Order this document by
MC68HC908RC24/D
Rev. 1

HC08

68HC908RC24

Advance Information

This document contains information on a new product.
Specifications and information herein are subject to change without notice.

MOTOROLA

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it 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.

List of Sections

List of Sections

Table of Contents

Section 1. General Description

Section 2. Memory Map

Section 3. Random-Access Memory (RAM)

Section 4. FLASH Memory

Section 5. Central Processor Unit (CPU)

Section 6. Resets and Interrupts

Section 7. Low-Power Modes

Section 8. Low-Voltage Inhibit (LVI)

Section 9. Oscillator

Section 10. Configuration Register (CONFIG)

Section 11. Input/Output Ports (I/O)

Section 12. Carrier Modulator Transmitter (CMT)

Section 13. Modulo Timer (TIM0I)

Section 14. External Interrupt (IRQ)

Section 15. Keyboard Interrupt Module (KBI)

Section 16. Computer Operating Properly (COP)

Section 17. Monitor ROM (MON)

Section 18. Break Module (BRK)

Section 19. Preliminary Electrical Specifications

Section 20. Mechanical Specifications

Section 21. Ordering Information

Table of Contents

List of Figures

List of Figures

Figure	Title	Page

List of Figures

List of Tables

List of Tables

Section 1. General Description

1.1 Contents

1.2 Features

Features of the MC68HC908RC24 include:

- High performance M68HC08 architecture
- Fully upward-compatible object code with M6805, M146805, and M68HC05 Families
- 8-MHz maximum oscillator frequency at 2.0-volt supply
- 24,064 byes of on-chip FLASH memory
- On-chip programming firmware for use with host personal computer
- FLASH memory data security(1)
- 352 bytes of on-chip random-access memory (RAM)
- Low-voltage inhibit (LVI) module:
  - 1.8-V detection forces the microprocessor unit (MCU) into low-power state
  - 2.0-V detection sets indicator flag
- Low-power design, fully static with low-power reset, stop, and wait modes
- Battery removal detection circuits
- Computer operating properly (COP) watchdog resets
- Carrier modulator transmitter (CMT) supporting baseband, pulse length modulator (PLM), and frequency shift keying (FSK) protocols
- 16-bit, modulo timer module (TIM0I)

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

- Available packaging:
  - 28-pin plastic dual in-line package (PDIP)
  - 28-pin small-outline integrated circuit package (SOIC)
- 20 bidirectional input/output (I/O) lines
- 8-bit keyboard wakeup port
- High-current infrared (IR) drive pin (IRO)
- High-current port pins (PC0–PC3)

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 × 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 MC68HC908RC24.

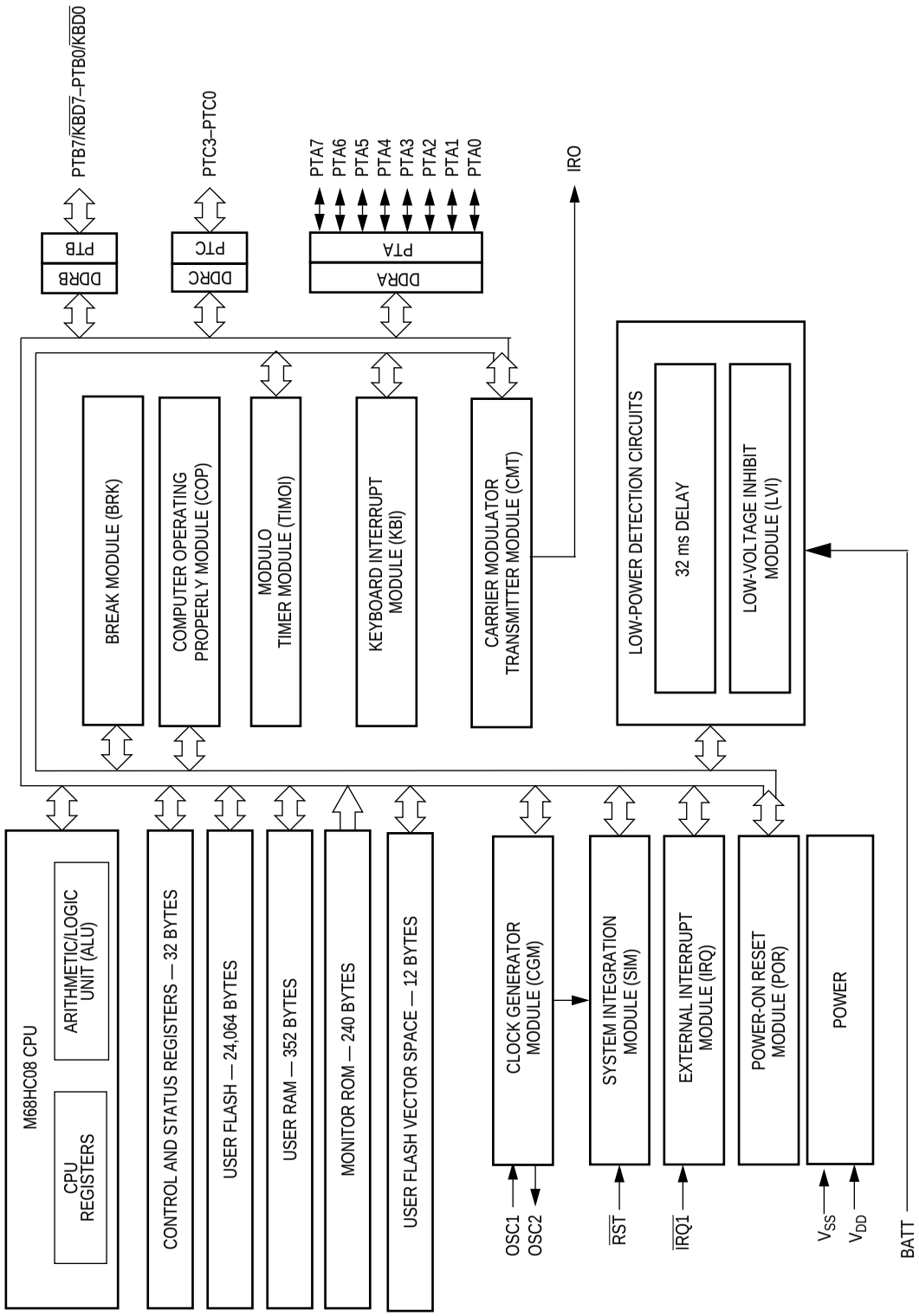


Figure 1-1. MCU Block Diagram for the MC68HC908RC24

1.4 Pin Assignments

Figure 1-2 shows the pin assignments for both the DIP and SOIC packages.

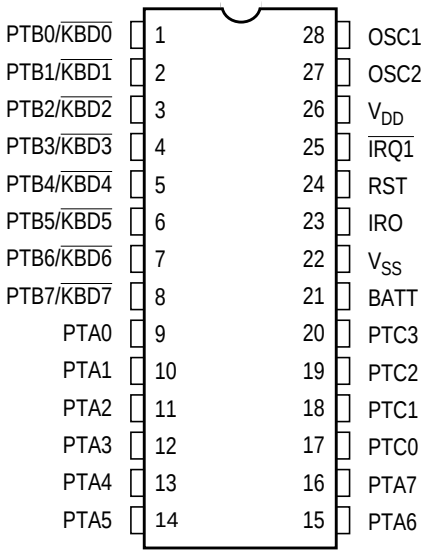


Figure 1-2. DIP and SOIC Pin Assignments

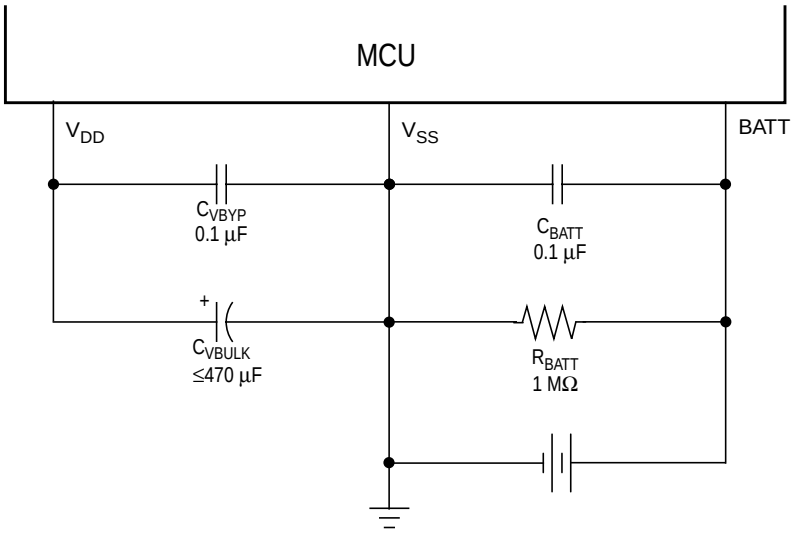
1.5 Pin Functions

This section provides a brief description of the pin functions. Where applicable, references are made to other sections for more detailed information.

1.5.1 V_{DD} and V_{SS}

V_{DD} is the supply node for the MCU. Under normal operation, power is supplied to V_{DD} from the BATT pin through an internal MOSFET device. When the battery is removed, power is temporarily supplied from an external electrolytic capacitor through V_{DD}. V_{SS} is ground.

Fast signal transitions on MCU pins place high, short-duration current demands on the power supply. To prevent noise problems, take special care to provide power supply bypassing at the MCU as **Figure 1-3** shows. Place the C_{VBYP} and C_{BATT} bypass capacitors as close to the MCU as possible. Use a high-frequency-response ceramic capacitor for C_{VBYP} and C_{BATT}. C_{VBULK} is used for bulk current requirements in applications that require the port pins to source high current levels during normal operation. C_{VBULK} is also used for RAM retention when batteries are removed.



Note: Component values shown represent typical applications.

Figure 1-3. Power Supply Bypassing

1.5.2 BATT

The voltage on BATT is passed to V_{DD} through an internal P-channel MOSFET transistor and supplies power to the MCU. This pin also detects battery removal and insertion. See **Section 7. Low-Power Modes**.

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

$\overline{\text{IRQ1}}$ is an asynchronous external interrupt pin. See **Section 14. External Interrupt (IRQ)**.

NOTE: *The IRQ circuits are powered by the chip V_{DD}. External circuits should drive the pin between V_{DD} and V_{SS} levels. If external circuits use a pullup device on $\overline{IRQ1}$, the pin must be pulled to V_{DD} and **not** to BATT.*

1.5.4 OSC1 and OSC2

The OSC1 and OSC2 pins are the connections for the on-chip oscillator circuit. See **Section 8. Low-Voltage Inhibit (LVI)**.

1.5.5 $\overline{\text{RST}}$

A logic 0 on the $\overline{\text{RST}}$ pin forces the MCU to a known startup state. $\overline{\text{RST}}$ is bidirectional, allowing a reset of the entire system. It is driven low when any internal reset source is asserted. See **Section 6. Resets and Interrupts**.

NOTE: *The reset circuits are powered by the chip V_{DD}. External circuits should drive the pin through a resistor between V_{DD} and V_{SS} levels. If external circuits use a pullup device on $\overline{RST}$, the pin must be pulled to V_{DD} and **not** to BATT.*

1.5.6 IRO

The IRO pin is the high-current source and sink output of the carrier modulator transmitter (CMT) subsystem which is suitable for driving infrared (IR) light-emitting diode (LED) biasing logic. See **Section 12. Carrier Modulator Transmitter (CMT)**.

1.5.7 Port A Input/Output Pins (PTA7–PTA0)

PTA7–PTA0 are general-purpose bidirectional I/O port pins. See **Section 11. Input/Output Ports (I/O)**.

1.5.8 Port B Input/Output Pins (PTB7/$\overline{\text{KBD7}}$–PTB0/$\overline{\text{KBD0}}$)

PTB7/$\overline{\text{KBD7}}$–PTB0/$\overline{\text{KBD0}}$ are general-purpose bidirectional I/O port pins. Any or all of the port B lines can be programmed to serve as external interrupt pins with internal pullups. See **Section 11. Input/Output Ports (I/O)**.

1.5.9 Port C Input/Output Pins (PTC7–PTC0)

PTC7–PTC0 are general-purpose bidirectional I/O port pins. PTC3–PTC0 have high-current drive capability. The state of any pin is software programmable and all port C lines are configured as input during power-on or reset. See **Section 11. Input/Output Ports (I/O)**.

NOTE: *On 28-pin package parts, PTC7–PTC4 pads are not bonded out. Set these ports to outputs after any reset to avoid floating inputs.*

Any unused inputs and I/O ports should be tied to an appropriate logic level (either V_{DD} or V_{SS}). Although the I/O ports of the MC68HC908RC24 do not require termination, termination is recommended to reduce the possibility of static damage.

Section 2. Memory Map

2.1 Contents

2.2 Introduction

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

- 24,064 bytes of FLASH memory
- 352 bytes of random-access memory (RAM)
- 12 bytes of user-defined vectors
- 240 bytes of monitor ROM (MON)

$0000 ↓ $001F	I/O REGISTERS 32 BYTES
$0020 ↓ $017F	RAM 352 BYTES
$0180 ↓ $9FFF	UNIMPLEMENTED 40,576 BYTES
$A000 ↓ $FDFF	FLASH MEMORY 24,064 BYTES
$FE00	BREAK STATUS REGISTER (BSR)
$FE01	RESET STATUS REGISTER (RSR)
$FE02	RESERVED
$FE03	BREAK FLAG CONTROL REGISTER (BFCR)
$FE04	INTERRUPT STATUS REGISTER 1 (INT1)
$FE05 ↓ $FE08	RESERVED 4 BYTES
$FE09	FLASH CONTROL REGISTER (FLCR)
$FE0A ↓ $FE0B	RESERVED 2 BYTES
$FE0C	BREAK ADDRESS HIGH REGISTER (BRKH)
$FE0D	BREAK ADDRESS LOW REGISTER (BRKL)
$FE0E	BREAK STATUS AND CONTROL REGISTER (BSCR)
$FE0F	RESERVED
$FE10 ↓ $FEFF	MONITOR ROM 240 BYTES
$FF00 ↓ $FF7F	UNIMPLEMENTED 128 BYTES
$FF80	FLASH BLOCK PROTECT REGISTER (FLBPR)
$FF81 ↓ $FFF3	UNIMPLEMENTED 115 BYTES
$FFF4 ↓ $FFFE	FLASH VECTORS 12 BYTES
$FFFF	COP CONTROL REGISTER (COPCTL)

Figure 2-1. Memory Map

2.3 I/O Section

Most of the control, status, and data registers are in the zero page area of $0000–$001F. Additional input/output (I/O) registers have these addresses:

- $FE00 — Break status register, BSR
- $FE01 — Reset status register, RSR
- $FE03 — Break flag control register, BFCR
- $FE04 — Interrupt status register 1, INT1
- $FE09 — FLASH control register, FLCR
- $FE0C and $FE0D — Break address registers, BRKH and BRKL
- $FE0E — Break status and control register, BSCR
- $FF80 — FLASH block protection register, FLBPR, in non-volatile FLASH memory
- $FFFF — Computer operating properly (COP) control register, COPCTL

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA) See page 111.	Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PTB) See page 113.	Read: Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PTC) See page 116.	Read:	0	0	0	0	PTC3	PTC2	PTC1	PTC0
		Write:								
		Reset:	Unaffected by reset							
$0003	Unimplemented									
$0004	Data Direction Register A (DDRA) See page 111.	Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB) See page 114.	Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Reset:	0	0	0	0	0	0	0	0
$0006	Data Direction Register C (DDRC) See page 117.	Read:	1	1	1	1	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007	Unimplemented									
$0008	Unimplemented									
$0009	Unimplemented									

Note: PTC7–PTC4 are not available. See **11.5 Port C**..

Note: DDRC7–DDRC4 are not available. Set DDRC7–DDRC4 to 1 on 28-pin packages as described in **11.5 Port C**.

= Unimplemented R = Reserved X = Indeterminate

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
$000A	Unimplemented									
$000B	Unimplemented									
$000C	Unimplemented									
$000D	Keyboard Status and Control Register (KBSCR) See page 165.	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$000E	Keyboard Interrupt Enable Register (KBIER) See page 166.	Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000F	IRQ Status and Control Register (ISCR) See page 157.	Read:	0	0	0	0	IRQF1	0	IMASK1	MODE1
		Write:						ACK1		
		Reset:	0	0	0	0	0	0	0	0
$0010	CMT Carrier Generator High Data Register 1 (CCH1) See page 135.	Read:	IROLN	CMTPOL	PH5	PH4	PH3	PH2	PH1	PH0
		Write:								
		Reset:	0	0	Unaffected by reset					
$0011	CMT Carrier Generator Low Data Register 1 (CCL1) See page 135.	Read:	IROLP	0	PL5	PL4	PL3	PL2	PL1	PL0
		Write:								
		Reset:	0	0	Unaffected by reset					
$0012	CMT Carrier Generator High Data Register 2 (CCH2) See page 135.	Read:	0	0	SH5	SH4	SH3	SH2	SH1	SH0
		Write:								
		Reset:	0	0	Unaffected by reset					
$0013	CMT Carrier Generator Low Data Register 2 (CCL2) See page 135.	Read:	0	0	SL5	SL4	SL3	SL2	SL1	SL0
		Write:								
		Reset:	0	0	Unaffected by reset					

= Unimplemented R = Reserved X = Indeterminate

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

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0014	CMT Modulator Control and Status Register (CMCS) See page 138.	Read:	EOCF	DIV2	0	EXSPC	BASE	MODE	EOCIE	MCGEN
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0015	CMT Modulator Data Register 1 (CMD1) See page 141.	Read:	MB11	MB10	MB9	MB8	SB11	SB10	SB9	SB8
		Write:								
		Reset:	Unaffected by reset							
$0016	CMT Modulator Data Register 2 (CMD2) See page 141.	Read:	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
		Write:								
		Reset:	Unaffected by reset							
$0017	CMT Modulator Data Register 3 (CMD3) See page 141.	Read:	SB7	SB6	SB5	SB4	SB3	SB2	SB1	SB0
		Write:								
		Reset:	Unaffected by reset							
$0018	TIM0I Status and Control Register (TSC) See page 148.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0019	TIM0I Counter Register High (TCNTH) See page 150.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001A	TIM0I Counter Register Low (TCNTL) See page 150.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001B	TIM0I Counter Modulo Register High (TMODH) See page 151.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$001C	TIM0I Counter Modulo Register Low (TMODL) See page 151.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1

= Unimplemented R = Reserved X = Indeterminate

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
$001D	LVI Status Register (LVISR) See page 98.	Read:	R	R	LOWV	0	0	0	0	R
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001E	Unimplemented									
$001F	Configuration Register (CONFIG) See page 106.	Read:	0	0	0	0	SSREC	COPRS	STOP	COPD
		Write:								
		Reset:	0	0	1	0	0	0	0	0
$FE00	Break Status Register (BSR) See page 191.	Read:	0	0	0	1	0	0	BW	0
		Write:	R	R	R	R	R	R	Note	R
		Reset:	0	0	0	1	0	0	0	0

Note: Writing a logic 0 clears BW.

$FE01	Reset Status Register (RSR) See page 73.	Read:	POR	PIN	COP	ILOP	ILAD	0	LPRST	0
		Write:								
		POR:	1	X	0	0	0	0	1	0
$FE02	Unimplemented									
$FE03	Break Flag Control Register (BFCR) See page 192.	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE04	Interrupt Status Register 1 (INT1) See page 81.	Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE05	Interrupt Status Register 2 (INT2) See page 82.	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE06	Reserved		R	R	R	R	R	R	R	R

= Unimplemented R = Reserved X = Indeterminate

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

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE07	Unimplemented									
$FE08	Reserved		R	R	R	R	R	R	R	R
$FE09	FLASH Control Register (FLCR) See page 44.	Read:	FDIV1	FDIV0	BLK1	BLK0	HVEN	VERF	ERASE	PGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0A	Unimplemented									
$FE0B	Unimplemented									
$FE0C	Break Address Register High (BRKH) See page 190.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0D	Break Address Register Low (BRKL) See page 190.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0E	Break Status and Control Register (BSCR) See page 189.	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FF80	FLASH Block Protect Register (FLBPR) See page 49.	Read:	0	0	0	0	BPR3	BPR2	BPR1	BPR0
		Write:								
			0	0	0	0	Unaffected by reset			
$FFFF	COP Control Register (COPCTL) See page 171.	Read:	Low byte of reset vector							
		Write:	Writing clears COP counter (any value)							
		Reset:	Unaffected by reset							

= Unimplemented R = Reserved X = Indeterminate

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

Table 2-1 is a list of vector locations.

Table 2-1. Vector Addresses

	Address	Vector
Low	$FFF4	Keyboard vector (high)
	$FFF5	Keyboard vector (low)
Priority ↑	$FFF6	TIM overflow vector (high)
	$FFF7	TIM overflow vector (low)
	$FFF8	CMT vector (high)
	$FFF9	CMT vector (low)
	$FFFA	IRQ1 vector (high)
	$FFFB	IRQ1 vector (low)
	$FFFC	SWI vector (high)
↓	$FFFD	SWI vector (low)
	$FFFE	Reset vector (high)
High	$FFFF	Reset vector (low)

2.4 Monitor ROM (MON)

The 240 bytes at addresses $FE10–$FEFF are reserved ROM addresses that contain the instructions for the monitor functions.

Memory Map

Section 3. Random-Access Memory (RAM)

3.1 Contents

3.2 Introduction

This section describes the 352 bytes of random-access memory (RAM).

3.3 Functional Description

Addresses $0020–$017F are RAM locations. The location of the stack RAM is programmable. The 16-bit stack pointer allows the stack to be anywhere in the 64-Kbyte memory space.

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

Within page zero are 224 bytes of RAM. Because the location of the stack RAM is programmable, all page zero RAM locations can be used for input/output (I/O) control and user data or code. When the stack pointer is moved from its reset location at $00FF out of page zero, direct addressing mode instructions can efficiently access all page zero RAM locations. Page zero RAM, therefore, provides ideal locations for frequently accessed global variables.

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

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

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

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

Section 4. FLASH Memory

4.1 Contents

4.2 Introduction

This section describes the operation of the embedded FLASH memory. This memory can be read, programmed, and erased. An internal supply is used for program and erase.

4.3 Functional Description

The FLASH memory is an array of 24,064 bytes with an additional 12 bytes of user vectors and one byte of block protection. An erased bit reads as a logic 0 and a programmed bit reads as a logic 1. Program and erase operations are facilitated through control bits in a memory mapped register. Details for these operations appear later in this section.

FLASH locations are in the ranges:

- $A000–$FDFF — User memory
- $FF80 — Block protect register, FLBPR
- $FFF4–$FFFF — Reserved for user-defined interrupt and reset vectors

For availability of programming tools and more information, contact a local Motorola representative.

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

4.4 FLASH Control Register

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

Address: $FE09

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	FDIV1	FDIV0	BLK1	BLK0	HVEN	VERF	ERASE	PGM
Reset:	0	0	0	0	0	0	0	0

Figure 4-1. FLASH Control Register (FLCR)

FDIV1 — Frequency Divide Control Bit

This read/write bit together with FDIV0 selects the factor by which the charge pump clock is divided from the system clock. See **4.5 Charge Pump Frequency Control**.

FDIV0 — Frequency Divide Control Bit

This read/write bit together with FDIV1 selects the factor by which the charge pump clock is divided from the system clock. See **4.5 Charge Pump Frequency Control**.

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

BLK1— Block Erase Control Bit

This read/write bit together with BLK0 allows erasing of blocks of varying size. See **4.6 FLASH Erase Operation** for a description of available block sizes.

BLK0 — Block Erase Control Bit

This read/write bit together with BLK1 allows erasing of blocks of varying size. See **4.6 FLASH Erase Operation** for a description of available block sizes.

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 be set only if either PGM or ERASE is high and the sequence for erase or program/verify is followed.

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

VERF — Verify Control Bit

This read/write bit configures the memory for verify operation. It cannot be set if the HVEN bit is high, and if it is high when HVEN is set, it will automatically return to 0.

1 = Verify operation selected
0 = Verify operation unselected

ERASE — Erase Control Bit

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

4.5 Charge Pump Frequency Control

The internal charge pump is designed to operate at greatest efficiency at a frequency of 2 MHz. **Table 4-1** shows how the FDIV bits are used to select a charge pump frequency and the recommended bus frequency ranges for each configuration. Program and erase operations cannot be performed if the pump clock frequency is below 2 MHz.

Table 4-1. Charge Pump Clock Frequency

FDIV1	FDIV0	Pump Clock Frequency	Use When Bus Frequency is
0	0	Bus frequency ÷ 1	2 MHz ± 10%
0	1	Bus frequency ÷ 2	4 MHz ± 10%
1	0	Bus frequency ÷ 2	4 MHz ± 10%
1	1	Bus frequency ÷ 4	8 MHz ± 10%

4.6 FLASH Erase Operation

Use this step-by-step procedure to erase a block of FLASH memory. Values for the time parameters are specified in **19.13 Memory Characteristics**.

1. Set the ERASE bit and the BLK0 and BLK1 bits in the FLASH control register. See **Table 4-2** for block sizes.
2. Read from the block protect register: address $FF80.
3. Write to any FLASH address with any data within the block address range desired.
4. Set the HVEN bit.
5. Wait for a time, t_{Erase}.
6. Clear the HVEN bit.
7. Wait for a time, t_{Kill}, for the high voltages to dissipate.
8. Clear the ERASE bit.
9. After a time, t_{HVD}, the memory can be accessed in read mode again.

NOTE: *While these operations must be performed in the order shown, other unrelated operations may occur between the steps.*

Table 4-2 shows the various block sizes which can be erased in one erase operation.

Table 4-2. Erase Block Sizes

BLK1	BLK0	Block Size, Addresses Cared
0	0	Full array: 24 Kbytes
0	1	Half array: 8 or 16 Kbytes (A14)
1	0	Eight rows: 512 bytes (A14–A9)
1	1	Single row: 64 bytes (A14–A6)

In step 3 of the erase operation, the cared addresses are latched and used to determine the location of the block to be erased. For the full array (BLK1 = BLK0 = 0), the only requirement is that the FLASH memory be selected. Writing to any address in the range $A000 to $FDFF or the vectors in the address range $FFF4 to $FFFF will enable the full array erase. In the half array case (BLK1 = 0, BLK0 = 1), the state of A14 determines whether the range $A000 to $BFFF is erased (A14 = 0) or the range from $C000 to $FFFF (A14 = 1) is erased.

4.7 FLASH Program/Verify Operation

Programming of the FLASH memory is done on a page basis. A page consists of eight consecutive bytes starting from address $XXX0 or $XXX8. The purpose of the verify mode is to ensure that data has been programmed with sufficient margin for long-term data retention. During verify, the control gates of the selected memory bits are held at a slightly negative voltage by an internal charge pump. Reading the data is the same as for ordinary read mode except that a built-in counter stretches the data access for an additional eight cycles to allow sensing of the lower cell current. A verify can only follow a program operation.

Execute these steps to program and verify the FLASH memory. Values for the time parameters are specified in **19.13 Memory Characteristics**.

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 block protect register.
3. Write data to the eight bytes of the page being programmed. This requires eight separate write operations.
4. Set the HVEN bit.
5. Wait for a time, t_{Step}.
6. Clear the HVEN bit.
7. Wait for a time, t_{HVTV}.
8. Set the VERF bit.
9. Wait for a time, t_{VTP}.
10. Clear the PGM bit.
11. Wait for a time, t_{HVD}.
12. Read back data in verify mode. This is done in eight separate read operations which are each stretched by eight cycles.
13. Clear the VERF bit.
14. After a time, $t_{Recovery}$, the memory can be accessed in read mode again.

NOTE: *While these operations must be performed in the order shown, other unrelated operations may occur between the steps.*

This program/verify sequence is repeated throughout the memory until all data is programmed. For minimum overall programming time and least program disturb effect, the sequence should be part of an intelligent operation which iterates per page (see **4.6 FLASH Erase Operation**).

4.8 Block Protection

Due to the ability of the on-board charge pump to erase and program the FLASH memory in the target application, provision is made for protecting blocks of memory from unintentional erase or program operations due to system malfunction. This protection is done by reserving a location in the memory for block protect information and requiring that this location be read from to enable setting of the HVEN bit.

When the block protect register is read, its contents are latched by the FLASH control logic. If the address range for an erase or program operation includes a protected block, the PGM or ERASE bit is cleared which prevents the HVEN bit in the FLASH control register from being set so that no high voltage is allowed in the array.

When the block protect register is erased (all 0s), the entire memory is accessible for program and erase. When bits within the register are programmed, they lock blocks of memory address ranges as shown in **4.9 FLASH Block Protect Register**. The presence of a voltage V_{TST} on the $\overline{IRQ1}$ pin will bypass the block protection so that all of the memory, including the block protect register, is open for program and erase operations.

4.9 FLASH Block Protect Register

The block protect register (FLBPR) is implemented as a byte within the FLASH memory. The erased state is logical 0. Each bit, when programmed, protects a range of addresses in the FLASH.

Address: $FF80

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	BPR3	BPR2	BPR1	BPR0
Write:								
Reset:	0	0	0	0	Unaffected by reset			

= Unimplemented

Figure 4-2. FLASH Block Protect Register (FLBPR)

BPR3 — Block Protect Register Bit 3

This bit protects the memory contents in the address range $C000 to $FFFF.

1 = Address range protected from erase or program
0 = Address range open to erase or program

BPR2 — Block Protect Register Bit 2

This bit protects the memory contents in the address range $A000 to $FFFF.

1 = Address range protected from erase or program
0 = Address range open to erase or program

BPR1 — Block Protect Register Bit 1

In a larger memory this bit would protect the memory contents in the address range $9000 to $FFFF. It is redundant in this implementation. Setting this bit locks everything from $A000 to $FFFF.

1 = Address range protected from erase or program
0 = Address range open to erase or program

BPR0 — Block Protect Register Bit 0

In a larger memory this bit would protect the memory contents in the address range $8000 to $FFFF. It is redundant in this implementation. Setting this bit locks everything from $A000 to $FFFF.

1 = Address range protected from erase or program
0 = Address range open to erase or program

By programming the block protect bits, a portion of the memory will be locked so that no further erase or program operations may be performed. Programming more than one bit at a time is redundant. If both bit 3 and bit 2 are set, for instance, the address range $A000 through $FFFF is locked. If all bits are erased, then all of the memory is available for erase and program. The presence of a voltage V_{TST} on the $\overline{IRQ1}$ pin will bypass the block protection so that all of the memory, including the block protect register, is open for program and erase operations.

Section 5. Central Processor Unit (CPU)

5.1 Contents

5.2 Introduction

The M68HC08 CPU 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.

5.3 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

5.4 CPU Registers

Figure 5-1 shows the five CPU registers. CPU registers are not part of the memory map.

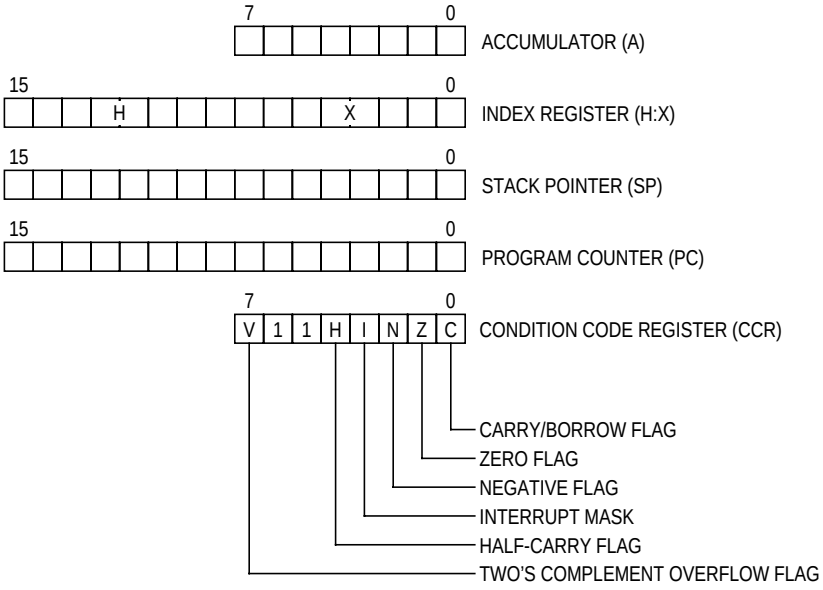


Figure 5-1. CPU Registers

5.4.1 Accumulator

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

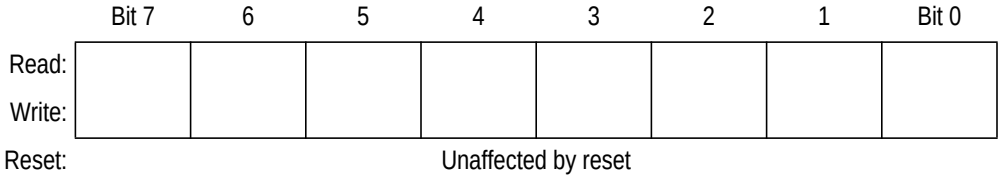


Figure 5-2. Accumulator (A)

5.4.2 Index Register

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

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

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

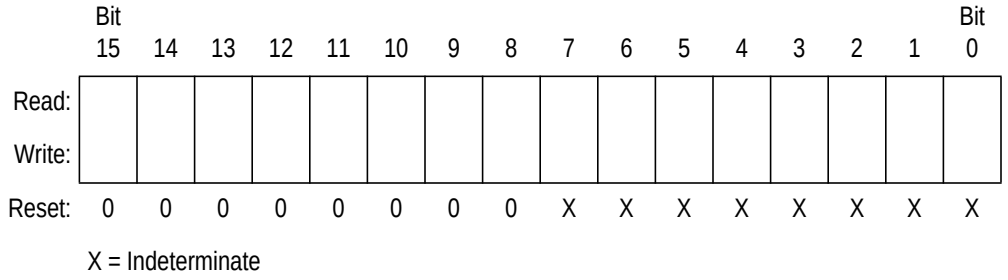


Figure 5-3. Index Register (H:X)

5.4.3 Stack Pointer

The stack pointer is a 16-bit register that contains the address of the next location on the stack. During a reset, the stack pointer is preset to $00FF. The reset stack pointer (RSP) instruction sets the least significant byte (LSB) to $FF and does not affect the most significant byte (MSB). 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.

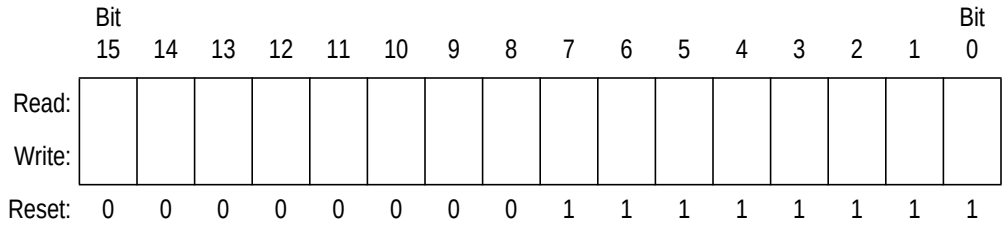


Figure 5-4. Stack Pointer (SP)

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

5.4.4 Program Counter

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

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

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

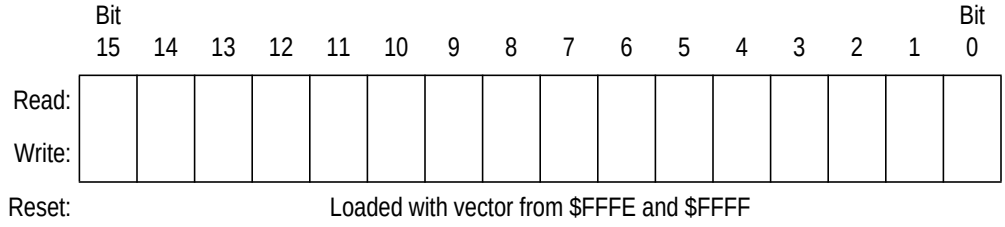


Figure 5-5. Program Counter (PC)

5.4.5 Condition Code Register

The 8-bit condition code register contains the interrupt mask and five flags that indicate the results of the instruction just executed. Bits 6 and 5 are set permanently to logic 1. The following paragraphs describe the functions of the condition code register.

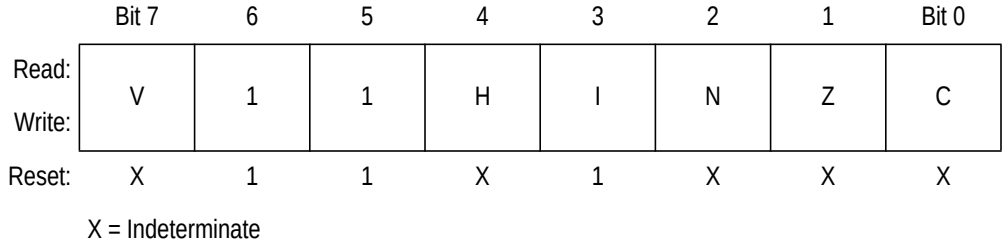


Figure 5-6. Condition Code Register (CCR)

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 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 produce 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

5.5 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.

5.6 Low-Power Modes

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

5.6.1 Wait Mode

The WAIT instruction:

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

5.6.2 Stop Mode

The STOP instruction:

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

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

5.7 CPU During Break Interrupts

If the break module is enabled, a break interrupt causes the CPU to execute the software interrupt instruction (SWI) at the completion of the current CPU instruction. (See **Section 18. Break Module (BRK)**.) The program counter vectors to $FFFC–$FFFD ($FEFC–$FEFD in monitor mode).

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.

5.8 Instruction Set Summary

Table 5-1 provides a summary of the M68HC08 instruction set.

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

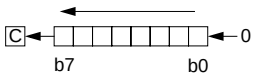
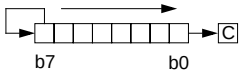
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X ADC *opr*,SP ADC *opr*,SP	Add with Carry	A ← (A) + (M) + (C)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A9 B9 C9 D9 E9 F9 9EE9 9ED9	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X ADD *opr*,SP ADD *opr*,SP	Add without Carry	A ← (A) + (M)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	AB BB CB DB EB FB 9EEB 9EDB	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
AIS #*opr*	Add Immediate Value (Signed) to SP	SP ← (SP) + (16 « M)	–	–	–	–	–	–	IMM	A7	ii	2
AIX #*opr*	Add Immediate Value (Signed) to H:X	H:X ← (H:X) + (16 « M)	–	–	–	–	–	–	IMM	AF	ii	2
AND #*opr* AND *opr* AND *opr* AND *opr*,X AND *opr*,X AND ,X AND *opr*,SP AND *opr*,SP	Logical AND	A ← (A) & (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A4 B4 C4 D4 E4 F4 9EE4 9ED4	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X ASL *opr*,SP	Arithmetic Shift Left (Same as LSL)		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR *opr*,X ASR *opr*,SP	Arithmetic Shift Right		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	37 47 57 67 77 9E67	dd ff ff	4 1 1 4 3 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + rel ? (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BCLR *n*, *opr*	Clear Bit n in M	Mn ← 0	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	4 4 4 4 4 4 4 4
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? (C) = 1	–	–	–	–	–	–	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? (Z) = 1	–	–	–	–	–	–	REL	27	rr	3
BGE *opr*	Branch if Greater Than or Equal To (Signed Operands)	PC ← (PC) + 2 + *rel* ? (N ⊕ V) = 0	–	–	–	–	–	–	REL	90	rr	3
BGT *opr*	Branch if Greater Than (Signed Operands)	PC ← (PC) + 2 + *rel* ? (Z) | (N ⊕ V) = 0	–	–	–	–	–	–	REL	92	rr	3

Table 5-1. 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				
BHCC *rel*	Branch if Half Carry Bit Clear	PC ← (PC) + 2 + *rel* ? (H) = 0	–	–	–	–	–	–	REL	28	rr	3
BHCS *rel*	Branch if Half Carry Bit Set	PC ← (PC) + 2 + *rel* ? (H) = 1	–	–	–	–	–	–	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? (C) | (Z) = 0	–	–	–	–	–	–	REL	22	rr	3
BHS *rel*	Branch if Higher or Same (Same as BCC)	PC ← (PC) + 2 + *rel* ? (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BIH *rel*	Branch if $\overline{\text{IRQ}}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 1	–	–	–	–	–	–	REL	2F	rr	3
BIL *rel*	Branch if $\overline{\text{IRQ}}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 0	–	–	–	–	–	–	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X BIT *opr*,SP BIT *opr*,SP	Bit Test	(A) & (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A5 B5 C5 D5 E5 F5 9EE5 9ED5	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
BLE *opr*	Branch if Less Than or Equal To (Signed Operands)	PC ← (PC) + 2 + *rel* ? (Z) | (N ⊕ V) = 1	–	–	–	–	–	–	REL	93	rr	3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? (C) = 1	–	–	–	–	–	–	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? (C) | (Z) = 1	–	–	–	–	–	–	REL	23	rr	3
BLT *opr*	Branch if Less Than (Signed Operands)	PC ← (PC) + 2 + *rel* ? (N ⊕ V) =1	–	–	–	–	–	–	REL	91	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? (I) = 0	–	–	–	–	–	–	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? (N) = 1	–	–	–	–	–	–	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? (I) = 1	–	–	–	–	–	–	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? (Z) = 0	–	–	–	–	–	–	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? (N) = 0	–	–	–	–	–	–	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel*	–	–	–	–	–	–	REL	20	rr	3
BRCLR *n*,*opr*,*rel*	Branch if Bit *n* in M Clear	PC ← (PC) + 3 + *rel* ? (Mn) = 0	–	–	–	–	–	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2	–	–	–	–	–	–	REL	21	rr	3
BRSET *n*,*opr*,*rel*	Branch if Bit *n* in M Set	PC ← (PC) + 3 + *rel* ? (Mn) = 1	–	–	–	–	–	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5

Table 5-1. 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				
BSET *n*,*opr*	Set Bit *n* in M	Mn ← 1	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	4 4 4 4 4 4 4 4
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	–	–	–	–	–	–	REL	AD	rr	4
CBEQ *opr,rel* CBEQA #*opr,rel* CBEQX #*opr,rel* CBEQ *opr*,X+,*rel* CBEQ X+,*rel* CBEQ *opr*,SP,*rel*	Compare and Branch if Equal	PC ← (PC) + 3 + rel ? (A) – (M) = $00 PC ← (PC) + 3 + rel ? (A) – (M) = $00 PC ← (PC) + 3 + rel ? (X) – (M) = $00 PC ← (PC) + 3 + rel ? (A) – (M) = $00 PC ← (PC) + 2 + rel ? (A) – (M) = $00 PC ← (PC) + 4 + rel ? (A) – (M) = $00	–	–	–	–	–	–	DIR IMM IMM IX1+ IX+ SP1	31 41 51 61 71 9E61	dd rr ii rr ii rr ff rr rr ff rr	5 4 4 5 4 6
CLC	Clear Carry Bit	C ← 0	–	–	–	–	–	0	INH	98		1
CLI	Clear Interrupt Mask	I ← 0	–	–	0	–	–	–	INH	9A		2
CLR *opr* CLRA CLRX CLRH CLR *opr*,X CLR ,X CLR *opr*,SP	Clear	M ← $00 A ← $00 X ← $00 H ← $00 M ← $00 M ← $00 M ← $00	0	–	–	0	1	–	DIR INH INH INH IX1 IX SP1	3F 4F 5F 8C 6F 7F 9E6F	dd ff ff	3 1 1 1 3 2 4
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X CMP *opr*,SP CMP *opr*,SP	Compare A with M	(A) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A1 B1 C1 D1 E1 F1 9EE1 9ED1	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
COM *opr* COMA COMX COM *opr*,X COM ,X COM *opr*,SP	Complement (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (M) X ← ($\overline{X}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	0	–	–	↕	↕	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd ff ff	4 1 1 4 3 5
CPHX #*opr* CPHX *opr*	Compare H:X with M	(H:X) – (M:M + 1)	↕	–	–	↕	↕	↕	IMM DIR	65 75	ii ii+1 dd	3 4
CPX #*opr* CPX *opr* CPX *opr* CPX ,X CPX *opr*,X CPX *opr*,X CPX *opr*,SP CPX *opr*,SP	Compare X with M	(X) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A3 B3 C3 D3 E3 F3 9EE3 9ED3	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
DAA	Decimal Adjust A	$(A)_{10}$	U	–	–	↕	↕	↕	INH	72		2

Table 5-1. 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				
DBNZ *opr,rel* DBNZA *rel* DBNZX *rel* DBNZ *opr*,X,*rel* DBNZ X,*rel* DBNZ *opr*,SP,*rel*	Decrement and Branch if Not Zero	A ← (A) – 1 or M ← (M) – 1 or X ← (X) – 1 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 4 + *rel* ? (result) ≠ 0	–	–	–	–	–	–	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	5 3 3 5 4 6
DEC *opr* DECA DECX DEC *opr*,X DEC ,X DEC *opr*,SP	Decrement	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1 M ← (M) – 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff	4 1 1 4 3 5
DIV	Divide	A ← (H:A)/(X) H ← Remainder	–	–	–	–	↕	↕	INH	52		7
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X EOR *opr*,SP EOR *opr*,SP	Exclusive OR M with A	A ← (A ⊕ M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A8 B8 C8 D8 E8 F8 9EE8 9ED8	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
INC *opr* INCA INCX INC *opr*,X INC ,X INC *opr*,SP	Increment	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1 M ← (M) + 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff	4 1 1 4 3 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff 	2 3 4 3 2
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + *n* (*n* = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Unconditional Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff 	4 5 6 5 4
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X LDA *opr*,SP LDA *opr*,SP	Load A from M	A ← (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A6 B6 C6 D6 E6 F6 9EE6 9ED6	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LDHX #*opr* LDHX *opr*	Load H:X from M	H:X ← (M:M + 1)	0	–	–	↕	↕	–	IMM DIR	45 55	ii jj dd	3 4
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X LDX *opr*,SP LDX *opr*,SP	Load X from M	X ← (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	AE BE CE DE EE FE 9EEE 9EDE	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5

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

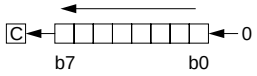
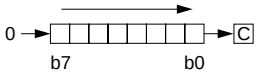
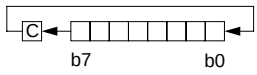
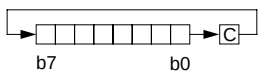
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X LSL *opr*,SP	Logical Shift Left (Same as ASL)		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
LSR *opr* LSRA LSR*X* LSR *opr*,X LSR ,X LSR *opr*,SP	Logical Shift Right		↕	–	–	0	↕	↕	DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff	4 1 1 4 3 5
MOV *opr,opr* MOV *opr*,X+ MOV #*opr,opr* MOV X+,*opr*	Move	$(M)_{Destination} \leftarrow (M)_{Source}$ H:X ← (H:X) + 1 (IX+D, DIX+)	0	–	–	↕	↕	–	DD DIX+ IMD IX+D	4E 5E 6E 7E	dd dd dd ii dd dd	5 4 4 4
MUL	Unsigned multiply	X:A ← (X) × (A)	–	0	–	–	–	0	INH	42		5
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X NEG *opr*,SP	Negate (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	30 40 50 60 70 9E60	dd ff ff	4 1 1 4 3 5
NOP	No Operation	None	–	–	–	–	–	–	INH	9D		1
NSA	Nibble Swap A	A ← (A[3:0]:A[7:4])	–	–	–	–	–	–	INH	62		3
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X ORA *opr*,SP ORA *opr*,SP	Inclusive OR A and M	A ← (A) | (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	AA BA CA DA EA FA 9EEA 9EDA	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
PSHA	Push A onto Stack	Push (A); SP ← (SP) – 1	–	–	–	–	–	–	INH	87		2
PSHH	Push H onto Stack	Push (H); SP ← (SP) – 1	–	–	–	–	–	–	INH	8B		2
PSHX	Push X onto Stack	Push (X); SP ← (SP) – 1	–	–	–	–	–	–	INH	89		2
PULA	Pull A from Stack	SP ← (SP + 1); Pull (A)	–	–	–	–	–	–	INH	86		2
PULH	Pull H from Stack	SP ← (SP + 1); Pull (H)	–	–	–	–	–	–	INH	8A		2
PULX	Pull X from Stack	SP ← (SP + 1); Pull (X)	–	–	–	–	–	–	INH	88		2
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X ROL *opr*,SP	Rotate Left through Carry		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	39 49 59 69 79 9E69	dd ff ff	4 1 1 4 3 5
ROR *opr* RORA RORX ROR *opr*,X ROR ,X ROR *opr*,SP	Rotate Right through Carry		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	4 1 1 4 3 5

Table 5-1. 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				
RSP	Reset Stack Pointer	SP ← $FF	–	–	–	–	–	–	INH	9C		1
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	↕	INH	80		7
RTS	Return from Subroutine	SP ← SP + 1; Pull (PCH) SP ← SP + 1; Pull (PCL)	–	–	–	–	–	–	INH	81		4
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X SBC *opr*,SP SBC *opr*,SP	Subtract with Carry	A ← (A) – (M) – (C)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A2 B2 C2 D2 E2 F2 9EE2 9ED2	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
SEC	Set Carry Bit	C ← 1	–	–	–	–	–	1	INH	99		1
SEI	Set Interrupt Mask	I ← 1	–	–	1	–	–	–	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X STA *opr*,SP STA *opr*,SP	Store A in M	M ← (A)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh ll ee ff ff ff ee ff	3 4 4 3 2 4 5
STHX *opr*	Store H:X in M	(M:M + 1) ← (H:X)	0	–	–	↕	↕	–	DIR	35	dd	4
STOP	Enable $\overline{\text{IRQ}}$ Pin; Stop Oscillator	I ← 0; Stop Oscillator	–	–	0	–	–	–	INH	8E		1
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X STX *opr*,SP STX *opr*,SP	Store X in M	M ← (X)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP1 SP2	BF CF DF EF FF 9EEF 9EDF	dd hh ll ee ff ff ff ee ff	3 4 4 3 2 4 5
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X SUB *opr*,SP SUB *opr*,SP	Subtract	A ← (A) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A0 B0 C0 D0 E0 F0 9EE0 9ED0	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	–	–	1	–	–	–	INH	83		9
TAP	Transfer A to CCR	CCR ← (A)	↕	↕	↕	↕	↕	↕	INH	84		2
TAX	Transfer A to X	X ← (A)	–	–	–	–	–	–	INH	97		1

Table 5-1. 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				
TPA	Transfer CCR to A	A ← (CCR)	–	–	–	–	–	–	INH	85		1
TST *opr* TSTA TSTX TST *opr*,X TST ,X TST *opr*,SP	Test for Negative or Zero	(A) – $00 or (X) – $00 or (M) – $00	0	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3D 4D 5D 6D 7D 9E6D	dd ff ff	3 1 1 3 2 4
TSX	Transfer SP to H:X	H:X ← (SP) + 1	–	–	–	–	–	–	INH	95		2
TXA	Transfer X to A	A ← (X)	–	–	–	–	–	–	INH	9F		1
TXS	Transfer H:X to SP	(SP) ← (H:X) – 1	–	–	–	–	–	–	INH	94		2

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

5.9 Opcode Map

The opcode map is provided in **Table 5-2**.

Table 5-2. Opcode Map

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

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

*Pre-byte for stack pointer indexed instructions

MSB / LSB	0
0	5 BRSET0 3 DIR

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

Section 6. Resets and Interrupts

6.1 Contents

6.2 Introduction

Resets and interrupts are responses to exceptional events during program execution. A reset re-initializes the MCU to its startup condition. An interrupt vectors the program counter to a service routine.

6.3 Resets

A reset immediately returns the MCU to a known startup condition and begins program execution from a user-defined memory location. **Figure 6-1** shows the structure of the reset circuits.

NOTE: *The reset circuits are powered by the chip V_{DD}. External circuits should drive the $\overline{RST}$ pin through a resistor between V_{DD} and V_{SS} levels. If external circuits use a pullup device on $\overline{RST}$, the pin must be pulled to V_{DD} and **not** to BATT.*

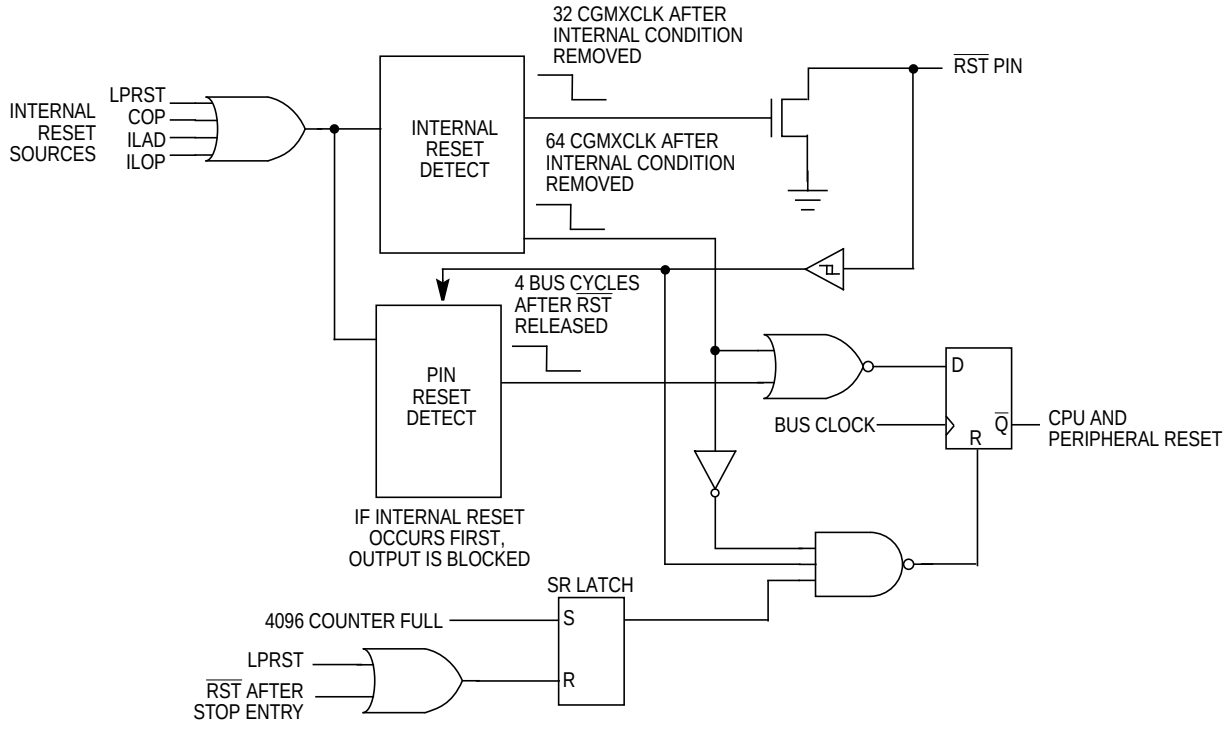


Figure 6-1. Reset Block Diagram

6.3.1 Reset Entry

Reset entry can be caused by an internal or external event. External events control reset by driving the $\overline{RST}$ pin.

Internal resets, which pull the $\overline{RST}$ pin low, have several sources:

- Computer operating properly (COP)
- Illegal opcode
- Illegal address
- Low-power reset

Reset entry immediately stops the operation of the instruction being executed.

6.3.1.1 External Reset

A logic 0 applied to the $\overline{RST}$ pin for a time, t_{IRL}, generates an external reset. After reset recovery, the PIN bit in the reset status register will be set.

6.3.1.2 COP Reset

A COP reset is an internal reset caused by an overflow of the COP counter. After reset recovery, the COP bit in the reset status register will be set.

To clear the COP counter and prevent a COP reset, write any value to the COP control register at location $FFFF.

6.3.1.3 Illegal Opcode Reset

An illegal opcode reset is an internal reset caused by an opcode that is not in the instruction set. After reset recovery, the ILOP bit in the reset status register will be set.

If the stop enable bit, STOP, in the configuration register is a logic 0, the STOP instruction causes an illegal opcode reset.

6.3.1.4 Illegal Address Reset

An illegal address reset is an internal reset caused by opcode fetch from an unmapped address. After reset recovery, the ILAD bit in the reset status register will be set.

A data fetch from an unmapped address does not generate a reset.

6.3.1.5 Low-Power Reset

The MC68HC908RC24 is designed for remote control applications and has on-chip circuits that force the MCU into low-power reset mode to preserve RAM contents. The low-power reset mode is entered whenever batteries have been removed or when the V_{DD} voltage is detected below the V_{LVR} voltage (a low battery condition exists). The V_{LVR} threshold is defined in **19.10 LVI Characteristics**.

A low-power reset:

- Puts the MCU into its low-power reset mode where the clocks to the CPU and modules are disabled
- Requires the BATT pin to be pulled low and battery reinsertion to exit this mode
- Upon exit, the system will go through the power-on reset recovery sequence.
- After reset recovery, the POR and LP bits in the reset status register will be set.

6.3.2 Reset Recovery

Reset recovery:

- Initializes certain control and status bits
- Loads the program counter with a user-defined reset vector address from locations $FFFE and $FFFF
- Selects CGMXCLK divided by four as the bus clock

6.3.2.1 External Reset Recovery

Figure 6-2 shows the relative timing of an external reset recovery.

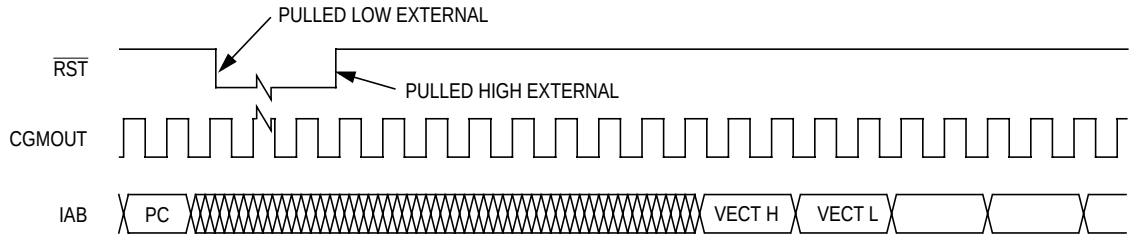


Figure 6-2. External Reset Recovery Timing

6.3.2.2 Active Reset Recovery

An active reset recovery occurs immediately after internal resets from a COP, illegal address, or illegal opcode have occurred. These reset sources pull the $\overline{\text{RST}}$ pin low for 32 CGMXCLK cycles to allow resetting of external devices. The MCU is held in reset for an additional 32 CGMXCLK cycles after releasing the $\overline{\text{RST}}$ pin. **Figure 6-3** shows the relative timing of an active reset recovery.

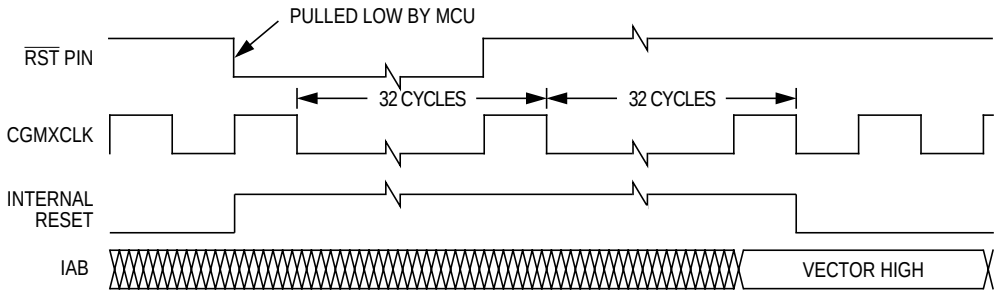


Figure 6-3. Active Reset Recovery Timing

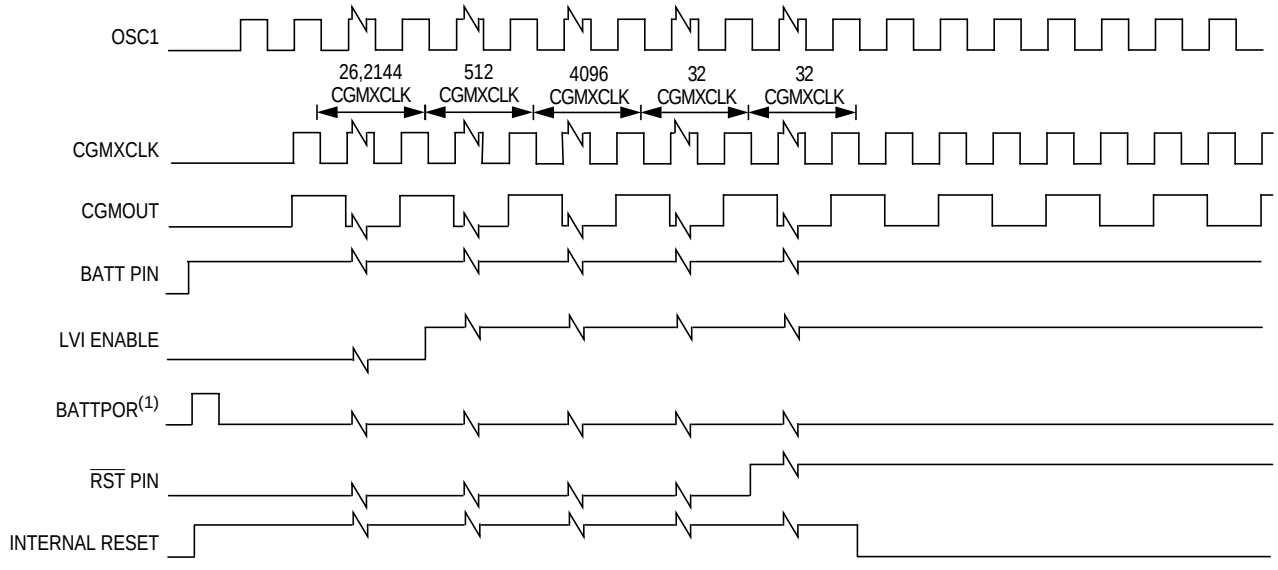
6.3.2.3 Power-On Reset (POR) Recovery

A power-on reset (POR) is an internal reset caused by a positive transition on the BATT pin. This will occur when the entire system is initially powered up or when the system is experiencing a battery change.

A power-on reset:

- After the batteries have been inserted, holds the clocks to the CPU and modules inactive for 262,144 CGMXCLK cycles for the external capacitor charge time
- Keeps the clocks inactive for an additional 512 CGMXCLK cycles for LVI enable time: the LVI will force re-entry into low-power reset mode if V_{DD} is below the V_{LVR} + H_{LVR} voltage
- Keeps the clocks inactive for a system stabilization delay of 4096 CGMXCLK cycles
- Drives the $\overline{RST}$ pin low during the startup process and releases the $\overline{RST}$ pin 32 CGMXCLK cycles after the system stabilization delay
- Releases the CPU to begin the reset vector sequence 64 CGMXCLK cycles after the system stabilization delay
- Sets the POR and LP bits in the reset status register and clears all other bits in the register

Figure 6-4 shows the relative timing of a power-on reset recovery.



Note 1: POR is an internally generated power-on reset pulse.

Figure 6-4. Power-on Reset (POR) Recovery

6.3.3 Reset Status Register

This read-only register contains flags to show reset sources. All flag bits are cleared automatically following a read of the register. Reset service can read the reset status register to clear the register after power-on reset and to determine the source of any subsequent reset.

The register is initialized on power-up as shown with the POR and the LP bits set and all other bits cleared. During a POR or any other internal reset, the $\overline{RST}$ pin is pulled low. After the pin is released, it will be sampled 32 CGMXCLK cycles later. If the pin is not above a V_{IH} at that time, the PIN bit in the RSR may be set in addition to whatever other bits are set.

NOTE: *Only a read of the reset status register clears all reset flags. After multiple resets from different sources without reading the register, multiple flags remain set.*

Address: $FE01

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

= Unimplemented X = Indeterminate

Figure 6-5. Reset Status Register (RSR)

POR — Power-On Reset Flag
- 1 = Power-on reset since last read of RSR
- 0 = Read of RSR since last power-on reset

PIN — External Reset Flag
- 1 = External reset since last read of RSR
- 0 = Power-on reset or read of RSR since last external reset

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

ILOP — Illegal Opcode Reset Bit

1 = Last reset caused by an illegal opcode
0 = POR or read of RSR

ILAD — Illegal Address Reset Bit

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

LPRST — Low-Power Mode Reset Bit

1 = Last reset caused by low-power supply voltage
0 = POR or read of RSR

6.3.4 Reset States

A brief description of how the various resets initialize the MCU is given here.

- Central processor unit (CPU):
  - Accumulator (A) — Unaffected
  - Index register (H:X) — Bits 15:8 cleared, bits 7:0 indeterminate
  - Stack pointer (SP) — Loaded with $FF
  - Program counter (PC) — Loaded with vector from locations $FFFE and $FFFF
  - Condition code register (CCR) — I bit is set, other bits indeterminate
- Resets:
  - Reset status register (RSR) — Bit indicating last cause of reset is set, other bits are unaffected by reset
- Interrupts:
  - Interrupt status register (INT1) — Cleared
- Low-voltage inhibit (LVI) reset:
  - LVI status register (LVISR) — Unaffected
- Configuration register (CONFIG):
  - CONFIG — Value fixed at time of manufacture

- Input/output (I/O) ports:
  - Data registers (PTA, PTB, and PTC) — Unaffected
  - Data direction registers (DDRA, DDRB, and DDRC) — Cleared
- Carrier modulator transmitter (CMT):
  - CMT primary carrier data registers (CCH1/CCL1) — IROLN, IROLP, and CMTPOL are cleared, all other bits are unaffected
  - CMT secondary carrier data registers (CCH2/CCL2) — Unaffected
  - CMT modulator control and status register (CMCS) — Cleared
  - CMT modulator data register (CMD1, CMD2, and CMD3) — Unaffected
- Modulo timer (TIM0I):
  - TIM status and control register (TSC) — TSTOP set, other bits cleared
  - TIM counter registers (TCNTH/TCNTL) — Cleared
  - TIM counter modulo registers (TMODH/TMODL) — Set
- External interrupt (IRQ):
  - IRQ status and control register (ISCR) — Cleared
- Keyboard interrupt module (KBI):
  - Keyboard status and control register (KBSCR) — Cleared
  - Keyboard interrupt enable register (KBIER) — Cleared
- Computer operating properly (COP):
  - COP control register (COPCTL) — Unaffected
- Break module (BRK):
  - Break status register (BSR) — Cleared
  - Break flag control register (BFCR) — Cleared
  - Break address registers (BRKH/BRKL) — Cleared
  - Break status and control register (BSCR) — Cleared

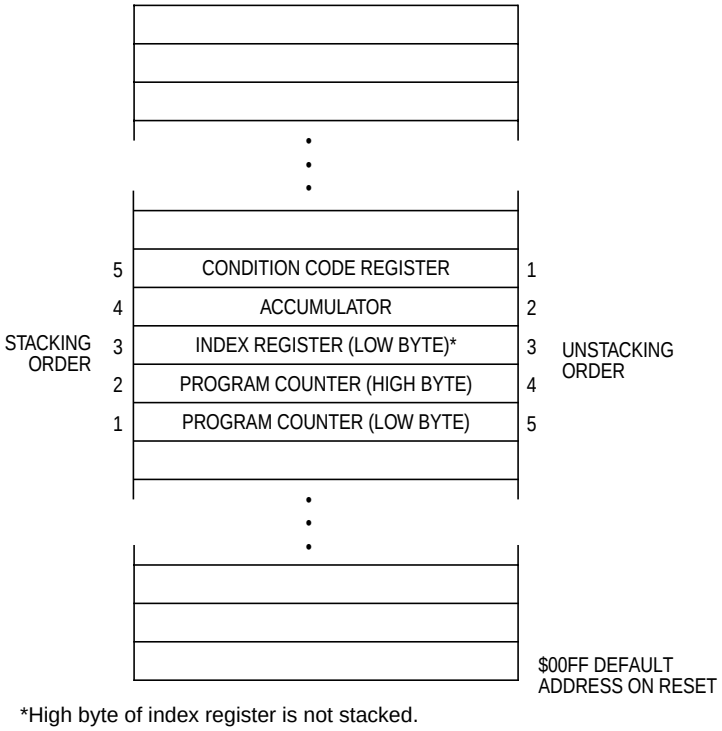
6.4 Interrupts

An interrupt temporarily changes the sequence of program execution to respond to a particular event. An interrupt does not stop the operation of the instruction being executed, but begins when the current instruction completes its operation.

6.4.1 Effects

An interrupt:

- Saves the CPU registers on the stack. At the end of the interrupt, the RTI instruction recovers the CPU registers from the stack so that normal processing can resume. See **Figure 6-6**.
- Sets the interrupt mask (I bit) to prevent additional interrupts. Once an interrupt is latched, no other interrupt can take precedence, regardless of its priority.
- Loads the program counter with a user-defined vector address



*High byte of index register is not stacked.

Figure 6-6. Interrupt Stacking Order

After every instruction, the CPU checks all pending interrupts if the I bit is not set. If more than one interrupt is pending when an instruction is done, the highest priority interrupt is serviced first. In the example shown in **Figure 6-7**, if an interrupt is pending upon exit from the interrupt service routine, the pending interrupt is serviced before the load accumulator from memory (LDA) instruction is executed.

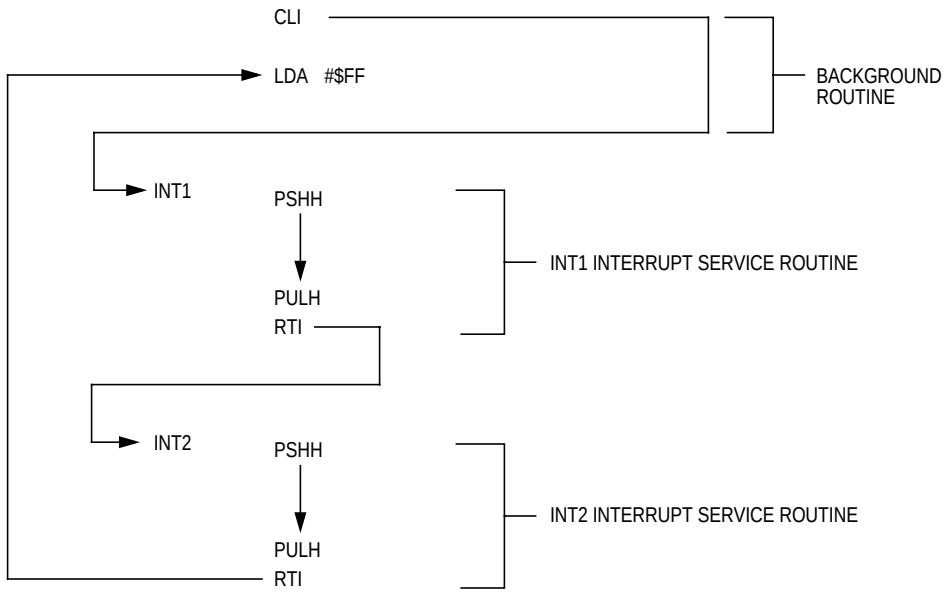


Figure 6-7. Interrupt Recognition Example

The LDA opcode is prefetched by both the interrupt 1 (INT1) and interrupt 2 (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, save the H register and then restore it prior to exiting the routine.*

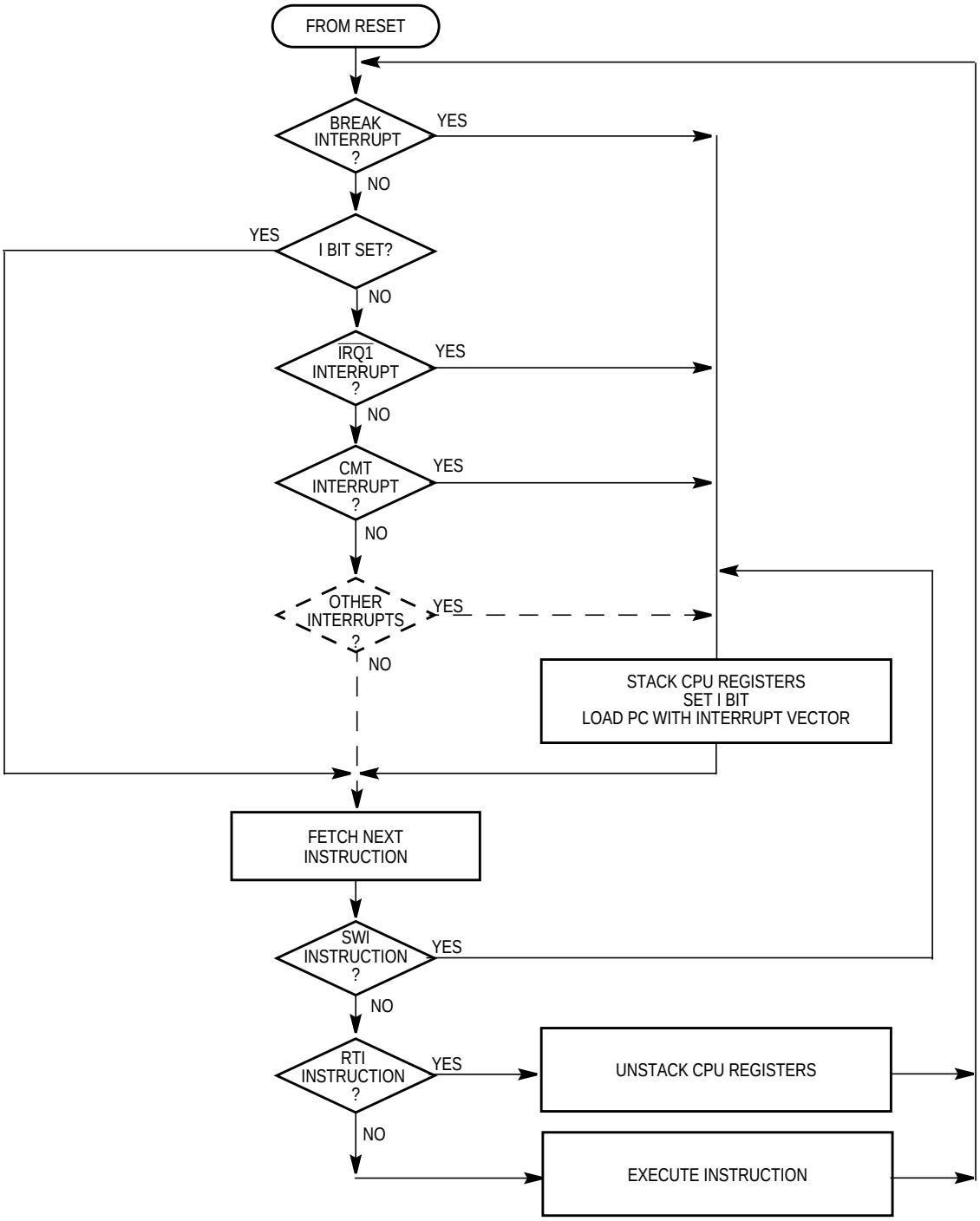


Figure 6-8. Interrupt Processing

6.4.2 Sources

The sources shown in **Table 6-1** can generate CPU interrupt requests.

Table 6-1. Interrupt Sources

Source	Flag	Mask(1)	INT Register Flag	Priority(2)	Vector Address
SWI instruction	None	None	None	0	$FFFC–$FFFD
$\overline{\text{IRQ1}}$ pin	IRQ1F	IMASK1	IF1	1	$FFFA–$FFFB
CMT end of cycle	EOCF	EOCIE	IF2	2	$FFF8–$FFF9
TIM0I overflow	TOF	TOIE	IF3	3	$FFF6–$FFF7
Keyboard pin	KEYF	IMASKK	IF4	4	$FFF4–$FFF5

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

6.4.2.1 SWI Instruction

The software interrupt instruction (SWI) causes a non-maskable interrupt.

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

6.4.2.2 Break Interrupt

The break module causes the CPU to execute an SWI instruction at a software-programmable break point.

6.4.2.3 $\overline{\text{IRQ1}}$ Pin

A logic 0 on the $\overline{\text{IRQ1}}$ pin latches an external interrupt request.

6.4.2.4 CMT Interrupt

The end-of-cycle flag (EOCF) can generate an interrupt request. The EOCF bit is set at the end of each modulation cycle. The EOC interrupt

enable (EOCIE) bit in the CMT control and status register enables CPU interrupt requests.

6.4.2.5 TIM0I Interrupt

The TIM0I overflow flag (TOF) can generate an interrupt request. The TOF bit is set when the TIM0I counter value rolls over to $0000 after matching the value in the TIM0I counter modulo registers. The TIM0I overflow interrupt enable bit, TOIE, enables CPU interrupt requests. TOF and TOIE are in the TIM0I status and control register.

6.4.2.6 $\overline{KBD0}$–$\overline{KBD7}$ Pins

A logic 0 on a keyboard interrupt pin latches an external interrupt request.

6.4.3 Interrupt Status Registers

The flags in the interrupt status registers identify maskable interrupt sources. **Table 6-2** summarizes the interrupt sources and the interrupt status register flags that they set. The interrupt status registers can be useful for debugging.

Table 6-2. Interrupt Source Flags

Interrupt Source	Interrupt Status Register Flag
Reset	—
SWI instruction	—
$\overline{IRQ1}$ pin	IF1
CMT	IF2
TIM0I	IF3
Keyboard pin	IF4

6.4.3.1 *Interrupt Status Register 1*

Address: $FE04

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

R = Reserved

Figure 6-9. Interrupt Status Register 1 (INT1)

IF6–IF5 — Interrupt Flags 6–5

Since the MC68HC908RC24 does not use these interrupt flags, these bits will always read 0.

IF4–IF1 — Interrupt Flags 4–1

These flags indicate the presence of interrupt requests from the sources shown in **Table 6-2**.

1 = Interrupt request present
0 = No interrupt request present

Bit 0 and Bit 1 — Always read 0

6.4.3.2 Interrupt Status Register 2

Address: $FE05

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

R = Reserved

Figure 6-10. Interrupt Status Register 2 (INT2)

IF14–IF7 — Interrupt Flags 14–7

Since the MC68HC908RC24 does not use these interrupt flags, these bits will always read 0.

Section 7. Low-Power Modes

7.1 Contents

7.2 Introduction

The MCU may enter three low-power modes. Two of the modes, wait mode and stop mode, are common to all HC08 MCUs and are entered through instruction execution. A third low-power mode, the low-power reset mode (LPRST), is entered when a low voltage condition exists.

7.3 Wait Mode

The WAIT instruction puts the MCU in a low-power standby mode in which the CPU clock is disabled but the bus clock continues to run.

7.4 Stop Mode

Stop mode is entered when a STOP instruction is executed. Both the CPU clock and the bus clock are disabled in stop mode.

7.5 Low-Power Reset Mode

The MC68HC908RC24 is designed for remote control applications and has on-chip circuits that force the MCU into low-power reset mode to preserve RAM contents. The low-power reset mode is entered whenever batteries have been removed or when the V_{DD} voltage is detected below the V_{LVR} voltage (a low battery condition exists). The V_{LVR} threshold is defined in **19.10 LVI Characteristics**.

A low-power reset:

- Puts the MCU into its low-power reset mode where the clocks to the CPU and modules are disabled
- Requires the BATT pin to be pulled low and battery reinsertion to exit this mode
- Upon exit, the system will go through the power-on reset recovery sequence.
- After reset recovery, the POR and LP bits in the reset status register will be set.

7.5.1 External Circuits

The required external components for low-power mode operation are shown in **Figure 7-1**. Refer to **19.11 Battery Detection Characteristics** for full component specifications. Power supply for the MCU is provided through the BATT pin.

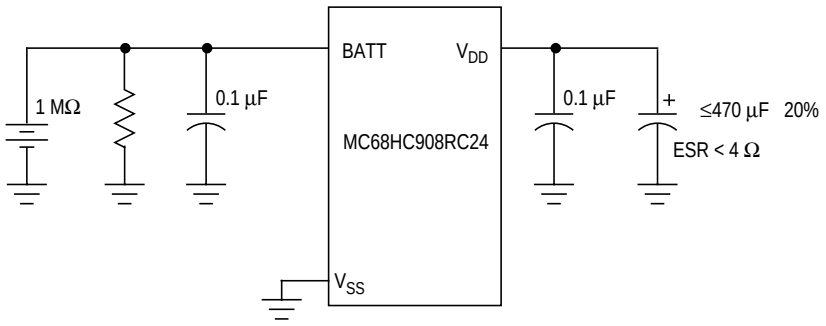


Figure 7-1. External Low-Power Circuit

NOTE: *Other pin connections necessary for proper MCU operation (for example, $\overline{IRQ1}$ and $\overline{RST}$) are not shown. The reset and external interrupt circuits are powered by the chip V_{DD}. External circuits should drive these pins between V_{DD} and V_{SS} levels. If external circuits use a pullup device on $\overline{RST}$ or $\overline{IRQ1}$, the pins must be pulled to V_{DD} and* ***not*** *to BATT.*

7.5.2 Low-Power Reset Operation

A block diagram of the low-power reset circuit is shown in **Figure 7-2**. The LPRST signal controls entry and exit from the low power reset mode. Low-power reset mode is entered whenever the batteries are removed or when V_{DD} is detected below V_{LVR} and is exited upon battery insertion.

The V_{DD} isolator is a MOSFET transistor connected to the BATT pin. This circuit blocks discharging of the external bulk capacitor when batteries have been removed.

When the batteries are removed, the external 1-MΩ pulldown resistor will force the BATT pin low. The external bulk capacitor continues to source the chip V_{DD}. The removed battery is detected and asserts LPRST. The low-power reset condition cannot be cleared until batteries have been reinstalled.

When new batteries are inserted, a 262,144 CGMXCLK cycle delay is started. This delay is used for charging the external bulk capacitor through the V_{DD} isolator. This delay (32 ms with an 8-MHz external crystal) provides enough time for the V_{DD} supply to get above V_{LVR}.

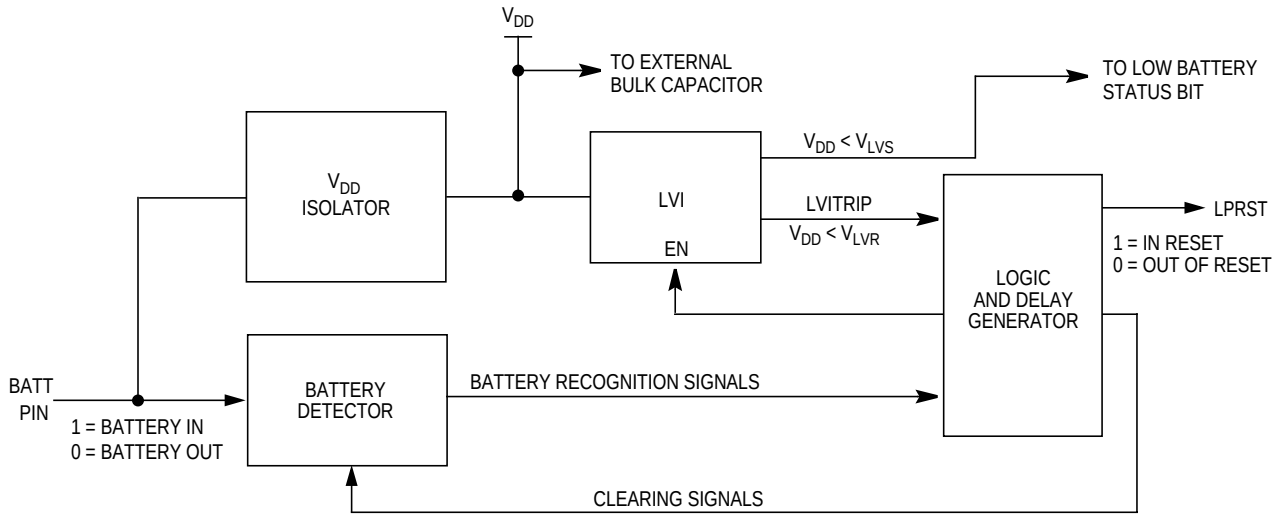


Figure 7-2. Internal Low-Battery Detection Block Diagram

After the 262,144 CGMXCLK cycle delay, a 512 CGMXCLK cycle delay occurs while the LVI is enabled and samples V_{DD}. The LVI is enabled at this time to sample V_{DD} to check if weak batteries were reinstalled. If V_{DD} is detected below $V_{LVR,}$ the LVI will cause the MCU to re-enter the low-power reset mode before the CPU begins operation. The low-power reset mode cannot be exited until batteries have been removed and reinserted.

The low-power reset condition is cleared after the 512 CGMXCLK cycle delay and a system stabilization delay of 4096 CGMXCLK cycles occurs. The complete power-on reset recovery sequence is detailed in **6.3.2.3 Power-On Reset (POR) Recovery**.

7.5.3 RAM Retention Determination

The external bulk capacitor is the power storage device used for RAM retention when the battery supply has been removed. The decay rate of the capacitor and the V_{DD} level at the time low-power reset mode is entered will determine actual retention time. The low-power mode is entered when the LVI forces entry into low-power reset mode or when batteries have been removed.

The LVI forced low-power reset provides the minimum data retention time. The retention time calculation is derived from:

$$i = C\frac{dV}{dt} \Rightarrow t_{RDR} = C\frac{\Delta V}{i}$$

where:

C = 100 µF

i = Typical I_{DD} current when battery removed @ 25°C = 100 nA

ΔV = Voltage difference of where the LVI forces entry into low-power reset mode ($V_{DD} = V_{LVR}$) to the guaranteed RAM retention voltage

Assuming a run current less than 10 mA, the voltage drop from the time the LVI senses the low voltage to the time the LVI forces entry into low-power reset mode is negligible (< 1 mV).

Example calculation:

$$t_{RDR} = 100\ \mu F\ \frac{1.8\ V - 1.3\ V}{0.1\ \mu A} = 500\ \text{seconds}$$

7.6 Central Processor Unit (CPU)

7.6.1 Wait Mode

The WAIT instruction:

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

7.6.2 Stop Mode

The STOP instruction:

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

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

7.7 Break Module (BRK)

7.7.1 Wait Mode

If enabled, the break module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if the BW bit in the break status register is set.

7.7.2 Stop Mode

The break module is inactive in stop mode. A break interrupt causes exit from stop mode and sets the BW bit in the break status register.

7.8 Computer Operating Properly Module (COP)

7.8.1 Wait Mode

The COP remains active in wait mode. To prevent a COP reset during wait mode, periodically clear the COP counter in a CPU interrupt routine.

7.8.2 Stop Mode

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

The STOP bit in the configuration register (CONFIG) enables the STOP instruction. To prevent inadvertently turning off the COP with a STOP instruction, disable the STOP instruction by clearing the STOP bit.

7.9 External Interrupt Module (IRQ)

7.9.1 Wait Mode

The IRQ module remains active in wait mode. Clearing the IMASK1 bit in the IRQ status and control register enables $\overline{\text{IRQ1}}$ CPU interrupt requests to bring the MCU out of wait mode.

7.9.2 Stop Mode

The IRQ module remains active in stop mode. Clearing the IMASK1 bit in the IRQ status and control register enables $\overline{\text{IRQ1}}$ CPU interrupt requests to bring the MCU out of stop mode.

7.10 Keyboard Interrupt Module (KBI)

7.10.1 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.

7.10.2 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.

7.11 Low-Voltage Inhibit Module (LVI)

7.11.1 Wait Mode

The LVI module remains active in wait mode. The LVI module can force entry into the low-power reset mode if a V_{DD} voltage below the V_{LVR} threshold is detected.

7.11.2 Stop Mode

The LVI module is disabled in stop mode. The LVI module cannot generate a reset and bring the MCU out of stop mode.

7.12 Carrier Modulator Transmitter (CMT)

7.12.1 Wait Mode

The CMT remains active in wait mode. Any enabled CPU interrupt request from the CMT can bring the MCU out of wait mode.

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

7.12.2 Stop Mode

The CMT is inactive in stop mode. The STOP instruction does not affect register states or the state of the CMT counter. CMT operation resumes when the MCU exits stop mode after an external interrupt.

7.13 Timer Interface Module (TIM0I)

7.13.1 Wait Mode

The TIM0I remains active in wait mode. Any enabled CPU interrupt request from the TIM0I can bring the MCU out of wait mode.

If TIM0I functions are not required during wait mode, reduce power consumption by stopping the TIM0I before executing the WAIT instruction.

7.13.2 Stop Mode

The TIM0I is inactive in stop mode. The STOP instruction does not affect register states or the state of the TIM0I counter. TIM0I operation resumes when the MCU exits stop mode after an external interrupt.

7.14 Exiting Wait Mode

These events restart the CPU clock and load the program counter with the reset vector or with an interrupt vector:

- External reset — A logic 0 on the $\overline{\text{RST}}$ pin resets the MCU and loads the program counter with the contents of locations $FFFE and $FFFF.
- External interrupt — A high-to-low transition on an external interrupt pin ($\overline{\text{IRQ1}}$ pin) loads the program counter with the contents of locations $FFFA and $FFFB.
- Break interrupt — A break interrupt loads the program counter with the contents of $FFFC and $FFFD.
- Computer operating properly module (COP) reset — A timeout of the COP counter resets the MCU and loads the program counter with the contents of $FFFE and $FFFF.
- Low-voltage inhibit module (LVI) reset — A V_{DD} voltage below the V_{LVR} voltage forces entry into the low-power reset mode.

- Carrier modulator transmitter module (CMT) interrupt — A CPU interrupt request from the CMT module (end of cycle detect) loads the program counter with the contents of $FFF8 and $FFF9.
- Timer interface module (TIM0I) interrupt — A CPU interrupt request from the TIM0I (TIM0I overflow) loads the program counter with the contents of $FFF6 and $FFE7.
- Keyboard module (KBI) interrupt — A CPU interrupt request from the KBI module loads the program counter with the contents of $FFF4 and $FFF5.

7.15 Exiting Stop Mode

These events restart the system clocks and load the program counter with the reset vector or with an interrupt vector:

- External reset — A logic 0 on the $\overline{\text{RST}}$ pin resets the MCU and loads the program counter with the contents of locations $FFFE and $FFFF.
- External interrupt — A high-to-low transition on an external interrupt pin loads the program counter with the contents of locations:
  - $FFFA and $FFFB ($\overline{\text{IRQ1}}$ pin)
  - $FFF4 and $FFF5 (keyboard interrupt pins)
- Break interrupt — A break interrupt loads the program counter with the contents of locations $FFFC and $FFFD.

Upon exit from stop mode, the system clocks begin running after the system stabilization delay. A 12-bit stop recovery counter inhibits the system clocks for 4096 CGMXCLK cycles after the reset or external interrupt.

During this stabilization period, the LVI is enabled and a low V_{DD} will force entry into the low-power reset mode.

The short stop recovery bit, SSREC, in the configuration register controls the system stabilization delay during stop recovery. When the

configuration is selected, stop recovery time is reduced from 4096 CGMXCLK cycles to 32 CGMXCLK cycles.

NOTE: *Use the full stop recovery time (SSREC = 0) in applications that use an external crystal.*

Section 8. Low-Voltage Inhibit (LVI)

8.1 Contents

8.2 Introduction

The low-voltage inhibit (LVI) module monitors the voltage on the V_{DD} pin and will set a low-voltage sense bit when V_{DD} voltage falls to the LVI sense voltage. The LVI will force a low-power reset when the V_{DD} voltage falls to the LVI trip voltage.

8.3 Features

Features of the LVI module include:

- Detects two levels of low battery condition
- Forces entry into low-power reset mode
- Disabled during stop mode
- Disabled during low-power reset mode

8.4 Functional Description

Figure 8-1 shows the structure of the LVI module, which contains a bandgap reference circuit and two comparators. The LVI monitors V_{DD} voltage during normal MCU operation and is disabled when in stop mode or low-power reset mode. When enabled, the LVI module forces entry into low-power reset mode when V_{DD} falls below the V_{LVR} threshold.

Once a low-power reset occurs, the MCU remains in low-power reset mode until the BATT pin sees a rising edge (batteries are removed and then reinstalled). That is, the LVI can force MCU entry into the low-power reset mode, but it is dependent on other chip circuits to bring the MCU out of a low-power reset. A low-power reset also drives the RST pin low to provide low-voltage protection to external peripheral devices.
See **7.5 Low-Power Reset Mode**.

In addition to forcing a reset condition, the LVI module has a second circuit dedicated to detecting low batteries. When V_{DD} falls below V_{LVS}, the output of the weak battery comparator asserts the LOWV flag in the LVI status register (LVISR). In applications that require detecting low batteries, software can monitor by polling the LOWV bit.

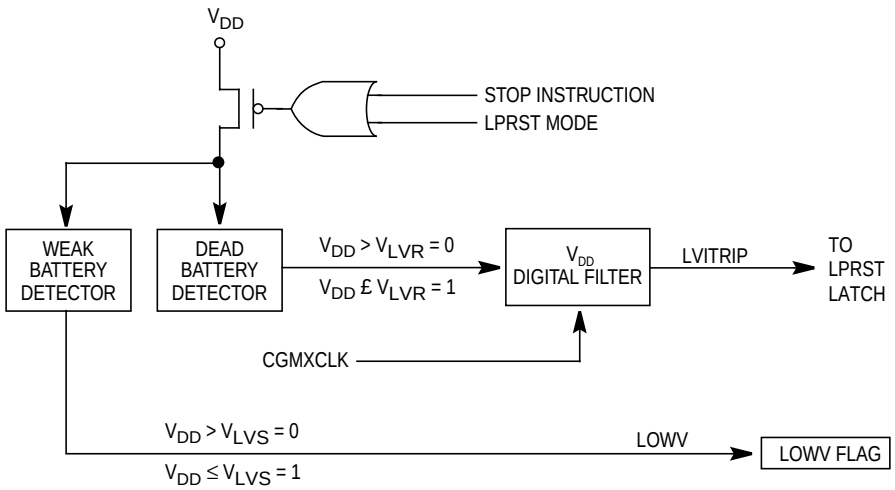


Figure 8-1. LVI Module Block Diagram

8.4.1 False Trip Protection

The V_{DD} pin level is digitally filtered to reduce false dead battery detection due to power supply noise. For the LVI module to reset due to a low-power supply, V_{DD} must remain at or below the V_{LVR} level for a minimum 32 to 40 CGMXCLK cycles. See **Table 8-1**.

Table 8-1. LVI Digital Filter Characteristics

V_{DD}		Result
At Level:	**For Number of CGMXCLK Cycles:**	
$V_{DD} > V_{LVR}$	Any	Filter counter remains clear
$V_{DD} < V_{LVR}$	< 32 CGMXCLK cycles	No reset, continue counting CGMXCLK
$V_{DD} < V_{LVR}$	Between 32 and 40 CGMXCLK cycles	LVI may force entry into LPRST after 32 CGMXCLK
$V_{DD} < V_{LVR}$	> 40 CGMXCLK cycles	LVI guaranteed to force entry into LPRST mode

8.4.2 Short Stop Recovery Option

The LVI has an enable time of t_{EN}. The system stabilization time for power on reset and long stop recovery (both 4096 CGMXCLK 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 mask option, the 32 CGMXCLK delay must be greater than the LVI turn on time to avoid a period in startup where the LVI is not protecting the MCU.

NOTE: *Use the full stop recovery time (SSREC = 0) in applications that use an external crystal.*

8.5 LVI Status Register

The LVI status register flags V_{DD} voltages below the V_{LVS} level.

Address: $001D

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

= Unimplemented R = Reserved

Figure 8-2. LVI Status Register (LVISR)

LOWV— LVI Low Indicator Bit

This read-only flag becomes set when the LVI is detecting V_{DD} voltage below the V_{LVS} threshold.

8.6 LVI Interrupts

The LVI module does not generate CPU interrupt requests.

8.7 Low-Power Modes

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

8.7.1 Wait Mode

The LVI module remains active in wait mode. The LVI module can force entry into the low-power reset mode if a V_{DD} voltage below the V_{LVR} voltage is detected.

8.7.2 Stop Mode

The LVI module is disabled in stop mode. The LVI module cannot generate a reset and bring the MCU out of stop mode.

Low-Voltage Inhibit (LVI)

Section 9. Oscillator

9.1 Contents

9.2 Introduction

The oscillator circuit is designed for use with crystals or ceramic resonators. The oscillator circuit generates the crystal clock signal, CGMXCLK, at the frequency of the crystal. This signal is divided by two before being passed on to the system integration module (SIM) for bus clock generation.

Figure 9-1 shows the structure of the oscillator. The oscillator requires various external components.

9.3 Oscillator External Connections

In its typical configuration, the oscillator requires five external components. The crystal oscillator is normally connected in a Pierce oscillator configuration, as shown in **Figure 9-1**. This figure shows only the logical representation of the internal components and may not represent actual circuitry. The oscillator configuration uses five components:

1. Crystal, X_1
2. Fixed capacitor, C_1
3. Tuning capacitor, C_2, which can also be a fixed capacitor
4. Feedback resistor, R_B
5. Optional series resistor, R_S

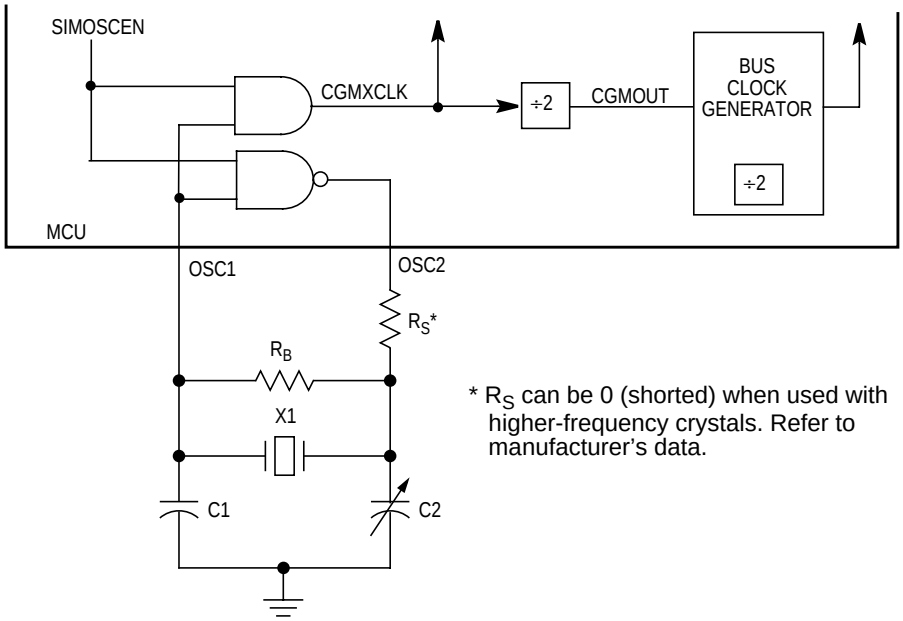


Figure 9-1. Oscillator External Connections

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

9.4 I/O Signals

This section describes the oscillator input/output (I/O) signals.

9.4.1 Crystal Amplifier Input Pin (OSC1)

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

9.4.2 Crystal Amplifier Output Pin (OSC2)

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

9.4.3 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal enables the oscillator.

9.4.4 External Clock Source (CGMXCLK)

CGMXCLK is the crystal oscillator output signal. It runs at the full speed of the crystal (f_{XCLK}) and comes directly from the crystal oscillator circuit. **Figure 9-1** shows only the logical relation of CGMXCLK to OSC1 and OSC2 and may not represent the actual circuitry. The duty cycle of CGMXCLK is unknown and may depend on the crystal and other external factors. Also, the frequency and amplitude of CGMXCLK can be unstable at startup.

9.4.5 Oscillator Out (CGMOUT)

CGMOUT is the clock output of the OSC circuits. This signal is used for generation of the MCU bus clocks. CGMOUT is a 50 percent duty cycle clock running at twice the bus frequency.

9.4.6 Bus Clocks

Several bus clocks that are derived from the CGMOUT signal are one fourth the CGMXCLK frequency.

9.5 Low-Power Modes

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

9.5.1 Wait Mode

The WAIT instruction has no effect on the oscillator logic. CGMXCLK and CGMOUT clocks continue to be driven out.

9.5.2 Stop Mode

The STOP instruction disables the CGMXCLK and CGMOUT outputs.

9.6 Oscillator During Break Mode

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

Section 10. Configuration Register (CONFIG)

10.1 Contents

10.2 Introduction

This section describes the configuration register (CONFIG). The configuration register enables or disables these options:

- Stop mode recovery time (32 CGMXCLK cycles or 4096 CGMXCLK cycles)
- COP timeout period ($2^{18} – 2^4$ or $2^{13} – 2^4$ CGMXCLK cycles)
- STOP instruction
- Computer operating properly module (COP)

10.3 Functional Description

The configuration register is used in the initialization of various options. The configuration register can be written once after each reset. All of the configuration register bits are cleared during reset. Since the various options affect the operation of the MCU, it is recommended that this register be written immediately after reset. The configuration register is located at $001F. For compatibility, a write to a ROM version of the MCU at this location will have no effect. The configuration register may be read at anytime.

NOTE: *The CONFIG module is known as an MOR (mask option register) on a ROM device.*

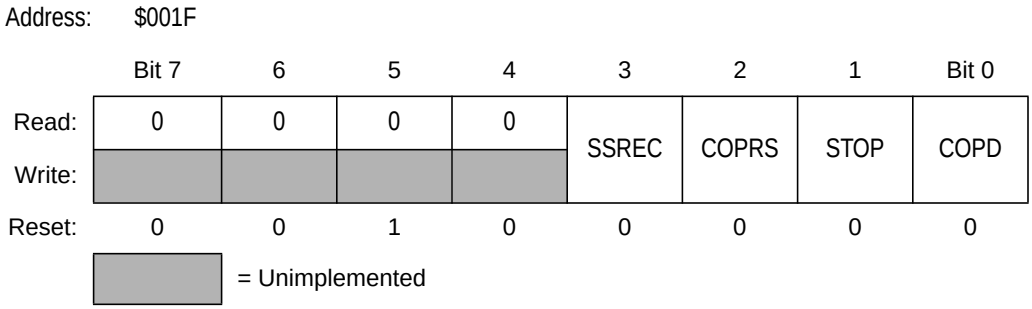


Figure 10-1. Configuration Register (CONFIG)

SSREC — Short Stop Recovery Bit

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

1 = Stop mode recovery after 32 CGMXCLK cycles
0 = Stop mode recovery after 4096 CGMXCLKC cycles

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

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

The LVI has an enable time of t_{EN}. The standard system stabilization time for power on reset and long stop recovery (both 4096 CGMXCLK 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 CGMXCLK 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.

COPRS — COP Rate Select Bit

COPRS selects the COP timeout period. Reset clears COPRS. (See **Section 16. Computer Operating Properly (COP)**.)

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

STOP — STOP Instruction Enable Bit

STOP enables the STOP instruction.

1 = STOP instruction enabled
0 = STOP instruction treated as illegal opcode

COPD — COP Disable Bit

COPD disables the COP module. (See **Section 16. Computer Operating Properly (COP)**.)

1 = COP module disabled
0 = COP module enabled

Configuration Register (CONFIG)

Section 11. Input/Output Ports (I/O)

11.1 Contents

11.2 Introduction

Twenty bidirectional input/output (I/O) pins form three parallel ports. All I/O pins are programmable as inputs or outputs.

NOTE: *Connect any unused I/O pins to an appropriate logic level, either V_{DD} or V_{SS}. Although the I/O ports do not require termination for proper operation, termination reduces excess current consumption and the possibility of electrostatic damage.*

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA) See page 111.	Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PTB) See page 113.	Read: Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PTC) See page 116.	Read:	0	0	0	0	PTC3	PTC2	PTC1	PTC0
		Write:								
		Reset:	Unaffected by reset							
Note: PTC7—PTC4 are not available.										
$0004	Data Direction Register A (DDRA) See page 111.	Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB) See page 114.	Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Reset:	0	0	0	0	0	0	0	0
$0006	Data Direction Register C (DDRC) See page 117.	Read:	1	1	1	1	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
Note: DDRC7–DDRC4 are not available. Set DDRC7–DDRC4 to 1 on 28-pin packages.										

= Unimplemented

Figure 11-1. I/O Port Register Summary

11.3 Port A

Port A is an 8-bit, general-purpose, bidirectional I/O port.

11.3.1 Port A Data Register

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

Address: $0000

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Reset:	Unaffected by reset							

Figure 11-2. Port A Data Register (PTA)

PTA7–PTA0 — Port A Data Bits

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

11.3.2 Data Direction Register A

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

Address: $0004

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset:	0	0	0	0	0	0	0	0

Figure 11-3. Data Direction Register A (DDRA)

DDRA7–DDRA0 — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA7–DDRA0, configuring all port A pins as inputs.

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

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

Figure 11-4 shows the port A I/O logic.

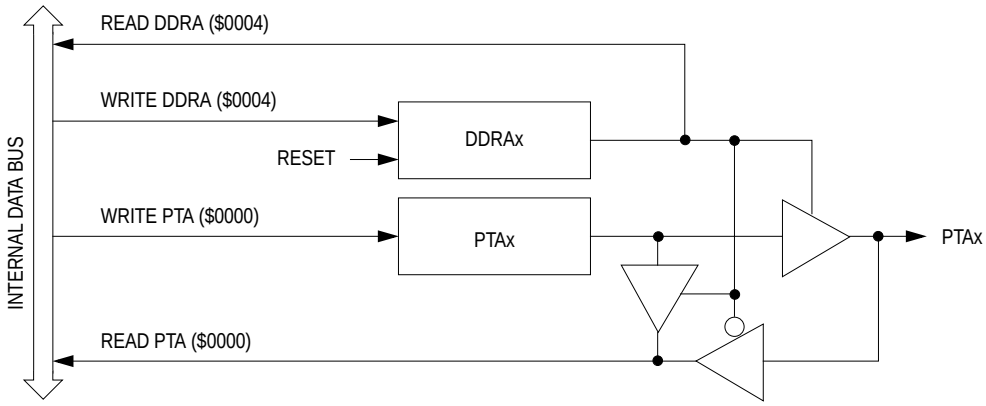


Figure 11-4. Port A I/O Circuit

When bit DDRAx is a logic 1, reading address $0000 reads the PTAx data latch. When bit DDRAx is a logic 0, reading address $0000 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. **Table 11-1** summarizes the operation of the port A pins.

Table 11-1. Port A Pin Functions

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

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

11.4 Port B

Port B is an 8-bit, general-purpose, bidirectional I/O port that shares its pins with the keyboard interrupt module (KBI).

11.4.1 Port B Data Register

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

Address: $0001

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Reset:	Unaffected by reset							

Figure 11-5. Port B Data Register (PTB)

PTB7–PTB0 — Port B Data Bits

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

The keyboard interrupt enable bits, KBIE7–KBIE0, in the keyboard interrupt control register (KBICR), enable the port B pins as external interrupt pins. (See **Section 15. Keyboard Interrupt Module (KBI)**.)

11.4.2 Data Direction Register B

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

Address: $0005

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Reset:	0	0	0	0	0	0	0	0

Figure 11-6. Data Direction Register B (DDRB)

DDRB7–DDRB0 — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB7–DDRB0, configuring all port B pins as inputs.

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

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

Figure 11-7 shows the port B I/O logic.

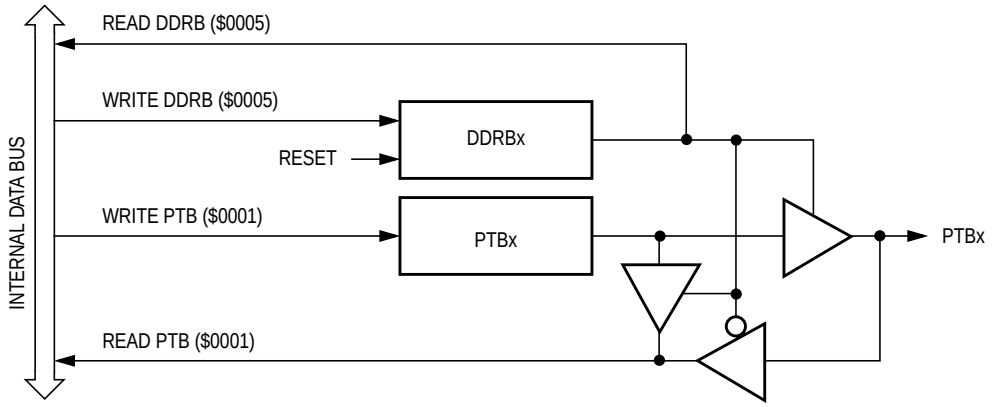


Figure 11-7. Port B I/O Circuit

When bit DDRBx is a logic 1, reading address $0001 reads the PTBx data latch. When bit DDRBx is a logic 0, reading address $0001 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. **Table 11-2** summarizes the operation of the port B pins.

Table 11-2. Port B Pin Functions

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

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

11.5 Port C

Port C is an 8-bit, general-purpose, bidirectional I/O port. PTC0–PTC3 have higher than standard current drive capabilities. Refer to **Section 19. Preliminary Electrical Specifications** for the port C drive specifications.

11.5.1 Port C Data Register

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

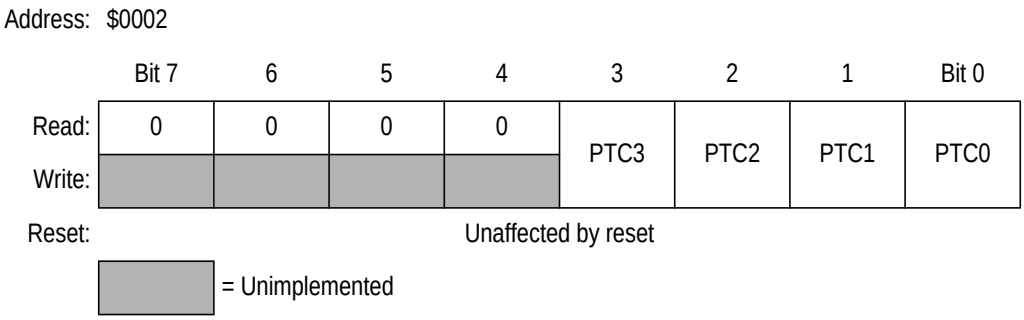


Figure 11-8. Port C Data Register (PTC)

PTC7–PTC4 — Port C Data Bits

PTC7–PTC4 pads are not bonded out. Set these ports to outputs by writing an $FX to data direction register C and write $00 to the port C data register after any reset to avoid floating inputs and erroneous data.

PTC[3:0] — Port C Data Bits

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

11.5.2 Data Direction Register C

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

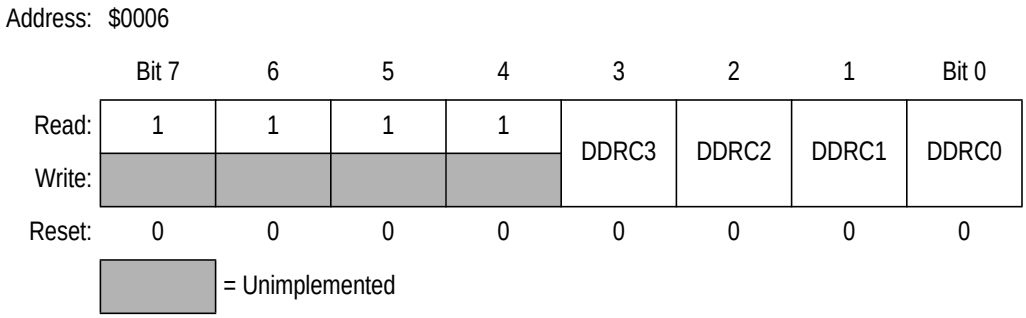


Figure 11-9. Data Direction Register C (DDRC)

DDRC7–DDRC4 — Data Direction Register C Bits

DDRC7–DDRC4 pads are not bonded out. Set these ports to outputs by writing an $FX to data direction register C and write $00 to the port C data register after any reset to avoid floating inputs and erroneous data.

DDRC3–DDRC0 — Data Direction Register C Bits

These read/write bits control port C data direction. Reset clears DDRC3–DDRC0, configuring all port C pins as inputs.

1 = Corresponding port C pin configured as output
0 = Corresponding port C pin configured as input

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

Figure 11-10 shows the port C I/O logic.

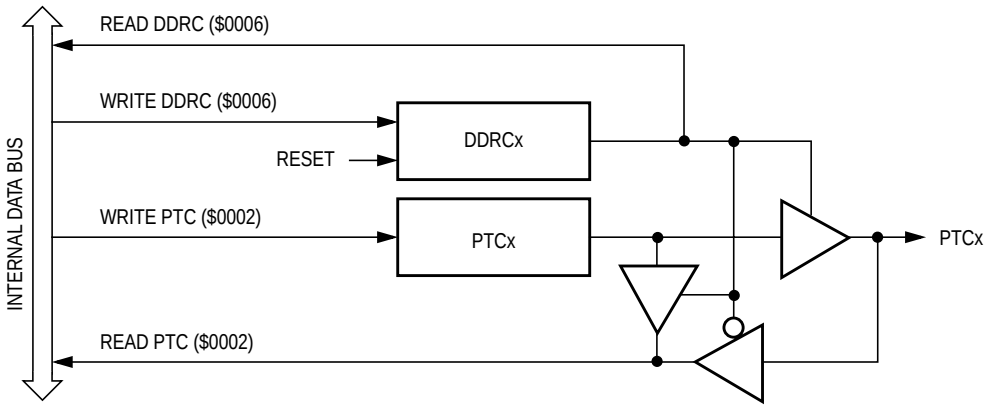


Figure 11-10. Port C I/O Circuit

When bit DDRCx is a logic 1, reading address $0002 reads the PTCx data latch. When bit DDRCx is a logic 0, reading address $0002 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. **Table 11-3** summarizes the operation of the port C pins.

Table 11-3. Port C Pin Functions

DDRC Bit	PTC Bit	I/O Pin Mode	Accesses to DDRC	Accesses to PTC	
			Read/Write	Read	Write
0	X(1)	Input, Hi-Z(2)	DDRC3–DDRC0	Pin	PTC3–PTC0(3)
1	X	Output	DDRC3–DDRC0	PTC3–PTC0	PTC3–PTC0

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

Section 12. Carrier Modulator Transmitter (CMT)

12.1 Contents

12.2 Introduction

The carrier modulator transmitter (CMT) module provides a means to generate the protocol timing and carrier signals for a wide variety of encoding schemes. The CMT incorporates hardware to off-load the critical and/or lengthy timing requirements associated with code generation from the central processor unit (CPU), releasing much of its bandwidth to handle other tasks such as code data generation, data decompression, or keyboard scanning.

The CMT does not include dedicated hardware configurations for specific protocols but is intended to be sufficiently programmable in its function to handle the timing requirements of most protocols with minimal CPU intervention. When the modulator is disabled, certain CMT registers can be used to change the state of the infrared out pin (IRO) directly. This feature allows for the generation of future protocols not readily producible by the current architecture.

12.3 Clock Generation

The CMT module uses the CGMOUT signal from the oscillator circuits as a clock. With an 8-MHz crystal, CGMOUT is a 4-MHz clock. The DIV2 bit in the CMT modulator control and status register (CMCS) provides an option to divide the CMT clock source by another divide-by-2 circuit, thus extending the CMT periods. The block diagram for CMT clock circuits is shown in **Figure 12-1**.

Table 12-1 shows the relationship between typical system clock values and the CMT operating speeds.

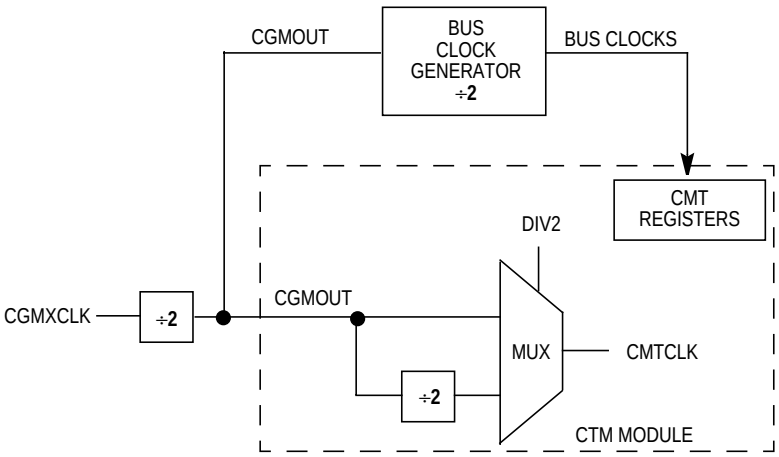


Figure 12-1. CMT Clock Generation

Table 12-1. System Clock Examples

Crystal (MHz)	Internal Bus (MHz)	CGMOUT (MHz) (Feed to CMT)	DIV2 Bit	Carrier Generator Resolution (µs)	Min Carrier Generator Period (µs)	Min Modulator Period (µs)
8	2	4	0	0.25	0.5	2
8	2	4	1	0.5	1	4
4	1	2	0	0.5	1	4
4	1	2	1	1	2	8

NOTE: *Period and frequency calculations shown in **Table 12-1** will be affected by the state of the DIV2 bit.*

When DIV2 is set:

$$CMTCLK = \frac{CGMOUT}{2} = \frac{CGMXCLK}{4}$$

When DIV2 is clear:

$$CMTCLK = CGMOUT = \frac{CGMXCLK}{2}$$

12.4 Overview

The module consists of:

- Carrier generator
- Modulator
- Transmitter output
- Registers

The block diagram is shown in **Figure 12-2**. The module has three main modes of operation: time, baseband, and frequency shift key (FSK).

When operating in time mode, the user independently defines the high and low times of the carrier signal to determine both period and duty cycle. The carrier generator resolution is 250 ns when operating with an 8-MHz crystal and the DIV2 bit in the CMCS clear. The carrier generator can generate signals with periods between 500 ns (2 MHz) and 31.5 µs (31.75 kHz) in steps of 250 ns. The possible duty cycle options will depend upon the number of counts required to complete the carrier period.

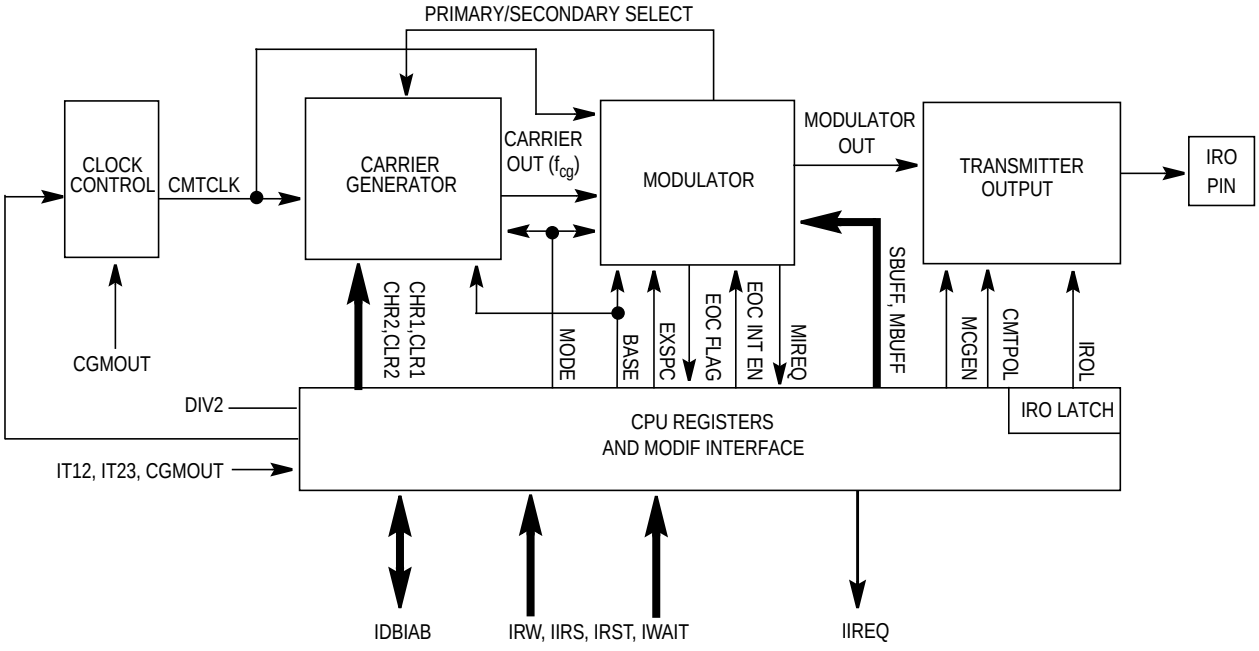


Figure 12-2. Carrier Modulator Transmitter Module Block Diagram

For example, an 800-kHz signal has a period of 1.25 µs and will therefore require 5 x 250 ns counts to generate. These counts may be split between high and low times so the duty cycles available will be:

- 20 percent (one high, four low)
- 40 percent (two high, three low)
- 60 percent (three high, two low)
- 80 percent (four high, one low)

For lower frequency signals with larger periods, higher resolution (as a percentage of the total period) duty cycles are possible.

When the BASE bit in the CMT modulator control and status register (CMCS) is set, the carrier output to the modulator is held high continuously to allow for the generation of baseband protocols.

A third mode allows the carrier generator to alternate between two sets of high and low times. When operating in FSK mode, the generator will toggle between the two sets when instructed by the modulator, allowing the user to dynamically switch between two carrier frequencies without CPU intervention.

The modulator provides a simple method to control protocol timing. The modulator has a resolution of 2 µs with an 8-MHz oscillator. It can count system clocks (to provide real-time control) or it can count carrier clocks (for self-clocked protocols). See **12.6 Modulator** for more details.

The transmitter output block controls the state of the infrared out pin (IRO). The modulator output is gated on to the IRO pin when the modulator/carrier generator is enabled. When the module is not enabled, the IRO pin is controlled by the state of the IRO latch. A polarity bit enables the IRO pin to be high true or low true. See **12.8 IRO Latch**.

A summary of the possible modes is shown in **Table 12-2**.

Table 12-2. CMT Modes of Operation

Mode	MCGEN Bit[1]	BASE Bit[2]	MODE Bit[2]	EXSPC Bit	Comment
Time	1	0	0	0	f_{cg} controlled by primary high and low registers. f_{cg} transmitted to IRO pin when modulator gate is open.
Baseband	1	1	0	0	f_{cg} is always high. IRO pin high when modulator gate is open.
FSK	1	0	1	0	f_{cg} control alternates between primary high/low registers and secondary high/low registers. f_{cg} transmitted to IRO pin when modulator gate is open.
Extended space	1	x	x	1	Setting the EXSPC bit causes subsequent modulator cycles to be spaces (modulator out not asserted) for the duration of the modulator period (mark and space times).
IRO latch	0	x	x	x	IRO latch controls state of IRO pin. The IRO latch can be written to on positive or negative edge of internal bus clock.

1. To prevent spurious operation, initialize all data and control registers before beginning a transmission (MCGEN = 1).
2. These bits are not double buffered and should not be changed during a transmission (while MCGEN = 1).

12.5 Carrier Generator

The carrier signal is generated by counting a register-selected number of input clocks (250 ns for an 8-MHz oscillator) for both the carrier high time and the carrier low time. The period is determined by the total number of clocks counted. The duty cycle is determined by the ratio of high time clocks to total clocks counted. The high and low time values are user programmable and are held in two registers.

An alternate set of high/low count values is held in another set of registers to allow the generation of dual frequency FSK (frequency shift keying) protocols without CPU intervention.

NOTE: *Data values for the high and low times should be non-zero to prevent spurious operation.*

The MCGEN bit in the CMCS must be set and the BASE bit in the CMCS must be cleared to enable carrier generator clocks. When the BASE bit in the CMT modulator control and status register (CMCS) is set, the carrier output to the modulator is held high continuously. The block diagram is shown in **Figure 12-3**.

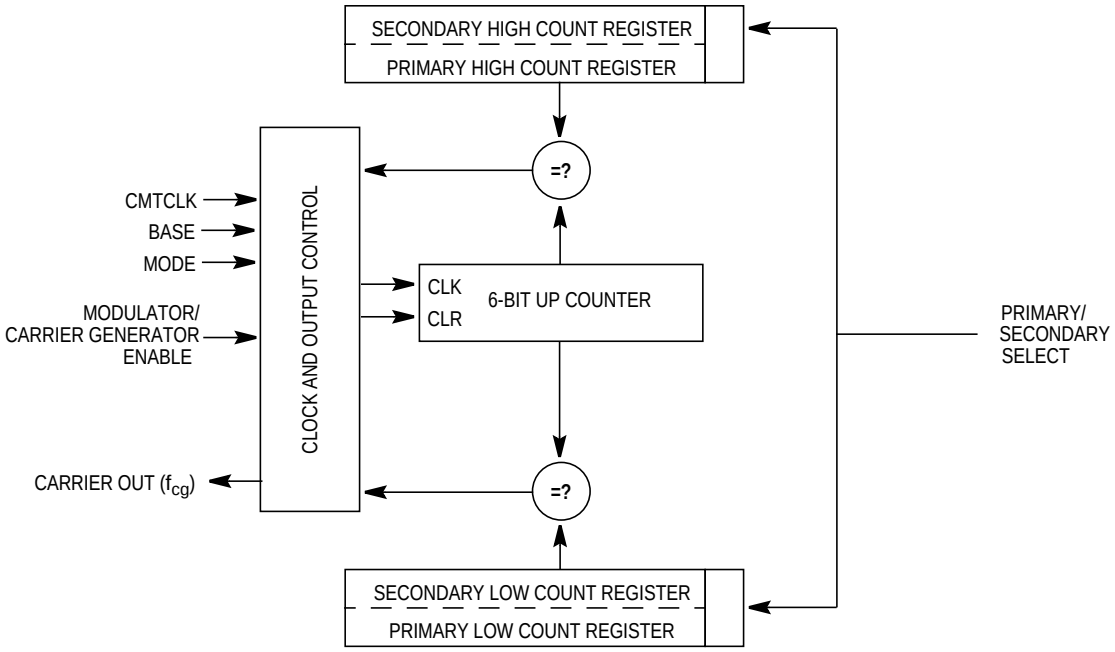


Figure 12-3. Carrier Generator Block Diagram

12.5.1 Time Counter

The high/low time counter is a 6-bit up counter. After each increment, the contents of the counter are compared with the appropriate high or low count value register. When the compare value is reached, the counter is reset (to a value of 1) and the compare is redirected to the other count value register.

Assuming that the high time count compare register is currently active, a valid compare will cause the carrier output to be driven low. The counter will continue to increment (starting at reset value of 1). When the value stored in the selected low count value register is reached, the counter will again be reset and the carrier output will be driven high.

The cycle repeats, automatically generating a periodic signal which is directed to the modulator. The lowest frequency (maximum period) and highest frequency (minimum period) which can be generated are:

f_{max} = CMTCLK ÷ (2 x 1) Hz

f_{min} = CMTCLK ÷ (2 x (2^6 – 1)) Hz

In the general case, the carrier generator output frequency is:

f_{cg} = CMTCLK ÷ (Highcount + Lowcount) Hz

Where:

0 < *Highcount* < 64
0 < *Lowcount* < 64

The duty cycle of the carrier signal is controlled by varying the ratio of high time to low + high time. As the input clock period is fixed, the duty cycle resolution will be proportional to the number of counts required to generate the desired carrier period.

$$\text{Duty Cycle} = \frac{\text{Highcount}}{\text{Highcount} + \text{Lowcount}}$$

12.6 Modulator

The modulator has three main modes of operation:

1. The modulator can gate the carrier onto the modulator output (TIME).
2. The modulator can control the logic level of the modulator output (BASEBAND).
3. The modulator can count carrier periods and instruct the carrier generator to alternate between two carrier frequencies whenever a modulation period (mark + space counts) expires (FSK)

The modulator includes a 13-bit down counter with underflow detection. The counter is loaded from the 12-bit modulation mark period buffer register, MBUFF. The most significant bit is loaded with a 0 and serves as a sign bit. When the counter holds a positive value, the modulator gate is open and the carrier signal is driven to the transmitter block.

When the counter underflows, the modulator gate is closed and a 12-bit comparator is enabled which compares the logical complement of the contents of the decrementing counter with the contents of the modulation space period register, SREG.

When a match is obtained, the cycle repeats by opening the modulator gate, reloading the counter with the contents of MBUFF and reloading SREG with the contents of SBUFF.

Should SREG = 0, the match will be immediate and no space period will be generated (for instance, for FSK protocols which require successive bursts of different frequencies).

The MCGEN bit in the CMCS must be set to enable the modulator timer. The 12-bit MBUFF and SBUFF registers are accessed through three 8-bit modulator period registers, CMD1, CMD2, and CMD3.

Refer to **Figure 12-4** for a block diagram of the modulator.

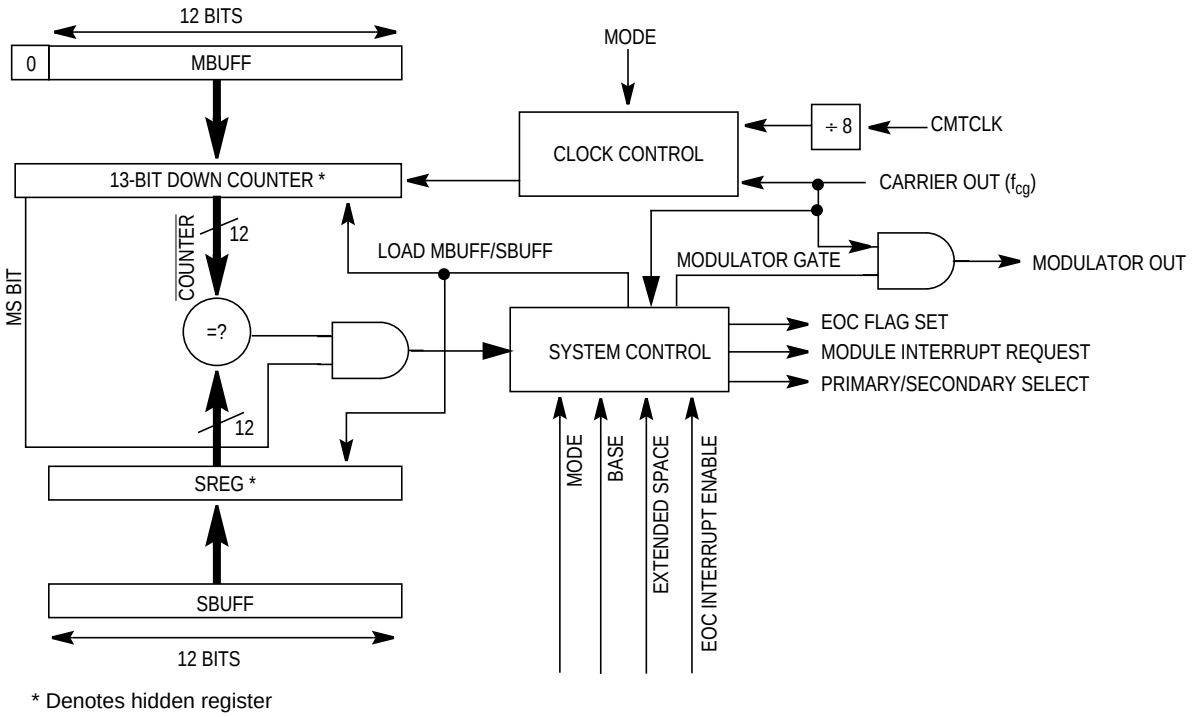


Figure 12-4. Modulator Block Diagram

12.6.1 Time Mode

When the modulator operates in time mode (BASE bit is clear, MODE bit is clear), the modulation mark period consists of an integer number of CMTCLK ÷ 8 clocks. The modulation space period consists of 0 or an integer number of CMTCLK ÷ 8 clocks. The mark and space periods are controlled by the MBUFF and SBUFF registers, respectively. With a 2-MHz crystal and DIV2 = 0, the modulator resolution is 4 µs and has a maximum mark and space period of about 16 ms each. See **Figure 12-5** for an example of the time mode output.

The mark and space time equations for time and baseband mode are:

$$t_{mark} = \frac{(MBUFF + 1) \times 8}{CMTCLK} secs$$

$$t_{space} = \frac{SBUFF \times 8}{CMTCLK} secs$$

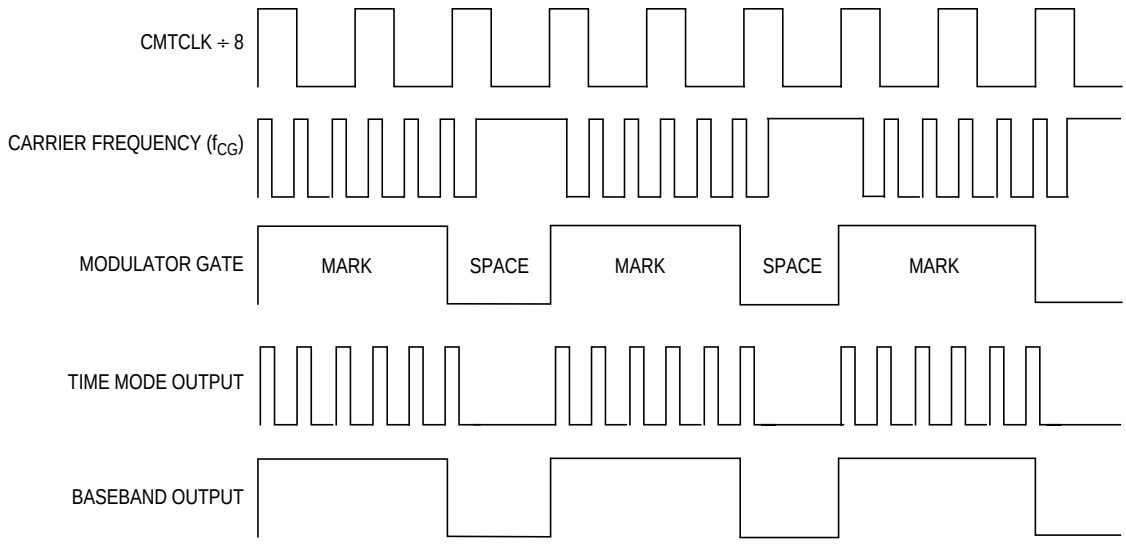


Figure 12-5. CMT Operation in Time Mode

12.6.1.1 Synchronization of Modulator and Carrier Generator in Time Mode

To prevent carrier glitches which could affect carrier spectral purity, the modulator control gate and carrier clock are synchronized. The carrier signal is activated when the modulator gate opens. The modulator gate can close only when the carrier signal is low (the output logic level during space periods is low).

When the modulator gate closes, the carrier is reset to a 1 on the next rising edge of CMTCLK. The carrier counter is held in the reset state until the beginning of the next mark period.

In some special cases, where the carrier is high for the duration of the space period, the modulator gate does not close and the carrier continues to be gated to the IRO output without interruption into the next mark period. These special cases can occur when modulation space periods are shorter than carrier high times.

12.6.2 Baseband Mode

Baseband mode (BASE bit is set, MODE bit is clear) is a derivative of time mode, where the mark and space period is based on (CMTCLK ÷ 8) counts. The mark and space calculations are the same as in time mode. In this mode the modulator output will be at a logic 1 for the duration of the mark period and at a logic 0 for the duration of a space period. See **Figure 12-5**.

12.6.3 FSK Mode

When the modulator operates in FSK mode, the modulation mark and space periods consist of an integer number of carrier clocks (space period can be 0). When the mark period expires, the space period is transparently started (as in time mode). The carrier generator toggles between primary and secondary data register values whenever the modulator space period expires.

The space period provides an interpulse gap (no carrier). If SBUFF = 0, then the modulator and carrier generator will switch between carrier frequencies without a gap or any carrier glitches (0 space).

Using timing data for carrier burst and interpulse gap length calculated by the CPU, FSK mode can automatically generate a phase-coherent, dual-frequency FSK signal with programmable burst and interburst gaps.

The mark and space time equations for FSK mode are:

$$t_{mark} = \frac{MBUFF + 1}{f_{CG}} secs$$

$$t_{space} = \frac{SBUFF}{f_{CG}} secs$$

Where f_{CG} is the frequency output from the carrier generator.

12.6.4 Extended Space Operation

In either time, baseband, or FSK mode, the space period can be made longer than the maximum possible value of SBUFF. Setting the EXSPC bit in the CMCS will force the modulator to treat the next modulation period (beginning with the next load of MBUFF/SBUFF) as a space period equal in length to the mark and space counts combined. Subsequent modulation periods will consist entirely of these extended space periods with no mark periods. Clearing EXSPC will return the modulator to standard operation at the beginning of the next modulation period.

12.6.4.1 EXSPC Operation in Time Mode

To calculate the length of an extended space in time or baseband modes, use the equation:

$$t_{exspace} = \frac{((SBUFF_1)+(MBUFF_2+1+SBUFF_2) +... (MBUFF_n+1+SBUFF_n)) \times 8}{CMTCLK} \text{ secs}$$

Where the subscripts 1, 2, ... n refer to the modulation periods that elapsed while the EXSPC bit was set.

For an example of extended space operation, see **Figure 12-6**.

NOTE: *The EXSPC feature can be used to emulate a zero mark event.*

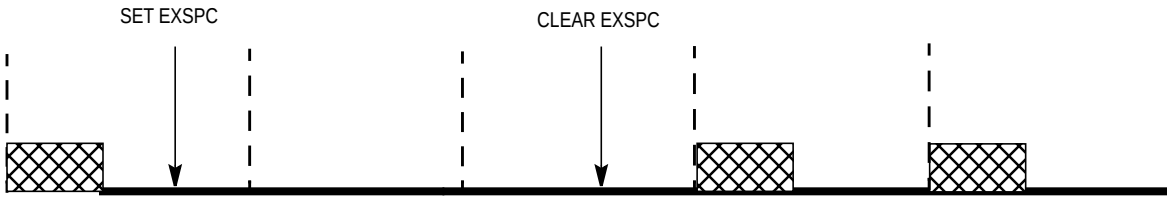


Figure 12-6. Extended Space Operation

12.6.4.2 EXSPC Operation in FSK Mode

In FSK mode, the modulator continues to count f_{CG} clocks, alternating between the primary and secondary registers at the end of each modulation period.

To calculate the length of an extended space in FSK mode, one needs to know whether the EXSPC bit was set on a primary or secondary modulation period. A status bit for the current modulation is not accessible to the CPU. If necessary, software should maintain tracking of the current modulation cycle (primary or secondary).

If the EXSPC bit was set during a primary modulation cycle, use the equation:

$$t_{EXSPACE} = \frac{SBUFF_1}{f_{CG1}} + \frac{MBUFF_2 + 1 + SBUFF_2}{f_{CG2}} + \frac{MBUFF_3 + 1 + SBUFF_3}{f_{CG1}} + ...secs$$

Where:

- The subscripts 1, 2, ... n refer to the modulation periods that elapsed while the EXSPC bit was set.
- f_{cg1} is the frequency output from the carrier generator for the primary registers.
- f_{cg2} is the frequency output from the carrier generator for the secondary registers.

If the EXSPC bit was set during a secondary modulation cycle, use the equation:

$$t_{EXSPACE} = \frac{SBUFF_1}{f_{CG2}} + \frac{MBUFF_2 + 1 + SBUFF_2}{f_{CG1}} + \frac{MBUFF_3 + 1 + SBUFF_3}{f_{CG2}} + ...secs$$

12.7 Transmitter

The transmitter output block controls the state of the infrared out pin (IRO). The modulator output is gated onto the IRO pin when the modulator/carrier generator is enabled. When the modulator/carrier generator is disabled, the IRO pin is controlled by the state of the IRO latch.

A polarity bit in the CCH1 register enables the IRO pin to be high true or low true.

12.8 IRO Latch

The IRO latch is accessible to the CPU through bit 7 of the carrier generator data registers CCH1 (IROLN bit) and CCL1 (IROLP bit). When the MCGEN bit is clear, the IRO latch is driven to the IRO pin.

Through the IROLN and IROLP bits, the IRO latch can be written on either edge of the internal bus clock, allowing for IR waveforms which have a resolution of twice the bus clock frequency (CGMOUT).

When IROLN is written, the IRO latch will be updated with the new data on the negative edge of the internal bus clock.

When IROLP is written, the IRO latch will be updated with the new data on the positive edge of the internal bus clock.

12.9 Flags and Interrupts

The end-of-cycle (EOCF) flag is set:

- When the modulator is not currently active and the MCGEN bit is set to begin the initial CMT transmission
- At the end of each modulation cycle (when the counter is reloaded from MBUFF) while the MCGEN bit is set

In the condition where the MCGEN bit is cleared and then set before the end of the modulation cycle, the EOC will not be set when the MCGEN is set, but will become set at the end of the current modulation cycle.

When the MCGEN becomes disabled, the CMT module does not set the EOC flag at the end of the last modulation cycle.

The EOCF bit is cleared by reading the CMT modulator control and status register (CMCS) followed by an access of CMD2 or CMD3.

If the EOC interrupt enable (EOCIE) bit was previously set, the CMT module will generate an interrupt request to the CPU. The EOCF bit must be cleared within the interrupt service routine (ISR) to prevent another interrupt being generated after exiting the ISR. If the EOC interrupt is not being used (EOCIE = 0), the EOCF flag need not be cleared.

The EOC interrupt is coincidental with reloading the down-counter with the contents of MBUFF and reloading the SREG with the contents of SBUFF. The EOC interrupt provides a means for the user to reload new mark/space values into the MBUFF and SBUFF registers. An MBUFF and/or SBUFF update will take effect at the end of the current modulation cycle.

NOTE: *The down-counter and SREG are updated at the end of every modulation cycle, irrespective of interrupt handling and the state of the EOCF flag.*

12.10 I/O Registers

These I/O registers control and monitor CMT operation:

- CMT carrier generator data registers (CCH1, CCL1, CCH2, CCL2)
- CMT modulator control and status register (CMCS)
- CMT modulator period data registers (CMD1, CMD2, CMD3)

12.10.1 Carrier Generator Data Registers (CCH1, CCL1, CCH2, and CCL2)

The carrier generator contains:

- One 8-bit data register — CMT primary high time, CCH1
- One 7-bit data register — CMT primary low time, CCL1
- Two 6-bit data registers:
  - CMT secondary high time, CCH2
  - CMT secondary low time, CCL2

Bit 7 of CCH1 and CCL1 is used to read and write the IRO latch.

Register Name and Address: CCH1—$0010

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	IROLN	CMTPOL	PH5	PH4	PH3	PH2	PH1	PH0
Reset:	0	0	Unaffected by reset					

Register Name and Address: CCL1—$0011

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IROLP	0	PL5	PL4	PL3	PL2	PL1	PL0
Write:								
Reset:	0	0	0	Unaffected by reset				

Figure 12-7. CMT Carrier Generator Data Register (CCH1, CCL1, CCH2, and CCL2)

Register Name and Address: CCH2—$0012

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	SH5	SH4	SH3	SH2	SH1	SH0
Write:								
Reset:	0	0	0	Unaffected by reset				

Register Name and Address: CCL2—$0013

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	SL5	SL4	SL3	SL2	SL1	SL0
Write:								
Reset:	0	0	0	Unaffected by reset				

= Unimplemented

Figure 12-7. CMT Carrier Generator Data Register (CCH1, CCL1, CCH2, and CCL2) (Continued)

PH0–PH5 and PL0–PL5 — Primary Carrier High and Low Time Data Values

When selected, these bits contain the number of input clocks required to generate the carrier high and low time periods. When operating in time mode (see **12.6.1 Time Mode**), this register pair is always selected. When operating in FSK mode (see **12.6.3 FSK Mode**), this register pair and the secondary register pair are alternately selected under control of the modulator. The primary carrier high and low time values are undefined out of reset. These bits must be written to non-zero values before the carrier generator is enabled to avoid spurious results.

NOTE: *Writing to CCH1 to update PH0–PH5 or to CCL1 to update PL0–PL5 will also update the IRO latch. When MCGEN (bit 0 in the CMCS) is clear, the IRO latch value appears on the IRO output pin. Care should be taken that bit 7 of the data to be written to CCH1 or CCL1 should contain the desired state of the IRO latch.*

Additionally, writing to CCH1 to update PH0–PH5 will also update the CMT polarity bit. Care should be taken that bit 6 of the data to be written to CCH1 should contain the desired state of the polarity bit.

SH0–SH5 and SL0–SL5 — Secondary Carrier High and Low Time Data Values

When selected, these bits contain the number of input clocks required to generate the carrier high and low time periods. When operating in time mode (see **12.6.1 Time Mode**), this register pair is never selected. When operating in FSK mode (see **12.6.3 FSK Mode**), this register pair and the primary register pair are alternately selected under control of the modulator. The secondary carrier high and low time values are undefined out of reset. These bits must be written to non-zero values before the carrier generator is enabled when operating in FSK mode.

CMTPOL — CMT Output Polarity Bit

This bit controls the polarity of the CMT output (IRO). When this bit is a 0, then the CMT output is active high. When this bit is set to 1 the CMT output is active low, in other words inverted. The reset state of this bit is 0.

IROLN and IROLP — IRO Latch Control Bit

Reading IROLN or IROLP reads the state of the IRO latch. Writing IROLN updates the IRO latch with the data being written on the negative edge of the internal processor clock (IT12). Writing IROLP updates the IRO latch on the positive edge of the internal processor clock; for example, one CGMOUT period later. The IRO latch is clear out of reset.

NOTE: *Writing to CCH1 to update IROLN or to CCL1 to update IROLP will also update the primary carrier high and low data values. Care should be taken that bits 5–0 of the data to be written to CCH1 or CCL1 should contain the desired values for the primary carrier high or low data.*

In addition, writing to CCH1 to update IROLN will update the CMT polarity bit. Care should be taken that bit 6 of the data to be written to CCH1 should contain the desired values for the polarity bit.

12.10.2 CMT Modulator Control and Status Register

The CMT modulator control and status register (CMCS) contains the modulator and carrier generator enable (MCGEN), interrupt enable (EOCIE), mode select (MODE), baseband enable (BASE), extended space (EXSPC), divide-by-two prescaler (DIV2) bit, and the end of cycle (EOCF) status bit.

Address: $0014

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	EOCF	DIV2	0	EXSPC	BASE	MODE	EOCIE	MCGEN
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 12-8. CMT Modulator Control and Status Register (CMCS)

EOCF — End-of-Cycle Status Flag

1 = End of modulator cycle has occurred.

0 = No end of modulation cycle occurrence since flag last cleared

EOCF is set when:

- The modulator is not currently active and the MCGEN bit is set to begin the initial CMT transmission.
- At the end of each modulation cycle while the MCGEN bit is set. This is recognized when a match occurs between the contents of the space period register, SREG, and the down counter. At this time, the counter is initialized with the (possibly new) contents of the mark period buffer, MBUFF, and the space period register, SREG, is loaded with the (possibly new) contents of the space period buffer, SBUFF.

This flag is cleared by a read of the CMCS followed by an access of CMD2 or CMD3. The EOC flag is cleared by reset.

In the condition where the MCGEN bit is cleared and then set before the end of the modulation cycle, the EOC will not be set when the MCGEN is set, but will become set at the end of the current modulation cycle.

When the MCGEN becomes disabled, the CMT module does not set the EOC flag at the end of the last modulation cycle.

DIV2 — Divide-by-Two Prescaler Bit

1 = Divide-by-two prescaler enabled
0 = Divide-by-two prescaler disabled

The divide-by-two prescaler causes the CMT to be clocked at the bus rate when enabled and 2 x the bus rate when disabled (CGMOUT). Since this bit is not double buffered, it should not be set during a transmission.

EXSPC — Extended Space Enable Bit

1 = Extended space enabled
0 = Extended space disabled

For a description of the extended space enable bit, see **12.6.4 Extended Space Operation**. This bit is cleared by reset.

BASE — Baseband Enable Bit

1 = Baseband enabled
0 = Baseband disabled

When set, the BASE bit disables the carrier generator and forces the carrier output high for generation of baseband protocols. When BASE is clear, the carrier generator is enabled and the carrier output toggles at the frequency determined by values stored in the carrier data registers. See **12.6.2 Baseband Mode**. This bit is cleared by reset. This bit is not double buffered and should not be written to during a transmission.

MODE — Mode Select Bit

1 = CMT operates in FSK mode.
0 = CMT operates in time mode.

For a description of CMT operation in time mode, see **12.6.1 Time Mode**. For a description of CMT operation in FSK mode, see **12.6.3 FSK Mode**. This bit is cleared by reset. This bit is not double buffered and should not be written to during a transmission.

EOCIE — End-of-Cycle Interrupt Enable Bit

1 = CPU interrupt enabled
0 = CPU interrupt disabled

A CPU interrupt will be requested when EOCF is set if EOCIE was previously set. If EOCIE is clear, EOCF will not request a CPU interrupt.

MCGEN — Modulator and Carrier Generator Enable Bit

1 = Modulator and carrier generator enabled
0 = Modulator and carrier generator disabled

Setting MCGEN will initialize the carrier generator and modulator and will enable all clocks. Once enabled, the carrier generator and modulator will function continuously. When MCGEN is cleared, the current modulator cycle will be allowed to expire before all carrier and modulator clocks are disabled (to save power) and the modulator output is forced low. To prevent spurious operation, the user should initialize all data and control registers before enabling the system. This bit is cleared by reset.

12.10.3 CMT Modulator Data Registers (CMD1, CMD2, and CMD3)

The 12-bit MBUFF and SBUFF registers are accessed through three 8-bit registers, CMD1, CMD2, and CMD3. CMD2 and CMD3 contain the least significant eight bits of MBUFF and SBUFF respectively. CMD1 contains the two most significant nibbles of MBUFF and SBUFF. In many applications, periods greater than those obtained by eight bits will not be required. Splitting the registers up in this manner allows the user to clear CMD1 and generate 8-bit periods with just two data writes.

Register Name and Address: CMD1—$0015

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	MB11	MB10	MB9	MB8	SB11	SB10	SB9	SB8
Reset:	Unaffected by reset							

Register Name and Address: CMD2—$0016

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Reset:	Unaffected by reset							

Register Name and Address: CMD3—$0017

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SB7	SB6	SB5	SB4	SB3	SB2	SB1	SB0
Reset:	Unaffected by reset							

Figure 12-9. CMT Modulator Data Registers (CMD1, CMD2, and CMD3)

12.11 Wait Mode Operation

During wait mode the CMT, if enabled, will continue to operate normally. However, there will be no new codes or changes of pattern mode while in wait mode, as the CPU is not operating.

12.12 Stop Mode Operation

During stop mode, clocks to the CMT module are halted. No registers are affected.

NOTE: *Notice that because the clocks are halted, the CMT will resume upon exit from stop. Software should ensure that the stop mode is not entered while the modulator is still in operation to prevent the IRO pin from being asserted while in stop mode. This may require a timeout period from the time that the MCGEN bit is cleared to allow the last modulator cycle to complete.*

Section 13. Modulo Timer (TIM0I)

13.1 Contents

13.2 Introduction

This section describes the modulo timer (TIM0I) which is a periodic interrupt timer whose counter is clocked internally via software programmable options. **Figure 13-1** is a block diagram of the TIM0I.

13.3 Features

Features of the TIM0I include:

- Programmable TIM0I clock input
- Free-running or modulo up-count operation
- TIM0I counter stop and reset bits

13.4 Functional Description

Figure 13-1 shows the structure of the TIM0I. The central component of the TIM0I is the 16-bit TIM0I counter that can operate as a free-running counter or a modulo up-counter. The counter provides the timing reference for the interrupt. The TIM0I counter modulo registers, TMODH and TMODL, control the modulo value of the counter. Software can read the counter value at any time without affecting the counting sequence.

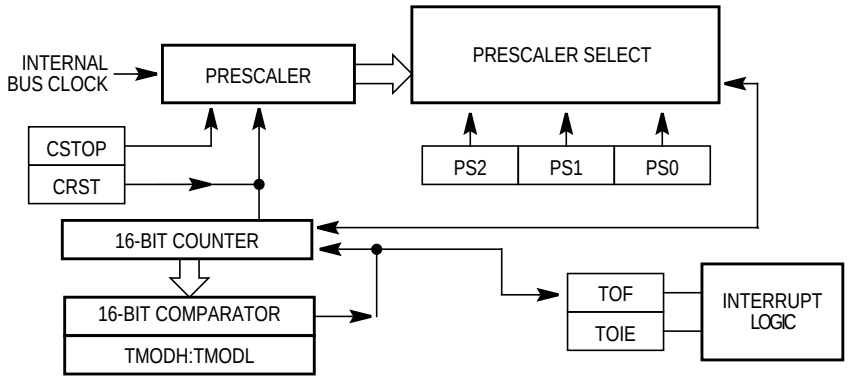


Figure 13-1. TIM0I Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0018	TIM Status and Control Register (TSC) See page 148.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0019	TIM Counter Register High (TCNTH) See page 150.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001A	TIM Counter Register Low (TCNTL) See page 150.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001B	TIM Counter Modulo Register High (TMODH) See page 151.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$001C	TIM Counter Modulo Register Low (TMODL) See page 151.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1

= Unimplemented

Table 13-1. TIM0I Register Summary

13.4.1 TIM0I Counter Prescaler

The clock source can be one of the seven prescaler outputs. The prescaler generates seven clock rates from the internal bus clock. The prescaler select bits, PS2–PS0, in the status and control register select the TIM0I clock source.

The value in the TIM0I counter modulo registers and the selected prescaler output determines the frequency of the periodic interrupt. The TIM0I overflow flag (TOF) is set when the TIM0I counter value rolls over to $0000 after matching the value in the TIM0I counter modulo registers. The TIM0I interrupt enable bit, TOIE, enables TIM0I overflow CPU interrupt requests. TOF and TOIE are in the TIM0I status and control register.

13.5 Interrupts

The TIM0I overflow flag (TOF) can generate an interrupt request. The TOF bit is set when the TIM0I counter value rolls over to $0000 after matching the value in the TIM0I counter modulo registers. The TIM0I overflow interrupt enable bit, TOIE, enables TIM0I overflow CPU interrupt requests. TOF and TOIE are in the TIM0I status and control register.

13.6 Low-Power Modes

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

13.6.1 Wait Mode

The TIM0I remains active in wait mode. Any enabled CPU interrupt request from the TIM0I can bring the MCU out of wait mode.

If TIM0I functions are not required during wait mode, reduce power consumption by stopping the TIM0I before executing the WAIT instruction.

13.6.2 Stop Mode

The TIM0I is inactive in stop mode. The STOP instruction does not affect register conditions or the state of the TIM0I counter. TIM0I operation resumes when the MCU exits stop mode after an external interrupt.

13.7 TIM0I During Break Interrupts

A break interrupt stops the TIM0I 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 **Section 18. Break Module (BRK)**.)

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

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

13.8 I/O Registers

These I/O registers control and monitor operation of the TIM0I:

- TIM0I status and control register (TSC)
- TIM0I counter registers (TCNTH and TCNTL)
- TIM0I counter modulo registers (TMODH and TMODL)

13.8.1 TIM0I Status and Control Register

The TIM0I status and control register:

- Enables TIM0I overflow interrupt
- Flags TIM0I overflows
- Stops the TIM0I counter
- Resets the TIM0I counter
- Prescales the TIM0I counter clock

Address: $0018

	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 13-2. TIM0I Status and Control Register (TSC)

TOF — TIM0I Overflow Flag Bit

This read/write flag is set when the TIM0I counter resets to $0000 after reaching the modulo value programmed in the TIM0I counter modulo registers. Clear TOF by reading the TIM0I status and control register when TOF is set and then writing a logic 0 to TOF. If another TIM0I overflow occurs before the clearing sequence is complete, then writing logic 0 to TOF has no effect. Therefore, a TOF interrupt request cannot be lost due to inadvertent clearing of TOF. Reset clears the TOF bit. Writing a logic 1 to TOF has no effect.

1 = TIM0I counter has reached modulo value.
0 = TIM0I counter has not reached modulo value.

TOIE — TIM0I Overflow Interrupt Enable Bit

This read/write bit enables TIM0I overflow interrupts when the TOF bit becomes set. Reset clears the TOIE bit.

1 = TIM0I overflow interrupts enabled
0 = TIM0I overflow interrupts disabled

TSTOP — TIM0I Stop Bit

This read/write bit stops the TIM0I counter. Counting resumes when TSTOP is cleared. Reset sets the TSTOP bit, stopping the TIM0I counter until software clears the TSTOP bit.

1 = TIM0I counter stopped
0 = TIM0I counter active

NOTE: *Do not set the TSTOP bit before entering wait mode if the TIM0I is required to exit wait mode.*

TRST — TIM0I Reset Bit

Setting this write-only bit resets the TIM0I counter and the TIM0I prescaler. Setting TRST has no effect on any other registers. Counting resumes from $0000. TRST is cleared automatically after the TIM0I counter is reset and always reads as logic 0. Reset clears the TRST bit.

1 = Prescaler and TIM0I counter cleared
0 = No effect

NOTE: *Setting the TSTOP and TRST bits simultaneously stops the TIM0I counter at a value of $0000.*

PS2–PS0 — Prescaler Select Bits

These read/write bits select one of the seven prescaler outputs as the input to the TIM0I counter as **Table 13-2** shows. Reset clears the PS2–PS0 bits.

Table 13-2. Prescaler Selection

PS2–PS0	TIM0I Clock Source
000	Internal bus clock ÷1
001	Internal bus clock ÷ 2
010	Internal bus clock ÷ 4
011	Internal bus clock ÷ 8
100	Internal bus clock ÷ 16
101	Internal bus clock ÷ 32
110	Internal bus clock ÷ 64
111	Internal bus clock ÷ 64

13.8.2 TIM0I Counter Registers

The two read-only TIM0I counter registers contain the high and low bytes of the value in the TIM0I 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 TIM0I counter registers. Setting the TIM0I reset bit (TRST) also clears the TIM0I counter registers.

NOTE: *If TCNTH is read 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.*

Register Name and Address: TCNTH—$0019

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

Register Name and Address: TCNTL—$001A

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

= Unimplemented

Figure 13-3. TIM0I Counter Registers (TCNTH and TCNTL)

13.8.3 TIM0I Counter Modulo Registers

The read/write TIM0I modulo registers contain the modulo value for the TIM0I counter. When the TIM0I counter reaches the modulo value, the overflow flag (TOF) becomes set, and the TIM0I counter resumes counting from $0000 at the next clock. Writing to the high byte (TMODH) inhibits the TOF bit and overflow interrupts until the low byte (TMODL) is written. Reset sets the TIM0I counter modulo registers.

Register Name and Address: TMODH—$001B

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

Register Name and Address: TMODL—$001C

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

Figure 13-4. TIM0I Counter Modulo Registers (TMODH and TMODL)

NOTE: *Reset the TIM0I counter before writing to the TIM0I counter modulo registers.*

At every system clock, the TMODH and TMODL value is compared to the TCNTH and TCNTL value. A valid compare flag is asserted on the first cycle in which the values match. If TMODH and TMODL are set to $0000, a TOF is generated on the first cycle in which the match occurs, but not subsequently. The circuit detects a change in the match value at each clock.

Section 14. External Interrupt (IRQ)

14.1 Contents

14.2 Introduction

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

14.3 Features

Features of the IRQ module include:

- A dedicated external interrupt pin ($\overline{\text{IRQ1}}$)
- IRQ1 interrupt control bits
- Hysteresis buffer
- Programmable edge-only or edge- and level-interrupt sensitivity
- Automatic interrupt acknowledge

14.4 Functional Description

A logic 0 applied to the external interrupt pin can latch a central processor unit (CPU) interrupt request. **Figure 14-1** shows the structure of the IRQ module.

NOTE: *The IRQ circuits are powered by the chip V_{DD}. External circuits should drive a logic 1 at the V_{DD} level. If external circuits use a pullup device on $\overline{IRQ1}$, the pin should be pulled to V_{DD} and **not** to BATT.*

Interrupt signals on the $\overline{IRQ1}$ pin are latched into the IRQ1 latch. An interrupt latch remains set until one of these actions occurs:

- Vector fetch — A vector fetch automatically generates an interrupt acknowledge signal that clears the latch that caused the vector fetch.
- Software clear — Software can clear an interrupt latch by writing to the appropriate acknowledge bit in the interrupt status and control register (ISCR). Writing a logic 1 to the ACK1 bit clears the IRQ1 latch.
- Reset — A reset automatically clears the interrupt latch.

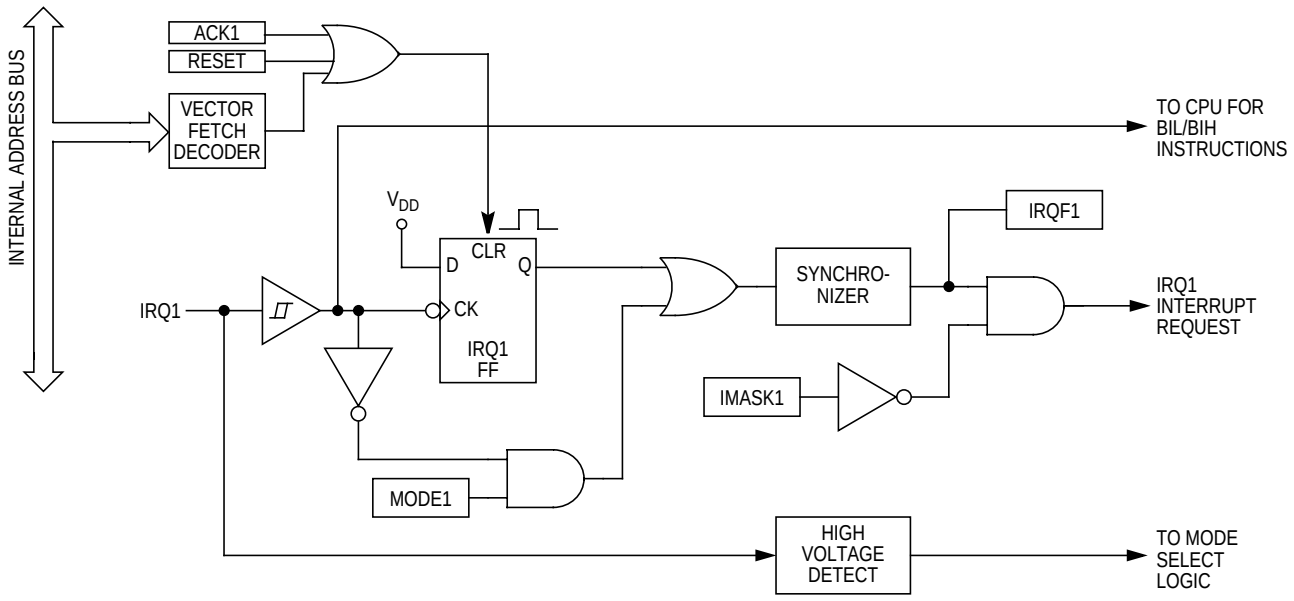


Figure 14-1. IRQ Module Block Diagram

The external interrupt pin is falling-edge-triggered and is software-configurable to be either falling-edge or low-level-triggered. The MODE1 bit in the ISCR controls the triggering sensitivity of the $\overline{\text{IRQ1}}$ pin.

When an interrupt pin is edge-triggered only, the interrupt remains set until a vector fetch, software clear, or reset occurs.

When an interrupt pin is both falling-edge and low-level-triggered, the interrupt remains set until both of these occur:

- Vector fetch or software clear
- Return of the interrupt pin to logic 1

The vector fetch or software clear may occur before or after the interrupt pin returns to logic 1. As long as the pin is low, the interrupt request remains pending. A reset will clear the latch and the MODE1 control bit, thereby clearing the interrupt even if the pin stays low.

When set, the IMASK1 bit in the ISCR masks all external interrupt requests. A latched interrupt request is not presented to the interrupt priority logic unless the IMASK1 bit is clear.

NOTE: *The interrupt mask (I) in the condition code register (CCR) masks all interrupt requests, including external interrupt requests.*

14.5 $\overline{\text{IRQ1}}$ Pin

A logic 0 on the $\overline{\text{IRQ1}}$ pin can latch an interrupt request into the IRQ1 latch. A vector fetch, software clear, or reset clears the IRQ1 latch.

If the MODE1 bit is set, the $\overline{\text{IRQ1}}$ pin is both falling-edge-sensitive and low-level-sensitive. With MODE1 set, both of these actions must occur to clear IRQ1:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the latch. Software may generate the interrupt acknowledge signal by writing a logic 1 to the ACK1 bit in the interrupt status and control register (ISCR). The ACK1 bit is useful in applications that poll the $\overline{\text{IRQ1}}$ pin and require software to clear the IRQ1 latch. Writing to the ACK1 bit prior to leaving an interrupt service routine can also prevent

spurious interrupts due to noise. Setting ACK1 does not affect subsequent transitions on the $\overline{\text{IRQ1}}$ pin. A falling edge that occurs after writing to the ACK1 bit generates another interrupt request. If the IRQ1 mask bit, IMASK1, is clear, the CPU loads the program counter with the vector address at locations $FFFA and $FFFB.

- Return of the $\overline{\text{IRQ1}}$ pin to logic 1 — As long as the $\overline{\text{IRQ1}}$ pin is at logic 0, IRQ1 remains active.

The vector fetch or software clear and the return of the $\overline{\text{IRQ1}}$ pin to logic 1 may occur in any order. The interrupt request remains pending as long as the $\overline{\text{IRQ1}}$ pin is at logic 0. A reset will clear the latch and the MODE1 control bit, thereby clearing the interrupt even if the pin stays low.

If the MODE1 bit is clear, the $\overline{\text{IRQ1}}$ pin is falling-edge-sensitive only. With MODE1 clear, a vector fetch or software clear immediately clears the IRQ1 latch.

The IRQF1 bit in the ISCR register can be used to check for pending interrupts. The IRQF1 bit is not affected by the IMASK1 bit, which makes it useful in applications where polling is preferred.

Use the BIH or BIL instruction to read the logic level on the $\overline{\text{IRQ1}}$ pin.

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

14.6 IRQ Module During Break Interrupts

The BCFE bit in the break flag control register (BFCR) enables software to clear the latch during the break state. See **Section 18. Break Module (BRK)**. To allow software to clear the IRQ1 latch during a break interrupt, write a logic 1 to the BCFE bit. If a latch is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect CPU interrupt flags during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), writing to the ACK1 bit in the IRQ status and control register during the break state has no effect on the IRQ interrupt flags.

14.7 IRQ Status and Control Register

The IRQ status and control register (ISCR) controls and monitors operation of the IRQ module. The ISCR has these functions:

- Shows the state of the IRQ1 flag
- Clears the IRQ1 latch
- Masks IRQ1 interrupt request
- Controls triggering sensitivity of the $\overline{\text{IRQ1}}$ interrupt pin

Address: $000F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	IRQF1	0	IMASK1	MODE1
Write:						ACK1		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-2. IRQ Status and Control Register (ISCR)

IRQF1 — IRQ1 Flag

This read-only status bit is high when the IRQ1 interrupt is pending.
1 = $\overline{\text{IRQ1}}$ interrupt pending
0 = $\overline{\text{IRQ1}}$ interrupt not pending

ACK1 — IRQ1 Interrupt Request Acknowledge Bit

Writing a logic 1 to this write-only bit clears the IRQ1 latch. ACK1 always reads as logic 0. Reset clears ACK1.

IMASK1 — IRQ1 Interrupt Mask Bit

Writing a logic 1 to this read/write bit disables IRQ1 interrupt requests. Reset clears IMASK1.
1 = IRQ1 interrupt requests disabled
0 = IRQ1 interrupt requests enabled

MODE1 — IRQ1 Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the $\overline{\text{IRQ1}}$ pin. Reset clears MODE1.
1 = $\overline{\text{IRQ1}}$ interrupt requests on falling edges and low levels
0 = $\overline{\text{IRQ1}}$ interrupt requests on falling edges only

External Interrupt (IRQ)

Section 15. Keyboard Interrupt Module (KBI)

15.1 Contents

15.2 Introduction

The keyboard interrupt module (KBI) provides eight independently maskable external interrupts.

15.3 Features

Features of the KBI include:

- Eight keyboard interrupt pins with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Hysteresis buffers
- Programmable edge-only or edge- and level- interrupt sensitivity
- Exit from low-power modes

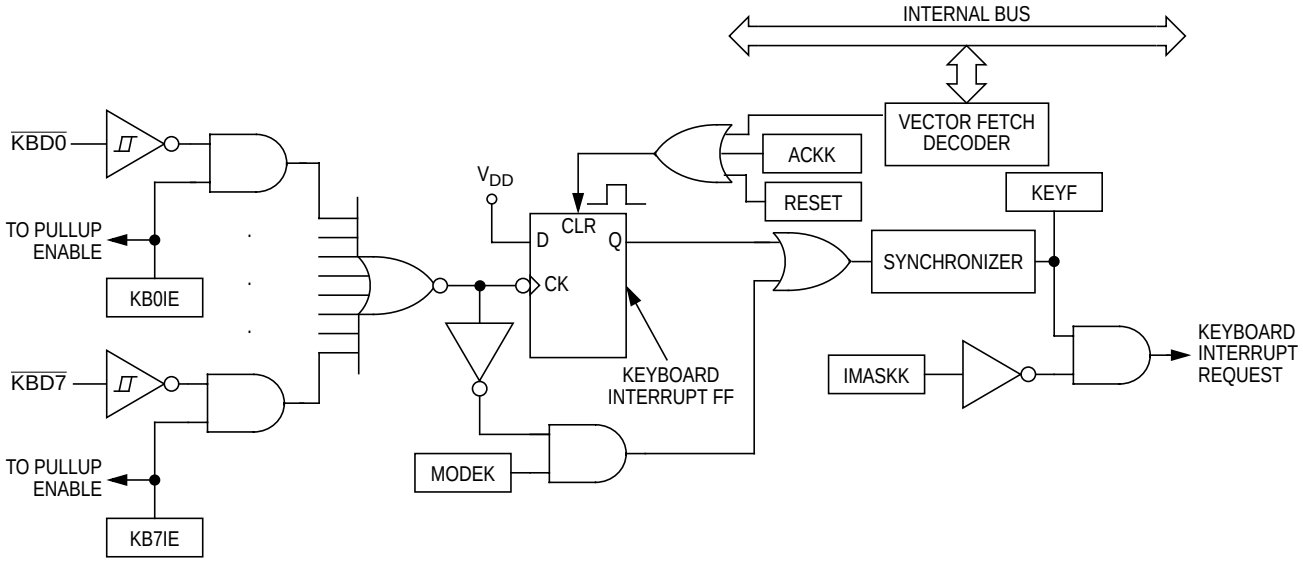


Figure 15-1. Keyboard Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000D	Keyboard Status and Control Register (KBSCR) See page 165.	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$000E	Keyboard Interrupt Enable Register (KBIER) See page 166.	Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 15-2. Keyboard Module I/O Register Summary

15.4 Functional Description

Writing to the KBIE7–KBIE0 bits in the keyboard interrupt enable register independently enables or disables each port B pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin also enables its internal pullup device. A logic 0 applied to an enabled keyboard interrupt pin latches a keyboard interrupt request.

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

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

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

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the interrupt request. Software may generate the interrupt acknowledge signal by writing a logic 1 to the ACKK bit in the keyboard status and control register (KBSCR). The ACKK bit is useful in applications that poll the keyboard interrupt pins and require software to clear the keyboard interrupt request. Writing to the ACKK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACKK does not affect subsequent transitions on the keyboard interrupt pins. A falling edge that occurs after writing to the ACKK bit latches another interrupt request. If the keyboard interrupt mask bit, IMASKK, is clear, the CPU loads the program counter with the vector address at locations $FFF4 and $FFF5.

- Return of all enabled keyboard interrupt pins to logic 1 — As long as any enabled keyboard interrupt pin is at logic 0, the keyboard interrupt remains set.

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

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

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

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

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

NOTE: *Setting a keyboard interrupt enable bit (KBIEx) forces the corresponding keyboard interrupt pin to be an input, overriding the data direction register. However, the data direction register bit must be a logic 0 for software to read the pin.*

15.5 Keyboard Initialization

When a keyboard interrupt pin is enabled, the pin may initially be low and cause a false interrupt to occur. A false interrupt on an edge-triggered pin can be acknowledged immediately after enabling the pin. A false interrupt on an edge- and level-triggered interrupt pin must be acknowledged after the pin has been pulled high.

The internal pullup device, the pin capacitance, as well as the external load will factor into the actual amount of time it takes for the pin to pull high. Considering only an internal pullup of 48 kΩ and pin capacitance of 8 pF, the pullup time will be on the order of 1 μs.

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.

Another way to avoid a false interrupt:

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

15.6 Low-Power Modes

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

15.6.1 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.

15.6.2 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.

15.7 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 logic 1 to the BCFE bit. If a latch is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect the latch during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), writing to the keyboard acknowledge bit (ACKK) in the keyboard status and control register during the break state has no effect. See **15.8.1 Keyboard Status and Control Register**.

15.8 I/O Registers

Two input/output registers control and monitor operation of the keyboard module:

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

15.8.1 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: $000D

	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 15-3. Keyboard Status and Control Register (KBSCR)

Bits 7–4 — Not used

These read-only bits always read as logic 0s.

KEYF — Keyboard Flag

This read-only bit is set when a keyboard interrupt is pending. Reset clears the KEYF bit.

1 = Keyboard interrupt pending
0 = No keyboard interrupt pending

ACKK — Keyboard Acknowledge Bit

Writing a logic 1 to this write-only bit clears the keyboard interrupt request. ACKK always reads as logic 0. Reset clears ACKK.

IMASKK — Keyboard Interrupt Mask Bit

Writing a logic 1 to this read/write bit prevents the output of the keyboard interrupt mask from generating interrupt requests. Reset clears the IMASKK bit.

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

MODEK — Keyboard Triggering Sensitivity Bit

This read/write bit controls the triggering sensitivity of the keyboard interrupt pins. Reset clears MODEK.

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

15.8.2 Keyboard Interrupt Enable Register

The keyboard interrupt enable register (KBIER) enables or disables each port B pin to operate as a keyboard interrupt pin.

Address: $000E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Reset:	0	0	0	0	0	0	0	0

Figure 15-4. Keyboard Interrupt Enable Register (KBIER)

KBIE7–KBIE0 — Keyboard Interrupt Enable Bits

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

1 = PTBx pin enabled as keyboard interrupt pin
0 = PTBx pin not enabled as keyboard interrupt pin

Section 16. Computer Operating Properly (COP)

16.1 Contents

16.2 Introduction

This 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. Clearing the COP counter periodically prevents a reset.

16.3 Functional Description

Figure 16-1 shows the structure of the COP module.

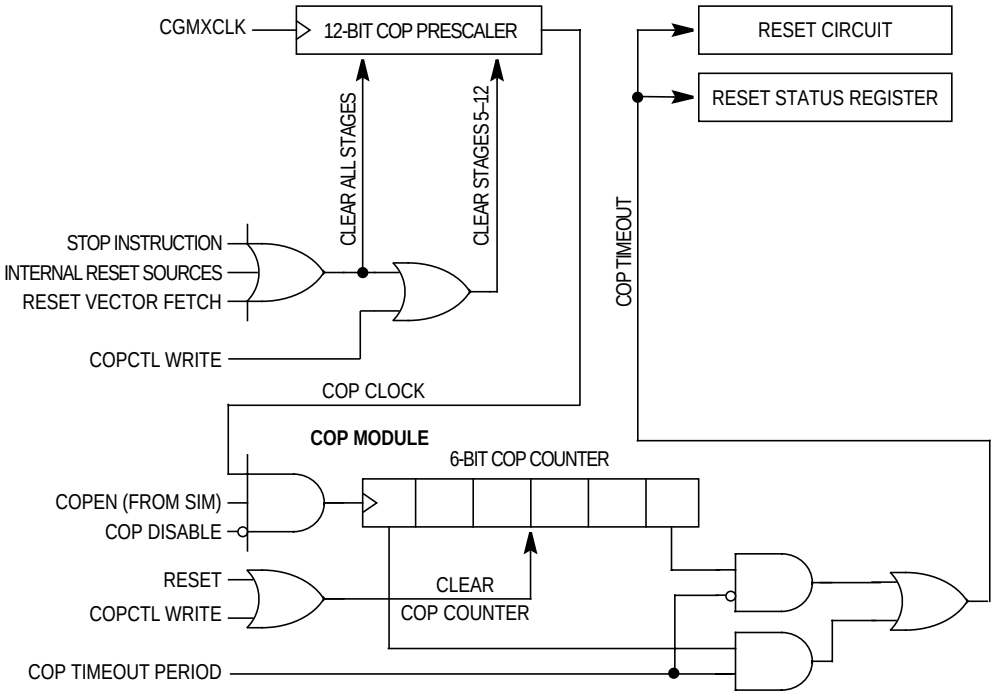


Figure 16-1. COP Block Diagram

The COP counter is a free-running 6-bit counter preceded by a 12-bit prescaler counter. If not cleared by software, the COP counter overflows and generates an asynchronous reset after $2^{18} - 2^4$ or $2^{13} - 2^4$ CGMXCLK cycles, depending on the state of the COP rate select bit COPRS, in the configuration register (CONFIG). With a $2^{18} - 2^4$ CGMXCLK cycle overflow option, a 4.9152-MHz crystal gives a COP timeout period is 53.3 ms. Writing any value to location $FFFF before an

overflow occurs prevents a COP reset by clearing the COP counter and stages 12 through 5 of the prescaler.

NOTE: *Service the COP immediately after reset and before entering or after exiting stop mode to guarantee the maximum time before the first COP counter overflow.*

A COP reset pulls the $\overline{\text{RST}}$ pin low for 32 CGMXCLK cycles and sets the COP bit in the reset status register (RSR).

In monitor mode, the COP is disabled if the $\overline{\text{RST}}$ pin or the $\overline{\text{IRQ1}}$ is held at V_{TST}. During the break state, V_{TST} on the $\overline{\text{RST}}$ pin disables the COP.

NOTE: *Place COP clearing instructions in the main program and not in an interrupt subroutine. Such an interrupt subroutine could keep the COP from generating a reset even while the main program is not working properly.*

16.4 I/O Signals

This section describes the signals shown in **Figure 16-1**.

16.4.1 CGMXCLK

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

16.4.2 STOP Instruction

The STOP instruction clears the COP prescaler.

16.4.3 COPCTL Write

Writing any value to the COP control register (COPCTL) (see **16.5 COP Control Register**) clears the COP counter and clears bits 12–5 of the prescaler. Reading the COP control register returns the low byte of the reset vector.

16.4.4 Power-On Reset

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

16.4.5 Internal Reset

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

16.4.6 Reset Vector Fetch

A reset vector fetch occurs when the vector address appears on the data bus. A reset vector fetch clears the COP prescaler.

16.4.7 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the configuration register (CONFIG).

16.4.8 COPRS (COP Rate Select)

The COPRS signal reflects the state of the COP rate select bit (COPRS) in the configuration register.

16.5 COP Control Register

The COP control register is located at address $FFFF and overlaps the reset vector. Writing any value to $FFFF clears the COP counter and starts a new timeout period. Reading location $FFFF returns the low byte of the reset vector. See **Figure 16-2**.

Address:	$FFFF							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Low byte of reset vector							
Write:	Writing clears COP counter (any value)							
Reset:	Unaffected by reset							

Figure 16-2. COP Control Register (COPCTL)

16.6 Interrupts

The COP does not generate central processor unit (CPU) interrupt requests.

16.7 Monitor Mode

The COP is disabled in monitor mode when V_{TST} is present on the $\overline{IRQ1}$ pin or on the $\overline{RST}$ pin.

16.8 Low-Power Modes

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

16.8.1 Wait Mode

The COP remains active during wait mode. To prevent a COP reset during wait mode, periodically clear the COP counter in a CPU interrupt routine.

16.8.2 Stop Mode

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

The STOP bit in the CONFIG register enables the STOP instruction. To prevent inadvertently turning off the COP with a STOP instruction, disable the STOP instruction by clearing the STOP bit.

16.9 COP Module During Break Mode

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

Section 17. Monitor ROM (MON)

17.1 Contents

17.2 Introduction

This section describes the monitor read-only memory (MON). The monitor ROM allows complete testing of the MCU through a single-wire interface with a host computer.

17.3 Features

Features of the monitor ROM include:

- Normal user-mode pin functionality
- One pin dedicated to serial communication between monitor ROM and host computer
- Standard mark/space non-return-to-zero (NRZ) communication with host computer
- Execution of code in random-access memory (RAM) or ROM
- ROM memory security[1]

17.4 Functional Description

The monitor ROM receives and executes commands from a host computer. **Figure 17-1** shows an example circuit used to enter monitor mode and communicate with a host computer via a standard RS-232 interface.

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

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

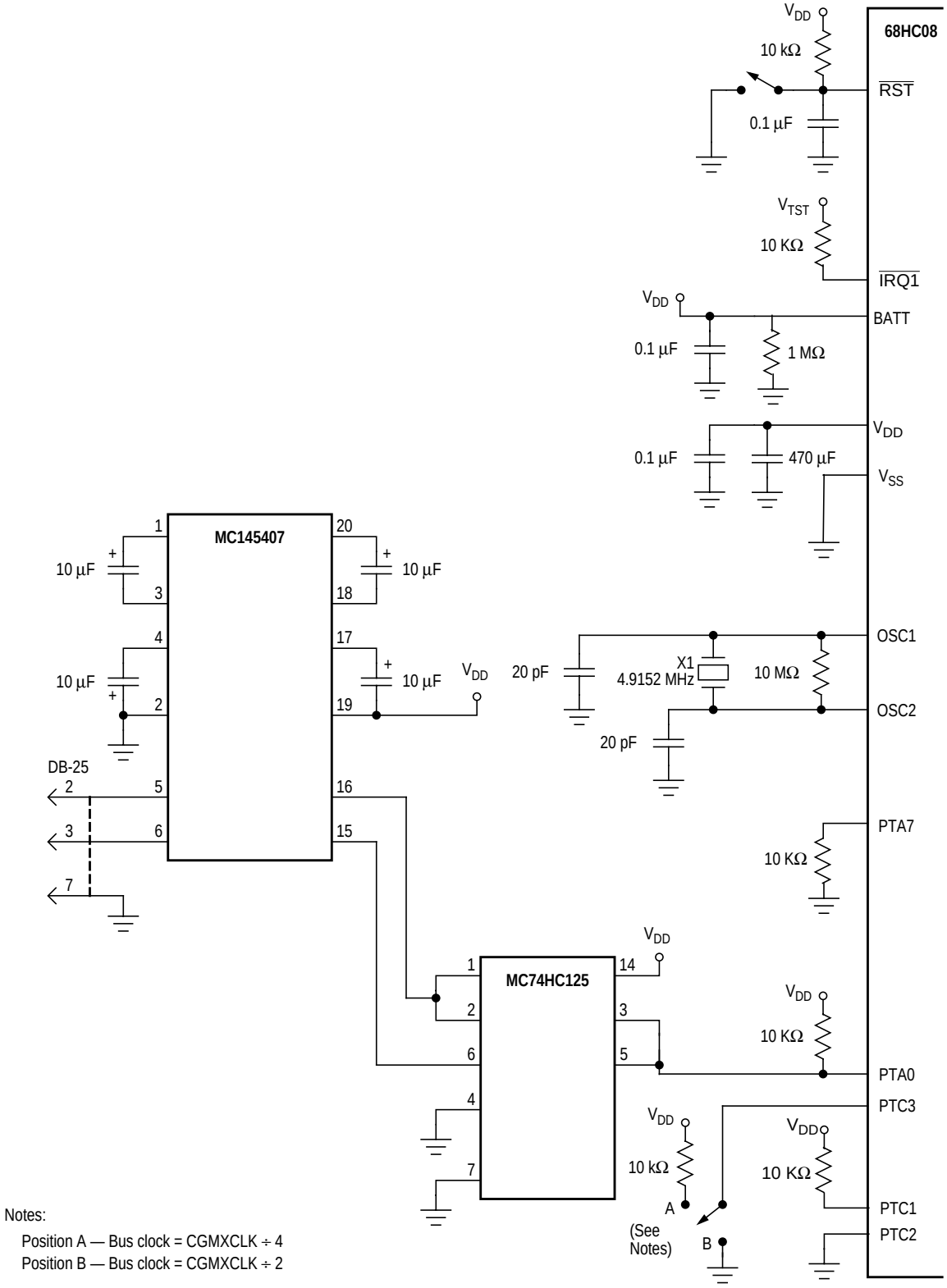


Figure 17-1. Monitor Mode Circuit

17.4.1 Entering Monitor Mode

Table 17-1 shows the pin conditions for entering monitor mode.

Table 17-1. Monitor Mode Entry

$\overline{IRQ1}$ Pin	PTA7 Pin	PTC1 Pin	PTC2 Pin	PTA0 Pin	PTC3 Pin	CGMOUT	Bus Frequency
V_{TST}	0	1	0	1	1	$\frac{CGMXCLK}{2}$	$\frac{CGMOUT}{2}$
					0	CGMXCLK	

If PTC3 is low upon monitor mode entry, CGMOUT is equal to the crystal frequency. The bus frequency in this case is a divide-by-two of the input clock. If PTC3 is high upon monitor mode entry, the bus frequency will be a divide-by-four of the input clock.

NOTE: *Holding the PTC3 pin low when entering monitor mode causes a bypass of a divide-by-two stage at the oscillator. The CGMOUT frequency is equal to the CGMXCLK frequency, and the OSC1 input directly generates internal bus clocks. In this case, the OSC1 signal must have a 50 percent duty cycle at maximum bus frequency.*

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

NOTE: *The PTA7 pin must remain at logic 0 for 24 bus cycles after the $\overline{RST}$ pin goes high.*

Once out of reset, the MCU waits for the host to send eight security bytes. (See **17.5 Security**.) After the security bytes, the MCU sends a break signal (10 consecutive logic 0s) to the host, indicating that it is ready to receive a command.

In monitor mode, the MCU uses different vectors for reset, software interrupt (SWI), 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.

The COP module is disabled in monitor mode as long as V_{TST} is applied to either the $\overline{IRQ}$ pin or the $\overline{RST}$ pin.

Table 17-2 summarizes the differences between user mode and monitor mode.

Table 17-2. Mode Differences

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

1. If the high voltage (V_{TST}) is removed from the $\overline{IRQ1}$ pin or the $\overline{RST}$ pin, the SIM asserts its COP enable output. The COP is a configuration option enabled or disabled by the COPD bit in the configuration register.

17.4.2 Data Format

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

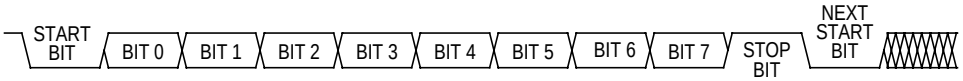


Figure 17-2. Monitor Data Format

17.4.3 Break Signal

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

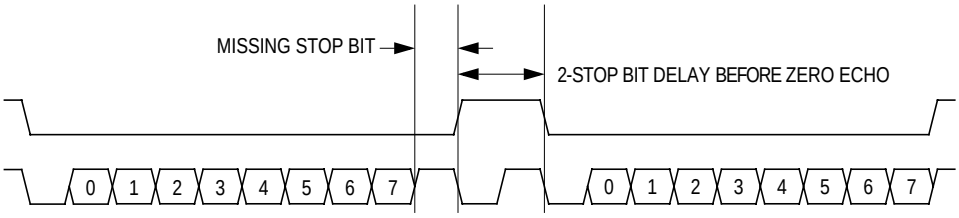


Figure 17-3. Break Transaction

17.4.4 Baud Rate

The communication baud rate is controlled by the crystal frequency and the state of the PTC3 pin upon entry into monitor mode. When PTC3 is high, the divide by ratio is 1024. If the PTC3 pin is at logic 0 upon entry into monitor mode, the divide by ratio is 512. **Table 17-3** lists crystal frequencies required to achieve standard baud rates. Other standard baud rates can be accomplished using higher crystal frequencies.

Table 17-3. Monitor Baud Rate Selection

Crystal Frequency (MHz)	PTC3 Pin	Baud Rate
4.9152	0	9600
4.9152	1	4800

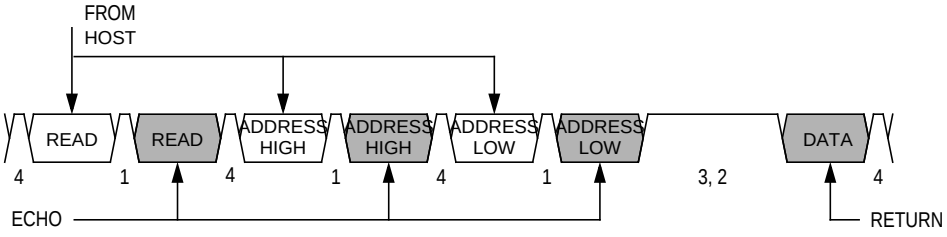
17.4.5 Commands

The monitor ROM firmware uses these commands:

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

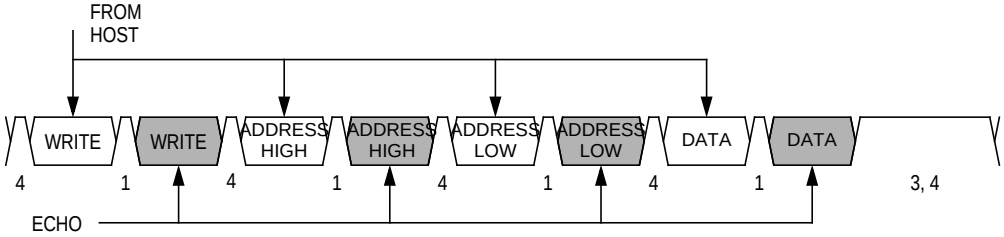
The monitor ROM firmware echoes each received byte back to the PTA0 pin for error checking. An 11-bit delay at the end of each command allows the host to send a break character to cancel the command. A delay of two bit times occurs before each echo and before READ, IREAD, or READSP data is returned.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

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



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

Figure 17-4. Read Transaction



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

Figure 17-5. Write Transaction

Refer to **Table 17-4**, **Table 17-5**, **Table 17-6**, **Table 17-7**, **Table 17-8**, and **Table 17-9** for brief descriptions of each monitor mode command.

Table 17-4. READ (Read Memory) Command

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

Command Sequence

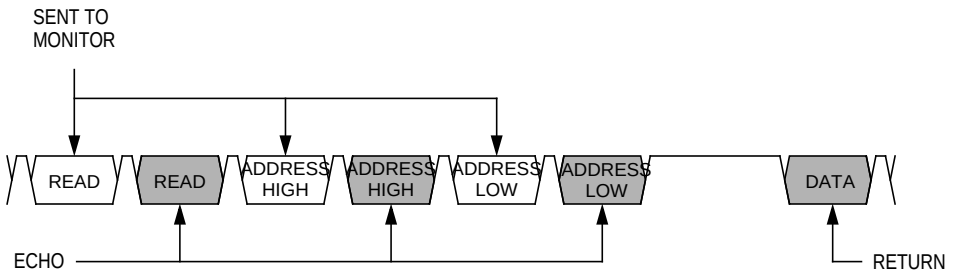


Table 17-5. WRITE (Write Memory) Command

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

Command Sequence

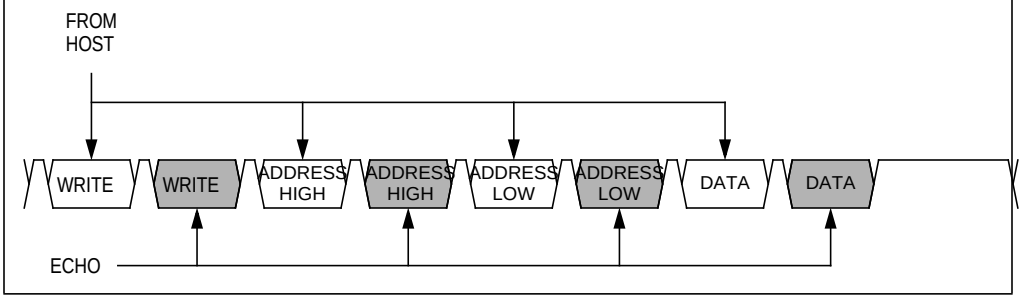


Table 17-6. IREAD (Indexed Read) Command

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

Command Sequence

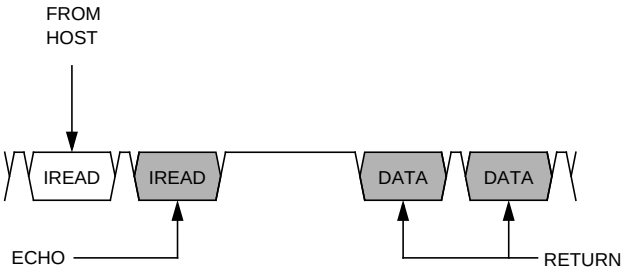
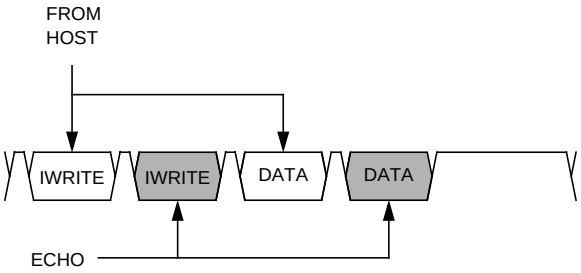


Table 17-7. IWRITE (Indexed Write) Command

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

Command Sequence



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

Table 17-8. READSP (Read Stack Pointer) Command

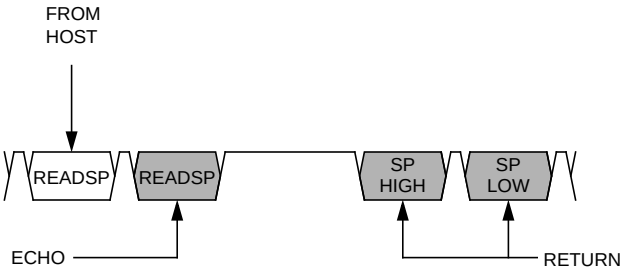
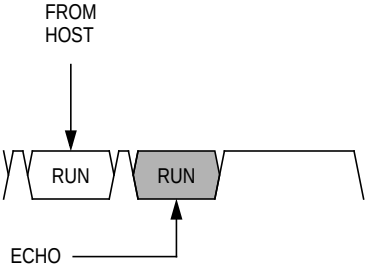
Description	Reads stack pointer
Operand	None
Data Returned	Returns incremented stack pointer value (SP + 1) in high byte:low byte order
Opcode	$0C
Command Sequence	

Table 17-9. RUN (Run User Program) Command

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

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

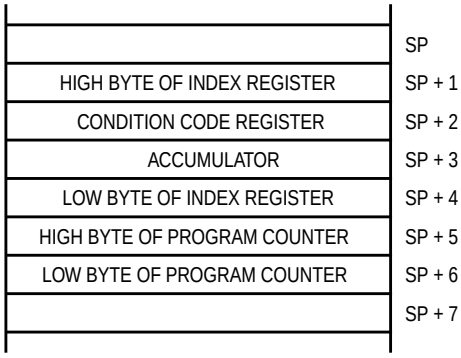


Figure 17-6. Stack Pointer at Monitor Mode Entry

17.5 Security

A security feature discourages unauthorized reading of ROM and 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 PA0.

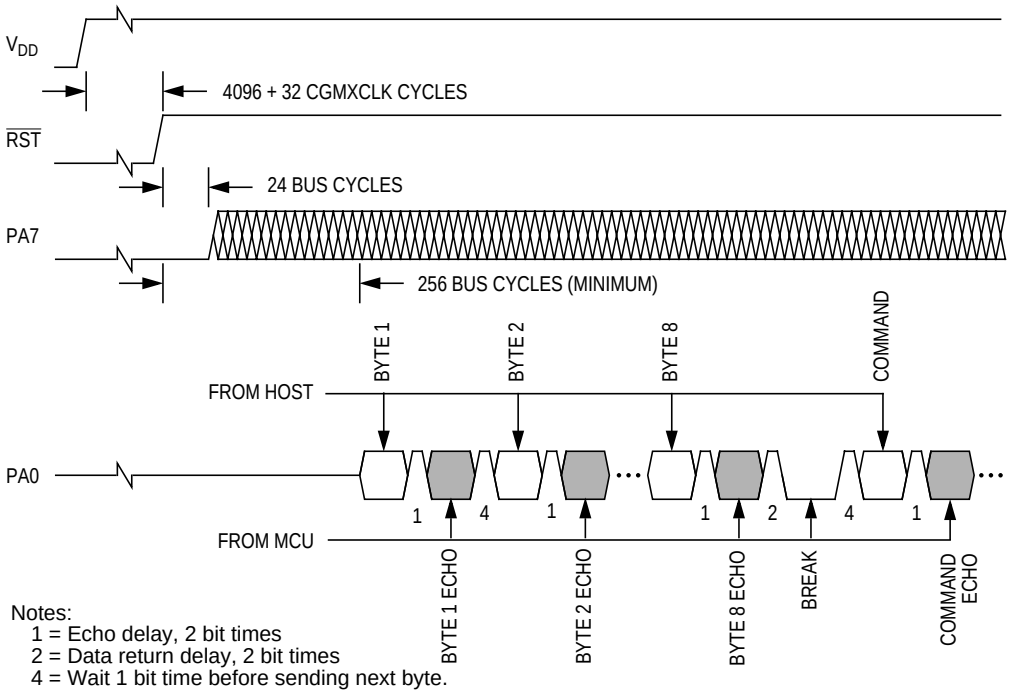


Figure 17-7. Monitor Mode Entry Timing

If the received bytes match those at locations $FFF6–$FFFD, the host bypasses the security feature and can read all ROM locations and execute code from ROM. Security remains bypassed until a power-on reset occurs. After the host bypasses security, any reset other than a power-on reset requires the host to send another eight bytes. If the reset was not a power-on reset, security remains bypassed regardless of the data that the host sends.

If the received bytes 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 ROM locations returns undefined data, and trying to execute code from ROM causes an illegal address reset. After the host fails to bypass security, any reset other than a power-on reset causes an endless loop of illegal address resets.

After receiving the eight security bytes from the host, the MCU transmits a break character signalling 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.*

Section 18. Break Module (BRK)

18.1 Contents

18.2 Introduction

This section describes the 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.

18.3 Features

Features of the break module 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

18.4 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal to the CPU. The CPU then loads the instruction register with a software interrupt instruction (SWI) after completion of the current CPU instruction. The program counter vectors to $FFFC and $FFFD ($FEFC and $FEFD in monitor mode).

These events can cause a break interrupt to occur:

- A CPU-generated address (the address in the program counter) matches the contents of the break address registers.
- Software writes a logic 1 to the BRKA bit in the break status and control register.

When a CPU-generated address matches the contents of the break address registers, the break interrupt begins after the CPU completes its current instruction. A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation. **Figure 18-1** shows the structure of the break module.

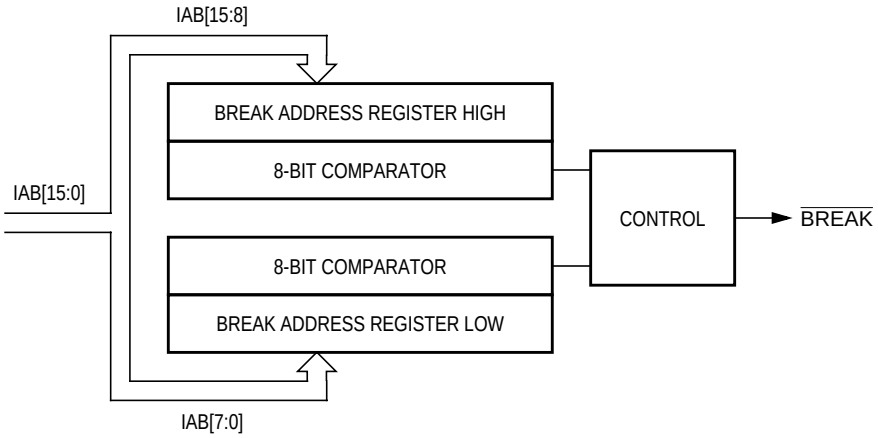


Figure 18-1. Break Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE00	Break Status Register (BSR) See page 191.	Read:	0	0	0	1	0	0	BW	0
		Write:	R	R	R	R	R	R	Note	R
		Reset:	0	0	0	1	0	0	0	0
$FE03	Break Flag Control Register (BFCR) See page 192.	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE0C	Break Address Register High (BRKH) See page 190.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0D	Break Address Register Low (BRKL) See page 190.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0E	Break Status and Control Register (BSCR) See page 189.	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

Note: Writing a logic 0 clears BW.

= Unimplemented R = Reserved

Figure 18-2. Break Module Register Summary

18.4.1 Flag Protection During Break Interrupts

The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state.

18.4.2 CPU During Break Interrupts

The CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with $FFFC and $FFFD ($FEFC and $FEFD in monitor mode)

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

18.4.3 TIM0I During Break Interrupts

A break interrupt stops the timer counter.

18.4.4 COP During Break Interrupts

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

18.5 Low-Power Modes

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

18.5.1 Wait Mode

If enabled, the break module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if SBSW is set. See **Section 7. Low-Power Modes**. Clear the BW bit by writing logic 0 to it.

18.5.2 Stop Mode

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

18.6 Break Module Registers

These registers control and monitor operation of the break module:

- Break status and control register (BSCR)
- Break address register high (BRKH)
- Break address register low (BRKL)
- Break status register (BSR)
- Break flag control register (BFCR)

18.6.1 Break Status and Control Register

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

Address: $FE0E

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

= Unimplemented

Figure 18-3. Break Status and Control Register (BSCR)

BRKE — Break Enable Bit

This read/write bit enables breaks on break address register matches. Clear BRKE by writing a logic 0 to bit 7. Reset clears the BRKE bit.

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

BRKA — Break Active Bit

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

1 = When read, break address match
0 = When read, no break address match

18.6.2 Break Address Registers

The break address registers (BRKH and BRKL) contain the high and low bytes of the desired breakpoint address. Reset clears the break address registers.

Register Name and Address: BRKH—$FE0C

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

Register Name and Address: BRKL—$FE0D

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

Figure 18-4. Break Address Registers (BRKH and BRKL)

18.6.3 Break Status Register

The break status register (BSR) contains a flag to indicate that a break caused an exit from wait mode. The flag is useful in applications requiring a return to wait mode after exiting from a break interrupt.

Address: $FE00

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

R = Reserved

Note: Writing a logic 0 clears BW.

Figure 18-5. Break Status Register (BSR)

BW — Break Wait Bit

This read/write bit is set when a break interrupt causes an exit from wait mode. Clear BW by writing a logic 0 to it. Reset clears BW.

1 = Break interrupt during wait mode

0 = No break interrupt during wait mode

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

This code works if the H register was stacked in the break interrupt routine. Execute this code at the end of the break interrupt routine.

```
 HIBYTE EQU  5
 LOBYTE EQU  6
;       If not BW, do RTI
        BRCLR BW,BSR, RETURN    ; See if wait mode or stop mode
                                ; was exited by break.
        TST   LOBYTE,SP         ; If RETURNLO is not 0,
        BNE   DOLO              ; then just decrement low byte.
        DEC   HIBYTE,SP         ; Else deal with high byte also.
 DOLO   DEC   LOBYTE,SP         ; Point to WAIT/STOP opcode.
 RETURN PULH                    ; Restore H register.
        RTI
```

18.6.4 Break Flag Control Register

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

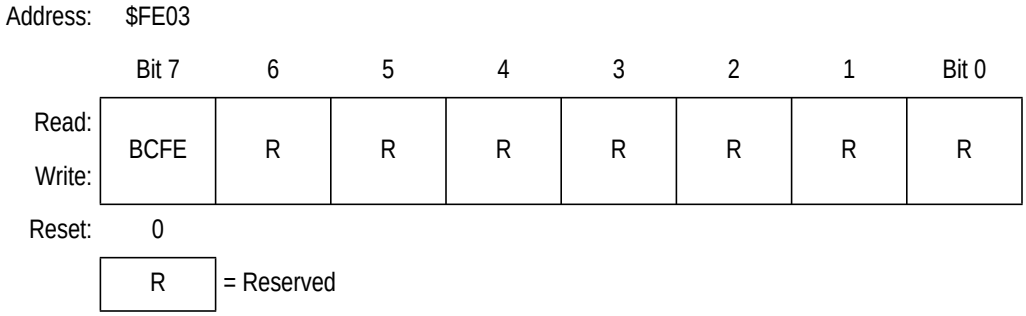


Figure 18-6. Break Flag Control Register (BFCR)

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 19. Preliminary Electrical Specifications

19.1 Contents

19.2 Introduction

This section contains electrical and timing specifications. These values are design targets and have not yet been fully tested.

19.3 Absolute Maximum Ratings

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

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to* ***19.6 3.3-Volt DC Electrical Characteristics*** *and* ***19.7 2.0-Volt DC Electrical Characteristics*** *for guaranteed operating conditions.*

Characteristic[1]	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
Maximum current per pin excluding V_{DD} and V_{SS}	I	± 25	mA
Storage temperature	T_{STG}	–55 to +150	°C
Maximum current out of V_{SS}	I_{MVSS}	100	mA
Maximum current into V_{DD}	I_{MVDD}	100	mA

1. Voltages referenced to V_{SS}.

NOTE: *This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{In} and V_{Out} be constrained to the range $V_{SS} \le (V_{In}$ or $V_{Out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD}.)*

19.4 Functional Operating Range

Characteristic	Symbol	Min	Max	Unit
Operating temperature range	T_A	0	70	°C
Operating voltage range (1)	V_{DD}	1.8	3.6	V
Battery input operating voltage range (2)	BATT	2.1	3.6	V

1. V_{DD} is the supply node for the MCU. V_{DD} is supplied from the BATT pin and is calculated as $V_{DD} = V_{BATT} – V_F$ where V_F is the forward voltage of the internal P-channel MOSFET. See **19.11 Battery Detection Characteristics**.
2. Minimum battery voltage is for guaranteed MCU operation. The MCU will operate lower when the LVI trip point (V_{LVR}) is less than maximum.

19.5 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance SOIC PDIP LQFP	θ_{JA}	 60 60 124	°C/W
I/O pin power dissipation	$P_{I/O}$	User determined	W
Power dissipation(1)	P_D	$(I_{BATT} \times V_{BATT}) + P_{I/O}$	W
Average junction temperature	T_J	$T_A + (P_D \times \theta_{JA})$	°C
Maximum junction temperature	T_{JM}	125	°C

1. Power dissipation is a function of temperature.

19.6 3.3-Volt DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage (I_{Load} = –2.5 mA) port A, port B (I_{Load} = –12 mA) IRO (I_{Load} = –3.0 mA) port C	 V_{OH} V_{OH} V_{OH}	 V_{DD} –0.3 V_{DD} –0.7 V_{DD} –0.3	 — — —	 — — —	 V V V
Output low voltage (I_{Load} = 4 mA) port A, port B (I_{Load} = 24 mA) IRO (I_{Load} = 14 mA) port C	 V_{OL} V_{OL} V_{OL}	 — — —	 — — —	 0.3 0.8 0.3	 V V V
Input high voltage All ports, $\overline{IRQ1}$, $\overline{RST}$, BATT, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input low voltage All ports, $\overline{IRQ1}$, $\overline{RST}$, BATT, OSC1	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
V_{DD} supply current Run, f_{OP} = 2.0 MHz[3] Wait[4] Stop[5] 25°C 0°C to 70°C LP reset with battery out [6] 25°C 0°C to 70°C	 I_{DD}	 — — — — — —	 3 1.25 1.0 1.0 0.25 0.25	 3.3 1.75 1.0 1.0 0.25 0.25	 mA mA µA µA µA mA
I/O ports hi-z leakage current	I_{IL}	—	—	± 4	µA
Input current	I_{In}	—	—	± 1	µA
Capacitance Ports (as input or output), $\overline{RST}$, $\overline{IRQ1}$, BATT	C_{Out} C_{In}	— —	— —	12 8	pF
Monitor mode entry voltage	V_{TST}	V_{DD} + 2.5	—	9	V
Pullup resistor PTB7/$\overline{KBD7}$–PTB0/$\overline{KBD0}$ 25°C 0°C to 70°C	R_{PU}	 27 24	 — —	 39 48	kΩ

1. V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range, 25°C only.
3. Run (operating) I_{DD} measured using external square wave clock source. LVI enabled. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD}. Measured with all modules enabled.
4. Wait I_{DD} measured using external square wave clock source ($f_{CGMXCLK}$ = 4 MHz); all inputs 0.2 V from rail; no dc loads; less than 100 pF on all outputs. C_L = 20 pF on OSC2; OSC2 capacitance linearly affects wait I_{DD}.
5. Stop I_{DD} measured with OSC1 grounded and no port pins sourcing current. LVI is disabled by stop entry.
6. I_{DD} measurement when BATT supply is removed and part is in LP reset.

19.7 2.0-Volt DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage (I_{Load} = –1.2 mA) port A, port B (I_{Load} = –6 mA) IRO (I_{Load} = –1.2 mA) port C	 V_{OH} V_{OH} V_{OH}	 V_{DD} –0.3 V_{DD} –0.7 V_{DD} –0.3	 — — —	 — — —	 V V V
Output low voltage (I_{Load} = 2 mA) port A,port B (I_{Load} = 11 mA) IRO (I_{Load} = 7 mA) port C	 V_{OL} V_{OL} V_{OL}	 — — —	 — — —	 0.3 0.8 0.3	 V V V
Input high voltage All ports, $\overline{IRQ1}$, $\overline{RST}$, BATT, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input low voltage All ports, $\overline{IRQ1}$, $\overline{RST}$, BATT, OSC1	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
V_{DD} supply current Run, f_{OP} = 2.0 MHz[3] Wait [4] Stop[5] or LP reset with battery in [6] 25°C 0°C to 70°C LP reset with battery out [7] 25°C 0°C to 70°C	 I_{DD}	 — — — — — —	 1 0.5 0.3 0.3 0.1 0.1	 1.1 0.6 0.5 0.5 0.1 0.1	 mA mA µA µA µA µA
I/O ports hi-z leakage current	I_{IL}	—	—	± 4	µA
Input current	I_{In}	—	—	± 1	µA
Capacitance Ports (as input or output), $\overline{RST}$, $\overline{IRQ1}$, BATT	C_{Out} C_{In}	— —	— —	12 8	pF
Monitor mode entry voltage	V_{TST}	V_{DD} + 2.5	—	9	V
Pullup resistor PTB7/$\overline{KBD7}$–PTB0/$\overline{KBD0}$ 25°C 0°C to 70°C	R_{PU}	 27 24	 — —	 39 48	kΩ

1. V_{DD} = 2.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range, 25°C only.
3. Run (operating) I_{DD} measured using external square wave clock source. LVI enabled. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD}. Measured with all modules enabled.
4. Wait I_{DD} measured using external square wave clock source ($f_{CGMXCLK}$ = 4 MHz); all inputs 0.2 V from rail; no dc loads; less than 100 pF on all outputs. C_L = 20 pF on OSC2; OSC2 capacitance linearly affects wait I_{DD}.
5. Stop I_{DD} measured with OSC1 grounded and no port pins sourcing current. LVI is disabled by stop entry.
6. I_{DD} measurement when BATT supply is below V_{LVR} and part is in LP reset.
7. I_{DD} measurement when BATT supply is removed and part is in LP reset.

19.8 Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency[2]	f_{OP}	—	2.0	MHz
$\overline{RST}$ input pulse width low[3]	t_{IRL}	125	—	ns
$\overline{IRQ1}$ interrupt pulse width low[4] (edge-triggered)	t_{ILIH}	125	—	ns

1. V_{DD} = 1.8 to 5.5 Vdc; V_{SS} = 0 Vdc; timing shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted
2. Some modules may require a minimum frequency greater than dc for proper operation; see appropriate table for this information.
3. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.
4. Minimum pulse width is for guaranteed interrupt. It is possible for a smaller pulse width to be recognized.

19.9 Oscillator Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Crystal frequency[1]	$f_{CGMXCLK}$[2]	1	—	8	MHz
External clock Reference frequency[1] [3]	$f_{CGMXCLK}$[2]	dc	—	8	MHz
Crystal load capacitance[4]	C_L	—	—	—	
Crystal fixed capacitance[4]	C_1	—	2 x C_L	—	
Crystal tuning capacitance[4]	C_2	—	2 x C_L	—	
Feedback bias resistor	R_B	—	10 MΩ	—	
Series resistor[4] [5]	R_S	—	—	—	

1. The CMT module is designed to function at $f_{CGMXCLK}$ = 8 MHz. The values given here are oscillator specifications.
2. $f_{OP} = f_{CGMXCLK/4}$
3. No more than 10% duty cycle deviation from 50%
4. Consult crystal vendor data sheet
5. Not required for high frequency crystals

19.10 LVI Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
LVI low battery sense voltage (1)	V_{LVS}	2.00	2.05	2.10	V
LVI trip voltage	V_{LVR}	1.85	1.9	1.95	V
LVI Trip voltage hysteresis	H_{LVR}	40	70	100	mV
V_{DD} slew rate, rising	SR_R	—	—	0.05	V/µs
V_{DD} slew rate, falling	SR_F	—	—	0.10	V/µs
Response time $R \leq SR_{MAX}$	t_{RESP}	—	—	$6.0 + \frac{40}{f_{CGMXCLK}}$	µs
Response time $SR > SR_{MAX}$	t_{RESP}	—	—	$\left(\frac{V_{DD}-V_{LVR}}{SR_{max}} - \frac{V_{DD}-V_{LVR}}{SR}\right) + 6.0 + \frac{40}{f_{CGMXCLK}}$	µs
LVI supply current @ V_{DD} = 5.5 V @ V_{DD} = 3.6 V @ V_{DD} =1.8 V	I_{LVI}	 — — —	 — — —	 100 50 25	 mA µA µA
Enable time (enable to output transition)	t_{EN}	—	—	50	µs

1. The LVI samples V_{DD}. V_{LVR} and V_{LVS} are V_{DD} voltages.

19.11 Battery Detection Characteristics

Characteristic	Symbol	Min	Max	Unit
Battery insertion rearm voltage(1)	V_{BR}	—	100	mV
BATT to V_{DD} voltage @ BATT = 2.1 V	V_F	—	200	mV
BATT to V_{DD} voltage @ BATT = 3.3 V	V_F	—	300	mV
Supply rise time ramp rate(2)	R_{POR}	0.035	—	V/ms

1. When supply is removed from the BATT pin, the pin voltage must fall below V_{BR} before supply is reapplied to guarantee that battery reinsertion is detected.
2. If minimum BATT pin voltage is not reached before the internal POR reset is released, $\overline{RST}$ must be driven low externally until minimum BATT voltage is reached.

19.12 Battery Circuit Component Specifications

Characteristic	Symbol	Typ	Max	Unit
External V_{DD} supply capacitor(1)	C_{VBULK}	—	470 ± 20%	µF
External V_{DD} bypass capacitor(2)	C_{VBYP}	0.1 ± 20%	—	µF
External BATT capacitor2	C_{BATT}	0.1 ± 20%	—	µF
External BATT resistor	R_{BATT}	1 ± 5%	—	MΩ

1. Capacitor is electrolytic, ± 20%, ESR < 4 Ω
2. Capacitor is ceramic, ± 20%

19.13 Memory Characteristics

Characteristic	Symbol	Min	Max	Unit
RAM data retention voltage	V_{RDR}	1.3	—	V
FLASH pages per row	—	8	8	Pages
FLASH bytes per page	—	8	8	Bytes
FLASH read bus clock frequency	f_{Read} (1)	32 K	8.4 M	Hz
FLASH charge pump clock frequency (see **4.5 Charge Pump Frequency Control**)	f_{Pump} (2)	1.8	2.3	MHz
FLASH block/bulk erase time	t_{Erase}	100	—	ms
FLASH high voltage kill time	t_{Kill}	200	—	µs
FLASH return to read time	t_{HVD}	50	—	µs
FLASH page program pulses	fls_{Pulses} (3)	1	20	Pulses
FLASH page program step size	t_{Step} (4)	0.8	5	ms
FLASH cumulative program time per row between erase cycles	t_{Row} (5)	—	800	ms
FLASH HVEN low to MARGIN high time	t_{HVTV}	50	—	µs
FLASH MARGIN high to PGM low time	t_{VTP}	150	—	µs
FLASH row erase endurance (6)	—	100	—	Cycles
FLASH row program endurance (7)	—	100	—	Cycles
FLASH data retention time (8)	—	10	—	Years

1. f_{Read} is defined as the frequency range for which the FLASH memory can be read.
2. f_{Pump} is defined as the charge pump clock frequency required for program, erase, and margin read operations.
3. fls_{Pulses} is defined as the number of pulses used to program the FLASH using the required smart program algorithm.
4. t_{Step} is defined as the amount of time during one page program cycle that HVEN is held high.
5. t_{Row} is defined as the cumulative time a row can see the program voltage before the row must be erased before further programming.
6. The minimum row endurance value specifies each row of the FLASH memory is guaranteed to work for at least this many erase / program cycles.
7. The minimum row endurance value specifies each row of the FLASH memory is guaranteed to work for at least this many erase / program cycles.
8. The FLASH is guaranteed to retain data over the entire temperature range for at least the minimum time specified.

Preliminary Electrical Specifications

Section 20. Mechanical Specifications

20.1 Contents

20.2 Introduction

This section provides package information for the MC68HC908RC24.

20.3 Available Packages

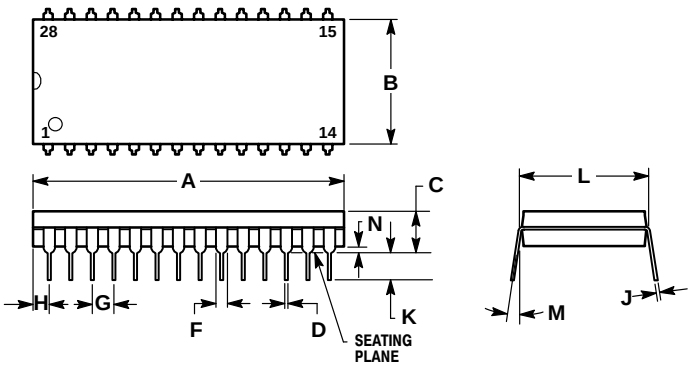
The sections **20.3.1 28-Pin Plastic Dual In-Line Package (Case 710)** and **20.3.2 28-Pin Small Outline Package (Case 751F)** show the latest information at the time of this publication.

To make sure that you have the latest package specifications, contact one of these sources:

- Local Motorola sales office
- Motorola Fax Back System (Mfax™)
  - Phone 1-602-244-6609
  - EMAIL RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/
- Worldwide Web (wwweb) home page at http://www.mcu.motsps.com

Follow Mfax or wwweb on-line instructions to retrieve the current mechanical specifications.

20.3.1 28-Pin Plastic Dual In-Line Package (Case 710)

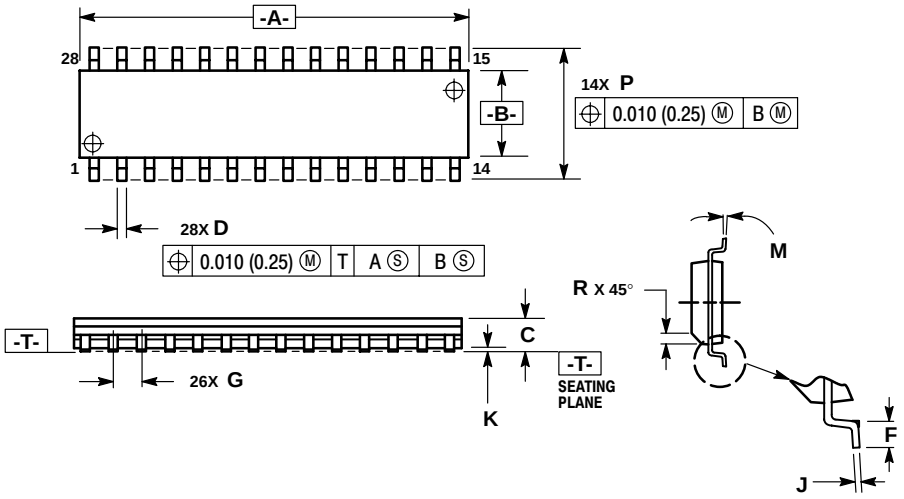


NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

Mfax is a trademark of Motorola, Inc.

20.3.2 28-Pin Small Outline Package (Case 751F)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 (0.005) TOTAL IN EXCESS OF D DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	17.80	18.05	0.701	0.711
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.41	0.90	0.016	0.035
G	1.27 BSC		0.050 BSC	
J	0.23	0.32	0.009	0.013
K	0.13	0.29	0.005	0.011
M	0°	8°	0°	8°
P	10.05	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

Mechanical Specifications

Section 21. Ordering Information

21.1 Contents

21.2 Introduction

This section contains instructions for ordering the MC68HC908RC24.

Packages available are:

- 28-pin plastic dual in-line package (PDIP)
- 28-pin small outline package (SOIC)

21.3 XC Order Numbers

Table 21-1. XC Order Numbers

MC Order Number(1)	Operating Temperature Range
XC68HC908RC24P	–0°C to + 70°C
XC68HC908RC24DW	–0°C to + 70°C

1. P = Dual in-line plastic (PDIP)
DW = Small outline (SOIC)

Ordering Information

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution, P.O. Box 5405, Denver, Colorado 80217, 1-800-441-2447 or 1-303-675-2140. Customer Focus Center, 1-800-521-6274

JAPAN: Motorola Japan Ltd.: SPD, Strategic Planning Office, 141, 4-32-1 Nishi-Gotanda, Shinagawa-ku, Tokyo, Japan, 03-5487-8488

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd., Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate, Tai Po, New Territories, Hong Kong, 852-26668334

Mfax™, Motorola Fax Back System: RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/; TOUCHTONE, 1-602-244-6609; US and Canada ONLY, 1-800-774-1848

HOME PAGE: http://motorola.com/sps/

MOTOROLA