



**MOTOROLA**  
intelligence everywhere™

*digital dna*™

Freescale Semiconductor, Inc.

*M68HC08*  
*Microcontrollers*

*MC68HC908QF4*

*Data Sheet*

MC68HC908QF4  
Rev. 1.0  
6/2004

[MOTOROLA.COM/SEMICONDUCTORS](http://MOTOROLA.COM/SEMICONDUCTORS)



## **MC68HC908QF4**

### **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://motorola.com/semiconductors/>

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.

Revision History

Date	Revision Level	Description	Page Number(s)
October, 2003	N/A	Initial release	N/A
June, 2004	1.0	Removed references to MC68HC908QF3, MC68HC908QF2, and MC68HC908QF1	Throughout
		<a href="#">17.4 Thermal Characteristics</a> — Updated 32-pin TQFP value	176
		<a href="#">18.2 MC Order Numbers</a> — Updated table entries for MC order numbers	193

## List of Sections

Section 1. General Description .....	15
Section 2. Memory .....	21
Section 3. Analog-to-Digital Converter (ADC) .....	37
Section 4. Auto Wakeup Module (AWU) .....	45
Section 5. Configuration Register (CONFIG) .....	51
Section 6. Computer Operating Properly (COP) .....	55
Section 7. Central Processor Unit (CPU) .....	59
Section 8. External Interrupt (IRQ) .....	73
Section 9. Keyboard Interrupt Module (KBI) .....	79
Section 10. Low-Voltage Inhibit (LVI) .....	87
Section 11. Oscillator Module (OSC).....	91
Section 12. PLL Tuned UHF Transmitter Module.....	101
Section 13. Input/Output (I/O) Ports .....	111
Section 14. System Integration Module (SIM) .....	119
Section 15. Timer Interface Module (TIM).....	137
Section 16. Development Support.....	155
Section 17. Electrical Specifications .....	175
Section 18. Ordering Information and Mechanical Specifications .....	193



**Table of Contents**

**Section 1. General Description**

1.1	Introduction . . . . .	15
1.2	Features . . . . .	15
1.3	MCU Block Diagram . . . . .	16
1.4	Pin Assignments . . . . .	17
1.5	Pin Functions . . . . .	18

**Section 2. Memory**

2.1	Introduction . . . . .	21
2.2	Unimplemented Memory Locations . . . . .	21
2.3	Reserved Memory Locations . . . . .	21
2.4	Input/Output (I/O) Section . . . . .	23
2.5	Random-Access Memory (RAM) . . . . .	29
2.6	FLASH Memory (FLASH) . . . . .	30
2.6.1	FLASH Control Register . . . . .	30
2.6.2	FLASH Page Erase Operation . . . . .	31
2.6.3	FLASH Mass Erase Operation . . . . .	32
2.6.4	FLASH Program Operation . . . . .	32
2.6.5	FLASH Protection . . . . .	33
2.6.6	FLASH Block Protect Register . . . . .	35
2.6.7	Wait Mode . . . . .	36
2.6.8	Stop Mode . . . . .	36

**Section 3. Analog-to-Digital Converter (ADC)**

3.1	Introduction . . . . .	37
3.2	Features . . . . .	37
3.3	Functional Description . . . . .	39
3.3.1	ADC Port I/O Pins . . . . .	40
3.3.2	Voltage Conversion . . . . .	40
3.3.3	Conversion Time . . . . .	40
3.3.4	Continuous Conversion . . . . .	40
3.3.5	Accuracy and Precision . . . . .	40
3.4	Interrupts . . . . .	41

3.5 Low-Power Modes ..... 41

3.5.1 Wait Mode ..... 41

3.5.2 Stop Mode ..... 41

3.6 Input/Output Signals ..... 41

3.7 Input/Output Registers ..... 41

3.7.1 ADC Status and Control Register ..... 42

3.7.2 ADC Data Register ..... 43

3.7.3 ADC Input Clock Register ..... 44

**Section 4. Auto Wakeup Module (AWU)**

4.1 Introduction ..... 45

4.2 Features ..... 45

4.3 Functional Description ..... 46

4.4 Wait Mode ..... 47

4.5 Stop Mode ..... 47

4.6 Input/Output Registers ..... 47

4.6.1 Port A I/O Register ..... 48

4.6.2 Keyboard Status and Control Register ..... 48

4.6.3 Keyboard Interrupt Enable Register ..... 49

**Section 5. Configuration Register (CONFIG)**

5.1 Introduction ..... 51

5.2 Functional Description ..... 51

**Section 6. Computer Operating Properly (COP)**

6.1 Introduction ..... 55

6.2 Functional Description ..... 55

6.3 I/O Signals ..... 56

6.3.1 BUSCLKX4 ..... 56

6.3.2 STOP Instruction ..... 56

6.3.3 COPCTL Write ..... 56

6.3.4 Power-On Reset ..... 56

6.3.5 Internal Reset ..... 56

6.3.6 COPD (COP Disable) ..... 57

6.3.7 COPRS (COP Rate Select) ..... 57

6.4 COP Control Register ..... 57

6.5 Interrupts ..... 57

6.6 Monitor Mode ..... 57



6.7 Low-Power Modes ..... 57  
 6.7.1 Wait Mode ..... 57  
 6.7.2 Stop Mode ..... 58  
 6.8 COP Module During Break Mode ..... 58

**Section 7. Central Processor Unit (CPU)**

7.1 Introduction ..... 59  
 7.2 Features ..... 59  
 7.3 CPU Registers ..... 60  
 7.3.1 Accumulator ..... 60  
 7.3.2 Index Register ..... 61  
 7.3.3 Stack Pointer ..... 61  
 7.3.4 Program Counter ..... 62  
 7.3.5 Condition Code Register ..... 62  
 7.4 Arithmetic/Logic Unit (ALU) ..... 64  
 7.5 Low-Power Modes ..... 64  
 7.5.1 Wait Mode ..... 64  
 7.5.2 Stop Mode ..... 64  
 7.6 CPU During Break Interrupts ..... 64  
 7.7 Instruction Set Summary ..... 65  
 7.8 Opcode Map ..... 71

**Section 8. External Interrupt (IRQ)**

8.1 Introduction ..... 73  
 8.2 Features ..... 73  
 8.3 Functional Description ..... 73  
 8.4  $\overline{\text{IRQ}}$  Pin ..... 76  
 8.5 IRQ Module During Break Interrupts ..... 76  
 8.6 IRQ Status and Control Register ..... 77

**Section 9. Keyboard Interrupt Module (KBI)**

9.1 Introduction ..... 79  
 9.2 Features ..... 79  
 9.3 Functional Description ..... 81  
 9.3.1 Keyboard Operation ..... 81  
 9.3.2 Keyboard Initialization ..... 83  
 9.4 Wait Mode ..... 83  
 9.5 Stop Mode ..... 83

9.6	Keyboard Module During Break Interrupts . . . . .	83
9.7	Input/Output Registers . . . . .	84
9.7.1	Keyboard Status and Control Register . . . . .	84
9.7.2	Keyboard Interrupt Enable Register . . . . .	85

**Section 10. Low-Voltage Inhibit (LVI)**

10.1	Introduction . . . . .	87
10.2	Features . . . . .	87
10.3	Functional Description . . . . .	87
10.3.1	Polled LVI Operation . . . . .	88
10.3.2	Forced Reset Operation . . . . .	88
10.3.3	Voltage Hysteresis Protection . . . . .	88
10.3.4	LVI Trip Selection . . . . .	89
10.4	LVI Status Register . . . . .	89
10.5	LVI Interrupts . . . . .	90
10.6	Low-Power Modes . . . . .	90
10.6.1	Wait Mode . . . . .	90
10.6.2	Stop Mode . . . . .	90

**Section 11. Oscillator Module (OSC)**

11.1	Introduction . . . . .	91
11.2	Features . . . . .	91
11.3	Functional Description . . . . .	91
11.3.1	Internal Oscillator . . . . .	91
11.3.1.1	Internal Oscillator Trimming . . . . .	93
11.3.1.2	Internal to External Clock Switching . . . . .	93
11.3.2	External Oscillator . . . . .	94
11.3.3	XTAL Oscillator . . . . .	94
11.3.4	RC Oscillator . . . . .	95
11.4	Oscillator Module Signals . . . . .	95
11.4.1	Crystal Amplifier Input Pin (OSC1) . . . . .	95
11.4.2	Crystal Amplifier Output Pin (OSC2/PTA4/BUSCLKX4) . . . . .	96
11.4.3	Oscillator Enable Signal (SIMOSCEN) . . . . .	96
11.4.4	XTAL Oscillator Clock (XTALCLK) . . . . .	96
11.4.5	RC Oscillator Clock (RCCLK) . . . . .	96
11.4.6	Internal Oscillator Clock (INTCLK) . . . . .	97
11.4.7	Oscillator Out 2 (BUSCLKX4) . . . . .	97
11.4.8	Oscillator Out (BUSCLKX2) . . . . .	97
11.5	Low Power Modes . . . . .	97
11.5.1	Wait Mode . . . . .	97
11.5.2	Stop Mode . . . . .	97

11.6	Oscillator During Break Mode	97
11.7	CONFIG2 Options	97
11.8	Input/Output (I/O) Registers	98
11.8.1	Oscillator Status Register	98
11.8.2	Oscillator Trim Register (OSCTRIM)	99

### Section 12. PLL Tuned UHF Transmitter Module

12.1	Introduction	101
12.2	Transmitter Functional Description	103
12.3	Phase-Lock Loop (PLL) and Local Oscillator	103
12.4	RF Output Stage	103
12.5	Modulation	104
12.6	Microcontroller Interfaces	104
12.7	State Machine	105
12.8	Power Management	107
12.9	Data Clock	107
12.10	Application Information	107
12.10.1	Application Schematics in OOK and FSK Modulation	107
12.10.2	Complete Application Schematic	109

### Section 13. Input/Output (I/O) Ports

13.1	Introduction	111
13.2	Port A	112
13.2.1	Port A Data Register	112
13.2.2	Data Direction Register A	113
13.2.3	Port A Input Pullup Enable Register	114
13.3	Port B	115
13.3.1	Port B Data Register	115
13.3.2	Data Direction Register B	115
13.3.3	Port B Input Pullup Enable Register	116

### Section 14. System Integration Module (SIM)

14.1	Introduction	119
14.2	$\overline{RST}$ and $\overline{IRQ}$ Pins Initialization	119
14.3	SIM Bus Clock Control and Generation	121
14.3.1	Bus Timing	122
14.3.2	Clock Start-Up from POR	122
14.3.3	Clocks in Stop Mode and Wait Mode	122

14.4	Reset and System Initialization	122
14.4.1	External Pin Reset	123
14.4.2	Active Resets from Internal Sources	123
14.4.2.1	Power-On Reset	124
14.4.2.2	Computer Operating Properly (COP) Reset	124
14.4.2.3	Illegal Opcode Reset	125
14.4.2.4	Illegal Address Reset	125
14.4.2.5	Low-Voltage Inhibit (LVI) Reset	125
14.5	SIM Counter	126
14.5.1	SIM Counter During Power-On Reset	126
14.5.2	SIM Counter During Stop Mode Recovery	126
14.5.3	SIM Counter and Reset States	126
14.6	Exception Control	126
14.6.1	Interrupts	126
14.6.1.1	Hardware Interrupts	128
14.6.1.2	SWI Instruction	129
14.6.2	Interrupt Status Registers	130
14.6.2.1	Interrupt Status Register 1	130
14.6.2.2	Interrupt Status Register 2	131
14.6.2.3	Interrupt Status Register 3	131
14.6.3	Reset	131
14.6.4	Break Interrupts	131
14.6.5	Status Flag Protection in Break Mode	132
14.7	Low-Power Modes	132
14.7.1	Wait Mode	132
14.7.2	Stop Mode	133
14.8	SIM Registers	135
14.8.1	SIM Reset Status Register	135
14.8.2	Break Flag Control Register	136

**Section 15. Timer Interface Module (TIM)**

15.1	Introduction	137
15.2	Features	137
15.3	Pin Name Conventions	137
15.4	Functional Description	139
15.4.1	TIM Counter Prescaler	141
15.4.2	Input Capture	141
15.4.3	Output Compare	141
15.4.3.1	Unbuffered Output Compare	141
15.4.3.2	Buffered Output Compare	142
15.4.4	Pulse Width Modulation (PWM)	142
15.4.4.1	Unbuffered PWM Signal Generation	143
15.4.4.2	Buffered PWM Signal Generation	144
15.4.4.3	PWM Initialization	144

15.5	Interrupts	145
15.6	Wait Mode	145
15.7	TIM During Break Interrupts	146
15.8	Input/Output Signals	146
15.8.1	TIM Clock Pin (PTA2/TCLK)	146
15.8.2	TIM Channel I/O Pins (PTA0/TCH0 and PTA1/TCH1)	146
15.9	Input/Output Registers	147
15.9.1	TIM Status and Control Register	147
15.9.2	TIM Counter Registers	149
15.9.3	TIM Counter Modulo Registers	149
15.9.4	TIM Channel Status and Control Registers	150
15.9.5	TIM Channel Registers	153

**Section 16. Development Support**

16.1	Introduction	155
16.2	Break Module (BRK)	155
16.2.1	Functional Description	155
16.2.1.1	Flag Protection During Break Interrupts	158
16.2.1.2	TIM During Break Interrupts	158
16.2.1.3	COP During Break Interrupts	158
16.2.2	Break Module Registers	159
16.2.2.1	Break Status and Control Register	159
16.2.2.2	Break Address Registers	160
16.2.2.3	Break Auxiliary Register	160
16.2.2.4	Break Status Register	161
16.2.2.5	Break Flag Control Register	161
16.2.3	Low-Power Modes	162
16.3	Monitor Module (MON)	162
16.3.1	Functional Description	162
16.3.1.1	Normal Monitor Mode	166
16.3.1.2	Forced Monitor Mode	168
16.3.1.3	Monitor Vectors	168
16.3.1.4	Data Format	169
16.3.1.5	Break Signal	169
16.3.1.6	Baud Rate	169
16.3.1.7	Commands	170
16.3.2	Security	173

**Section 17. Electrical Specifications**

17.1	Introduction	175
17.2	Absolute Maximum Ratings	175
17.3	Functional Operating Range	176

17.4	Thermal Characteristics . . . . .	176
17.5	DC Electrical Characteristics . . . . .	177
17.6	Control Timing . . . . .	178
17.7	Typical 3.0-V Output Drive Characteristics . . . . .	179
17.8	Oscillator Characteristics . . . . .	180
17.9	Supply Current Characteristics . . . . .	181
17.10	Analog-to-Digital (ADC) Converter Characteristics . . . . .	183
17.10.1	ADC Electrical Operating Conditions . . . . .	183
17.10.2	ADC Performance Characteristics . . . . .	183
17.11	Timer Interface Module Characteristics . . . . .	184
17.12	Memory Characteristics . . . . .	185
17.13	UHF Transmitter Module . . . . .	186
17.13.1	UHF Module Electrical Characteristics . . . . .	186
17.13.2	UHF Module Output Power Measurement . . . . .	190

**Section 18. Ordering Information  
and Mechanical Specifications**

18.1	Introduction . . . . .	193
18.2	MC Order Numbers . . . . .	193
18.3	32-Pin Plastic Low-Profile Quad Flat Pack (Case No. 873A) . . . . .	194

### Section 1. General Description

#### 1.1 Introduction

The MC68HC908QF4 MCU is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). Optimized for low-power operation and available in a small 32-pin low-profile quad flat pack (LQFP), this MCU is well suited for remote keyless entry (RKE) transmitter designs, tire pressure monitoring (TPM), or other remote sensing and wireless RF data transmission applications.

All MCUs in the M68HC908 Family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

#### 1.2 Features

Features of the MC68HC908QF4 MCU include:

- High-performance M68HC08 architecture
- Fully upward-compatible object code with M6805, M146805, and M68HC05 Families
- Operating voltage range of 2.2 to 3.6 V
- Maximum internal bus frequency of 2 MHz
- Trimmable internal oscillator
  - 4-MHz operating frequency for a 1-MHz bus frequency
  - 8-bit trim capability allows 0.4% accuracy<sup>(1)</sup>
  - $\pm 25$  percent accuracy untrimmed
- Auto wakeup from STOP capability
- 4096 bytes of on-chip FLASH memory
- FLASH program memory security<sup>(2)</sup>
- 128 bytes of on-chip RAM
- 16-bit, 2-channel timer interface module (TIM)
- 4 channel, 8-bit analog-to-digital converter (ADC)

---

1. The oscillator frequency is guaranteed to  $\pm 5\%$  over temperature and voltage range after trimming.

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

- 13 general-purpose input/output (I/O) ports:
  - Six shared with keyboard wakeup function
  - Three shared with the timer module, IRQ
  - Port A pins have 3-mA sink capabilities
- Low-voltage inhibit (LVI) module with selectable trip points:
  - 2.12 V detection forces MCU into reset
  - 2.32 V detection sets indicator flag
- 6-bit keyboard interrupt with wakeup feature (KBI)
- External asynchronous interrupt pin with internal pullup ( $\overline{\text{IRQ}}$ )
- Ultra high frequency (UHF) RF transmitter:
  - Ultra low sleep mode current
  - ASK and FSK modulation selectable
- System protection features:
  - Computer operating properly (COP) reset
  - Low-voltage detection with reset
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- 32-pin plastic LQFP package
- Power saving stop and wait modes
- Master reset pin ( $\overline{\text{RST}}$ ) shared with general-purpose I/O pin

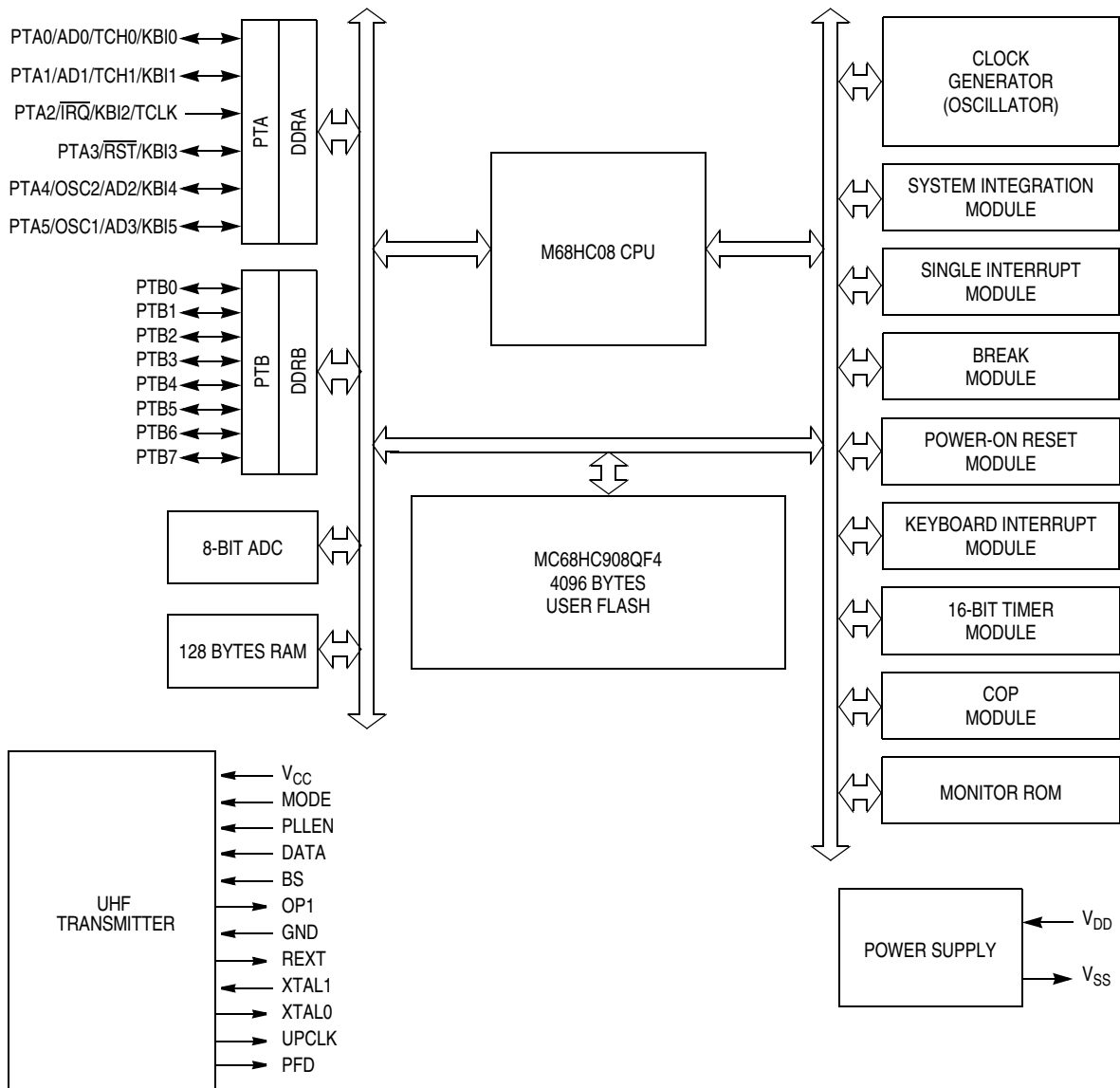
Features of the CPU08 include:

- 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  $8 \times 8$  multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Third party C language support

### 1.3 MCU Block Diagram

**Figure 1-1** shows the structure of the MC68HC908QF4 MCU.





$\overline{RST}$ ,  $\overline{IRQ}$ : Pins have internal (about 30K Ohms) pull up  
 PTA[0:5]: High current sink and source capability  
 PTA[0:5]: Pins have programmable keyboard interrupt and pull up

Figure 1-1. Block Diagram

### 1.4 Pin Assignments

The MC68HC908QF4 is available in a 32-pin plastic low-profile quad flat pack (LQFP). [Figure 1-2](#) shows the pin assignment for this package.

1.5 Pin Functions

Table 1-1 provides a description of the pin functions other than those dedicated to the UHF module which are shown in Table 1-2.

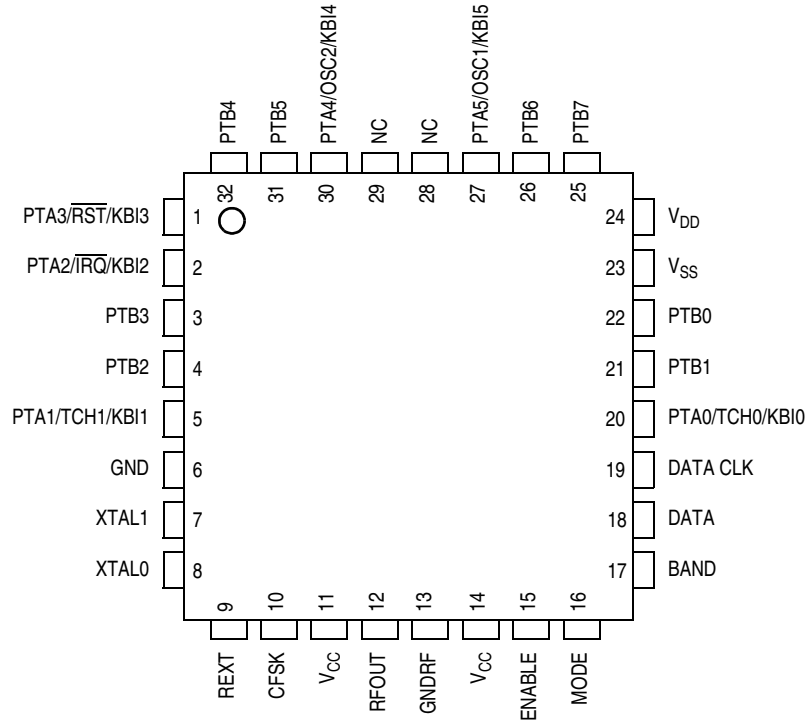


Figure 1-2. MC68HC908QF4 Pin Assignments

Table 1-1. Pin Functions

Pin Name	Description	Input/Output
$V_{DD}$	Power supply	Power
$V_{SS}$	Power supply ground	Power
PTA0	PTA0 — General purpose I/O port	Input/Output
	TCH0 — Timer Channel 0 I/O	Input/Output
	KBI0 — Keyboard interrupt input 0	Input
PTA1	PTA1 — General purpose I/O port	Input/Output
	TCH1 — Timer Channel 1 I/O	Input/Output
	KBI1 — Keyboard interrupt input 1	Input

**Table 1-1. Pin Functions (Continued)**

Pin Name	Description	Input/Output
PTA2	PTA2 — General purpose input-only port	Input
	$\overline{IRQ}$ — External interrupt with programmable pullup and Schmitt trigger input	Input
	KBI2 — Keyboard interrupt input 2	Input
PTA3	PTA3 — General purpose I/O port	Input/Output
	$\overline{RST}$ — Reset input, active low with internal pullup and Schmitt trigger	Input
	KBI3 — Keyboard interrupt input 3	Input
PTA4	PTA4 — General purpose I/O port	Input/Output
	OSC2 — XTAL oscillator output (XTAL option only) RC or internal oscillator output (OSC2EN = 1 in PTAPUE register)	Output Output
	KBI4 — Keyboard interrupt input 4	Input
PTA5	PTA5 — General purpose I/O port	Input/Output
	OSC1 — XTAL, RC, or external oscillator input	Input
	KBI5 — Keyboard interrupt input 5	Input
PTB[0:7]	8 general-purpose I/O ports	Input/Output

**Table 1-2. UHF Transmitter Pins**

Pin	Function	Description
6	GND	Ground
7	XTAL1	Reference oscillator input
8	XTAL0	Reference oscillator output
9	REXT	Output amplifier current setting resistor
10	CFSK	FSK switch output
11	V <sub>CC</sub>	Power supply
12	RFOUT	Power amplifier output
13	GNDRF	Power amplifier ground
14	V <sub>CC</sub>	Power supply
15	ENABLE	Enable input
16	MODE	Modulation type selection input
17	BAND	Frequency band selection
18	DATA	Data input
19	DATACLK	Clock output to the microcontroller



## Section 2. Memory

### 2.1 Introduction

The central processor unit (CPU08) can address 64 Kbytes of memory space. The memory map, shown in [Figure 2-1](#), includes:

- 4096 bytes of user FLASH
- 128 bytes of random access memory (RAM)
- 48 bytes of user-defined vectors, located in FLASH
- 416 bytes of monitor read-only memory (ROM)
- 1536 bytes of FLASH program and erase routines, located in ROM

### 2.2 Unimplemented Memory Locations

Accessing an unimplemented location can have unpredictable effects on MCU operation. In [Figure 2-1](#) and in register figures in this document, unimplemented locations are shaded.

### 2.3 Reserved Memory Locations

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

Memory

\$0000 ↓ \$003F	I/O REGISTERS 64 BYTES
\$0040 ↓ \$007F	RESERVED 64 BYTES
\$0080 ↓ \$00FF	RAM 128 BYTES
\$0100 ↓ \$27FF	UNIMPLEMENTED 9984 BYTES
\$2800 ↓ \$2DFF	AUXILIARY ROM 1536 BYTES
\$2E00 ↓ \$EDFF	UNIMPLEMENTED 49152 BYTES
\$EE00 ↓ \$FDFE	FLASH MEMORY 4096 BYTES
\$FE00 ↓ \$FE0F	SYSTEM REGISTERS
\$FE10 ↓ \$FFAF	MONITOR ROM 416 BYTES
\$FFB0 ↓ \$FFBD	FLASH 14 BYTES
\$FFBE	FLASH BLOCK PROTECT REGISTER (FLBPR)
\$FFBF	RESERVED FLASH
\$FFC0	INTERNAL OSCILLATOR TRIM VALUE
\$FFC1	RESERVED FLASH
\$FFC2 ↓ \$FFCF	FLASH 14 BYTES
\$FFD0 ↓ \$FFFF	USER VECTORS 48 BYTES

Figure 2-1. Memory Map

## 2.4 Input/Output (I/O) Section

Addresses \$0000–\$003F, shown in **Figure 2-2**, contain most of the control, status, and data registers. Additional I/O registers have these addresses:

- \$FE00 — Break status register, BSR
- \$FE01 — Reset status register, SRSR
- \$FE02 — Break auxiliary register, BRKAR
- \$FE03 — Break flag control register, BFCR
- \$FE04 — Interrupt status register 1, INT1
- \$FE05 — Interrupt status register 2, INT2
- \$FE06 — Interrupt status register 3, INT3
- \$FE07 — Reserved
- \$FE08 — FLASH control register, FLCR
- \$FE09 — Break address register high, BRKH
- \$FE0A — Break address register low, BRKL
- \$FE0B — Break status and control register, BRKSCR
- \$FE0C — LVI status register, LVISR
- \$FE0D — Reserved
- \$FFBE — FLASH block protect register, FLBPR
- \$FFC0 — Internal OSC trim value — Optional
- \$FFFF — COP control register, COPCTL

Memory

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0	
\$0000	Port A Data Register (PTA) <a href="#">See page 112.</a>	Read:	R	AWUL	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:	UNAFFECTED BY RESET							
\$0001	Port B Data Register (PTB) <a href="#">See page 115.</a>	Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:	Unaffected by reset							
\$0002 ↓ \$0003	Unimplemented									
\$0004	Data Direction Register A (DDRA) <a href="#">See page 113.</a>	Read:	R	R	DDRA5	DDRA4	DDRA3	0	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0005	Data Direction Register B (DDRB) <a href="#">See page 115.</a>	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0006 ↓ \$000A	Unimplemented									
\$000B	Port A Input Pullup Enable Register (PTAPUE) <a href="#">See page 114.</a>	Read:	OSC2EN	0	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000C	Port B Input Pullup Enable Register (PTBPUE) <a href="#">See page 116.</a>	Read:	PTBPUE7	PTBPUE6	PTBPUE5	PTBPUE4	PTBPUE3	PTBPUE2	PTBPUE1	PTBPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000D ↓ \$0019	Unimplemented									
\$001A	Keyboard Status and Control Register (KBSCR) <a href="#">See page 84.</a>	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
\$001B	Keyboard Interrupt Enable Register (KBIER) <a href="#">See page 85.</a>	Read:	0	AWUIE	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$001C	Unimplemented									

= Unimplemented    
 R = Reserved    
 U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 1 of 5)



Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0	
\$001D	IRQ Status and Control Register (INTSCR) <a href="#">See page 77.</a>	Read:	0	0	0	0	IRQF1	0	IMASK1	MODE1
		Write:						ACK1		
		Reset:	0	0	0	0	0	0	0	0
\$001E	Configuration Register 2 (CONFIG2) <sup>(1)</sup> <a href="#">See page 51.</a>	Read:	IRQPUD	IRQEN	R	OSCOPT1	OSCOPT0	R	R	RSTEN
		Write:								
		Reset:	0	0	0	0	0	0	0	0 <sup>(2)</sup>
		1. One-time writable register after each reset. 2. RSTEN reset to 0 by a power-on reset (POR) only.								
\$001F	Configuration Register 1 (CONFIG1) <sup>(1)</sup> <a href="#">See page 52.</a>	Read:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVDLVR	SSREC	STOP	COPD
		Write:								
		Reset:	0	0	0	0	0 <sup>(2)</sup>	0	0	0
		1. One-time writable register after each reset. Exceptions are LVDLVR and LVIRSTD bits. 2. LVDLVR reset to 0 by a power-on reset (POR) only.								
\$0020	TIM Status and Control Register (TSC) <a href="#">See page 147.</a>	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
\$0021	TIM Counter Register High (TCNTH) <a href="#">See page 149.</a>	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0022	TIM Counter Register Low (TCNTL) <a href="#">See page 149.</a>	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0023	TIM Counter Modulo Register High (TMODH) <a href="#">See page 149.</a>	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
\$0024	TIM Counter Modulo Register Low (TMODL) <a href="#">See page 149.</a>	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
\$0025	TIM Channel 0 Status and Control Register (TSC0) <a href="#">See page 150.</a>	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
\$0026	TIM Channel 0 Register High (TCH0H) <a href="#">See page 153.</a>	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:								

Indeterminate after reset

= Unimplemented     
 R = Reserved     
 U = Unaffected

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

Memory

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0		
\$0027	TIM Channel 0 Register Low (TCH0L) <a href="#">See page 153.</a>	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
		Write:									
		Reset:	Indeterminate after reset								
\$0028	TIM Channel 1 Status and Control Register (TSC1) <a href="#">See page 150.</a>	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX	
		Write:	0								
		Reset:	0	0	0	0	0	0	0	0	
\$0029	TIM Channel 1 Register High (TCH1H) <a href="#">See page 153.</a>	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
		Write:									
		Reset:	Indeterminate after reset								
\$002A	TIM Channel 1 Register Low (TCH1L) <a href="#">See page 153.</a>	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
		Write:									
		Reset:	Indeterminate after reset								
\$002B ↓ \$0035	Unimplemented										
\$0036	Oscillator Status Register (OSCSTAT) <a href="#">See page 98.</a>	Read:	R	R	R	R	R	R	ECGON	ECGST	
		Write:									
		Reset:	0	0	0	0	0	0	0	0	
\$0037	Unimplemented										
\$0038	Oscillator Trim Register (OSCTRIM) <a href="#">See page 99.</a>	Read:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0	
		Write:									
		Reset:	1	0	0	0	0	0	0	0	
\$0039 ↓ \$003F	Unimplemented										
\$FE00	Break Status Register (BSR) <a href="#">See page 161.</a>	Read:	R	R	R	R	R	R	SBSW	R	
		Write:								See note 1	
		Reset:	0								
1. Writing a 0 clears SBSW.											
\$FE01	SIM Reset Status Register (SRSR) <a href="#">See page 135.</a>	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0	
		Write:									
		POR:	1	0	0	0	0	0	0	0	

= Unimplemented    
  = Reserved    
 U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 3 of 5)

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0	
\$FE02	Break Auxiliary Register (BRKAR) <a href="#">See page 160.</a>	Read:	0	0	0	0	0	0	BDCOP	
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE03	Break Flag Control Register (BFCR) <a href="#">See page 161.</a>	Read:	BCFE	R	R	R	R	R	R	
		Write:								
		Reset:	0							
\$FE04	Interrupt Status Register 1 (INT1) <a href="#">See page 77.</a>	Read:	0	IF5	IF4	IF3	0	IF1	0	
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE05	Interrupt Status Register 2 (INT2) <a href="#">See page 77.</a>	Read:	IF14	0	0	0	0	0	0	
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE06	Interrupt Status Register 3 (INT3) <a href="#">See page 77.</a>	Read:	0	0	0	0	0	0	IF15	
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE07	Reserved	R	R	R	R	R	R	R	R	
\$FE08	FLASH Control Register (FLCR) <a href="#">See page 30.</a>	Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE09	Break Address High Register (BRKH) <a href="#">See page 160.</a>	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE0A	Break Address low Register (BRKL) <a href="#">See page 160.</a>	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE0B	Break Status and Control Register (BRKSCR) <a href="#">See page 159.</a>	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE0C	LVI Status Register (LVISR) <a href="#">See page 89.</a>	Read:	LVIOUT	0	0	0	0	0	0	R
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE0D ↓ \$FE0F	Reserved for FLASH Test	R	R	R	R	R	R	R	R	

= Unimplemented     
 R = Reserved     
 U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 5)


Memory

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$FFB0 ↓ \$FFBD	Unimplemented								
\$FFBE	FLASH Block Protect Register (FLBPR) <a href="#">See page 35.</a>	Read: BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	0
		Write:							
		Reset:	0	0	0	0	0	0	0
\$FFBF	Unimplemented								
\$FFC0	Internal Oscillator Trim Value (Optional)	Read: TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
		Write:							
		Reset:	1	0	0	0	0	0	0
\$FFC1	Reserved	R	R	R	R	R	R	R	R
\$FFC2 ↓ \$FFCF	Unimplemented								
\$FFFF	COP Control Register (COPCTL) <a href="#">See page 57.</a>	Read: LOW BYTE OF RESET VECTOR							
		Write: WRITING CLEARS COP COUNTER (ANY VALUE)							
		Reset: Unaffected by reset							
		<div style="display: flex; align-items: center; gap: 10px;"> <span><span style="background-color: #cccccc; border: 1px solid black; width: 20px; height: 10px; display: inline-block;"></span> = Unimplemented</span> <span><span style="border: 1px solid black; padding: 2px 5px;">R</span> = Reserved</span> <span>U = Unaffected</span> </div>							

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

Table 2-1 shows the MC68HC908QF4 reset and interrupt vectors.

Table 2-1. Vector Addresses

Vector Priority	Vector	Address	Vector
Lowest  Highest	IF14	\$FFE0	Keyboard vector (high)
		\$FFE1	Keyboard vector (low)
	IF13 ↓ IF6	—	Not used
	IF5	\$FFF2	TIM overflow vector (high)
		\$FFF3	TIM overflow vector (low)
	IF4	\$FFF4	TIM channel 1 vector (high)
		\$FFF5	TIM channel 1 vector (low)
	IF3	\$FFF6	TIM channel 0 vector (high)
		\$FFF7	TIM channel 0 vector (low)
	IF2	—	Not used
	IF1	\$FFFA	$\overline{IRQ}$ vector (high)
		\$FFFB	$\overline{IRQ}$ vector (low)
	—	\$FFFC	SWI vector (high)
		\$FFFD	SWI vector (low)
	—	\$FFFE	Reset vector (high)
		\$FFFF	Reset vector (low)

## 2.5 Random-Access Memory (RAM)

Addresses \$0080–\$00FF 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.

Before processing an interrupt, the central processor unit (CPU) uses five bytes of the stack to save the contents of the CPU registers.

**NOTE:** For M6805, M146805, and M68HC05 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.

## 2.6 FLASH Memory (FLASH)

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

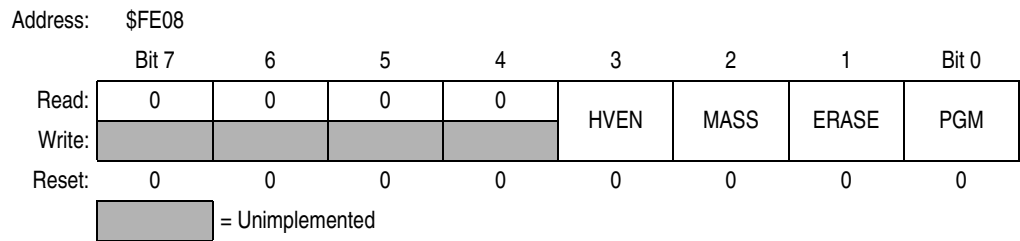
The FLASH memory consists of an array of 4096 bytes with an additional 48 bytes for user vectors. The minimum size of FLASH memory that can be erased is 64 bytes; and the maximum size of FLASH memory that can be programmed in a program cycle is 32 bytes (a row). Program and erase operations are facilitated through control bits in the FLASH control register (FLCR). Details for these operations appear later in this section. The address ranges for the user memory and vectors are:

- \$EE00 – \$FDFF; user memory, 4096 bytes
- \$FFD0 – \$FFFF; user interrupt vectors, 48 bytes.

**NOTE:** An erased bit reads as 1 and a programmed bit reads as 0. A security feature prevents viewing of the FLASH contents.<sup>(1)</sup>

### 2.6.1 FLASH Control Register

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



**Figure 2-3. FLASH Control Register (FLCR)**

**HVEN** — High Voltage Enable Bit

This read/write bit enables high voltage from the charge pump to the memory for either program or erase operation. It can only be set if either PGM = 1 or ERASE = 1 and the proper sequence for program or erase is followed.

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

**MASS** — Mass Erase Control Bit

This read/write bit configures the memory for mass erase operation.

- 1 = Mass erase operation selected
- 0 = Mass erase operation unselected

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

**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 unselected

**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 unselected

**2.6.2 FLASH Page Erase Operation**

Use the following procedure to erase a page of FLASH memory. A page consists of 64 consecutive bytes starting from addresses \$XX00, \$XX40, \$XX80, or \$XXC0. The 48-byte user interrupt vectors area also forms a page. Any FLASH memory page can be erased alone.

1. Set the ERASE bit and clear the MASS bit in the FLASH control register.
2. Read the FLASH block protect register.
3. Write any data to any FLASH location within the address range of the block to be erased.
4. Wait for a time,  $t_{NVS}$  (minimum 10  $\mu$ s).
5. Set the HVEN bit.
6. Wait for a time,  $t_{Erase}$  (minimum 1 ms or 4 ms).
7. Clear the ERASE bit.
8. Wait for a time,  $t_{NVH}$  (minimum 5  $\mu$ s).
9. Clear the HVEN bit.
10. After time,  $t_{RCV}$  (typical 1  $\mu$ s), the memory can be accessed in read mode again.

**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.*

In applications that need up to 10,000 program/erase cycles, use the 4 ms page erase specification to get improved long-term reliability. Any application can use this 4 ms page erase specification. However, in applications where a FLASH location will be erased and reprogrammed less than 1000 times, and speed is important, use the 1 ms page erase specification to get a lower minimum erase time.

### 2.6.3 FLASH Mass Erase Operation

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

1. Set both the ERASE bit and the MASS bit in the FLASH control register.
2. Read from the FLASH block protect register.
3. Write any data to any FLASH address<sup>(1)</sup> within the FLASH memory address range.
4. Wait for a time,  $t_{NVS}$  (minimum 10  $\mu$ s).
5. Set the HVEN bit.
6. Wait for a time,  $t_{Erase}$  (minimum 4 ms).
7. Clear the ERASE and MASS bits.

**NOTE:** Mass erase is disabled whenever any block is protected (FLBPR does not equal \$FF).

8. Wait for a time,  $t_{NVH1}$  (minimum 100  $\mu$ s).
9. Clear the HVEN bit.
10. After time,  $t_{RCV}$  (typical 1  $\mu$ s), the memory can be accessed in read mode again.

**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.

### 2.6.4 FLASH Program Operation

Programming of the FLASH memory is done on a row basis. A row consists of 32 consecutive bytes starting from addresses \$XX00, \$XX20, \$XX40, \$XX60, \$XX80, \$XXA0, \$XXC0, or \$XXE0. Use the following step-by-step procedure to program a row of FLASH memory

Figure 2-4 shows a flowchart of the programming algorithm.

**NOTE:** Only bytes which are currently \$FF may be programmed.

1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
2. Read from the FLASH block protect register.
3. Write any data to any FLASH location within the address range desired.
4. Wait for a time,  $t_{NVS}$  (minimum 10  $\mu$ s).
5. Set the HVEN bit.
6. Wait for a time,  $t_{PGS}$  (minimum 5  $\mu$ s).

1. When in monitor mode, with security sequence failed (see 16.3.2 Security), write to the FLASH block protect register instead of any FLASH address.



7. Write data to the FLASH address being programmed<sup>(1)</sup>.
8. Wait for time,  $t_{PROG}$  (minimum 30  $\mu$ s).
9. Repeat step 7 and 8 until all desired bytes within the row are programmed.
10. Clear the PGM bit<sup>(1)</sup>.
11. Wait for time,  $t_{NVH}$  (minimum 5  $\mu$ s).
12. Clear the HVEN bit.
13. After time,  $t_{RCV}$  (typical 1  $\mu$ s), the memory can be accessed in read mode again.

**NOTE:** The COP register at location \$FFFF should not be written between steps 5-12, when the HVEN bit is set. Since this register is located at a valid FLASH address, unpredictable behavior may occur if this location is written while HVEN is set.

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 [17.12 Memory Characteristics](#).

### 2.6.5 FLASH Protection

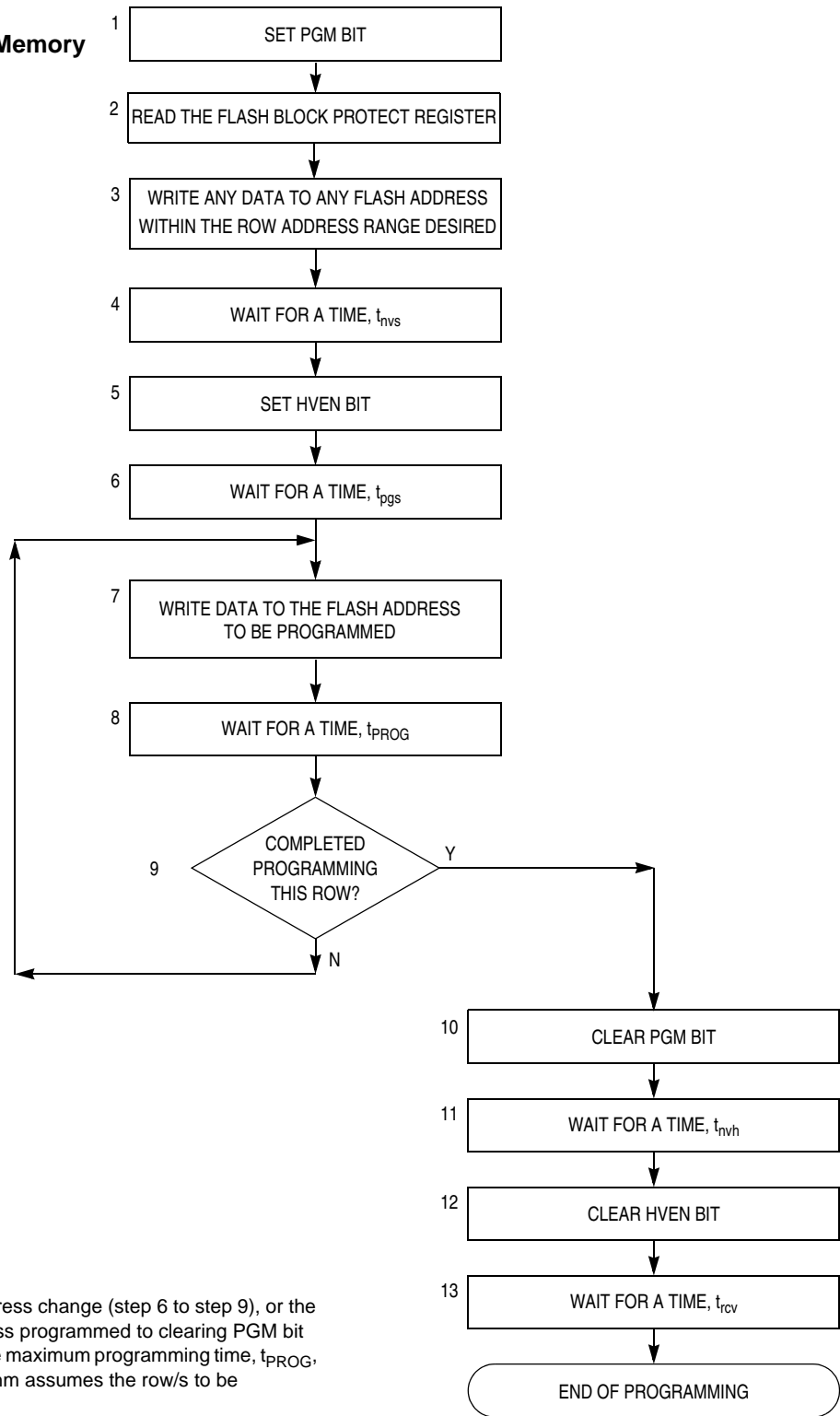
Due to the ability of the on-board charge pump to erase and program the FLASH memory in the target application, provision is made to protect blocks of memory from unintentional erase or program operations due to system malfunction. This protection is done by use of a FLASH block protect register (FLBPR).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

**NOTE:** In performing a program or erase operation, the FLASH block protect register must be read after setting the PGM or ERASE bit and before asserting the HVEN bit.

When the FLBPR is programmed with all 0 s, the entire memory is protected from being programmed and erased. When all the bits are erased (all 1's), the entire memory is accessible for program and erase.

1. The time between each FLASH address change, or the time between the last FLASH address programmed to clearing PGM bit, must not exceed the maximum programming time,  $t_{PROG}$  maximum.

Algorithm for Programming a Row (32 Bytes) of FLASH Memory



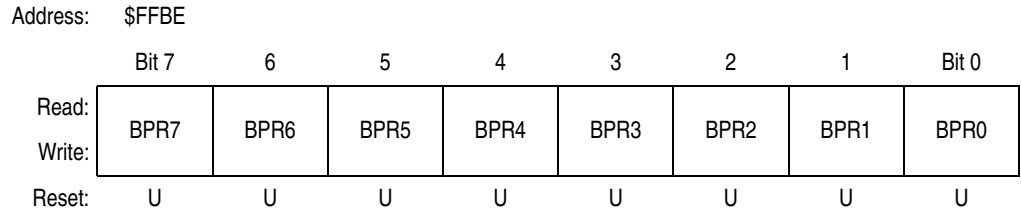
Notes:  
 The time between each FLASH address change (step 6 to step 9), or the time between the last FLASH address programmed to clearing PGM bit (step 6 to step 9) must not exceed the maximum programming time,  $t_{PROG}$ , maximum. This row program algorithm assumes the row/s to be programmed are initially erased.

Figure 2-4. FLASH Programming Flowchart

When bits within the FLBPR are programmed, they lock a block of memory. The address ranges are shown in **2.6.6 FLASH Block Protect Register**. Once the FLBPR is programmed with a value other than \$FF, any erase or program of the FLBPR or the protected block of FLASH memory is prohibited. Mass erase is disabled whenever any block is protected (FLBPR does not equal \$FF). The FLBPR itself can be erased or programmed only with an external voltage,  $V_{TST}$ , present on the  $\overline{IRQ}$  pin. This voltage also allows entry from reset into the monitor mode.

**2.6.6 FLASH Block Protect Register**

The FLASH block protect register is implemented as a byte within the FLASH memory, and therefore can only be written during a programming sequence of the FLASH memory. The value in this register determines the starting address of the protected range within the FLASH memory.



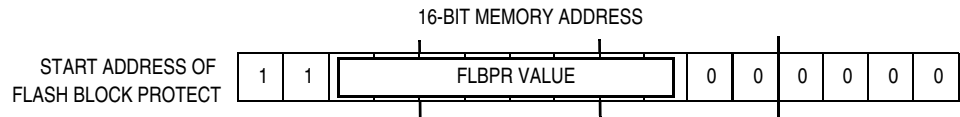
U = Unaffected by reset. Initial value from factory is 1.  
Write to this register is by a programming sequence to the FLASH memory.

**Figure 2-5. FLASH Block Protect Register (FLBPR)**

**BPR[7:0] — FLASH Protection Register Bits [7:0]**

These eight bits in FLBPR represent bits [13:6] of a 16-bit memory address. Bits [15:14] are 1s and bits [5:0] are 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. With this mechanism, the protect start address can be XX00, XX40, XX80, or XXC0 within the FLASH memory. See **Figure 2-6** and **Table 2-2**.



**Figure 2-6. FLASH Block Protect Start Address**

Table 2-2. Examples of Protect Start Address

BPR[7:0]	Start of Address of Protect Range
\$00–\$B8	The entire FLASH memory is protected.
\$B9 (1011 1001)	\$EE40 (1110 1110 0100 0000)
\$BA (1011 1010)	\$EE80 (1110 1110 1000 0000)
\$BB (1011 1011)	\$EEC0 (1110 1110 1100 0000)
\$BC (1011 1100)	\$EF00 (1110 1111 0000 0000)
and so on...	
\$DE (1101 1110)	\$F780 (1111 0111 1000 0000)
\$DF (1101 1111)	\$F7C0 (1111 0111 1100 0000)
\$FE (1111 1110)	\$FF80 (1111 1111 1000 0000) FLBPR, OSCTRIM, and vectors are protected
\$FF	The entire FLASH memory is not protected.

### 2.6.7 Wait Mode

Putting the MCU into wait mode while the FLASH is in read mode does not affect the operation of the FLASH memory directly, but there will not be any memory activity since the CPU is inactive.

The WAIT instruction should not be executed while performing a program or erase operation on the FLASH, or the operation will discontinue and the FLASH will be on standby mode.

### 2.6.8 Stop Mode

Putting the MCU into stop mode while the FLASH is in read mode does not affect the operation of the FLASH memory directly, but there will not be any memory activity since the CPU is inactive.

The STOP instruction should not be executed while performing a program or erase operation on the FLASH, or the operation will discontinue and the FLASH will be on standby mode

**NOTE:** Standby mode is the power-saving mode of the FLASH module in which all internal control signals to the FLASH are inactive and the current consumption of the FLASH is at a minimum.

### Section 3. Analog-to-Digital Converter (ADC)

#### 3.1 Introduction

This section describes the analog-to-digital converter (ADC). The ADC is an 8-bit, 4-channel analog-to-digital converter.

#### 3.2 Features

Features of the ADC module include:

- 4 channels with multiplexed input
- Linear successive approximation with monotonicity
- 8-bit resolution
- Single or continuous conversion
- Conversion complete flag or conversion complete interrupt
- Selectable ADC clock frequency

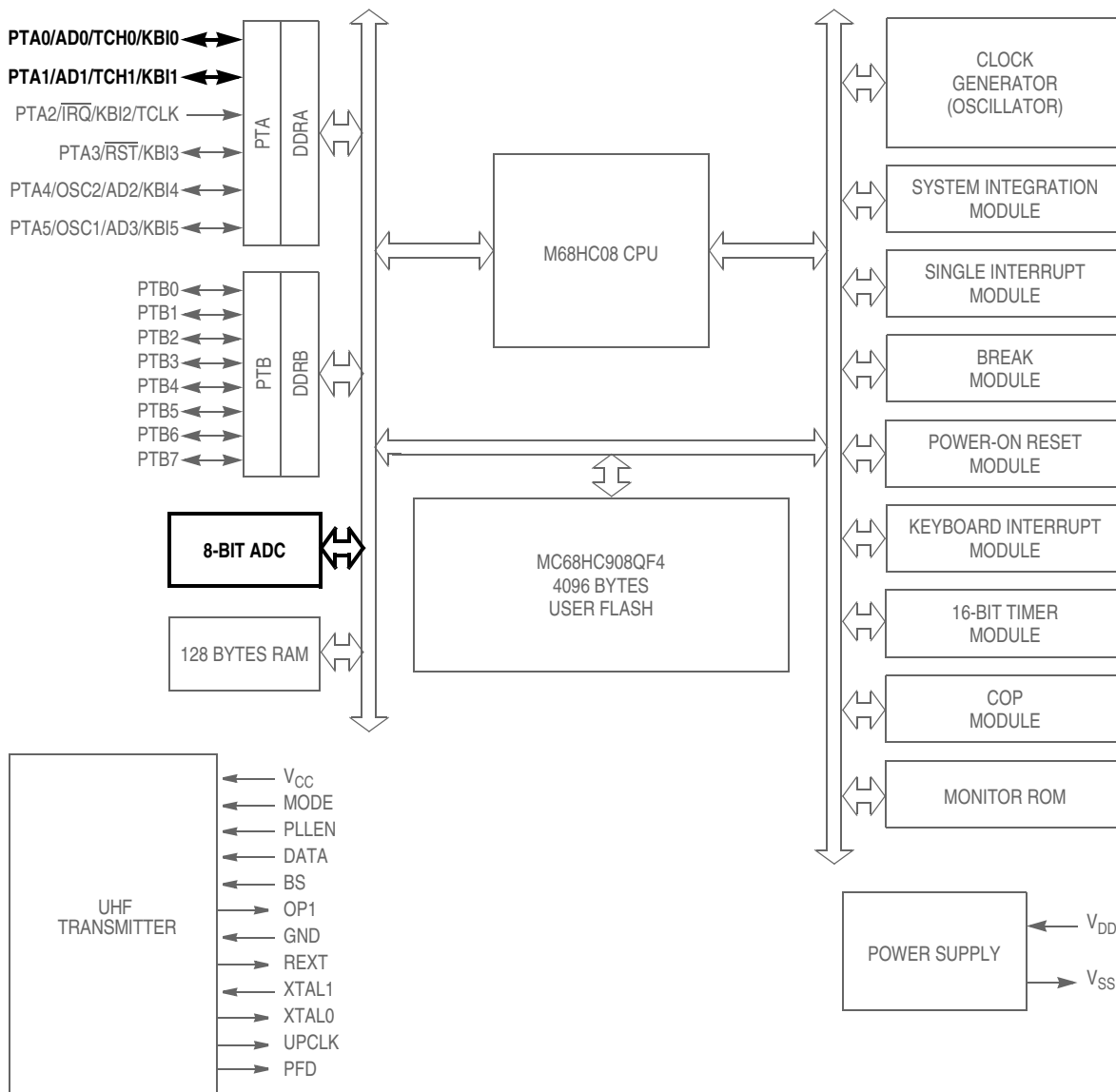
Figure 3-1 provides a summary of the input/output (I/O) registers.

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0	
\$003C	ADC Status and Control Register (ADSCR) <a href="#">See page 42.</a>	Read:	COCO	AIEN	ADCO	CH4	CH3	CH2	CH1	CH0
		Write:								
		Reset:	0	0	0	1	1	1	1	1
\$003D	Unimplemented									
\$003E	ADC Data Register (ADR) <a href="#">See page 43.</a>	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
\$003F	ADC Input Clock Register (ADICLK) <a href="#">See page 44.</a>	Read:	ADIV2	ADIV1	ADIV0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 3-1. ADC I/O Register Summary

Analog-to-Digital Converter (ADC)



$\overline{RST}$ ,  $\overline{IRQ}$ : Pins have internal (about 30K Ohms) pull up  
 PTA[0:5]: High current sink and source capability  
 PTA[0:5]: Pins have programmable keyboard interrupt and pull up

Figure 3-2. Block Diagram Highlighting ADC Block and Pins

### 3.3 Functional Description

Four ADC channels are available for sampling external sources at pins PTA0, PTA1, PTA4, and PTA5. An analog multiplexer allows the single ADC converter to select one of the four ADC channels as an ADC voltage input (ADCVIN). ADCVIN is converted by the successive approximation register-based counters. The ADC resolution is eight bits. When the conversion is completed, ADC puts the result in the ADC data register and sets a flag or generates an interrupt.

Figure 3-3 shows a block diagram of the ADC.

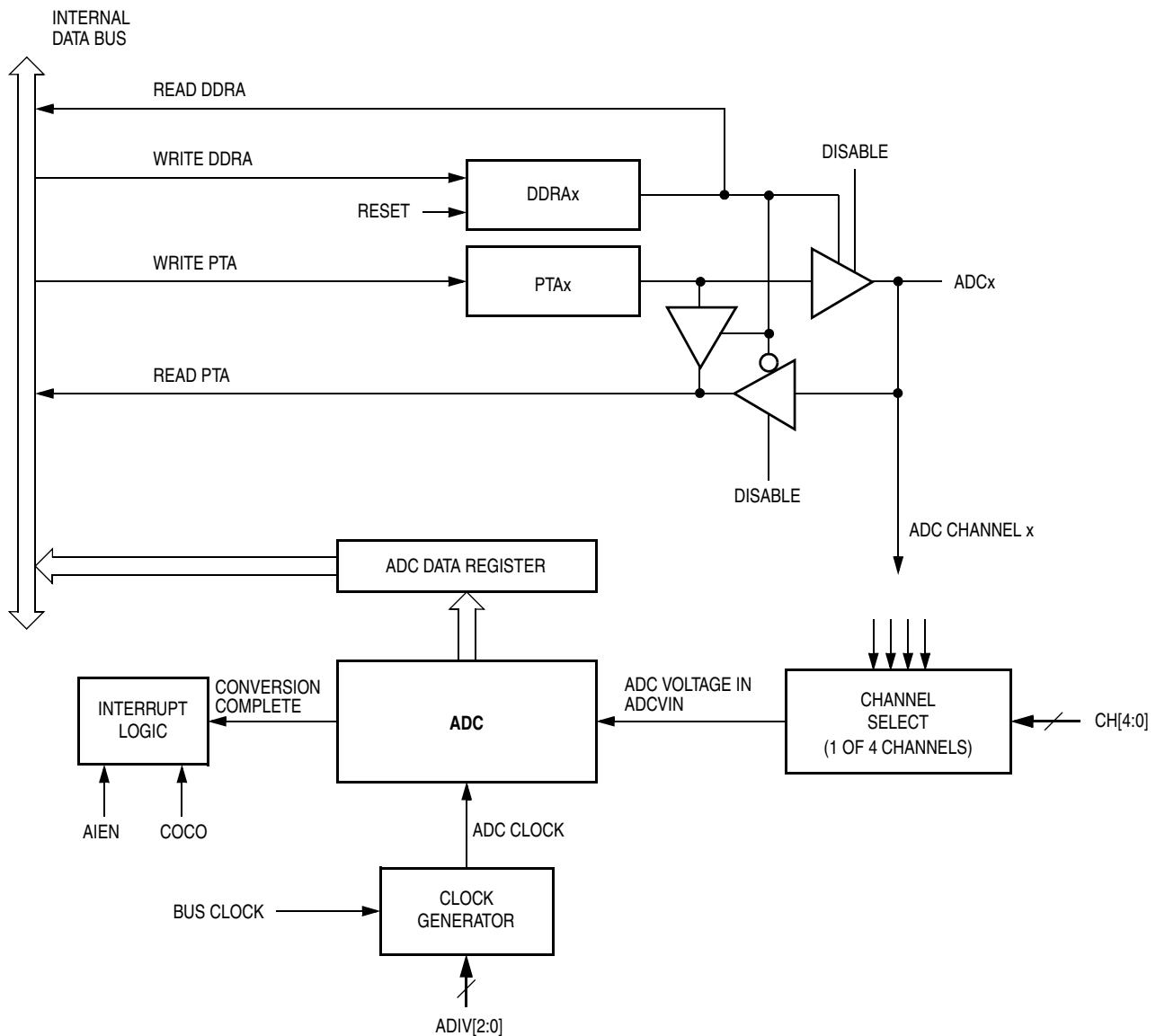


Figure 3-3. ADC Block Diagram

### 3.3.1 ADC Port I/O Pins

PTA0, PTA1, PTA4, and PTA5 are general-purpose I/O pins that are shared with the ADC channels. The channel select bits (ADC status and control register (ADSCR), \$003C), 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 data direction register (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 a 0 if the corresponding DDR bit is at 0. If the DDR bit is 1, the value in the port data latch is read.

### 3.3.2 Voltage Conversion

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

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

### 3.3.3 Conversion Time

Sixteen 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 16  $\mu$ s to complete. With a 1-MHz ADC internal clock the maximum sample rate is 62.5 kHz.

$$\text{Conversion Time} = \frac{16 \text{ ADC Clock Cycles}}{\text{ADC Clock Frequency}}$$

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

### 3.3.4 Continuous Conversion

In the continuous conversion mode (ADCO = 1), the ADC continuously converts the selected channel filling the ADC data register (ADR) 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 (ADSCR, \$003C) is set after each conversion and will stay set until the next read of the ADC data register.

When a conversion is in process and the ADSCR is written, the current conversion data should be discarded to prevent an incorrect reading.

### 3.3.5 Accuracy and Precision

The conversion process is monotonic and has no missing codes.



## 3.4 Interrupts

When the AIEN bit is set, the ADC module is capable of generating a central processor unit (CPU) interrupt after each ADC conversion. A CPU interrupt is generated if the COCO bit is at 0. The COCO bit is not used as a conversion complete flag when interrupts are enabled.

## 3.5 Low-Power Modes

The following subsections describe the ADC in low-power modes.

### 3.5.1 Wait Mode

The ADC continues normal operation during wait mode. Any enabled CPU interrupt request from the ADC can bring the microcontroller unit (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 CH[4:0] bits in ADSCR to 1s before executing the WAIT instruction.

### 3.5.2 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 using ADC data after exiting stop mode.

## 3.6 Input/Output Signals

The ADC module has four channels that are shared with I/O port A.

ADC voltage in (ADCVIN) is the input voltage signal from one of the four ADC channels to the ADC module.

## 3.7 Input/Output Registers

These I/O registers control and monitor ADC operation:

- ADC status and control register (ADSCR)
- ADC data register (ADR)
- ADC clock register (ADICLK)

3.7.1 ADC Status and Control Register

The following paragraphs describe the function of the ADC status and control register (ADSCR). When a conversion is in process and the ADSCR is written, the current conversion data should be discarded to prevent an incorrect reading.

Address: \$003C

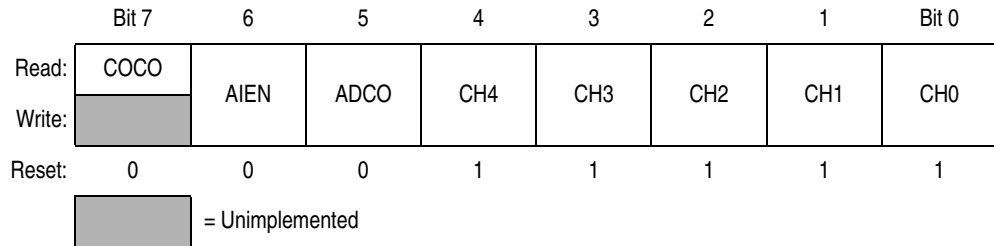


Figure 3-4. ADC Status and Control Register (ADSCR)

COCO — Conversions Complete Bit

In non-interrupt mode (AIEN = 0), COCO is a read-only bit that is set at the end of each conversion. COCO will stay set until cleared by a read of the ADC data register. Reset clears this bit.

In interrupt mode (AIEN = 1), COCO is a read-only bit that is not set at the end of a conversion. It always reads as a 0.

1 = Conversion completed (AIEN = 0)

0 = Conversion not completed (AIEN = 0) or CPU interrupt enabled (AIEN = 1)

**NOTE:** The write function of the COCO bit is reserved. When writing to the ADSCR register, always have a 0 in the COCO bit position.

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 ADR is read or ADSCR is written. Reset clears the AIEN bit.

1 = ADC interrupt enabled

0 = ADC interrupt disabled

ADCO — ADC Continuous Conversion Bit

When set, the ADC will convert samples continuously and update ADR 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

CH[4:0] — ADC Channel Select Bits

CH4, CH3, CH2, CH1, and CH0 form a 5-bit field which is used to select one of the four ADC channels. The five select bits are detailed in Table 3-1. 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. The ADC subsystem is turned off when the channel select bits are all set to 1. This feature allows for reduced power consumption for the MCU when the ADC is not used. Reset sets all of these bits to a 1.

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

**Table 3-1. MUX Channel Select**

CH4	CH3	CH2	CH1	CH0	ADC Channel	Input Select
0	0	0	0	0	AD0	PTA0
0	0	0	0	1	AD1	PTA1
0	0	0	1	0	AD2	PTA4
0	0	0	1	1	AD3	PTA5
0	0	1	0	0	—	Unused <sup>(1)</sup>
↓	↓	↓	↓	↓	—	
1	1	0	1	0	—	
1	1	0	1	1	—	Reserved
1	1	1	0	0	—	Unused
1	1	1	0	1	—	V <sub>DDA</sub> <sup>(2)</sup>
1	1	1	1	0	—	V <sub>SSA</sub> <sup>(2)</sup>
1	1	1	1	1	—	ADC power off

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.

### 3.7.2 ADC Data Register

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

Address: \$003E



**Figure 3-5. ADC Data Register (ADR)**

3.7.3 ADC Input Clock Register

This register selects the clock frequency for the ADC.

Address: \$003F

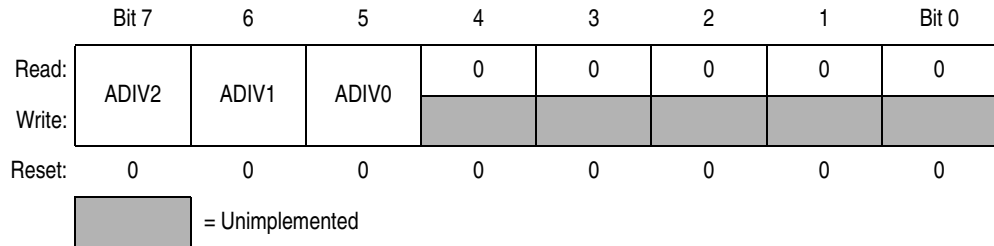


Figure 3-6. ADC Input Clock Register (ADICLK)

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 3-2 shows the available clock configurations. The ADC clock should be set according to the MCU operating voltage. Lower operating voltages will require lower ADC clock frequencies for best accuracy. The analog input level should remain stable for the entire conversion time (maximum = 17 ADC clock cycles).

Table 3-2. ADC Clock Divide Ratio

ADIV2	ADIV1	ADIV0	ADC Clock Rate
0	0	0	Bus clock ÷ 1
0	0	1	Bus clock ÷ 2
0	1	0	Bus clock ÷ 4
0	1	1	Bus clock ÷ 8
1	X	X	Bus clock ÷ 16

X = don't care

Section 4. Auto Wakeup Module (AWU)

4.1 Introduction

This section describes the auto wakeup module (AWU). The AWU generates a periodic interrupt during stop mode to wake the part up without requiring an external signal. Figure 4-2 is a block diagram of the AWU.

4.2 Features

Features of the auto wakeup module include:

- One internal interrupt with separate interrupt enable bit, sharing the same keyboard interrupt vector and keyboard interrupt mask bit
- Exit from low-power stop mode without external signals
- Selectable timeout periods
- Dedicated low power internal oscillator separate from the main system clock sources

Figure 4-1 provides a summary of the input/output (I/O) registers used in conjunction with the AWU.

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0	
\$0000	Port A Data Register (PTA) <a href="#">See page 48.</a>	Read:	0	AWUL	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:	Unaffected by reset							
\$001A	Keyboard Status and Control Register (KBSCR) <a href="#">See page 48.</a>	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
\$001B	Keyboard Interrupt Enable Register (KBIER) <a href="#">See page 49.</a>	Read:	0							
		Write:		AWUIE	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figure 4-1. AWU Register Summary

### 4.3 Functional Description

The function of the auto wakeup logic is to generate periodic wakeup requests to bring the microcontroller unit (MCU) out of stop mode. The wakeup requests are treated as regular keyboard interrupt requests, with the difference that instead of a pin, the interrupt signal is generated by an internal logic.

Writing the AWUIE bit in the keyboard interrupt enable register enables or disables the auto wakeup interrupt input (see [Figure 4-2](#)). A logic 1 applied to the AWUIREQ input with auto wakeup interrupt request enabled, latches an auto wakeup interrupt request.

Auto wakeup latch, AWUL, can be read directly from the bit 6 position of port A data register (PTA). This is a read-only bit which is occupying an empty bit position on PTA. No PTA associated registers, such as PTA6 data direction or PTA6 pullup exist for this bit.

Entering stop mode will enable the auto wakeup generation logic. An internal RC oscillator (exclusive for the auto wakeup feature) drives the wakeup request generator. Once the overflow count is reached in the generator counter, a wakeup request, AWUIREQ, is latched and sent to the KBI logic. See [Figure 4-1](#).

Wakeup interrupt requests will only be serviced if the associated interrupt enable bit, AWUIE, in KBIER is set. The AWU shares the keyboard interrupt vector.

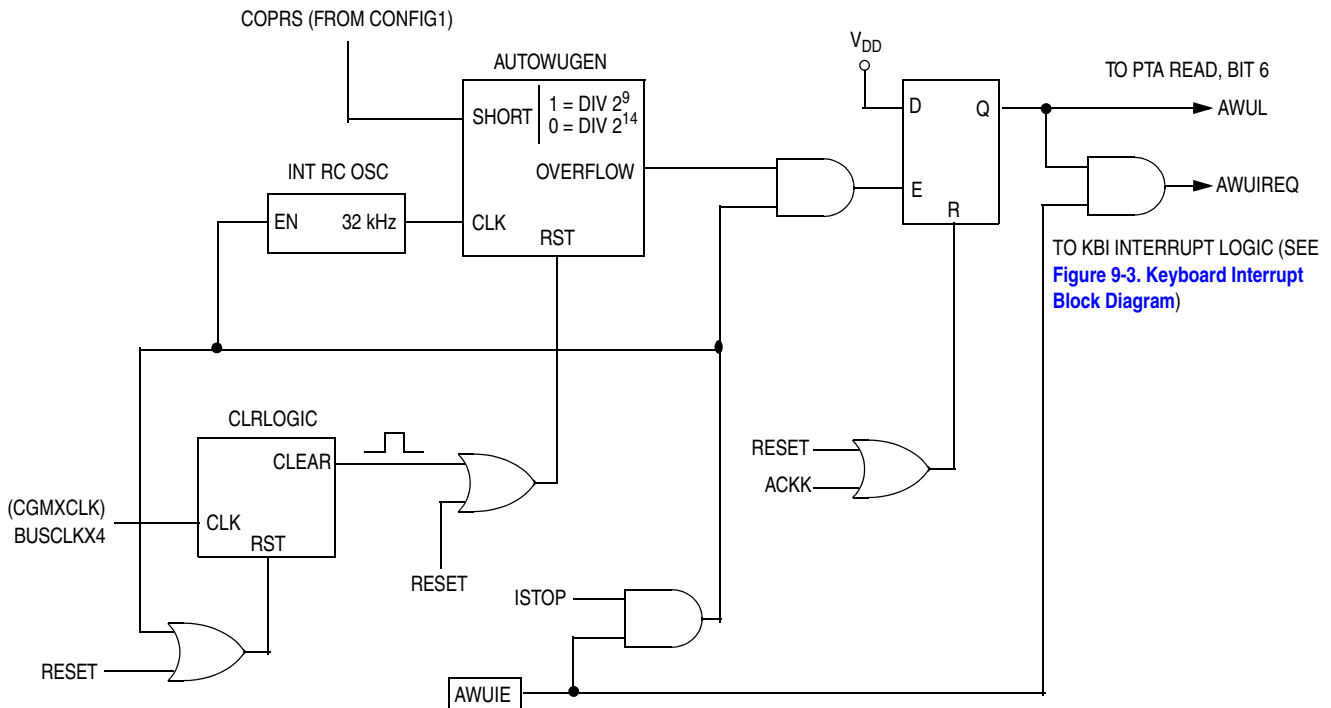


Figure 4-2. Auto Wakeup Interrupt Request Generation Logic

The overflow count can be selected from two options defined by the COPRS bit in CONFIG1. This bit was “borrowed” from the computer operating properly (COP) using the fact that the COP feature is idle (no MCU clock available) in stop mode. The typical values of the periodic wakeup request are (at room temperature):

- COPRS = 0: 875 ms @ 3.0 V, 1.1 s @ 2.3 V
- COPRS = 1: 22 ms @ 3.0 V, 27 ms @ 2.3 V

The auto wakeup RC oscillator is highly dependent on operating voltage and temperature. This feature is not recommended for use as a time-keeping function.

The wakeup request is latched to allow the interrupt source identification. The latched value, AWUL, can be read directly from the bit 6 position of PTA data register. This is a read-only bit which is occupying an empty bit position on PTA. No PTA associated registers, such as PTA6 data, PTA6 direction, and PTA6 pullup exist for this bit. The latch can be cleared by writing to the ACKK bit in the KBSCR register. Reset also clears the latch. AWUIE bit in KBI interrupt enable register (see [Figure 4-2](#)) has no effect on AWUL reading.

The AWU oscillator and counters are inactive in normal operating mode and become active only upon entering stop mode.

## 4.4 Wait Mode

The AWU module remains inactive in wait mode.

## 4.5 Stop Mode

When the AWU module is enabled (AWUIE = 1 in the keyboard interrupt enable register) it is activated automatically upon entering stop mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of stop mode. The AWU counters start from ‘0’ each time stop mode is entered.

## 4.6 Input/Output Registers

The AWU shares registers with the keyboard interrupt (KBI) module and the port A I/O module. The following I/O registers control and monitor operation of the AWU:

- Port A data register (PTA)
- Keyboard interrupt status and control register (KBSCR)
- Keyboard interrupt enable register (KBIER)

Auto Wakeup Module (AWU)

4.6.1 Port A I/O Register

The port A data register (PTA) contains a data latch for the state of the AWU interrupt request, in addition to the data latches for port A.

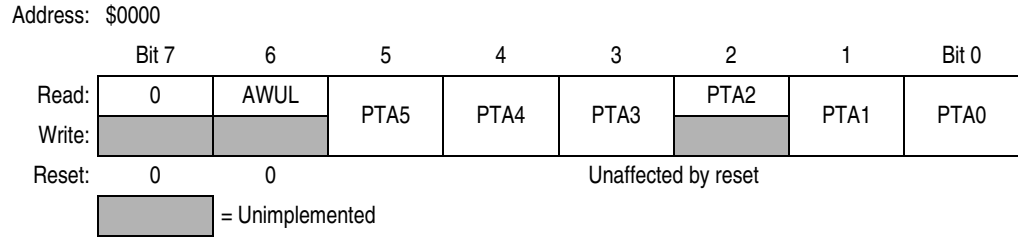


Figure 4-3. Port A Data Register (PTA)

AWUL — Auto Wakeup Latch

This is a read-only bit which has the value of the auto wakeup interrupt request latch. The wakeup request signal is generated internally. There is no PTA6 port or any of the associated bits such as PTA6 data direction or pullup bits.

- 1 = Auto wakeup interrupt request is pending
- 0 = Auto wakeup interrupt request is not pending

**NOTE:** PTA5–PTA0 bits are not used in conjunction with the auto wakeup feature. To see a description of these bits, see [13.2.1 Port A Data Register](#).

4.6.2 Keyboard Status and Control Register

The keyboard status and control register (KBSCR):

- Flags keyboard/auto wakeup interrupt requests
- Acknowledges keyboard/auto wakeup interrupt requests
- Masks keyboard/auto wakeup interrupt requests

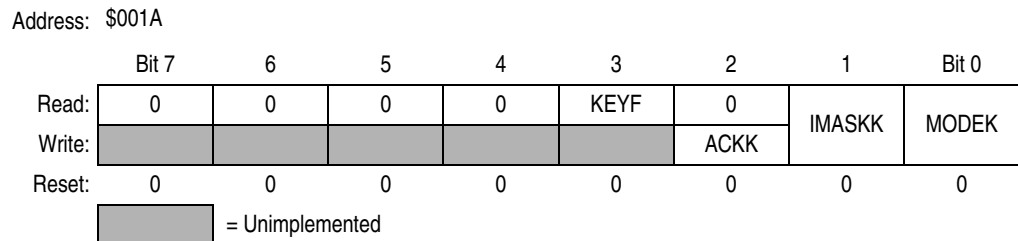


Figure 4-4. Keyboard Status and Control Register (KBSCR)

Bits 7–4 — Not used

These read-only bits always read as 0s.

KEYF — Keyboard Flag Bit

This read-only bit is set when a keyboard interrupt is pending on port A or auto wakeup. Reset clears the KEYF bit.

- 1 = Keyboard/auto wakeup interrupt pending
- 0 = No keyboard/auto wakeup interrupt pending



**ACKK** — Keyboard Acknowledge Bit

Writing a 1 to this write-only bit clears the keyboard/auto wakeup interrupt request on port A and auto wakeup logic. ACKK always reads as 0. Reset clears ACKK.

**IMASKK**— Keyboard Interrupt Mask Bit

Writing a 1 to this read/write bit prevents the output of the keyboard interrupt mask from generating interrupt requests on port A or auto wakeup. Reset clears the IMASKK bit.

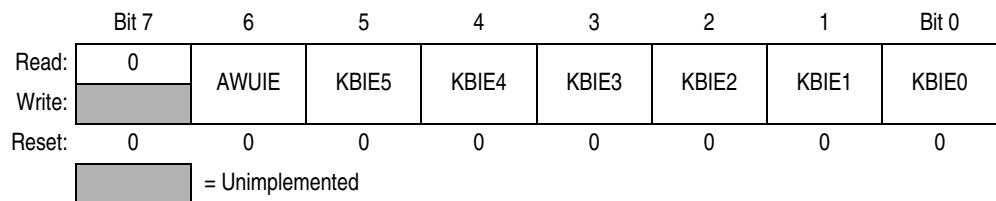
- 1 = Keyboard/auto wakeup interrupt requests masked
- 0 = Keyboard/auto wakeup interrupt requests not masked

**NOTE:** *MODEK is not used in conjunction with the auto wakeup feature. To see a description of this bit, see [9.7.1 Keyboard Status and Control Register](#).*

**4.6.3 Keyboard Interrupt Enable Register**

The keyboard interrupt enable register (KBIER) enables or disables the auto wakeup to operate as a keyboard/auto wakeup interrupt input.

Address: \$001B



**Figure 4-5. Keyboard Interrupt Enable Register (KBIER)**

**AWUIE** — Auto Wakeup Interrupt Enable Bit

This read/write bit enables the auto wakeup interrupt input to latch interrupt requests. Reset clears AWUIE.

- 1 = Auto wakeup enabled as interrupt input
- 0 = Auto wakeup not enabled as interrupt input

**NOTE:** *KBIE5–KBIE0 bits are not used in conjunction with the auto wakeup feature. To see a description of these bits, see [9.7.2 Keyboard Interrupt Enable Register](#).*



## Section 5. Configuration Register (CONFIG)

### 5.1 Introduction

This section describes the configuration registers (CONFIG1 and CONFIG2). The configuration registers enable or disable the following options:

- Stop mode recovery time ( $32 \times \text{BUSCLKX4}$  cycles or  $4096 \times \text{BUSCLKX4}$  cycles)
- STOP instruction
- Computer operating properly module (COP)
- COP reset period (COPRS):  $(2^{13}-2^4) \times \text{BUSCLKX4}$  or  $(2^{18}-2^4) \times \text{BUSCLKX4}$
- Low-voltage inhibit (LVI) enable and trip voltage selection
- OSC option selection
- $\overline{\text{IRQ}}$  pin
- $\overline{\text{RST}}$  pin
- Auto wakeup timeout period

### 5.2 Functional Description

The configuration registers are used in the initialization of various options. The configuration registers can be written once after each reset. Exceptions are bits LVDLVR and LVIRSTD which may be written at any time. Most of the configuration register bits are cleared during reset. Since the various options affect the operation of the microcontroller unit (MCU) it is recommended that this register be written immediately after reset. The configuration registers are located at \$001E and \$001F, and may be read at anytime.

Address: \$001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IRQPUD	IRQEN	R	OSCOPT1	OSCOPT0	R	R	RSTEN
Write:								
Reset:	0	0	0	0	0	0	0	U
POR:	0	0	0	0	0	0	0	0

R = Reserved                      U = Unaffected

Figure 5-1. Configuration Register 2 (CONFIG2)

Configuration Register (CONFIG)

- IRQPUD —  $\overline{\text{IRQ}}$  Pin Pullup Control Bit
  - 1 = Internal pullup is disconnected
  - 0 = Internal pullup is connected between  $\overline{\text{IRQ}}$  pin and  $V_{DD}$
- IRQEN —  $\overline{\text{IRQ}}$  Pin Function Selection Bit
  - 1 = Interrupt request function active in pin
  - 0 = Interrupt request function inactive in pin
- OSCOPT1 and OSCOPT0 — Selection Bits for Oscillator Option
  - (0, 0) Internal oscillator
  - (0, 1) External oscillator
  - (1, 0) External RC oscillator
  - (1, 1) External XTAL oscillator
- RSTEN —  $\overline{\text{RST}}$  Pin Function Selection
  - 1 = Reset function active in pin
  - 0 = Reset function inactive in pin

**NOTE:** The RSTEN bit is cleared by a power-on reset (POR) only. Other resets will leave this bit unaffected.

Address: \$001F

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

U = Unaffected

**Figure 5-2. Configuration Register 1 (CONFIG1)**

- COPRS (Out of STOP Mode) — COP Reset Period Selection Bit
  - 1 = COP reset short cycle =  $(2^{13} - 2^4) \times \text{BUSCLKX4}$
  - 0 = COP reset long cycle =  $(2^{18} - 2^4) \times \text{BUSCLKX4}$
- COPRS (In STOP Mode) — Auto Wakeup Period Selection Bit
  - 1 = Auto wakeup short cycle =  $(2^9) \times \text{INTRCOSC}$
  - 0 = Auto wakeup long cycle =  $(2^{14}) \times \text{INTRCOSC}$
- LVISTOP — LVI Enable in Stop Mode Bit
 

When the LVIPWRD bit is clear, setting the LVISTOP bit enables the LVI to operate during stop mode. Reset clears LVISTOP.

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

## LVIRSTD — LVI Reset Disable Bit

LVIRSTD disables the reset signal from the LVI module. Unlike other configuration bits, the LVIRSTD can be written at any time.

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

## LVIPWRD — LVI Power Disable Bit

LVIPWRD disables the LVI module.

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

## LVDLVR — Low Voltage Detect or Low Voltage Reset Mode Bit

LVDLVR selects the trip voltage of the LVI module. LVD trip voltage can be used as a low voltage warning, while LVR will commonly be used as a reset condition. Unlike other CONFIG bits, LVDLVR can be written multiple times after reset.

- 1 = LVI trip voltage level set to LVD trip voltage
- 0 = LVI trip voltage level set to LVR trip voltage

**NOTE:** *The LVDLVR bit is cleared by a power-on reset (POR) only. Other resets will leave this bit unaffected.*

## SSREC — Short Stop Recovery Bit

SSREC enables the CPU to exit stop mode with a delay of 32 BUSCLKX4 cycles instead of a 4096 BUSCLKX4 cycle delay.

- 1 = Stop mode recovery after 32 BUSCLKX4 cycles
- 0 = Stop mode recovery after 4096 BUSCLKX4 cycles

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

When using the LVI during normal operation but disabling during stop mode, the LVI will have an enable time of  $t_{EN}$ . The system stabilization time for power-on reset and long stop recovery (both 4096 BUSCLKX4 cycles) gives a delay longer than the LVI enable time for these startup scenarios. There is no period where the MCU is not protected from a low-power condition. However, when using the short stop recovery configuration option, the 32 BUSCLKX4 delay must be greater than the LVI's turn on time to avoid a period in startup where the LVI is not protecting the MCU.

## 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.

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



## Section 6. Computer Operating Properly (COP)

### 6.1 Introduction

The computer operating properly (COP) module contains a free-running counter that generates a reset if allowed to overflow. The COP module helps software recover from runaway code. Prevent a COP reset by clearing the COP counter periodically. The COP module can be disabled through the COPD bit in the configuration 1 (CONFIG1) register.

### 6.2 Functional Description

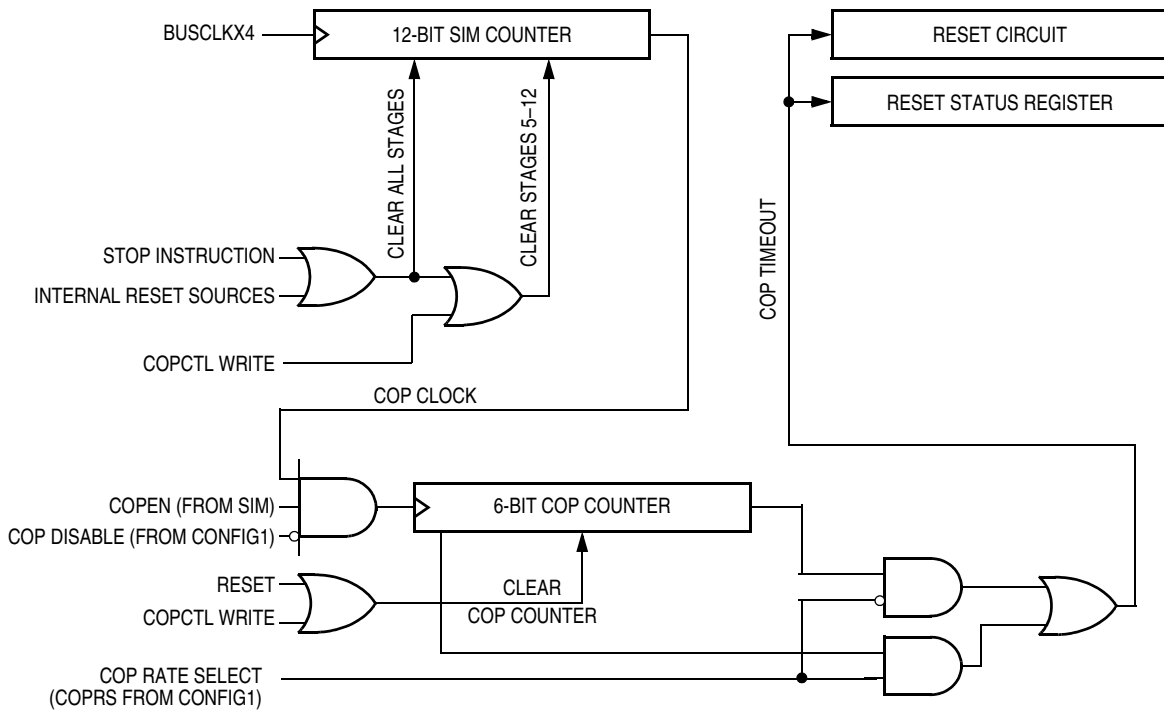


Figure 6-1. COP Block Diagram

The COP counter is a free-running 6-bit counter preceded by the 12-bit system integration module (SIM) counter. If not cleared by software, the COP counter overflows and generates an asynchronous reset after  $2^{18} - 2^4$  or  $2^{13} - 2^4$  BUSCLKX4 cycles; depending on the state of the COP rate select bit, COPRS, in configuration register 1. With a  $2^{18} - 2^4$  BUSCLKX4 cycle overflow option, the internal 12.8-MHz oscillator gives a COP timeout period of 20.48 ms. Writing any value to location \$FFFF before an overflow occurs prevents a COP reset by clearing the COP counter and stages 12–5 of the SIM counter.

**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 (if the RSTEN bit is set in the CONFIG1 register) for  $32 \times \text{BUSCLKX4}$  cycles and sets the COP bit in the reset status register (RSR). See [14.8.1 SIM Reset Status Register](#).

**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.

### 6.3 I/O Signals

The following paragraphs describe the signals shown in [Figure 6-1](#).

#### 6.3.1 BUSCLKX4

BUSCLKX4 is the oscillator output signal. BUSCLKX4 frequency is equal to the crystal frequency or the RC-oscillator frequency.

#### 6.3.2 STOP Instruction

The STOP instruction clears the SIM counter.

#### 6.3.3 COPCTL Write

Writing any value to the COP control register (COPCTL) (see [6.4 COP Control Register](#)) clears the COP counter and clears stages 12–5 of the SIM counter. Reading the COP control register returns the low byte of the reset vector.

#### 6.3.4 Power-On Reset

The power-on reset (POR) circuit in the SIM clears the SIM counter  $4096 \times \text{BUSCLKX4}$  cycles after power up.

#### 6.3.5 Internal Reset

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



### 6.3.6 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the configuration register (CONFIG). See [Section 5. Configuration Register \(CONFIG\)](#).

### 6.3.7 COPRS (COP Rate Select)

The COPRS signal reflects the state of the COP rate select bit (COPRS) in the configuration register 1 (CONFIG1). See [Section 5. Configuration Register \(CONFIG\)](#).

## 6.4 COP Control Register

The COP control register (COPCTL) 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.

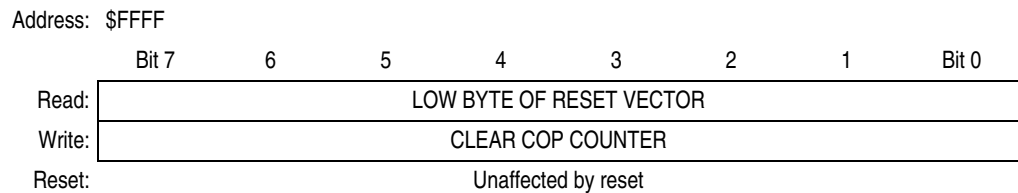


Figure 6-2. COP Control Register (COPCTL)

## 6.5 Interrupts

The COP does not generate CPU interrupt requests.

## 6.6 Monitor Mode

The COP is disabled in monitor mode when  $V_{TST}$  is present on the  $\overline{IRQ}$  pin.

## 6.7 Low-Power Modes

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

### 6.7.1 Wait Mode

The COP continues to operate during wait mode. To prevent a COP reset during wait mode, periodically clear the COP counter.

### 6.7.2 Stop Mode

Stop mode turns off the BUSCLKX4 input to the COP and clears the SIM counter. Service the COP immediately before entering or after exiting stop mode to ensure a full COP timeout period after entering or exiting stop mode.

### 6.8 COP Module During Break Mode

The COP is disabled during a break interrupt with monitor mode when BDCOP bit is set in break auxiliary register (BRKAR).

## Section 7. Central Processor Unit (CPU)

### 7.1 Introduction

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

### 7.2 Features

Features of the CPU include:

- 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

7.3 CPU Registers

Figure 7-1 shows the five CPU registers. CPU registers are not part of the memory map.

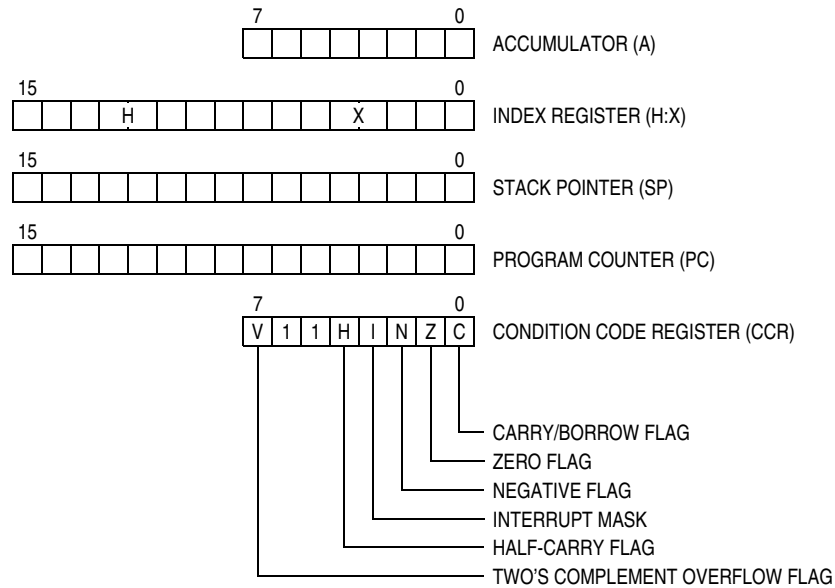


Figure 7-1. CPU Registers

7.3.1 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.

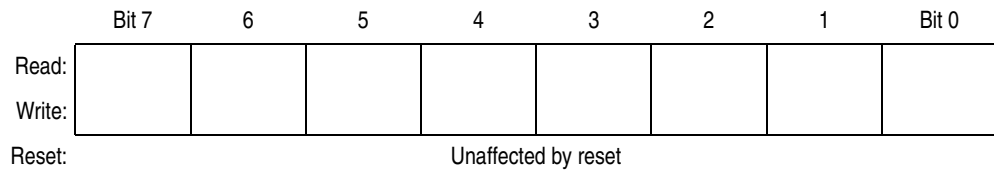


Figure 7-2. Accumulator (A)

### 7.3.2 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.

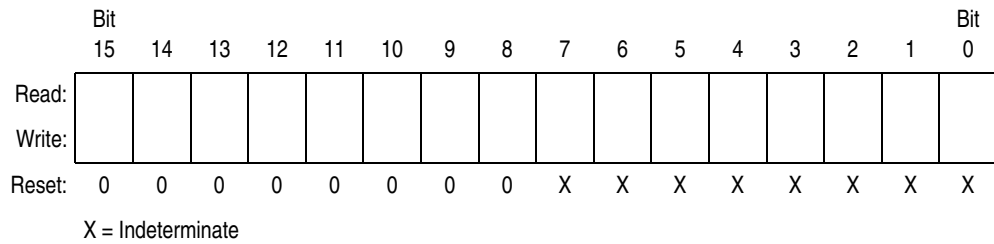


Figure 7-3. Index Register (H:X)

### 7.3.3 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.

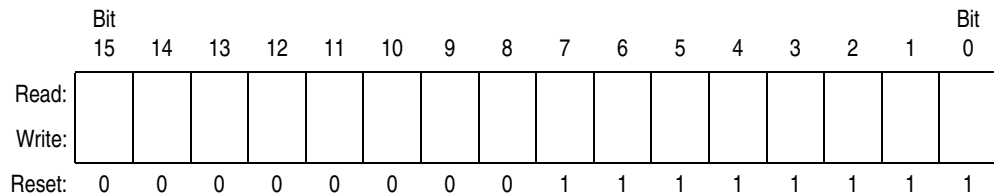


Figure 7-4. Stack Pointer (SP)

**NOTE:** The location of the stack is arbitrary and may be relocated anywhere in random-access memory (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.

### 7.3.4 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.

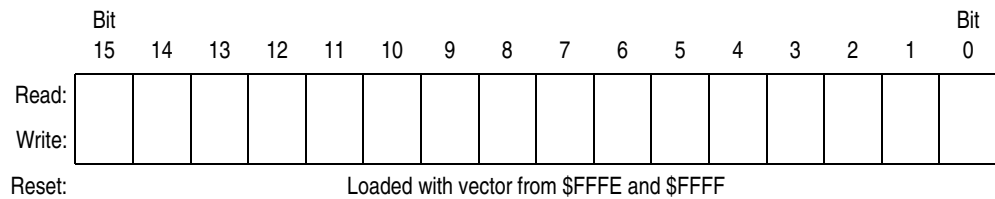


Figure 7-5. Program Counter (PC)

### 7.3.5 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 1. The following paragraphs describe the functions of the condition code register.

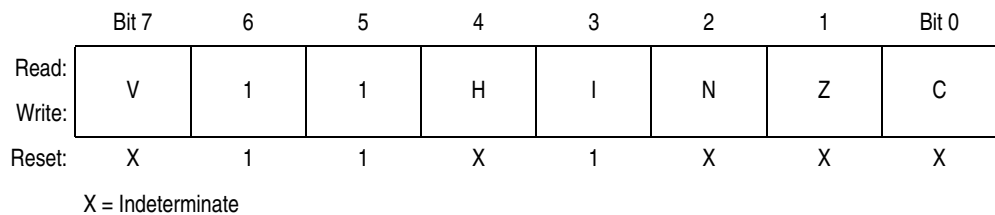


Figure 7-6. Condition Code Register (CCR)

#### 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

### 7.4 Arithmetic/Logic Unit (ALU)

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

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

### 7.5 Low-Power Modes

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

#### 7.5.1 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

#### 7.5.2 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.

### 7.6 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.

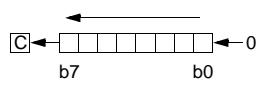
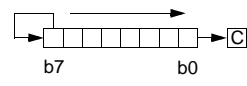
A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation if the break interrupt has been deasserted.



7.7 Instruction Set Summary

Table 7-1 provides a summary of the M68HC08 instruction set.

Table 7-1. Instruction Set Summary (Sheet 1 of 7)

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 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 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 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 1 1 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 1 1 ff ff	4 1 1 4 3 5
BCC rel	Branch if Carry Bit Clear	$PC \leftarrow (PC) + 2 + rel ? (C) = 0$	-	-	-	-	-	-	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
BCS rel	Branch if Carry Bit Set (Same as BLO)	$PC \leftarrow (PC) + 2 + rel ? (C) = 1$	-	-	-	-	-	-	REL	25	rr	3
BEQ rel	Branch if Equal	$PC \leftarrow (PC) + 2 + rel ? (Z) = 1$	-	-	-	-	-	-	REL	27	rr	3

Freescale Semiconductor, Inc.

Table 7-1. Instruction Set Summary (Sheet 2 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
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 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
BRCLR <i>n,opr,rel</i>	Branch if Bit <i>n</i> in M Clear	$PC \leftarrow (PC) + 3 + rel ? (Mn) = 0$	-	-	-	-	-	†	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN <i>rel</i>	Branch Never	$PC \leftarrow (PC) + 2$	-	-	-	-	-	-	REL	21	rr	3

**Table 7-1. Instruction Set Summary (Sheet 3 of 7)**

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
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 <i>,X</i> 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
									IX1	6F	ff	3
									IX	7F		2
									SP1	9E6F	ff	4
CMP # <i>opr</i> CMP <i>opr</i> CMP <i>opr</i> CMP <i>opr,X</i> CMP <i>opr,X</i> CMP <i>,X</i> CMP <i>opr,SP</i> CMP <i>opr,SP</i>	Compare A with M	$(A) - (M)$	†						IMM	A1	ii	2
									DIR	B1	dd	3
									EXT	C1	hh ll	4
									IX2	D1	ee ff	4
									IX1	E1	ff	3
									IX	F1		2
									SP1	9EE1	ff	4
									SP2	9ED1	ee ff	5
COM <i>opr</i> COMA COM X COM <i>opr,X</i> COM <i>,X</i> COM <i>opr,SP</i>	Complement (One's Complement)	$M \leftarrow (\overline{M}) = \$FF - (M)$ $A \leftarrow (\overline{A}) = \$FF - (M)$ $X \leftarrow (\overline{X}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$							DIR	33	dd	4
									INH	43		1
									INH	53		1
									IX1	63	ff	4
									IX	73		3
									SP1	9E63	ff	5
CPHX # <i>opr</i> CPHX <i>opr</i>	Compare H:X with M	$(H:X) - (M:M + 1)$	†						IMM	65	ii ii+1	3
									DIR	75	dd	4

Table 7-1. Instruction Set Summary (Sheet 4 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
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) <sub>10</sub>	U	–	–	†	†	†	INH	72		2
DBNZ opr,rel DBNZA rel DBNZX rel DBNZ opr,X,rel DBNZ X,rel DBNZ opr,SP,rel	Decrement and Branch if Not Zero	A ← (A) – 1 or M ← (M) – 1 or X ← (X) – 1 PC ← (PC) + 3 + rel ? (result) ≠ 0 PC ← (PC) + 2 + rel ? (result) ≠ 0 PC ← (PC) + 2 + rel ? (result) ≠ 0 PC ← (PC) + 3 + rel ? (result) ≠ 0 PC ← (PC) + 2 + rel ? (result) ≠ 0 PC ← (PC) + 4 + rel ? (result) ≠ 0	–	–	–	–	–	–	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	5 3 3 5 4 6
DEC opr DECA DECX DEC opr,X DEC ,X DEC opr,SP	Decrement	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1 M ← (M) – 1	†	–	–	†	†	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff ff ff	4 1 1 4 3 5
DIV	Divide	A ← (H:A)/(X) H ← Remainder	–	–	–	–	†	†	INH	52		7
EOR #opr EOR opr EOR opr EOR opr,X EOR opr,X EOR ,X EOR opr,SP EOR opr,SP	Exclusive OR M with A	A ← (A ⊕ M)	0	–	–	†	†	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A8 B8 C8 D8 E8 F8 9EE8 9ED8	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
INC opr INCA INCX INC opr,X INC ,X INC opr,SP	Increment	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1 M ← (M) + 1	†	–	–	†	†	–	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff ff ff	4 1 1 4 3 5
JMP opr JMP opr JMP opr,X JMP opr,X JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff	2 3 4 3 2
JSR opr JSR opr JSR opr,X JSR opr,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Unconditional Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	4 5 6 5 4
LDA #opr LDA opr LDA opr LDA opr,X LDA opr,X LDA ,X LDA opr,SP LDA opr,SP	Load A from M	A ← (M)	0	–	–	†	†	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A6 B6 C6 D6 E6 F6 9EE6 9ED6	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5

Freescale Semiconductor, Inc.

Table 7-1. Instruction Set Summary (Sheet 5 of 7)

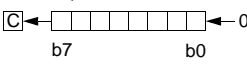
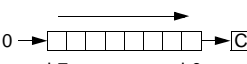
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
LDHX #opr LDHX opr	Load H:X from M	$H:X \leftarrow (M:M + 1)$	0	-	-	†	†	-	IMM DIR	45 55	ii jj dd	3 4
LDX #opr LDX opr LDX opr LDX opr,X LDX opr,X LDX ,X LDX opr,SP LDX opr,SP	Load X from M	$X \leftarrow (M)$	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	Logical Shift Left (Same as ASL)		†	-	-	†	†	†	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
LSR opr LSRA LSRX LSR opr,X LSR ,X LSR opr,SP	Logical Shift Right		†	-	-	0	†	†	DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff	4 1 1 4 3 5
MOV opr,opr MOV opr,X+ MOV #opr,opr MOV X+,opr	Move	$(M)_{\text{Destination}} \leftarrow (M)_{\text{Source}}$ $H:X \leftarrow (H:X) + 1$ (IX+D, DIX+)	0	-	-	†	†	-	DD DIX+ IMD IX+D	4E 5E 6E 7E	dd dd dd ii dd dd	5 4 4 4
MUL	Unsigned multiply	$X:A \leftarrow (X) \times (A)$	-	0	-	-	-	0	INH	42		5
NEG opr NEGA NEGX NEG opr,X NEG ,X NEG opr,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 #opr ORA opr ORA opr ORA opr,X ORA opr,X ORA ,X ORA opr,SP ORA opr,SP	Inclusive OR A and M	$A \leftarrow (A)   (M)$	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

Table 7-1. Instruction Set Summary (Sheet 6 of 7)

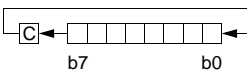
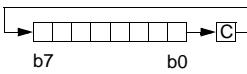
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ROL <i>opr</i> ROLA ROLX ROL <i>opr,X</i> ROL ,X ROL <i>opr,SP</i>	Rotate Left through Carry		↑	-	-	↑	↑	↑	DIR INH IX1 SP1	39 49 59 69 79 9E69	dd ff ff	4 1 1 4 3 5
ROR <i>opr</i> RORA RORX ROR <i>opr,X</i> ROR ,X ROR <i>opr,SP</i>	Rotate Right through Carry		↑	-	-	↑	↑	↑	DIR INH IX1 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 ,X 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 ,X 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 Interrupts, Stop Processing, Refer to MCU Documentation	I ← 0; Stop Processing	-	-	0	-	-	-	INH	8E		1
STX <i>opr</i> STX <i>opr</i> STX <i>opr,X</i> STX <i>opr,X</i> STX ,X 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
SUB # <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> SUB <i>opr,X</i> SUB <i>opr,X</i> SUB ,X SUB <i>opr,SP</i> SUB <i>opr,SP</i>	Subtract	A ← (A) - (M)	↑	-	-	↑	↑	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A0 B0 C0 D0 E0 F0 9EE0 9ED0	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5

Table 7-1. Instruction Set Summary (Sheet 7 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) - 1; Push (PCH) SP ← (SP) - 1; Push (X) SP ← (SP) - 1; Push (A) SP ← (SP) - 1; Push (CCR) SP ← (SP) - 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	-	-	1	-	-	-	INH	83		9
TAP	Transfer A to CCR	CCR ← (A)	†	†	†	†	†	†	INH	84		2
TAX	Transfer A to X	X ← (A)	-	-	-	-	-	-	INH	97		1
TPA	Transfer CCR to A	A ← (CCR)	-	-	-	-	-	-	INH	85		1
TST <i>opr</i> TSTA TSTX TST <i>opr,X</i> TST <i>,X</i> TST <i>opr,SP</i>	Test for Negative or Zero	(A) - \$00 or (X) - \$00 or (M) - \$00	0	-	-	†	†	-	DIR INH INH IX1 IX SP1	3D 4D 5D 6D 7D 9E6D	dd ff ff	3 1 1 3 2 4
TSX	Transfer SP to H:X	H:X ← (SP) + 1	-	-	-	-	-	-	INH	95		2
TXA	Transfer X to A	A ← (X)	-	-	-	-	-	-	INH	9F		1
TXS	Transfer H:X to SP	(SP) ← (H:X) - 1	-	-	-	-	-	-	INH	94		2
WAIT	Enable Interrupts; Wait for Interrupt	I bit ← 0; Inhibit CPU clocking until interrupted	-	-	0	-	-	-	INH	8F		1

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

7.8 Opcode Map

See [Table 7-2](#).

**Central Processor Unit (CPU)**

**Table 7-2. Opcode Map**

MSB LSB	Bit Manipulation			Branch			Read-Modify-Write			Control			Register/Memory								
	DIR	DIR	REL	DIR	DIR	REL	INH	INH	IX1	SP1	IX	INH	INH	IMM	DIR	EXT	IX2	SP2	IX1	SP1	IX
0	5 DIR	3 DIR	2	3	4	5	1 INH	4 INH	6 IX1	9E6	7	8	9	A	B	C	D	9ED	E	9EE	F
1	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	5 IX1	NEG SP1	3 IX	7 INH	3 INH	2 SUB	3 DIR	4 EXT	4 SUB	5 SUB	3 SUB	4 SUB	2 SUB
2	5 DIR	4 DIR	3 REL	4 REL	5 INH	7 INH	5 INH	3 INH	3 IX1	CBEQ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 CMP	5 CMP	3 CMP	4 CMP	2 CMP
3	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC
4	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC
5	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC
6	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC
7	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC
8	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC
9	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC
A	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC
B	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC
C	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC
D	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC
E	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC
F	5 DIR	4 DIR	3 REL	4 REL	1 INH	4 INH	1 INH	4 INH	3 IX1	DBNZ SP1	4 IX+	4 INH	2 INH	2 SBC	3 DIR	4 EXT	4 SBC	5 SBC	3 SBC	4 SBC	2 SBC

INH Inherent  
 IMM Immediate  
 DIR Direct  
 EXT Extended  
 DD Direct-Direct  
 IX+D Indexed-Direct  
 \*Pre-byte for stack pointer indexed instructions

REL Relative  
 IX Indexed, No Offset  
 IX1 Indexed, 8-Bit Offset  
 IX2 Indexed, 16-Bit Offset  
 Post Increment  
 IX1+ Indexed, 1-Byte Offset with Post Increment

SP1 Stack Pointer, 8-Bit Offset  
 SP2 Stack Pointer, 16-Bit Offset  
 Indexed, No Offset with Post Increment

High Byte of Opcode in Hexadecimal  
 Low Byte of Opcode in Hexadecimal  
 Cycles  
 Opcode Mnemonic  
 Number of Bytes / Addressing Mode

MSB	0
LSB	5
MSB	0
LSB	BRSET0
	3 DIR



### Section 8. External Interrupt (IRQ)

#### 8.1 Introduction

The  $\overline{\text{IRQ}}$  pin (external interrupt), shared with PTA2 (general purpose input) and keyboard interrupt (KBI), provides a maskable interrupt input.

#### 8.2 Features

Features of the IRQ module include the following:

- External interrupt pin,  $\overline{\text{IRQ}}$
- $\overline{\text{IRQ}}$  interrupt control bits
- Hysteresis buffer
- Programmable edge-only or edge and level interrupt sensitivity
- Automatic interrupt acknowledge
- Selectable internal pullup resistor

#### 8.3 Functional Description

$\overline{\text{IRQ}}$  pin functionality is enabled by setting configuration register 2 (CONFIG2) IRQEN bit accordingly. A zero disables the IRQ function and  $\overline{\text{IRQ}}$  will assume the other shared functionalities. A one enables the IRQ function.

A falling edge on the external interrupt pin can latch a central processor unit (CPU) interrupt request. **Figure 8-2** shows the structure of the IRQ module.

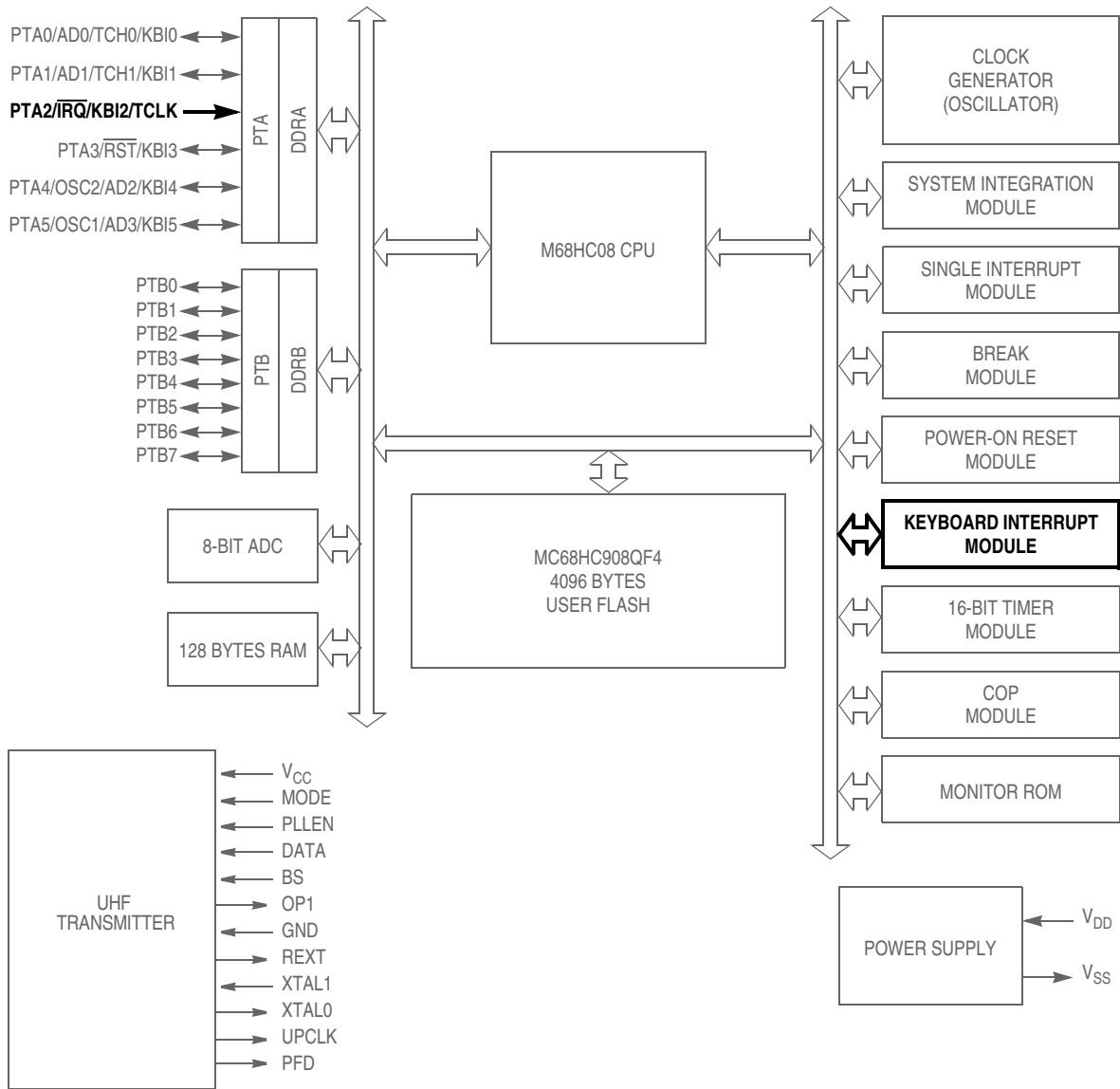
Interrupt signals on the  $\overline{\text{IRQ}}$  pin are latched into the IRQ latch. An interrupt latch remains set until one of the following actions occurs:

- Vector fetch — A vector fetch automatically generates an interrupt acknowledge signal that clears the IRQ latch.
- Software clear — Software can clear the interrupt latch by writing to the acknowledge bit in the interrupt status and control register (INTSCR). Writing a 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 out of reset 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 the interrupt pin is edge-triggered only (MODE = 0), the CPU interrupt request remains set until a vector fetch, software clear, or reset occurs.

External Interrupt (IRQ)



**RST, IRQ:** Pins have internal (about 30K Ohms) pull up  
 PTA[0:5]: High current sink and source capability  
 PTA[0:5]: Pins have programmable keyboard interrupt and pull up

Figure 8-1. Block Diagram Highlighting IRQ Block and Pins

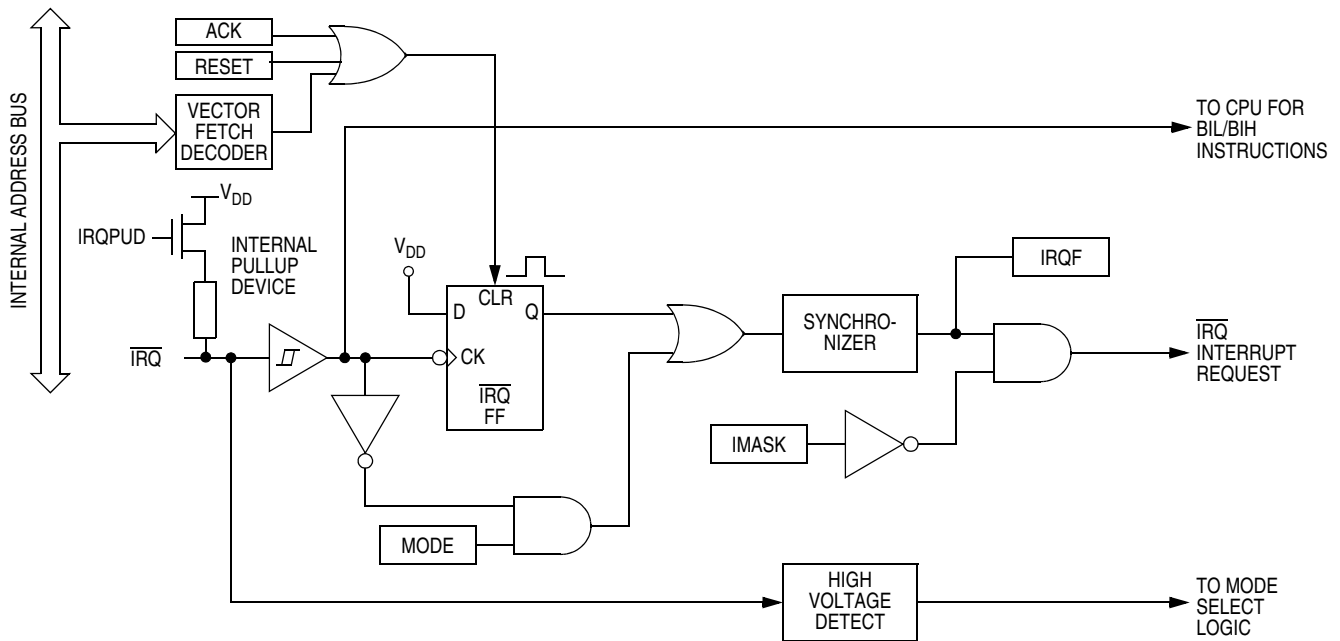


Figure 8-2. IRQ Module Block Diagram

When the interrupt pin is both falling-edge and low-level triggered (MODE = 1), the CPU interrupt request 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. See [14.6 Exception Control](#).

Figure 8-3 provides a summary of the IRQ I/O register.

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0	
\$001D	IRQ Status and Control Register (INTSCR) <i>See page 77.</i>	Read:	0	0	0	0	IRQF	0	IMASK	MODE
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 8-3. IRQ I/O Register Summary

## 8.4 $\overline{\text{IRQ}}$ Pin

A falling edge 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 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 latches 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.

**NOTE:** When the  $\overline{\text{IRQ}}$  function is enabled in the CONFIG2 register, the BIH and BIL instructions can be used to read the logic level on the  $\overline{\text{IRQ}}$  pin. If the  $\overline{\text{IRQ}}$  function is disabled, these instructions will behave as if the  $\overline{\text{IRQ}}$  pin is a logic 1, regardless of the actual level on the pin. Conversely, when the  $\overline{\text{IRQ}}$  function is enabled, bit 2 of the port A data register will always read a 0.

**NOTE:** When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine. An internal pullup resistor to  $V_{DD}$  is connected to the  $\overline{\text{IRQ}}$  pin; this can be disabled by setting the IRQPUD bit in the CONFIG2 register (\$001E).

## 8.5 IRQ Module During Break Interrupts

The system integration module (SIM) controls whether the IRQ latch can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear the latches during the break state. See [Section 14. System Integration Module \(SIM\)](#).

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

To protect the latches during the break state, write a 0 to the BCFE bit. With BCFE at 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 latch.

## 8.6 IRQ Status and Control Register


The IRQ status and control register (ISCR) controls and monitors operation of the IRQ module, see [Section 5. Configuration Register \(CONFIG\)](#).

The ISCR has the following functions:

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

Address: \$001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	IRQF		IMASK	MODE
Write:						ACK		
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

**Figure 8-4. IRQ Status and Control Register (INTSCR)**

**IRQF — IRQ Flag**

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

**IMASK — IRQ Interrupt Mask Bit**

Writing a 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 9. Keyboard Interrupt Module (KBI)

### 9.1 Introduction

The keyboard interrupt module (KBI) provides six independently maskable external interrupts, which are accessible via the PTA0–PTA5 pins.

### 9.2 Features

Features of the keyboard interrupt module include:

- Six keyboard interrupt pins with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Software configurable pullup device if input pin is configured as input port bit
- Programmable edge-only or edge and level interrupt sensitivity
- Exit from low-power modes

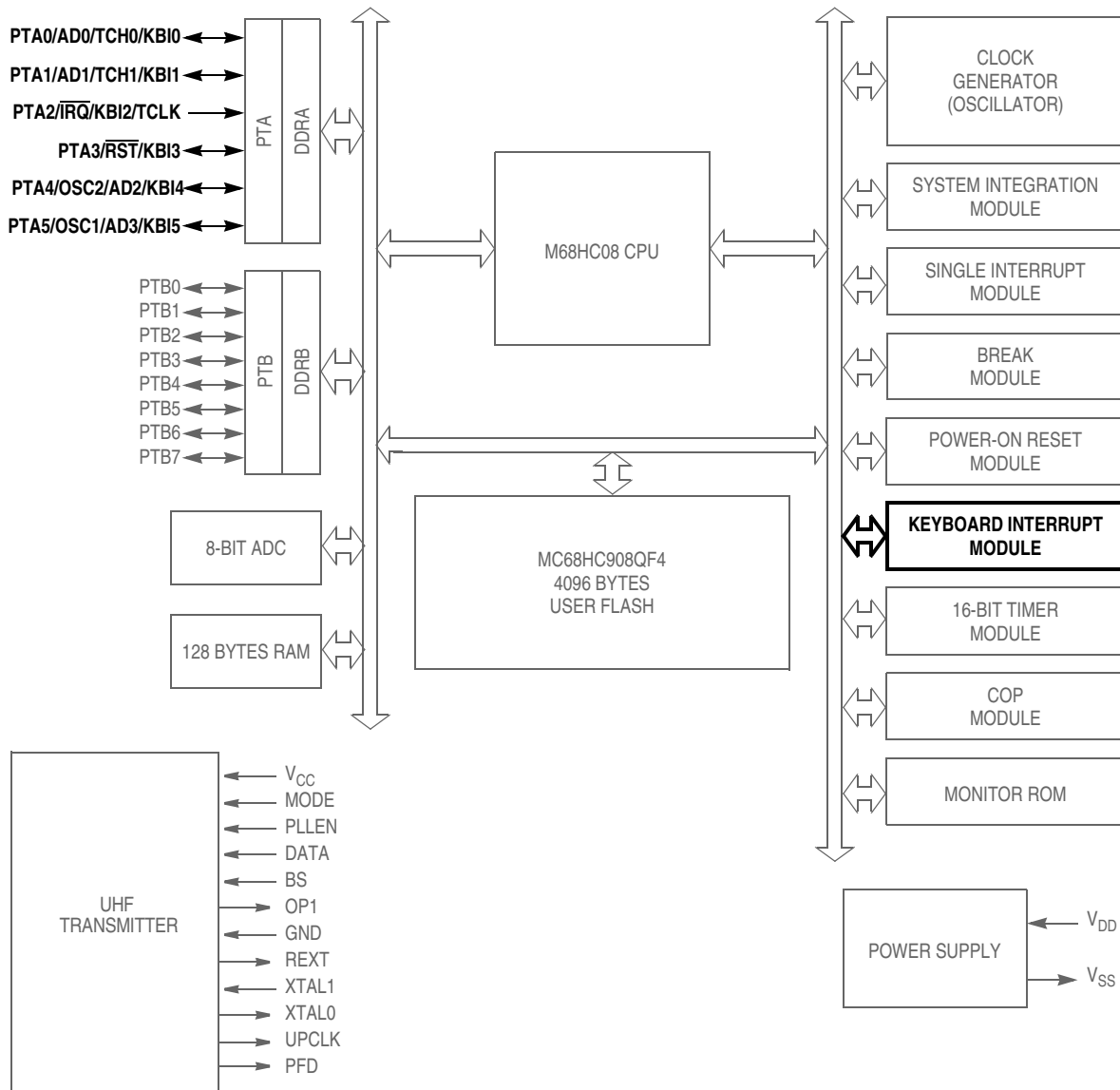
Figure 9-1 provides a summary of the input/output (I/O) registers

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0	
\$001A	Keyboard Status and Control Register (KBSCR) <a href="#">See page 84.</a>	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:	[Unimplemented]							
		Reset:	0	0	0	0	0	0	0	0
\$001B	Keyboard Interrupt Enable Register (KBIER) <a href="#">See page 85.</a>	Read:	0	AWUIE	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:	[Unimplemented]							
		Reset:	0	0	0	0	0	0	0	0

[Unimplemented] = Unimplemented

Figure 9-1. KBI I/O Register Summary

Keyboard Interrupt Module (KBI)



$\overline{RST}$ ,  $\overline{IRQ}$ : Pins have internal (about 30K Ohms) pull up  
 PTA[0:5]: High current sink and source capability  
 PTA[0:5]: Pins have programmable keyboard interrupt and pull up

Figure 9-2. Block Diagram Highlighting KBI Block and Pins



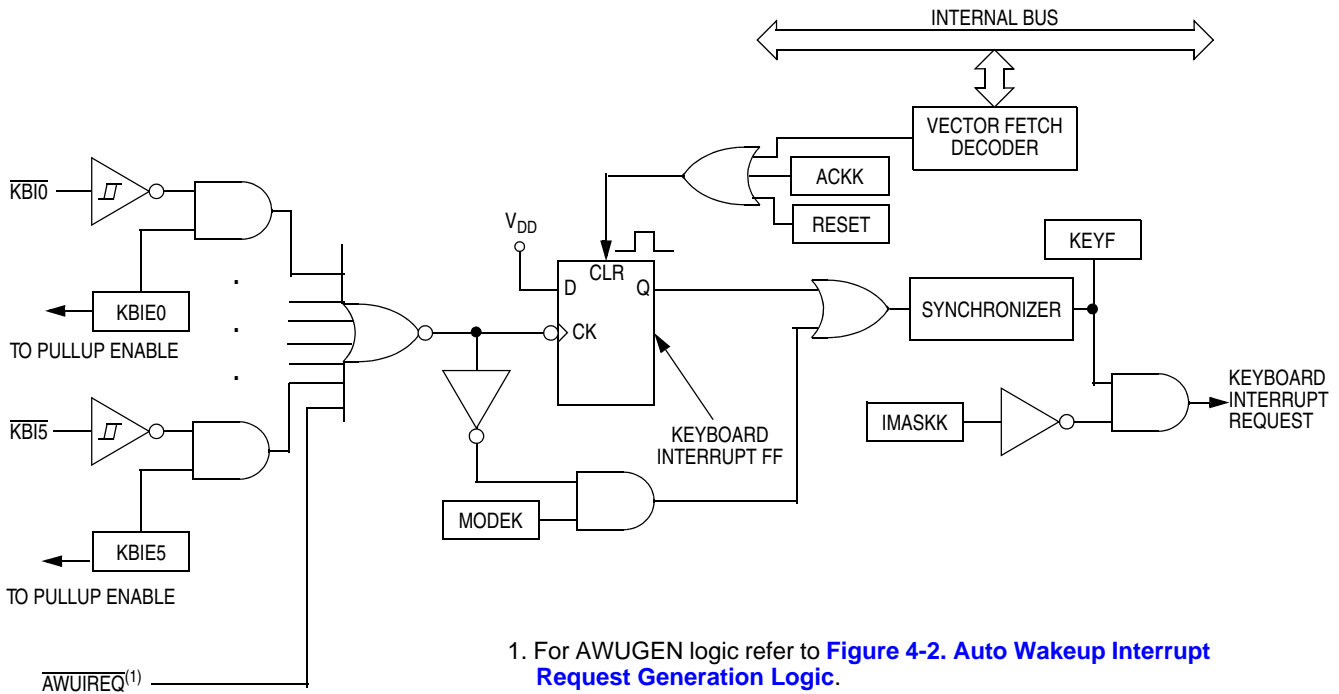


Figure 9-3. Keyboard Interrupt Block Diagram

### 9.3 Functional Description

The keyboard interrupt module controls the enabling/disabling of interrupt functions on the six port A pins. These six pins can be enabled/disabled independently of each other.

#### 9.3.1 Keyboard Operation

Writing to the KBIE0–KBIE5 bits in the keyboard interrupt enable register (KBIER) independently enables or disables each port A pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin in port A also enables its internal pullup device irrespective of PTAPUE<sub>x</sub> bits in the port A input pullup enable register (see [13.2.3 Port A Input Pullup Enable Register](#)). A logic 0 applied to an enabled keyboard interrupt pin latches a keyboard interrupt request.

A keyboard interrupt is latched when one or more keyboard interrupt inputs goes low after all were high. The MODEK bit in the keyboard status and control register controls the triggering mode of the keyboard interrupt.

- If the keyboard interrupt is edge-sensitive only, a falling edge on a keyboard interrupt input does not latch an interrupt request if another keyboard pin is already low. To prevent losing an interrupt request on one input because another input is still low, software can disable the latter input while it is low.
- If the keyboard interrupt is falling edge and low-level sensitive, an interrupt request is present as long as any keyboard interrupt input is low.

If the MODEK bit is set, the keyboard interrupt inputs are both falling edge and low-level sensitive, and both of the following actions must occur to clear a keyboard interrupt request:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the interrupt request. Software may generate the interrupt acknowledge signal by writing a 1 to the ACKK bit in the keyboard status and control register (KBSCR). The ACKK bit is useful in applications that poll the keyboard interrupt inputs and require software to clear the keyboard interrupt request. Writing to the ACKK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACKK does not affect subsequent transitions on the keyboard interrupt inputs. A falling edge that occurs after writing to the ACKK bit latches another interrupt request. If the keyboard interrupt mask bit, IMASKK, is clear, the central processor unit (CPU) loads the program counter with the vector address at locations \$FFE0 and \$FFE1.
- Return of all enabled keyboard interrupt inputs to logic 1 — As long as any enabled keyboard interrupt pin is at logic 0, the keyboard interrupt remains set. The auto wakeup interrupt input, AWUIREQ, will be cleared only by writing to ACKK bit in KBSCR or reset.

The vector fetch or software clear and the return of all enabled keyboard interrupt pins to logic 1 may occur in any order.

If the MODEK bit is clear, the keyboard interrupt pin is falling-edge sensitive only. With MODEK clear, a vector fetch or software clear immediately clears the keyboard interrupt request.

Reset clears the keyboard interrupt request and the MODEK bit, clearing the interrupt request even if a keyboard interrupt input stays at logic 0.

The keyboard flag bit (KEYF) in the keyboard status and control register can be used to see if a pending interrupt exists. The KEYF bit is not affected by the keyboard interrupt mask bit (IMASKK) which makes it useful in applications where polling is preferred.

To determine the logic level on a keyboard interrupt pin, use the data direction register to configure the pin as an input and then read the data register.

**NOTE:** *Setting a keyboard interrupt enable bit (KBIE<sub>x</sub>) forces the corresponding keyboard interrupt pin to be an input, overriding the data direction register. However, the data direction register bit must be a 0 for software to read the pin.*

### 9.3.2 Keyboard Initialization

When a keyboard interrupt pin is enabled, it takes time for the internal pullup to reach a logic 1. Therefore a false interrupt can occur as soon as the pin is enabled.

To prevent a false interrupt on keyboard initialization:

1. Mask keyboard interrupts by setting the IMASKK bit in the keyboard status and control register.
2. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.
3. Write to the ACKK bit in the keyboard status and control register to clear any false interrupts.
4. Clear the IMASKK bit.

An interrupt signal on an edge-triggered pin can be acknowledged immediately after enabling the pin. An interrupt signal on an edge- and level-triggered interrupt pin must be acknowledged after a delay that depends on the external load.

Another way to avoid a false interrupt:

1. Configure the keyboard pins as outputs by setting the appropriate DDRA bits in the data direction register A.
2. Write 1s to the appropriate port A data register bits.
3. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.

### 9.4 Wait Mode

The keyboard module remains active in wait mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of wait mode.

### 9.5 Stop Mode

The keyboard module remains active in stop mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of stop mode.

### 9.6 Keyboard Module During Break Interrupts

The system integration module (SIM) controls whether the keyboard interrupt latch can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state.

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

To protect the latch during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), writing to the keyboard acknowledge bit (ACKK) in the keyboard status and control register during the break state has no effect.

### 9.7 Input/Output Registers

The following I/O registers control and monitor operation of the keyboard interrupt module:

- Keyboard interrupt status and control register (KBSCR)
- Keyboard interrupt enable register (KBIER)


#### 9.7.1 Keyboard Status and Control Register

The keyboard status and control register (KBSCR):

- Flags keyboard interrupt requests
- Acknowledges keyboard interrupt requests
- Masks keyboard interrupt requests
- Controls keyboard interrupt triggering sensitivity

Address: \$001A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
Write:						ACKK		
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

**Figure 9-4. Keyboard Status and Control Register (KBSCR)**

Bits 7–4 — Not used

These read-only bits always read as 0s.

**KEYF** — Keyboard Flag Bit

This read-only bit is set when a keyboard interrupt is pending on port A or auto wakeup. Reset clears the KEYF bit.

- 1 = Keyboard interrupt pending
- 0 = No keyboard interrupt pending

**ACKK** — Keyboard Acknowledge Bit

Writing a 1 to this write-only bit clears the keyboard interrupt request on port A and auto wakeup logic. ACKK always reads as 0. Reset clears ACKK.

**IMASKK— Keyboard Interrupt Mask Bit**

Writing a 1 to this read/write bit prevents the output of the keyboard interrupt mask from generating interrupt requests on port A or auto wakeup. Reset clears the IMASKK bit.

- 1 = Keyboard interrupt requests masked
- 0 = Keyboard interrupt requests not masked

**MODEK — Keyboard Triggering Sensitivity Bit**

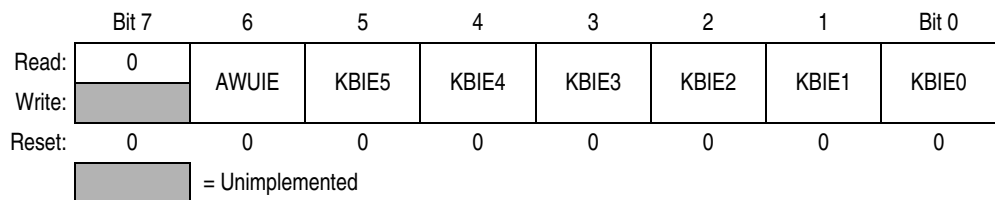
This read/write bit controls the triggering sensitivity of the keyboard interrupt pins on port A and auto wakeup. Reset clears MODEK.

- 1 = Keyboard interrupt requests on falling edges and low levels
- 0 = Keyboard interrupt requests on falling edges only

**9.7.2 Keyboard Interrupt Enable Register**

The port A keyboard interrupt enable register (KBIER) enables or disables each port A pin or auto wakeup to operate as a keyboard interrupt input.

Address: \$001B



**Figure 9-5. Keyboard Interrupt Enable Register (KBIER)**

**KBIE5–KBIE0 — Port A Keyboard Interrupt Enable Bits**

Each of these read/write bits enables the corresponding keyboard interrupt pin on port A to latch interrupt requests. Reset clears the keyboard interrupt enable register.

- 1 = KBIx pin enabled as keyboard interrupt pin
- 0 = KBIx pin not enabled as keyboard interrupt pin

**NOTE:** *AWUIE bit is not used in conjunction with the keyboard interrupt feature. To see a description of this bit, see [Section 4. Auto Wakeup Module \(AWU\)](#).*



## Section 10. Low-Voltage Inhibit (LVI)

### 10.1 Introduction

This section describes the low-voltage inhibit (LVI) module, which monitors the voltage on the  $V_{DD}$  pin and can force a reset when the  $V_{DD}$  voltage falls below the LVI trip falling voltage,  $V_{TRIPF}$ .

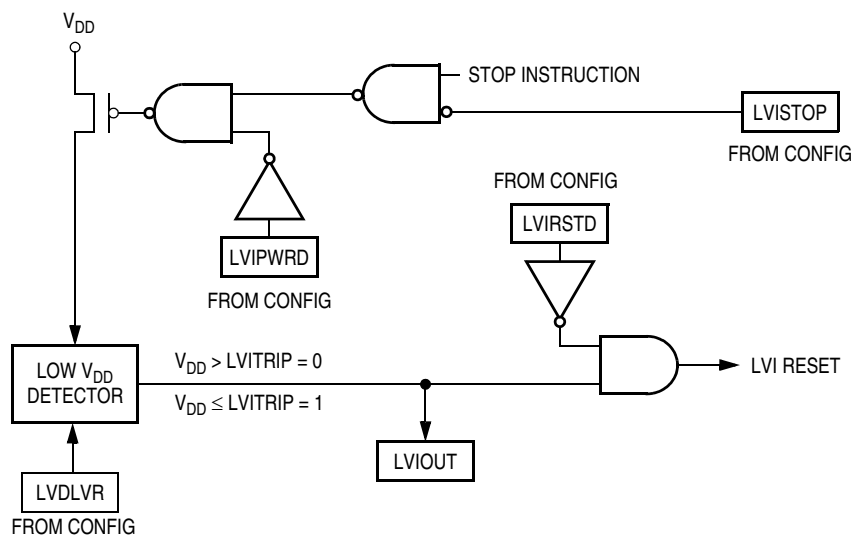
### 10.2 Features

Features of the LVI module include:

- Programmable LVI reset
- Programmable power consumption
- Selectable LVI trip voltage
- Programmable stop mode operation

### 10.3 Functional Description

**Figure 10-1** shows the structure of the LVI module. LVISTOP, LVIPWRD, LVDLVR, and LVIRSTD are user selectable options found in the configuration register (CONFIG1). See **Section 5. Configuration Register (CONFIG)**.



**Figure 10-1. LVI Module Block Diagram**

The LVI is enabled out of reset. The LVI module contains a bandgap reference circuit and comparator. Clearing the LVI power disable bit (LVIPWRD) enables the LVI to monitor  $V_{DD}$  voltage. Clearing the LVI reset disable bit (LVIRSTD) enables the LVI module to generate a reset when  $V_{DD}$  falls below a voltage,  $V_{TRIPF}$  or  $V_{DTRIPF}$ . Setting the LVI enable in stop mode bit (LVISTOP) enables the LVI to operate in stop mode. Setting the LVD or LVR trip point bit (LVDLVR) selects the LVD trip point voltage. The actual trip thresholds are specified in [17.5 DC Electrical Characteristics](#). Either trip level can be used as a detect or reset.

**NOTE:** After a power-on reset, the LVI's default mode of operation is LVR trip voltage. If a higher trip voltage is desired, the user must set the LVDLVR bit to raise the trip point to the LVD voltage.

If the user requires the higher trip voltage and sets the LVDLVR bit after power-on reset while the VDD supply is not above the  $V_{TRIPR}$  for LVD mode, the microcontroller unit (MCU) will immediately go into reset. The next time the LVI releases the reset, the supply will be above the  $V_{TRIPR}$  for LVD mode.

Once an LVI reset occurs, the MCU remains in reset until  $V_{DD}$  rises above a voltage,  $V_{TRIPR}$ , which causes the MCU to exit reset. See [Section 14. System Integration Module \(SIM\)](#) for the reset recovery sequence.

The output of the comparator controls the state of the LVIOOUT flag in the LVI status register (LVISR) and can be used for polling LVI operation when the LVI reset is disabled.

### 10.3.1 Polled LVI Operation

In applications that can operate at  $V_{DD}$  levels below the  $V_{TRIPF}$  level, software can monitor  $V_{DD}$  by polling the LVIOOUT bit. In the configuration register, the LVIPWRD bit must be cleared to enable the LVI module, and the LVIRSTD bit must be set to disable LVI resets.

### 10.3.2 Forced Reset Operation

In applications that require  $V_{DD}$  to remain above the  $V_{TRIPF}$  level, enabling LVI resets allows the LVI module to reset the MCU when  $V_{DD}$  falls below the  $V_{TRIPF}$  level. In the configuration register, the LVIPWRD and LVIRSTD bits must be cleared to enable the LVI module and to enable LVI resets.

### 10.3.3 Voltage Hysteresis Protection

Once the LVI has triggered (by having  $V_{DD}$  fall below  $V_{TRIPF}$ ), the LVI will maintain a reset condition until  $V_{DD}$  rises above the rising trip point voltage,  $V_{TRIPR}$ . This prevents a condition in which the MCU is continually entering and exiting reset if  $V_{DD}$  is approximately equal to  $V_{TRIPF}$ .  $V_{TRIPR}$  is greater than  $V_{TRIPF}$  by the hysteresis voltage,  $V_{HYS}$ .



### 10.3.4 LVI Trip Selection

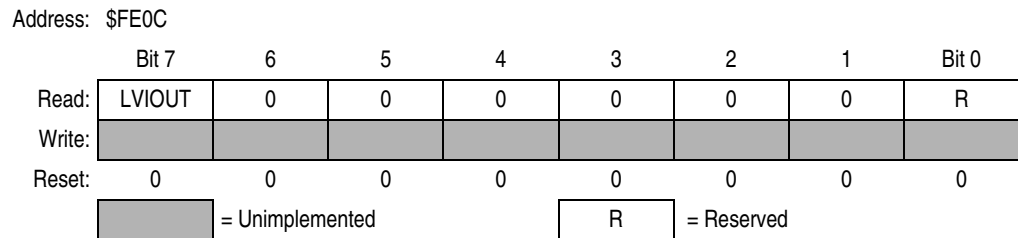
The LVDLVR bit in the configuration register selects whether the LVI is configured for LVD (low voltage detect) or LVR (low voltage reset) protection. The LVD trip voltage can be used as a low voltage warning. The LVR trip voltage will commonly be configured as a reset condition since it is very close to the minimum operating voltage of the device. The LVDLVR bit can be written to anytime so that battery applications can make use of the LVI as both a warning indicator and to generate a system reset.

Polling and forced reset operation modes can be combined to take full advantage of LVD and LVR trip voltages selection. LVD (LVDLVR = 1) in polling mode (LVIRSTD = 1) can be used as a low voltage warning in a slowly and continuously falling  $V_{DD}$  application (for example, battery applications). Once LVD has been identified, the part can be set to LVR (LVDLVR = 0) and reset enabled (LVIRSTD = 0). So, as  $V_{DD}$  continues to fall the part will reset when LVR trip voltage is reached. Unlike other bits in CONFIG registers, LVIRSTD and LVDLVR bits are allowed to be written multiple times after reset.

**NOTE:** *The microcontroller is guaranteed to operate at a minimum supply voltage. The trip point ( $V_{TRIPF}$  [LVD] or  $V_{TRIPF}$  [LVR]) may be lower than this. See [17.5 DC Electrical Characteristics](#) for the actual trip point voltages.*

### 10.4 LVI Status Register

The LVI status register (LVISR) indicates if the  $V_{DD}$  voltage was detected below the  $V_{TRIPF}$  level while LVI resets have been disabled.



**Figure 10-2. LVI Status Register (LVISR)**

#### LVIOUT — LVI Output Bit

This read-only flag becomes set when the  $V_{DD}$  voltage falls below the  $V_{TRIPF}$  trip voltage and is cleared when  $V_{DD}$  voltage rises above  $V_{TRIPR}$ . The difference in these threshold levels results in a hysteresis that prevents oscillation into and out of reset (see [Table 10-1](#)). Reset clears the LVIOUT bit.

Table 10-1. LVIOUT Bit Indication

V <sub>DD</sub>	LVIOUT
V <sub>DD</sub> > V <sub>TRIPR</sub>	0
V <sub>DD</sub> < V <sub>TRIPF</sub>	1
V <sub>TRIPF</sub> < V <sub>DD</sub> < V <sub>TRIPR</sub>	Previous value

### 10.5 LVI Interrupts

The LVI module does not generate interrupt requests.

### 10.6 Low-Power Modes

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

#### 10.6.1 Wait Mode

If enabled, the LVI module remains active in wait mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of wait mode.

#### 10.6.2 Stop Mode

When the LVIPWRD bit in the configuration register is cleared and the LVISTOP bit in the configuration register is set, the LVI module remains active in stop mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of stop mode.

### Section 11. Oscillator Module (OSC)

#### 11.1 Introduction

The oscillator module is used to provide a stable clock source for the microcontroller system and bus. The oscillator module generates two output clocks, BUSCLKX2 and BUSCLKX4. The BUSCLKX4 clock is used by the system integration module (SIM) and the computer operating properly module (COP). The BUSCLKX2 clock is divided by two in the SIM to be used as the bus clock for the microcontroller. Therefore the bus frequency will be one fourth of the BUSCLKX4 frequency.

#### 11.2 Features

The oscillator has these four clock source options available:

1. Internal oscillator: An internally generated, fixed frequency clock, trimmable to  $\pm 5\%$ . This is the default option out of reset.
2. External oscillator: An external clock that can be driven directly into OSC1.
3. External RC: A built-in oscillator module (RC oscillator) that requires an external R connection only. The capacitor is internal to the chip.
4. External crystal: A built-in oscillator module (XTAL oscillator) that requires an external crystal or ceramic-resonator.

#### 11.3 Functional Description

The oscillator contains these major subsystems:

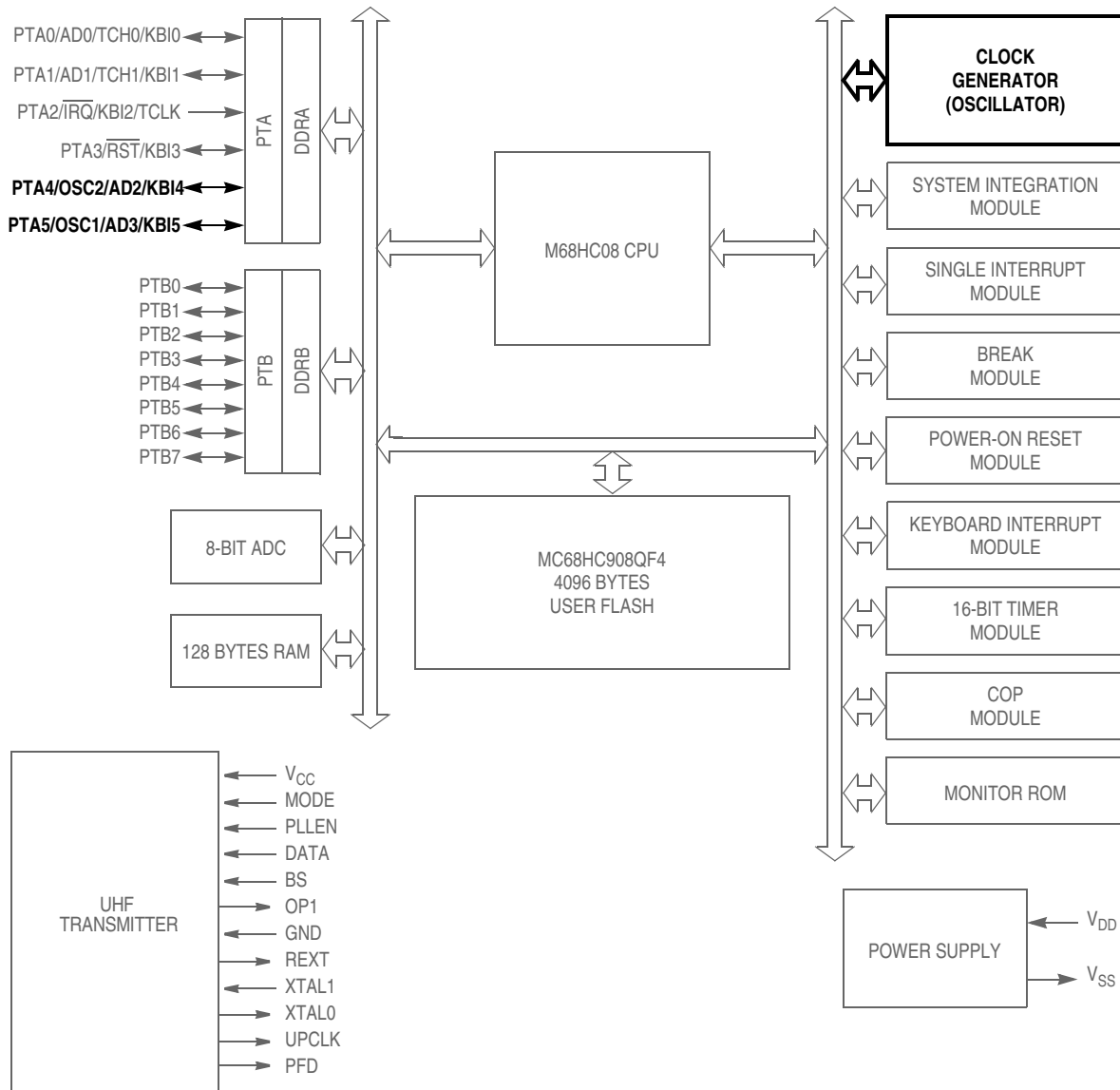
- Internal oscillator circuit
- Internal or external clock switch control
- External clock circuit
- External crystal circuit
- External RC clock circuit

##### 11.3.1 Internal Oscillator

The internal oscillator circuit is designed for use with no external components to provide a clock source with tolerance less than  $\pm 25\%$  untrimmed. An 8-bit trimming register allows adjustment to a tolerance of less than  $\pm 5\%$ .

The internal oscillator will generate a clock of 4.0 MHz typical (INTCLK) resulting in a bus speed (internal clock  $\div$  4) of 1.0 MHz.

Oscillator Module (OSC)



$\overline{RST}$ ,  $\overline{IRQ}$ : Pins have internal (about 30K Ohms) pull up  
**PTA[0:5]: High current sink and source capability**  
**PTA[0:5]: Pins have programmable keyboard interrupt and pull up**

Figure 11-1. Block Diagram Highlighting OSC Block and Pins

**Figure 11-3** shows how BUSCLKX4 is derived from INTCLK and, like the RC oscillator, OSC2 can output BUSCLKX4 by setting OSC2EN in PTAPUE register. See **Section 13. Input/Output (I/O) Ports**.

#### 11.3.1.1 Internal Oscillator Trimming

The 8-bit trimming register, OSCTRIM, allows a clock period adjust of +127 and –128 steps. Increasing OSCTRIM value increases the clock period. Trimming allows the internal clock frequency to be set to 4.0 MHz  $\pm$ 5%.

All devices are programmed with a trim value in a reserved FLASH location, \$FFC0. This value can be copied from the FLASH to the OSCTRIM register (\$0038) during reset initialization.

Reset loads OSCTRIM with a default value of \$80.

**WARNING:** *Bulk FLASH erasure will set location \$FFC0 to \$FF and the factory programmed value will be lost.*

#### 11.3.1.2 Internal to External Clock Switching

When external clock source (external OSC, RC, or XTAL) is desired, the user must perform the following steps:

1. For external crystal circuits only, OSCOPT[1:0] = 1:1: To help precharge an external crystal oscillator, set PTA4 (OSC2) as an output and drive high for several cycles. This may help the crystal circuit start more robustly.
2. Set CONFIG2 bits OSCOPT[1:0] according to **11.7 CONFIG2 Options**. The oscillator module control logic will then set OSC1 as an external clock input and, if the external crystal option is selected, OSC2 will also be set as the clock output.
3. Create a software delay to wait the stabilization time needed for the selected clock source (crystal, resonator, RC) as recommended by the component manufacturer. A good rule of thumb for crystal oscillators is to wait 4096 cycles of the crystal frequency, i.e., for a 4-MHz crystal, wait approximately 1 msec.
4. After the manufacturer's recommended delay has elapsed, the ECGON bit in the OSC status register (OSCSTAT) needs to be set by the user software.
5. After ECGON set is detected, the OSC module checks for oscillator activity by waiting two external clock rising edges.
6. The OSC module then switches to the external clock. Logic provides a glitch free transition.
7. The OSC module first sets the ECGST bit in the OSCSTAT register and then stops the internal oscillator.

**NOTE:** *Once transition to the external clock is done, the internal oscillator will only be reactivated with reset. No post-switch clock monitor feature is implemented (clock does not switch back to internal if external clock dies).*

### 11.3.2 External Oscillator

The external clock option is designed for use when a clock signal is available in the application to provide a clock source to the microcontroller. The OSC1 pin is enabled as an input by the oscillator module. The clock signal is used directly to create BUSCLKX4 and also divided by two to create BUSCLKX2.

In this configuration, the OSC2 pin cannot output BUSCLKX4. So the OSC2EN bit in the port A pullup enable register will be clear to enable PTA4 I/O functions on the pin.

### 11.3.3 XTAL Oscillator

The XTAL oscillator circuit is designed for use with an external low-frequency crystal or ceramic resonator to provide an accurate clock source. In this configuration, the OSC2 pin is dedicated to the external crystal circuit. The OSC2EN bit in the port A pullup enable register has no effect when this clock mode is selected.

In its typical configuration, the XTAL oscillator is connected in a Pierce oscillator configuration, as shown in [Figure 11-2](#). 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,  $R_S$

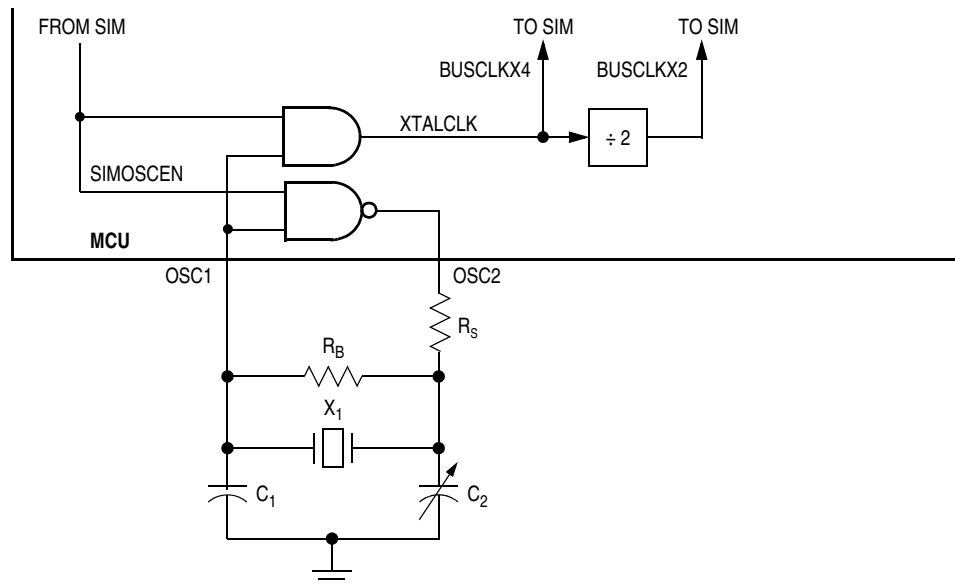


Figure 11-2. XTAL Oscillator External Connections

### 11.3.4 RC Oscillator

The RC oscillator circuit is designed for use with external R to provide a clock source with tolerance less than 25%.

In its typical configuration, the RC oscillator requires two external components, one R and one C. In the MC68HLC908QF4, the capacitor is internal to the chip. The R value should have a tolerance of 1% or less, to obtain a clock source with less than 25% tolerance. The oscillator configuration uses one component,  $R_{EXT}$ .

In this configuration, the OSC2 pin can be left in the reset state as PTA4. Or, the OSC2EN bit in the port A pullup enable register can be set to enable the OSC2 output function on the pin. Enabling the OSC2 output slightly increases the external RC oscillator frequency,  $f_{RCCLK}$ .

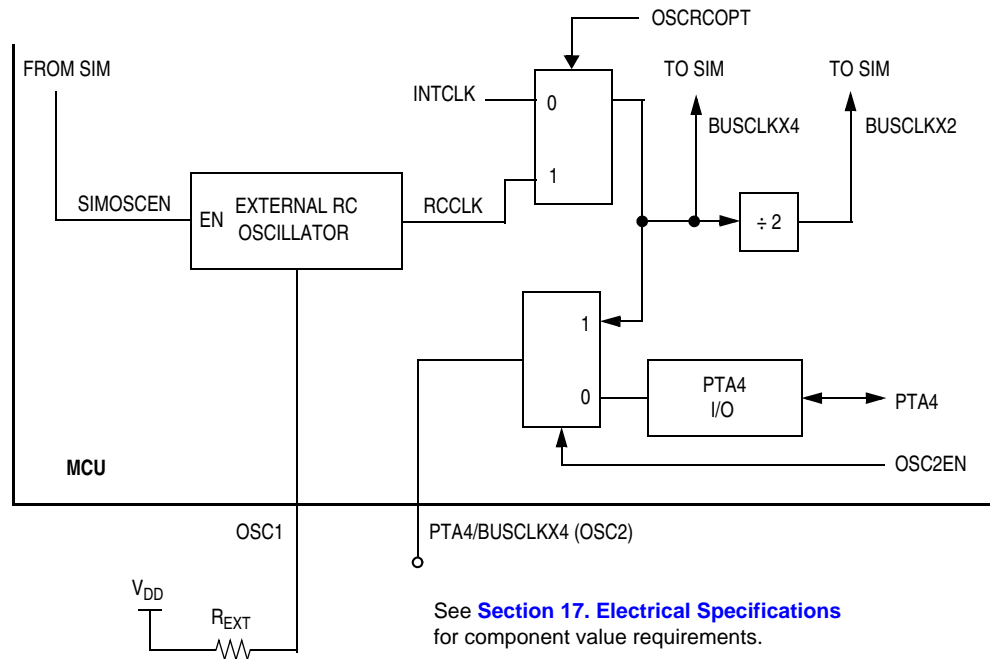


Figure 11-3. RC Oscillator External Connections

## 11.4 Oscillator Module Signals

The following paragraphs describe the signals that are inputs to and outputs from the oscillator module.

### 11.4.1 Crystal Amplifier Input Pin (OSC1)

The OSC1 pin is either an input to the crystal oscillator amplifier, an input to the RC oscillator circuit, or an external clock source.

For the internal oscillator configuration, the OSC1 pin can assume other functions according to [Table 1-1. Pin Functions](#).

#### 11.4.2 Crystal Amplifier Output Pin (OSC2/PTA4/BUSCLKX4)

For the XTAL oscillator device, the OSC2 pin is the crystal oscillator inverting amplifier output.

For the external clock option, the OSC2 pin is dedicated to the PTA4 I/O function. The OSC2EN bit has no effect.

For the internal oscillator or RC oscillator options, the OSC2 pin can assume other functions according to [Table 1-1. Pin Functions](#), or the output of the oscillator clock (BUSCLKX4).

**Table 11-1. OSC2 Pin Function**

Option	OSC2 Pin Function
XTAL oscillator	Inverting OSC1
External clock	PTA4 I/O
Internal oscillator or RC oscillator	Controlled by OSC2EN bit in PTAPUE register OSC2EN = 0: PTA4 I/O OSC2EN = 1: BUSCLKX4 output

#### 11.4.3 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal comes from the system integration module (SIM) and enables/disables either the XTAL oscillator circuit, the RC oscillator, or the internal oscillator.

#### 11.4.4 XTAL Oscillator Clock (XTALCLK)

XTALCLK is the XTAL oscillator output signal. It runs at the full speed of the crystal ( $f_{XCLK}$ ) and comes directly from the crystal oscillator circuit. [Figure 11-2](#) shows only the logical relation of XTALCLK to OSC1 and OSC2 and may not represent the actual circuitry. The duty cycle of XTALCLK is unknown and may depend on the crystal and other external factors. Also, the frequency and amplitude of XTALCLK can be unstable at start up.

#### 11.4.5 RC Oscillator Clock (RCCLK)

RCCLK is the RC oscillator output signal. Its frequency is directly proportional to the time constant of external R and internal C. [Figure 11-3](#) shows only the logical relation of RCCLK to OSC1 and may not represent the actual circuitry.



#### 11.4.6 Internal Oscillator Clock (INTCLK)

INTCLK is the internal oscillator output signal. Its nominal frequency is fixed to 4.0 MHz, but it can be also trimmed using the oscillator trimming feature of the OSCTRIM register (see [11.3.1.1 Internal Oscillator Trimming](#)).

#### 11.4.7 Oscillator Out 2 (BUSCLKX4)

BUSCLKX4 is the same as the input clock (XTALCLK, RCCLK, or INTCLK). This signal is driven to the SIM module and is used to determine the COP cycles.

#### 11.4.8 Oscillator Out (BUSCLKX2)

The frequency of this signal is equal to half of the BUSCLKX4, this signal is driven to the SIM for generation of the bus clocks used by the CPU and other modules on the MCU. BUSCLKX2 will be divided again in the SIM and results in the internal bus frequency being one fourth of either the XTALCLK, RCCLK, or INTCLK frequency.

### 11.5 Low Power Modes

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

#### 11.5.1 Wait Mode

The WAIT instruction has no effect on the oscillator logic. BUSCLKX2 and BUSCLKX4 continue to drive to the SIM module.

#### 11.5.2 Stop Mode

The STOP instruction disables either the XTALCLK, the RCCLK, or INTCLK output, hence BUSCLKX2 and BUSCLKX4.

### 11.6 Oscillator During Break Mode

The oscillator continues to drive BUSCLKX2 and BUSCLKX4 when the device enters the break state.

### 11.7 CONFIG2 Options

Two CONFIG2 register options affect the operation of the oscillator module: OSCOPT1 and OSCOPT0. All CONFIG2 register bits will have a default configuration. Refer to [Section 5. Configuration Register \(CONFIG\)](#) for more information on how the CONFIG2 register is used.

Table 11-2 shows how the OSCOPT bits are used to select the oscillator clock source.

Table 11-2. Oscillator Modes

OSCOPT1	OSCOPT0	Oscillator Modes
0	0	Internal Oscillator
0	1	External Oscillator
1	0	External RC
1	1	External Crystal

### 11.8 Input/Output (I/O) Registers

The oscillator module contains these two registers:

1. Oscillator status register (OSCSTAT)
2. Oscillator trim register (OSCTRIM)

#### 11.8.1 Oscillator Status Register

The oscillator status register (OSCSTAT) contains the bits for switching from internal to external clock sources.

Address: \$0036

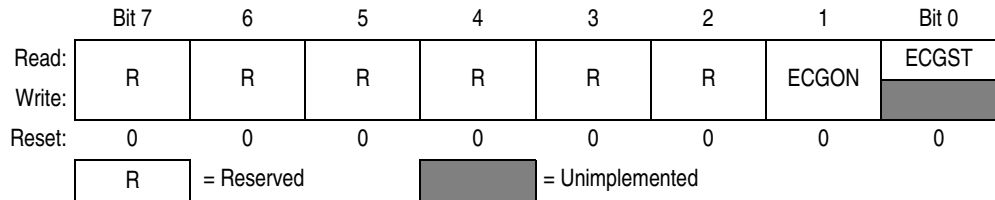


Figure 11-4. Oscillator Status Register (OSCSTAT)

#### ECGON — External Clock Generator On Bit

This read/write bit enables external clock generator, so that the switching process can be initiated. This bit is forced low during reset. This bit is ignored in monitor mode with the internal oscillator bypassed.

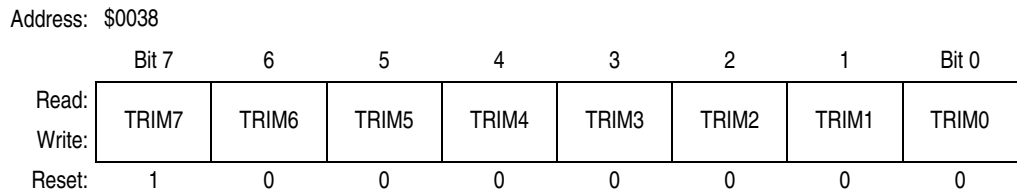
- 1 = External clock generator enabled
- 0 = External clock generator disabled

#### ECGST — External Clock Status Bit

This read-only bit indicates whether or not an external clock source is engaged to drive the system clock.

- 1 = An external clock source engaged
- 0 = An external clock source disengaged

**11.8.2 Oscillator Trim Register (OSCTRIM)**



**Figure 11-5. Oscillator Trim Register (OSCTRIM)**

**TRIM7–TRIM0 — Internal Oscillator Trim Factor Bits**

These read/write bits change the size of the internal capacitor used by the internal oscillator. By measuring the period of the internal clock and adjusting this factor accordingly, the frequency of the internal clock can be fine tuned. Increasing (decreasing) this factor by one increases (decreases) the period by approximately 0.2% of the untrimmed period (the period for TRIM = \$80). The trimmed frequency is guaranteed not to vary by more than  $\pm 5\%$  over the full specified range of temperature and voltage. The reset value is \$80, which sets the frequency to 4.0 MHz (1.0 MHz bus speed)  $\pm 25\%$ .

Freescale Semiconductor, Inc.



Section 12. PLL Tuned UHF Transmitter Module

12.1 Introduction

This section describes the integrated radio frequency (RF) module. This module integrates an ultra high frequency (UHF) transmitter offering these key features:

- Switchable frequency bands: 315, 434, and 868 MHz
- On/off keying (OOK) and frequency shift keying (FSK) modulation
- Adjustable output power range
- Fully integrated voltage-controlled oscillator (VCO)
- Supply voltage range: 1.9 to 3.6 V
- Very low standby current: 0.1 nA @ T<sub>A</sub> = 25°C
- Low supply voltage shutdown
- Data clock output for microcontroller
- Low external component count

Architecture of the module is described in [Figure 12-1](#).

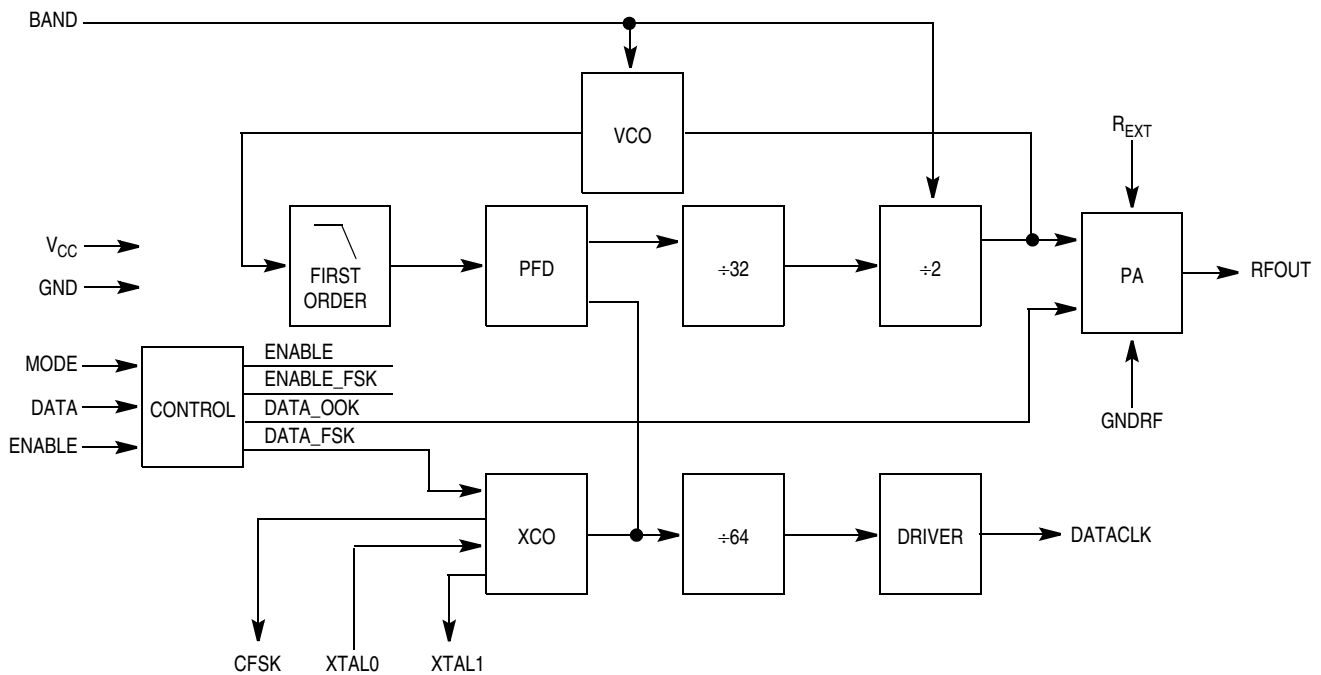
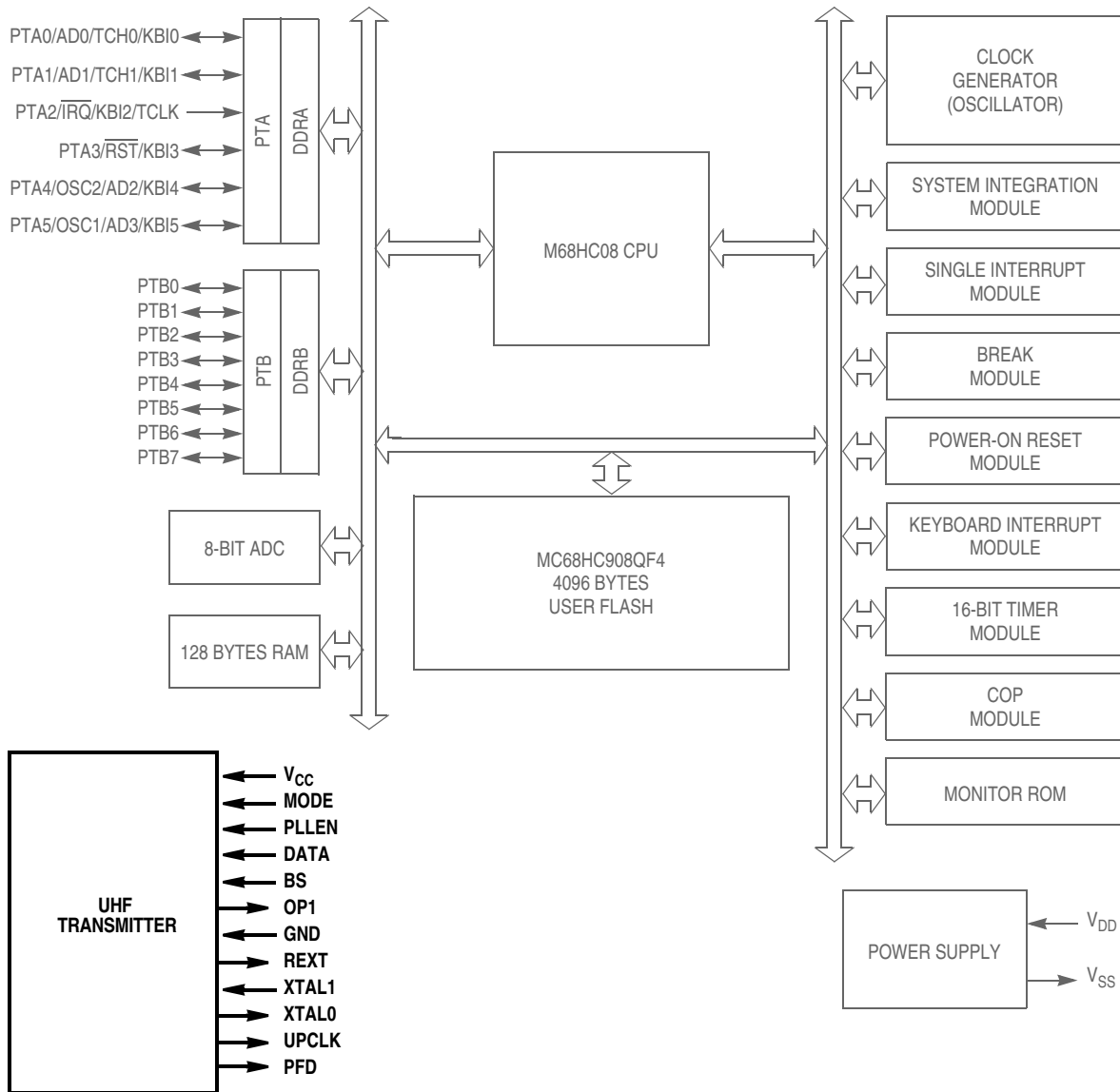


Figure 12-1. Simplified Integrated RF Module Block Diagram

PLL Tuned UHF Transmitter Module



$\overline{RST}$ ,  $\overline{IRQ}$ : Pins have internal (about 30K Ohms) pull up  
 PTA[0:5]: High current sink and source capability  
 PTA[0:5]: Pins have programmable keyboard interrupt and pull up

Figure 12-2. Block Diagram Highlighting PLL Tuned UHF Transmitter Block and Pins

Freescale Semiconductor, Inc.

## 12.2 Transmitter Functional Description

The transmitter is a phase-locked loop (PLL) tuned low-power UHF transmitter. The different modes of operation are controlled by the microcontroller through several digital input pins. The power supply voltage ranges from 1.9 V to 3.6 V allowing operation with a single lithium cell.

## 12.3 Phase-Lock Loop (PLL) and Local Oscillator

The VCO is a completely integrated relaxation oscillator. The phase frequency detector (PFD) and the loop filter are fully integrated. The exact output frequency is equal to:

$$f_{RFOUT} = f_{XTAL} \times \text{PLL divider ratio}$$

The frequency band of operation is selected through the BAND pin. [Table 12-1](#) provides details for each frequency band selection.

**Table 12-1. Frequency Band Selection and Associated Divider Ratios**

BAND Input Level	Frequency Band (MHz)	PLL Divider Ratio	Crystal Oscillator Frequency (MHz)
High	315	32	9.84
	434		13.56
Low	868	64	

An out-of-lock function is performed by monitoring the internal PFD output voltage. When it exceeds its limits, the RF output stage is disabled.

## 12.4 RF Output Stage

The output stage is a single-ended square wave switched current source. Harmonics will be present in the output current drive. Their radiated absolute level depends on the antenna characteristics and output power. Typical application demonstrates compliance to European Telecommunications Standards Institute (ETSI) standard. A resistor  $R_{EXT}$  connected to the REXT pin controls the output power allowing a tradeoff between radiated power and current consumption. The output voltage is internally clamped to:

$$V_{CC} \pm 2 V_{BE} \text{ (typically } V_{CC} \pm 1.5 \text{ V @ } T_A = 25^\circ\text{C).}$$

## 12.5 Modulation

If a low-logic level is applied on pin MODE, then the on/off keying (OOK) modulation is selected. This modulation is performed by switching on/off the RF output stage. The logic level applied on pin DATA controls the output stage state:

- DATA = 0 → output stage off
- DATA = 1 → output stage on

If a high-logic level is applied on pin MODE, then frequency shift keying (FSK) modulation is selected. This modulation is achieved by modulating the frequency of the reference oscillator. This frequency change is performed by switching the external crystal load capacitor. The logic level applied on pin DATA controls the internal switch connected to pin CFSK:

- DATA = 0 → switch off
- DATA = 1 → switch on

In case of [Figure 12-6](#), where the two capacitors C6 and C9 are in series:

- DATA = 0 leads to the high value of the carrier frequency
- DATA = 1 leads to the low value of the carrier frequency

This crystal pulling solution implies that the RF output frequency deviation equals the crystal frequency deviation multiplied by the PLL divider ratio (see [Table 12-1](#)).

## 12.6 Microcontroller Interfaces

Four digital input pins (ENABLE, DATA, BAND, and MODE) enable the circuit to be controlled by a microcontroller. It is recommended to configure the band frequency and the modulation type before enabling the circuit. In a typical application the input pins BAND and MODE are hardwired.

One digital output (DATACLK) provides the microcontroller a reference frequency for data clocking. This frequency is equal to the crystal oscillator frequency divided by 64 (see [Table 12-2](#)).

**Table 12-2. DATACLK Frequency versus Crystal Oscillator Frequency**

Crystal Oscillator Frequency (MHz)	DATACLK Frequency (kHz)
9.84	154
13.56	212



12.7 State Machine

Figure 12-3 details the main state machine.

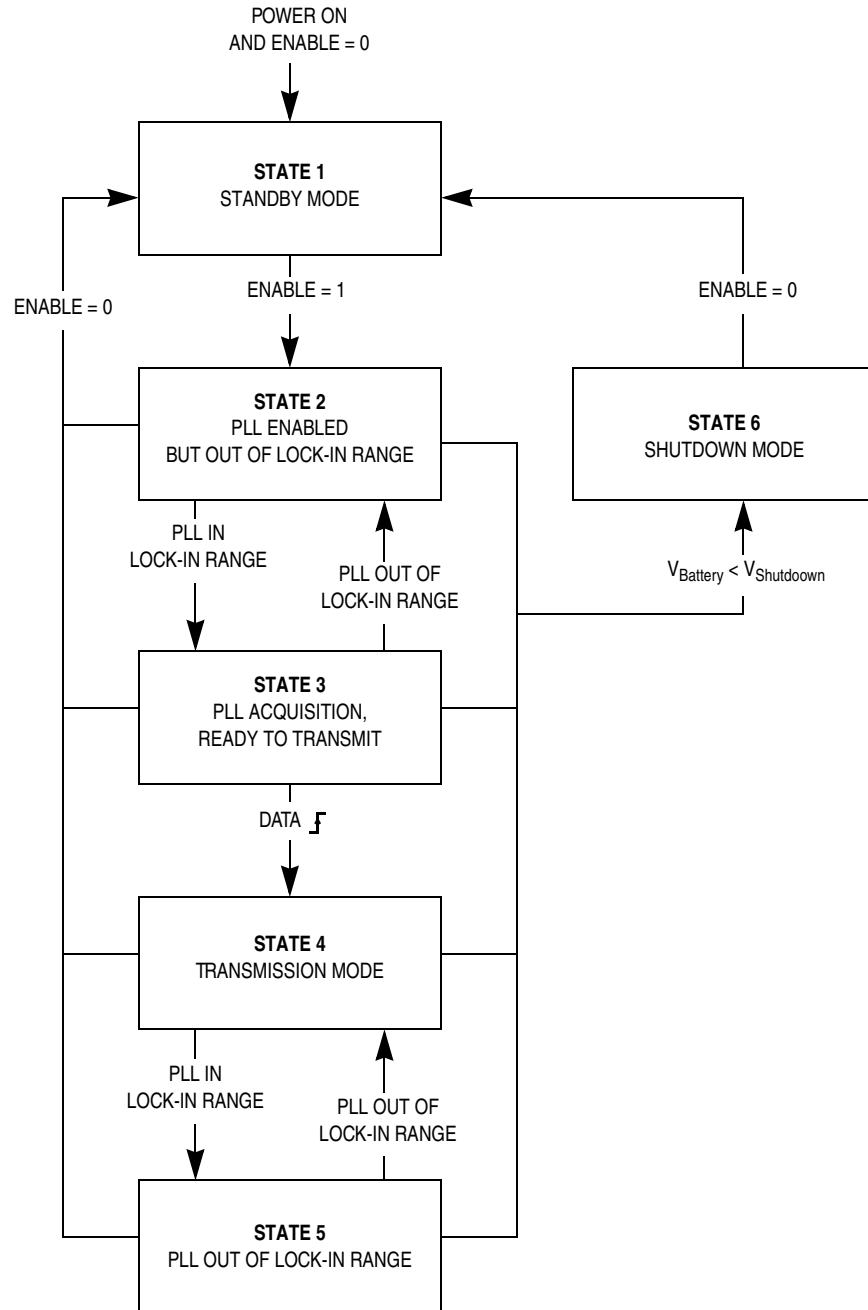


Figure 12-3. Main State Machine

**State 1**

The circuit is in standby mode and draws only a leakage current from the power supply.

**State 2**

In this state, the PLL is enabled but out of the lock-in range. Therefore the RF output stage is switched off preventing any data transmission. Data clock is available on pin DATACLK. In normal operation, this state is transitional.

**State 3**

In this state, the PLL is within the lock-in range.

If  $t < t_{PLL\_Lock\_In}$ , then the PLL can still be in acquisition mode.

If  $t \geq t_{PLL\_Lock\_In}$ , then the PLL is locked.

The circuit is ready to transmit in band and is waiting for the first data (see [Figure 12-4](#)).

**State 4**

A rising edge on pin DATA starts the transmission. Data entered on pin DATA are output on pin RFOUT. The modulation is the one selected through the level applied on pin MODE.

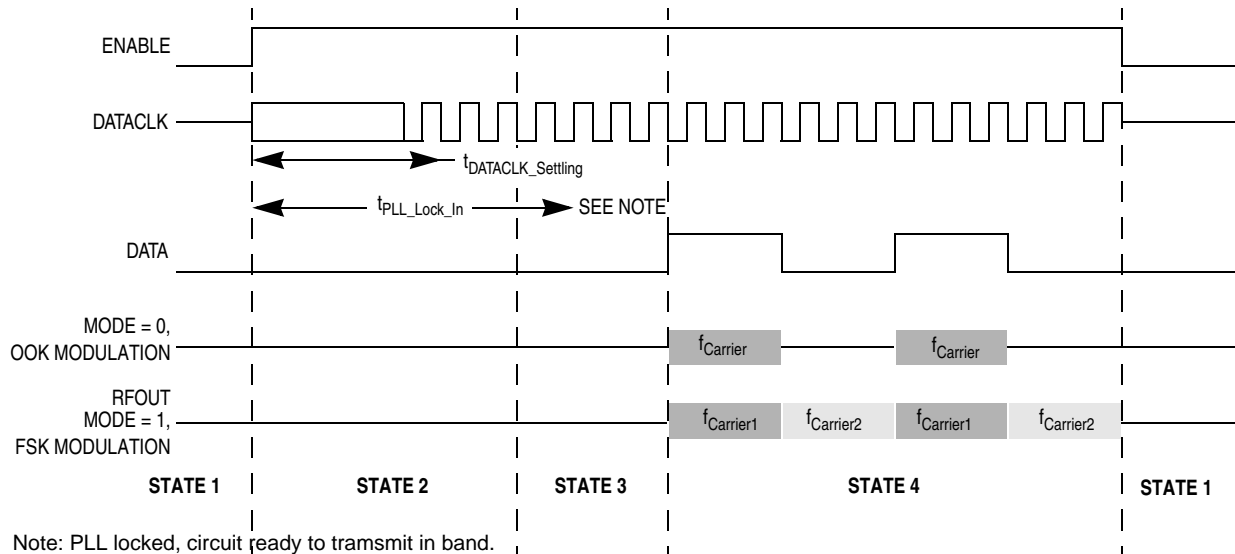
**State 5**

An out-of-lock condition has been detected. The RF output stage is switched off preventing any data transmission. Data clock is available on pin DATACLK.

**State 6**

When the supply voltage falls below the shutdown voltage threshold ( $V_{SDWN}$ ) the whole circuit is switched off. Applying a low level on pin ENABLE is the only condition to get out of this state.

**Figure 12-4** shows the waveforms of the main signals for a typical application cycle



**Figure 12-4. Signals, Waveforms, and Timing Definitions**

## 12.8 Power Management

When the battery voltage falls below the shutdown voltage threshold ( $V_{SDWN}$ ) the whole circuit is switched off.

**NOTE:** After this shutdown, the circuit is latched until a low level is applied on pin ENABLE (see state 6 under [12.7 State Machine](#)).

## 12.9 Data Clock

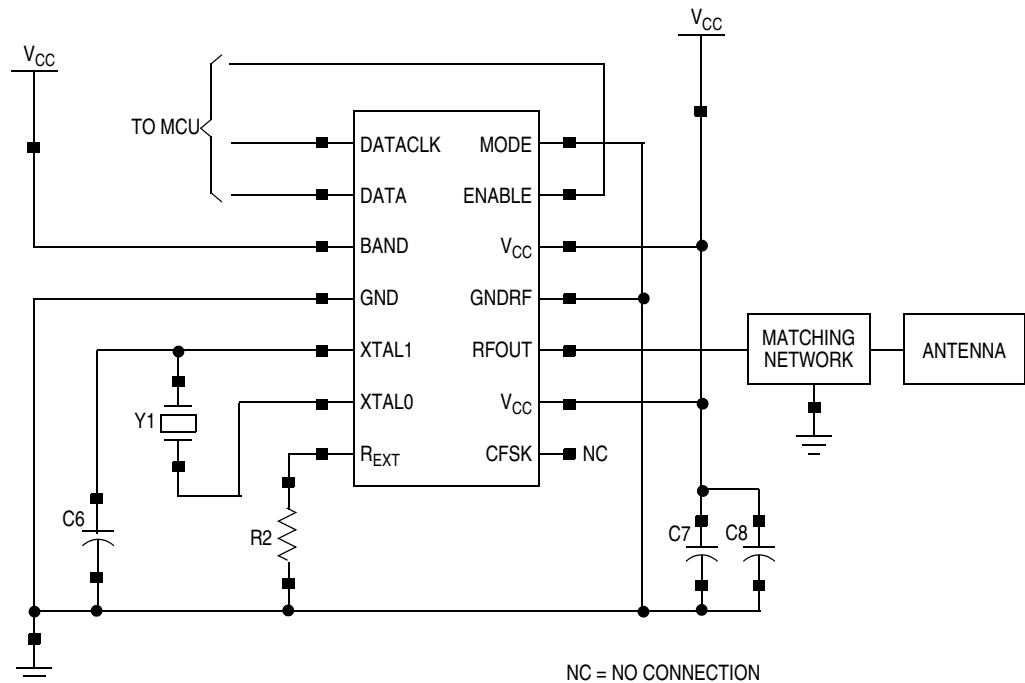
When the data clock starts, the high-to-low ratio may be uneven. Similarly the clock is switched off asynchronously so the last period length is not guaranteed.

## 12.10 Application Information

This subsection provides application information for the usage of the UHF transmitter module.

### 12.10.1 Application Schematics in OOK and FSK Modulation

**Figure 12-5** and **Figure 12-6** show application schematics in OOK and FSK modulation for the 315-MHz and 434-MHz frequency bands. For 868-MHz band application, the input pin BAND must be wired to GND. See component description in **Table 12-4** and **Table 12-5**.



**Figure 12-5. Application Schematic in OOK Modulation, 315-MHz and 434-MHz Frequency Bands**

PLL Tuned UHF Transmitter Module

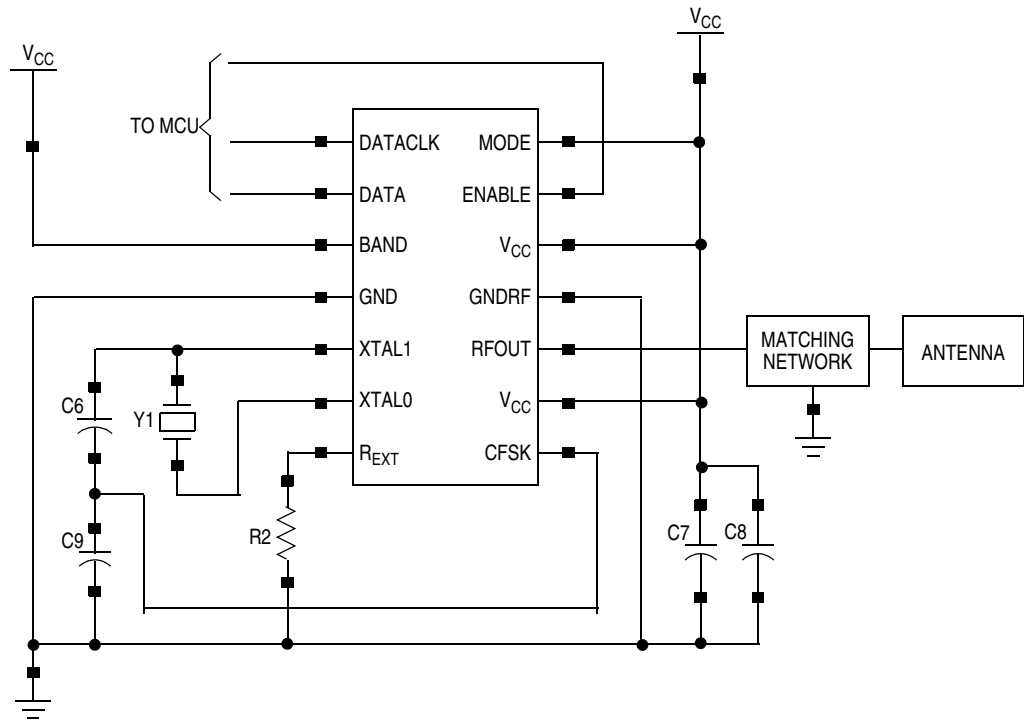


Figure 12-6. Application Schematic in FSK Modulation, 315-MHz and 434-MHz Frequency Bands

Table 12-3. Component Description

Component	Function	Value	Unit
Y1	Crystal	315-MHz band: 9.84, see <a href="#">Table 12-5</a>	MHz
		434-MHz band: 13.56, see <a href="#">Table 12-5</a>	MHz
		868-MHz band: 13.56, see <a href="#">Table 12-5</a>	MHz
R2	RF output level setting resistor (R <sub>EXT</sub> )	12	kΩ
C6	Crystal load capacitor	OOK modulation: 18	pF
		FSK modulation: 22	pF
C7	Power supply decoupling capacitor	10	nF
C8		100	pF
C9	Crystal pulling capacitor for FSK modulation only	See <a href="#">Table 12-5</a>	pF

A example of crystal reference is: Tokyo Denpa TTS-3B 13568.750 kHz, see [Table 12-4](#).

**Table 12-4. Recommended Crystal Characteristics (SMD Ceramic Package)**

Parameter	Value	Unit
Load capacitance	20	pF
Motional capacitance	6.7	fF
Static capacitance	2	pF
Loss resistance	40	W

**Table 12-5. Crystal Pulling Capacitor Value versus Carrier Frequency Total Deviation**

Carrier Frequency (MHz)	Carrier Frequency Total Deviation (kHz)	Capacitor Value (pF)
434	40	18
	70	10
	100	6.8
868	80	18
	140	10
	200	6.8

**12.10.2 Complete Application Schematic**

[Figure 12-7](#) gives a complete application schematic using the Motorola MC68HC908RF2. OOK modulation is selected,  $f_{Carrier} = 433.92$  MHz.



Section 13. Input/Output (I/O) Ports

13.1 Introduction

The MC68HC908QF4 has thirteen bidirectional pins and one input only pin. 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.

Figure 13-1 provides a summary of the I/O registers.

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0	
\$0000	Port A Data Register (PTA) <a href="#">See page 112.</a>	Read:	R	AWUL	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:	Unaffected by reset							
\$0001	Port B Data Register (PTB) <a href="#">See page 115.</a>	Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:	Unaffected by reset							
\$0004	Data Direction Register A (DDRA) <a href="#">See page 113.</a>	Read:	R	R	DDRA5	DDRA4	DDRA3	0	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0005	Data Direction Register B (DDRB) <a href="#">See page 115.</a>	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000B	Port A Input Pullup Enable Register (PTAPUE) <a href="#">See page 114.</a>	Read:	OSC2EN		PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000C	Port B Input Pullup Enable Register (PTBPUE) <a href="#">See page 116.</a>	Read:	PTBPUE7	PTBPUE6	PTBPUE5	PTBPUE4	PTBPUE3	PTBPUE2	PTBPUE1	PTBPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

R = Reserved       = Unimplemented

Figure 13-1. I/O Port Register Summary

13.2 Port A

Port A is a 6-bit special function port that shares all six of its pins with the keyboard interrupt (KBI) module (see [Section 9. Keyboard Interrupt Module \(KBI\)](#)). Each port A pin also has a software configurable pullup device if the corresponding port pin is configured as an input port.

**NOTE:** PTA2 is input only.

When the  $\overline{IRQ}$  function is enabled in the configuration register 2 (CONFIG2), bit 2 of the port A data register (PTA) will always read a 0. In this case, the BIH and BIL instructions can be used to read the logic level on the PTA2 pin. When the  $\overline{IRQ}$  function is disabled, these instructions will behave as if the PTA2 pin is a logic 1. However, reading bit 2 of PTA will read the actual logic level on the pin.

13.2.1 Port A Data Register

The port A data register (PTA) contains a data latch for each of the six port A pins.

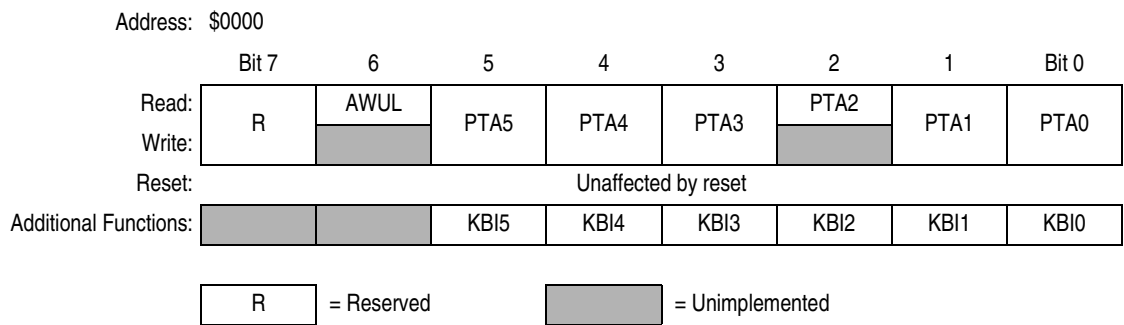


Figure 13-2. Port A Data Register (PTA)

PTA[5:0] — Port A Data Bits

These read/write bits are software programmable. Data direction of each port A pin is under the control of the corresponding bit in data direction register A. Reset has no effect on port A data.

AWUL — Auto Wakeup Latch Data Bit

This is a read-only bit which has the value of the auto wakeup interrupt request latch. The wakeup request signal is generated internally (see [Section 4. Auto Wakeup Module \(AWU\)](#)). There is no PTA6 port nor any of the associated bits such as PTA6 data register, pullup enable or direction.

KBI[5:0] — Port A Keyboard Interrupts

The keyboard interrupt enable bits, KBIE5–KBIE0, in the keyboard interrupt control enable register (KBIER) enable the port A pins as external interrupt pins (see [Section 9. Keyboard Interrupt Module \(KBI\)](#)).



13.2.2 Data Direction Register A

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

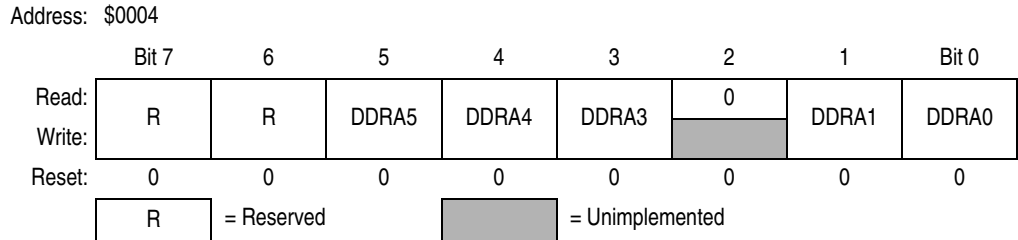


Figure 13-3. Data Direction Register A (DDRA)

DDRA[5:0] — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA[5:0], configuring all port A pins as inputs.

- 1 = Corresponding port A pin configured as output
- 0 = Corresponding port A pin configured as input

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

Figure 13-4 shows the port A I/O logic.

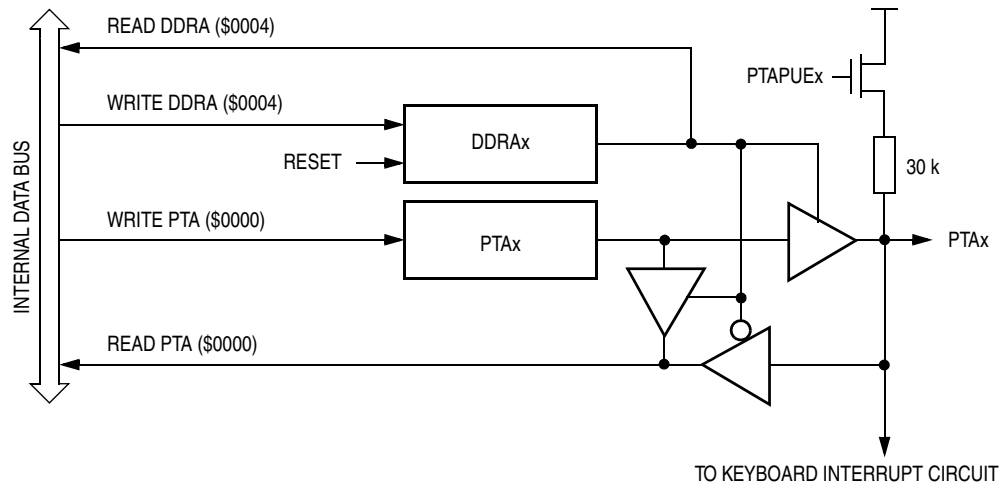


Figure 13-4. Port A I/O Circuit

**NOTE:** Figure 13-4 does not apply to PTA2

When DDRAx is a 1, reading address \$0000 reads the PTAx data latch. When DDRAx is a 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.

13.2.3 Port A Input Pullup Enable Register

The port A input pullup enable register (PTAPUE) contains a software configurable pullup device for each of the six port A pins. Each bit is individually configurable and requires the corresponding data direction register, DDRAx, to be configured as input. Each pullup device is automatically and dynamically disabled when its corresponding DDRAx bit is configured as output.

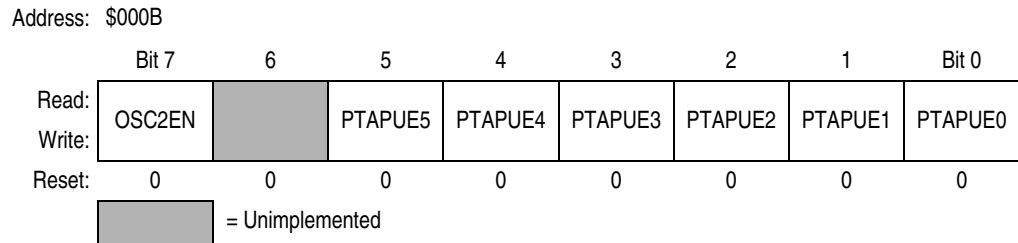


Figure 13-5. Port A Input Pullup Enable Register (PTAPUE)

OSC2EN — Enable PTA4 on OSC2 Pin

This read/write bit configures the OSC2 pin function when internal oscillator or RC oscillator option is selected. This bit has no effect for the XTAL or external oscillator options.

- 1 = OSC2 pin outputs the internal or RC oscillator clock (BUSCLKX4)
- 0 = OSC2 pin configured for PTA4 I/O, having all the interrupt and pullup functions

PTAPUE[5:0] — Port A Input Pullup Enable Bits

These read/write bits are software programmable to enable pullup devices on port A pins.

- 1 = Corresponding port A pin configured to have internal pull if its DDRA bit is set to 0
- 0 = Pullup device is disconnected on the corresponding port A pin regardless of the state of its DDRA bit

Table 13-1 summarizes the operation of the port A pins.

Table 13-1. Port A Pin Functions

PTAPUE Bit	DDRA Bit	PTA Bit	I/O Pin Mode	Accesses to DDRA		Accesses to PTA	
				Read/Write		Read	Write
1	0	X <sup>(1)</sup>	Input, V <sub>DD</sub> <sup>(2)</sup>	DDRA5–DDRA0		Pin	PTA5–PTA0 <sup>(3)</sup>
0	0	X	Input, Hi-Z <sup>(4)</sup>	DDRA5–DDRA0		Pin	PTA5–PTA0 <sup>(3)</sup>
X	1	X	Output	DDRA5–DDRA0		PTA5–PTA0	PTA5–PTA0 <sup>(5)</sup>

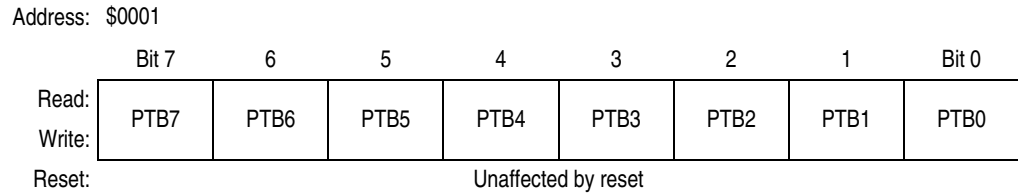
1. X = don't care
2. I/O pin pulled to V<sub>DD</sub> by internal pullup.
3. Writing affects data register, but does not affect input.
4. Hi-Z = high impedance
5. Output does not apply to PTA2

### 13.3 Port B

Port B is an 8-bit general purpose I/O port.

#### 13.3.1 Port B Data Register

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



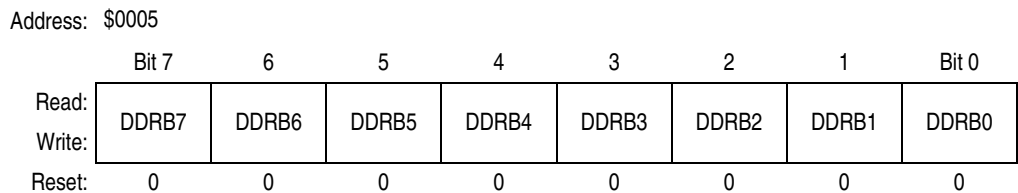
**Figure 13-6. Port B Data Register (PTB)**

#### PTB[7:0] — Port B Data Bits

These read/write bits are software programmable. Data direction of each port B pin is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.

#### 13.3.2 Data Direction Register B

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



**Figure 13-7. Data Direction Register B (DDRB)**

#### DDRB[7:0] — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB[7:0], configuring all port B pins as inputs.

- 1 = Corresponding port B pin configured as output
- 0 = Corresponding port B pin configured as input

**NOTE:** Avoid glitches on port B pins by writing to the port B data register before changing data direction register B bits from 0 to 1. [Figure 13-8](#) shows the port B I/O logic.

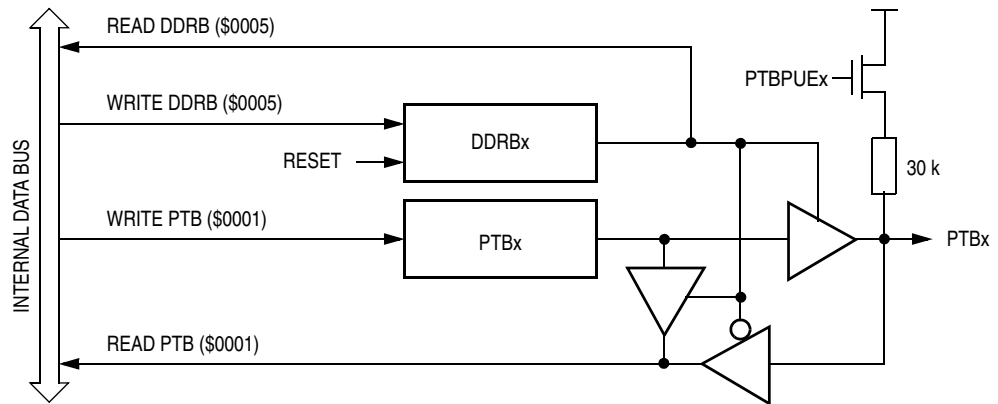


Figure 13-8. Port B I/O Circuit

When DDRBx is a 1, reading address \$0001 reads the PTBx data latch. When DDRBx is a 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 13-2](#) summarizes the operation of the port B pins.

Table 13-2. Port B Pin Functions

DDRB Bit	PTB Bit	I/O Pin Mode	Accesses to DDRB		Accesses to PTB	
			Read/Write	Read	Write	
0	X <sup>(1)</sup>	Input, Hi-Z <sup>(2)</sup>	DDRB7–DDRB0	Pin	PTB7–PTB0 <sup>(3)</sup>	
1	X	Output	DDRB7–DDRB0	Pin	PTB7–PTB0	

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

### 13.3.3 Port B Input Pullup Enable Register

The port B input pullup enable register (PTBPUE) contains a software configurable pullup device for each of the eight port B pins. Each bit is individually configurable and requires the corresponding data direction register, DDRBx, be configured as input. Each pullup device is automatically and dynamically disabled when its corresponding DDRBx bit is configured as output.

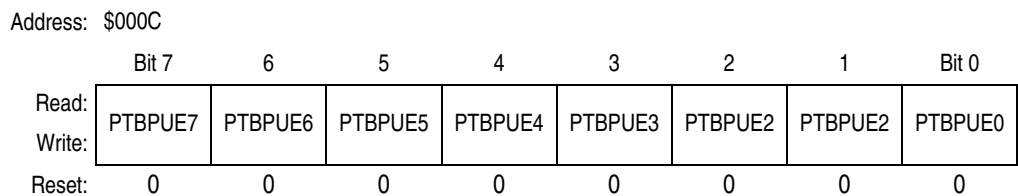


Figure 13-9. Port B Input Pullup Enable Register (PTBPUE)

**PTBPUE[7:0] — Port B Input Pullup Enable Bits**

These read/write bits are software programmable to enable pullup devices on port B pins

1 = Corresponding port B pin configured to have internal pull if its DDRB bit is set to 0

0 = Pullup device is disconnected on the corresponding port B pin regardless of the state of its DDRB bit.

**Table 13-3** summarizes the operation of the port B pins.

**Table 13-3. Port B Pin Functions**

PTBPUE Bit	DDR B Bit	PTB Bit	I/O Pin Mode	Accesses to DDRB	Accesses to PTB	
				Read/Write	Read	Write
1	0	X <sup>(1)</sup>	Input, V <sub>DD</sub> <sup>(2)</sup>	DDR B7–DDR B0	Pin	PTB7–PTB0 <sup>(3)</sup>
0	0	X	Input, Hi-Z <sup>(4)</sup>	DDR B7–DDR B0	Pin	PTB7–PTB0 <sup>(3)</sup>
X	1	X	Output	DDR B7–DDR B0	PTB7–PTB0	PTB7–PTB0

1. X = don't care
2. I/O pin pulled to V<sub>DD</sub> by internal pullup.
3. Writing affects data register, but does not affect input.
4. Hi-Z = high impedance



## Section 14. System Integration Module (SIM)

### 14.1 Introduction

This section describes the system integration module (SIM), which supports up to 24 external and/or internal interrupts. Together with the central processor unit (CPU), the SIM controls all microcontroller unit (MCU) activities. A block diagram of the SIM is shown in [Figure 14-1](#). [Figure 14-2](#) is 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 computer operating properly (COP) timeout
- Interrupt control:
  - Acknowledge timing
  - Arbitration control timing
  - Vector address generation
- CPU enable/disable timing

### 14.2 $\overline{\text{RST}}$ and $\overline{\text{IRQ}}$ Pins Initialization

$\overline{\text{RST}}$  and  $\overline{\text{IRQ}}$  pins come out of reset as PTA3 and PTA2 respectively.  $\overline{\text{RST}}$  and  $\overline{\text{IRQ}}$  functions can be activated by programming CONFIG2 accordingly. Refer to [Section 5. Configuration Register \(CONFIG\)](#).

System Integration Module (SIM)

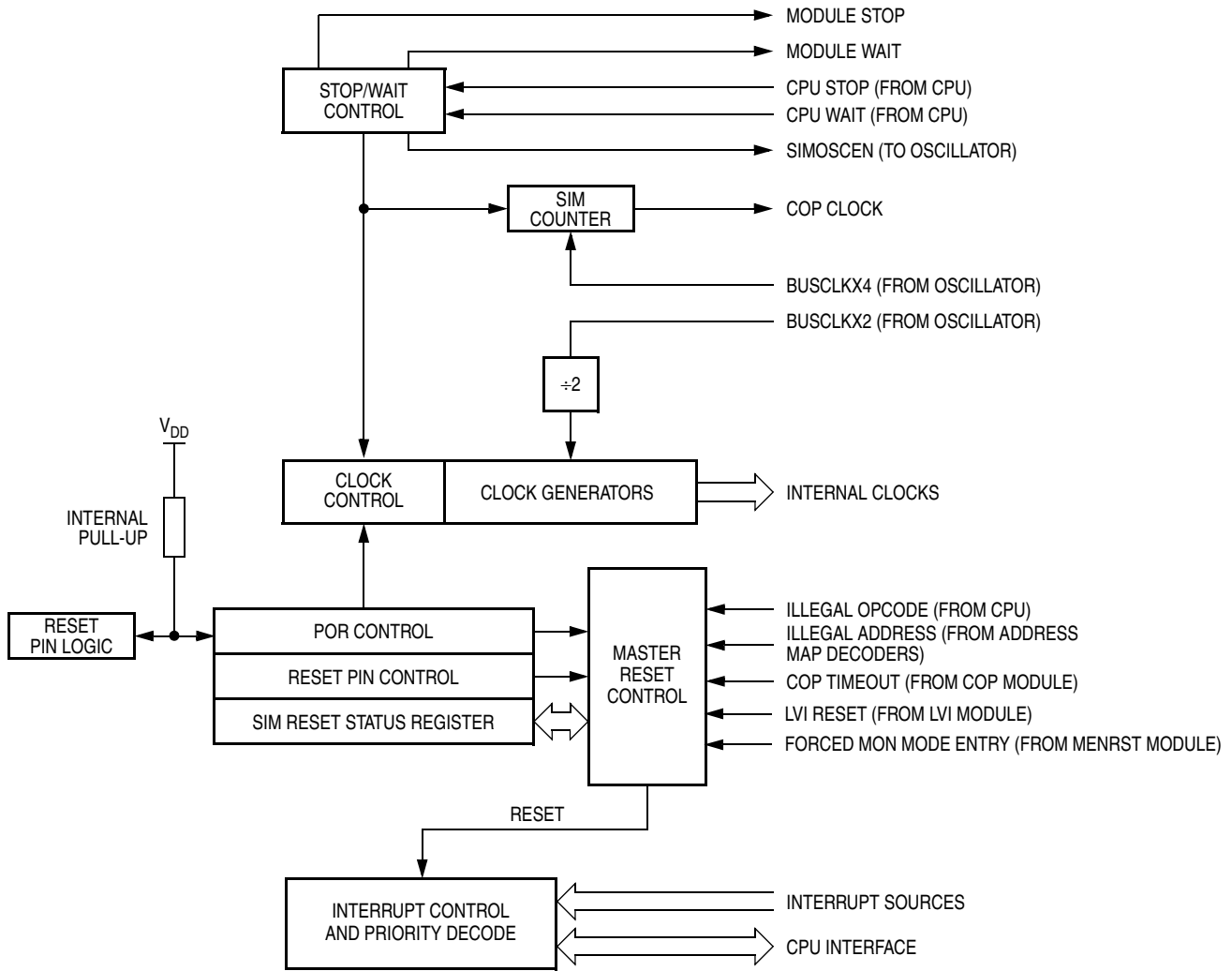


Figure 14-1. SIM Block Diagram

Table 14-1. Signal Name Conventions

Signal Name	Description
BUSCLKX4	Buffered clock from the internal, RC or XTAL oscillator circuit.
BUSCLKX2	The BUSCLKX4 frequency divided by two. This signal is again divided by two in the SIM to generate the internal bus clocks (bus clock = BUSCLKX4 ÷ 4).
Address bus	Internal address bus
Data bus	Internal data bus
PORRST	Signal from the power-on reset module to the SIM
IRST	Internal reset signal
R/ $\bar{W}$	Read/write signal



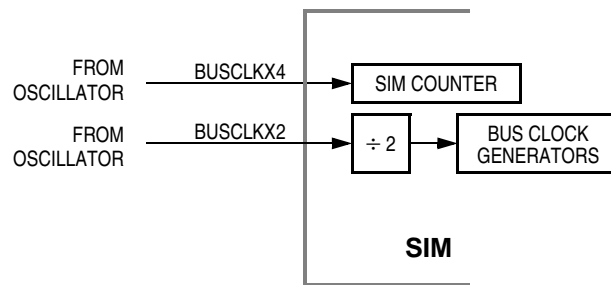
Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$FE00	Break Status Register (BSR) <a href="#">See page 161.</a>	Read:	R	R	R	R	R	R	SBSW	R
		Write:							Note 1	
		Reset:	0	0	0	0	0	0	0	0
1. Writing a 0 clears SBSW.										
\$FE01	SIM Reset Status Register (SRSR) <a href="#">See page 135.</a>	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
		Write:								
\$FE02	Reserved	POR:	1	0	0	0	0	0	0	0
\$FE03	Break Flag Control Register (BFCR) <a href="#">See page 136.</a>	Read:								
		Write:	BCFE	R	R	R	R	R	R	R
		Reset:	0							
\$FE04	Interrupt Status Register 1 (INT1) <a href="#">See page 130.</a>	Read:	0	IF5	IF4	IF3	0	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE05	Interrupt Status Register 2 (INT2) <a href="#">See page 131.</a>	Read:	IF14	0	0	0	0	0	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE06	Interrupt Status Register 3 (INT3) <a href="#">See page 131.</a>	Read:	0	0	0	0	0	0	0	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented     
 R = Reserved

**Figure 14-2. SIM I/O Register Summary**

### 14.3 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, BUSCLKX2, as shown in [Figure 14-3](#).



**Figure 14-3. SIM Clock Signals**

### 14.3.1 Bus Timing

In user mode, the internal bus frequency is the oscillator frequency (BUSCLKX4) divided by four.

### 14.3.2 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 BUSCLKX4 cycle POR time out has completed. The IBUS clocks start upon completion of the time out.

### 14.3.3 Clocks in Stop Mode and Wait Mode

Upon exit from stop mode by an interrupt or reset, the SIM allows BUSCLKX4 to clock the SIM counter. The CPU and peripheral clocks do not become active until after the stop delay time out. This time out is selectable as 4096 or 32 BUSCLKX4 cycles. See [14.7.2 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.

## 14.4 Reset and System Initialization

The MCU has these reset sources:

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

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

An internal reset clears the SIM counter (see [14.5 SIM Counter](#)), but an external reset does not. Each of the resets sets a corresponding bit in the SIM reset status register (SRSR). See [14.8 SIM Registers](#).

14.4.1 External Pin Reset

The  $\overline{RST}$  pin circuits include an internal pullup device. Pulling the asynchronous  $\overline{RST}$  pin low halts all processing. The PIN bit of the SIM reset status register (SRSR) is set as long as  $\overline{RST}$  is held low for at least the minimum  $t_{RL}$  time. **Figure 14-4** shows the relative timing. The  $\overline{RST}$  pin function is only available if the RSTEN bit is set in the CONFIG1 register.

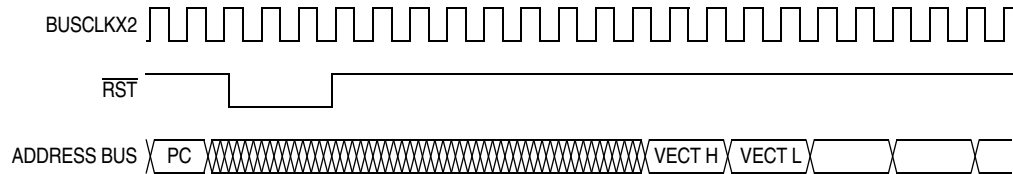


Figure 14-4. External Reset Timing

14.4.2 Active Resets from Internal Sources

The  $\overline{RST}$  pin is initially setup as a general-purpose input after a POR. Setting the RSTEN bit in the CONFIG1 register enables the pin for the reset function. This section assumes the RSTEN bit is set when describing activity on the  $\overline{RST}$  pin.

All internal reset sources actively pull the  $\overline{RST}$  pin low for 32 BUSCLKX4 cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles (see **Figure 14-5**). An internal reset can be caused by an illegal address, illegal opcode, COP time out, LVI, or POR (see **Figure 14-6**).

**NOTE:** For POR and LVI resets, the SIM cycles through 4096 BUSCLKX4 cycles during which the SIM forces the  $\overline{RST}$  pin low. The internal reset signal then follows the sequence from the falling edge of  $\overline{RST}$  shown in **Figure 14-5**.

The COP reset is asynchronous to the bus clock.

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

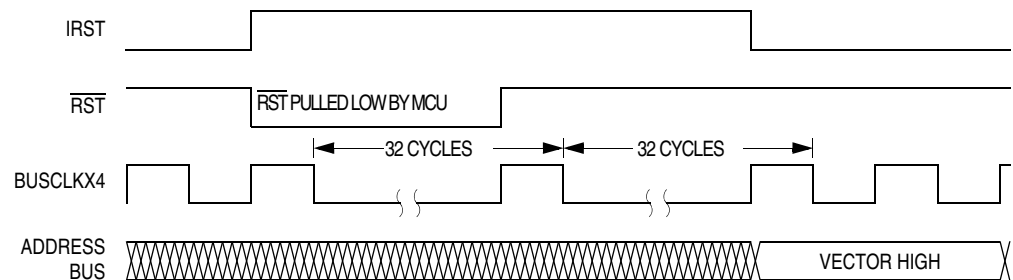


Figure 14-5. Internal Reset Timing

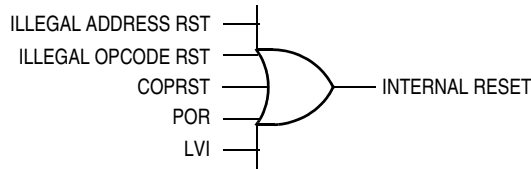


Figure 14-6. Sources of Internal Reset

Table 14-2. Reset Recovery Timing

Reset Recovery Type	Actual Number of Cycles
POR/LVI	4163 (4096 + 64 + 3)
All others	67 (64 + 3)

14.4.2.1 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 SIM counter counts out 4096 BUSCLKX4 cycles. Sixty-four BUSCLKX4 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 BUSCLKX4.
- Internal clocks to the CPU and modules are held inactive for 4096 BUSCLKX4 cycles to allow stabilization of the oscillator.
- The POR bit of the SIM reset status register (SRSR) is set.

See [Figure 14-7](#).

14.4.2.2 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  $\overline{RST}$  pin for all internal reset sources.

To prevent a COP module time out, write any value to location \$FFFF. Writing to location \$FFFF clears the COP counter and stages 12–5 of the SIM counter. The SIM counter output, which occurs at least every  $(2^{12} - 2^4)$  BUSCLKX4 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 time out.

The COP module is disabled during a break interrupt with monitor mode when BDCOP bit is set in break auxiliary register (BRKAR).

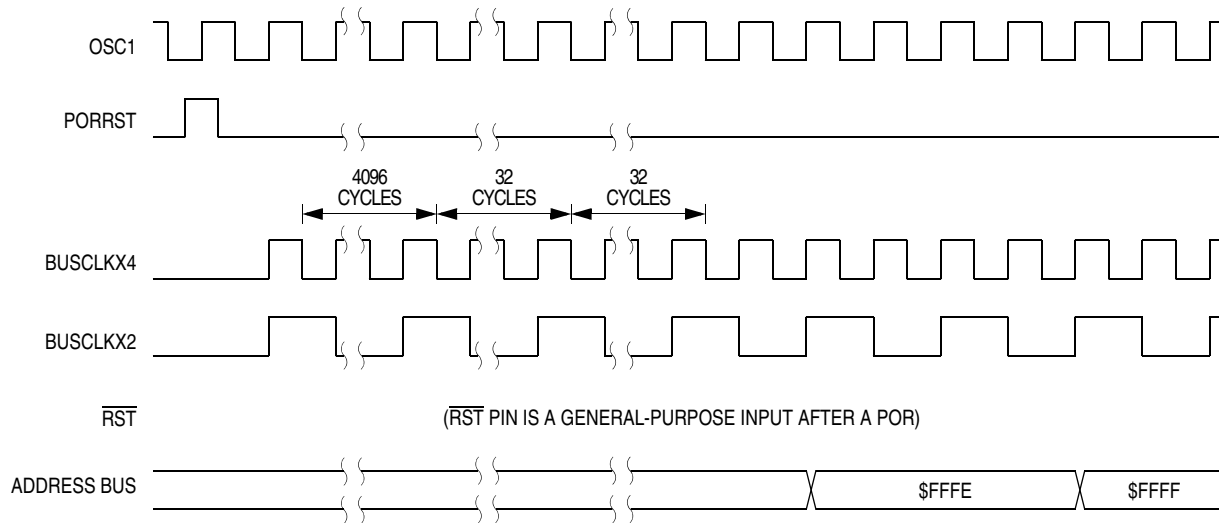


Figure 14-7. POR Recovery

14.4.2.3 Illegal Opcode Reset

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

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

14.4.2.4 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. See [Figure 2-1. Memory Map](#) for memory ranges.

14.4.2.5 Low-Voltage Inhibit (LVI) Reset

The LVI asserts its output to the SIM when the  $V_{DD}$  voltage falls to the LVI trip voltage  $V_{TRIPF}$ . The LVI bit in the SIM reset status register (SRSR) is set, and the external reset pin ( $\overline{\text{RST}}$ ) is held low while the SIM counter counts out 4096 BUSCLKX4 cycles after  $V_{DD}$  rises above  $V_{TRIPR}$ . Sixty-four BUSCLKX4 cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur. The SIM actively pulls down the ( $\overline{\text{RST}}$ ) pin for all internal reset sources.

## 14.5 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 uses 12 stages for counting, followed by a 13th stage that triggers a reset of SIM counters and supplies the clock for the COP module. The SIM counter is clocked by the falling edge of BUSCLKX4.

### 14.5.1 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.

### 14.5.2 SIM Counter During Stop Mode Recovery

The SIM counter also is used for stop mode recovery. The STOP instruction clears the SIM counter. After an interrupt, break, or reset, the SIM senses the state of the short stop recovery bit, SSREC, in the configuration register 1 (CONFIG1). If the SSREC bit is a 1, then the stop recovery is reduced from the normal delay of 4096 BUSCLKX4 cycles down to 32 BUSCLKX4 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 in the configuration register 1 (CONFIG1).

### 14.5.3 SIM Counter and Reset States

External reset has no effect on the SIM counter (see [14.7.2 Stop Mode](#) for details.) The SIM counter is free-running after all reset states. See [14.4.2 Active Resets from Internal Sources](#) for counter control and internal reset recovery sequences.

## 14.6 Exception Control

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

1. Interrupts
  - a. Maskable hardware CPU interrupts
  - b. Non-maskable software interrupt instruction (SWI)
2. Reset
3. Break interrupts

### 14.6.1 Interrupts

An interrupt temporarily changes the sequence of program execution to respond to a particular event. [Figure 14-8](#) 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 14-9](#) shows interrupt entry timing. [Figure 14-10](#) shows interrupt recovery timing.

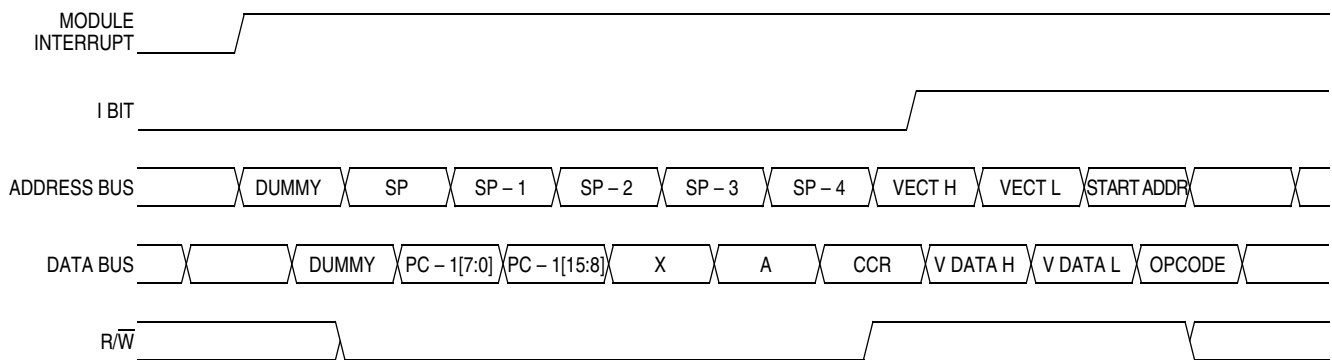


Figure 14-9. Interrupt Entry

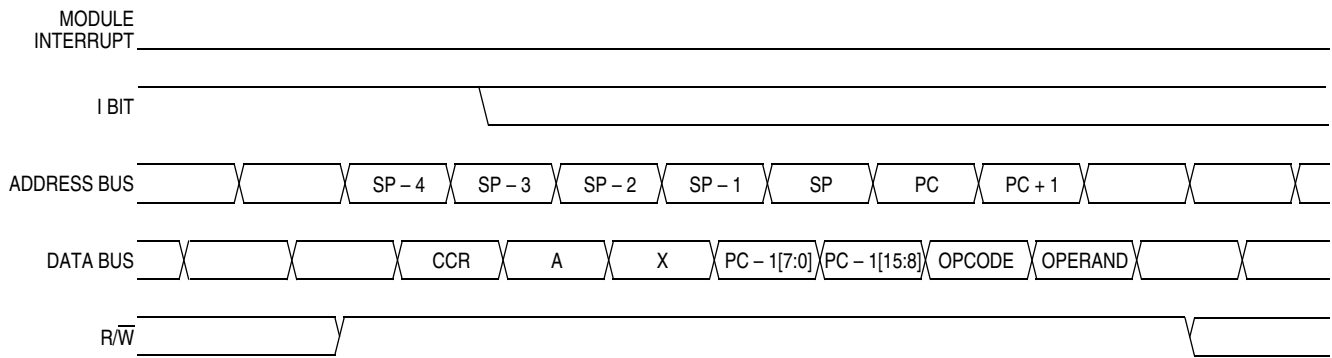


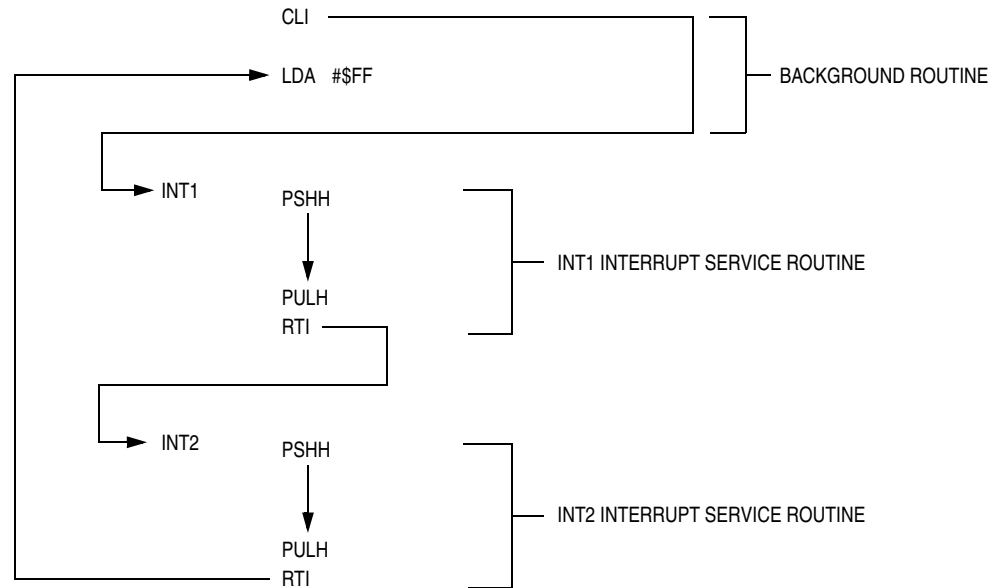
Figure 14-10. Interrupt Recovery

14.6.1.1 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 14-11** demonstrates what happens when two interrupts are pending. If an interrupt is pending upon exit from the original interrupt service routine, the pending interrupt is serviced before the LDA instruction is executed.



**Figure 14-11. Interrupt Recognition Example**

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

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

### 14.6.1.2 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.*

14.6.2 Interrupt Status Registers

The flags in the interrupt status registers identify maskable interrupt sources. **Table 14-3** summarizes the interrupt sources and the interrupt status register flags that they set. The interrupt status registers can be useful for debugging.

Table 14-3. Interrupt Sources

Priority	Source	Flag	Mask <sup>(1)</sup>	INT Register Flag	Vector Address
Highest ↑ ↓ Lowest	Reset	—	—	—	\$FFFE–\$FFFF
	SWI instruction	—	—	—	\$FFFC–\$FFFD
	$\overline{\text{IRQ}}$ pin	IRQF	IMASK	IF1	\$FFFA–\$FFFB
	Timer channel 0 interrupt	CH0F	CH0IE	IF3	\$FFF6–\$FFF7
	Timer channel 1 interrupt	CH1F	CH1IE	IF4	\$FFF4–\$FFF5
	Timer overflow interrupt	TOF	TOIE	IF5	\$FFF2–\$FFF3
	Keyboard interrupt	KEYF	IMASKK	IF14	\$FFE0–\$FFE1
	ADC conversion complete interrupt	COCO	AIEN	IF15	\$FFDE–\$FFDF

1. The I bit in the condition code register is a global mask for all interrupt sources except the SWI instruction.

14.6.2.1 Interrupt Status Register 1

Address: \$FE04

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

R = Reserved

Figure 14-12. Interrupt Status Register 1 (INT1)

IF1 and IF3–IF5 — Interrupt Flags

These flags indicate the presence of interrupt requests from the sources shown in **Table 14-3**.

- 1 = Interrupt request present
- 0 = No interrupt request present

Bit 0, 1, 3, and 7 — Always read 0

14.6.2.2 Interrupt Status Register 2

Address: \$FE05

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF14	0	0	0	0	0	0	0
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 14-13. Interrupt Status Register 2 (INT2)

IF14 — Interrupt Flags

This flag indicates the presence of interrupt requests from the sources shown in [Table 14-3](#).

- 1 = Interrupt request present
- 0 = No interrupt request present

Bit 0–6 — Always read 0

14.6.2.3 Interrupt Status Register 3

Address: \$FE06

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

R = Reserved

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

IF15 — Interrupt Flags

These flags indicate the presence of interrupt requests from the sources shown in [Table 14-3](#).

- 1 = Interrupt request present
- 0 = No interrupt request present

Bit 1–7 — Always read 0

14.6.3 Reset

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

14.6.4 Break Interrupts

The break module can stop normal program flow at a software programmable break point by asserting its break interrupt output. (See [Section 16. Development Support](#).) 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.

### 14.6.5 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 break flag control register (BFCR).

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.

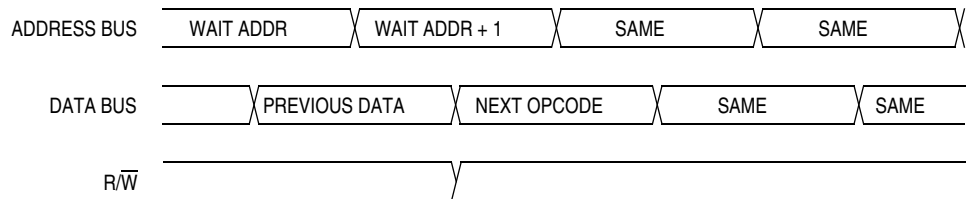
Setting the BCFE bit enables the clearing mechanisms. Once cleared in break mode, a flag remains cleared even when break mode is exited. Status flags with a two-step clearing mechanism — for example, a read of one register followed by the read or write of another — are protected, even when the first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

## 14.7 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.

### 14.7.1 Wait Mode

In wait mode, the CPU clocks are inactive while the peripheral clocks continue to run. **Figure 14-15** shows the timing for wait mode entry.



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

**Figure 14-15. Wait Mode Entry Timing**

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 in emulation mode). A break interrupt during wait mode sets the SIM break stop/wait bit, SBSW, in the break status register (BSR). If the COP disable bit, COPD, in the configuration register is 0, then the computer operating properly module (COP) is enabled and remains active in wait mode.

Figure 14-16 and Figure 14-17 show the timing for wait recovery.

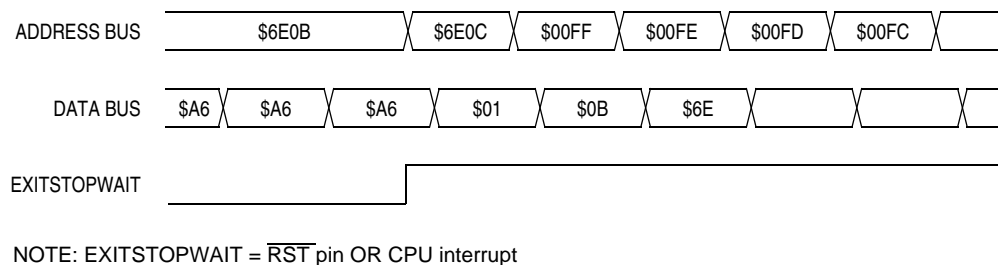
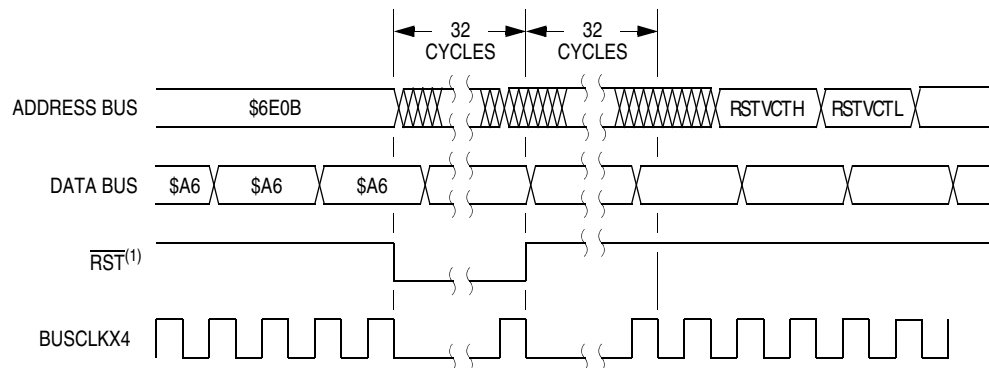


Figure 14-16. Wait Recovery from Interrupt



1.  $\overline{\text{RST}}$  is only available if the RSTEN bit in the CONFIG1 register is set.

Figure 14-17. Wait Recovery from Internal Reset

### 14.7.2 Stop Mode

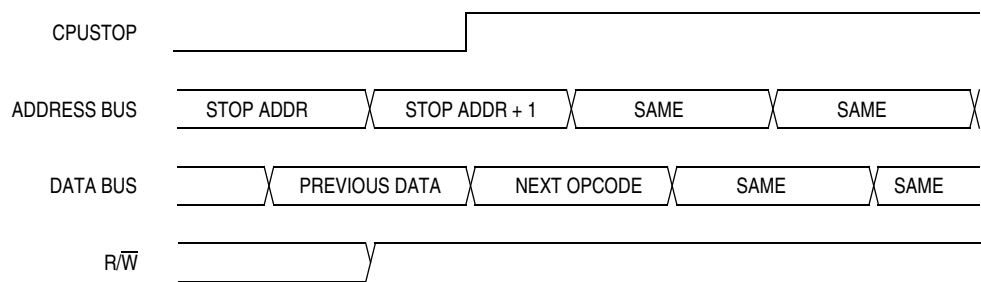
In stop mode, the SIM counter is reset and the system clocks are disabled. An interrupt request from a module can cause an exit from stop mode. Stacking for interrupts begins after the selected stop recovery time has elapsed. Reset or break also causes an exit from stop mode.

The SIM disables the oscillator signals (BUSCLKX2 and BUSCLKX4) in stop mode, stopping the CPU and peripherals. Stop recovery time is selectable using the SSREC bit in the configuration register 1 (CONFIG1). If SSREC is set, stop recovery is reduced from the normal delay of 4096 BUSCLKX4 cycles down to 32. This is ideal for the internal oscillator, RC oscillator, and external oscillator options which 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.

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 14-18 shows stop mode entry timing and Figure 14-19 shows the stop mode recovery time from interrupt or break

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



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

Figure 14-18. Stop Mode Entry Timing

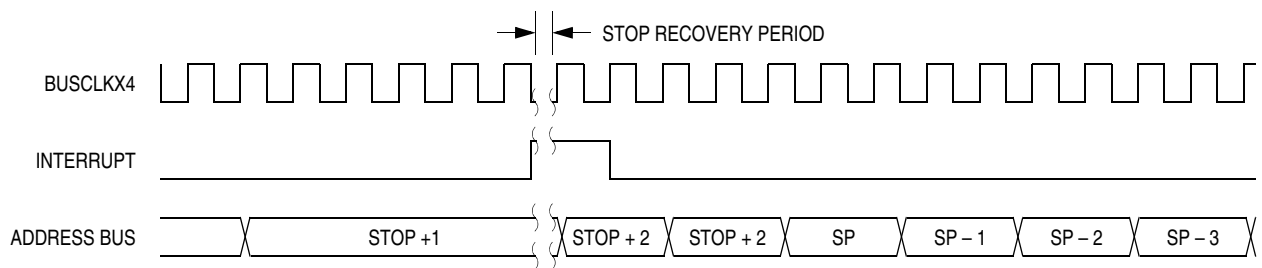


Figure 14-19. Stop Mode Recovery from Interrupt

## 14.8 SIM Registers

The SIM has three memory mapped registers. [Table 14-4](#) shows the mapping of these registers.

**Table 14-4. SIM Registers**


Address	Register	Access Mode
\$FE00	BSR	User
\$FE01	SRSR	User
\$FE03	BFCR	User

### 14.8.1 SIM Reset Status Register

This register contains seven 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	MODRST	LVI	0
Write:								
POR:	1	0	0	0	0	0	0	0

 = Unimplemented

**Figure 14-20. 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 (illegal attempt to fetch an opcode from an unimplemented address)

- 1 = Last reset caused by an opcode fetch from an illegal address
- 0 = POR or read of SRSR

MODRST — Monitor Mode Entry Module Reset Bit

1 = Last reset caused by monitor mode entry when vector locations \$FFFE and \$FFFF are \$FF after POR while  $\overline{IRQ} = V_{DD}$

0 = POR or read of SRSR

LVI — Low Voltage Inhibit Reset bit

1 = Last reset caused by LVI circuit

0 = POR or read of SRSR

### 14.8.2 Break Flag Control Register

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

Address: \$FE03

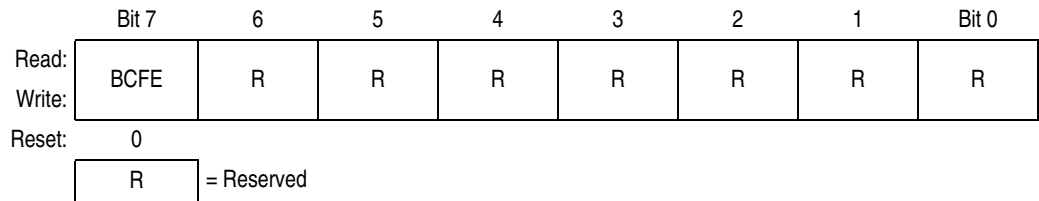


Figure 14-21. Break Flag Control Register (BFCR)

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 15. Timer Interface Module (TIM)

### 15.1 Introduction

This section describes the timer interface module (TIM). The TIM is a two-channel timer that provides a timing reference with input capture, output compare, and pulse-width-modulation functions. [Figure 15-2](#) is a block diagram of the TIM.

### 15.2 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
  - 7-frequency internal bus clock prescaler selection
  - External TIM clock input
- Free-running or modulo up-count operation
- Toggle any channel pin on overflow
- TIM counter stop and reset bits

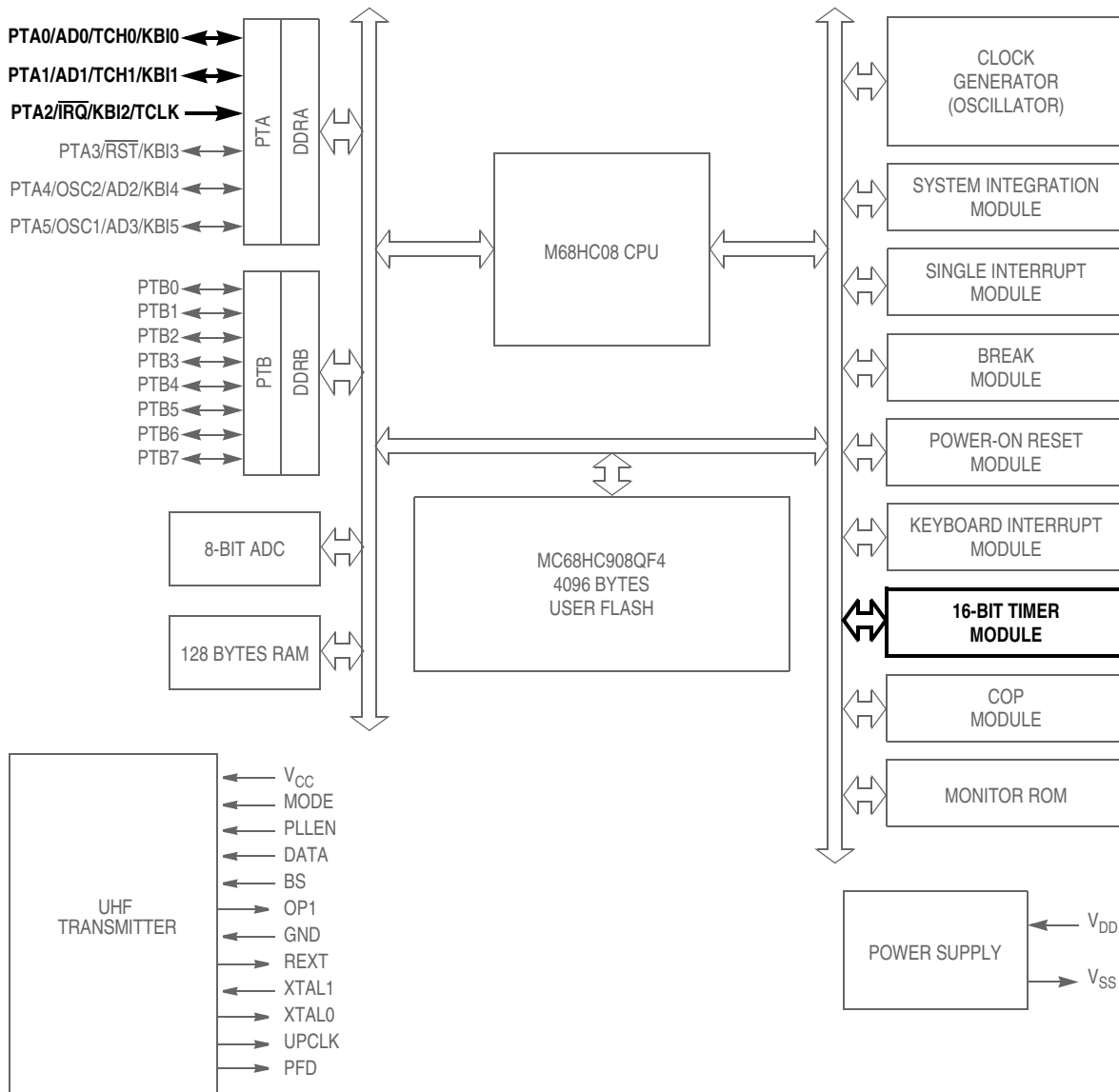
### 15.3 Pin Name Conventions

The TIM shares two input/output (I/O) pins with two port A I/O pins. The full names of the TIM 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**

TIM Generic Pin Names:	TCH0	TCH1	TCLK
Full TIM Pin Names:	PTA0/TCH0	PTA1/TCH1	PTA2/TCLK

Timer Interface Module (TIM)



**RST, IRQ:** Pins have internal (about 30K Ohms) pull up  
**PTA[0:5]:** High current sink and source capability  
**PTA[0:5]:** Pins have programmable keyboard interrupt and pull up

Figure 15-1. Block Diagram Highlighting TIM Block and Pins

15.4 Functional Description

Figure 15-2 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.

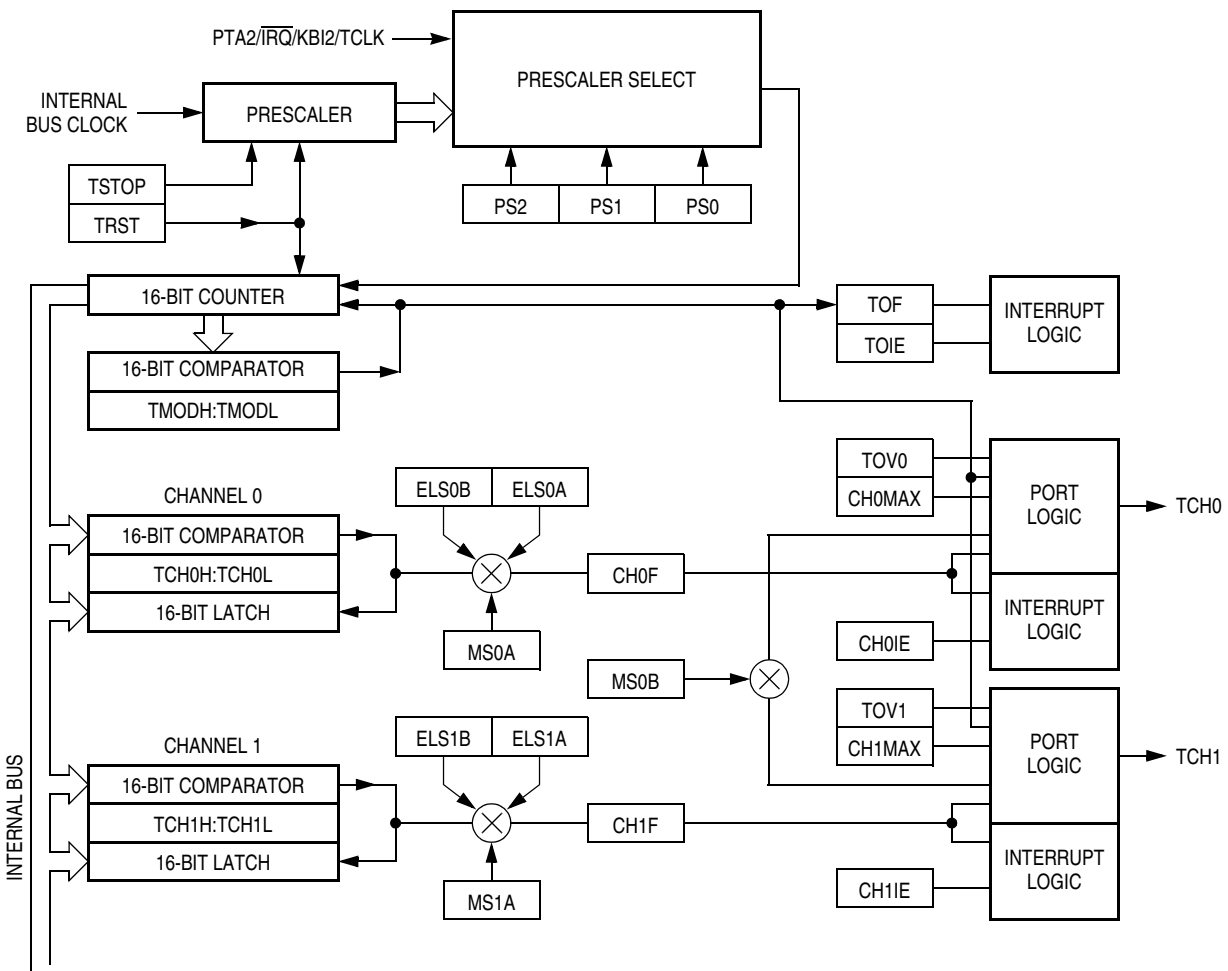


Figure 15-2. TIM Block Diagram

## Timer Interface Module (TIM)

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0		
\$0020	TIM Status and Control Register (TSC) <a href="#">See page 147.</a>	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0	
		Write:	0								
		Reset:	0	0	1	0	0	0	0	0	
\$0021	TIM Counter Register High (TCNTH) <a href="#">See page 149.</a>	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
		Write:									
		Reset:	0	0	0	0	0	0	0	0	
\$0022	TIM Counter Register Low (TCNTL) <a href="#">See page 149.</a>	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
		Write:									
		Reset:	0	0	0	0	0	0	0	0	
\$0023	TIM Counter Modulo Register High (TMODH) <a href="#">See page 149.</a>	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
		Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
		Reset:	1	1	1	1	1	1	1	1	
\$0024	TIM Counter Modulo Register Low (TMODL) <a href="#">See page 149.</a>	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
		Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
		Reset:	1	1	1	1	1	1	1	1	
\$0025	TIM Channel 0 Status and Control Register (TSC0) <a href="#">See page 150.</a>	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX	
		Write:	0								
		Reset:	0	0	0	0	0	0	0	0	
\$0026	TIM Channel 0 Register High (TCH0H) <a href="#">See page 153.</a>	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
		Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
		Reset:	Indeterminate after reset								
\$0027	TIM Channel 0 Register Low (TCH0L) <a href="#">See page 153.</a>	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
		Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
		Reset:	Indeterminate after reset								
\$0028	TIM Channel 1 Status and Control Register (TSC1) <a href="#">See page 150.</a>	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX	
		Write:	0								
		Reset:	0	0	0	0	0	0	0	0	
\$0029	TIM Channel 1 Register High (TCH1H) <a href="#">See page 153.</a>	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
		Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
		Reset:	Indeterminate after reset								
\$002A	TIM Channel 1 Register Low (TCH1L) <a href="#">See page 153.</a>	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
		Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
		Reset:	Indeterminate after reset								

= Unimplemented

**Figure 15-3. TIM I/O Register Summary**

### 15.4.1 TIM Counter Prescaler

The TIM clock source is one of the seven prescaler outputs or the TIM clock pin, TCLK. 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.

### 15.4.2 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 central processor unit (CPU) interrupt requests.

### 15.4.3 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.

#### 15.4.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in [15.4.3 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.

#### 15.4.3.2 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.*

#### 15.4.4 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 15-4](#) shows, the output compare value in the TIM channel registers determines the pulse width of the PWM signal. The time between overflow and output compare is the pulse width. Program the TIM to clear the channel pin on output compare if the state of the PWM pulse is logic 1 (ELSxA = 0). Program the TIM to set the pin if the state of the PWM pulse is logic 0 (ELSxA = 1).

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 [15.9.1 TIM Status and Control Register](#).

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%.

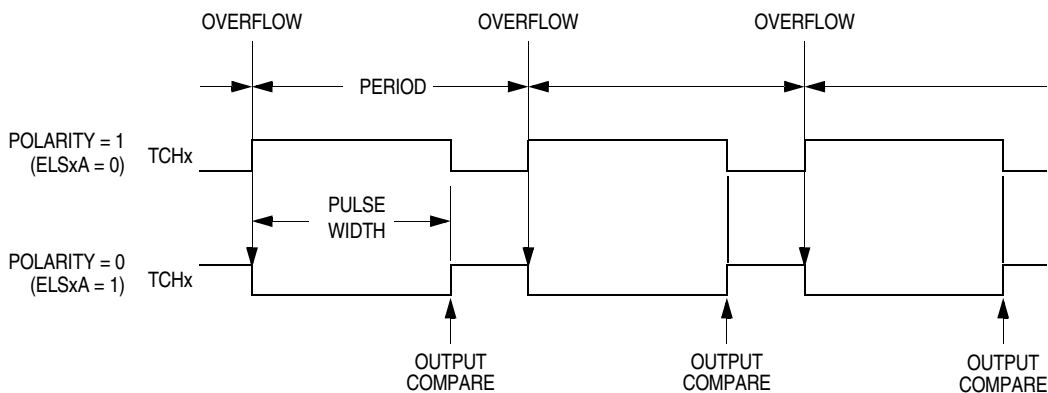


Figure 15-4. PWM Period and Pulse Width

#### 15.4.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in [15.4.4 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.*

## 15.4.4.2 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.*

## 15.4.4.3 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 15-3](#).
  - b. Write 1 to the toggle-on-overflow bit, TOVx.
  - c. Write 1:0 (polarity 1 — to clear output on compare) or 1:1 (polarity 0 — 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 15-3](#).

**NOTE:** *In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error*



or noise. Toggling on output compare can also cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.

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

Setting MS0B links channels 0 and 1 and configures them for buffered PWM operation. The TIM channel 0 registers (TCH0H:TCH0L) initially control the buffered PWM output. TIM status control register 0 (TSCR0) controls and monitors the PWM signal from the linked channels. MS0B takes priority over MS0A.

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 [15.9.4 TIM Channel Status and Control Registers](#).

## 15.5 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.

## 15.6 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.

## 15.7 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 [16.2.2.5 Break Flag Control Register](#).

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

To protect status bits during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a 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 0. After the break, doing the second step clears the status bit.

## 15.8 Input/Output Signals

Port A shares three of its pins with the TIM. Two TIM channel I/O pins are PTA0/TCH0 and PTA1/TCH1 and an alternate clock source is PTA2/TCLK.

### 15.8.1 TIM Clock Pin (PTA2/TCLK)

PTA2/TCLK is an external clock input that can be the clock source for the TIM counter instead of the prescaled internal bus clock. Select the PTA2/TCLK input by writing 1s to the three prescaler select bits, PS[2–0]. (See [15.9.1 TIM Status and Control Register](#).) When the PTA2/TCLK pin is the TIM clock input, it is an input regardless of port pin initialization.

### 15.8.2 TIM Channel I/O Pins (PTA0/TCH0 and PTA1/TCH1)

Each channel I/O pin is programmable independently as an input capture pin or an output compare pin. PTA0/TCH0 can be configured as a buffered output compare or buffered PWM pin.

### 15.9 Input/Output 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)


#### 15.9.1 TIM Status and Control Register

The TIM status and control register (TSC) does the following:

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

Address: \$0020

	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 15-5. 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 0 to TOF. If another TIM overflow occurs before the clearing sequence is complete, then writing 0 to TOF has no effect. Therefore, a TOF interrupt request cannot be lost due to inadvertent clearing of TOF. Reset clears the TOF bit. Writing a 1 to TOF has no effect.

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

#### TOIE — TIM Overflow Interrupt Enable Bit

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

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

TSTOP — TIM Stop Bit

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

- 1 = TIM counter stopped
- 0 = TIM counter active

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

TRST — TIM Reset Bit

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

- 1 = Prescaler and TIM counter cleared
- 0 = No effect

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

PS[2:0] — Prescaler Select Bits

These read/write bits select either the PTA2/TCLK pin or one of the seven prescaler outputs as the input to the TIM counter as Table 15-2 shows. Reset clears the PS[2:0] bits.

**Table 15-2. Prescaler Selection**

PS2	PS1	PS0	TIM Clock Source
0	0	0	Internal bus clock ÷ 1
0	0	1	Internal bus clock ÷ 2
0	1	0	Internal bus clock ÷ 4
0	1	1	Internal bus clock ÷ 8
1	0	0	Internal bus clock ÷ 16
1	0	1	Internal bus clock ÷ 32
1	1	0	Internal bus clock ÷ 64
1	1	1	PTA2/TCLK

15.9.2 TIM Counter Registers

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

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

Address:	\$0021								TCNTH
	Bit 7	6	5	4	3	2	1	Bit 0	
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
Write:									
Reset:	0	0	0	0	0	0	0	0	
Address:	\$0022								TCNTL
	Bit 7	6	5	4	3	2	1	Bit 0	
Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Write:									
Reset:	0	0	0	0	0	0	0	0	


 = Unimplemented

Figure 15-6. TIM Counter Registers (TCNTH:TCNTL)

15.9.3 TIM Counter Modulo Registers

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

Address:	\$0023								TMODH
	Bit 7	6	5	4	3	2	1	Bit 0	
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
Write:									
Reset:	1	1	1	1	1	1	1	1	
Address:	\$0024								TMODL
	Bit 7	6	5	4	3	2	1	Bit 0	
Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Write:									
Reset:	1	1	1	1	1	1	1	1	

Figure 15-7. TIM Counter Modulo Registers (TMODH:TMODL)

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

15.9.4 TIM Channel Status and Control Registers

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

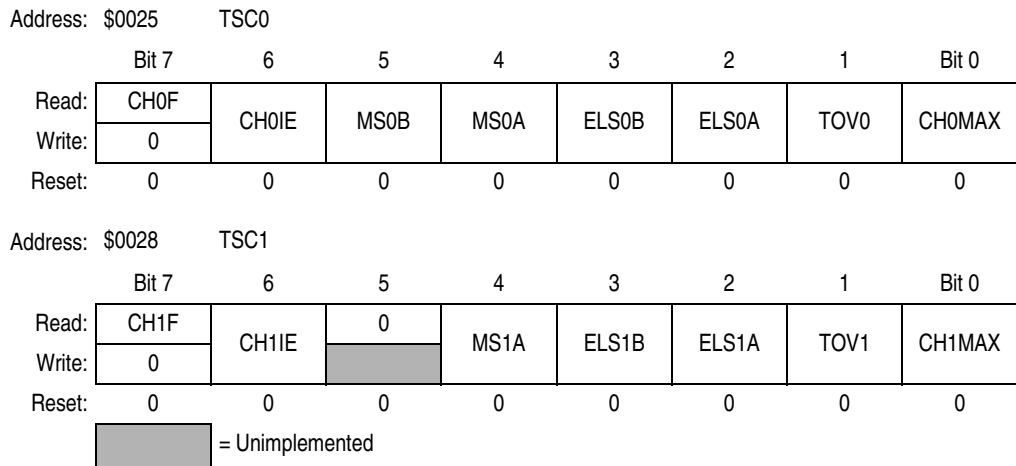


Figure 15-8. TIM Channel Status and Control Registers (TSC0:TSC1)

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.

Clear CHxF by reading the TIM channel x status and control register with CHxF set and then writing a 0 to CHxF. If another interrupt request occurs before the clearing sequence is complete, then writing a 0 to CHxF has no effect.

Therefore, an interrupt request cannot be lost due to inadvertent clearing of CHxF.

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

1 = Input capture or output compare on channel x

0 = No input capture or output compare on channel x

**CHxIE** — Channel x Interrupt Enable Bit

This read/write bit enables TIM CPU interrupt service requests on channel x. Reset clears the CHxIE bit.

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

**MSxB** — Mode Select Bit B

This read/write bit selects buffered output compare/PWM operation. MSxB exists only in the 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 ≠ 00, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation.

See [Table 15-3](#).

- 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 15-3](#)). Reset clears the MSxA bit.

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

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

**Table 15-3. 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	0	Output compare or PWM	Software compare only
0	1	0	1		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

ELSxB and ELSxA — Edge/Level Select Bits

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

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

When ELSxB and ELSxA are both clear, channel x is not connected to an I/O port, and pin TCHx is available as a general-purpose I/O pin. **Table 15-3** shows how ELSxB and ELSxA work. Reset clears the ELSxB and ELSxA bits.

**NOTE:** After initially enabling a TIM channel register for input capture operation and selecting the edge sensitivity, clear CHxF to ignore any erroneous edge detection flags.

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 a 1, setting the CHxMAX bit forces the duty cycle of buffered and unbuffered PWM signals to 100%. As **Figure 15-9** 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.

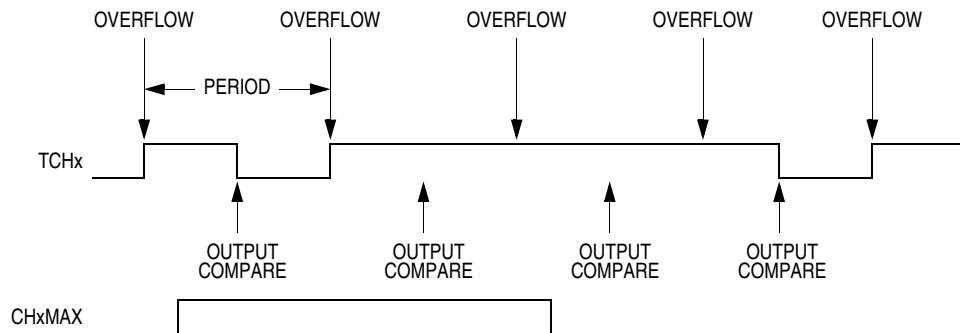


Figure 15-9. CHxMAX Latency



15.9.5 TIM Channel Registers

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

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

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

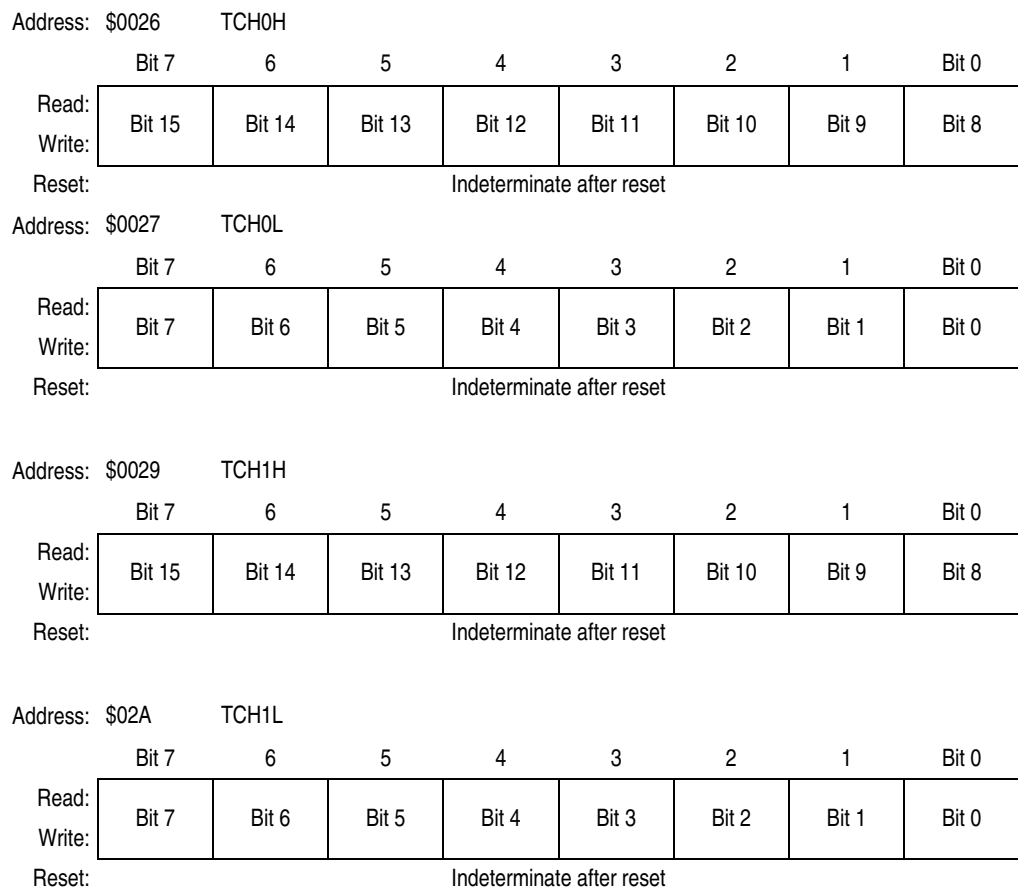


Figure 15-10. TIM Channel Registers (TCH0H/L:TCH1H/L)



## Section 16. Development Support

### 16.1 Introduction

This section describes the break module, the monitor read-only memory (MON), and the monitor mode entry methods.

### 16.2 Break Module (BRK)

The break module can generate a break interrupt that stops normal program flow at a defined address to enter a background program.

Features include:

- Accessible input/output (I/O) registers during the break Interrupt
- Central processor unit (CPU) generated break interrupts
- Software-generated break interrupts
- Computer operating properly (COP) disabling during break interrupts

#### 16.2.1 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal ( $\overline{\text{BKPT}}$ ) to the system integration module (SIM). The SIM then causes the CPU to load the instruction register with a software interrupt instruction (SWI). The program counter vectors to \$FFFC and \$FFFD (\$FEFC and \$FEFD in monitor mode).

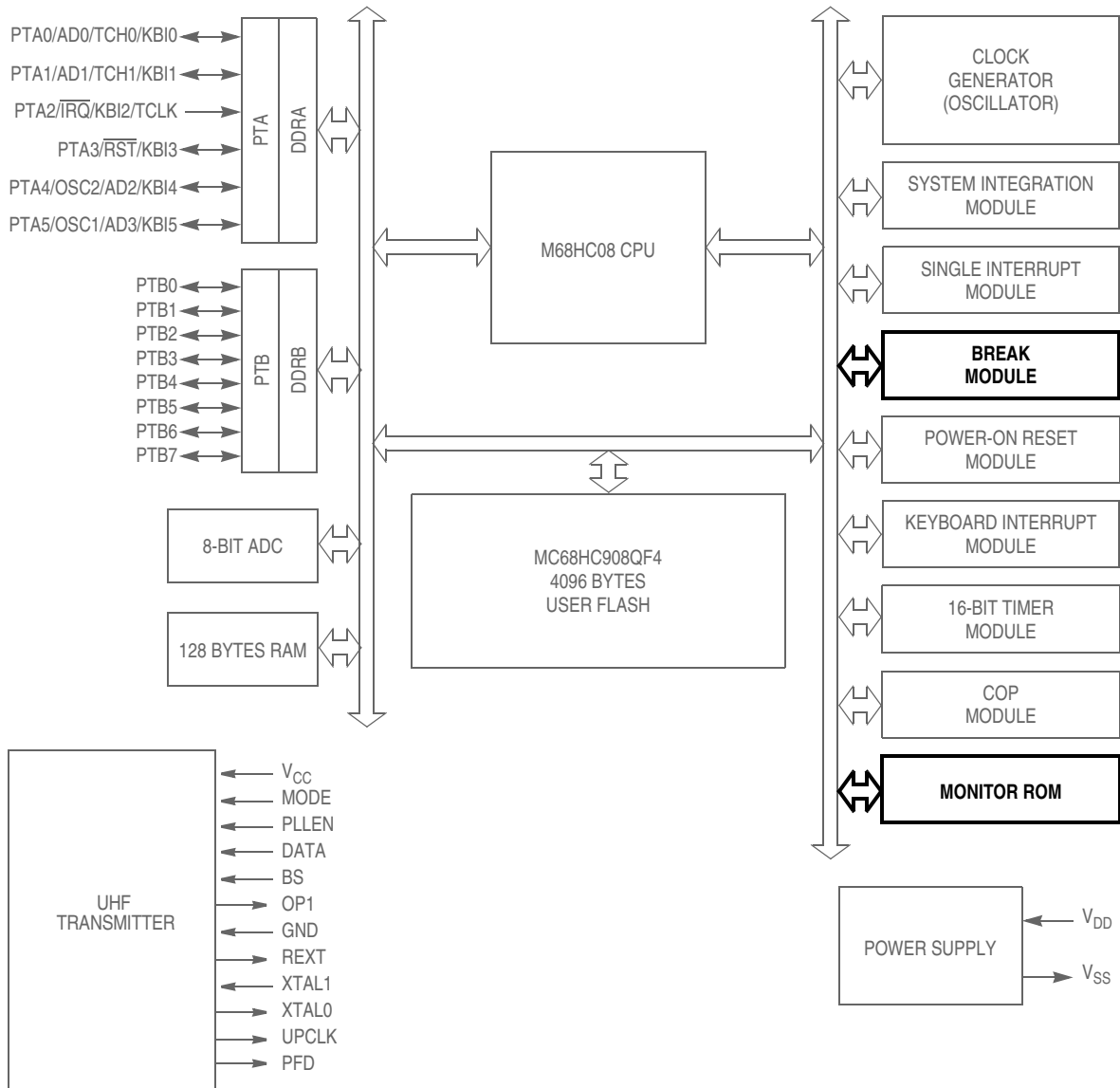
The following events can cause a break interrupt to occur:

- A CPU generated address (the address in the program counter) matches the contents of the break address registers.
- Software writes a 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 is generated. A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the microcontroller unit (MCU) to normal operation.

**Figure 16-2** shows the structure of the break module.

**Figure 16-3** provides a summary of the I/O registers.



$\overline{RST}$ ,  $\overline{IRQ}$ : Pins have internal (about 30K Ohms) pull up  
 PTA[0:5]: High current sink and source capability  
 PTA[0:5]: Pins have programmable keyboard interrupt and pull up

Figure 16-1. Block Diagram Highlighting BRK and MON Blocks

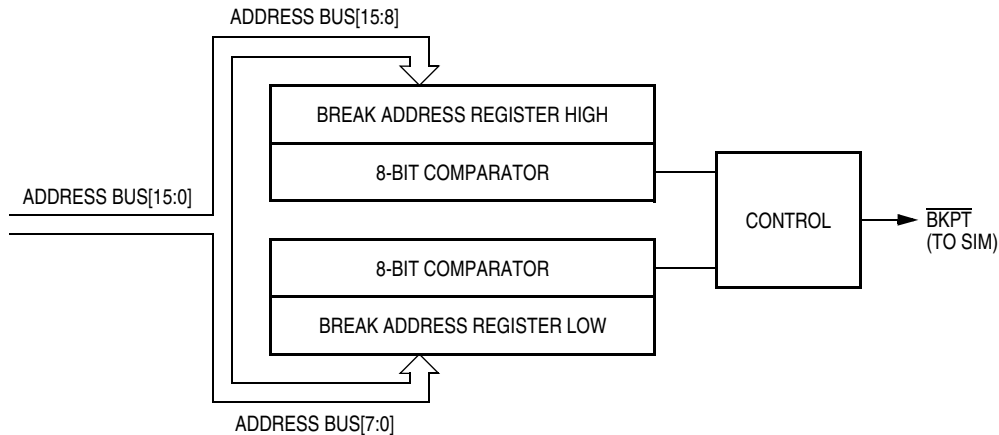


Figure 16-2. Break Module Block Diagram

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0		
\$FE00	Break Status Register (BSR) <a href="#">See page 161.</a>	Read:	R	R	R	R	R	SBSW	R		
		Write:						Note <sup>(1)</sup>			
		Reset:	0								
\$FE02	Break Auxiliary Register (BRKAR) <a href="#">See page 160.</a>	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
		Write:									
		Reset:	0	0	0	0	0	0	0	0	0
\$FE03	Break Flag Control Register (BFCR) <a href="#">See page 161.</a>	Read:	BCFE	R	R	R	R	R	R	R	
		Write:									
		Reset:	0								
\$FE09	Break Address High Register (BRKH) <a href="#">See page 160.</a>	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
		Write:									
		Reset:	0	0	0	0	0	0	0	0	0
\$FE0A	Break Address Low Register (BRKL) <a href="#">See page 160.</a>	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
		Write:									
		Reset:	0	0	0	0	0	0	0	0	0
\$FE0B	Break Status and Control Register (BRKSCR) <a href="#">See page 159.</a>	Read:	BRKE	BRKA	0	0	0	0	0	0	
		Write:									
		Reset:	0	0	0	0	0	0	0	0	0

1. Writing a 0 clears SBSW.

= Unimplemented      R = Reserved

Figure 16-3. Break I/O Register Summary

When the internal address bus matches the value written in the break address registers or when software writes a 1 to the BRKA bit in the break status and control register, 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 timing is:

- When a break address is placed at the address of the instruction opcode, the instruction is not executed until after completion of the break interrupt routine.
- When a break address is placed at an address of an instruction operand, the instruction is executed before the break interrupt.
- When software writes a 1 to the BRKA bit, the break interrupt occurs just before the next instruction is executed.

By updating a break address and clearing the BRKA bit in a break interrupt routine, a break interrupt can be generated continuously.

**CAUTION:** *A break address should be placed at the address of the instruction opcode. When software does not change the break address and clears the BRKA bit in the first break interrupt routine, the next break interrupt will not be generated after exiting the interrupt routine even when the internal address bus matches the value written in the break address registers.*

#### 16.2.1.1 Flag Protection During Break Interrupts

The system integration module (SIM) controls whether or not module status bits 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 [16.2.2.5 Break Flag Control Register](#) and the **Break Interrupts** subsection for each module.

#### 16.2.1.2 TIM During Break Interrupts

A break interrupt stops the timer counter.

#### 16.2.1.3 COP During Break Interrupts

The COP is disabled during a break interrupt with monitor mode when BDCOP bit is set in break auxiliary register (BRKAR).

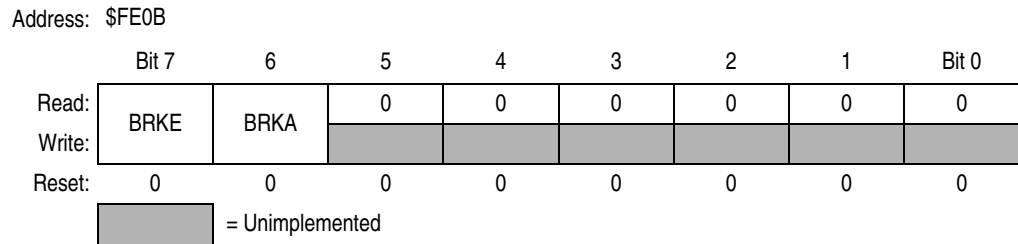
## 16.2.2 Break Module Registers

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)
- Break status register (BSR)
- Break flag control register (BFCR)

### 16.2.2.1 Break Status and Control Register

The break status and control register (BRKSCR) contains break module enable and status bits.



**Figure 16-4. 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 0 to bit 7. Reset clears the BRKE bit.

- 1 = Breaks enabled on 16-bit address match
- 0 = Breaks disabled

#### BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a 1 to BRKA generates a break interrupt. Clear BRKA by writing a 0 to it before exiting the break routine. Reset clears the BRKA bit.

- 1 = Break address match
- 0 = No break address match

16.2.2.2 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.

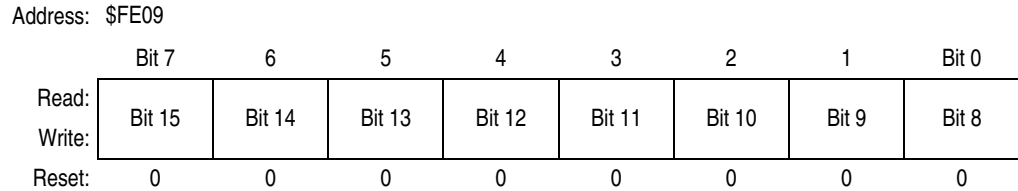


Figure 16-5. Break Address Register High (BRKH)

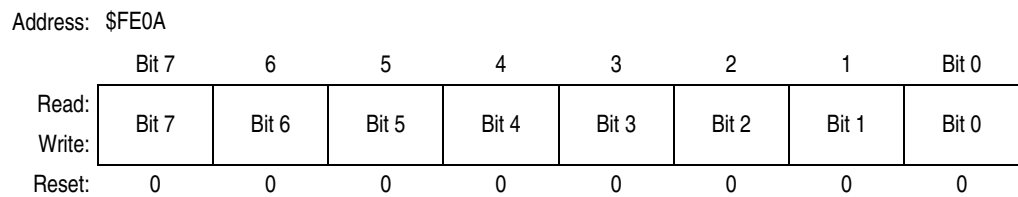


Figure 16-6. Break Address Register Low (BRKL)

16.2.2.3 Break Auxiliary Register

The break auxiliary register (BRKAR) contains a bit that enables software to disable the COP while the MCU is in a state of break interrupt with monitor mode.

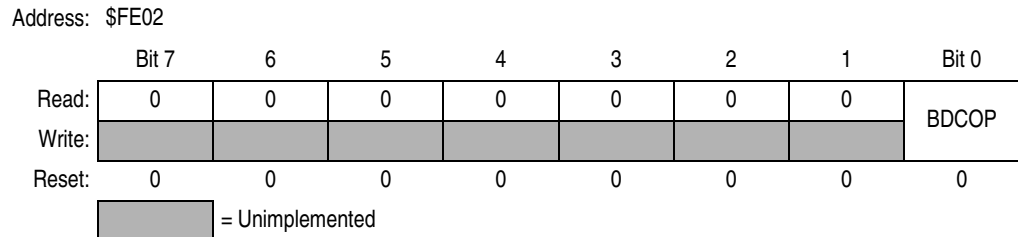


Figure 16-7. Break Auxiliary Register (BRKAR)

BDCOP — Break Disable COP Bit

This read/write bit disables the COP during a break interrupt. Reset clears the BDCOP bit.

- 1 = COP disabled during break interrupt
- 0 = COP enabled during break interrupt.





### 16.2.3 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power- consumption standby modes. If enabled, the break module will remain enabled in wait and stop modes. However, since the internal address bus does not increment in these modes, a break interrupt will never be triggered.

## 16.3 Monitor Module (MON)

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

Features include:

- Normal user-mode pin functionality on most pins
- One pin dedicated to serial communication between MCU and host computer
- Standard non-return-to-zero (NRZ) communication with host computer
- Execution of code in random-access memory (RAM) or FLASH
- FLASH memory security feature<sup>(1)</sup>
- FLASH memory programming interface
- Use of external 9.8304 MHz oscillator to generate internal frequency of 2.4576 MHz
- Simple internal oscillator mode of operation (no external clock or high voltage)
- Monitor mode entry without high voltage,  $V_{TST}$ , if reset vector is blank (\$FFFE and \$FFFF contain \$FF)
- Standard monitor mode entry if high voltage is applied to  $\overline{IRQ}$

### 16.3.1 Functional Description

**Figure 16-10** shows a simplified diagram of monitor mode entry.

The monitor module receives and executes commands from a host computer. **Figure 16-11**, **Figure 16-12**, and **Figure 16-13** show example circuits used to enter monitor mode and communicate with a host computer via a standard RS-232 interface.

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

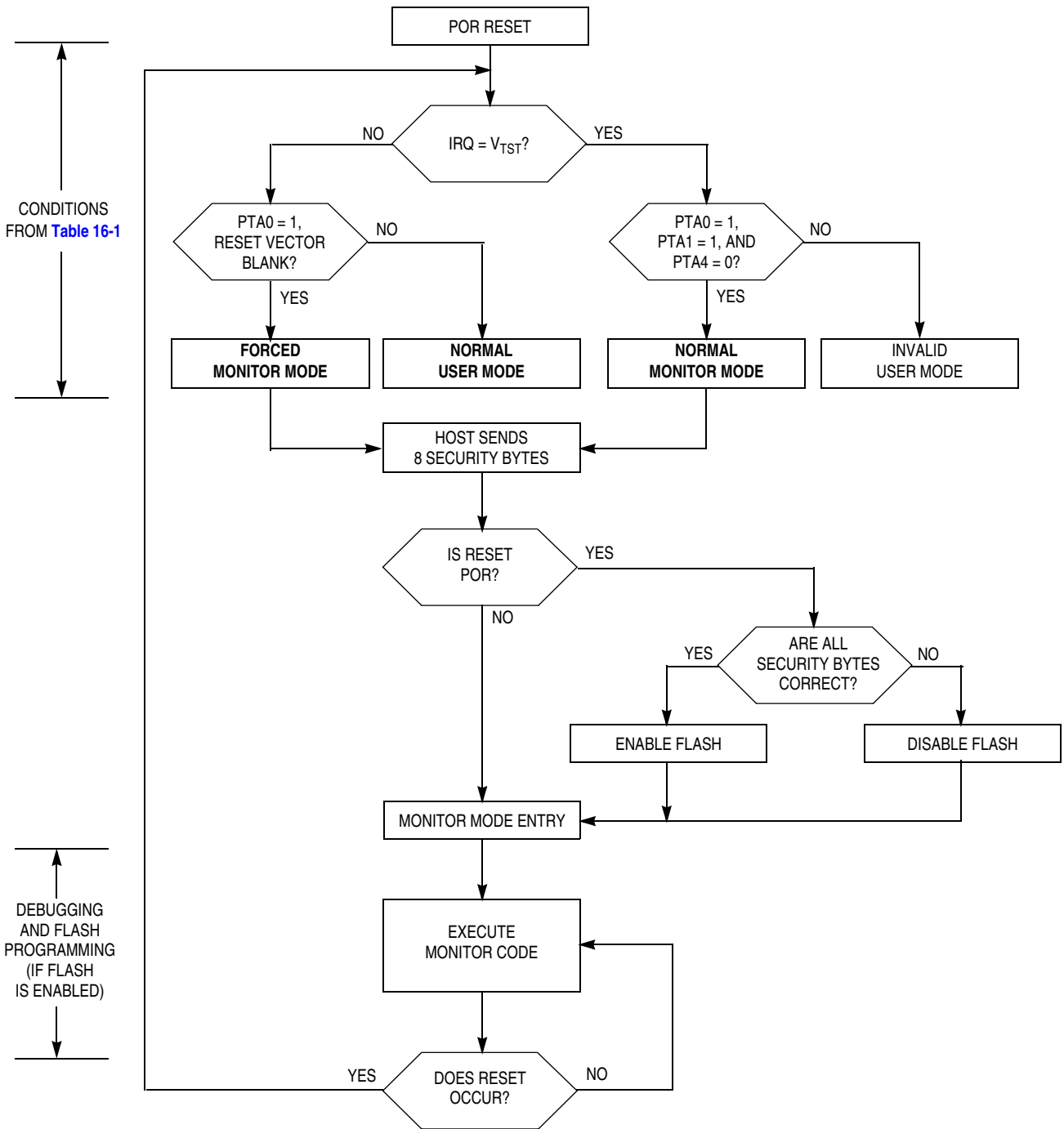


Figure 16-10. Simplified Monitor Mode Entry Flowchart

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

The monitor code has been updated from previous versions of the monitor code to allow enabling the internal oscillator to generate the internal clock. This addition, which is enabled when  $\overline{\text{IRQ}}$  is held low out of reset, is intended to support serial communication/programming at 4800 baud in monitor mode by using the internal oscillator, and the internal oscillator user trim value OSCTRIM (FLASH location \$FFC0, if programmed) to generate the desired internal frequency (1.0 MHz). Since this feature is enabled only when  $\overline{\text{IRQ}}$  is held low out of reset, it cannot be used when the reset vector is programmed (i.e., the value is not \$FFFF) because entry into monitor mode in this case requires  $V_{\text{TST}}$  on  $\overline{\text{IRQ}}$ . The  $\overline{\text{IRQ}}$  pin must remain low during this monitor session in order to maintain communication.

**Table 16-1** shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a power-on reset (POR) and will allow communication at 9600 baud provided one of the following sets of conditions is met:

- If \$FFFE and \$FFFF do not contain \$FF (programmed state):
  - The external clock is 9.8304 MHz
  - $\overline{\text{IRQ}} = V_{\text{TST}}$
- If \$FFFE and \$FFFF contain \$FF (erased state):
  - The external clock is 9.8304 MHz
  - $\overline{\text{IRQ}} = V_{\text{DD}}$  (this can be implemented through the internal  $\overline{\text{IRQ}}$  pullup)
- If \$FFFE and \$FFFF contain \$FF (erased state):
  - $\overline{\text{IRQ}} = V_{\text{SS}}$  (internal oscillator is selected, no external clock required)

The rising edge of the internal  $\overline{\text{RST}}$  signal latches the monitor mode. Once monitor mode is latched, the values on PTA1 and PTA4 pins can be changed.

Once out of reset, the MCU waits for the host to send eight security bytes (see **16.3.2 Security**). After the security bytes, the MCU sends a break signal (10 consecutive logic 0s) to the host, indicating that it is ready to receive a command.

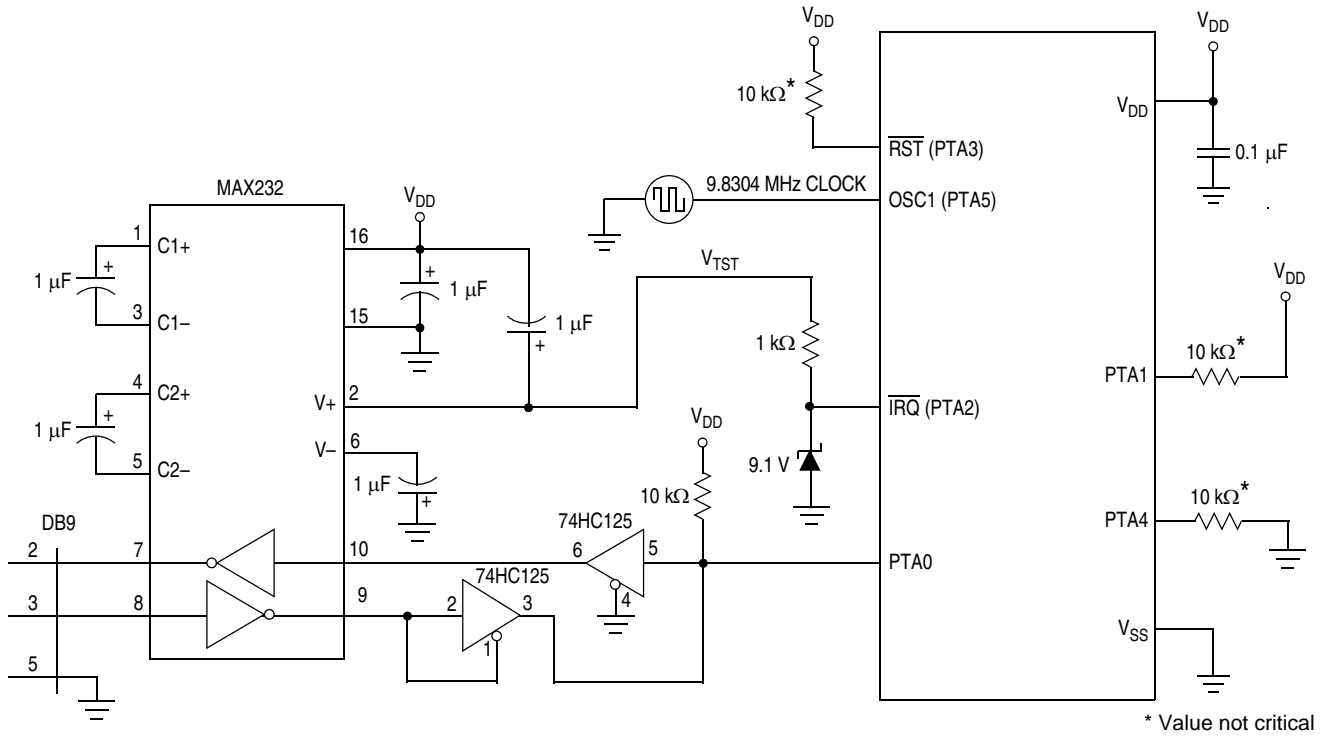


Figure 16-11. Monitor Mode Circuit (External Clock, with High Voltage)

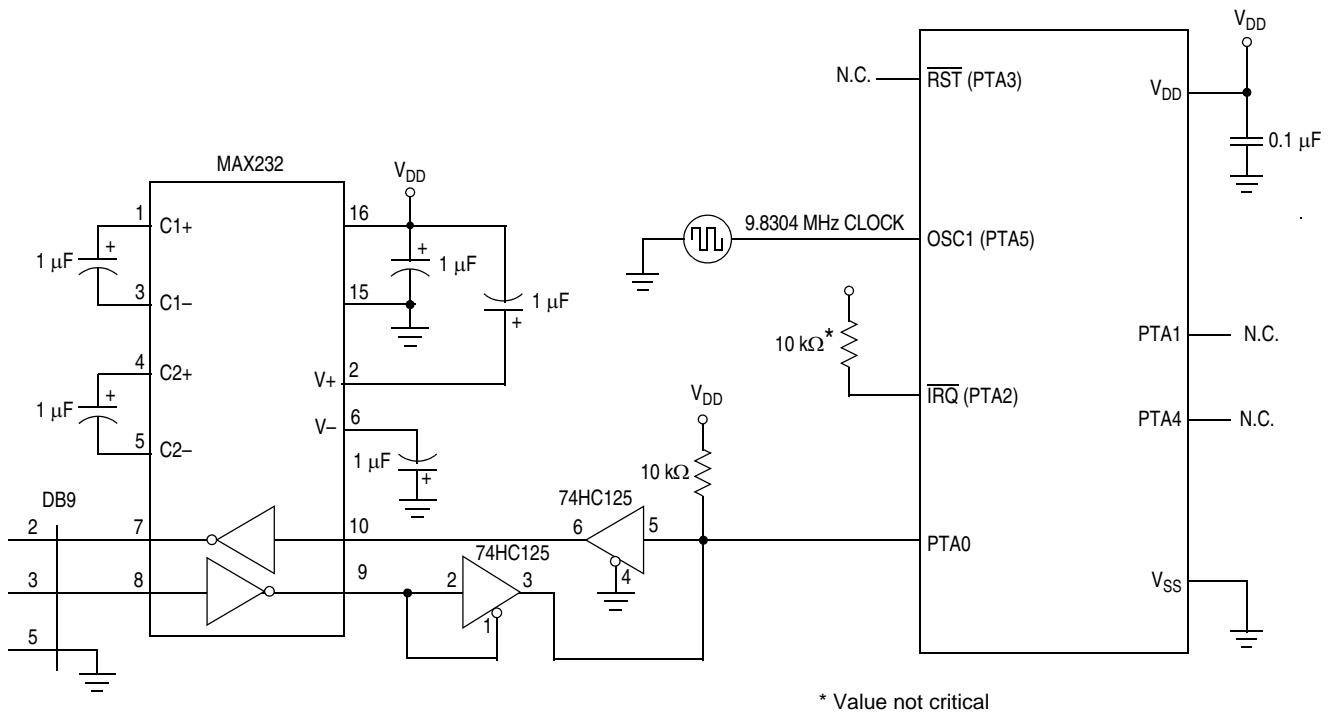


Figure 16-12. Monitor Mode Circuit (External Clock, No High Voltage)

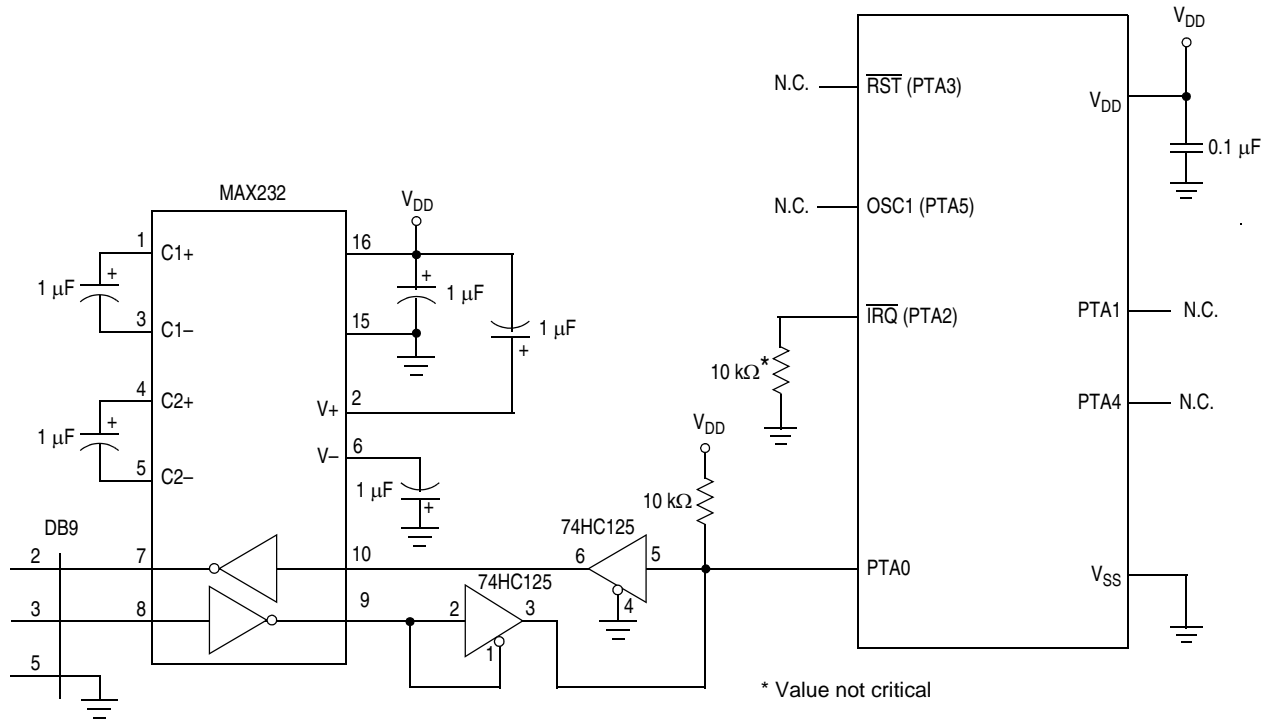


Figure 16-13. Monitor Mode Circuit (Internal Clock, No High Voltage)

### 16.3.1.1 Normal Monitor Mode

$\overline{\text{RST}}$  and OSC1 functions will be active on the PTA3 and PTA5 pins respectively as long as  $V_{\text{TST}}$  is applied to the  $\overline{\text{IRQ}}$  pin. If the  $\overline{\text{IRQ}}$  pin is lowered (no longer  $V_{\text{TST}}$ ) then the chip will still be operating in monitor mode, but the pin functions will be determined by the settings in the configuration registers (see [Section 5. Configuration Register \(CONFIG\)](#)) when  $V_{\text{TST}}$  was lowered. With  $V_{\text{TST}}$  lowered, the BIH and BIL instructions will read the  $\overline{\text{IRQ}}$  pin state only if IRQEN is set in the CONFIG2 register.

If monitor mode was entered with  $V_{\text{TST}}$  on  $\overline{\text{IRQ}}$ , then the COP is disabled as long as  $V_{\text{TST}}$  is applied to  $\overline{\text{IRQ}}$ .

**Table 16-1. Monitor Mode Signal Requirements and Options**

Mode	IRQ (PTA2)	RST (PTA3)	Reset Vector	Serial Communication	Mode Selection		COP	Communication Speed			Comments
					PTA1	PTA4		External Clock	Bus Frequency	Baud Rate	
Normal Monitor	V <sub>TST</sub>	V <sub>DD</sub>	X	PTA0 1	1	0	Disabled	9.8304 MHz	2.4576 MHz	9600	Provide external clock at OSC1.
Forced Monitor	V <sub>DD</sub>	X	\$FFFF (blank)	1	X	X	Disabled	9.8304 MHz	2.4576 MHz	9600	Provide external clock at OSC1.
	V <sub>SS</sub>	X	\$FFFF (blank)	1	X	X	Disabled	X	1.0 MHz (Trimmed)	4800	Internal clock is active.
User	X	X	Not \$FFFF	X	X	X	Enabled	X	X	X	
MON08 Function [Pin No.]	V <sub>TST</sub> [6]	RST [4]	—	COM [8]	MOD0 [12]	MOD1 [10]	—	OSC1 [13]	—	—	

1. PTA0 must have a pullup resistor to V<sub>DD</sub> in monitor mode.
2. Communication speed in the table is an example to obtain a baud rate of 9600. Baud rate using external oscillator is bus frequency / 256 and baud rate using internal oscillator is bus frequency / 206.
3. External clock is a 9.8304 MHz oscillator on OSC1.
4. X = don't care
5. MON08 pin refers to P&E Microcomputer Systems' MON08-Cyclone 2 by 8-pin connector.

NC	1	2	GND
NC	3	4	RST
NC	5	6	IRQ
NC	7	8	PTA0
NC	9	10	PTA4
NC	11	12	PTA1
OSC1	13	14	NC
V <sub>DD</sub>	15	16	NC

## 16.3.1.2 Forced Monitor Mode

If entering monitor mode without high voltage on  $\overline{IRQ}$ , then startup port pin requirements and conditions, (PTA1/PTA4) are not in effect. This is to reduce circuit requirements when performing in-circuit programming.

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

If monitor mode was entered as a result of the reset vector being blank, the COP is always disabled regardless of the state of  $\overline{IRQ}$ .

If the voltage applied to the  $\overline{IRQ}$  is less than  $V_{TST}$ , the MCU will come out of reset in user mode. Internal circuitry monitors the reset vector fetches and will assert an internal reset if it detects that the reset vectors are erased (\$FF). When the MCU comes out of reset, it is forced into monitor mode without requiring high voltage on the  $\overline{IRQ}$  pin. Once out of reset, the monitor code is initially executing with the internal clock at its default frequency.

If  $\overline{IRQ}$  is held high, all pins will default to regular input port functions except for PTA0 and PTA5 which will operate as a serial communication port and OSC1 input respectively (refer to [Figure 16-12](#)). That will allow the clock to be driven from an external source through OSC1 pin.

If  $\overline{IRQ}$  is held low, all pins will default to regular input port function except for PTA0 which will operate as serial communication port. Refer to [Figure 16-13](#).

Regardless of the state of the  $\overline{IRQ}$  pin, it will not function as a port input pin in monitor mode. Bit 2 of the Port A data register will always read 0. The BIH and BIL instructions will behave as if the  $\overline{IRQ}$  pin is enabled, regardless of the settings in the configuration register. See [Section 5. Configuration Register \(CONFIG\)](#).

The COP module is disabled in forced monitor mode. Any reset other than a power-on reset (POR) will automatically force the MCU to come back to the forced monitor mode.

## 16.3.1.3 Monitor Vectors

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

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



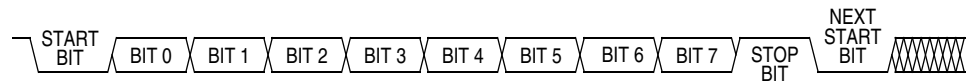
**Table 16-2** summarizes the differences between user mode and monitor mode regarding vectors.

**Table 16-2. Mode Difference**

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

16.3.1.4 Data Format

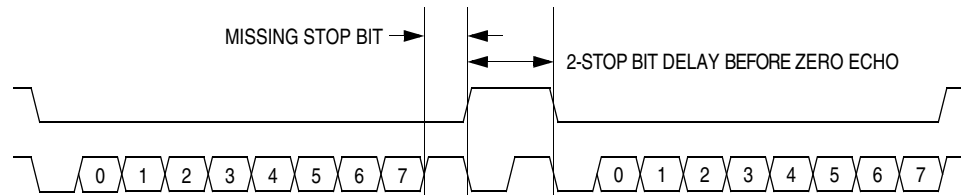
Communication with the monitor ROM is in standard non-return-to-zero (NRZ) mark/space data format. Transmit and receive baud rates must be identical.



**Figure 16-14. Monitor Data Format**

16.3.1.5 Break Signal

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



**Figure 16-15. Break Transaction**

16.3.1.6 Baud Rate

The monitor communication baud rate is controlled by the frequency of the external or internal oscillator and the state of the appropriate pins as shown in **Table 16-1**.

**Table 16-1** also lists the bus frequencies to achieve standard baud rates. The effective baud rate is the bus frequency divided by 256 when using an external oscillator. When using the internal oscillator in forced monitor mode, the effective baud rate is the bus frequency divided by 206.

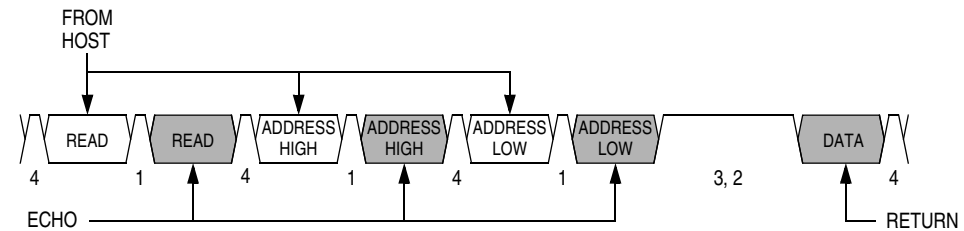
16.3.1.7 Commands

The monitor ROM firmware uses these commands:

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

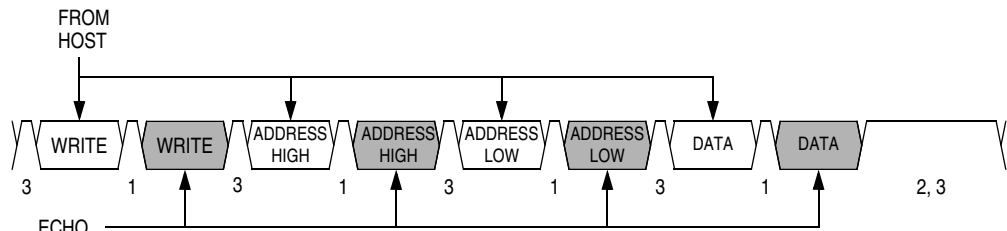
The monitor ROM firmware echoes each received byte back to the PTA0 pin for error checking. An 11-bit delay at the end of each command allows the host to send a break character to cancel the command. A delay of two bit times occurs before each echo and before READ, IREAD, or READSP data is returned.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

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



Notes:  
 1 = Echo delay, approximately 2 bit times  
 2 = Data return delay, approximately 2 bit times  
 3 = Cancel command delay, 11 bit times  
 4 = Wait 1 bit time before sending next byte.

Figure 16-16. Read Transaction



Notes:  
 1 = Echo delay, approximately 2 bit times  
 2 = Cancel command delay, 11 bit times  
 3 = Wait 1 bit time before sending next byte.

Figure 16-17. Write Transaction

A brief description of each monitor mode command is given in [Table 16-3](#) through [Table 16-8](#).

**Table 16-3. READ (Read Memory) Command**

<b>Description</b>	Read byte from memory
<b>Operand</b>	2-byte address in high-byte:low-byte order
<b>Data Returned</b>	Returns contents of specified address
<b>Opcode</b>	\$4A
<b>Command Sequence</b>	
<p>The diagram illustrates the command sequence for the READ command. It starts with a 'READ' signal sent to the monitor. This is followed by an 'ECHO' signal that repeats the 'READ' signal. Next, the 'ADDRESS HIGH' signal is sent to the monitor, followed by its 'ECHO'. Then, the 'ADDRESS LOW' signal is sent to the monitor, followed by its 'ECHO'. Finally, the 'DATA' signal is returned from the monitor.</p>	

**Table 16-4. WRITE (Write Memory) Command**

<b>Description</b>	Write byte to memory
<b>Operand</b>	2-byte address in high-byte:low-byte order; low byte followed by data byte
<b>Data Returned</b>	None
<b>Opcode</b>	\$49
<b>Command Sequence</b>	
<p>The diagram illustrates the command sequence for the WRITE command. It starts with a 'WRITE' signal sent from the host. This is followed by an 'ECHO' signal that repeats the 'WRITE' signal. Next, the 'ADDRESS HIGH' signal is sent from the host, followed by its 'ECHO'. Then, the 'ADDRESS LOW' signal is sent from the host, followed by its 'ECHO'. Finally, the 'DATA' signal is sent from the host, followed by its 'ECHO'.</p>	

**Table 16-5. IREAD (Indexed Read) Command**

<b>Description</b>	Read next 2 bytes in memory from last address accessed
<b>Operand</b>	2-byte address in high byte:low byte order
<b>Data Returned</b>	Returns contents of next two addresses
<b>Opcode</b>	\$1A
<b>Command Sequence</b>	
<p>The diagram illustrates the command sequence for the IREAD command. It starts with an 'IREAD' signal sent from the host. This is followed by an 'ECHO' signal that repeats the 'IREAD' signal. Then, two 'DATA' signals are returned from the monitor.</p>	

Table 16-6. IWRITE (Indexed Write) Command

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

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

Table 16-7. READSP (Read Stack Pointer) Command

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

Table 16-8. RUN (Run User Program) Command

<b>Description</b>	Executes PULH and RTI instructions
<b>Operand</b>	None
<b>Data Returned</b>	None
<b>Opcode</b>	\$28
<b>Command Sequence</b>	

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

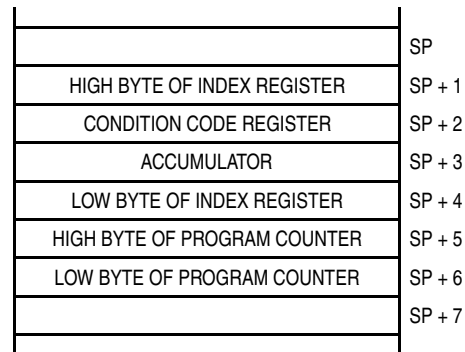


Figure 16-18. Stack Pointer at Monitor Mode Entry

16.3.2 Security

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

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

During monitor mode entry, the MCU waits after the power-on reset for the host to send the eight security bytes on pin PTA0. If the received bytes match those at locations \$FFF6–\$FFFD, the host bypasses the security feature and can read all FLASH locations and execute code from FLASH. Security remains bypassed until a power-on reset occurs. If the reset was not a power-on reset, security remains bypassed and security code entry is not required. See Figure 16-19.

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

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

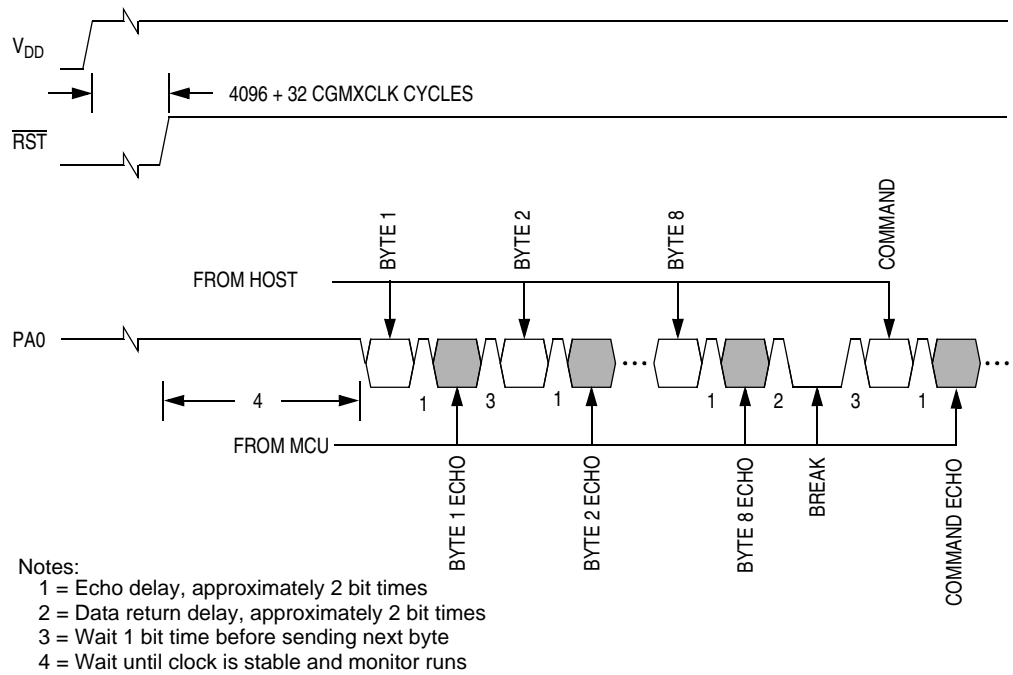


Figure 16-19. Monitor Mode Entry Timing

To determine whether the security code entered is correct, check to see if bit 6 of RAM address \$80 is set. If it is, then the correct security code has been entered and FLASH can be accessed.

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

## Section 17. Electrical Specifications

### 17.1 Introduction

This section contains electrical and timing specifications.

### 17.2 Absolute Maximum Ratings

Maximum ratings are the extreme limits to which the microcontroller unit (MCU) can be exposed without permanently damaging it.

**NOTE:** *This device is not guaranteed to operate properly at the maximum ratings. Refer to [17.5 DC Electrical Characteristics](#) for guaranteed operating conditions.*

Characteristic <sup>(1)</sup>	Symbol	Value	Unit
Supply voltage	$V_{DD}$	-0.3 to +6.0	V
Input voltage	$V_{IN}$	$V_{SS} - 0.3$ to $V_{DD} + 0.3$	V
Mode entry voltage, $\overline{IRQ}$ pin	$V_{TST}$	$V_{SS} - 0.3$ to +9.1	V
Maximum current per pin excluding PTA0–PTA5, $V_{DD}$ , and $V_{SS}$	I	±15	mA
Maximum current for pins PTA0–PTA5	$I_{PTA0-IPTA5}$	±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

1. Voltages references 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  $V_{DD}$ .)*

17.3 Functional Operating Range

Characteristic	Symbol	Value	Unit	Temp Code
Operating temperature range ( $T_L$ to $T_H$ )	$T_A$	-40 to 85 0 to 70	°C	C —
Operating voltage range <sup>(1)</sup> ( $V_{DDMIN}$ to $V_{DDMAX}$ ) -40 to 85°C 0 to 70°C	$V_{DD}$	2.4 to 3.6 2.2 to 3.6	V	C —

1.  $V_{DD}$  must be above  $V_{TRIPR}$  upon power on.

17.4 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance 32-pin TQFP	$\theta_{JA}$	72	°C/W
I/O pin power dissipation	$P_{I/O}$	User determined	W
Power dissipation <sup>(1)</sup>	$P_D$	$P_D = (I_{DD} \times V_{DD})$ $+ P_{I/O} = K/(T_J + 273^\circ\text{C})$	W
Constant <sup>(2)</sup>	K	$P_D \times (T_A + 273^\circ\text{C})$ $+ P_D^2 \times \theta_{JA}$	W/°C
Average junction temperature	$T_J$	$T_A + (P_D \times \theta_{JA})$	°C
Maximum junction temperature	$T_{JM}$	150	°C

1. Power dissipation is a function of temperature.
2. K constant unique to the device. K can be determined for a known  $T_A$  and measured  $P_D$ . With this value of K,  $P_D$  and  $T_J$  can be determined for any value of  $T_A$ .



17.5 DC Electrical Characteristics

Characteristic <sup>(1)</sup>	Symbol	Min	Typ <sup>(2)</sup>	Max	Unit
Output high voltage (for $V_{DD} > 2.7$ V) $I_{Load} = -4$ mA $I_{Load} = -10$ mA, PTA0, PTA1, PTA3–PTA5 only	$V_{OH}$	$V_{DD}-0.8$ $V_{DD}-0.8$	— —	— —	V
Output high voltage (for $V_{DDMIN} < V_{DD} < V_{DDMAX}$ ) $I_{Load} = -2$ mA $I_{Load} = -5$ mA, PTA0, PTA1, PTA3–PTA5 only	$V_{OH}$	$V_{DD}-0.8$ $V_{DD}-0.8$	— —	— —	V
Output low voltage (for $V_{DD} > 2.7$ V) $I_{Load} = 4$ mA $I_{Load} = 10$ mA, PTA0, PTA1, PTA3–PTA5 only	$V_{OL}$	— —	— —	0.8 0.8	V
Output low voltage (for $V_{DDMIN} < V_{DD} < V_{DDMAX}$ ) $I_{Load} = 2$ mA $I_{Load} = 5$ mA, PTA0, PTA1, PTA3–PTA5 only	$V_{OL}$	— —	— —	0.8 0.8	V
Maximum combined $I_{OH}$ (all I/O pins)	$I_{OHT}$	—	—	50	mA
Maximum combined $I_{OL}$ (all I/O pins)	$I_{OLT}$	—	—	50	mA
Input high voltage PTA0–PTA5, PTB0–PTB7	$V_{IH}$	$0.7 \times V_{DD}$	—	$V_{DD}$	V
Input low voltage PTA0–PTA5, PTB0–PTB7	$V_{IL}$	$V_{SS}$	—	$0.3 \times V_{DD}$	V
Input hysteresis	$V_{HYS}$	$0.06 \times V_{DD}$	—	—	V
DC injection current, all ports	$I_{INJ}$	-2	—	+2	mA
Total dc current injection (sum of all I/O)	$I_{INJTOT}$	-25	—	+25	mA
Digital I/O ports Hi-Z leakage current Typical at 25°C	$I_{IL}$	-1 —	— $\pm 0.1$	+1 —	$\mu$ A
Digital input only ports leakage current (PA2/IRQ/KBI2)	$I_{IN}$	-1	—	+1	$\mu$ A
Capacitance Ports (as input) Ports (as output)	$C_{IN}$ $C_{OUT}$	— —	— —	12 8	pF
POR rearm voltage <sup>(3)</sup>	$V_{POR}$	0	—	100	mV
POR rise time ramp rate <sup>(4)</sup>	$R_{POR}$	0.035	—	—	V/ms
Monitor mode entry voltage	$V_{TST}$	$V_{DD} + 2.5$	—	9.1	V
Pullup resistors <sup>(5)</sup> PTA0–PTA5, PTB0–PTB7	$R_{PU}$	16	26	36	k $\Omega$

— Continued on next page

Electrical Specifications

Characteristic <sup>(1)</sup>	Symbol	Min	Typ <sup>(2)</sup>	Max	Unit
Low-voltage inhibit reset, trip falling voltage (LVR)	$V_{TRIPF}$	2.00	2.12	2.24	V
Low-voltage inhibit reset, trip rising voltage (LVR)	$V_{TRIPR}$	2.04	2.18	2.30	V
Low-voltage inhibit reset/recover hysteresis	$V_{HYS}$	—	60	—	mV
Low-voltage detect, trip falling voltage (LVD)	$V_{DTRIPF}$	2.20	2.32	2.44	V
Low-voltage detect, trip rising voltage (LVD)	$V_{DTRIPR}$	2.21	2.33	2.45	V
Low-voltage detect reset/recover hysteresis	$V_{DHYS}$	—	10	—	mV

- $V_{DD} = V_{DDMIN}$  to  $V_{DDMAX}$ ,  $V_{SS} = 0$  Vdc,  $T_A = T_L$  to  $T_H$ , unless otherwise noted.
- Typical values reflect average measurements at  $V_{DD} = 3.0$  V, 25°C only.
- Maximum is highest voltage that POR is guaranteed.
- If minimum  $V_{DD}$  is not reached before the internal POR reset is released, the LVI will hold the part in reset until minimum  $V_{DD}$  is reached.
- $R_{PU}$  is measured at  $V_{DD} = 3.0$  V.

17.6 Control Timing

Characteristic <sup>(1)</sup>	Symbol	Min	Max	Unit
Internal operating frequency	$f_{OP}$ ( $f_{Bus}$ )	—	2	MHz
Internal clock period ( $1/f_{OP}$ )	$t_{cyc}$	500	—	ns
$\overline{RST}$ input pulse width low	$t_{RL}$	400	—	ns
$\overline{IRQ}$ interrupt pulse width low (edge-triggered)	$t_{LIH}$	400	—	ns
$\overline{IRQ}$ interrupt pulse period	$t_{LIL}$	Note <sup>(2)</sup>	—	$t_{cyc}$

- $V_{DD} > 2.2$  V,  $V_{SS} = 0$  Vdc; timing shown with respect to 20%  $V_{DD}$  and 70%  $V_{DD}$  unless otherwise noted.
- The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1  $t_{cyc}$ .

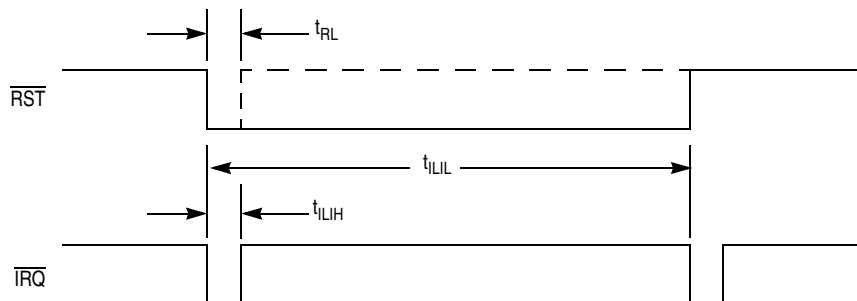


Figure 17-1.  $\overline{RST}$  and  $\overline{IRQ}$  Timing

17.7 Typical 3.0-V Output Drive Characteristics

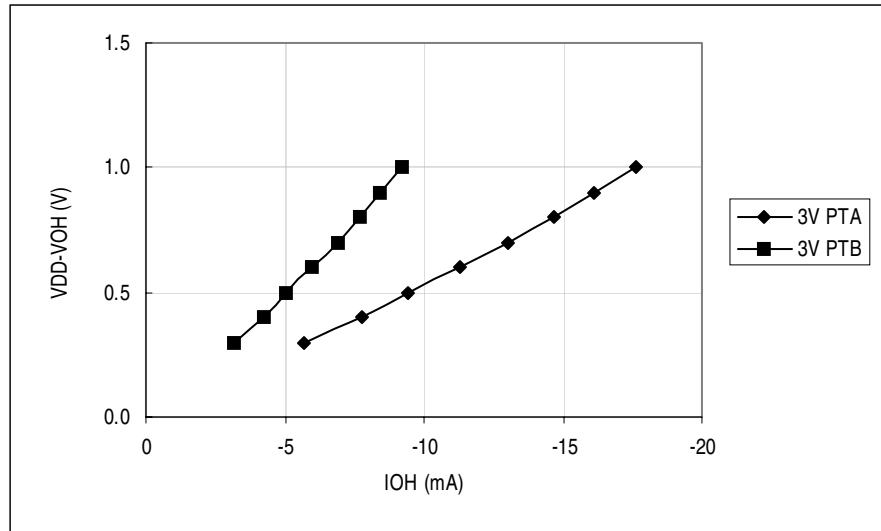


Figure 17-2. Typical 3-Volt Output High Voltage versus Output High Current (25°C)

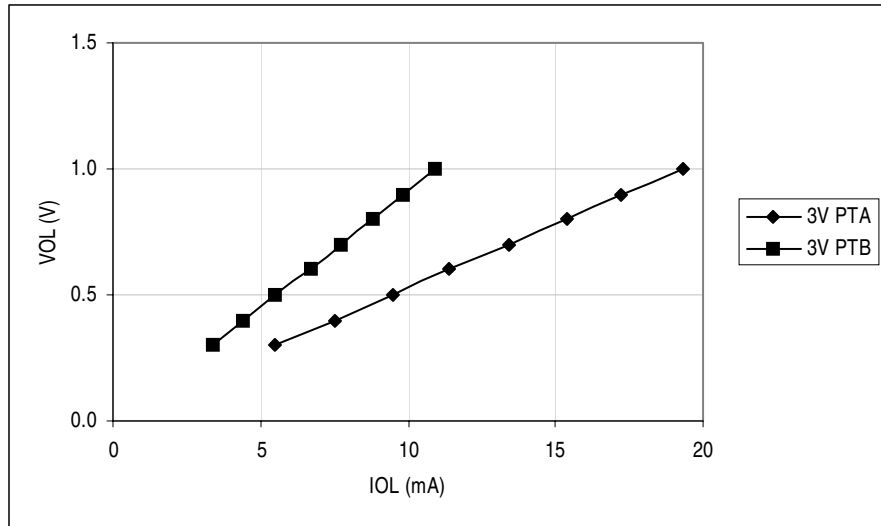


Figure 17-3. Typical 3-Volt Output Low Voltage versus Output Low Current (25°C)

17.8 Oscillator Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Internal oscillator frequency <sup>(1)</sup>	$f_{INTCLK}$	—	4.0	—	MHz
Crystal frequency, XTALCLK <sup>(1)</sup>	$f_{OSCCLK}$	30	32.768	100	kHz
External RC oscillator frequency, RCCLK <sup>(1)</sup>	$f_{RCCLK}$	2	—	8	MHz
External clock reference frequency <sup>(1), (2)</sup>	$f_{OSCCLK}$	dc	—	8	MHz
Crystal load capacitance <sup>(3)</sup>	$C_L$	—	12.5	—	pF
Crystal fixed capacitance <sup>(3)</sup>	$C_1$	—	$2 \times C_L$	—	—
Crystal tuning capacitance <sup>(3)</sup>	$C_2$	—	$2 \times C_L$	—	—
Feedback bias resistor	$R_B$	—	10	—	M $\Omega$
Series resistor	$R_S$	270	330	360	k $\Omega$
RC oscillator external resistor	$R_{EXT}$	See <a href="#">Figure 17-4</a>			—

1. Bus frequency,  $f_{OP}$ , is oscillator frequency divided by 4.
2. No more than 10% duty cycle deviation from 50%.
3. Consult crystal vendor data sheet.

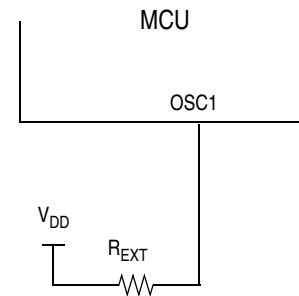
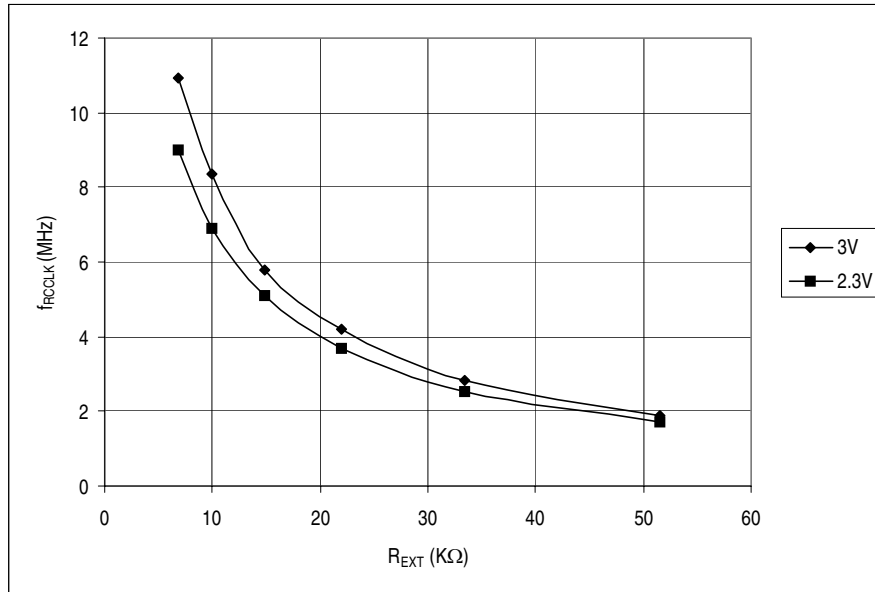


Figure 17-4. Typical RC Oscillator Frequency versus  $R_{EXT}$  (25°C)

17.9 Supply Current Characteristics

Characteristic	Voltage	Bus Freq. (MHz)	Symbol	Typ	Max	Unit	
Run mode $V_{DD}$ supply current <sup>(1)</sup>	3.0	1	$R_{I_{DD}}$	1.5	2.5	mA	
	2.2	1		1.0	1.5		
WAIT mode $V_{DD}$ supply current <sup>(2)</sup>	3.0	1	$W_{I_{DD}}$	1.2	2.0	mA	
	2.2	1		1.0	1.0		
Stop mode $V_{DD}$ supply current <sup>(3)</sup>	3.0		$S_{I_{DD}}$	0.006	—	$\mu$ A	
				0 to 70°C	0.08		—
				-40 to 85°C	0.12		2.0
	25°C with auto wake-up enabled			5.70	—		
	Incremental current with LVI enabled at 25°C			110	—		
	2.2			0.005	—		$\mu$ A
0 to 70°C	0.08	—					
-40 to 85°C	0.12	1.0					
25°C with auto wake-up enabled	1.30	—					
Incremental current with LVI enabled at 25°C	100	—					

1. 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. All ports configured as inputs. Measured with all modules except ADC enabled.
2. Wait (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. All ports configured as inputs. Measured with all modules except ADC enabled.
3. Stop  $I_{DD}$  measured with all ports driven 0.2 V or less from rail. No dc loads. On the 8-pin versions, port B is configured as inputs with pullups enabled.

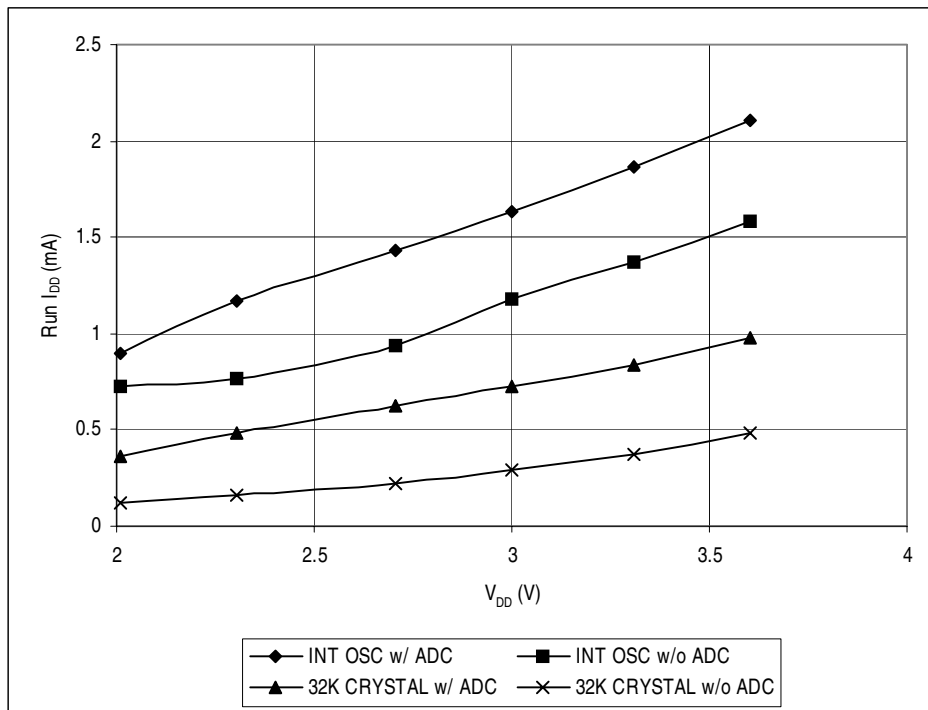


Figure 17-5. Typical Run Current versus  $V_{DD}$  (25°C)  
( $f_{BUS} = 1$  MHz for Internal Oscillator,  $f_{BUS} = 8$  kHz for Crystal Oscillator)

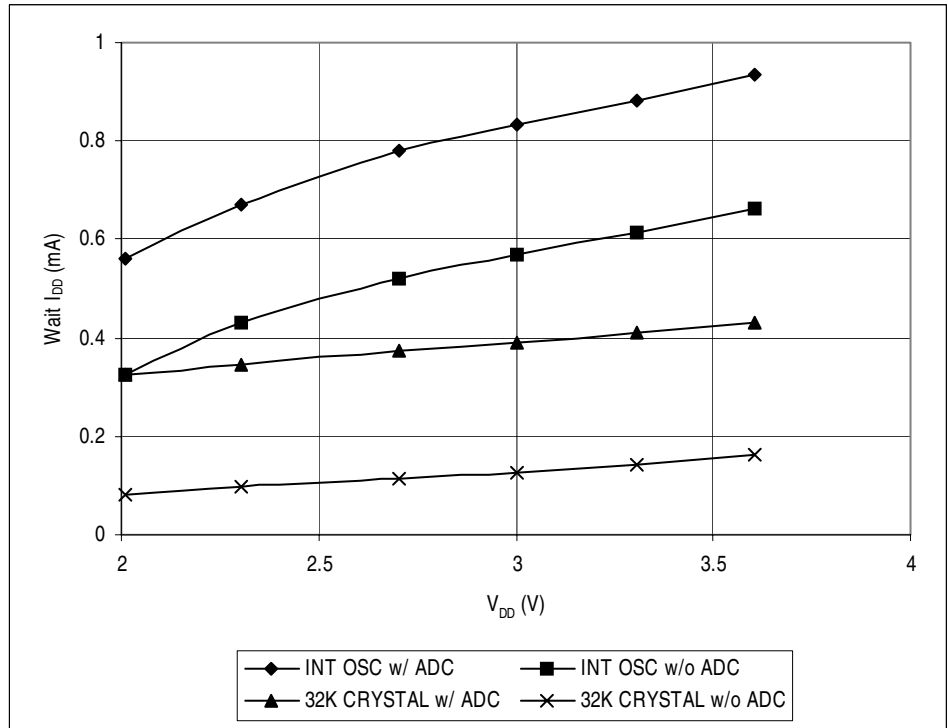


Figure 17-6. Typical Wait Current versus V<sub>DD</sub> (25°C)  
 f<sub>Bus</sub> = 1 MHz for Internal Oscillator, f<sub>Bus</sub> = 8 kHz for Crystal Oscillator)

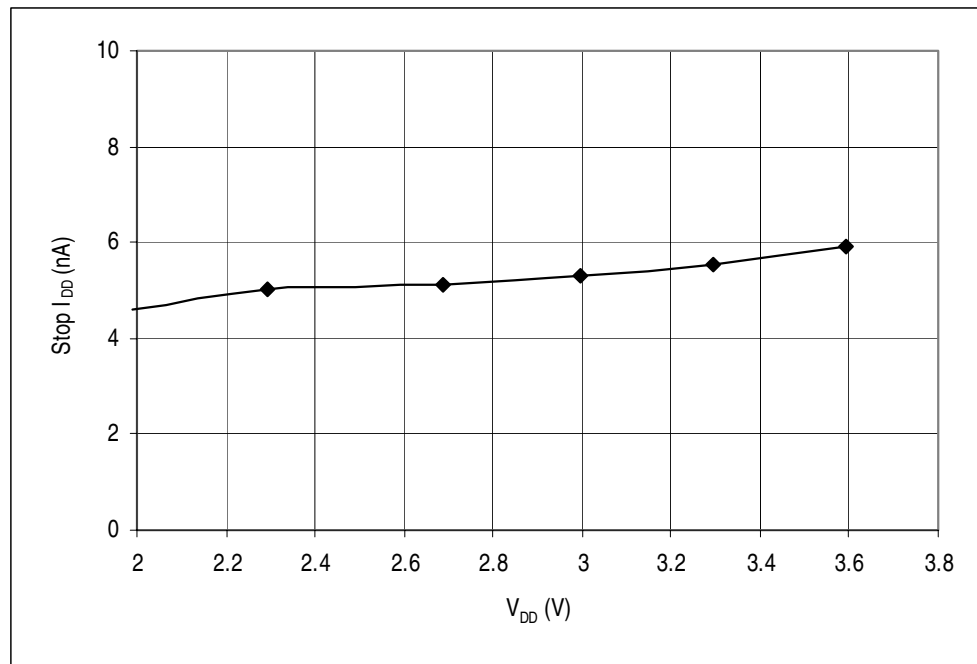


Figure 17-7. Typical Stop Current versus V<sub>DD</sub> (25°C)

## 17.10 Analog-to-Digital (ADC) Converter Characteristics

### 17.10.1 ADC Electrical Operating Conditions

The ADC accuracy characteristics below are guaranteed over two operating conditions as stated here.

	Characteristic	Symbol	Min	Max	Unit
<b>Condition A</b>	ATD supply	$V_{DD}$	2.7	3.6	V
	ADC internal clock	$f_{ADIC}$	0.008	1	MHz
	Ambient temperature	$T_A$	$T_L$	$T_H$	°C
<b>Condition B</b>	ATD supply	$V_{DD}$	2.3	2.7	V
	ADC internal clock	$f_{ADIC}$	8	63	kHz
	Ambient temperature	$T_A$	0	$T_H$	°C

### 17.10.2 ADC Performance Characteristics

Characteristic	Symbol	Min	Max	Unit	Comments
Input voltages	$V_{ADIN}$	$V_{SS}$	$V_{DD}$	V	—
Resolution (1 LSB)	Condition A Condition B RES	10.5 8.99	14.1 10.5	mV	—
Absolute accuracy (Total unadjusted error)	Condition A Condition B $E_{TUE}$	— —	$\pm 1.5$ $\pm 2.0$	LSB	Includes quantization
Conversion range	$V_{AIN}$	$V_{SS}$	$V_{DD}$	V	—
Power-up time	$t_{ADPU}$	16	—	$t_{ADIC}$ cycles	$t_{ADIC} = 1/f_{ADIC}$
Conversion time	$t_{ADC}$	16	17	$t_{ADIC}$ cycles	$t_{ADIC} = 1/f_{ADIC}$
Sample time <sup>(1)</sup>	$t_{ADS}$	5	—	$t_{ADIC}$ cycles	$t_{ADIC} = 1/f_{ADIC}$
Zero input reading <sup>(2)</sup>	$Z_{ADI}$	00	01	Hex	$V_{IN} = V_{SS}$
Full-scale reading <sup>(3)</sup>	$F_{ADI}$	FE	FF	Hex	$V_{IN} = V_{DD}$
Input capacitance	$C_{ADI}$	—	8	pF	Not tested
Input leakage <sup>(3)</sup>	$I_{IL}$	—	$\pm 1$	$\mu A$	—
ADC supply current ( $V_{DD} = 3 V$ )	$I_{ADAD}$	Typical = 0.45		mA	Enabled

1. Source impedances greater than 10 k $\Omega$  adversely affect internal RC charging time during input sampling.
2. Zero-input/full-scale reading requires sufficient decoupling measures for accurate conversions.
3. The external system error caused by input leakage current is approximately equal to the product of R source and input current.

17.11 Timer Interface Module Characteristics

Characteristic	Symbol	Min	Max	Unit
Timer input capture pulse width	$t_{TH}, t_{TL}$	2	—	$t_{cyc}$
Timer input capture period	$t_{TLTL}$	Note <sup>(1)</sup>	—	$t_{cyc}$
Timer input clock pulse width	$t_{TCL}, t_{TCH}$	$t_{cyc} + 5$	—	ns

1. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1  $t_{cyc}$ .

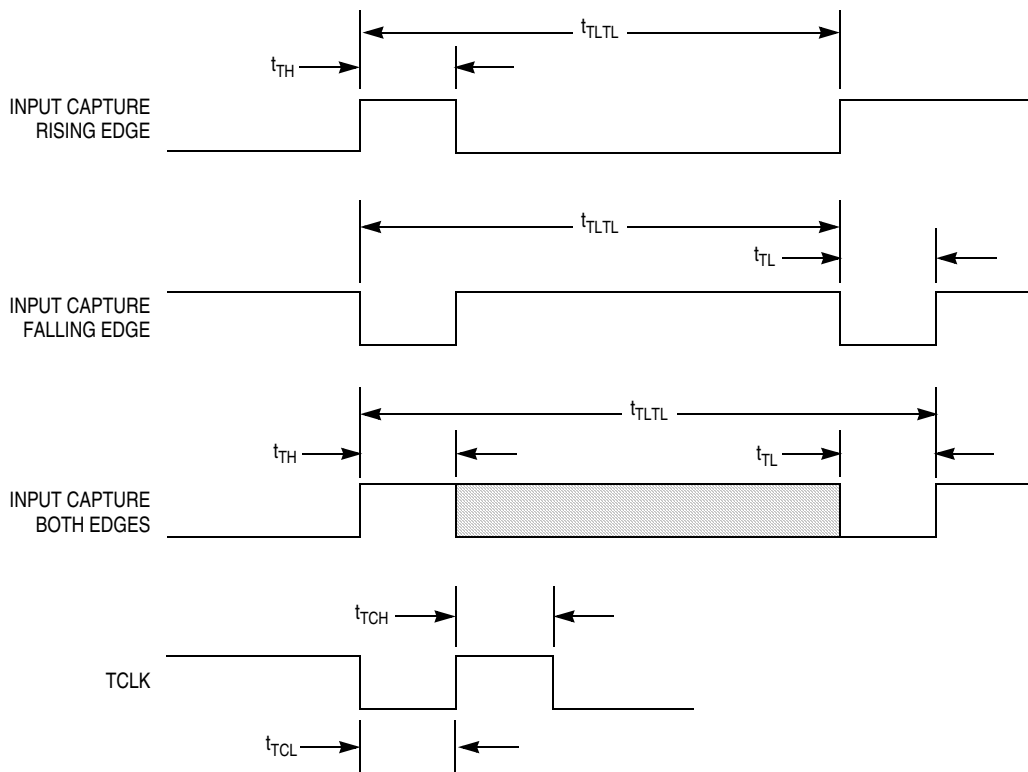


Figure 17-8. Timer Input Timing



17.12 Memory Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
RAM data retention voltage	$V_{RDR}$	1.3	—	—	V
FLASH program bus clock frequency	—	1	—	—	MHz
FLASH PGM/ERASE supply voltage ( $V_{DD}$ )	$V_{PGM/ERASE}$	2.7	—	3.6	V
FLASH read bus clock frequency	$f_{Read}^{(1)}$	0	—	2	MHz
FLASH page erase time <1 k cycles >1 k cycles	$t_{Erase}$	0.9 3.6	1 4	1.1 5.5	ms
FLASH mass erase time	$t_{MErase}$	4	—	—	ms
FLASH PGM/ERASE to HVEN setup time	$t_{NVS}$	10	—	—	$\mu$ s
FLASH high-voltage hold time	$t_{NVH}$	5	—	—	$\mu$ s
FLASH high-voltage hold time (mass erase)	$t_{NVHL}$	100	—	—	$\mu$ s
FLASH program setup time	$t_{PGS}$	5	—	—	$\mu$ s
FLASH program time	$t_{PROG}$	30	—	40	$\mu$ s
FLASH return to read time	$t_{RCV}^{(2)}$	1	—	—	ms
FLASH cumulative program hv period	$t_{HV}^{(3)}$	—	—	4	ms
FLASH endurance <sup>(4)</sup>	—	10 k	100 k	—	Cycles
FLASH data retention time <sup>(5)</sup>	—	15	100	—	Years

- $f_{Read}$  is defined as the frequency range for which the FLASH memory can be read.
- $t_{RCV}$  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 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:  $t_{NVS} + t_{NVH} + t_{PGS} + (t_{PROG} \times 32) \leq t_{HV}$  maximum.
- Typical endurance was evaluated for this product family. For additional information on how Motorola defines *Typical Endurance*, please refer to Engineering Bulletin EB619.
- Typical data retention values are based on intrinsic capability of the technology measured at high temperature and de-rated to 25°C using the Arrhenius equation. For additional information on how Motorola defines *Typical Data Retention*, please refer to Engineering Bulletin EB618.

### 17.13 UHF Transmitter Module

This subsection provides electrical specifications and timing definitions for the UHF transmitter module.

#### 17.13.1 UHF Module Electrical Characteristics

Unless otherwise specified:

- $V_{CC} = 3\text{ V}$
- $R_{EXT} = 12\text{ k}\Omega$
- Operating temperature range ( $T_A$ ) =  $-40^\circ\text{C}$  to  $85^\circ\text{C}$
- RF output frequency:  $f_{Carrier} = 433.92\text{ MHz}$
- Reference frequency:  $f_{Reference} = 13.56\text{ MHz}$
- OOK modulation selected
- Output load is  $50\ \Omega$  resistor (see [Figure 17-12](#))

Values refer to the circuit shown in the recommended application schematic (see [Figure 12-5. Application Schematic in OOK Modulation, 315-MHz and 434-MHz Frequency Bands](#)). Typical values reflect average measurement at  $V_{CC} = 3\text{ V}$ ,  $T_A = 25^\circ\text{C}$ .

Parameter	Test Conditions and Comments	Min	Typ	Max	Unit
<b>General Parameters</b>					
Supply current in standby mode	$T_A \leq 25^\circ\text{C}$	—	0.1	5	nA
	$T_A = 60^\circ\text{C}$	—	7	30	nA
	$T_A = 85^\circ\text{C}$	—	40	100	nA
Supply current in transmission mode	315 and 434 MHz bands, continuous wave, $T_A \leq 85^\circ\text{C}$	—	11.6	13.5	mA
	315 and 434 MHz bands, DATA = 0, $-40^\circ\text{C} \leq T_A \leq 85^\circ\text{C}$	—	4.4	6.0	mA
	868 MHz band, DATA = 0, $-40^\circ\text{C} \leq T_A \leq 85^\circ\text{C}$	—	4.6	6.2	mA
	868 MHz band, continuous wave, $-40^\circ\text{C} \leq T_A \leq 85^\circ\text{C}$	—	11.8	15.1	mA

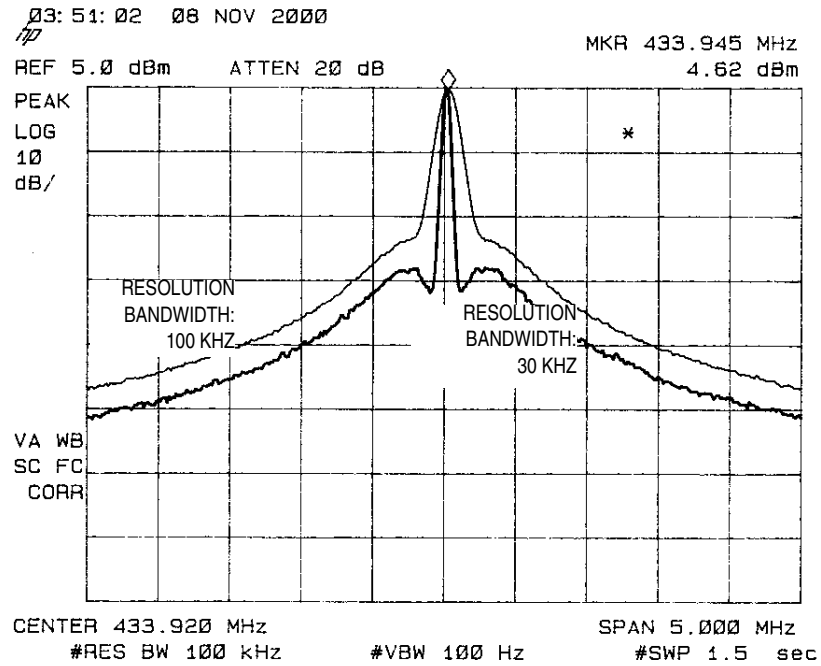
— Continued on next page

Parameter	Test Conditions and Comments	Min	Typ	Max	Unit
Supply voltage		—	3	3.6	V
Shutdown voltage threshold	$T_A = -40^\circ\text{C}$	—	2.04	2.11	V
	$T_A = -20^\circ\text{C}$	—	1.99	2.06	V
	$T_A = 25^\circ\text{C}$	—	1.86	1.95	V
	$T_A = 60^\circ\text{C}$	—	1.76	1.84	V
	$T_A = 85^\circ\text{C}$	—	1.68	1.78	V
	$T_A = 125^\circ\text{C}$	—	1.56	1.67	V
<b>RF Parameters (assuming a 50 <math>\Omega</math> matching network connected to the D.U.T. output)</b>					
$R_{EXT}$ value		12	—	21	k $\Omega$
Output power	315 and 434 MHz bands, with 50 $\Omega$ matching network	—	5	—	dBm
	868 MHz band, with 50 $\Omega$ matching network	—	1	—	dBm
	315 and 434 MHz bands, $-40^\circ\text{C} \leq T_A \leq 125^\circ\text{C}$	-3	0	3	dBm
	868 MHz band, $-40^\circ\text{C} \leq T_A \leq 125^\circ\text{C}$	-7	-3	0	dBm
Current and output power variation vs $R_{EXT}$ value	314 and 434 MHz bands, with 50 $\Omega$ matching network	—	-0.35	—	dB/k $\Omega$ mA/k $\Omega$
Harmonic 2 level	315 and 434 MHz bands, with 50 $\Omega$ matching network	—	-34	—	dBc
	868 MHz band, with 50 $\Omega$ matching network	—	-49	—	dBc
	315 and 434 MHz bands	—	-23	-17	dBc
	868 MHz band	—	-38	-27	dBc
Harmonic 3 level	315 and 434MHz bands, with 50 $\Omega$ matching network	—	-32	—	dBc
	868 MHz band, with 50 $\Omega$ matching network	—	-57	—	dBc
	315 and 434 MHz bands	—	-21	-15	dBc
	868 MHz band	—	-48	-39	dBc
Spurious level @ $f_{Carrier} \pm f_{DATACLK}$	315 and 434 MHz bands	—	-36	-24	dBc
	868 MHz band	—	-29	-17	dBc
Spurious level @ $f_{Carrier} \pm f_{Reference}$	315 MHz band	—	-37	-30	dBc
	434 MHz band	—	-44	-34	dBc
	868 MHz band	—	-37	-27	dBc

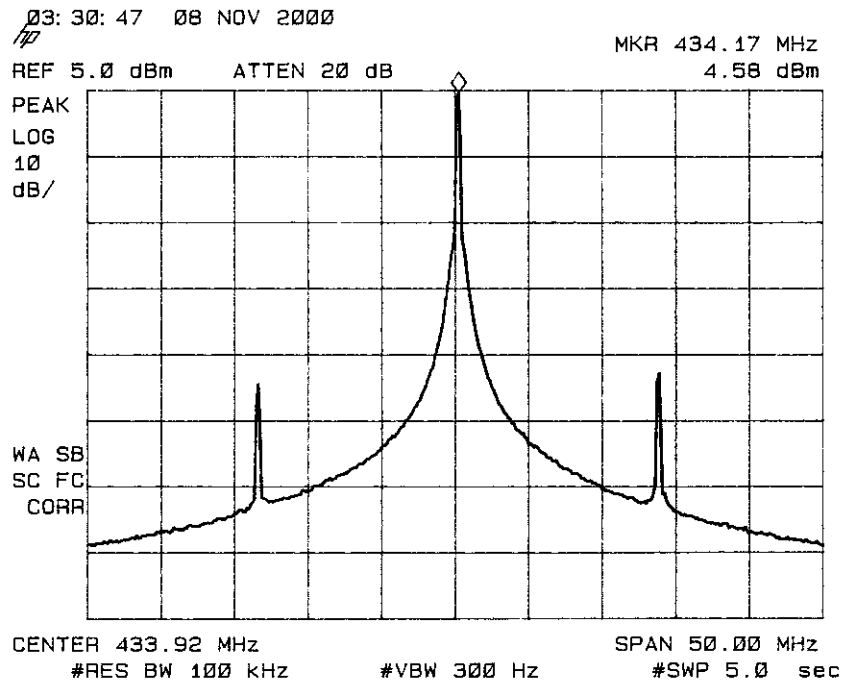
— Continued on next page

Electrical Specifications

Parameter	Test Conditions and Comments	Min	Typ	Max	Unit
Spurious level @ $f_{Carrier}/2$	315 MHz bands	—	-62	-53	dBc
	434 MHz bands	—	-80	-60	dBc
	868 MHz band	—	-45	-39	dBc
RF spectrum	434 MHz bands	See <a href="#">Figure 17-9</a> , <a href="#">Figure 17-10</a> , and <a href="#">Figure 17-11</a>			—
Phase noise	315 and 434 MHz bands, $\pm 175$ kHz from $f_{Carrier}$	—	-75	-68	dBc/Hz
	868 MHz band, $\pm 175$ kHz from $f_{Carrier}$	—	-73	-66	dBc/Hz
PLL lock-in time, $t_{PLL\_Lock\_In}$	$f_{Carrier}$ within 30 kHz from the final value, crystal series resistor = 150 $\Omega$	—	400	1600	$\mu$ s
XTAL1 input capacitance		—	1	2	pF
Crystal resistance	OOK modulation	—	20	200	$\Omega$
	FSK modulation	—	20	50	
OOK modulation depth		75	90	—	dBc
Data rate		—	—	10	kBit/s
<b>Microcontroller Interfaces</b>					
Input low voltage	Pins BAND, MODE, ENABLE, and DATA	0	—	$0.3 \times V_{CC}$	V
Input high voltage		$0.7 \times V_{CC}$	—	$V_{CC}$	V
Input hysteresis voltage		—	—	150	mV
Input current	Pins BAND, MODE, DATA @ high level	—	—	100	nA
ENABLE pulldown resistor		—	180	—	k $\Omega$
DATACLK output low voltage	$C_{Load} = 2$ pF	0	—	$0.25 \times V_{CC}$	V
DATACLK output high voltage		$0.75 \times V_{CC}$	—	$V_{CC}$	V
DATACLK rising time	$C_{Load} = 2$ pF, measured from 20% to 80% of the voltage swing	—	250	500	ns
DATACLK falling time		—	150	400	ns
DATACLK settling time, $t_{DATACLK\_Settling}$	45 < duty cycle $f_{DATACLK}$ < 55%	—	800	1800	$\mu$ s



**Figure 17-9. RF Spectrum at 434-MHz Frequency Band Displayed with a 5-MHz Span**



**Figure 17-10. RF Spectrum at 434-MHz Frequency Band Displayed with a 50-MHz Span**

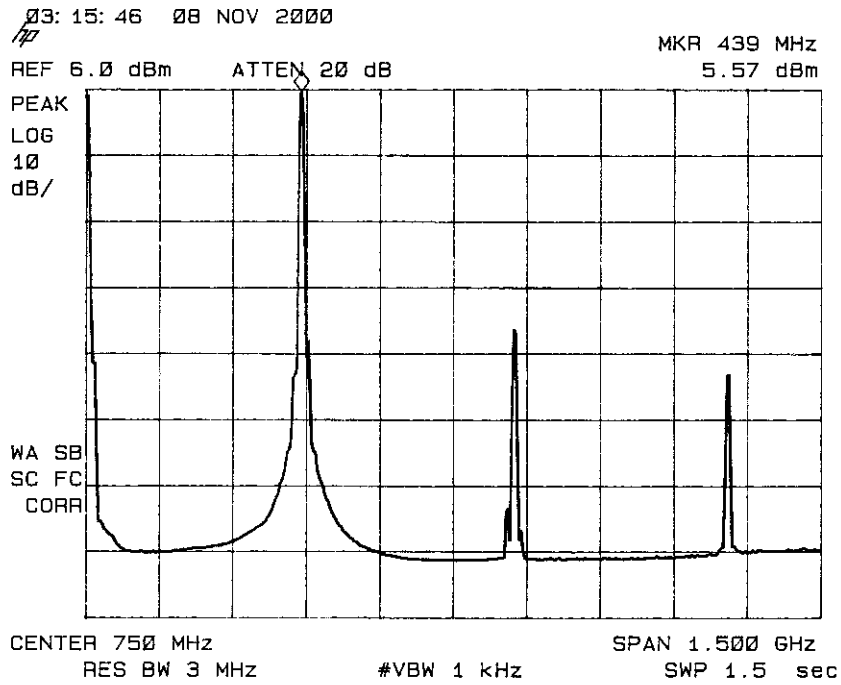


Figure 17-11. RF Spectrum at 434-MHz Frequency Band Displayed with a 1.5-GHz Span

17.13.2 UHF Module Output Power Measurement

The RF output levels given in the [17.13.1 UHF Module Electrical Characteristics](#) are measured with a 50-Ω load directly connected to the pin RFOUT as shown in figure [Figure 17-12](#). This wideband coupling method gives results independent of the application.

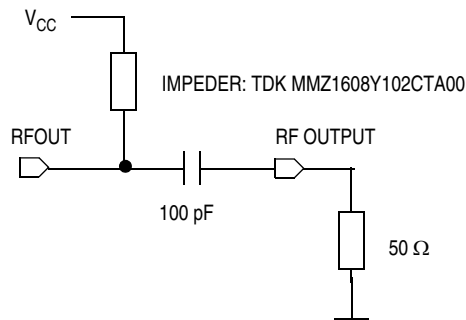
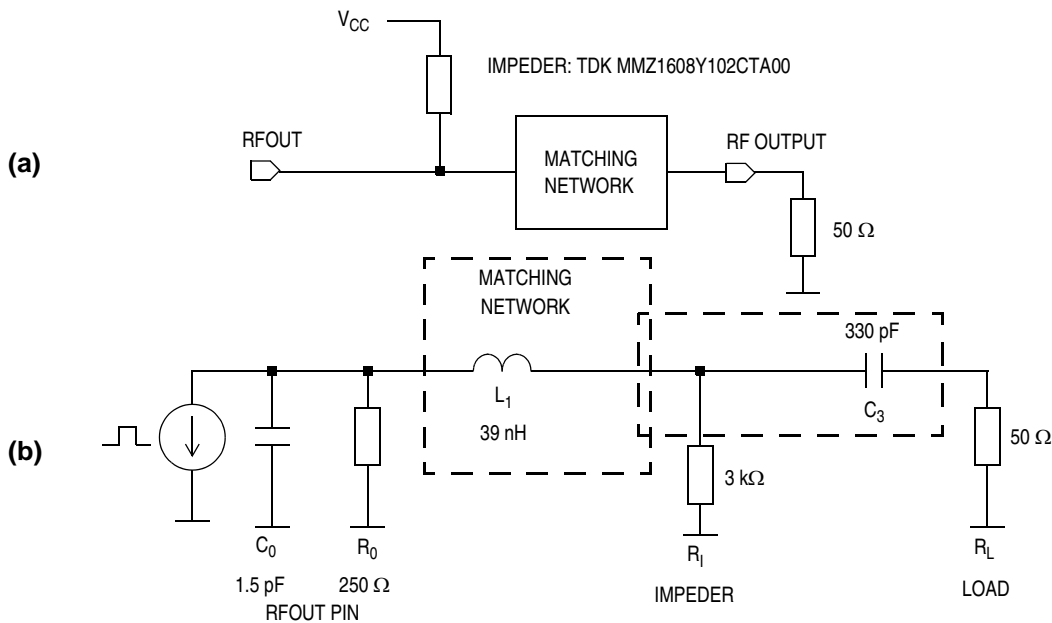


Figure 17-12. Output Power Measurement Configurations

The configuration shown in **Figure 17-13(a)** provides a better efficiency in terms of output power and harmonics rejection. Schematic in **Figure 17-13(b)** gives the equivalent circuit of the pin RFOUT and impededer as well as the matching network components for 434-MHz frequency band.

**NOTE:** Note that the impededer is moved to the load side to decrease its influence (similar to dc bias through the antenna).

**Figure 17-14** gives the output power versus the  $R_{EXT}$  resistor value, in both cases with 50- $\Omega$  load and with matching network.



**Figure 17-13. Output Characteristic and Matching Network for 434-MHz Frequency Band**

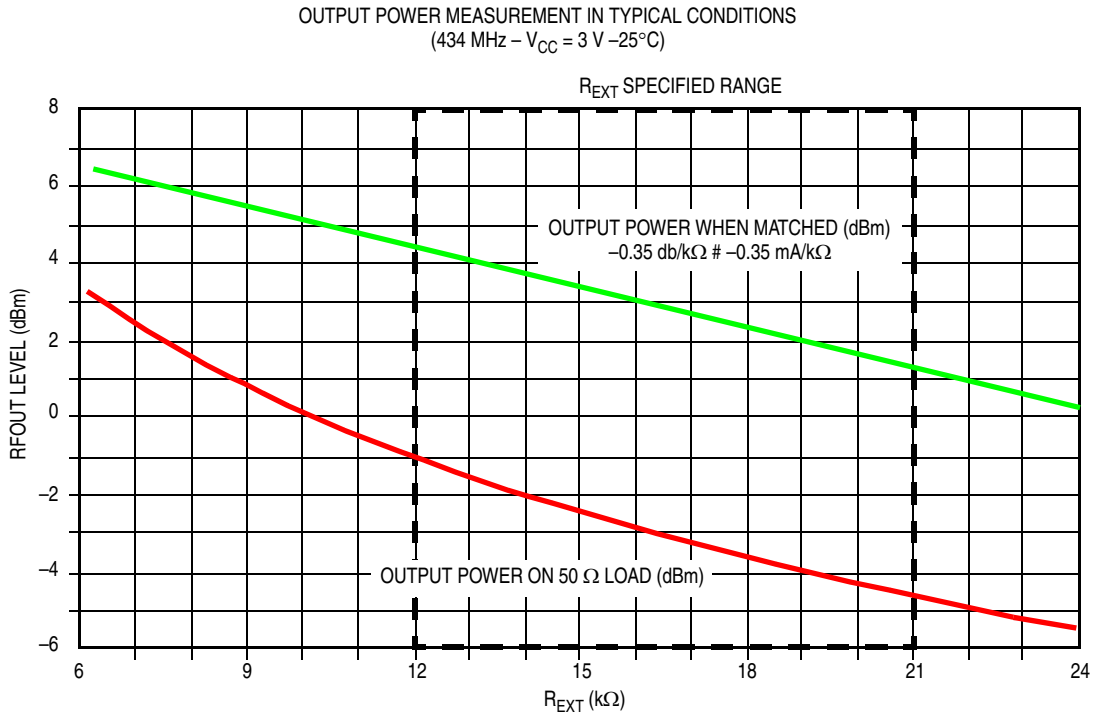


Figure 17-14. Output Power at 434-MHz Frequency Band versus  $R_{EXT}$  Value



## Section 18. Ordering Information and Mechanical Specifications

### 18.1 Introduction

This section provides ordering information and mechanical specifications for the 32-pin low-profile quad flat pack (LQFP).

The package outline given here reflects the latest package drawing at the time of publication. To make sure that you have the latest package specification, contact your local Motorola Sales Office.

### 18.2 MC Order Numbers

Table 18-1. Available MC Order Numbers

MC Order Number	Operating Temperature Range	Package
MC908QF4CFJ	-40°C to +85°C	32-pin LQFP
MC908QF4FJ	0°C to +70°C	32-pin LQFP

Temperature and package designators:  
 C = -40°C to +85°C  
 FJ = Low-profile quad flat pack (LQFP)

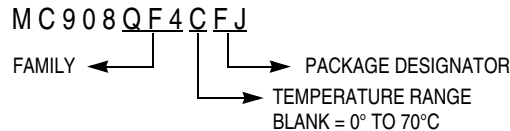
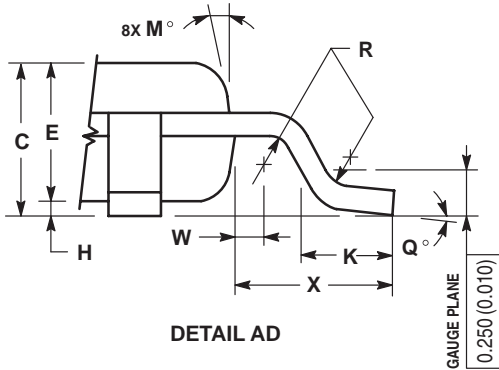
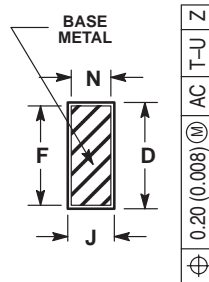
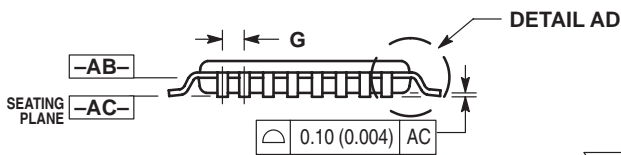
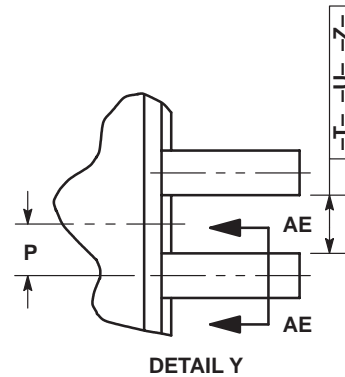
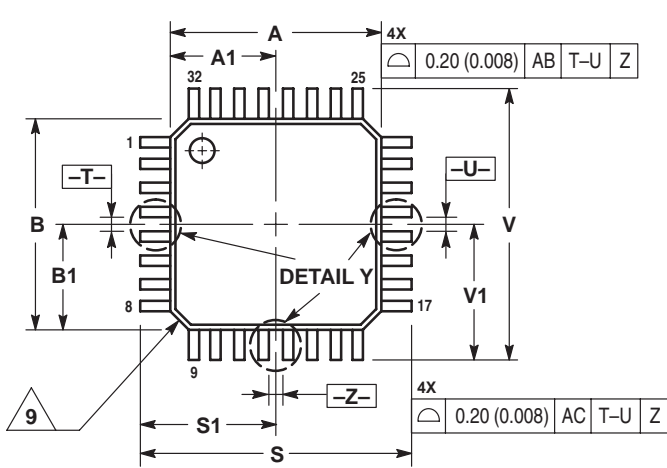


Figure 18-1. Device Numbering System

18.3 32-Pin Plastic Low-Profile Quad Flat Pack (Case No. 873A)



NOTES:

- DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
- CONTROLLING DIMENSION: MILLIMETER.
- DATUM PLANE -AB- IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
- DATUMS -T-, -U-, AND -Z- TO BE DETERMINED AT DATUM PLANE -AB-.
- DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -AC-.
- DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.250 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -AB-.
- DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE D DIMENSION TO EXCEED 0.520 (0.020).
- MINIMUM SOLDER PLATE THICKNESS SHALL BE 0.0076 (0.0003).
- EXACT SHAPE OF EACH CORNER MAY VARY FROM DEPICTION.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	7.000 BSC		0.276 BSC	
A1	3.500 BSC		0.138 BSC	
B	7.000 BSC		0.276 BSC	
B1	3.500 BSC		0.138 BSC	
C	1.400	1.600	0.055	0.063
D	0.300	0.450	0.012	0.018
E	1.350	1.450	0.053	0.057
F	0.300	0.400	0.012	0.016
G	0.800 BSC		0.031 BSC	
H	0.050	0.150	0.002	0.006
J	0.090	0.200	0.004	0.008
K	0.500	0.700	0.020	0.028
M	12° REF		12° REF	
N	0.090	0.160	0.004	0.006
P	0.400 BSC		0.016 BSC	
Q	1°	5°	1°	5°
R	0.150	0.250	0.006	0.010
S	9.000 BSC		0.354 BSC	
S1	4.500 BSC		0.177 BSC	
V	9.000 BSC		0.354 BSC	
V1	4.500 BSC		0.177 BSC	
W	0.200 REF		0.008 REF	
X	1.000 REF		0.039 REF	



## HOW TO REACH US:

### USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution  
P.O. Box 5405  
Denver, Colorado 80217  
1-800-521-6274 or 480-768-2130

### JAPAN:

Motorola Japan Ltd.  
SPS, Technical Information Center  
3-20-1, Minami-Azabu, Minato-ku  
Tokyo 106-8573, Japan  
81-3-3440-3569

### ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.  
Silicon Harbour Centre  
2 Dai King Street  
Tai Po Industrial Estate  
Tai Po, N.T., Hong Kong  
852-26668334

### HOME PAGE:

<http://motorola.com/semiconductors>



Information in this document is provided solely to enable system and software implementers to use Motorola 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.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola 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 Motorola 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. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.

MOTOROLA and the Stylized M Logo are registered in the US Patent and Trademark Office. All other product or service names are the property of their respective owners. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

© Motorola Inc. 2004

MC68HC908QF4  
Rev. 1.0  
6/2004

**For More Information On This Product,  
Go to: [www.freescale.com](http://www.freescale.com)**