
| 20.6.3 | SIM Break Status Register | 268 |
| 20.6.4 | SIM Break Flag Control Register | 270 |

20.2 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.

20.3 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

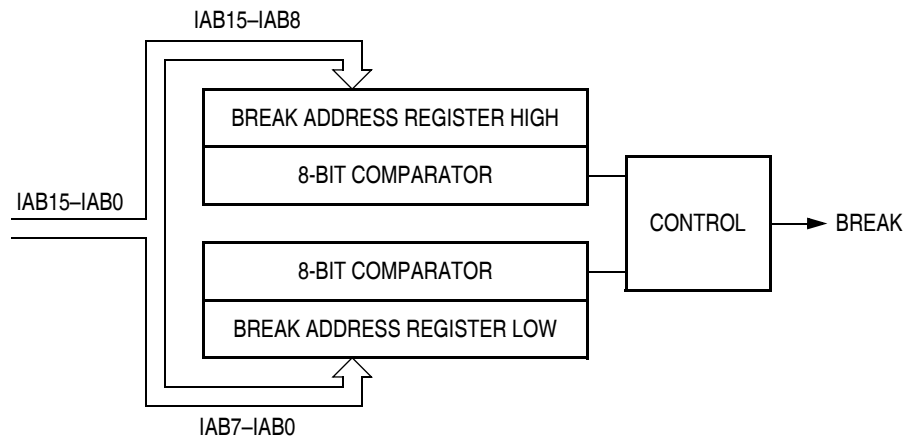
20.4 Functional Description

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 20-1](#) shows the structure of the break module.


Figure 20-1. Break Module Block Diagram
Table 20-1. Break Module I/O Register Summary

| 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 | SBSW | R | |
| | | Write: | | | | | | Note | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| \$FE03 | SIM Break Flag Control Register (SBFCR) | Read: | BCFE | 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

20.4.1 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.

20.4.2 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.

20.4.3 TIM During Break Interrupts

A break interrupt stops the timer counters.

20.4.4 COP During Break Interrupts

The COP is disabled during a break interrupt when V_{TST} is present on the \overline{RST} pin.

20.5 Low-Power Modes

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

20.5.1 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 7. System Integration Module \(SIM\)](#)). Clear the SBSW bit by writing logic 0 to it.

20.5.2 Stop Mode

A break interrupt causes exit from stop mode and sets the SBSW bit in the break status register.

20.6 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)

20.6.1 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 20-2. 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

20.6.2 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.

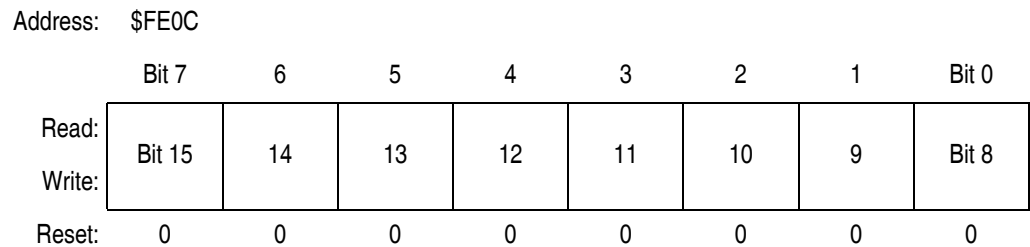


Figure 20-3. Break Address Register High (BRKH)

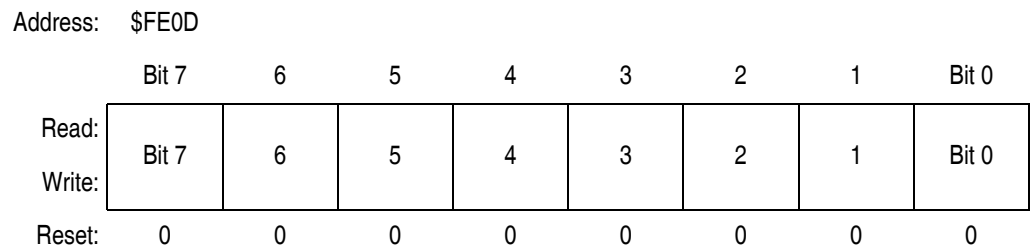


Figure 20-4. Break Address Register Low (BRKL)

20.6.3 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.

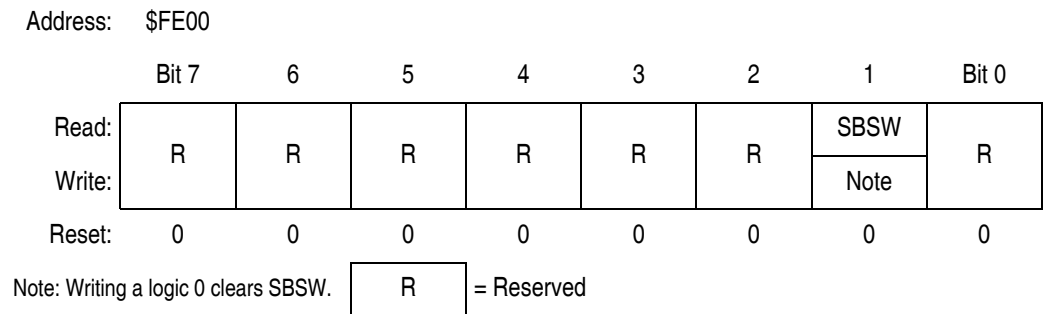


Figure 20-5. SIM Break Status Register (SBSR)

SBSW — SIM Break Stop/Wait Bit

This status bit is useful in applications requiring a return to wait or stop mode after exiting from a break interrupt. 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 one 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
    
```

20.6.4 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

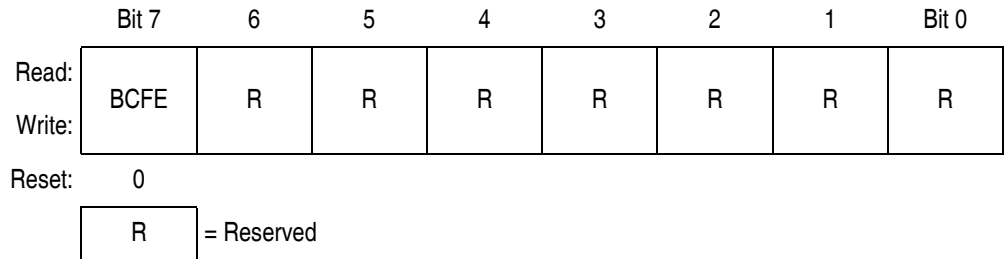


Figure 20-6. SIM Break Flag Control Register (SBFCR)

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 21. Electrical Specifications

21.1 Contents

| | | |
|---------|--|-----|
| 21.2 | Introduction | 271 |
| 21.3 | Absolute Maximum Ratings | 272 |
| 21.4 | Functional Operating Range. | 273 |
| 21.5 | Thermal Characteristics | 273 |
| 21.6 | DC Electrical Characteristics | 274 |
| 21.7 | Control Timing | 275 |
| 21.8 | Oscillator Characteristics | 275 |
| 21.9 | ADC Characteristics | 276 |
| 21.10 | USB DC Electrical Characteristics | 277 |
| 21.11 | USB Low Speed Source Electrical Characteristics. | 278 |
| 21.12 | Timer Interface Module Characteristics | 278 |
| 21.13 | DDC12AB/MMIIC Timing | 279 |
| 21.13.1 | DDC12AB/MMIIC Interface Input Signal Timing | 279 |
| 21.13.2 | DDC12AB/MMIIC Interface Output Signal Timing | 279 |
| 21.14 | Sync Processor Timing. | 280 |
| 21.15 | Memory Characteristics | 281 |

21.2 Introduction

This section contains electrical and timing specifications.

21.3 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 [21.6 DC Electrical Characteristics](#) for guaranteed operating conditions.*

| Characteristic | Symbol | Value | Unit |
|--|------------|------------------------------|------|
| Supply Voltage | V_{DD} | -0.3 to +5.5 | V |
| Input Voltage | V_{IN} | $V_{SS}-0.3$ to $V_{DD}+0.3$ | V |
| Maximum Current Per Pin Excluding V_{DD} and V_{SS} | I | ±25 | mA |
| Storage Temperature | T_{STG} | -55 to +150 | °C |
| Maximum Current Out of V_{SS} | I_{MVSS} | 100 | mA |
| Maximum Current Into V_{DD} | I_{MVDD} | 100 | mA |

NOTE:

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 $V_{SS} \leq (V_{IN} \text{ or } V_{OUT}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD} .)*

21.4 Functional Operating Range

| Characteristic | Symbol | Value | Unit |
|-----------------------------|----------|------------|------|
| Operating Temperature Range | T_A | 0 to 85 | °C |
| Operating Voltage Range | V_{DD} | 4.5 to 5.5 | V |

21.5 Thermal Characteristics

| Characteristic | Symbol | Value | Unit |
|---|---------------|--|------|
| Thermal Resistance QFP (44 Pins) SDIP (42 Pins) | θ_{JA} | 95 60 | °C/W |
| I/O Pin Power Dissipation | $P_{I/O}$ | User Determined | W |
| Power Dissipation ⁽¹⁾ | P_D | $P_D = (I_{DD} \times V_{DD}) + P_{I/O} =$ $K/(T_J + 273 \text{ °C})$ | W |
| Constant ⁽²⁾ | K | $P_D \times (T_A + 273 \text{ °C})$ $+ P_D^2 \times \theta_{JA}$ | W/°C |
| Average Junction Temperature | T_J | $T_A + (P_D \times \theta_{JA})$ | °C |
| Maximum Junction Temperature | T_{JM} | 100 | °C |

NOTES:

1. Power dissipation is a function of temperature.
2. K is a 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 .

21.6 DC Electrical Characteristics

| Characteristic | Symbol | Min | Typ ⁽²⁾ | Max | Unit |
|---|-----------------------|--|--------------------|--|----------------------|
| Output High Voltage ($I_{LOAD} = -2.0\text{mA}$) All ports (except PTD0 and PTD1) | V_{OH} | $V_{DD} - 0.8$ | — | — | V |
| Output Low Voltage ($I_{LOAD} = 1.6\text{mA}$) All ports (except PTD0 and PTD1) | V_{OL} | — | — | 0.4 | V |
| Input High Voltage All ports (except PTD0 and PTD1), \overline{IRQ} , \overline{RST} OSC1, PTD0, PTD1 VSYNC, HSYNC | V_{IH} | $0.7 \times V_{DD}$ $0.7 \times V_{REG}$ 2.0 | — — — | V_{DD} V_{REG} V_{DD} | V V V |
| Input Low Voltage All ports (except PTD0 and PTD1), \overline{IRQ} , \overline{RST} OSC1, PTD0, PTD1 VSYNC, HSYNC | V_{IL} | V_{SS} V_{SS} V_{SS} | — — — | $0.2 \times V_{DD}$ $0.2 \times V_{REG}$ 0.8 | V V V |
| V_{DD} Supply Current Run, USB active Run, USB suspended ⁽³⁾ Wait ⁽⁴⁾ Stop ⁽⁵⁾ 0°C to 85°C | I_{DD} | — — — — | 10 8 4 2 | 15 12 8 5 | mA mA mA mA |
| 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 ReArm 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} | $V_{DD} + 2.5$ | — | 9 | V |
| Pull-up Resistor \overline{RST} , \overline{IRQ} | R_{PU} | 20 | 45 | 65 | k Ω |
| Low-Voltage Inhibit, trip falling voltage | V_{TRIPF} | 3.4 | 3.6 | 3.8 | V |
| Low-Voltage Inhibit, trip rising voltage | V_{TRIPR} | 3.6 | 3.8 | 4.0 | V |
| Low-Voltage Inhibit Reset/Recover Hysteresis | V_{HYS} | — | 200 | — | mV |

NOTES:

- $V_{DD} = 5.0\text{Vdc} \pm 10\%$, $V_{SS} = 0\text{Vdc}$, $T_A = T_L$ to T_H , unless otherwise noted.
- Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
- 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. $C_L = 15\text{pF}$ on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD} . Measured with all modules enabled.
- Wait I_{DD} measured using external square wave clock source ($f_{OSCCLK} = 24\text{MHz}$); all inputs 0.2 V from rail; no dc loads; less than 100 pF on all outputs. $C_L = 15\text{pF}$ on OSC2; USB in suspend mode, 15 k $\Omega \pm 5\%$ termination resistors on D+ and D- pins; all ports configured as inputs; OSC2 capacitance linearly affects wait I_{DD} .
- STOP I_{DD} measured with USB in suspend mode, OSC1 grounded, 1.5 k $\Omega \pm 1\%$ pull-up resistor on D+ pin and 15 k $\Omega \pm 1\%$ pull-down resistors on D+ and D- pins, no port pins sourcing current.
- Maximum is highest voltage that POR is guaranteed.
- 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.

21.7 Control Timing

| Characteristic | Symbol | Min | Max | Unit |
|---|-----------|-----|-----|------|
| Internal Operating Frequency ⁽²⁾ | f_{OP} | — | 6 | MHz |
| RST Input Pulse Width Low ⁽³⁾ | t_{IRL} | 50 | — | ns |

NOTES:

- $V_{DD} = 4.5$ to 5.5 Vdc, $V_{SS} = 0$ Vdc; timing shown with respect to 20% V_{DD} and 70% V_{DD} , unless otherwise noted.
- Some modules may require a minimum frequency greater than dc for proper operation; see appropriate table for this information.
- Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.

21.8 Oscillator Characteristics

| Characteristic | Symbol | Min | Typ | Max | Unit |
|--|--------------|-----|-----|-----|-----------|
| Crystal Frequency ⁽¹⁾ | f_{OSCCLK} | — | 24 | — | MHz |
| External Clock Reference Frequency ^{(1), (2)} | f_{OSCCLK} | dc | — | 24 | MHz |
| Crystal Load Capacitance ⁽³⁾ | C_L | — | 30 | — | pF |
| Crystal Fixed Capacitance ⁽³⁾ | C_1 | — | 15 | — | pF |
| Crystal Tuning Capacitance ⁽³⁾ | C_2 | — | 15 | — | pF |
| Feedback Bias Resistor | R_B | — | 10 | — | $M\Omega$ |
| Series Resistor ^{(3), (4)} | R_S | — | — | — | |

NOTES:

- The sync processor module is designed to function at $f_{OSCCLK} = 24$ MHz. The values given here are oscillator specifications.
- No more than 10% duty cycle deviation from 50%
- Quoted values are for reference only. Actual values depend on application and crystal performance. Please consult crystal vendor data sheet
- Not Required for high frequency crystals

21.9 ADC Characteristics

| Characteristic ⁽¹⁾ | Symbol | Min | Max | Unit | Comments |
|--|------------|------------------------|------------------------|------------------|---|
| Supply voltage | V_{DDAD} | 4.5 (V_{DD} min) | 5.5 (V_{DD} max) | V | |
| Input voltages | V_{ADIN} | 0 | $\frac{2}{3} V_{DD}$ | V | |
| Resolution | B_{AD} | 8 | 8 | Bits | |
| Absolute accuracy ($V_{SS} = 0$ V, $V_{DD} = 5$ V \pm 10%) | A_{AD} | — | ± 2 | LSB | Includes quantization |
| ADC internal clock | f_{ADIC} | 0.375 | 6 | MHz | $t_{AIC} = 1/f_{ADIC}$, tested only at 1.5 MHz |
| Conversion range | R_{AD} | V_{SS} | $\frac{2}{3} V_{DD}$ | V | |
| Power-up time | t_{ADPU} | 16 | | t_{AIC} cycles | |
| Conversion time | t_{ADC} | 12 | 13 | t_{AIC} cycles | |
| Sample time ⁽²⁾ | t_{ADS} | 4 | — | t_{AIC} cycles | |
| Zero input reading ⁽³⁾ | Z_{ADI} | 00 | 02 | Hex | |
| Full-scale reading ⁽³⁾ | F_{ADI} | FD | FF | Hex | |
| Input capacitance | C_{ADI} | — | 8 | pF | Not tested |
| Input leakage ⁽⁴⁾ Port C | — | — | ± 1 | μ A | |

NOTES:

- $V_{DD} = 5.0$ Vdc \pm 10%, $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H , unless otherwise noted.
- Source impedances greater than 10 k Ω adversely affect internal RC charging time during input sampling.
- Zero-input/full-scale reading requires sufficient decoupling measures for accurate conversions.
- The external system error caused by input leakage current is approximately equal to the product of R source and input current.

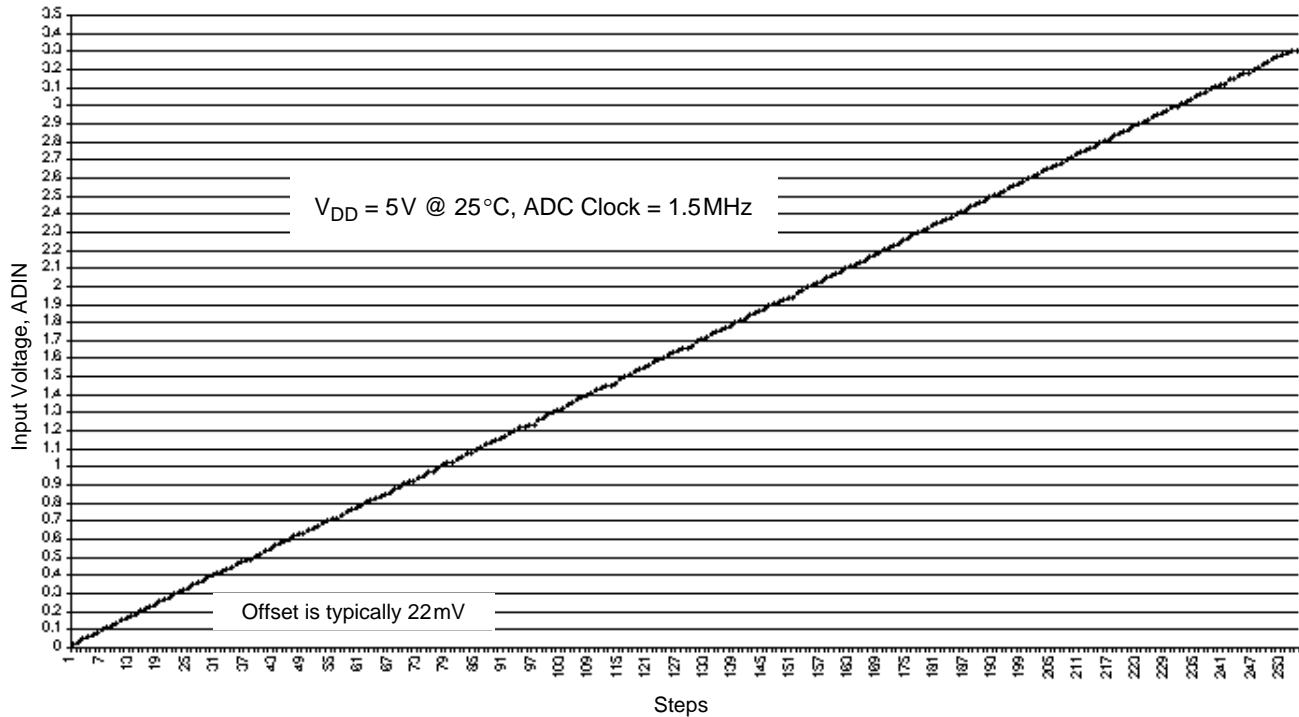


Figure 21-1. ADC Input Voltage vs. Step Readings

21.10 USB DC Electrical Characteristics

| Characteristic | Symbol | Conditions | Min | Typ | Max | Unit |
|--|--------------|-------------------------|-----|-----|-----|---------|
| Hi-Z State Data Line Leakage | I_{LO} | $0V < V_{IN} < 3.3V$ | -10 | | +10 | μA |
| Differential Input Sensitivity | V_{DI} | $ (D+) - (D-) $ | 0.2 | | | V |
| Differential Common Mode Range | V_{CM} | Includes V_{DI} range | 0.8 | | 2.5 | V |
| Single Ended Receiver Threshold | V_{SE} | | 0.8 | | 2.0 | V |
| Static Output Low | V_{OL} | R_L of 1.5k to 3.6V | | | 0.3 | V |
| Static Output High | V_{OH} | R_L of 15k to GND | 2.8 | | 3.6 | V |
| Regulator Supply Voltage ^{(2), (3)} | V_{REGOUT} | $I_L = 4$ mA | 3.0 | 3.3 | 3.6 | V |

NOTES:

- $V_{DD} = 5.0$ Vdc $\pm 10\%$, $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H , unless otherwise noted.
- Transceiver pullup resistor of $1.5K\Omega \pm 5\%$ between V_{DD3} and D- and $15K\Omega \pm 5\%$ to ground termination resistors on D+ and D-.
- No external current draw besides the USB required external resistors should be connected to the V_{DD3} pin.

21.11 USB Low Speed Source Electrical Characteristics

| Characteristic | Symbol | Conditions (Notes 1,2,3) | Min | Typ | Max | Unit |
|---|----------------------------|---|-----------------|----------------|-----------------|---------------|
| Transition time: Rise Time | T_R | Notes 4, 5, 8 $C_L=50\text{pF}$ | 75 | — | — | ns |
| Fall Time | T_F | $C_L=350\text{pF}$ $C_L=50\text{pF}$ $C_L=350\text{pF}$ | 75 | — | 300 | |
| Rise/Fall Time Matching | T_{RFM} | T_R/T_F | 80 | — | 120 | % |
| Output Signal Crossover Voltage | V_{CRS} | | 1.3 | — | 2.0 | V |
| Low Speed Data Rate | T_{DRATE} | $1.5\text{Mbs} \pm 1.5\%$ | 1.4775 676.8 | 1.500 666.0 | 1.5225 656.8 | Mbs ns |
| Source Differential Driver Jitter To Next Transition For Paired Transitions | T_{UDJ1} T_{UDJ2} | $C_L=350\text{pF}$ Notes 6, 7 | -25 -10 | — — | 25 10 | ns ns |
| Receiver Data Jitter Tolerance To Next Transition For Paired Transitions | T_{DJR1} T_{DJR2} | $C_L=350\text{pF}$ Note 7 | -75 -45 | — — | 75 45 | ns ns |
| Source EOP Width | TEOPT | Note 7 | 1.25 | — | 1.50 | μs |
| Differential to EOP Transition Skew | TDEOP | Note 7 | -40 | — | 100 | ns |
| Receiver EOP Width Must Reject as EOP Must Accept | T_{EOPR1} T_{EOPR2} | Note 7 | 330 675 | — — | — — | ns ns |

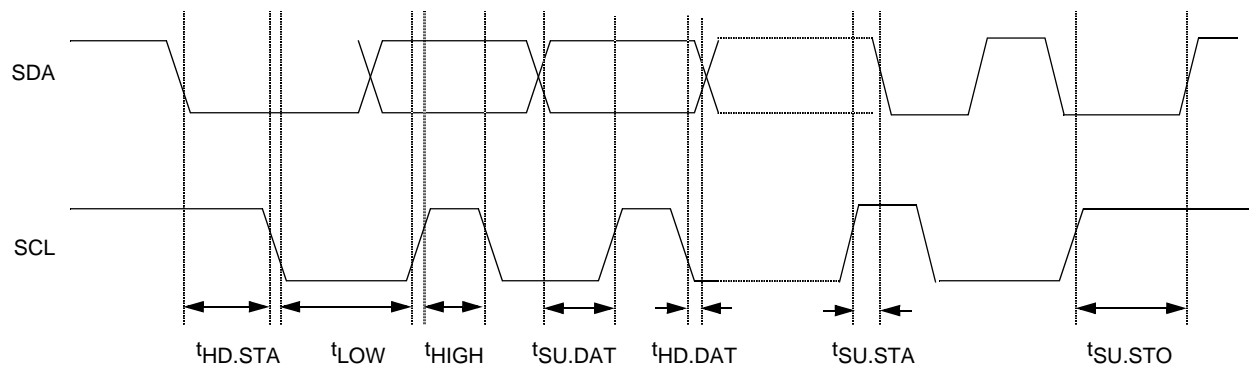
NOTES:

1. All voltages measured from local ground, unless otherwise specified.
2. All timings use a capacitive load of 50pF, unless otherwise specified.
3. Low speed timings have a 1.5k Ω pull-up to 2.8V on the D- data line.
4. Measured from 10% to 90% of the data signal.
5. The rising and falling edges should be smoothly transitioning (monotonic).
6. Timing differences between the differential data signals.
7. Measured at crossover point of differential data signals.
8. Capacitive loading includes 50pF of tester capacitance.

21.12 Timer Interface Module Characteristics

| Characteristic | Symbol | Min | Max | Unit |
|---------------------------|--------------------|------------------|-----|------|
| Input Capture Pulse Width | t_{TIH}, t_{TIL} | 125 | — | ns |
| Input Clock Pulse Width | t_{TCH}, t_{TCL} | $(1/f_{OP}) + 5$ | — | ns |

21.13 DDC12AB/MMIIC Timing



21.13.1 DDC12AB/MMIIC Interface Input Signal Timing

| Characteristic | Symbol | Min | Max | Unit |
|--|--------------|-----|-----|-----------|
| START condition hold time | $t_{HD.STA}$ | 2 | — | t_{CYC} |
| Clock low period | t_{LOW} | 4 | — | t_{CYC} |
| Clock high period | t_{HIGH} | 4 | — | t_{CYC} |
| Data set-up time | $t_{SU.DAT}$ | 250 | — | ns |
| Data hold time | $t_{HD.DAT}$ | 0 | — | ns |
| START condition set-up time (for repeated START condition only) | $t_{SU.STA}$ | 2 | — | t_{CYC} |
| STOP condition set-up time | $t_{SU.STO}$ | 2 | — | t_{CYC} |

NOTES:

1. $V_{DD} = 5.0 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$; timing shown with respect to 20% V_{DD} and 70% V_{DD} , unless otherwise noted.

21.13.2 DDC12AB/MMIIC Interface Output Signal Timing

| Characteristic | Symbol | Min | Max | Unit |
|----------------------------------|--------------|-----------|-----|---------------|
| SDA/SCL rise time ⁽²⁾ | t_R | — | 1 | μs |
| SDA/SCL fall time | t_F | — | 300 | ns |
| Data set-up time | $t_{SU.DAT}$ | t_{LOW} | — | ns |
| Data hold time | $t_{HD.DAT}$ | 0 | — | ns |

NOTES:

1. $V_{DD} = 5.0 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$; timing shown with respect to 20% V_{DD} and 70% V_{DD} , unless otherwise noted.
2. With 200pF loading on the SDA/SCL pins.

21.14 Sync Processor Timing

| Characteristic | Symbol | Min | Max | Unit |
|-------------------------------------|-------------|-----|------|---------|
| VSYNC input sync pulse | $t_{VI.SP}$ | 8 | 2048 | μs |
| HSYNC input sync pulse | $t_{HI.SP}$ | 0.1 | 6 | μs |
| VSYNC to VSYNCO delay (8pF loading) | t_{VVd} | 30 | 40 | μs |
| HSYNC to HSYNCO delay (8pF loading) | t_{HHd} | 30 | 40 | μs |

NOTES:

1. $V_{DD} = 5.0 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$; timing shown with respect to 20% V_{DD} and 70% V_{DD} , unless otherwise noted.

21.15 Memory Characteristics

| Characteristic | Symbol | Min | Max | Unit |
|--|--------------------|-----|------|---------|
| RAM data retention voltage | V_{RDR} | 2 | — | V |
| FLASH program bus clock frequency | — | 1 | — | MHz |
| FLASH read bus clock frequency | $f_{Read}^{(1)}$ | 32k | 8.4M | Hz |
| FLASH page erase time | $t_{Erase}^{(2)}$ | 2 | — | ms |
| FLASH mass erase time | $t_{MErase}^{(3)}$ | 4 | — | ms |
| FLASH PGM/ERASE to HVEN set up time | t_{nvs} | 5 | — | μs |
| FLASH high-voltage hold time | t_{nvh} | 5 | — | μs |
| FLASH high-voltage hold time (mass erase) | t_{nvhl} | 100 | — | μs |
| FLASH program hold time | t_{pgs} | 10 | — | μs |
| FLASH program time | t_{PROG} | 20 | 40 | μs |
| FLASH return to read time | $t_{rcv}^{(4)}$ | 1 | — | μs |
| FLASH cumulative program hv period | $t_{HV}^{(5)}$ | — | 4 | ms |
| FLASH row erase endurance ⁽⁶⁾ | — | 10k | — | Cycles |
| FLASH row program endurance ⁽⁷⁾ | — | 10k | — | Cycles |
| FLASH data retention time ⁽⁸⁾ | — | 10 | — | Years |

Notes:

- f_{Read} is defined as the frequency range for which the FLASH memory can be read.
- 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.
- If the mass erase time is longer than t_{MErase} (Min), there is no erase-disturb, but it reduces the endurance of the FLASH memory.
- t_{rcv} 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.
- t_{HV} is defined as the cumulative high voltage programming time to the same row before next erase.
 t_{HV} must satisfy this condition: $t_{nvs} + t_{nvh} + t_{pgs} + (t_{PROG} \times 64) \leq t_{HV} \text{ max.}$
- The minimum row endurance value specifies each row of the FLASH memory is guaranteed to work for at least this many erase / program cycles.
- The minimum row endurance value specifies each row of the FLASH memory is guaranteed to work for at least this many erase / program cycles.
- The FLASH is guaranteed to retain data over the entire operating temperature range for at least the minimum time specified.



Section 22. Mechanical Specifications

22.1 Contents

22.2 Introduction283

22.4 42-Pin Shrink Dual in-Line Package (SDIP)284

22.5 44-Pin Plastic Quad Flat Pack (QFP)285

22.2 Introduction

This section gives the dimensions for:

- 42-pin shrink dual in-line package (case #858)
- 44-pin plastic quad flat pack (case #824A)

22.3 42-Pin Shrink Dual in-Line Package (SDIP)

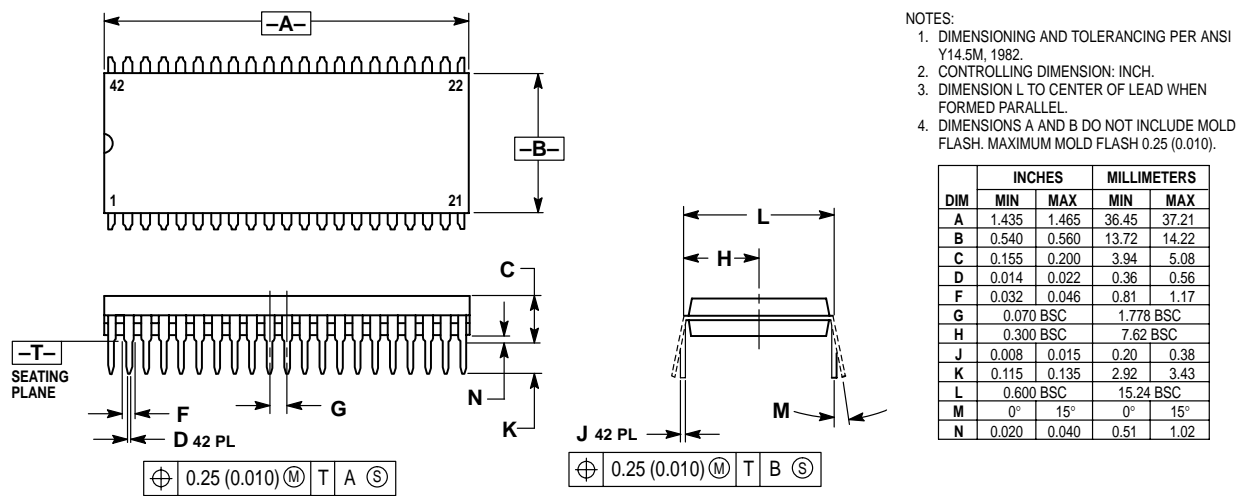


Figure 22-1. 42-Pin SDIP (Case #858)

22.4 44-Pin Plastic Quad Flat Pack (QFP)

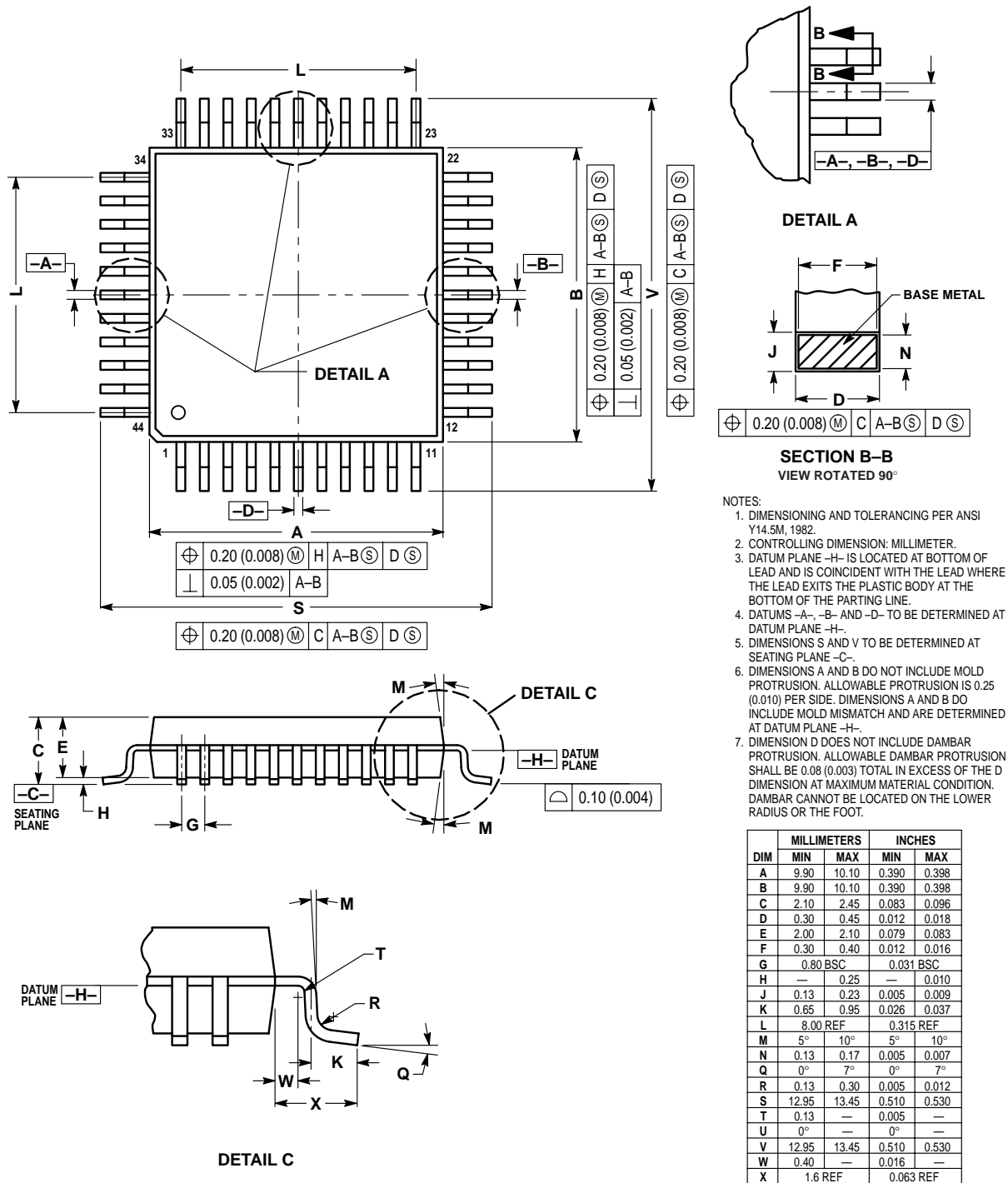


Figure 22-2. 44-Pin QFP (Case #824A)

Section 23. Ordering Information

23.1 Contents

| | | |
|------|------------------------|-----|
| 23.2 | Introduction | 285 |
| 23.3 | MC Order Numbers | 285 |

23.2 Introduction

This section contains ordering numbers for the MC68HC908BD48.

23.3 MC Order Numbers

Table 23-1. MC Order Numbers

| MC Order Number | Operating Temperature Range | Package |
|------------------|-----------------------------|-------------|
| MC68HC908BD48IB | 0 °C to +85 °C | 42-pin SDIP |
| MC68HC908BD48IFB | 0 °C to +85 °C | 44-pin QFP |



How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

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

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Rev. 2.1
MC68HC908BD48/D
August 1, 2005

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.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. The ARM POWERED logo is a registered trademark of ARM Limited. ARM7TDMI-S is a trademark of ARM Limited. Java and all other Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. The Bluetooth trademarks are owned by their proprietor and used by Freescale Semiconductor, Inc. under license.

© Freescale Semiconductor, Inc. 2005. All rights reserved.