MC68HC908KX8
MC68HC908KX2
MC68HC08KX8

Data Sheet

M68HC08
Microcontrollers

freescale.com

freescale™
semiconductor

MC68HC908KX8
MC68HC908KX2
MC68HC08KX8

Data Sheet

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

http://www.freescale.com

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

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

Revision History

Date	Revision Level	Description	Page Number(s)
April, 2001	0.1	Label for pin 9 corrected in Figure 1-1 and Figure 1-2	19, 20
		$FF is the erase state of the FLASH, not $00.	82, 252, 255
		First bulleted paragraph under the subsection 15.5 Interrupts reworded for clarity	177
		Revision to the description of the CHxMAX bit and the note that follows that description	183
		Forced monitor mode information added to Table 16-1.	192
		In Figure 16-10. Monitor Data Format, resistor value for connection between VTST and IRQ1 changed from 10 kΩ to 1 kΩ.	194
February, 2002	1.0	7.2 Features — Corrected third bullet	71
		7.7.3 ICG Trim Register — Corrected description of the TRIM7:TRIM0 bits	97
		14.2 Features — Corrected divide by factors in first bullet	165
		Figure 14-1. Timebase Block Diagram — Corrected divide-by-2 blocks	166
		Table 14-1. Timebase Divider Selection — Corrected last divider tap entry	167
		Section 15. Timer Interface Module (TIM) — Timer discrepancies corrected throughout this section	169
		17.4 Thermal Characteristics — Corrected SOIC thermal resistance and maximum junction temperature	202
		17.5 5.0-Vdc DC Electrical Characteristics and — Corrected footnote for VDD supply current in stop mode	203 and 204
		Appendix B. MC68HC08KX8 — Added to supply exception information for the MC68HC08KX8	215
March, 2004	2.0	Reformatted to current publication standards	Throughout
		2.7 FLASH Page Erase Operation — Updated procedure	33
		2.8 FLASH Mass Erase Operation — Updated procedure	33
		2.9 FLASH Program/Read Operation — Updated procedure	34
		Figure 5-1. COP Block Diagram — Updated figure	53
		Table 6-1. Instruction Set Summary — Added WAIT instruction	69
		Section 7. Internal Clock Generator Module (ICG) — Updated with new information	71 through 98
		14.2 Features — Corrected values given in the first bullet	165
		Table 15-3. Mode, Edge, and Level Selection — Reworked for clarity	182
		17.11 Memory Characteristics — Updated table with new information	210
July, 2005	2.1	Updated to meet Freescale identity guidelines.	Throughout

List of Chapters

Table of Contents

Chapter 1
General Description

Chapter 2
Memory

Chapter 3
Analog-to-Digital Converter (ADC)

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Table of Contents

Chapter 7
Internal Clock Generator Module (ICG)

Chapter 8
External Interrupt (IRQ)

Chapter 9
Keyboard Interrupt Module (KBI)

Chapter 10
Low-Voltage Inhibit (LVI)

Chapter 11
Input/Output (I/O) Ports (PORTS)

Chapter 12
Serial Communications Interface Module (SCI)

Chapter 13
System Integration Module (SIM)

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Chapter 17
Electrical Specifications

Chapter 18
Ordering Information and Mechanical Specifications

Appendix A
MC68HC908KX2

Appendix B
MC68HC08KX8

Chapter 1
General Description

1.1 Introduction

The MC68HC908KX8 is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCU). The M68HC08 Family is based on the customer-specified integrated circuit (CSIC) design strategy. All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

The information contained is this document pertains to the MC68HC908KX2 and the MC68HC08KX8 with the exceptions found in:

- Appendix A MC68HC908KX2
- Appendix B MC68HC08KX8

1.2 Features

Features include:

- High-performance M68HC08 architecture
- Fully upward-compatible object code with M6805, M146805, and M68HC05 Families
- Maximum internal bus frequencies of:
  - 8 MHz at 5.0 V
  - 4 MHz at 3.0 V
- Internal oscillator requiring no external components:
  - Software selectable bus frequencies
  - 25 percent accuracy with trim capability to 2 percent
  - Clock monitor
  - Option to allow use of external clock source or external crystal/ceramic resonator
- Eight Kbytes of on-chip, in-circuit programmable FLASH memory
- FLASH program memory security[1]
- On-chip programming firmware for use with host personal computer which does not require high voltage for entry
- 192 bytes of on-chip random-access memory (RAM)
- 16-bit, 2-channel timer interface (TIM) module
- 4-channel, 8-bit, analog-to-digital converter (ADC) with high-voltage reference (V_{REFH}) double bonded to V_{DD} pin

---

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

- Serial communications interface (SCI) module
- 5-bit keyboard interrupt (KBI) with wakeup feature
- 13 general-purpose input/output (I/O) ports:
  - Five shared with KBI and TIM, with 15-mA source/15-mA sink capabilities and with programmable pullups on general- purpose input ports
  - Four shared with ADC
  - Two shared with SCI
- Low-voltage inhibit (LVI) module with software selectable trip points, 2.6-V or 4.3-V trip point
- Timebase module (TBM) with
  - Clock prescaler for eight user-selectable, periodic real-time interrupts
  - Active clock source in stop mode for periodic wakeup from stop using external crystal or internal oscillator
- External asynchronous interrupt pin with internal pullup ($\overline{\text{IRQ1}}$)
- System protection features:
  - Computer operating properly (COP) reset
  - Low-voltage detection with reset
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- 16-pin plastic dual in-line (PDIP) or small outline (SOIC) package
- Low-power design fully static with stop and wait modes
- Internal power-up reset circuit requiring no external pins
- -40°C to $+125^{\circ}$C operation

Features of the CPU08 include:

- Enhanced HC05 programming model
- Extensive loop control functions
- 16 addressing modes, eight more than the M68HC05
- 16-bit index register and stack pointer
- Memory-to-memory data transfers
- Fast 8 $\times$ 8 multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Third party C language support

1.3 MCU Block Diagram

Figure 1-1 shows the structure of the MC68HC908KX8.

INTERNAL BUS

M68HC08 CPU

CPU REGISTERS

ARITHMETIC/LOGIC UNIT

CONTROL AND STATUS REGISTERS — 78 BYTES

USER FLASH — 7680 BYTES

USER RAM — 192 BYTES

MONITOR ROM — 295 BYTES

USER FLASH VECTOR SPACE — 36 BYTES

FLASH BURN-IN ROM — 1024 BYTES

INTERNAL CLOCK GENERATOR MODULE (SOFTWARE SELECTABLE)

SYSTEM INTEGRATION MODULE

$\overline{IRQ1}$[1]

IRQ MODULE

V_{DD}

V_{SS}

POWER

POWER-ON RESET MODULE

SECURITY MODULE

COMPUTER OPERATING PROPERLY MODULE

LOW-VOLTAGE INHIBIT MODULE

2-CHANNEL TIMER INTERFACE MODULE

KEYBOARD INTERRUPT MODULE

ANALOG-TO-DIGITAL CONVERTER MODULE

SERIAL COMMUNICATION INTERFACE MODULE

PROGRAMMABLE TIME BASE MODULE

BREAK MODULE

DDRB PTB

PTB0/AD0
PTB1/AD1
PTB2/AD2
PTB3/AD3
PTB4/RxD
PTB5/TxD
PTB6/(OSC1)[4]
PTB7/(OSC2)/$\overline{RST}$[4]

DDRA PTA

PTA0/$\overline{KBD0}$[2], [3]
PTA1/$\overline{KBD1}$[2], [3]
PTA2/$\overline{KBD2}$/TCH0[2], [3]
PTA3/$\overline{KBD3}$/TCH1[2], [3]
PTA4/$\overline{KBD4}$[2], [3]

Notes:
1. Pin contains integrated pullup resistor
2. High-current source/sink pin
3. Pin contains software selectable pullup resistor if general function I/O pin is configured as input.

Figure 1-1. MC68HC908KX8 MCU Block Diagram

1.4 Pin Assignments

Figure 1-2 shows the pin assignments for MC68HC908KX8.
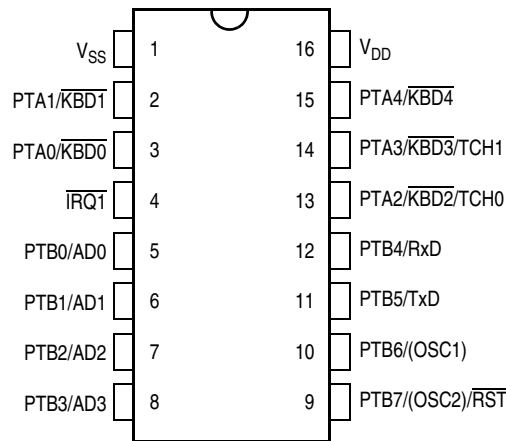
Figure 1-2. PDIP and SOIC Pin Assignments

1.4.1 Supply Pins (V_{DD} and V_{SS})

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

Fast signal transitions on MCU pins place high, short-duration current demands on the power supply. To prevent noise problems, take special care to provide power supply bypassing at the MCU as shown in Figure 1-3. Place the bypass capacitors as close to the MCU power pins as possible. Use high-frequency response ceramic capacitors for C_{Bypass}. C_{Bulk} are optional bulk current bypass capacitors for use in applications that require the port pins to source high-current levels.
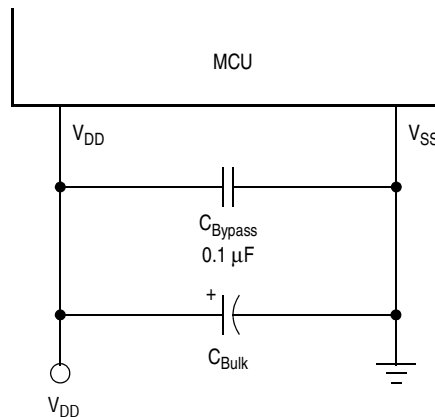
Note: Component values shown represent typical applications.

Figure 1-3. Power Supply Bypassing

1.4.2 Oscillator Pins (OSC1 and OSC2)

The OSC1 and OSC2 pins are available through programming options in the configuration register. These pins then become the connections to an external clock source or crystal/ceramic resonator. PTB7 and PTB6 are not available for the crystal/ceramic resonator option and PTB6 is unavailable for the external clock source option.

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

$\overline{\text{IRQ1}}$ is an asynchronous external interrupt pin with an internal pullup resistor. See Chapter 8 External Interrupt (IRQ).

1.4.4 Port A Input/Output (I/O) Pins (PTA4/$\overline{\text{KBD4}}$–PTA0/$\overline{\text{KBD0}}$)

PTA4/$\overline{\text{KBD4}}$–PTA0/$\overline{\text{KBD0}}$ is a 5-bit special-function port that shares its pins with the keyboard interrupt (KBI) module and the 2-channel timer module (TIM).

- Any or all of the port A pins can be programmed to serve as keyboard interrupt pins. The respective pin utilizes an internal pullup resistor when enabled. See Chapter 9 Keyboard Interrupt Module (KBI).

- Each port A pin contains a software selectable internal pullup resistor when the general-function I/O port is configured as an input. See Chapter 11 Input/Output (I/O) Ports (PORTS). The pullup resistor is automatically disabled once a TIM special function is enabled for that pin.

- All port A pins are high-current source/sink pins.

> **NOTE**
> *Any unused inputs and I/O ports should be tied to an appropriate logic level (either V_{DD} or V_{SS}). Although the I/O ports of the MC68HC908KX8 do not require termination, termination is recommended to reduce the possibility of static damage.*

1.4.5 Analog Reference Pin (V_{REFH})

The V_{REFH} pin is the analog reference voltage for the analog-to-digital converter (ADC) module. The voltage is supplied through a double-bond to the V_{DD} pin. See Chapter 17 Electrical Specifications for ADC parameters.

1.4.6 Port B Input/Output (I/O) Pins (PTB7/(OSC2)/$\overline{\text{RST}}$–PTB0/AD0)

PTB7/(OSC2)/$\overline{\text{RST}}$–PTB0/AD0 are general-purpose bidirectional I/O port pins, all sharing special functions.

- PTB7 and PTB6 share with the on-chip oscillator circuit through configuration options. See 7.3.3 External Clock Generator.

- PTB5 and PTB4 share with the SCI module. See Chapter 12 Serial Communications Interface Module (SCI).

- PTB3–PTB0 share with the ADC module. See Chapter 3 Analog-to-Digital Converter (ADC).

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Freescale Semiconductor · 21

Chapter 2
Memory

2.1 Introduction

The central processor unit (CPU08) can address 64 Kbytes of memory space.

The memory map, shown in Figure 2-1, includes:

- 7680 bytes of FLASH memory
- 192 bytes of random-access memory (RAM)
- 36 bytes of user-defined vectors
- 295 bytes of monitor read-only memory (ROM)

2.2 I/O Registers

Most of the control, status, and data registers are in the zero-page area of $0000–$003F. Additional input/output (I/O) registers have the following addresses:

- $FE01 — SIM reset status register, SRSR
- $FE04 — Interrupt status register 1, INT1
- $FE05 — Interrupt status register 2, INT2
- $FE06 — Interrupt status register 3, INT3
- $FE08 — FLASH control register, FLCR
- $FE09 — Break address register high, BRKH
- $FE0A — Break address register low, BRKL
- $FE0B — Break status and control register, BRKSCR
- $FE0C — LVI status register, LVISR
- $FF7E — FLASH block protect register, FLBPR
          in non-volatile FLASH memory
- $FFFF — COP control register, COPCTL

A summary of the available registers is provided in Figure 2-2. Table 2-1 is a list of vector locations.

2.3 Monitor ROM

The 295 bytes at addresses $FE20–$FF46 are reserved ROM addresses that contain the instructions for the monitor functions.

$0000 ↓ $003F	I/O REGISTERS (64 BYTES)
$0040 ↓ $00FF	RAM (192 BYTES)
$0100 ↓ $0FFF	UNIMPLEMENTED (3839 BYTES)
$1000 ↓ $13FF	FLASH BURN-IN ROM (1024 BYTES)
$1400 ↓ $DFFF	UNIMPLEMENTED (52,224 BYTES)
$E000 ↓ $FDFF	USER FLASH MEMORY (7680 BYTES)

$FE00	RESERVED
$FE01	SIM RESET STATUS REGISTER (SRSR)
$FE02	RESERVED
$FE03	RESERVED
$FE04	INTERRUPT STATUS REGISTER 1 (INT1)
$FE05	INTERRUPT STATUS REGISTER 2 (INT2)
$FE06	INTERRUPT STATUS REGISTER 3 (INT3)
$FE07	RESERVED
$FE08	FLASH CONTROL REGISTER (FLCR)
$FE09	BREAK ADDRESS REGISTER HIGH (BRKH)
$FE0A	BREAK ADDRESS REGISTER LOW (BRKL)
$FE0B	BREAK STATUS AND CONTROL REGISTER (BRKSCR)
$FE0C	LVI STATUS REGISTER (LVISR)
$FE0D ↓ $FE1F	UNIMPLEMENTED (18 BYTES)
$FE20 ↓ $FF46	MONITOR ROM (295 BYTES)
$FF47 ↓ $FF7D	UNIMPLEMENTED (57 BYTES)
$FF7E	FLASH BLOCK PROTECT REGISTER (FLBPR)
$FF7F ↓ $FFDB	UNIMPLEMENTED (90 BYTES)
$FFDC ↓ $FFFF	FLASH VECTORS (36 BYTES)

Figure 2-1. Memory Map

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA) See page 106.	Read:	0	0	0	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:				Unaffected by reset				
$0001	Port B Data Register (PTB) See page 108.	Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:				Unaffected by reset				
$0002	Unimplemented									
$0003	Unimplemented									
$0004	Data Direction Register A (DDRA) See page 106.	Read:	0	0	0	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB) See page 109.	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0006 ↓ $000C	Unimplemented									
$000D	Port A Input Pullup Enable Register (PTAPUE) See page 108.	Read:	0	0	0	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000E ↓ $0012	Unimplemented									
$0013	SCI Control Register 1 (SCC1) See page 125.	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0014	SCI Control Register 2 (SCC2) See page 127.	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0015	SCI Control Register 3 (SCC3) See page 129.	Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
		Write:								
		Reset:	U	U	0	0	0	0	0	0
$0016	SCI Status Register 1 (SCS1) See page 130.	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:								
		Reset:	1	1	0	0	0	0	0	0

	= Unimplemented		R	= Reserved	U = Unaffected

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

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0017	SCI Status Register 2 (SCS2) See page 132.	Read:	0	0	0	0	0	0	BKF	RPF
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0018	SCI Data Register (SCDR) See page 133.	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:				Unaffected by reset				
$0019	SCI Baud Rate Register (SCBR) See page 133.	Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001A	Keyboard Status and Control Register (KBSCR) See page 99.	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$001B	Keyboard Interrupt Enable Register (KBIER) See page 100.	Read:	0	0	0	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001C	Timebase Control Register (TBCR) See page 154.	Read:	TBIF	TBR2	TBR1	TBR0	0	TBIE	TBON	R
		Write:					TACK			
		Reset:	0	0	0	0	0	0	0	0
$001D	IRQ Status and Control Register (ISCR) See page 94.	Read:	0	0	0	0	IRQF1	0	IMASK1	MODE1
		Write:	R	R	R	R	R	ACK1		
		Reset:	0	0	0	0	0	0	0	0
$001E	Configuration Register 2[1] (CONFIG2) See page 48.	Read:	R	0	EXT-XTALEN	EXT-SLOW	EXT-CLKEN	0	OSCENIN-STOP	SCIBDSRC
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001F	Configuration Register 1[1] (CONFIG1) See page 47.	Read:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3	SSREC	STOP	COPD
		Write:								
		POR Reset:	0	0	0	0	0	0	0	0
		Other Resets:	0	0	0	0	U	0	0	0

1. LVI5OR3 is only writable after a power-on reset (POR). Bit 6 of CONFIG1 is read-only and will read 0.
   All other bits in CONFIG1 and CONFIG2 are one-time writable after any reset.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0020	Timer Status and Control Register (TSC) See page 163.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0021	Timer Counter Register High (TCNTH) See page 164.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0

☐ = Unimplemented   R = Reserved   U = Unaffected

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
$0022	Timer Counter Register Low (TCNTL) See page 164.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0023	Timer Counter Modulo Register High (TMODH) See page 165.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0024	Timer Counter Modulo Register Low (TMODL) See page 165.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0025	Timer Channel 0 Status and Control Register (TSC0) See page 165.	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0026	Timer Channel 0 Register High (TCH0H) See page 168.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0027	Timer Channel 0 Register Low (TCH0L) See page 168.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0028	Timer Channel 1 Status and Control Register (TSC1) See page 165.	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0029	Timer Channel 1 Register High (TCH1H) See page 168.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$002A	Timer Channel 1 Register Low (TCH1L) See page 168.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$002B ↓ $0035	Unimplemented									
$0036	ICG Control Register (ICGCR) See page 87.	Read:	CMIE	CMF / 0[1]	CMON	CS	ICGON	ICGS	ECGON	ECGS
		Write:								
		Reset:	0	0	0	0	1	0	0	0

1. See 7.7.1 ICG Control Register for method of clearing the CMF bit.

$0037	ICG Multiplier Register (ICGMR) See page 88.	Read:		N6	N5	N4	N3	N2	N1	N0
		Write:								
		Reset:	0	0	0	1	0	1	0	1

☐ = Unimplemented   R = Reserved   U = Unaffected

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

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0038	ICG Trim Register (ICGTR) See page 89.	Read:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
		Write:								
		Reset:	1	0	0	0	0	0	0	0
$0039	ICG Divider Control Register (ICGDVR) See page 89.	Read:					DDIV3	DDIV2	DDIV1	DDIV0
		Write:								
		Reset:	0	0	0	0	U	U	U	U
$003A	ICG DCO Stage Control Register (ICGDSR) See page 89.	Read:	DSTG7	DSTG6	DSTG5	DSTG4	DSTG3	DSTG2	DSTG1	DSTG0
		Write:	R	R	R	R	R	R	R	R
		Reset:	U	U	U	U	U	U	U	U
$003B	Reserved		R	R	R	R	R	R	R	R
$003C	Analog-to-Digital Status and Control Register (ADSCR) See page 43.	Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
		Write:	R							
		Reset:	0	0	0	1	1	1	1	1
$003D	Analog-to-Digital Data Register (ADR) See page 45.	Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		Write:	R	R	R	R	R	R	R	R
		Reset:				Indeterminate after reset				
$003E	Analog-to-Digital Input Clock Register (ADCLK) See page 45.	Read:	ADIV2	ADIV1	ADIV0	ADICLK	0	0	0	R
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$003F	Unimplemented									
$FE00	SIM Break Status Register (SBSR)[1] See page 172.	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

1. Writing a 0 clears BW.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE01	SIM Reset Status Register (SRSR) See page 148.	Read:	POR	0	COP	ILOP	ILAD	MENRST	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
$FE02	Break Auxiliary Register (BRKAR) See page 173.	Read:	0	0	0	0	0	0	0	BDCOP
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE03	SIM Break Flag Control Register (SBFCR) See page 173.	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE04	Interrupt Status Register 1 (INT1) See page 149.	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

| = Unimplemented | R = Reserved | U = Unaffected |

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 5)
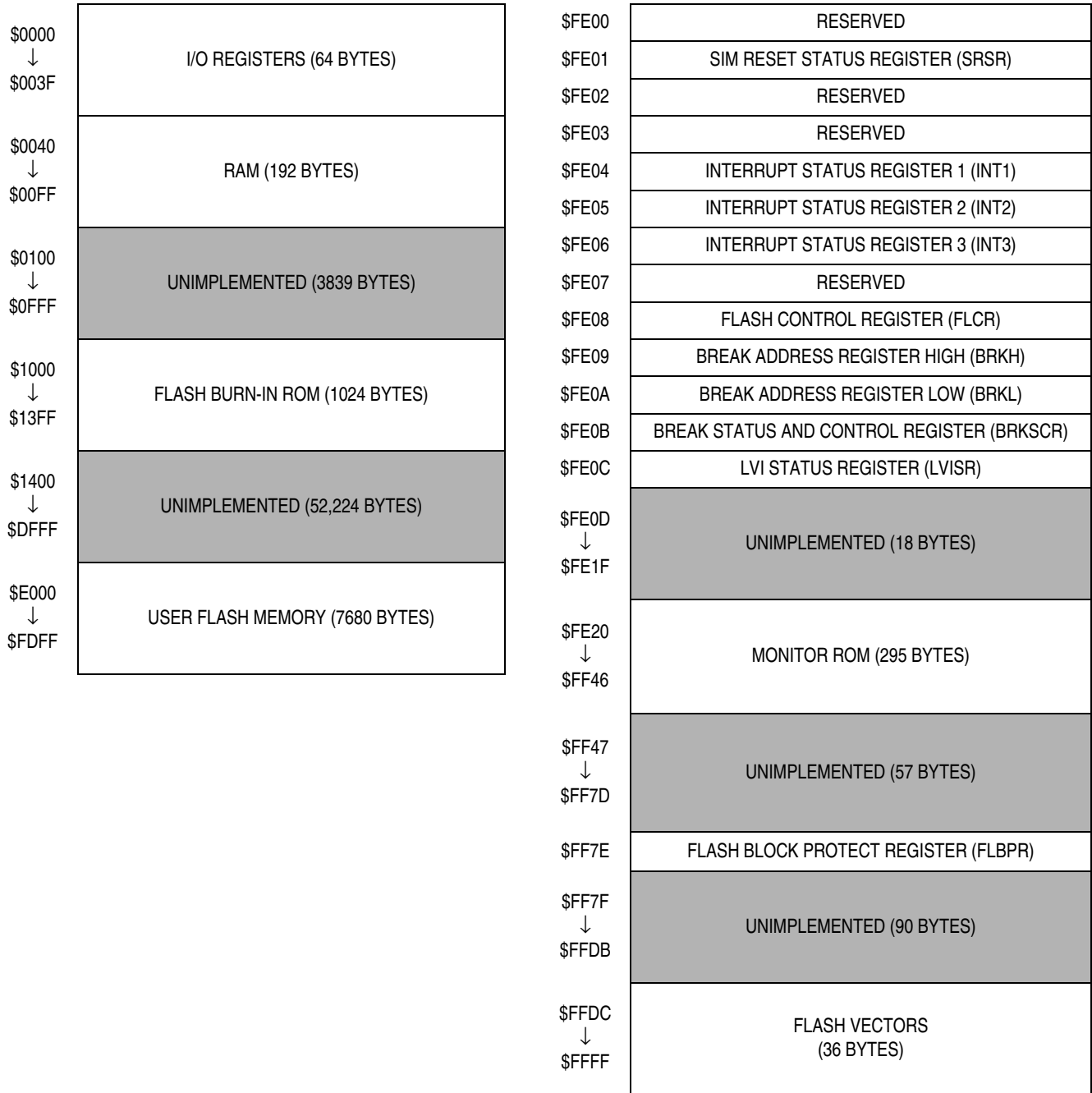
Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE05	Interrupt Status Register 2 (INT2) See page 150.	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE06	Interrupt Status Register 3 (INT3) See page 150.	Read:	IF22	IF21	IF20	IF19	IF18	IF17	IF16	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE07	FLASH Test Control Register (FLTCR)	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$FE08	FLASH Control Register (FLCR) See page 31.	Read:	0	0	0	0	HVEN	MARGIN	ERASE	PGM
		Write:					HVEN	MARGIN	ERASE	PGM
		Reset:	0	0	0	0	0	0	0	0
$FE09	Break Address Register High (BRKH) See page 172.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	0	0	0	0	0	0	0	0
$FE0A	Break Address Register Low (BRKL) See page 172.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	0	0	0	0	0	0	0	0
$FE0B	Break Status and Control Register (BRKSCR) See page 171.	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:	BRKE	BRKA						
		Reset:	0	0	0	0	0	0	0	0
$FE0C	LVI Status Register (LVISR) See page 103.	Read:	LVIOUT	0	0	0	0	0	0	R
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FF7E	FLASH Block Protect Register (FLBPR)[1] See page 36.	Read:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Write:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Reset:				Unaffected by reset				

1. Non-volatile FLASH register

$FFFF	COP Control Register (COPCTL) See page 53.	Read:	Low byte of reset vector							
		Write:	Writing clears COP counter (any value)							
		Reset:	Unaffected by reset							

	= Unimplemented	R	= Reserved	U = Unaffected

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

Table 2-1. Vector Locations

	Address	Vector
Low	$FFDC	Timebase module vector (high)
	$FFDD	Timebase module vector (low)
	$FFDE	ADC conversion complete vector (high)
	$FFDF	ADC conversion complete vector (low)
	$FFE0	Keyboard vector (high)
	$FFE1	Keyboard vector (low)
	$FFE2	SCI transmit vector (high)
	$FFE3	SCI transmit vector (low)
	$FFE4	SCI receive vector (high)
	$FFE5	SCI receive vector (low)
	$FFE6	SCI receive error vector (high)
	$FFE7	SCI receive error vector (low)
	$FFE8	Reserved
	$FFE9	Reserved
	$FFEA	Reserved
Priority	$FFEB	Reserved
	$FFEC	Reserved
	$FFED	Reserved
	$FFEE	Reserved
	$FFEF	Reserved
	$FFF0	Reserved
	$FFF1	Reserved
	$FFF2	TIM overflow vector (high)
	$FFF3	TIM overflow vector (low)
	$FFF4	TIM channel 1 vector (high)
	$FFF5	TIM channel 1 vector (low)
	$FFF6	TIM channel 0 vector (high)
	$FFF7	TIM channel 0 vector (low)
	$FFF8	CMIREQ vector (high)
	$FFF9	CMIREQ vector (low)
	$FFFA	IRQ1 vector (high)
	$FFFB	IRQ1 vector (low)
	$FFFC	SWI vector (high)
	$FFFD	SWI vector (low)
High	$FFFE	Reset vector (high)
	$FFFF	Reset vector (low)

2.4 Random-Access Memory (RAM)

Addresses $0040–$00FF are RAM locations. The location of the stack RAM is programmable. The 16-bit stack pointer allows the stack to be anywhere in the 64-Kbyte memory space.

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

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

> ***NOTE***
> *For M6805, M146805 and M68HC05compatibility, 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 could overwrite data in the RAM during a subroutine or during the interrupt stacking operation.*

2.5 FLASH Memory (FLASH)

The FLASH memory is an array of 7,680 bytes with an additional 36 bytes of user vectors and one byte used for block protection.

> ***NOTE***
> *An erased bit reads as 1 and a programmed bit reads as 0.*

The program and erase operations are facilitated through control bits in the FLASH control register (FLCR). See 2.6 FLASH Control Register.

The FLASH is organized internally as an 8192-word by 8-bit complementary metal-oxide semiconductor (CMOS) page erase, byte (8-bit) program embedded FLASH memory. Each page consists of 64 bytes. The page erase operation erases all words within a page. A page is composed of two adjacent rows.

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

2.6 FLASH Control Register

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

Address: $FE08

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

= Unimplemented

Figure 2-3. FLASH Control Register (FLCR)

---

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

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

HVEN — High-Voltage Enable Bit

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

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

MASS — Mass Erase Control Bit

Setting this read/write bit configures the 8-Kbyte FLASH array for mass erase operation.

    1 = MASS erase operation selected
    0 = MASS erase operation unselected

ERASE — Erase Control Bit

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

    1 = Erase operation selected
    0 = Erase operation unselected

PGM — Program Control Bit

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

    1 = Program operation selected
    0 = Program operation unselected

2.7 FLASH Page Erase Operation

Use this step-by-step procedure to erase a page (64 bytes) of FLASH memory to read as 1:

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

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

2.8 FLASH Mass Erase Operation

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

1. Set both the ERASE bit and the MASS bit in the FLASH control register.
2. Read the FLASH block protect register.
3. Write any data to any FLASH address[1] within the FLASH memory address range.
4. Wait for a time, t_{NVS} (minimum 10 μs).
5. Set the HVEN bit.
6. Wait for a time, t_{MErase} (minimum 4 ms).
7. Clear the ERASE and MASS bits.

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

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

> ### *NOTE*
> *Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order shown, other unrelated operations may occur between the steps.*

---

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

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

2.9 FLASH Program/Read Operation

Programming of the FLASH memory is done on a row basis. A row consists of 32 consecutive bytes starting from addresses $XX00, $XX20, $XX40, $XX60, $XX80, $XXA0, $XXC0, and $XXE0. Use this step-by-step procedure to program a row of FLASH memory (Figure 2-4 is a flowchart representation).

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

1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
2. Read the FLASH block protect register.
3. Write any data to any FLASH location within the address range desired.
4. Wait for a time, t_{NVS} (minimum 10 μs).
5. Set the HVEN bit.
6. Wait for a time, t_{PGS} (minimum 5 μs).
7. Write data to the FLASH address being programmed[1].
8. Wait for time, t_{PROG} (minimum 30 μs).
9. Repeat step 7 and 8 until all desired bytes within the row are programmed.
10. Clear the PGM bit[1].
11. Wait for time, t_{NVH} (minimum 5 μs).
12. Clear the HVEN bit.
13. After time, t_{RCV} (typical 1 μs), the memory can be accessed in read mode again.

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

> ***NOTE***
> *Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order shown, other unrelated operations may occur between the steps. Do not exceed t_{PROG} maximum. See 17.11 Memory Characteristics.*

---

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

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

**Algorithm for programming
a row (32 bytes) of FLASH memory**

1 | SET PGM BIT

2 | READ THE FLASH BLOCK PROTECT REGISTER

3 | WRITE ANY DATA TO ANY FLASH ADDRESS WITHIN THE ROW ADDRESS RANGE DESIRED

4 | WAIT FOR A TIME, t_{NVS}

5 | SET HVEN BIT

6 | WAIT FOR A TIME, t_{PGS}

7 | WRITE DATA TO THE FLASH ADDRESS TO BE PROGRAMMED

8 | WAIT FOR A TIME, t_{PROG}

COMPLETED PROGRAMMING THIS ROW?   YES / NO

10 | CLEAR PGM BIT

11 | WAIT FOR A TIME, t_{NVH}

12 | CLEAR HVEN BIT

13 | WAIT FOR A TIME, t_{RCV}

END OF PROGRAMMING

Notes:

The time between each FLASH address change (step 7 to step 7),
or the time between the last FLASH address programmed
to clearing PGM bit (step 7 to step 10)
must not exceed the maximum programming
time, t_{PROG} maximum.

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

Figure 2-4. FLASH Programming Flowchart

2.10 FLASH Block Protection

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

NOTE

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

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

When bits within the FLBPR are programmed, they lock a block of memory address ranges as shown in 2.11 FLASH Block Protect Register. Once the FLBPR is programmed with a value other than $FF, any erase or program of the FLBPR or the protected block of FLASH memory is prohibited. The FLBPR itself can be erased or programmed only with an external voltage, V_{TST}, present on the $\overline{IRQ}$ pin. This voltage also allows entry from reset into the monitor mode.

2.11 FLASH Block Protect Register

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

Address:	$FF7E							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
Reset:	U	U	U	U	U	U	U	U

U = Unaffected by reset. Initial value from factory is 1.
Write to this register is by a programming sequence to the FLASH memory.
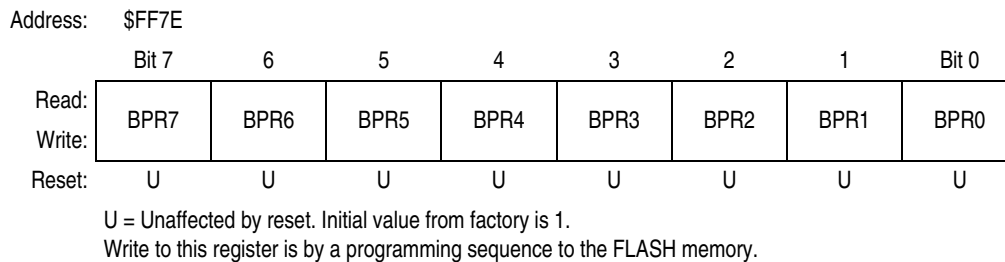
Figure 2-5. FLASH Block Protect Register (FLBPR)

BPR7–BPR0 — FLASH Block Protect Bits

These eight bits represent bits 13–6 of a 16-bit memory address. Bits 15 and 14 are 1s and bits 5–0 are 0s.

The resultant 16-bit address is used for specifying the start address of the FLASH memory for block protection. The FLASH is protected from this start address to the end of FLASH memory, at $FFFF. With this mechanism, the protect start address can be $XX00, $XX40, etc., (64 bytes page boundaries) within the FLASH memory. See Figure 2-6 and Table 2-2.

16-BIT MEMORY ADDRESS

START ADDRESS OF FLASH
BLOCK PROTECT

| 1 | 1 | FLBPR VALUE | 0 | 0 | 0 | 0 | 0 | 0 |
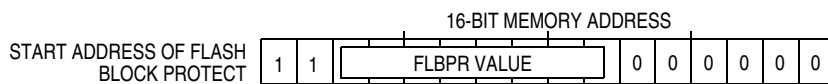
Figure 2-6. FLASH Block Protect Start Address

Table 2-2. Protect Start Address Examples

BPR7–BPR0	Start of Address of Protect Range[1]
$80	The entire FLASH memory is protected.
$81 (**1000 0001**)	$E040 (111**0 0000 01**00 0000)
$82 (**1000 0010**)	$E080 (111**0 0000 10**00 0000)
and so on...	
$FE (**1111 1110**)	$FF80 (111**1 1111 10**00 0000)
$FF	The entire FLASH memory is not protected.

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

2.12 Wait Mode

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

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

2.13 Stop Mode

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

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

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

Chapter 3
Analog-to-Digital Converter (ADC)

3.1 Introduction

This section describes the 8-bit analog-to-digital converter (ADC).

3.2 Features

Features of the ADC module include:
- Four channels with multiplexed input
- Linear successive approximation
- 8-bit resolution
- Single or continuous conversion
- Conversion complete flag or conversion complete interrupt
- Selectable ADC clock

3.3 Functional Description

The ADC provides four pins for sampling external sources at pins PTB3–PTB0. An analog multiplexer allows the single ADC converter to select one of four ADC channels as ADC voltage in (ADCVIN). ADCVIN is converted by the successive approximation register-based counters. When the conversion is completed, ADC places the result in the ADC data register and sets a flag or generates an interrupt. See Figure 3-2.

The MC68HC908KX8 uses V_{DD} as the high voltage reference.

3.3.1 ADC Port I/O Pins

PTB3–PTB0 are general-purpose input/output (I/O) pins that are shared with the ADC channels.

The channel select bits define which ADC channel/port pin will be used as the input signal. The ADC overrides the port I/O logic by forcing that pin as input to the ADC. The remaining ADC channels/port pins are controlled by the port I/O logic and can be used as general-purpose I/O. Writes to the port register or DDR will not have any effect on the port pin that is selected by the ADC. Read of a port pin which is in use by the ADC will return a logic 0 if the corresponding DDR bit is at 0. If the DDR bit is at 1, the value in the port data latch is read.
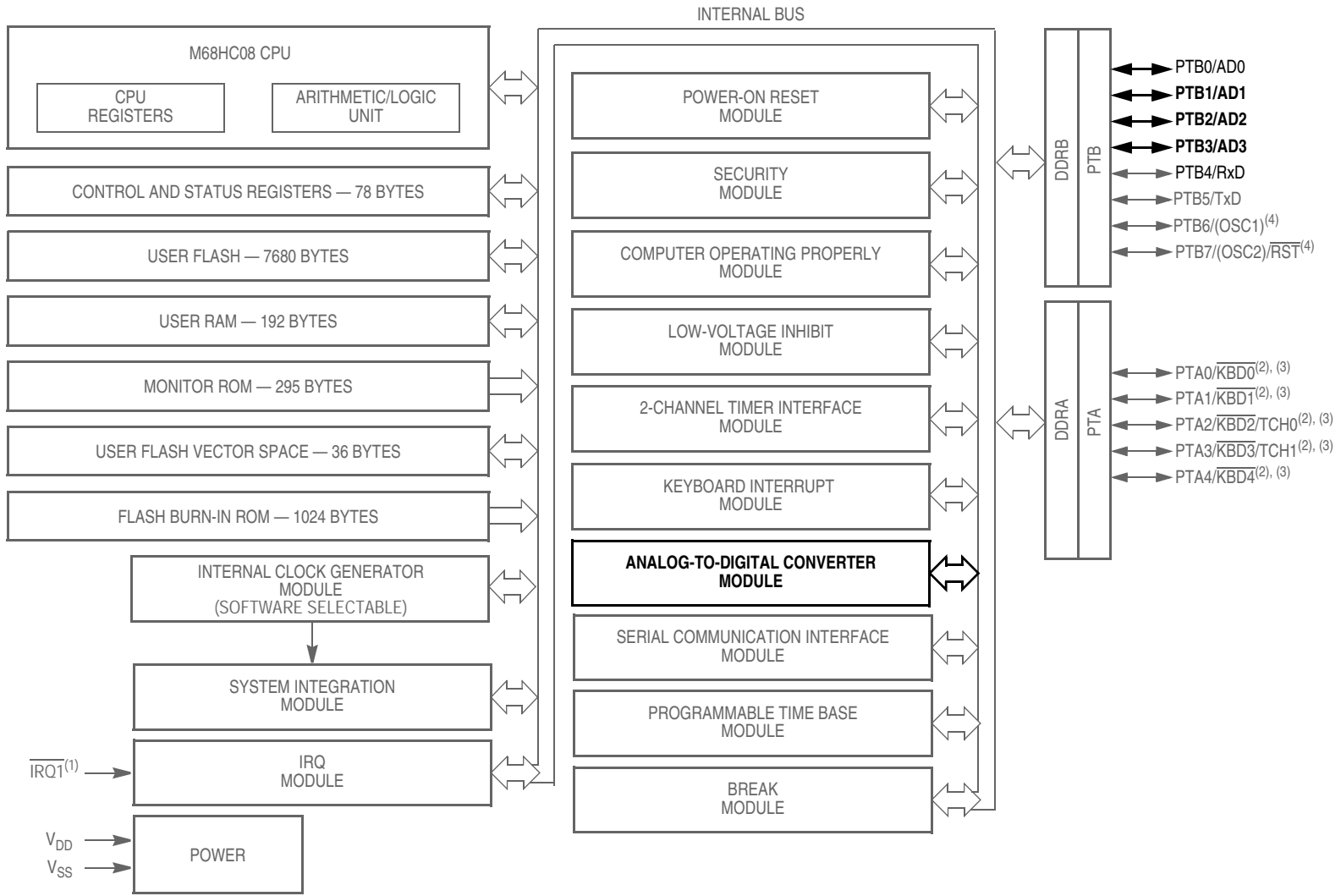
Figure 3-1. Block Diagram Highlighting ADC Block and Pins

Notes:
1. Pin contains integrated pullup resistor
2. High-current source/sink pin
3. Pin contains software selectable pullup resistor if general function I/O pin is configured as input.
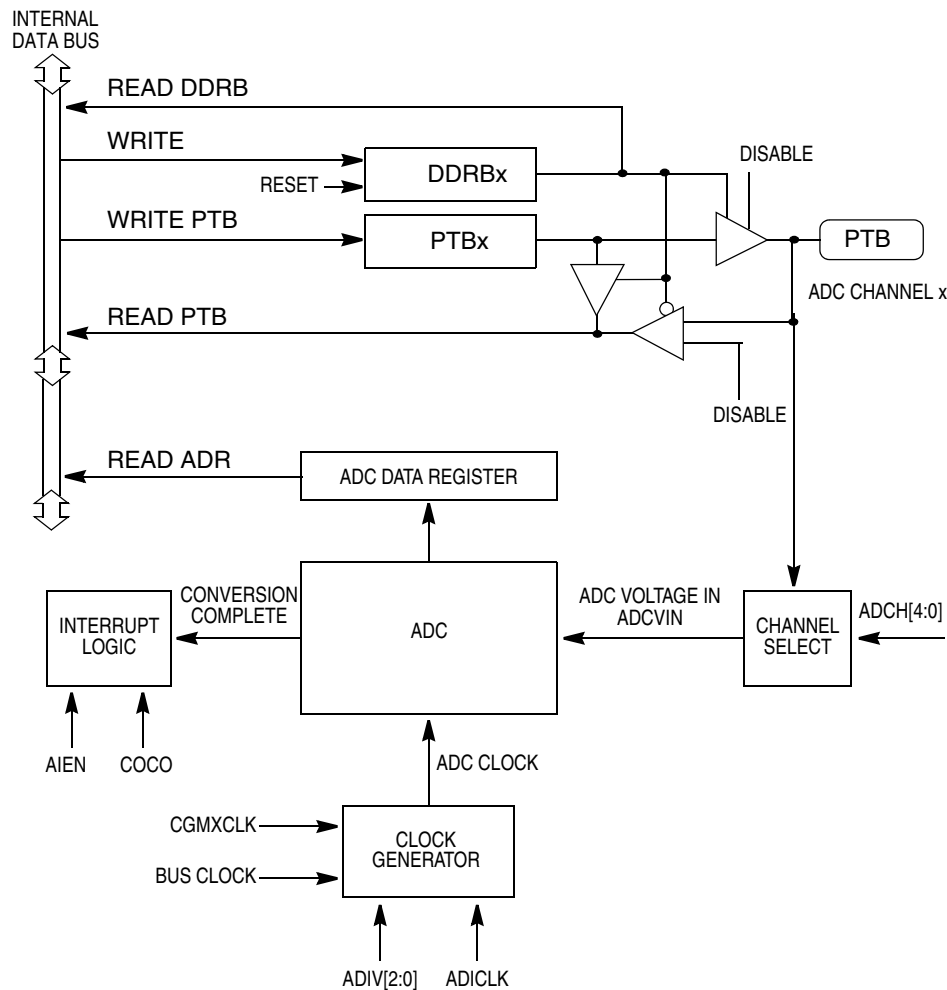
Figure 3-2. ADC Block Diagram

3.3.2 Voltage Conversion

When the input voltage to the ADC equals V_{REFH} (see 17.9 Trimmed Accuracy of the Internal Clock Generator), the ADC converts the signal to $FF (full scale). If the input voltage equals V_{SS}, the ADC converts it to $00. Input voltages between V_{REFH} and V_{SS} are a straight-line linear conversion. All other input voltages will result in $FF if greater than V_{REFH} and $00 if less than V_{SS}.

> *NOTE*
> *Input voltage should not exceed the high-voltage reference, which in turn should not exceed supply voltages.*

3.3.3 Conversion Time

Conversion starts after a write to the ADSCR (ADC status control register, $003C) and requires between 16 and 17 ADC clock cycles to complete. Conversion time in terms of the number of bus cycles is a function of CGMXCLK frequency, bus frequency, the ADIV prescaler bits, and the ADICLK bit. For example, with a CGMXCLK frequency of 8 MHz, bus frequency of 2 MHz, and fixed ADC clock frequency of 1 MHz, one conversion will take between 16 and 17 µs and there will be 32 bus cycles between each conversion. Sample rate is approximately 60 kHz.

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Refer to 17.9 Trimmed Accuracy of the Internal Clock Generator.

$$\text{Conversion time} = \frac{\text{16 to 17 ADC clock cycles}}{\text{ADC clock frequency}}$$

Number of bus cycles = conversion time x bus frequency

3.3.4 Continuous Conversion

In continuous conversion mode, the ADC data register will be filled with new data after each conversion. Data from the previous conversion will be overwritten whether that data has been read or not. Conversions will continue until the ADCO bit (ADC status control register, $003C) is cleared. The COCO bit is set after the first conversion and will stay set until the next write of the ADC status and control register or the next read of the ADC data register.

3.3.5 Accuracy and Precision

The conversion process is monotonic and has no missing codes. See 17.9 Trimmed Accuracy of the Internal Clock Generator for accuracy information.

3.4 Interrupts

When the AIEN bit is set, the ADC module is capable of generating a CPU interrupt after each ADC conversion. A CPU interrupt is generated if the COCO bit (ADC status control register, $003C) is at 0. If the COCO bit is set, a direct-memory access (DMA) interrupt is generated.

> ***NOTE***
> *Because the* MC68HC908KX8 *does not have a DMA module, the COCO bit should not be set while interrupts are enabled (AIEN = 1).*

The COCO bit is not used as a conversion complete flag when interrupts are enabled.

3.5 Low-Power Modes

The following subsections describe the low-power modes.

3.5.1 Wait Mode

The ADC continues normal operation during wait mode. Any enabled CPU interrupt request from the ADC can bring the MCU out of wait mode. If the ADC is not required to bring the MCU out of wait mode, power down the ADC by setting the ADCH[4:0] bits in the ADC status and control register before executing the WAIT instruction.

3.5.2 Stop Mode

The ADC module is inactive after the execution of a STOP instruction. Any pending conversion is aborted. ADC conversions resume when the MCU exits stop mode. Allow one conversion cycle to stabilize the analog circuitry before attempting a new ADC conversion after exiting stop mode.

3.6 I/O Signals

The ADC module has four channels that are shared with port B pins. Refer to 17.9 Trimmed Accuracy of the Internal Clock Generator for voltages referenced here.

3.6.1 ADC Analog Power and ADC Voltage Reference Pins

The ADC analog portion uses V_{DD} as its power pin and V_{SS} as its ground pin.

Due to pin limitations, the V_{REFL} signal is internally connected to V_{SS} on the MC68HC908KX8. On the MC68HC908KX8, the V_{REFH} signal is internally connected to V_{DD}.

3.6.2 ADC Voltage In (ADCVIN)

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

3.7 I/O Registers

These I/O registers control and monitor ADC operation:
- ADC status and control register, ADSCR
- ADC data register, ADR
- ADC clock register, ADICLK

3.7.1 ADC Status and Control Register

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

Address: $003C

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

R		= Reserved

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

COCO — Conversions Complete Bit
When the AIEN bit is a 0, the COCO is a read-only bit which is set each time a conversion is completed. This bit is cleared whenever the ADC status and control register is written or whenever the ADC data register is read.

When the AIEN bit is a 1, the ADC module is capable of generating a CPU interrupt after each ADC conversion. A CPU interrupt is generated if the COCO bit (ADC status control register, $003C) is at 0. If the COCO bit is at 1, a DMA interrupt is generated. Reset clears this bit.
  1 = Conversion completed (AIEN = 0)
  0 = Conversion not completed (AIEN = 0)
      or CPU interrupts enabled (AIEN = 1)

NOTE
Because the MC68HC908KX8 does not have a DMA module, the COCO bit should not be set while interrupts are enabled (AIEN = 1).

AIEN — ADC Interrupt Enable Bit

When this bit is set, an interrupt is generated at the end of an ADC conversion. The interrupt signal is cleared when the ADR register is read or the ADSCR register is written. Reset clears the AIEN bit.

    1 = ADC interrupt enabled

    0 = ADC interrupt disabled

ADCO — ADC Continuous Conversion Bit

When set, the ADC will convert samples continuously and update the ADR register at the end of each conversion. Only one conversion is allowed when this bit is cleared. Reset clears the ADCO bit.

    1 = Continuous ADC conversion

    0 = One ADC conversion

ADCH4–ADCH0 — ADC Channel Select Bits

ADCH4–ADCH0 form a 5-bit field which is used to select the input for the A/D measurement. The choices are one of four ADC channels, as well as V_{REFH} and V_{SS}. Input selection is detailed in Table 3-1. Care should be taken when using a port pin as both an analog and a digital input simultaneously to prevent switching noise from corrupting the analog signal.

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

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

Table 3-1. Mux Channel Select

ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	Input Select
0	0	0	0	0	PTB0
0	0	0	0	1	PTB1
0	0	0	1	0	PTB2
0	0	0	1	1	PTB3
0	0	1	0	0	Unused[1]
—	—	—	—	—	—
1	1	1	0	0	Unused [1]
1	1	1	0	1	V_{REFH}[2]
1	1	1	1	0	V_{SSAD} [2]
1	1	1	1	1	ADC power off

1. If any unused channels are selected, the resulting ADC conversion will be unknown.
2. The voltage levels supplied from internal reference nodes as specified in the table are used to verify the operation of the ADC converter both in production test and for user applications.

3.7.2 ADC Data Register

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

Address: $003D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Write:	R	R	R	R	R	R	R	R
Reset:				Indeterminate after reset				

R	= Reserved
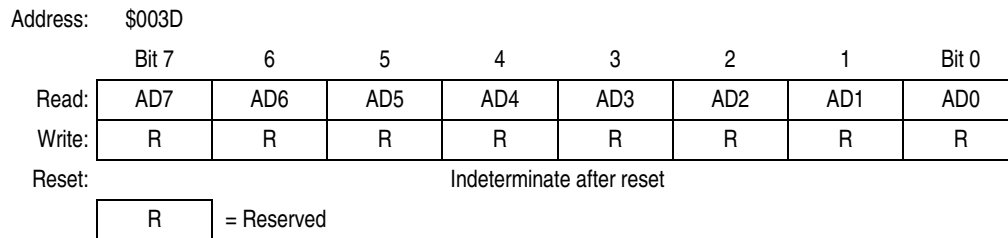
Figure 3-4. ADC Data Register (ADR)

3.7.3 ADC Input Clock Register

This register selects the clock frequency for the ADC.

Address: $003E

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

	= Unimplemented		R	= Reserved
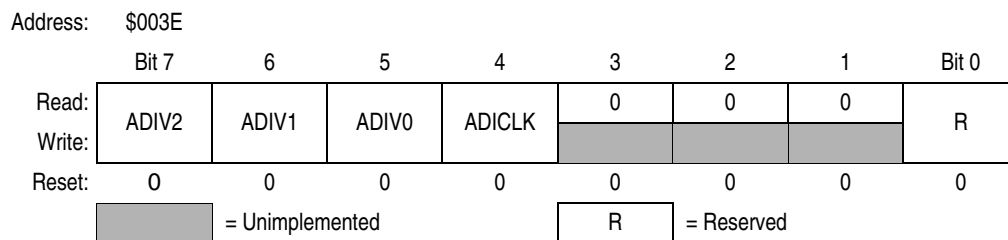
Figure 3-5. ADC Input Clock Register (ADICLK)

ADIV2–ADIV0 — ADC Clock Prescaler Bits
    ADIV2, ADIV1, and ADIV0 form a 3-bit field which selects the divide ratio used by the ADC to generate the internal ADC clock. Table 3-2 shows the available clock configurations. The ADC clock should be set to approximately 1 MHz.

Table 3-2. ADC Clock Divide Ratio

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

X = don't care

ADICLK — ADC Input Clock Select Bit
    ADICLK selects either bus clock or the oscillator output clock (CGMXCLK) as the input clock source to generate the internal ADC rate clock. Reset selects CGMXCLK as the ADC clock source.
        1 = Internal bus clock
        0 = Oscillator output clock (CGMXCLK)

The ADC requires a clock rate of approximately 1 MHz for correct operation. If the selected clock source is not fast enough, the ADC will generate incorrect conversions. See 17.9 Trimmed Accuracy of the Internal Clock Generator.

$$f_{ADIC} = \frac{f_{CGMXCLK} \text{ or bus frequency}}{ADIV[2:0]} \cong 1 \text{ MHz}$$

> ***NOTE***
> *During the conversion process, changing the ADC clock will result in an incorrect conversion.*

Chapter 4
Configuration Register (CONFIG)

4.1 Introduction

This section describes the configuration registers, CONFIG1 and CONFIG2. The configuration registers control these options:

- Stop mode recovery time, 32 CGMXCLK cycles or 4096 CGMXCLK cycles
- Computer operating properly (COP) timeout period, 2^{18}–2^4 or 2^{13}–2^4 CGMXCLK cycles
- STOP instruction
- Computer operating properly (COP) module
- Low-voltage inhibit (LVI) module control and voltage trip point selection
- Enable/disable the oscillator (OSC) during stop mode
- Serial communications interface (SCI) clock source selection
- External clock/crystal source control
- Enable/disable for the FLASH charge-pump regulator

4.2 Functional Description

The configuration registers are used in the initialization of various options and 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 microcontroller unit (MCU), it is recommended that these registers be written immediately after reset. The configuration registers are located at $001E and $001F. For compatibility, a write to a read-only memory (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. On a ROM device, the options are fixed at the time of device fabrication and are neither writable nor changeable by the user.*
>
> *On a FLASH device, the CONFIG registers are special registers containing one-time writable latches after each reset. Upon a reset, the CONFIG registers default to predetermined settings as shown in Figure 4-1 and Figure 4-2.*

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	0	EXT-XTALEN	EXT-SLOW	EXT-CLKEN	0	OSCENIN-STOP	SCIBD-SRC
Write:								
Reset:	0	0	0	0	0	0	0	0

[ ] = Unimplemented   R = Reserved
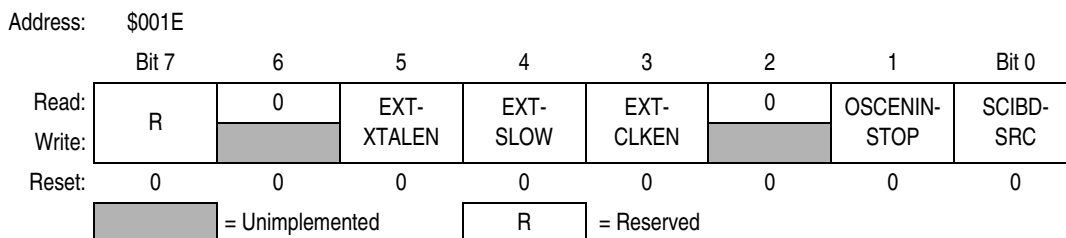
Figure 4-1. Configuration Register 2 (CONFIG2)

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3[1]	SSREC	STOP	COPD
Reset:	0	0	0	0	0	0	0	0
Other Resets:	0	0	0	0	U	0	0	0

1. The LVI5OR3 bit is cleared only by a power-on reset (POR).

U = Unaffected

Figure 4-2. Configuration Register 1 (CONFIG1)

EXTCLKEN — External Clock Enable Bit

EXTCLKEN enables an external clock source or crystal/ceramic resonator to be used as a clock input. Setting this bit enables PTB6/(OSC1) pin to be a clock input pin. Clearing this bit (default setting) allows the PTB6/(OSC1) and PTB7/(OSC2)/$\overline{\text{RST}}$ pins to function as a general-purpose input/output (I/O) pin. Refer to Table 4-1 for configuration options for the external source. See Chapter 7 Internal Clock Generator Module (ICG) for a more detailed description of the external clock operation.

    1 = Allows PTB6/(OSC1) to be an external clock connection
    0 = PTB6/(OSC1) and PTB7/(OSC2)/$\overline{\text{RST}}$ function as I/O port pins (default).

Table 4-1. External Clock Option Settings

External Clock Configuration Bits		Pin Function		Description
EXTCLKEN	EXTXTALEN	PTB6/(OSC1)	PTB7/(OSC2)/$\overline{\text{RST}}$	
0	0	PTB6	PTB7	Default setting — external oscillator disabled
0	1	PTB6	PTB7	External oscillator disabled since EXTCLKEN not set
1	0	OSC1	PTB7	External oscillator configured for an external clock source input (square wave) on OSC1
1	1	OSC1	OSC2	External oscillator configured for an external crystal configuration on OSC1 and OSC2. System will also operate with square-wave clock source in OSC1.

EXTSLOW — Slow External Crystal Enable Bit

The EXTSLOW bit has two functions. It configures the ICG module for a fast (1 MHz to 8 MHz) or slow (30 kHz to 100 kHz) speed crystal. The option also configures the clock monitor operation in the ICG module to expect an external frequency higher (307.2 kHz to 32 MHz) or lower (60 Hz to 307.2 kHz) than the base frequency of the internal oscillator. See Chapter 7 Internal Clock Generator Module (ICG).

    1 = ICG set for slow external crystal operation
    0 = ICG set for fast external crystal operation

EXTXTALEN — External Crystal Enable Bit

EXTXTALEN enables the external oscillator circuits to be configured for a crystal configuration where the PTB6/(OSC1) and PTB7/(OSC2)/$\overline{\text{RST}}$ pins are the connections for an external crystal.

NOTE
This bit does not function without setting the EXTCLKEN bit also.

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Clearing the EXTXTALEN bit (default setting) allows the PTB7/(OSC2)/$\overline{RST}$ pin to function as a general-purpose I/O pin. Refer to Table 4-1 for configuration options for the external source. See Chapter 7 Internal Clock Generator Module (ICG) for a more detailed description of the external clock operation.

EXTXTALEN, when set, also configures the clock monitor to expect an external clock source in the valid range of crystals (30 kHz to 100 kHz or 1 MHz to 8 MHz). When EXTXTALEN is clear, the clock monitor will expect an external clock source in the valid range for externally generated clocks when using the clock monitor (60 Hz to 32 MHz).

EXTXTALEN, when set, also configures the external clock stabilization divider in the clock monitor for a 4096-cycle timeout to allow the proper stabilization time for a crystal. When EXTXTALEN is clear, the stabilization divider is configured to 16 cycles since an external clock source does not need a startup time.

 1 = Allows PTB7/(OSC2)/$\overline{RST}$ to be an external crystal connection.
 0 = PTB7/(OSC2)/$\overline{RST}$ functions as an I/O port pin (default).

OSCENINSTOP — Oscillator Enable In Stop Mode Bit

OSCENINSTOP, when set, will enable the internal clock generator module to continue to generate clocks (either internal, ICLK, or external, ECLK) in stop mode. See Chapter 7 Internal Clock Generator Module (ICG). This function is used to keep the timebase running while the rest of the microcontroller stops. See Chapter 14 Timebase Module (TBM). When clear, all clock generation will cease and both ICLK and ECLK will be forced low during stop mode. The default state for this option is clear, disabling the ICG in stop mode.

 1 = Oscillator enabled to operate during stop mode
 0 = Oscillator disabled during stop mode (default)

NOTE
This bit has the same functionality as the OSCSTOPENB CONFIG bit in MC68HC908GP20 and MC68HC908GR8 parts.

SCIBDSRC — SCI Baud Rate Clock Source Bit

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

 1 = Internal data bus clock is used as clock source for SCI.
 0 = CGMXCLK is used as clock source for SCI.

COPRS — COP Rate Select Bit

COPD selects the COP timeout period. Reset clears COPRS. See Chapter 5 Computer Operating Properly Module (COP).

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

LVISTOP — LVI Enable in Stop Mode Bit

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

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

LVIRSTD — LVI Reset Disable Bit

LVIRSTD disables the reset signal from the LVI module. See Chapter 10 Low-Voltage Inhibit (LVI).

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

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

LVIPWRD — LVI Power Disable Bit

LVIPWRD disables the LVI module. See Chapter 10 Low-Voltage Inhibit (LVI).
    1 = LVI module power disabled
    0 = LVI module power enabled

LVI5OR3 — LVI 5-V or 3-V Operating Mode Bit

LVI5OR3 selects the voltage operating mode of the LVI module (see See Chapter 10 Low-Voltage Inhibit (LVI).). The voltage mode selected for the LVI should match the operating V_{DD}. See Chapter 17 Electrical Specifications for the LVI's voltage trip points for each of the modes.
    1 = LVI operates in 5-V mode.
    0 = LVI operates in 3-V mode.

NOTE

The LVI5OR3 bit is cleared by a power-on reset (POR) only. Other resets will leave this bit unaffected.

SSREC — Short Stop Recovery Bit

SSREC enables the CPU to exit stop mode with a delay of 32 CGMXCLK cycles instead of a 4096-CGMXCLK cycle delay.
    1 = Stop mode recovery after 32 CGMXCLK cycles
    0 = Stop mode recovery after 4096 CGMXCLCK cycles

NOTE

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

If the system clock source selected is the internal oscillator or the external crystal and the OSCENINSTOP configuration bit is not set, the oscillator will be disabled during stop mode. The short stop recovery does not provide enough time for oscillator stabilization and thus the SSREC bit should not be set.

When using the LVI during normal operation but disabling during stop mode, the LVI will have an enable time of t_{EN}. The system stabilization time for power-on reset and long stop recovery (both 4096 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.

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 Chapter 5 Computer Operating Properly Module (COP).
    1 = COP module disabled
    0 = COP module enabled

Chapter 5
Computer Operating Properly Module (COP)

5.1 Introduction

The computer operating properly (COP) module contains a free-running counter that generates a reset if allowed to overflow. The COP module helps software to recover from a runaway code. Periodically clearing the COP counter will prevent a COP reset from occurring. The COP module can be disabled through the COPD bit in the configuration (CONFIG) register.

5.2 Block Diagram
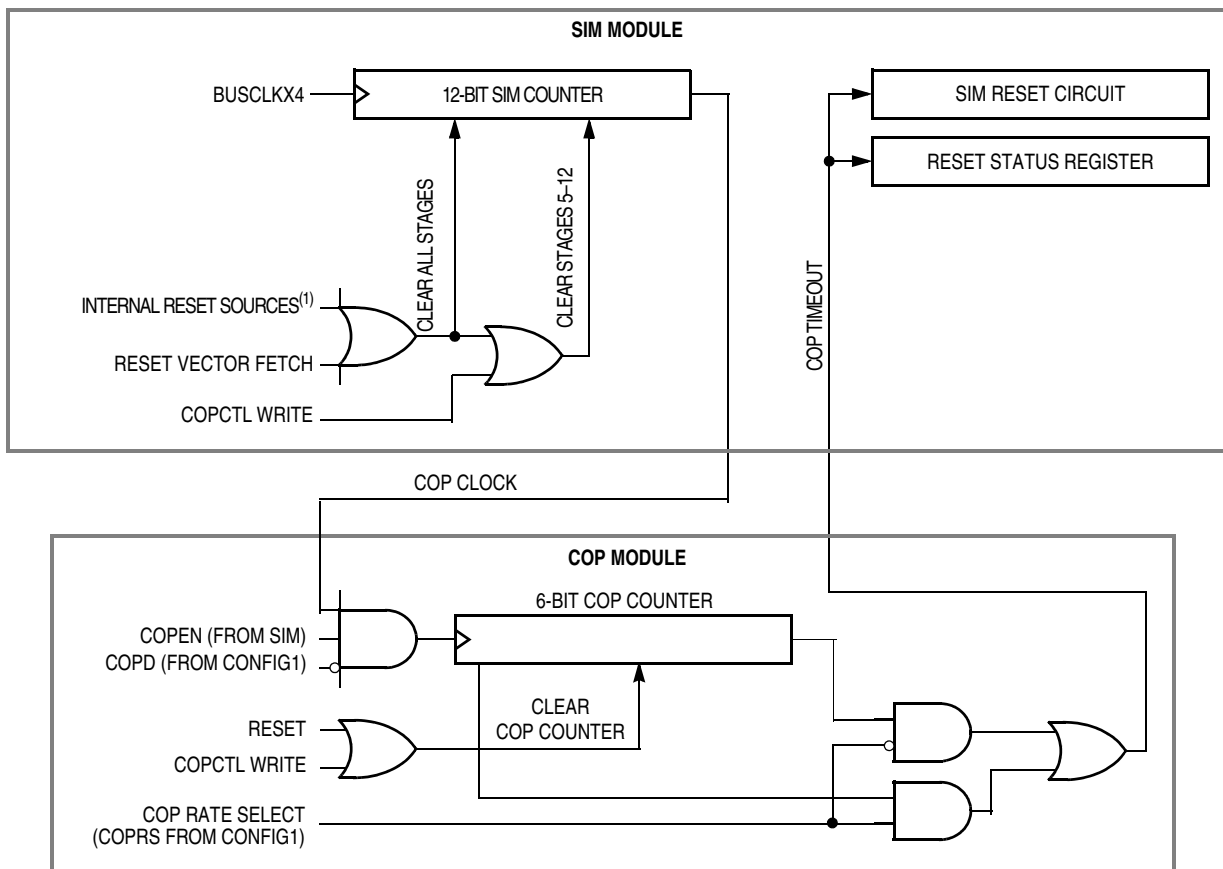
Figure 5-1. COP Block Diagram

5.3 Functional Description

The COP counter is a free-running 6-bit counter preceded by a 12-bit prescaler. If not cleared by software, the COP counter overflows and generates an asynchronous reset after $2^{13}-2^4$ or $2^{18}-2^4$ CGMXCLK cycles, depending on the state of the COP rate select bit, COPRS, in the configuration register. With a $2^{18}-2^4$ CGMXCLK cycle overflow option, a 4.9152-MHz CGMXCLK frequency gives a COP timeout period of 53.3 ms. Writing any value to location $FFFF before an overflow occurs prevents a COP reset by clearing the COP counter and stages 5–12 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 an internal reset for 64 CGMXCLK cycles and sets the COP bit in the system integration module (SIM) reset status register (SRSR).

In monitor mode, the COP is disabled if the $\overline{\text{IRQ1}}$ pin is held at V_{TST}.

> *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.*

5.4 I/O Signals

The following paragraphs describe the signals shown in Figure 5-1.

5.4.1 CGMXCLK

CGMXCLK is the internal clock generator (ICG) module's oscillator output signal. CGMXCLK is selected from either the internal clock source or the external crystal.

5.4.2 STOP Instruction

The STOP instruction clears the COP prescaler.

5.4.3 COPCTL Write

Writing any value to the COP control register (COPCTL) clears the COP counter and clears stages 12–5 of the COP prescaler. Reading the COP control register returns the low byte of the reset vector.

5.4.4 Power-On Reset

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

5.4.5 Internal Reset

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

5.4.6 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.

5.4.7 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the configuration register. See Chapter 4 Configuration Register (CONFIG).

5.4.8 COPRS (COP Rate Select)

The COPRS signal reflects the state of the COP rate select bit (COPRS) in the configuration register. See Chapter 4 Configuration Register (CONFIG).

5.5 COP Control Register

The COP control register (COPCTL) is located at address $FFFF and overlaps the reset vector. Writing any value to $FFFF clears the COP counter and stages 12–5 of the COP prescaler and starts a new timeout period. Reading location $FFFF returns the low byte of the reset vector.

Address: $FFFF

	Bit 7	6	5	4	3	2	1	Bit 0
Read:				Low byte of reset vector				
Write:				Clear COP counter				
Reset:				Unaffected by reset				
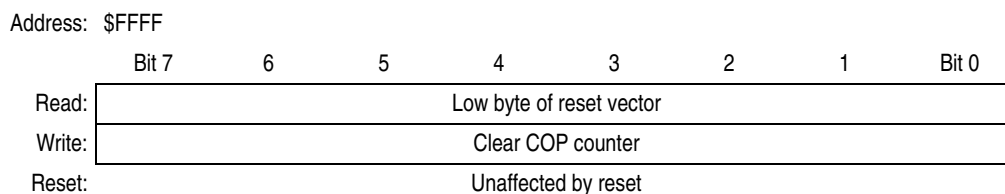
Figure 5-2. COP Control Register (COPCTL)

5.6 Interrupts

The COP does not generate CPU interrupt requests.

5.7 Monitor Mode

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

5.8 Low-Power Modes

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

5.8.1 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.

5.8.2 Stop Mode

Stop mode holds the 12-bit prescaler counter in reset until after stop mode is exited. Service the COP immediately before entering or after exiting stop mode to ensure a full COP timeout period after entering or exiting stop mode.

To prevent inadvertently turning off the COP with a STOP instruction, a configuration option is available that disables the STOP instruction. When the STOP bit in the configuration has the STOP instruction disabled, execution of a STOP instruction results in an illegal opcode reset.

Chapter 6
Central Processor Unit (CPU)

6.1 Introduction

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

6.2 Features

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

6.3 CPU Registers

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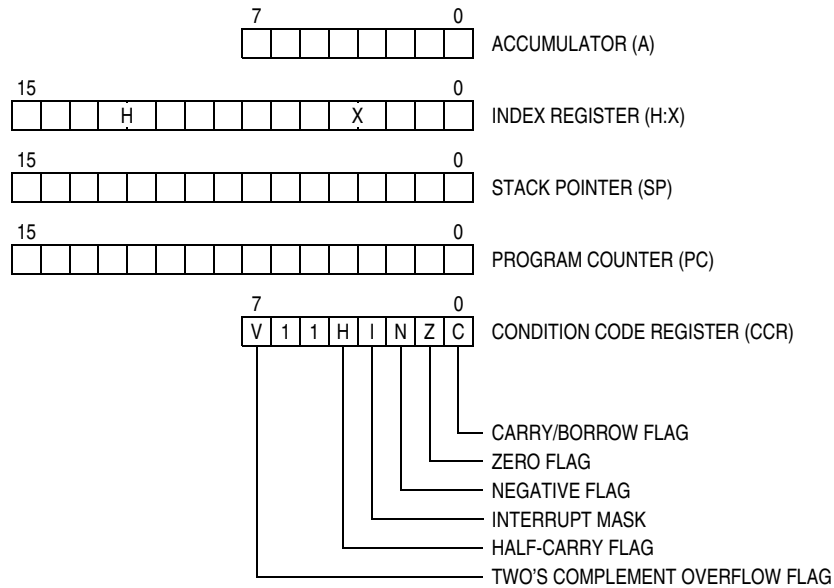
Figure 6-1. CPU Registers

6.3.1 Accumulator

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and the results of arithmetic/logic operations.
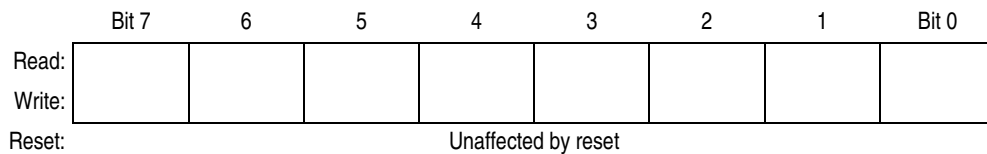
Figure 6-2. Accumulator (A)

6.3.2 Index Register

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

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

The index register can serve also as a temporary data storage location.
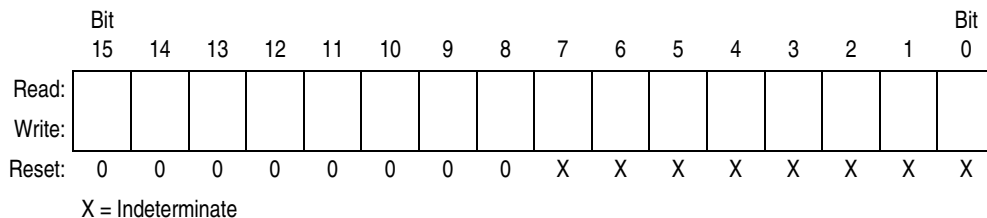
Figure 6-3. Index Register (H:X)

6.3.3 Stack Pointer

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

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

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
Read:																
Write:																
Reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
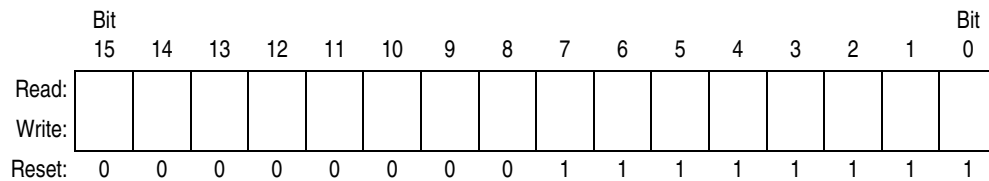
Figure 6-4. Stack Pointer (SP)

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

6.3.4 Program Counter

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

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

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

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
Read:																
Write:																
Reset:						Loaded with vector from $FFFE and $FFFF										
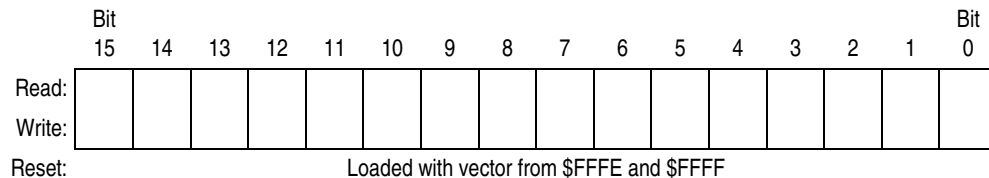
Figure 6-5. Program Counter (PC)

6.3.5 Condition Code Register

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

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

X = Indeterminate

Figure 6-6. Condition Code Register (CCR)

V — Overflow Flag
The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag.
1 = Overflow
0 = No overflow

H — Half-Carry Flag
The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C flags to determine the appropriate correction factor.
1 = Carry between bits 3 and 4
0 = No carry between bits 3 and 4

I — Interrupt Mask
When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the interrupt vector is fetched.
1 = Interrupts disabled
0 = Interrupts enabled

NOTE

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

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

N — Negative Flag
The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result.
1 = Negative result
0 = Non-negative result

Z — Zero Flag

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

    1 = Zero result

    0 = Non-zero result

C — Carry/Borrow Flag

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

    1 = Carry out of bit 7

    0 = No carry out of bit 7

6.4 Arithmetic/Logic Unit (ALU)

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

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

6.5 Low-Power Modes

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

6.5.1 Wait Mode

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

6.5.2 Stop Mode

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

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

6.6 CPU During Break Interrupts

If a break module is present on the MCU, the CPU starts a break interrupt by:
- Loading the instruction register with the SWI instruction
- Loading the program counter with $FFFC:$FFFD or with $FEFC:$FEFD in monitor mode

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

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

6.7 Instruction Set Summary

Table 6-1 provides a summary of the M68HC08 instruction set.

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

Source Form	Operation	Description	V	H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
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)	C ← [b7 ... b0] ← 0	↕	–	–	↕	↕	↕	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	[b7 ... b0] → C	↕	–	–	↕	↕	↕	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
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

Table 6-1. Instruction Set Summary (Sheet 2 of 6)

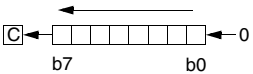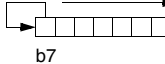| Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
BHS *rel*	Branch if Higher or Same (Same as BCC)	PC ← (PC) + 2 + *rel* ? (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BIH *rel*	Branch if $\overline{\text{IRQ}}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 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
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

Table 6-1. Instruction Set Summary (Sheet 3 of 6)

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

Table 6-1. Instruction Set Summary (Sheet 4 of 6)

Source Form	Operation	Description	V	H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff	2 3 4 3 2
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Unconditional Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	4 5 6 5 4
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X LDA *opr*,SP LDA *opr*,SP	Load A from M	A ← (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A6 B6 C6 D6 E6 F6 9EE6 9ED6	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LDHX #*opr* LDHX *opr*	Load H:X from M	H:X ← (M:M + 1)	0	–	–	↕	↕	–	IMM DIR	45 55	ii jj dd	3 4
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X LDX *opr*,SP LDX *opr*,SP	Load X from M	X ← (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	AE BE CE DE EE FE 9EEE 9EDE	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X LSL *opr*,SP	Logical Shift Left (Same as ASL)	C ← [b7 ... b0] ← 0	↕	–	–	↕	↕	↕	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 → [b7 ... b0] → C	↕	–	–	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 ← (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
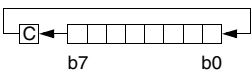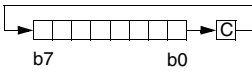
MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Table 6-1. Instruction Set Summary (Sheet 5 of 6)

Source Form	Operation	Description	V	H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
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	(diagram)	↕	–	–	↕	↕	↕	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	(diagram)	↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	4 1 1 4 3 5
RSP	Reset Stack Pointer	SP ← $FF	–	–	–	–	–	–	INH	9C		1
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	↕	INH	80		7
RTS	Return from Subroutine	SP ← SP + 1; Pull (PCH) SP ← SP + 1; Pull (PCL)	–	–	–	–	–	–	INH	81		4
SBC #opr SBC opr SBC opr SBC opr,X SBC opr,X SBC ,X SBC opr,SP SBC opr,SP	Subtract with Carry	A ← (A) − (M) − (C)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A2 B2 C2 D2 E2 F2 9EE2 9ED2	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
SEC	Set Carry Bit	C ← 1	–	–	–	–	–	1	INH	99		1
SEI	Set Interrupt Mask	I ← 1	–	–	1	–	–	–	INH	9B		2
STA opr STA opr STA opr,X STA opr,X STA ,X STA opr,SP STA opr,SP	Store A in M	M ← (A)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh ll ee ff ff ff ee ff	3 4 4 3 2 4 5
STHX opr	Store H:X in M	(M:M + 1) ← (H:X)	0	–	–	↕	↕	–	DIR	35	dd	4
STOP	Enable Interrupts, Stop Processing, Refer to MCU Documentation	I ← 0; Stop Processing	–	–	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

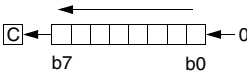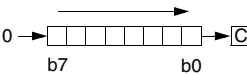**Table 6-1. Instruction Set Summary (Sheet 6 of 6)**

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) − 1; Push (PCH) SP ← (SP) − 1; Push (X) SP ← (SP) − 1; Push (A) SP ← (SP) − 1; Push (CCR) SP ← (SP) − 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	–	–	1	–	–	–	INH	83		9
TAP	Transfer A to CCR	CCR ← (A)	↕	↕	↕	↕	↕	↕	INH	84		2
TAX	Transfer A to X	X ← (A)	–	–	–	–	–	–	INH	97		1
TPA	Transfer CCR to A	A ← (CCR)	–	–	–	–	–	–	INH	85		1
TST 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
WAIT	Enable Interrupts; Wait for Interrupt	I bit ← 0; Inhibit CPU clocking until interrupted	–	–	0	–	–	–	INH	8F		1

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

6.8 Opcode Map

See .

Table 6-2. Opcode Map

LSB \ MSB	Bit Manip. DIR 0	Bit Manip. DIR 1	Branch REL 2	RMW DIR 3	RMW INH 4	RMW INH 5	RMW IX1 6	RMW SP1 9E6	RMW IX 7	Control INH 8	Control INH 9	Reg/Mem IMM A	Reg/Mem DIR B	Reg/Mem EXT C	Reg/Mem IX2 D	Reg/Mem SP2 9ED	Reg/Mem IX1 E	Reg/Mem SP1 9EE	Reg/Mem IX 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		2 CPHX 2 DIR	1 TPA 1 INH	2 TSX 1 INH	2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	4 BIT 3 IX2	5 BIT 4 SP2	3 BIT 2 IX1	4 BIT 3 SP1	2 BIT 1 IX
6	5 BRSET3 3 DIR	4 BSET3 2 DIR	3 BNE 2 REL	4 ROR 2 DIR	1 RORA 1 INH	1 RORX 1 INH	4 ROR 2 IX1	5 ROR 3 SP1	3 ROR 1 IX	2 PULA 1 INH		2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	4 LDA 3 IX2	5 LDA 4 SP2	3 LDA 2 IX1	4 LDA 3 SP1	2 LDA 1 IX
7	5 BRCLR3 3 DIR	4 BCLR3 2 DIR	3 BEQ 2 REL	4 ASR 2 DIR	1 ASRA 1 INH	1 ASRX 1 INH	4 ASR 2 IX1	5 ASR 3 SP1	3 ASR 1 IX	2 PSHA 1 INH	1 TAX 1 INH	2 AIS 2 IMM	3 STA 2 DIR	4 STA 3 EXT	4 STA 3 IX2	5 STA 4 SP2	3 STA 2 IX1	4 STA 3 SP1	2 STA 1 IX
8	5 BRSET4 3 DIR	4 BSET4 2 DIR	3 BHCC 2 REL	4 LSL 2 DIR	1 LSLA 1 INH	1 LSLX 1 INH	4 LSL 2 IX1	5 LSL 3 SP1	3 LSL 1 IX	2 PULX 1 INH	1 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	4 EOR 3 IX2	5 EOR 4 SP2	3 EOR 2 IX1	4 EOR 3 SP1	2 EOR 1 IX
9	5 BRCLR4 3 DIR	4 BCLR4 2 DIR	3 BHCS 2 REL	4 ROL 2 DIR	1 ROLA 1 INH	1 ROLX 1 INH	4 ROL 2 IX1	5 ROL 3 SP1	3 ROL 1 IX	2 PSHX 1 INH	1 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	4 ADC 3 IX2	5 ADC 4 SP2	3 ADC 2 IX1	4 ADC 3 SP1	2 ADC 1 IX
A	5 BRSET5 3 DIR	4 BSET5 2 DIR	3 BPL 2 REL	4 DEC 2 DIR	1 DECA 1 INH	1 DECX 1 INH	4 DEC 2 IX1	5 DEC 3 SP1	3 DEC 1 IX	2 PULH 1 INH	2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	4 ORA 3 IX2	5 ORA 4 SP2	3 ORA 2 IX1	4 ORA 3 SP1	2 ORA 1 IX
B	5 BRCLR5 3 DIR	4 BCLR5 2 DIR	3 BMI 2 REL	5 DBNZ 3 DIR	3 DBNZA 2 INH	3 DBNZX 2 INH	5 DBNZ 3 IX1	6 DBNZ 4 SP1	4 DBNZ 2 IX	2 PSHH 1 INH	2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	4 ADD 3 IX2	5 ADD 4 SP2	3 ADD 2 IX1	4 ADD 3 SP1	2 ADD 1 IX
C	5 BRSET6 3 DIR	4 BSET6 2 DIR	3 BMC 2 REL	4 INC 2 DIR	1 INCA 1 INH	1 INCX 1 INH	4 INC 2 IX1	5 INC 3 SP1	3 INC 1 IX	1 CLRH 1 INH	1 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2		3 JMP 2 IX1		2 JMP 1 IX
D	5 BRCLR6 3 DIR	4 BCLR6 2 DIR	3 BMS 2 REL	3 TST 2 DIR	1 TSTA 1 INH	1 TSTX 1 INH	3 TST 2 IX1	4 TST 3 SP1	2 TST 1 IX		1 NOP 1 INH	4 BSR 2 REL	4 JSR 2 DIR	5 JSR 3 EXT	6 JSR 3 IX2		5 JSR 2 IX1		4 JSR 1 IX
E	5 BRSET7 3 DIR	4 BSET7 2 DIR	3 BIL 2 REL		5 MOV 3 DD	4 MOV 2 DIX+	4 MOV 3 IMD		4 MOV 2 IX+D	1 STOP 1 INH	*	2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	4 LDX 3 IX2	5 LDX 4 SP2	3 LDX 2 IX1	4 LDX 3 SP1	2 LDX 1 IX
F	5 BRCLR7 3 DIR	4 BCLR7 2 DIR	3 BIH 2 REL	3 CLR 2 DIR	1 CLRA 1 INH	1 CLRX 1 INH	3 CLR 2 IX1	4 CLR 3 SP1	2 CLR 1 IX	1 WAIT 1 INH	1 TXA 1 INH	2 AIX 2 IMM	3 STX 2 DIR	4 STX 3 EXT	4 STX 3 IX2	5 STX 4 SP2	3 STX 2 IX1	4 STX 3 SP1	2 STX 1 IX

INH Inherent
IMM Immediate
DIR Direct
EXT Extended
DD Direct-Direct
IX+D Indexed-Direct

REL Relative
IX Indexed, No Offset
IX1 Indexed, 8-Bit Offset
IX2 Indexed, 16-Bit Offset
IMD Immediate-Direct
DIX+ Direct-Indexed

SP1 Stack Pointer, 8-Bit Offset
SP2 Stack Pointer, 16-Bit Offset
IX+ Indexed, No Offset with Post Increment
IX1+ Indexed, 1-Byte Offset with Post Increment

*Pre-byte for stack pointer indexed instructions

MSB / LSB — 0 = High Byte of Opcode in Hexadecimal

Low Byte of Opcode in Hexadecimal — 0

5 = Cycles
BRSET0 = Opcode Mnemonic
3 DIR = Number of Bytes / Addressing Mode

Chapter 7
Internal Clock Generator Module (ICG)

7.1 Introduction

The internal clock generator module (ICG) is used to create a stable clock source for the microcontroller without using any external components. The ICG generates the oscillator output clock (CGMXCLK), which is used by the computer operating properly (COP), low-voltage inhibit (LVI), and other modules. The ICG also generates the clock generator output (CGMOUT), which is fed to the system integration module (SIM) to create the bus clocks. The bus frequency will be one-fourth the frequency of CGMXCLK and one-half the frequency of CGMOUT. Finally, the ICG generates the timebase clock (TBMCLK), which is used in the timebase module (TBM).

7.2 Features

Features of the ICG include:

- Selectable external clock generator, either one pin external source or two pin crystal, multiplexed with port pins
- Internal clock generator with programmable frequency output in integer multiples of a nominal frequency (307.2 kHz $\pm$ 25%)
- Frequency adjust (trim) register to improve variability to $\pm$ 2%
- Bus clock software selectable from either internal or external clock (bus frequency range from 76.8 kHz $\pm$ 25% to 9.75 MHz $\pm$ 25% in 76.8 kHz increments — note that for the MC68HC908KX8, MC68HC908KX2, and MC68HC08KX8, do not exceed the maximum bus frequency of 8 MHz at 5.0 V and 4 MHz at 3.0 V
- Timebase clock automatically selected externally, if external clock is available
- Clock monitor for both internal and external clocks

7.3 Functional Description

As shown in Figure 7-1, the ICG contains these major submodules:

- Clock enable circuit
- Internal clock generator
- External clock generator
- Clock monitor circuit
- Clock selection circuit
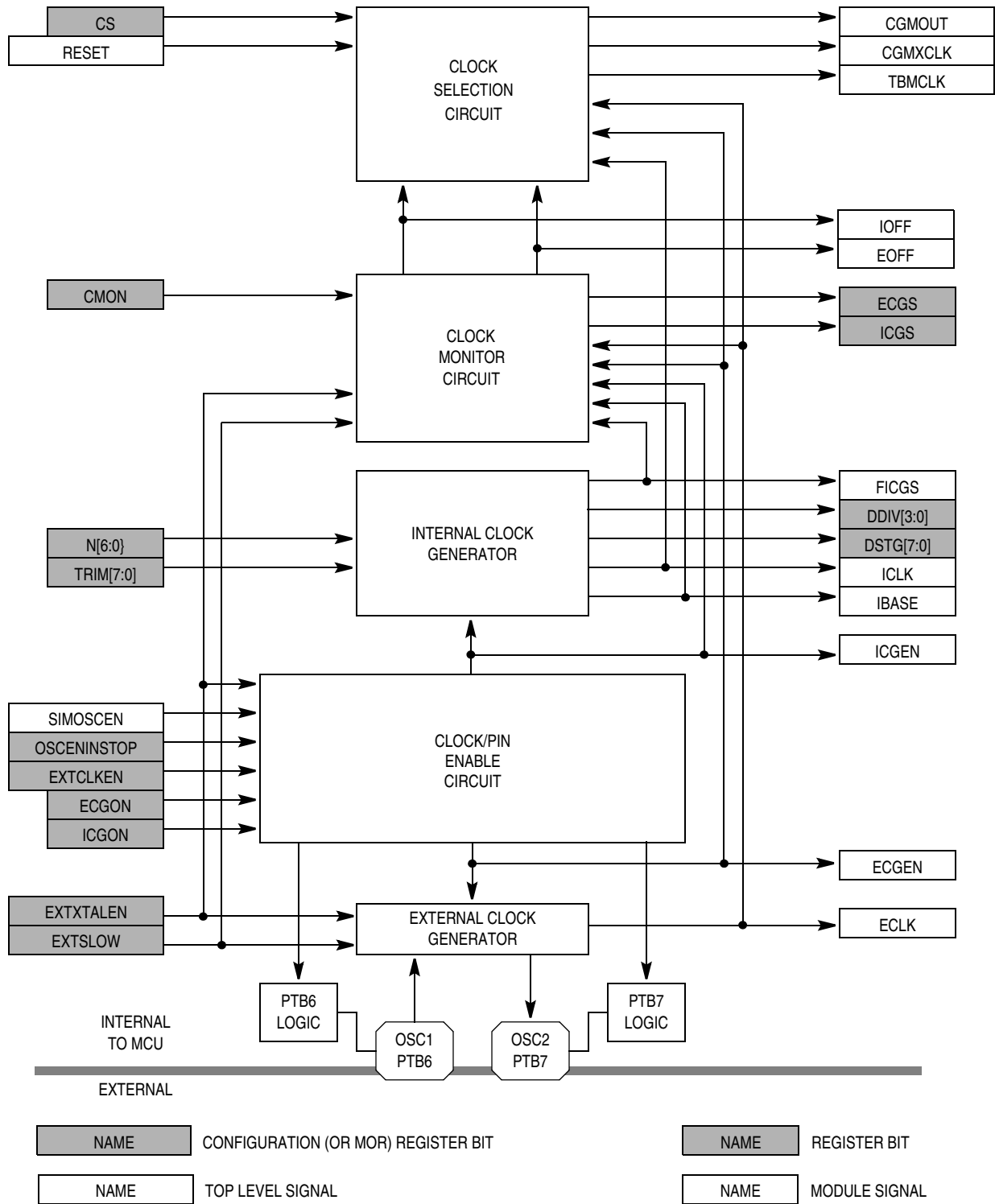
Figure 7-1. ICG Module Block Diagram

7.3.1 Clock Enable Circuit

The clock enable circuit is used to enable the internal clock (ICLK) or external clock (ECLK) and the port logic which is shared with the oscillator pins (OSC1 and OSC2). The clock enable circuit generates an ICG stop (ICGSTOP) signal which stops all clocks (ICLK, ECLK, and the low-frequency base clock, IBASE). ICGSTOP is set and the ICG is disabled in stop mode if the oscillator enable in stop (OSCENINSTOP) bit in the CONFIG (or MOR) register is clear. The ICG clocks will be enabled in stop mode if OSCENINSTOP is high.

The internal clock enable signal (ICGEN) turns on the ICG which generates ICLK. ICGEN is set (active) whenever the ICGON bit is set and the ICGSTOP signal is clear. When ICGEN is clear, ICLK and IBASE are both low.

The external clock enable signal (ECGEN) turns on the external clock generator which generates ECLK. ECGEN is set (active) whenever the ECGON bit is set and the ICGSTOP signal is clear. ECGON cannot be set unless the external clock enable (EXTCLKEN) bit in the CONFIG (or MOR) register is set. When ECGEN is clear, ECLK is low.

The port B6 enable signal (PB6EN) turns on the port B6 logic. Since port B6 is on the same pin as OSC1, this signal is only active (set) when the external clock function is not desired. Therefore, PB6EN is clear when ECGON is set. PB6EN is not gated with ICGSTOP, which means that if the ECGON bit is set, the port B6 logic will remain disabled in stop mode.

The port B7 enable signal (PB7EN) turns on the port B7 logic. Since port B7 is on the same pin as OSC2, this signal is only active (set) when two-pin oscillator function is not desired. Therefore, PB7EN is clear when ECGON and the external crystal enable (EXTXTALEN) bit in the CONFIG (or MOR) register are both set. PB6EN is not gated with ICGSTOP, which means that if ECGON and EXTXTALEN are set, the port B7 logic will remain disabled in stop mode.

7.3.2 Internal Clock Generator

The ICG, shown in Figure 7-2, creates a low frequency base clock (IBASE), which operates at a nominal frequency (f_{NOM}) of 307.2 kHz $\pm$ 25%, and an internal clock (ICLK) which is an integer multiple of IBASE. This multiple is the ICG multiplier factor (N), which is programmed in the ICG multiplier register (ICGMR). The ICG is turned off and the output clocks (IBASE and ICLK) are held low when the ICG enable signal (ICGEN) is clear.

The ICG contains:

- A digitally controlled oscillator
- A modulo "N" divider
- A frequency comparator, which contains voltage and current references, a frequency to voltage converter, and comparators
- A digital loop filter
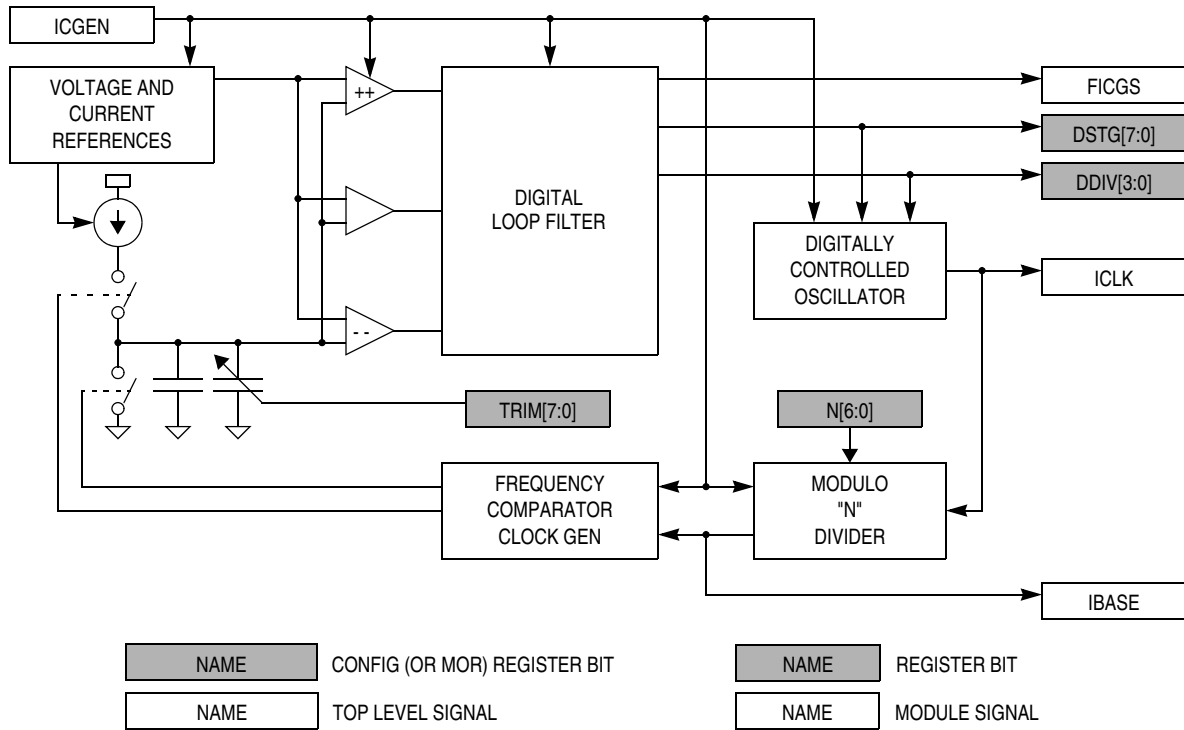
Figure 7-2. Internal Clock Generator Block Diagram

7.3.2.1 Digitally Controlled Oscillator

The digitally controlled oscillator (DCO) is an inaccurate oscillator which generates the internal clock (ICLK). The clock period of ICLK is dependent on the digital loop filter outputs (DSTG[7:0] and DDIV[3:0]). Because there is only a limited number of bits in DDIV and DSTG, the precision of the output (ICLK) is restricted to a precision of approximately ±0.202% to ±0.368% when measured over several cycles (of the desired frequency). Additionally, since the propagation delays of the devices used in the DCO ring oscillator are a measurable fraction of the bus clock period, reaching the long-term precision may require alternately running faster and slower than desired, making the worst case cycle-to-cycle frequency variation ±6.45% to ±11.8% (of the desired frequency). The valid values of DDIV:DSTG range from $000 to $9FF. For more information on the quantization error in the DCO, see 7.4.4 Quantization Error in DCO Output.

7.3.2.2 Modulo "N" Divider

The modulo "N" divider creates the low frequency base clock (IBASE) by dividing the internal clock (ICLK) by the ICG multiplier factor (N), contained in the ICG multiplier register (ICGMR). When N is programmed to a $01 or $00, the divider is disabled and ICLK is passed through to IBASE undivided. When the ICG is stable, the frequency of IBASE will be equal to the nominal frequency (f_{NOM}) of 307.2 kHz ± 25%.

7.3.2.3 Frequency Comparator

The frequency comparator effectively compares the low frequency base clock (IBASE) to a nominal frequency, f_{NOM}. First, the frequency comparator converts IBASE to a voltage by charging a known capacitor with a current reference for a period dependent on IBASE. This voltage is compared to a voltage

reference with comparators, whose outputs are fed to the digital loop filter. The dependence of these outputs on the capacitor size, current reference, and voltage reference causes up to $\pm25\%$ error in f_{NOM}.

7.3.2.4 Digital Loop Filter

The digital loop filter (DLF) uses the outputs of the frequency comparator to adjust the internal clock (ICLK) clock period. The DLF generates the DCO divider control bits (DDIV[3:0]) and the DCO stage control bits (DSTG[7:0]), which are fed to the DCO. The DLF first concatenates the DDIV and DSTG registers (DDIV[3:0]:DSTG[7:0]) and then adds or subtracts a value dependent on the relative error in the low frequency base clock's period, as shown in Table 7-1. In some extreme error conditions, such as operating at a V_{DD} level which is out of specification, the DLF may attempt to use a value above the maximum ($9FF) or below the minimum ($000). In both cases, the value for DDIV will be between $A and $F. In this range, the DDIV value will be interpreted the same as $9 (the slowest condition). Recovering from this condition requires subtracting (increasing frequency) in the normal fashion until the value is again below $9FF (if the desired value is $9xx, the value may settle at $Axx through $Fxx — this is an acceptable operating condition). If the error is less than $\pm15\%$, the ICG's filter stable indicator (FICGS) is set, indicating relative frequency accuracy to the clock monitor.

All FLASH mask sets other than 0K45D, 1K45D, 0L09H, 1L09H have 15% comparators that improve stability at low temperatures.

Table 7-1. Correction Sizes from DLF to DCO

Frequency Error of IBASE compared to f_{NOM}	DDVI[3:0]:DSTG[7:0] Correction	Current to New DDIV[3:0]:DSTG[7:0] [1]		Relative Correction in DCO	
IBASE < 0.85 f_{NOM}	−32 (−$020)	Minimum	$xFF to $xDF	−2/31	−6.45%
		Maximum	$x20 to $x00	−2/19	−10.5%
0.85 f_{NOM} < IBASE IBASE < f_{NOM}	−1 (−$001)	Minimum	$xFF to $xFE	−0.0625/31	−0.202%
		Maximum	$x01 to $x00	−0.0625/17.0625	−0.366%
f_{NOM} < IBASE IBASE < 1.15 f_{NOM}	+1 (+$001)	Minimum	$xFE to $xFF	+0.0625/30.9375	+0.202%
		Maximum	$x00 to $x01	+0.0625/17	+0.368%
1.15 f_{NOM} < IBASE	+32 (+$020)	Minimum	$xDF to $xFF	+2/29	+6.90%
		Maximum	$x00 to $x20	+2/17	+11.8%

1. x =Maximum error is independent of value in DDIV[3:0]. DDIV increments or decrements when an addition to DSTG[7:0] carries or borrows.

7.3.3 External Clock Generator

The ICG also provides for an external oscillator or external clock source, if desired. The external clock generator, shown in Figure 7-3, contains an external oscillator amplifier and an external clock input path.

7.3.3.1 External Oscillator Amplifier

The external oscillator amplifier provides the gain required by an external crystal connected in a Pierce oscillator configuration. The amount of this gain is controlled by the slow external (EXTSLOW) bit in the CONFIG (or MOR) register. When EXTSLOW is set, the amplifier gain is reduced for operating low-frequency crystals (32 kHz to 100 kHz). When EXTSLOW is clear, the amplifier gain will be sufficient for 1 MHz to 8 MHz crystals. EXTSLOW must be configured correctly for the given crystal or the circuit may not operate.
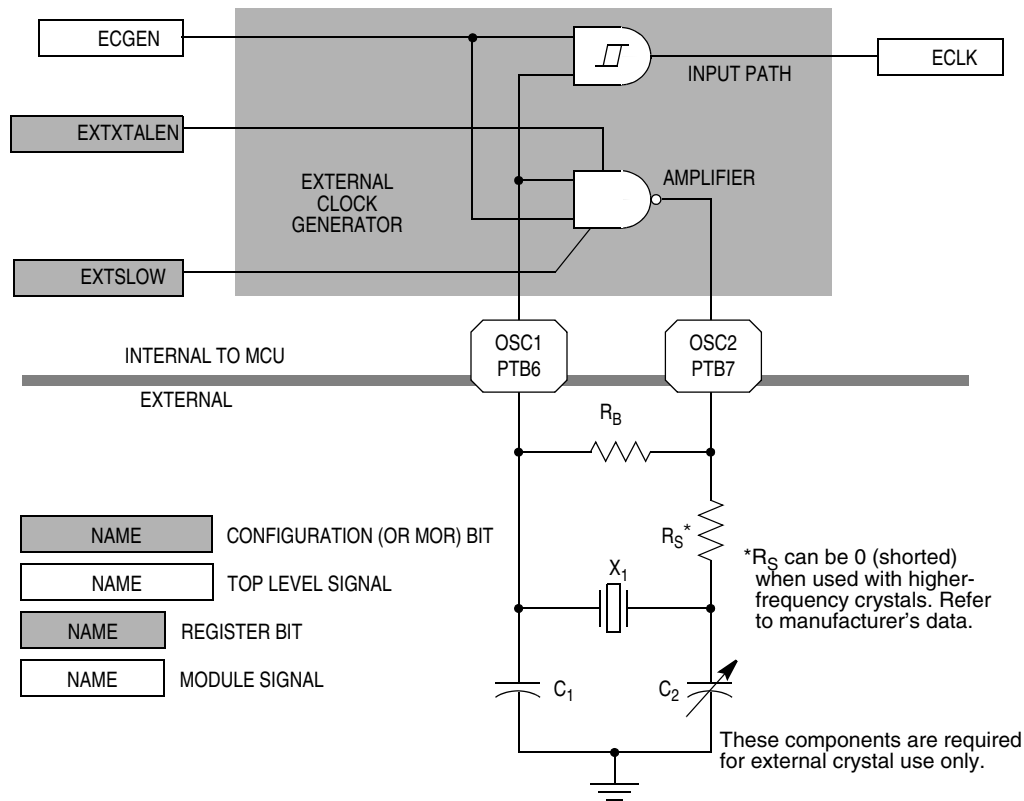
Figure 7-3. External Clock Generator Block Diagram

The amplifier is enabled when the external clock generator enable (ECGEN) signal is set and when the external crystal enable (EXTXTALEN) bit in the CONFIG (or MOR) register is set. ECGEN is controlled by the clock enable circuit (see 7.3.1 Clock Enable Circuit), and indicates that the external clock function is desired. When enabled, the amplifier will be connected between the PTB6/(OSC1) and PTB7/(OSC2)/$\overline{\text{RST}}$ pins. Otherwise, the PTB7/(OSC2)/$\overline{\text{RST}}$ pin reverts to its port function. In its typical configuration, the external oscillator requires five external components:

- Crystal, X_1
- Fixed capacitor, C_1
- Tuning capacitor, C_2 (can also be a fixed capacitor)
- Feedback resistor, R_B
- Series resistor, R_S (included in the diagram to follow strict Pierce oscillator guidelines and may not be required for all ranges of operation, especially with high frequency crystals. Refer to the crystal manufacturer's data for more information.)

7.3.3.2 External Clock Input Path

The external clock input path is the means by which the microcontroller uses an external clock source. The input to the path is the PTB6/(OSC1) pin and the output is the external clock (ECLK). The path, which contains input buffering, is enabled when the external clock generator enable signal (ECGEN) is set. When not enabled, the PTB6/(OSC1) pin reverts to its port function.

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

7.3.4 Clock Monitor Circuit

The ICG contains a clock monitor circuit which, when enabled, will continuously monitor both the external clock (ECLK) and the internal clock (ICLK) to determine if either clock source has been corrupted. The clock monitor circuit, shown in Figure 7-4, contains these blocks:

- Clock monitor reference generator
- Internal clock activity detector
- External clock activity detector
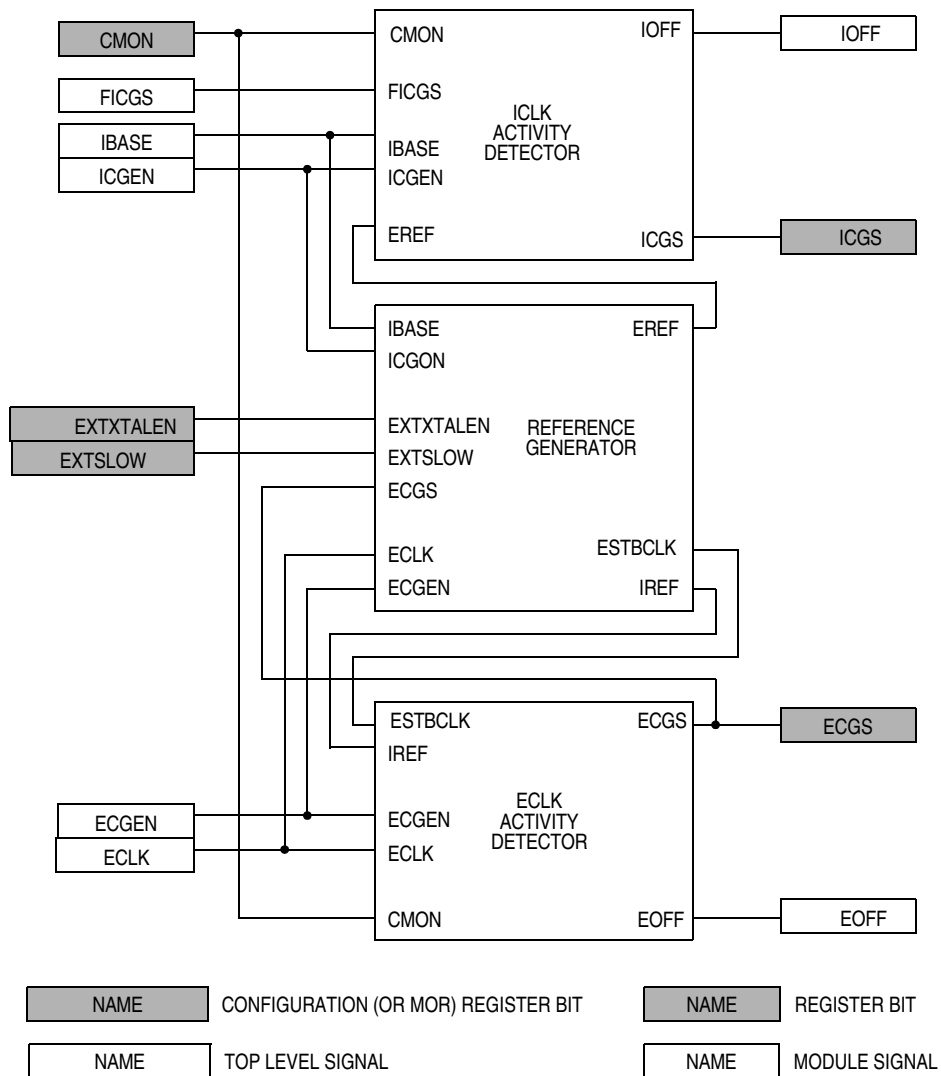
Figure 7-4. Clock Monitor Block Diagram

7.3.4.1 Clock Monitor Reference Generator

The clock monitor uses a reference based on one clock source to monitor the other clock source. The clock monitor reference generator generates the external reference clock (EREF) based on the external clock (ECLK) and the internal reference clock (IREF) based on the internal clock (ICLK). To simplify the circuit, the low frequency base clock (IBASE) is used in place of ICLK because it always operates at or

near 307.2 kHz. For proper operation, EREF must be at least twice as slow as IBASE and IREF must be at least twice as slow as ECLK.

To guarantee that IREF is slower than ECLK and EREF is slower than IBASE, one of the signals is divided down. Which signal is divided and by how much is determined by the external slow (EXTSLOW) and external crystal enable (EXTXTALEN) bits in the CONFIG (or MOR) register, according to the rules in Table 7-2.

NOTE
Each signal (IBASE and ECLK) is always divided by four. A longer divider is used on either IBASE or ECLK based on the EXTSLOW bit.

Table 7-2. Clock Monitor Reference Divider Ratios

ICGON	ECGON	ECGS	EXTSLOW	EXTXTALEN	External Frequency		EREF Divider Ratio	EREF Frequency	ESTBCLK Divider Ratio	ESTBCLK Frequency	IREF Divider Ratio[1]	IREF Frequency
0	x	x	x	x	U		U	U	U	U	Off	0
x	0	0	x	x	0		Off	0	Off	0	U	U
1	1	0	x	0	Minimum	60 Hz	Off	0	16 (ECLK)	3.75 Hz	1*4	76.8 kHz ± 25%
					Maximum	32 MHz				2.0 MHz		
1	1	0	x	1	Minimum	30 kHz	Off	0	4096 (ECLK)	7.324 kHz	1*4	76.8 kHz ± 25%
					Maximum	8 MHz				1.953 kHz		
1	1	1	0	0	Minimum	307.2 kHz	128*4	600 Hz	16 (ECLK)	19.2 kHz	1*4	76.8 kHz ± 25%
					Maximum	32 MHz		62.5 kHz		2.0 MHz		
1	1	1	0	1	Minimum	1 MHz	128*4	1.953 kHz	4096 (ECLK)	244 Hz	1*4	76.8 kHz ± 25%
					Maximum	8 MHz		15.63 kHz		1.953 kHz		
1	1	1	1	0	Minimum	60 Hz	1*4	15 Hz	16 (IBASE)[2]	19.2 kHz ± 25%	4096*4	18.75 Hz ± 125%
					Maximum	307.2 kHz		76.8 kHz				
1	1	1	1	1	Minimum	30 kHz	1*4	7.5 kHz	4096 (IBASE)[2]	75 Hz ± 25%	16*4	4.8 kHz ± 25%
					Maximum	100 kHz		25.0 kHz				

1. U = Unaffected; refer to section of table where ICGON or ECGON is set to 1.
2. IBASE is always used as the internal frequency (307.2 kHz).

To conserve size, the long divider (divide by 4096) is also used as an external crystal stabilization divider. The divider is reset when the external clock generator is turned off or in STOP (ECGEN is clear). When the external clock generator is first turned on, the external clock generator stable bit (ECGS) will be clear. This condition automatically selects ECLK as the input to the long divider. The external stabilization clock (ESTBCLK) will be ECLK divided by 16 when EXTXTALEN is low or 4096 when EXTXTALEN is high. This time-out allows the crystal to stabilize. The falling edge of ESTBCLK is used to set ECGS (ECGS will set after a full 16 or 4096 cycles). When ECGS is set, the divider returns to its normal function. ESTBCLK may be generated by either IBASE or ECLK, but any clocking will only reinforce the set condition. If ECGS is cleared because the clock monitor determined that ECLK was inactive, the divider will revert to a stabilization divider. Since this will change the EREF and IREF divide ratios, it is important to turn the clock monitor off (CMON = 0) after inactivity is detected to ensure valid recovery.

7.3.4.2 Internal Clock Activity Detector

The internal clock activity detector, shown in Figure 7-5, looks for at least one falling edge on the low-frequency base clock (IBASE) every time the external reference (EREF) is low. Since EREF is less than half the frequency of IBASE, this should occur every time. If it does not occur two consecutive times, the internal clock inactivity indicator (IOFF) is set. IOFF will be cleared the next time there is a falling edge of IBASE while EREF is low.

The internal clock stable bit (ICGS) is also generated in the internal clock activity detector. ICGS is set when the internal clock generator's filter stable signal (FICGS) indicates that IBASE is within about 15% of the target 307.2 kHz $\pm$ 25% for two consecutive measurements. ICGS is cleared when FICGS is clear, the internal clock generator is turned off or in STOP (ICGEN is clear), or when IOFF is set.
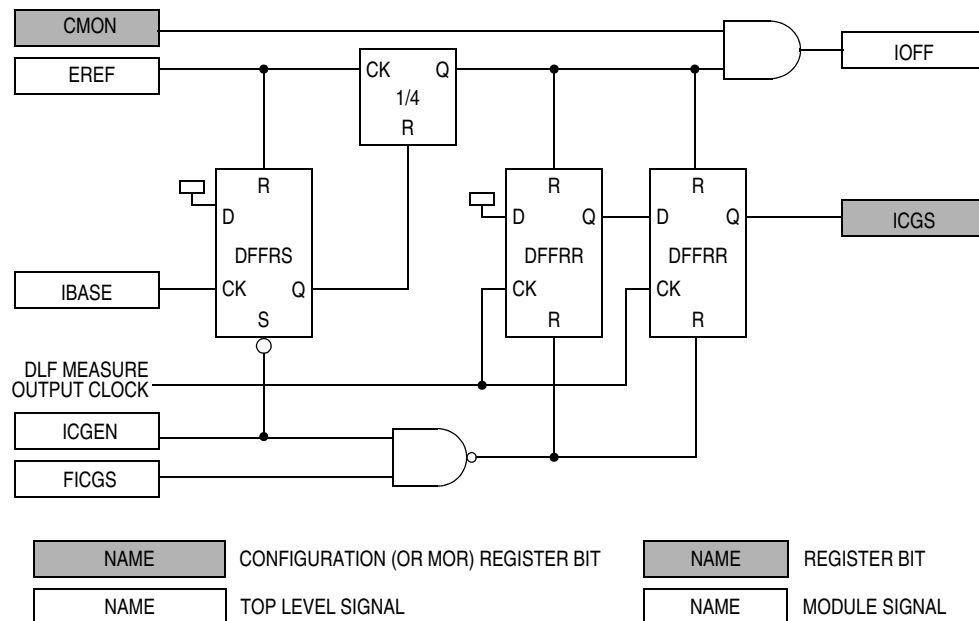
Figure 7-5. Internal Clock Activity Detector

7.3.4.3 External Clock Activity Detector

The external clock activity detector, shown in Figure 7-6, looks for at least one falling edge on the external clock (ECLK) every time the internal reference (IREF) is low. Since IREF is less than half the frequency of ECLK, this should occur every time. If it does not occur two consecutive times, the external clock inactivity indicator (EOFF) is set. EOFF will be cleared the next time there is a falling edge of ECLK while IREF is low.

The external clock stable bit (ECGS) is also generated in the external clock activity detector. ECGS is set on a falling edge of the external stabilization clock (ESTBCLK). This will be 4096 ECLK cycles after the external clock generator on bit is set or the MCU exits STOP (ECGEN = 1) if the external crystal enable (EXTXTALEN) in the CONFIG (or MOR) register is set, or 16 cycles when EXTXTALEN is clear. ECGS is cleared when the external clock generator is turned off or in STOP (ECGEN is clear) or when EOFF is set.

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1
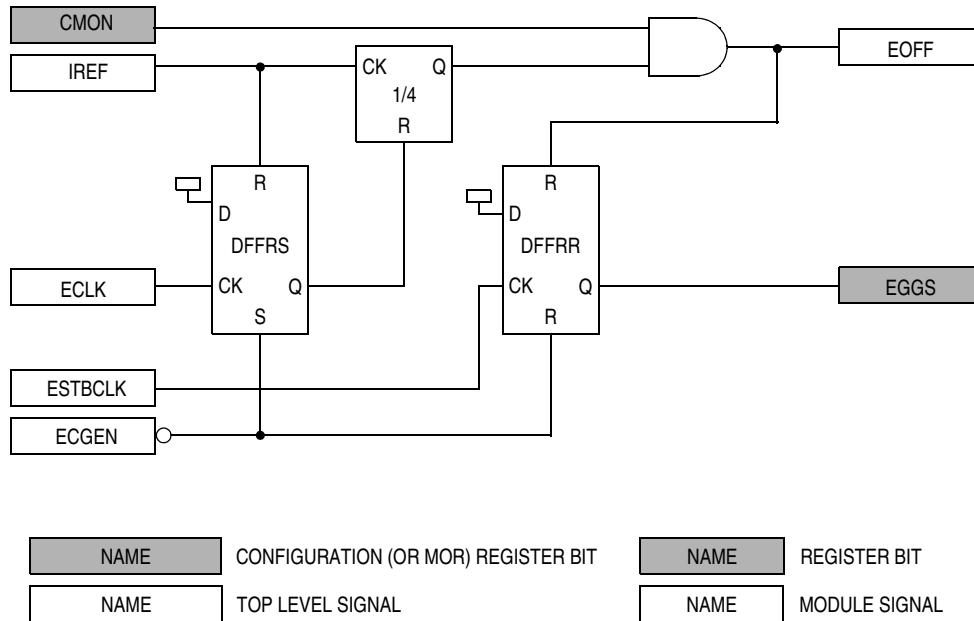
Figure 7-6. External Clock Activity Detector

7.3.5 Clock Selection Circuit

The clock selection circuit, shown in Figure 7-7, contains two clock switches which generate the oscillator output clock (CGMXCLK) and the timebase clock (TBMCLK) from either the internal clock (ICLK) or the external clock (ECLK). The clock selection circuit also contains a divide-by-two circuit which creates the clock generator output clock (CGMOUT), which generates the bus clocks.
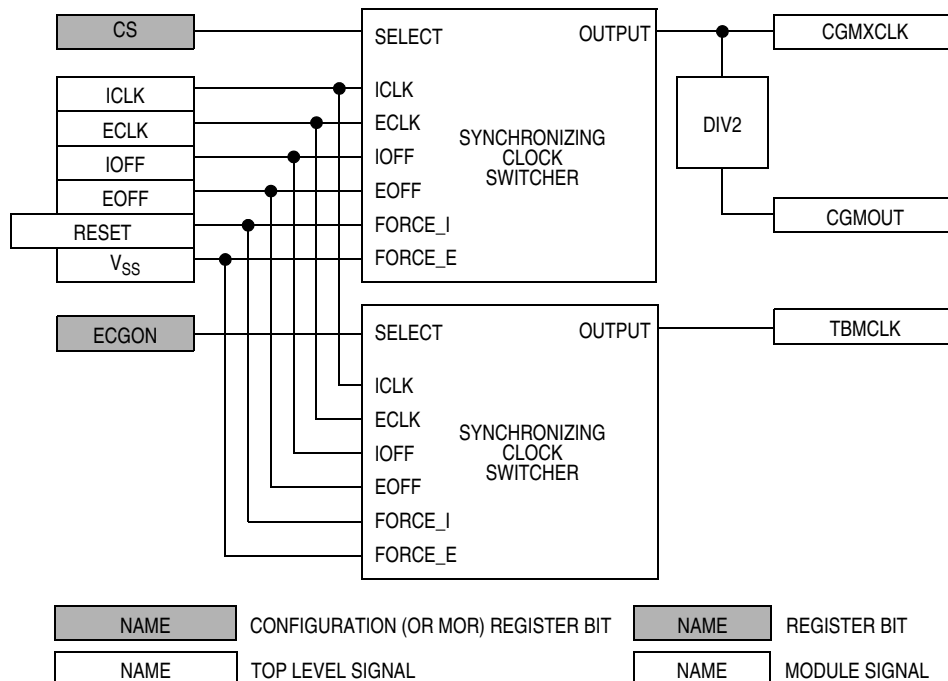
Figure 7-7. Clock Selection Circuit Block Diagram

7.3.5.1 Clock Selection Switches

The first switch creates the oscillator output clock (CGMXCLK) from either the internal clock (ICLK) or the external clock (ECLK), based on the clock select bit (CS set selects ECLK, clear selects ICLK). When switching the CS bit, both ICLK and ECLK must be on (ICGON and ECGON set). The clock being switched to must also be stable (ICGS or ECGS set).

The second switch creates the timebase clock (TBMCLK) from ICLK or ECLK based on the external clock on bit. When ECGON is set, the switch automatically selects the external clock, regardless of the state of the ECGS bit.

7.3.5.2 Clock Switching Circuit

To robustly switch between the internal clock (ICLK) and the external clock (ECLK), the switch assumes the clocks are completely asynchronous, so a synchronizing circuit is required to make the transition. When the select input (the clock select bit for the oscillator output clock switch or the external clock on bit for the timebase clock switch) is changed, the switch will continue to operate off the original clock for between 1 and 2 cycles as the select input is transitioned through one side of the synchronizer. Next, the output will be held low for between 1 and 2 cycles of the new clock as the select input transitions through the other side. Then the output starts switching at the new clock's frequency. This transition guarantees that no glitches will be seen on the output even though the select input may change asynchronously to the clocks. The unpredictability of the transition period is a necessary result of the asynchronicity.

The switch automatically selects ICLK during reset. When the clock monitor is on (CMON is set) and it determines one of the clock sources is inactive (as indicated by the IOFF or EOFF signals), the circuit is forced to select the active clock. There are no clocks for the inactive side of the synchronizer to properly operate, so that side is forced deselected. However, the active side will not be selected until 1 to 2 clock cycles after the IOFF or EOFF signal transitions.

7.4 Usage Notes

The ICG has several features which can provide protection to the microcontroller if properly used. There are other features which can greatly simplify usage if certain techniques are employed. This subsection will describe several possible ways to use the ICG and its features. These techniques are not the only ways to use the ICG, and may not be optimum for all environments. In any case, these techniques should only be used as a template, and the user should modify them according to the application's requirements.

These notes include:

- Switching clock sources
- Enabling the clock monitor
- Using clock monitor interrupts
- Quantization error in DCO output
- Switching internal clock frequencies
- Nominal frequency settling time
- Improving frequency settling time
- Trimming frequency

7.4.1 Switching Clock Sources

Switching from one clock source to another requires both clock sources to be enabled and stable. A simple flow requires:

1. Enable desired clock source
2. Wait for it to become stable
3. Switch clocks
4. Disable previous clock source

The key point to remember in this flow is that the clock source cannot be switched (CS cannot be written) unless the desired clock is on and stable. A short assembly code example of how to employ this flow is shown in Figure 7-8. This code is for illustrative purposes only and does not represent valid syntax for any particular assembler.

```
                        ;Clock Switching Code Example
                        ;This code switches from Internal to External clock
                        ;Clock Monitor and interrupts are not enabled
start   lda     #$13    ;Mask for CS, ECGON, ECGS
                        ; If switching from External to Internal, mask is $0C.
loop    **      **      ;Other code here, such as writing the COP, since ECGS may
                        ; take some time to set
        sta     icgcr   ;Try to set CS, ECGON and clear ICGON. ICGON will not
                        ; clear until CS is set, and CS will not set until
                        ; ECGON and ECGS are set.
        cmpa    icgcr   ;Check to see if ECGS set, then CS set, then ICGON clear
        bne     loop    ;Keep looping until ICGON is clear.
```

Figure 7-8. Code Example for Switching Clock Sources

7.4.2 Enabling the Clock Monitor

Many applications require the clock monitor to determine if one of the clock sources has become inactive, so the other can be used to recover from a potentially dangerous situation. Using the clock monitor requires both clocks to be active (ECGON and ICGON both set). To enable the clock monitor, both clocks must also be stable (ECGS and ICGS both set). This is to prevent the use of the clock monitor when a clock is first turned on and potentially unstable.

Enabling the clock monitor and clock monitor interrupts requires a flow similar to the one below:

1. Enable the alternate clock source
2. Wait for both clock sources to be stable
3. Switch to the desired clock source if necessary
4. Enable the clock monitor
5. Enable clock monitor interrupts

These events must happen in sequence. A short assembly code example of how to employ this flow is shown in Figure 7-9. This code is for illustrative purposes only and does not represent valid syntax for any particular assembler.

```
                                 ;Clock Monitor Enabling Code Example
                                 ;This code turns on both clocks, selects the desired
                                 ; one, then turns on the Clock Monitor and Interrupts
start  lda     #$AF             ;Mask for CMIE, CMON, ICGON, ICGS, ECGON, ECGS
                                 ; If Internal Clock desired, mask is $AF
                                 ; If External Clock desired, mask is $BF
                                 ; If interrupts not desired mask is $2F int; $3F ext
loop   **      **               ;Other code here, such as writing the COP, since ECGS
                                 ; and ICGS may take some time to set.
       sta     icgcr            ;Try to set CMIE. CMIE wont set until CMON set; CMON
                                 ; won't set until ICGON, ICGS, ECGON, ECGS set.
       brset   6,ICGCR,error    ;Verify CMF is not set
       cmpa    icgcr            ;Check if ECGS set, then CMON set, then CMIE set
       bne     loop             ;Keep looping until CMIE is set.
```

Figure 7-9. Code Example for Enabling the Clock Monitor

7.4.3 Using Clock Monitor Interrupts

The clock monitor circuit can be used to recover from perilous situations such as crystal loss. To use the clock monitor effectively, the following notes should be observed:

- Enable the clock monitor and clock monitor interrupts.

- The first statement in the clock monitor interrupt service routine (CMISR) should be a read to the ICG control register (ICGCR) to verify the clock monitor flag (CMF) is set. This is also the first step in clearing the CMF bit.

- The second statement in the CMISR should be a write to the ICGCR to clear the CMF bit (write the bit low). Writing the bit high will not affect it. This statement does not need to immediately follow the first, but must be contained in the CMISR.

- The third statement in the CMISR should be to clear the CMON bit. This is required to ensure proper reconfiguration of the reference dividers. This statement must also be contained in the CMISR.

- Although the clock monitor can only be enabled when both clocks are stable (ICGS is set or ECGS is set), it will remain set if one of the clocks goes unstable.

- The clock monitor only works if the external slow (EXTSLOW) bit in the CONFIG (or MOR) register is set to the correct value.

- The internal and external clocks must both be enabled and running in order to use the clock monitor.

- When the clock monitor detects inactivity, the inactive clock is automatically deselected and the active clock selected as the source for CGMXCLK and TBMCLK. The CMISR can use the state of the CS bit to check which clock is inactive.

- When the clock monitor detects inactivity, the application may have been subjected to extreme conditions which may have affected other circuits. The CMISR should take any appropriate precautions.

7.4.4 Quantization Error in DCO Output

The digitally controlled oscillator (DCO) is comprised of three major sub-blocks:

- Binary weighted divider
- Variable-delay ring oscillator
- Ring oscillator fine-adjust circuit

Each of these blocks affects the clock period of the internal clock (ICLK). Since these blocks are controlled by the digital loop filter (DLF) outputs DDIV and DSTG, the output of the DCO can only change in quantized steps as the DLF increments or decrements its output. The following subsections describe how each block will affect the output frequency.

7.4.4.1 Digitally Controlled Oscillator

The digitally controlled oscillator (DCO) is an inaccurate oscillator which generates the internal clock (ICLK), whose clock period is dependent on the digital loop filter outputs (DSTG[7:0] and DDIV[3:0]). Because of the digital nature of the DCO, the clock period of ICLK will change in quantized steps. This will create a clock period difference, or quantization error (Q-ERR) from one cycle to the next. Over several cycles or for longer periods, this error is divided out until it reaches a minimum error of 0.202% to 0.368%. The dependence of this error on the DDIV[3:0] value and the number of cycles the error is measured over is shown in Table 7-3.

Table 7-3. Quantization Error in ICLK

DDIV[3:0]	ICLK Cycles	Bus Cycles	τ_{ICLK} Q-ERR
%0000 (min)	1	NA	6.45% – 11.8%
%0000 (min)	4	1	1.61% – 2.94%
%0000 (min)	≥ 32	≥ 8	0.202% – 0.368%
%0001	1	NA	3.23% – 5.88%
%0001	4	1	0.806% – 1.47%
%0001	≥ 16	≥ 4	0.202% – 0.368%
%0010	1	NA	1.61% – 2.94%
%0010	4	1	0.403% – 0.735%
%0010	≥ 8	≥ 2	0.202% – 0.368%
%0011	1	NA	0.806% – 1.47%
%0011	≥ 4	≥ 1	0.202% – 0.368%
%0100	1	NA	0.403% – 0.735%
%0100	≥ 2	≥ 1	0.202% – 0.368%
%0101 – %1001 (max)	≥ 1	≥ 1	0.202% – 0.368%

7.4.4.2 Binary Weighted Divider

The binary weighted divider divides the output of the ring oscillator by a power of 2, specified by the DCO divider control bits (DDIV[3:0]). DDIV maximizes at %1001 (values of %1010 through %1111 are interpreted as %1001), which corresponds to a divide by 512. When DDIV is %0000, the ring oscillator's output is divided by 1. Incrementing DDIV by one will double the period; decrementing DDIV will halve the period. The DLF cannot directly increment or decrement DDIV; DDIV is only incremented or decremented when an addition or subtraction to DSTG carries or borrows.

7.4.4.3 Variable-Delay Ring Oscillator

The variable-delay ring oscillator's period is adjustable from 17 to 31 stage delays, in increments of two, based on the upper three DCO stage control bits (DSTG[7:5]). A DSTG[7:5] of %000 corresponds to 17 stage delays; DSTG[7:5] of %111 corresponds to 31 stage delays. Adjusting the DSTG[5] bit has a 6.45% to 11.8% effect on the output frequency. This also corresponds to the size correction made when the frequency error is greater than ±15%. The value of the binary weighted divider does not affect the relative change in output clock period for a given change in DSTG[7:5].

7.4.4.4 Ring Oscillator Fine-Adjust Circuit

The ring oscillator fine-adjust circuit causes the ring oscillator to effectively operate at non-integer numbers of stage delays by operating at two different points for a variable number of cycles specified by the lower five DCO stage control bits (DSTG[4:0]). For example, when DSTG[7:5] is %011, the ring oscillator nominally operates at 23 stage delays. When DSTG[4:0] is %00000, the ring will always operate at 23 stage delays. When DSTG[4:0] is %00001, the ring will operate at 25 stage delays for one of 32 cycles and at 23 stage delays for 31 of 32 cycles. Likewise, when DSTG[4:0] is %11111, the ring operates at 25 stage delays for 31 of 32 cycles and at 23 stage delays for one of 32 cycles. When DSTG[7:5] is %111, similar results are achieved by including a variable divide-by-two, so the ring operates at 31 stages for some cycles and at 17 stage delays, with a divide-by-two for an effective 34 stage delays, for the remainder of the cycles. Adjusting the DSTG[0] bit has a 0.202% to 0.368% effect on the output clock period. This corresponds to the minimum size correction made by the DLF, and the inherent, long term quantization error in the output frequency.

7.4.5 Switching Internal Clock Frequencies

The frequency of the internal clock (ICLK) may need to be changed for some applications. For example, if the reset condition does not provide the correct frequency, or if the clock is slowed down for a low power mode (or sped up after a low-power mode), the frequency must be changed by programming the internal clock multiplier factor (N). The frequency of ICLK is N times the frequency of IBASE, which is 307.2 kHz ± 25%.

Before switching frequencies by changing the N value, the clock monitor must be disabled. This is because when N is changed, the frequency of the low-frequency base clock (IBASE) will change proportionally until the digital loop filter has corrected the error. Since the clock monitor uses IBASE, it could erroneously detect an inactive clock. The clock monitor cannot be re-enabled until the Internal Clock is stable again (ICGS is set).

The following flow is an example of how to change the clock frequency:
- Verify there is no clock monitor Interrupt by reading the CMF bit
- Turn off the clock monitor
- If desired, switch to the external clock (see 7.4.1 Switching Clock Sources)
- Change the value of N
- Switch back to internal (see 7.4.1 Switching Clock Sources), if desired
- Turn on the clock monitor (see 7.4.2 Enabling the Clock Monitor), if desired

7.4.6 Nominal Frequency Settling Time

Because the clock period of the internal clock (ICLK) is dependent on the digital loop filter outputs (DDIV and DSTG) which cannot change instantaneously, ICLK will temporarily operate at an incorrect clock

period when any of the operating condition changes. This happens whenever the part is reset, the ICG multiply factor (N) is changed, the ICG trim factor (TRIM) is changed, or the internal clock is enabled after inactivity (STOP or disabled operation). The time that the ICLK takes to adjust to the correct period is known as the settling time.

Settling time depends primarily on how many corrections it takes to change the clock period, and the period of each correction. Since the corrections require four periods of the low-frequency base clock ($4 * \tau_{IBASE}$), and since ICLK is N (the ICG multiply factor for the desired frequency) times faster than IBASE, each correction takes $4 * N * \tau_{ICLK}$. The period of ICLK, however, will vary as the corrections occur.

7.4.6.1 Settling To Within 15%

All FLASH mask sets other than 0K45D, 1K45D, 0L09H, 1L09H have 15% comparators that improve stability at low temperatures.

When the error is greater than 15%, the filter takes eight corrections to double or halve the clock period. Due to how the DCO increases or decreases the clock period, the total period of these eight corrections is approximately 11 times the period of the fastest correction. (If the corrections were perfectly linear, the total period would be 11.5 times the minimum period; however, the ring must be slightly nonlinear.) Therefore, the total time it takes to double or halve the clock period is $44 * N * t_{ICLKFAST}$.

If the clock period needs more than doubled or halved, the same relationship applies, only for each time the clock period needs doubled, the total number of cycles doubles. That is, when transitioning from fast to slow, going from the initial speed to half speed takes $44 * N * t_{ICLKFAST}$; from half speed to quarter speed takes $88 * N * t_{ICLKFAST}$; going from quarter speed to eighth speed takes $176 * N * t_{ICLKFAST}$; and so on. This series can be expressed as $(2^x - 1) * 44 * N * t_{ICLKFAST}$, where x is the number of times the speed needs doubled or halved. Since 2^x happens to be equal to $\tau_{ICLKSLOW}/\tau_{ICLKFAST}$, the equation reduces to $44 * N * (\tau_{ICLKSLOW} - \tau_{ICLKFAST})$.

NOTE

Increasing speed takes much longer than decreasing speed since N is higher. This can be expressed in terms of the initial clock period (τ_1) minus the final clock period (τ_2) as such:

$$\tau_{15} = \text{abs}[44N(\tau_1 - \tau_2)]$$

Once the clock period is within 15% of the desired clock period, the internal clock stable bit (ICGS) will be set and the clock frequency is usable, although the error will be as high as 15%.

7.4.6.2 Total Settling Time

Once the clock period is within 15% of the desired clock period, the filter starts making minimum adjustments. In this mode, each correction will adjust the frequency between 0.202% and 0.368%. A maximum of 88 corrections will be required to get to the minimum error. Each correction takes approximately the same period of time, or $4 * \tau_{IBASE}$. This makes 88 corrections ($352 * \tau_{IBASE}$) to get from 15% to the minimum error. The total time to the minimum error is:

$$\tau_{tot} = \text{abs}[44N(\tau_1 - \tau_2)] + 352\tau_{IBASE}$$

The equations for τ_{15}, τ_5, and τ_{tot} are dependent on the actual initial and final clock periods τ_1 and τ_2, not the nominal. This means the variability in the ICLK frequency due to process, temperature and voltage must be considered. Additionally, other process factors and noise can affect the actual tolerances of the points at which the filter changes modes. This means a worst case adjustment of up to 35% (ICLK clock

period tolerance plus 10%) must be added. This adjustment can be reduced with trimming. Table 7-4 shows some typical values for settling time.

Table 7-4. Typical Settling Time Examples

τ_1	τ_2	N	τ_{15}	τ_{tot}
1/ (6.45 MHz)	1/ (25.8 MHz)	84	430 μs	1165 μs
1/ (25.8 MHz)	1/ (6.45 MHz)	21	107 μs	840 μs
1/ (25.8 MHz)	1/ (307.2 kHz)	1	141 μs	875 μs
1/ (307.2 kHz)	1/ (25.8 MHz)	84	11.9 ms	12.6 ms

7.4.7 Trimming Frequency on the Internal Clock Generator

The unadjusted frequency of the low-frequency base clock (IBASE), when the comparators in the frequency comparator indicate zero error, will vary as much as ±25% due to process, temperature, and voltage dependencies. These dependencies are in the voltage and current references, the offset of the comparators, and the internal capacitor. The voltage and temperature dependencies have been designed to be a maximum of approximately ±1% error. The process dependencies account for the rest.

Fortunately, for an individual part, the process dependencies are constant. An individual part can operate at approximately ±2% variance from its unadjusted operating point over the entire spec range of the application. If the unadjusted operating point can be changed, the entire variance can be limited to ±2%.

The method of changing the unadjusted operating point is by changing the size of the capacitor. This capacitor is designed with 639 equally sized units. 384 of these units are always connected. The remaining 255 units are put in by adjusting the ICG trim factor (TRIM). The default value for TRIM is $80, or 128 units, making the default capacitor size 512. Each unit added or removed will adjust the output frequency by about ±0.195% of the unadjusted frequency (adding to TRIM will decrease frequency). Therefore, the frequency of IBASE can be changed to ±25% of its unadjusted value, which is enough to cancel the process variability mentioned before.

The best way to trim the internal clock is to use the timer to measure the width of an input pulse on an input capture pin (this pulse must be supplied by the application and should be as long or wide as possible). Considering the prescale value of the timer and the theoretical (zero error) frequency of the bus (307.2 kHz *N/4), the error can be calculated. This error, expressed as a percentage, can be divided by 0.195% and the resultant factor added or subtracted from TRIM. This process should be repeated to eliminate any residual error.

7.5 Low-Power Modes

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

7.5.1 Wait Mode

The ICG remains active in wait mode. If enabled, the ICG interrupt to the CPU can bring the MCU out of wait mode.

In some applications, low power consumption is desired in wait mode and a high frequency clock is not needed. In these applications, reduce power consumption by either selecting a low-frequency external clock and turn the internal clock generator off, or reduce the bus frequency by minimizing the ICG multiplier factor (N) before executing the WAIT instruction.

7.5.2 Stop Mode

The value of the oscillator enable in stop (OSCENINSTOP) bit in the CONFIG (or MOR) register determines the behavior of the ICG in stop mode. If OSCENINSTOP is low, the ICG is disabled in stop and, upon execution of the STOP instruction, all ICG activity will cease and the output clocks (CGMXCLK, CGMOUT, and TBMCLK) will be held low. Power consumption will be minimal.

If OSCENINSTOP is high, the ICG is enabled in stop and activity will continue. This is useful if the timebase module (TBM) is required to bring the MCU out of stop mode. ICG interrupts will not bring the MCU out of stop mode in this case.

During STOP, if OSCENINSTOP is low, several functions in the ICG are affected. The stable bits (ECGS and ICGS) are cleared, which will enable the external clock stabilization divider upon recovery. The clock monitor is disabled (CMON = 0) which will also clear the clock monitor interrupt enable (CMIE) and clock monitor flag (CMF) bits. The CS, ICGON, ECGON, N, TRIM, DDIV, and DSTG bits are unaffected.

7.6 CONFIG (or MOR) Register Options

There are four CONFIG (or MOR) register options that affect the functionality of the ICG. These options are:
- EXTCLKEN (external clock enable)
- EXTXTALEN (external crystal enable)
- EXTSLOW (slow external clock)
- OSCENINSTOP (oscillator enable in stop)

All CONFIG (or MOR) register options will have a default setting.

7.6.1 External Clock Enable (EXTCLKEN)

External clock enable (EXTCLKEN), when set, enables the ECGON bit to be set. ECGON turns on the external clock input path through the PTB6/(OSC1) pin. When EXTCLKEN is clear, ECGON cannot be set and PTB6/(OSC1) will always perform the PTB6 function.

The default state for this option is clear.

7.6.2 External Crystal Enable (EXTXTALEN)

External crystal enable (EXTXTALEN), when set, will enable an amplifier to drive the PTB7/(OSC2)/$\overline{\text{RST}}$ pin from the PTB6/(OSC1) pin. The amplifier will only drive if the external clock enable (EXTCLKEN) bit and the ECGON bit are also set. If EXTCLKEN or ECGON are clear, PTB7/(OSC2)/$\overline{\text{RST}}$ will perform the PTB7 function. When EXTXTALEN is clear, PTB7/(OSC2)/$\overline{\text{RST}}$ will always perform the PTB7 function.

EXTXTALEN, when set, also configures the clock monitor to expect an external clock source in the valid range of crystals (30 kHz to 100 kHz or 1 MHz to 8 MHz). When EXTXTALEN is clear, the clock monitor will expect an external clock source in the valid range for externally generated clocks when using the clock monitor (60 Hz to 32 MHz).

EXTXTALEN, when set, also configures the external clock stabilization divider in the clock monitor for a 4096 cycle time-out to allow the proper stabilization time for a crystal. When EXTXTALEN is clear, the stabilization divider is configured to 16 cycles since an external clock source does not need a start-up time.

The default state for this option is clear.

7.6.3 Slow External Clock (EXTSLOW)

Slow external clock (EXTSLOW), when set, will decrease the drive strength of the oscillator amplifier, enabling low-frequency crystal operation (30 kHz–100 kHz) if properly enabled with the external clock enable (EXTCLKEN) and external crystal enable (EXTXTALEN) bits. When clear, EXTSLOW enables high-frequency crystal operation (1 MHz to 8 MHz).

EXTSLOW, when set, also configures the clock monitor to expect an external clock source that is slower than the low-frequency base clock (60 Hz–307.2 kHz). When EXTSLOW is clear, the clock monitor will expect an external clock faster than the low-frequency base clock (307.2 kHz–32 MHz).

The default state for this option is clear.

7.6.4 Oscillator Enable In Stop (OSCENINSTOP)

Oscillator enable in stop (OSCENINSTOP), when set, will enable the ICG to continue to generate clocks (either CGMXCLK, CGMOUT, or TBMCLK) in stop mode. This function is used to keep the timebase running while the rest of the microcontroller stops. When OSCENINSTOP is clear, all clock generation will cease and CGMXCLK, CGMOUT, and TBMCLK will be forced low during stop.

The default state for this option is clear.

7.7 I/O Registers

The ICG contains five registers, summarized in Figure 7-10. These registers are:

- ICG control register
- ICG multiplier register
- ICG trim register
- ICG DCO divider control register
- ICG DCO stage control register

Several of the bits in these registers have interaction where the state of one bit may force another bit to a particular state or prevent another bit from being set or cleared. A summary of this interaction is shown in Table 7-5.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0035	ICG Control Register (ICGCR) See page 87.	Read:	CMIE	CMF	CMON	CS	ICGON	ICGS	ECGON	ECGS
		Write:		0[1]						
		Reset:	0	0	0	0	1	0	0	0

1. See CMF bit description for method of clearing.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0037	ICG Multiplier Register (ICGMR) See page 88.	Read:		N6	N5	N4	N3	N2	N1	N0
		Write:								
		Reset:	0	0	0	1	0	1	0	1
$0038	ICG Trim Register (ICGTR) See page 89.	Read:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
		Write:								
		Reset:	1	0	0	0	0	0	0	0

☐ = Unimplemented   R = Reserved   U = Unaffected

Figure 7-10. ICG Module I/O Register Summary

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Freescale Semiconductor · 85

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0039	ICG DCO Divider Control Register (ICGDVR) See page 89.	Read:					DDIV3	DDIV2	DDIV1	DDIV0
		Write:								
		Reset:	0	0	0	0	U	U	U	U
$003A	ICG DCO Stage Control Register (ICGDSR) See page 89.	Read:	DSTG7	DSTG6	DSTG5	DSTG4	DSTG3	DSTG2	DSTG1	DSTG0
		Write:	R	R	R	R	R	R	R	R
		Reset:	U	U	U	U	U	U	U	U

= Unimplemented   R = Reserved   U = Unaffected

Figure 7-10. ICG Module I/O Register Summary (Continued)

Table 7-5. ICG Module Register Bit Interaction Summary

Condition	Register Bit Results for Given Condition											
	CMIE	CMF	CMON	CS	ICGON	ICGS	ECGON	ECGS	N[6:0]	TRIM[7:0]	DDIV[3:0]	DSTB[7:0]
Reset	0	0	0	0	1	0	0	0	$15	$80	—	—
OSCENINSTOP = 0, STOP = 1	0	0	0	—	—	0	—	0	—	—	—	—
EXTCLKEN = 0	0	0	0	0	1	—	0	0	—	—	uw	uw
CMF = 1	—	(1)	1	—	1	—	1	—	uw	uw	uw	uw
CMON = 0	0	0	(0)	—	—	—	—	—	—	—	—	—
CMON = 1	—	—	(1)	—	1	—	1	—	uw	uw	uw	uw
CS = 0	—	—	—	(0)	1	—	—	—	—	—	uw	uw
CS = 1	—	—	—	(1)	—	—	1	—	—	—	—	—
ICGON = 0	0	0	0	1	(0)	0	1	—	—	—	—	—
ICGON = 1	—	—	—	—	(1)	—	—	—	—	—	uw	uw
ICGS = 0	us	—	us	uc	—	(0)	—	—	—	—	—	—
ECGON = 0	0	0	0	0	1	—	(0)	0	—	—	uw	uw
ECGS = 0	us	—	us	us	—	—	—	(0)	—	—	—	—
IOFF = 1	—	1*	(1)	1	(1)	0	(1)	—	uw	uw	uw	uw
EOFF = 1	—	1*	(1)	0	(1)	—	(1)	0	uw	uw	uw	uw
N = written	(0)	(0)	(0)	—	—	0*	—	—	—	—	—	—
TRIM = written	(0)	(0)	(0)	—	—	0*	—	—	—	—	—	—

—Register bit is unaffected by the given condition.
0, 1Register bit is forced clear or set (respectively) in the given condition.
0*, 1*Register bit is temporarily forced clear or set (respectively) in the given condition.
(0), (1)Register bit must be clear or set (respectively) for the given condition to occur.
us, uc, uwRegister bit cannot be set, cleared, or written (respectively) in the given condition.

7.7.1 ICG Control Register

The ICG control register (ICGCR) contains the control and status bits for the internal clock generator, external clock generator, and clock monitor as well as the Clock Select and Interrupt Enable bits.

Address: $0036

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CMIE	CMF	CMON	CS	ICGON	ICGS	ECGON	ECGS
Write:		0[(1)]						
Reset:	0	0	0	0	1	0	0	0

▨ = Unimplemented   1. See CMF bit description for method of clearing
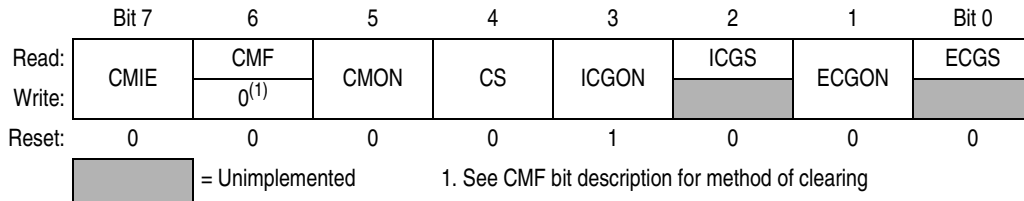
Figure 7-11. ICG Control Register (ICGCR)

CMIE — Clock Monitor Interrupt Enable Bit
This read/write bit enables clock monitor interrupts. An interrupt will occur when both CMIE and CMF are set. CMIE can be set when the CMON bit has been set for at least one cycle. CMIE is forced clear when CMON is clear or during reset.
  1 = Clock monitor interrupts enabled
  0 = Clock monitor interrupts disabled

CMF — Clock Monitor Interrupt Flag
This read-only bit is set when the clock monitor determines that either ICLK or ECLK becomes inactive and the CMON bit is set. This bit is cleared by first reading the bit while it is set, followed by writing the bit low. This bit is forced clear when CMON is clear or during reset.
  1 = Either ICLK or ECLK have become inactive
  0 = ICLK and ECLK have not become inactive since the last read of the ICGCR, or the clock monitor
      is disabled

CMON — Clock Monitor On Bit
This read/write bit enables the clock monitor. CMON can be set when both ICLK and ECLK have been on and stable for at least one bus cycle (ICGON, ECGON, ICGS, and ECGS are all set). CMON is forced set when CMF is set, to avoid inadvertent clearing of CMF. CMON is forced clear when either ICGON or ECGON are clear, during STOP with OSCENINSTOP low, or during reset.
  1 = Clock monitor output enabled
  0 = Clock monitor output disabled

CS — Clock Select Bit
This read/write bit determines which clock will generate the oscillator output clock (CGMXCLK). This bit can be set when ECGON and ECGS have been set for at least one bus cycle and can be cleared when ICGON and ICGS have been set for at least one bus cycle. This bit is forced set when the clock monitor determines the internal clock (ICLK) is inactive or when ICGON is clear. This bit is forced clear when the clock monitor determines that the external clock (ECLK) is inactive, when ECGON is clear, or during reset.
  1 = External clock (ECLK) sources CGMXCLK
  0 = Internal clock (ICLK) sources CGMXCLK

ICGON — Internal Clock Generator On Bit
This read/write bit enables the internal clock generator. ICGON can be cleared when the CS bit has been set and the CMON bit has been clear for at least one bus cycle. ICGON is forced set when the CMON bit is set, the CS bit is clear, or during reset.
    1 = Internal clock generator enabled
    0 = Internal clock generator disabled

ICGS — Internal Clock Generator Stable Bit
This read-only bit indicates when the internal clock generator has determined that the internal clock (ICLK) is within about 15% of the desired value. This bit is forced clear when the clock monitor determines the ICLK is inactive, when ICGON is clear, when the ICG multiplier register (ICGMR) is written, when the ICG trim register (ICGTR) is written, during STOP with OSCENINSTOP low, or during reset.
    1 = Internal clock is within 15% of the desired value
    0 = Internal clock may not be within 15% of the desired value

ECGON — External Clock Generator On Bit
This read/write bit enables the external clock generator. ECGON can be cleared when the CS and CMON bits have been clear for at least one bus cycle. ECGON is forced set when the CMON bit or the CS bit is set. ECGON is forced clear during reset.
    1 = External clock generator enabled
    0 = External clock generator disabled

ECGS — External Clock Generator Stable Bit
This read-only bit indicates when at least 4096 external clock (ECLK) cycles have elapsed since the external clock generator was enabled. This is not an assurance of the stability of ECLK but is meant to provide a start-up delay. This bit is forced clear when the clock monitor determines ECLK is inactive, when ECGON is clear, during STOP with OSCENINSTOP low, or during reset.
    1 = 4096 ECLK cycles have elapsed since ECGON was set
    0 = External clock is unstable, inactive, or disabled

7.7.2 ICG Multiplier Register

Address: $0037

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:		N6	N5	N4	N3	N2	N1	N0
Reset:	0	0	0	1	0	1	0	1

☐ = Unimplemented
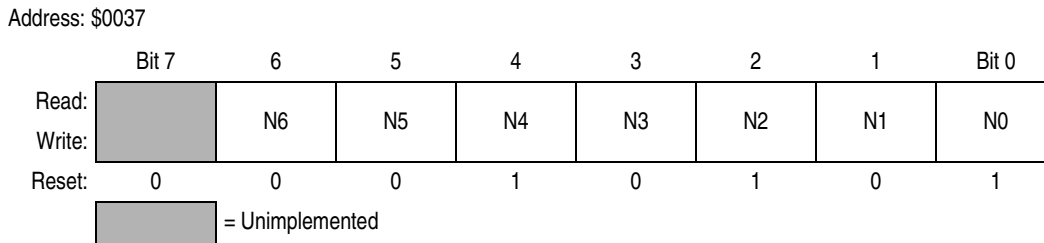
Figure 7-12. ICG Multiplier Register (ICGMR)

N6:N0 — ICG Multiplier Factor Bits
These read/write bits change the multiplier used by the internal clock generator. The internal clock (ICLK) will be (307.2 kHz ± 25%) * N. A value of $00 in this register is interpreted the same as a value of $01. This register cannot be written when the CMON bit is set. Reset sets this factor to $15 (decimal 21) for default frequency of 6.45 MHz ± 25% (1.613 MHz ± 25% bus).

7.7.3 ICG Trim Register

Address: $0038

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
Reset:	1	0	0	0	0	0	0	0

            = Unimplemented
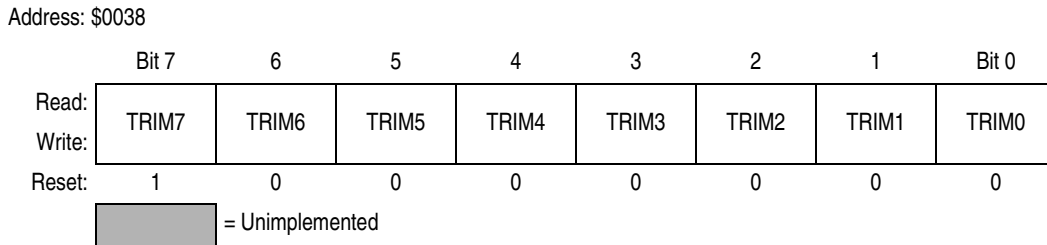
Figure 7-13. ICG Trim Register (ICGTR)

TRIM7:TRIM0 — ICG Trim Factor Bits
These read/write bits change the size of the internal capacitor used by the internal clock generator. By testing the frequency of the internal clock and incrementing or decrementing this factor accordingly, the accuracy of the internal clock can be improved to ±2%. Incrementing this register by one decreases the frequency by 0.195% of the unadjusted value. Decrementing this register by one increases the frequency by 0.195%. This register cannot be written when the CMON bit is set. Reset sets these bits to $80, centering the range of possible adjustment.

7.7.4 ICG DCO Divider Register

Address: $0039

	Bit 7	6	5	4	3	2	1	Bit 0
Read:					DDIV3	DDIV2	DDIV1	DDIV0
Write:								
Reset:	0	0	0	0	U	U	U	U

        = Unimplemented   U = Undefined
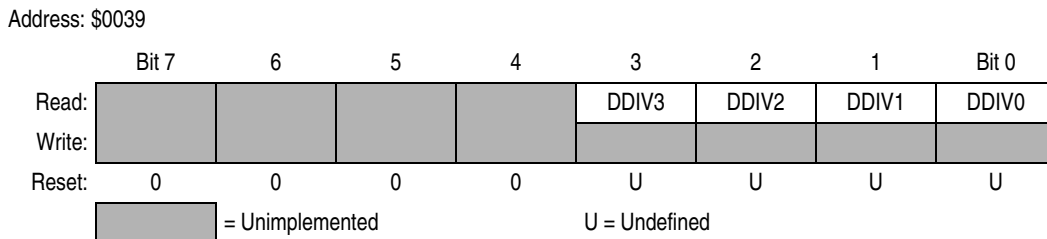
Figure 7-14. ICG DCO Divider Control Register (ICGDVR)

DDIV3:DDIV0 — ICG DCO Divider Control Bits
These bits indicate the number of divide-by-twos (DDIV) that follow the digitally controlled oscillator. When ICGON is set, DDIV is controlled by the digital loop filter. The range of valid values for DDIV is from $0 to $9. Values of $A–$F are interpreted the same as $9. Since the DCO is active during reset, reset has no effect on DSTG and the value may vary.

7.7.5 ICG DCO Stage Register

Address: $003A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	DSTG7	DSTG6	DSTG5	DSTG4	DSTG3	DSTG2	DSTG1	DSTG0
Write:	R	R	R	R	R	R	R	R
Reset:	U	U	U	U	U	U	U	U

      = Unimplemented   R   = Reserved   U = Unaffected
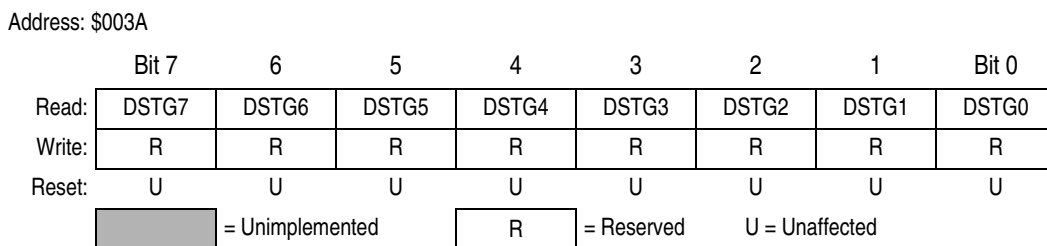
Figure 7-15. ICG DCO Stage Control Register (ICGDSR)

DSTG7:DSTG0 — ICG DCO Stage Control Bits

These bits indicate the number of stages (above the minimum) in the digitally controlled oscillator. The total number of stages is approximately equal to $1FF, so changing DSTG from $00 to $FF will approximately double the period. Incrementing DSTG will increase the period (decrease the frequency) by 0.202% to 0.368% (decrementing has the opposite effect). DSTG cannot be written when ICGON is set to prevent inadvertent frequency shifting. When ICGON is set, DSTG is controlled by the digital loop filter. Since the DCO is active during reset, reset has no effect on DSTG and the value may vary.

Chapter 8
External Interrupt (IRQ)

8.1 Introduction

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

8.2 Features

Features of the IRQ module include:

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

8.3 Functional Description

A logic 0 applied to the external interrupt pin can latch a central processor unit (CPU) interrupt request. Figure 8-2 shows the structure of the IRQ module.

Interrupt signals on the $\overline{\text{IRQ1}}$ 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 1 to the ACK1 bit clears the IRQ1 latch.
- Reset — A reset automatically clears the interrupt latch.

The external interrupt pin is falling-edge triggered and is software- configurable to be both falling-edge and 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 latch 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 latch remains set until both of these occur:

- Vector fetch or software clear
- Return of the interrupt pin to logic 1
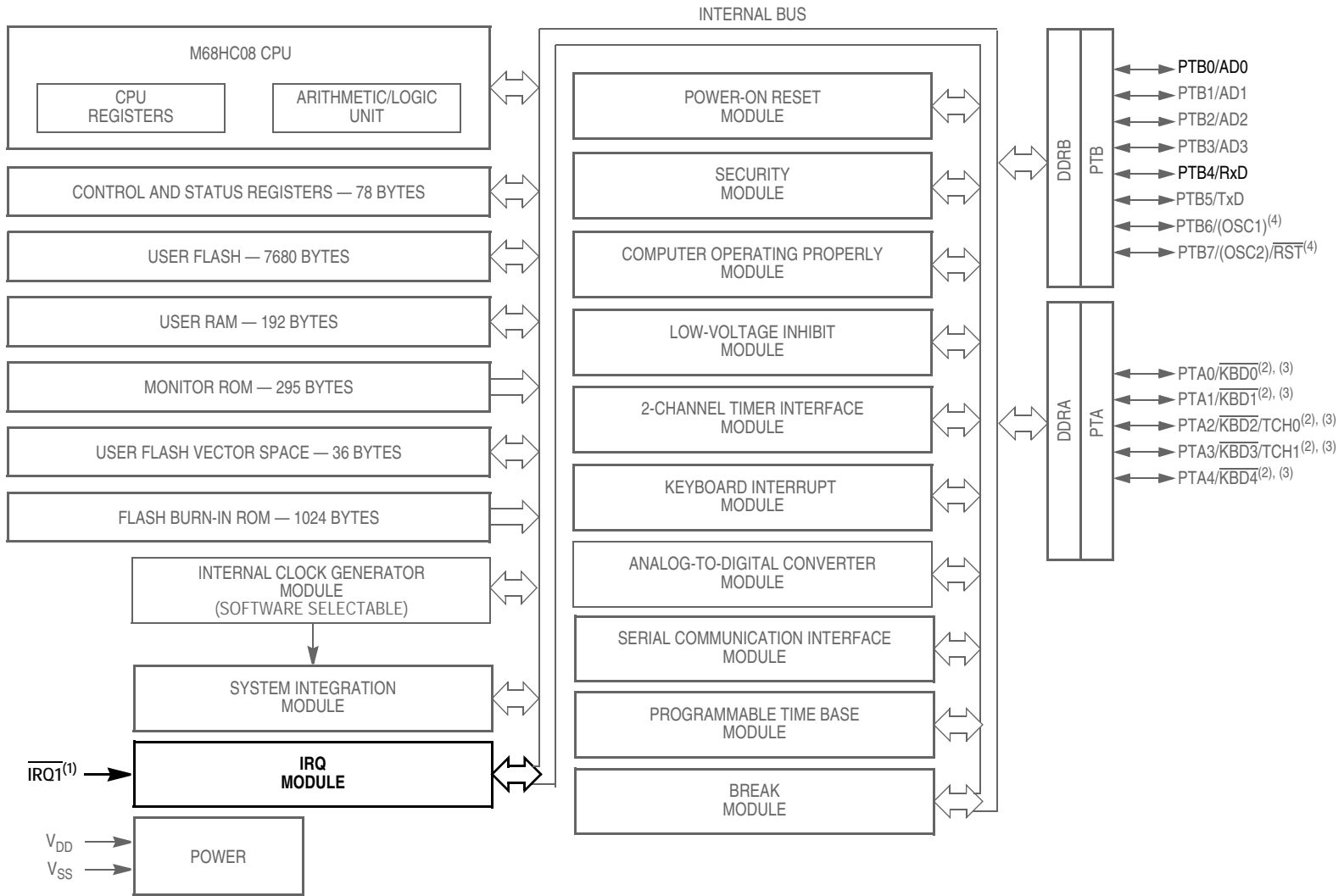
Figure 8-1. Block Diagram Highlighting IRQ Block and Pins

Notes:
1. Pin contains integrated pullup resistor
2. High-current source/sink pin
3. Pin contains software selectable pullup resistor if general function I/O pin is configured as input.
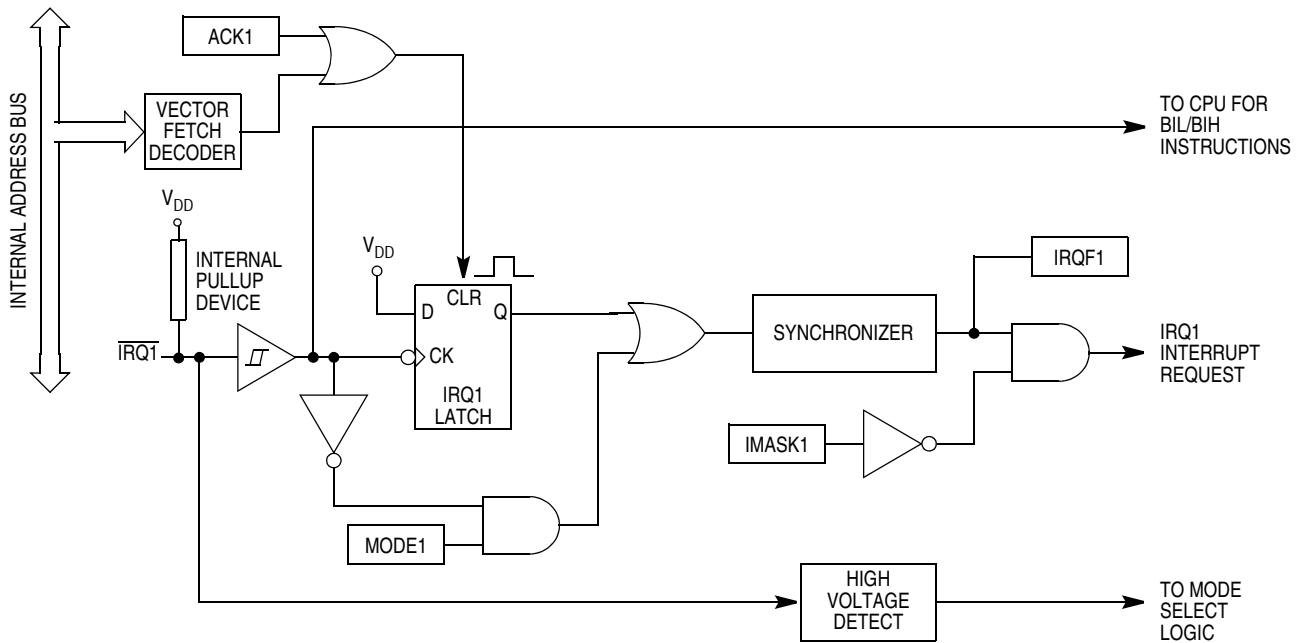
Figure 8-2. IRQ Block Diagram

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.*

8.4 $\overline{\text{IRQ1}}$ 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 the following actions must occur to clear the IRQ1 latch:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the latch. Software may generate the interrupt acknowledge signal by writing a 1 to the 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 can also prevent spurious interrupts due to noise. Setting ACK1 does not affect subsequent transitions on the $\overline{\text{IRQ1}}$ pin. A falling edge on the $\overline{\text{IRQ1}}$ pin that occurs after writing to the ACK1 bit latches 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, the IRQ1 latch remains set.

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

The vector fetch or software clear and the return of the $\overline{IRQ1}$ pin to logic 1 can occur in any order. The interrupt request remains pending as long as the $\overline{IRQ1}$ 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 $\overline{IRQ1}$ 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 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 branch if interrupt pin is high (BIH) or branch if interrupt pin is low (BIL) instruction to read the logic level on the $\overline{IRQ1}$ pin.

NOTE

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

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

Address: $001D

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

R = Reserved   U = Unaffected

Figure 8-3. IRQ Status and Control Register (ISCR)

IRQF1 — IRQ1 Flag Bit
This read-only status bit is high when the IRQ1 interrupt is pending.
1 = IRQ1 interrupt pending
0 = IRQ1 interrupt not pending

ACK1 — IRQ1 Interrupt Request Acknowledge Bit
Writing a 1 to this write-only bit clears the IRQ1 latch. ACK1 always reads as 0. Reset clears ACK1.

IMASK1 — IRQ1 Interrupt Mask Bit
Writing a 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 $\overline{IRQ1}$ pin. Reset clears MODE1.
1 = IRQ1 interrupt requests on falling edges and low levels
0 = IRQ1 interrupt requests on falling edges only

Chapter 9
Keyboard Interrupt Module (KBI)

9.1 Introduction

The keyboard interrupt module (KBI) provides five independently maskable external interrupt pins.

9.2 Features

KBI features include:

- Five keyboard interrupt pins, on the MC68HC08KX8, are with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Hysteresis buffers
- Programmable edge-only or edge- and level-interrupt sensitivity
- Automatic interrupt acknowledge
- Exit from low-power modes
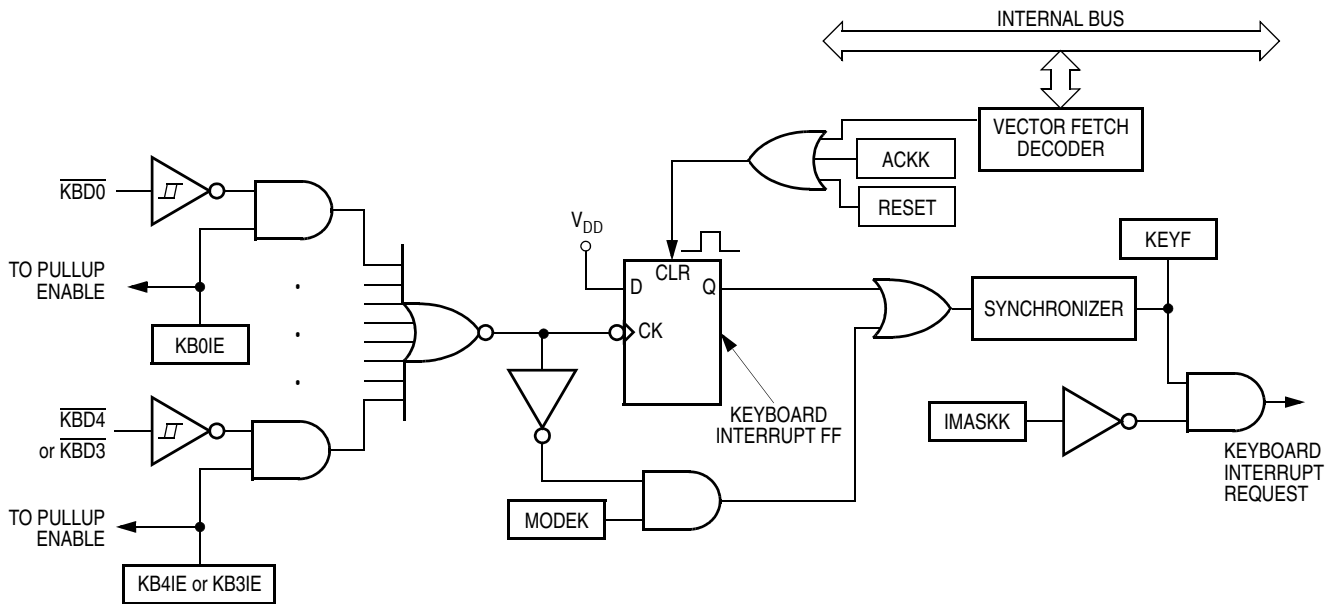
Figure 9-1. Keyboard Module Block Diagram
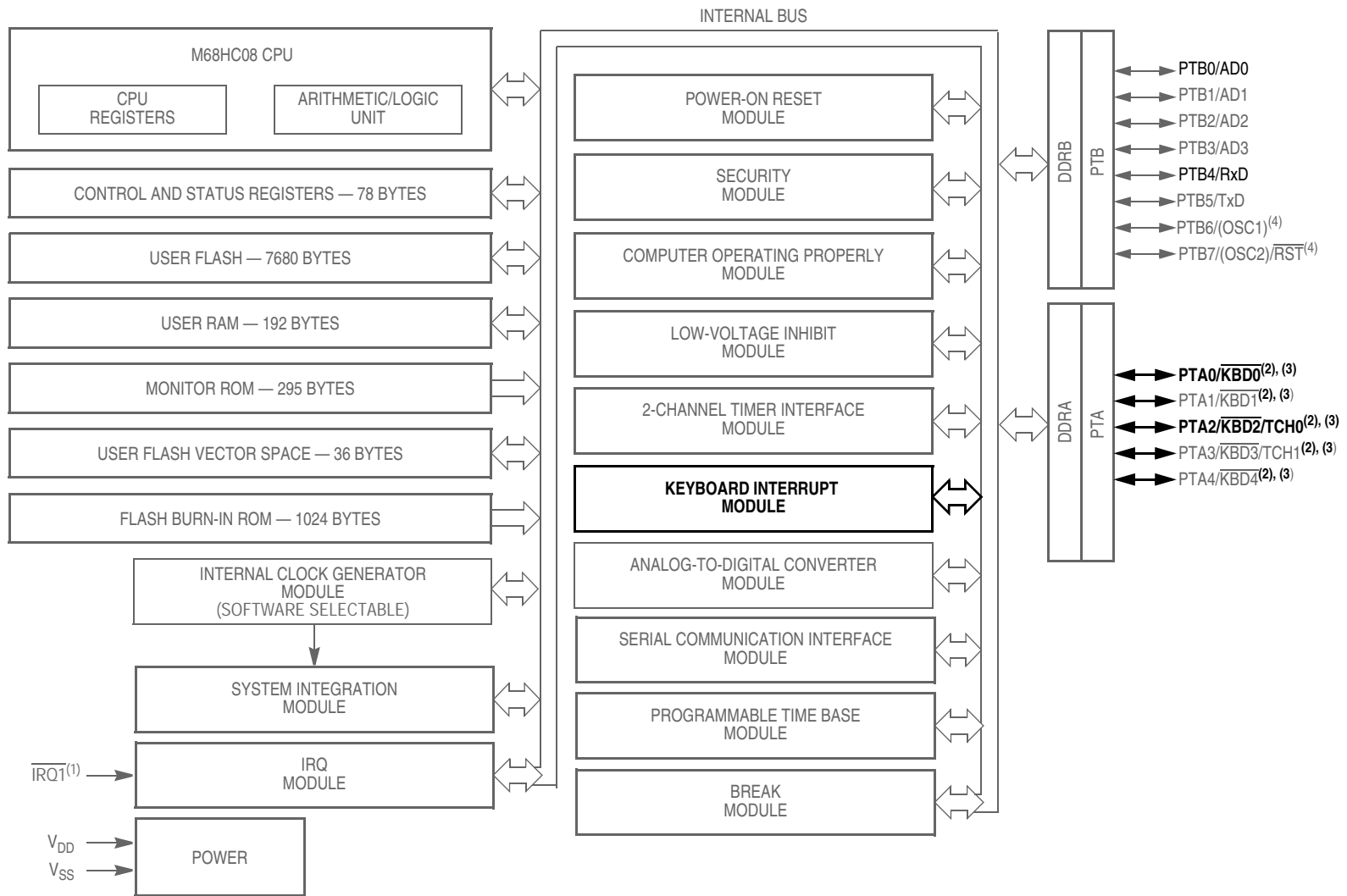
Notes:
1. Pin contains integrated pullup resistor
2. High-current source/sink pin
3. Pin contains software selectable pullup resistor if general function I/O pin is configured as input.

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

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001A	Keyboard Status and Control Register (KBSCR) See page 99.	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$001B	Keyboard Interrupt Enable Register (KBIER) See page 100.	Read:	0	0	0	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

☐ = Unimplemented

Figure 9-3. I/O Register Summary

9.3 Functional Description

Writing to the KBIE4–KBIE0 bits in the keyboard interrupt enable register independently enables or disables each port A 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 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 also can prevent spurious interrupts due to noise. Setting ACKK does not affect subsequent transitions on the keyboard interrupt pins. A falling edge that occurs after writing to the ACKK bit latches another interrupt request. If the keyboard interrupt mask bit, IMASKK, is clear, the CPU loads the program counter with the vector address at locations $FFE0 and $FFE1.

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

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

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

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

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

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

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

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

9.5 Low-Power Modes

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

9.5.1 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.

9.5.2 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.

9.6 I/O Registers

Two registers control and monitor operation of the keyboard module:

- • Keyboard status and control register, KBSCR
- • Keyboard interrupt enable register, KBIER

9.6.1 Keyboard Status and Control Register

The keyboard status and control register (KBSCR):

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

Address: $001A

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

= Unimplemented

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

Bits 7–4 — Not used

These read-only bits always read as 0s.

KEYF — Keyboard Flag Bit

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

    1 = Keyboard interrupt pending
    0 = No keyboard interrupt pending

ACKK — Keyboard Acknowledge Bit

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

IMASKK — Keyboard Interrupt Mask Bit

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

9.6.2 Keyboard Interrupt Enable Register

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

Address: $001B

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

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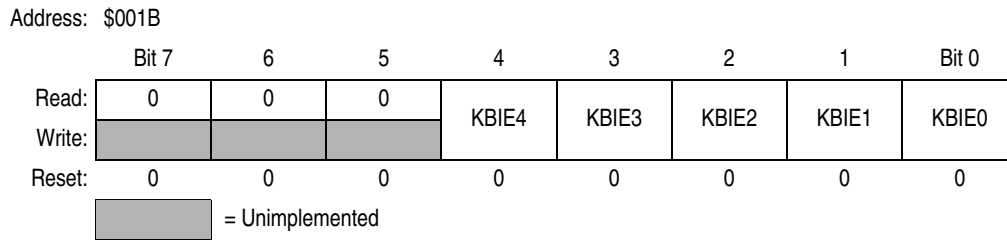
Figure 9-5. Keyboard Interrupt Enable Register (KBIER)

KBIE4–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 = PAx pin enabled as keyboard interrupt pin
0 = PAx pin not enabled as keyboard interrupt pin

Chapter 10
Low-Voltage Inhibit (LVI)

10.1 Introduction

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

10.2 Features

Features of the LVI module include:
- Programmable LVI reset
- Programmable power consumption
- Selectable LVI trip voltage
- Programmable stop mode operation

10.3 Functional Description

Figure 10-1 shows the structure of the LVI module. LVISTOP, LVIPWRD, LVI5OR3, and LVIRSTD are user selectable options found in the configuration register (CONFIG1). See Chapter 4 Configuration Register (CONFIG).
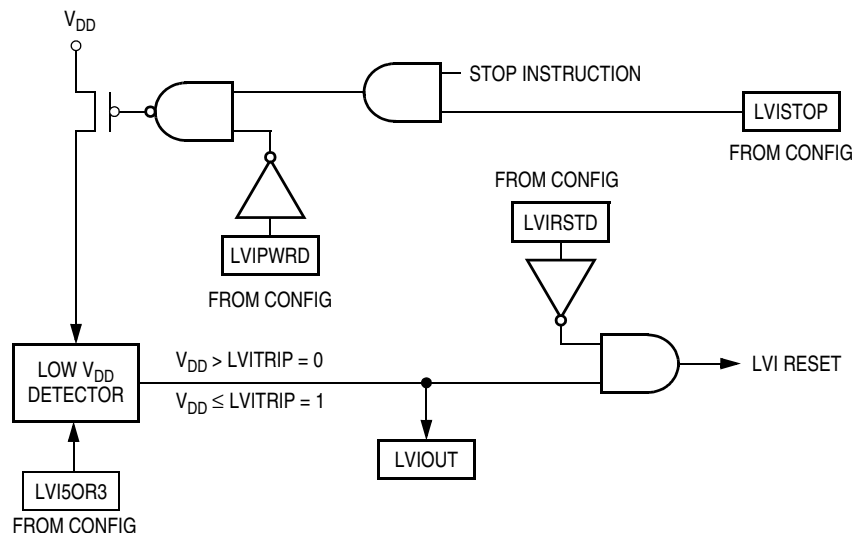
Figure 10-1. LVI Module Block Diagram

The LVI is enabled out of reset. The LVI module contains a bandgap reference circuit and comparator. Clearing the LVI power disable bit, LVIPWRD, enables the LVI to monitor V_{DD} voltage. Clearing the LVI reset disable bit, LVIRSTD, enables the LVI module to generate a reset when V_{DD} falls below a voltage, V_{TRIPF}. Setting the LVI enable in stop mode bit, LVISTOP, enables the LVI to operate in stop mode.

Setting the LVI 5-V or 3-V trip point bit, LVI5OR3, enables the trip point voltage, V_{TRIPF}, to be configured for 5-V operation. Clearing the LVI5OR3 bit enables the trip point voltage, V_{TRIPF}, to be configured for 3-V operation. The actual trip thresholds are specified in 17.5 5.0-Vdc DC Electrical Characteristics and .

> ***NOTE***
> *After a power-on reset, the LVI's default mode of operation is 3 volts. If a 5-V system is used, the user must set the LVI5OR3 bit to raise the trip point to 5-V operation.*
>
> *If the user requires 5-V mode and sets the LVI5OR3 bit after power-on reset while the V_{DD} supply is not above the V_{TRIPR} for 5-V mode, the MCU will immediately go into reset. The next time the LVI releases the reset, the supply will be above the V_{TRIPR} for 5-V mode.*

Once an LVI reset occurs, the MCU remains in reset until V_{DD} rises above a voltage, V_{TRIPR}, which causes the MCU to exit reset. See Chapter 13 System Integration Module (SIM) for the reset recovery sequence.

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

10.3.1 Polled LVI Operation

In applications that can operate at V_{DD} levels below the V_{TRIPF} level, software can monitor V_{DD} by polling the LVIOUT bit. In the configuration register, the LVIPWRD bit must be at 0 to enable the LVI module, and the LVIRSTD bit must be at 1 to disable LVI resets.

10.3.2 Forced Reset Operation

In applications that require V_{DD} to remain above the V_{TRIPF} level, enabling LVI resets allows the LVI module to reset the MCU when V_{DD} falls below the V_{TRIPF} level. In the configuration register, the LVIPWRD and LVIRSTD bits must be at 0 to enable the LVI module and to enable LVI resets.

10.3.3 Voltage Hysteresis Protection

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

10.3.4 LVI Trip Selection

The LVI5OR3 bit in the configuration register selects whether the LVI is configured for 5-V or 3-V protection.

> ***NOTE***
> *The microcontroller is guaranteed to operate at a minimum supply voltage. The trip point (V_{TRIPF} [5 V] or V_{TRIPF} [3 V]) may be lower than this. See 17.5 5.0-Vdc DC Electrical Characteristics and   for the actual trip point voltages.*

10.4 LVI Status Register

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

Address: $FE0C

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

= Unimplemented   R   = Reserved
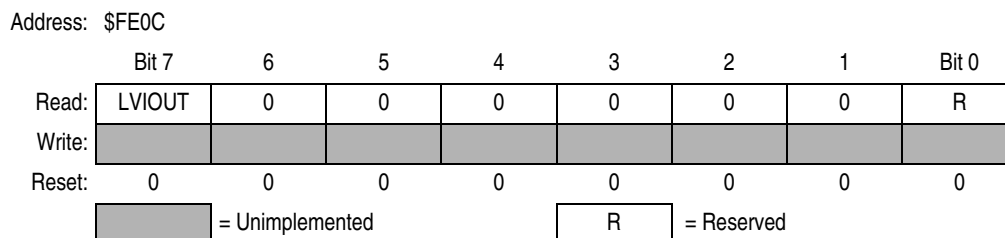
Figure 10-2. LVI Status Register (LVISR)

LVIOUT — LVI Output Bit

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

Table 10-1. LVIOUT Bit Indication

V_{DD}	LVIOUT
$V_{DD} > V_{TRIPR}$	0
$V_{DD} < V_{TRIPF}$	1
$V_{TRIPF} < V_{DD} < V_{TRIPR}$	Previous value

10.5 LVI Interrupts

The LVI module does not generate interrupt requests.

10.6 Low-Power Modes

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

10.6.1 Wait Mode

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

10.6.2 Stop Mode

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

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Chapter 11
Input/Output (I/O) Ports (PORTS)

11.1 Introduction

Thirteen bidirectional input/output (I/O) pins form two parallel ports in the 16-pin plastic dual in-line package (PDIP) and small outline integrated circuit (SOIC) package in the MC68HC908KX8 part. All I/O pins are programmable as inputs or outputs. Port A has software selectable pullup resistors if the port is used as a general-function input port.

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.

See Figure 11-1 for a summary of the I/O port registers.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA) See page 106.	Read:	0	0	0	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:				Unaffected by reset				
$0001	Port B Data Register (PTB) See page 108.	Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:				Unaffected by reset				
$0004	Data Direction Register A (DDRA) See page 106.	Read:	0	0	0	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB) See page 109.	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000D	Port A Input Pullup Enable Register (PTAPUE) See page 108.	Read:	0	0	0	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-1. I/O Port Register Summary

11.2 Port A

Port A is a 5-bit special function port on the MC68HC908KX8 that shares all of its pins with the keyboard interrupt module (KBI) and the 2-channel timer. Port A contains software programmable pullup resistors enabled when a port pin is used as a general-function input. Port A pins are also high-current port pins with 15-mA source/15-mA sink capabilities.

11.2.1 Port A Data Register

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

Address: $0000

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	PTA4	PTA3	PTA2	PTA1	PTA0
Write:								
Reset:				Unaffected by reset				
Alternate Function:				KBD4	KBD3	KBD2	KBD1	KBD0
Alternate Function:				VREFH	TCH1	TCH0		

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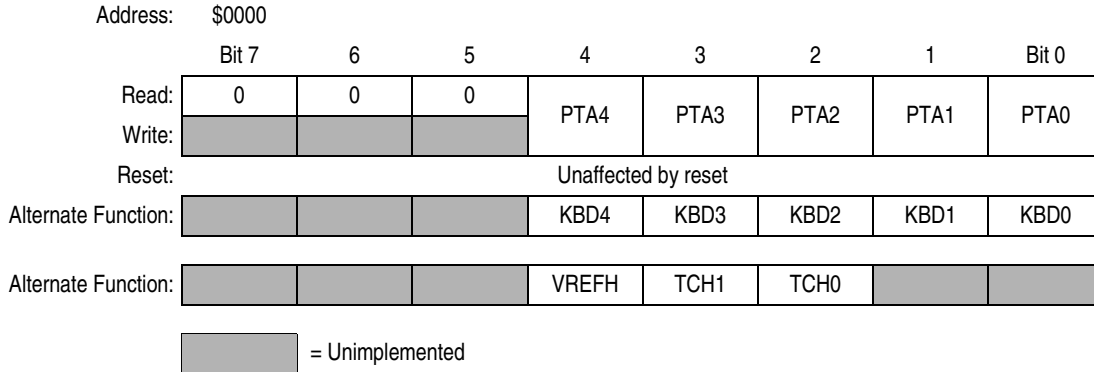
Figure 11-2. Port A Data Register (PTA)

PTA4–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.

KBD4–KBD0 — Keyboard Wakeup Bits
The keyboard interrupt enable bits, KBIE4–KBIE0, in the keyboard interrupt control register, enable the port A pins as external interrupt pins. See Chapter 9 Keyboard Interrupt Module (KBI).

TCH1 and TCH0 — Timer Channel I/O Bits
The PTA3/$\overline{KBD3}$/TCH1 and PTA2/$\overline{KBD2}$/TCH0 pins are the TIM input capture/output compare pins. The edge/level select bits, ELSxB and ELSxA, determine whether the pins are timer channel I/O pins or general-purpose I/O pins. See Chapter 9 Keyboard Interrupt Module (KBI).

11.2.2 Data Direction Register A

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

Address: $0004

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

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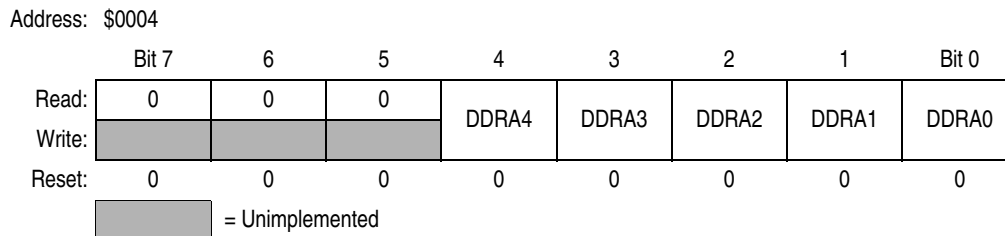
Figure 11-3. Data Direction Register A (DDRA)

DDRA4–DDRA0 — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA4–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.
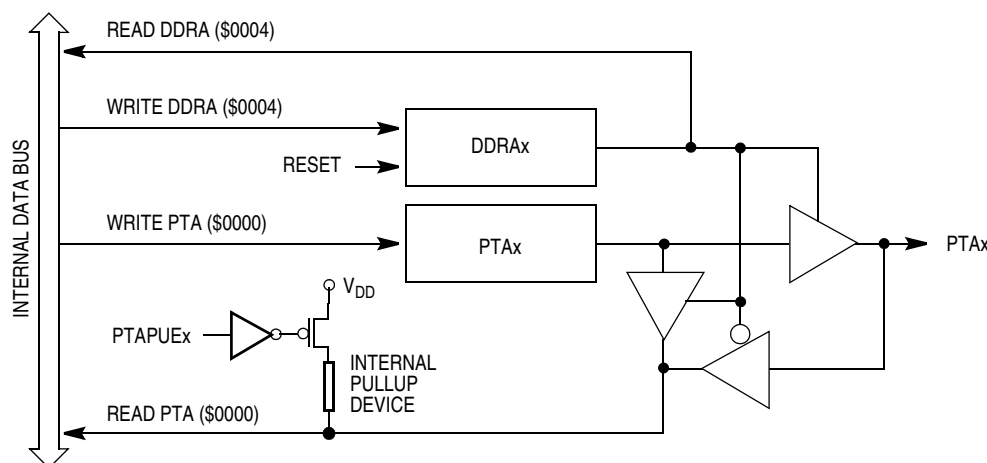
Figure 11-4. Port A I/O Circuit

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

Table 11-1. Port A Pin Functions

PTAPUE Bit	DDRA Bit	PTA Bit	I/O Pin Mode	Accesses to DDRA	Accesses to PTA	
				Read/Write	Read	Write
1	0	X	Input, V_{DD}[1]	DDRA4–DDRA0	Pin	PTA4–PTA0[2]
0	0	X	Input, Hi-Z	DDRA4–DDRA0	Pin	PTA4–PTA0[3]
X	1	X	Output	DDRA4–DDRA0	PTA4–PTA0	PTA4–PTA0

X = Don't care
Hi-Z = High impedance

1. I/O pin pulled up to V_{DD} by internal pulllup device
2. Writing affects data register, but does not affect input.

11.2.3 Port A Input Pullup Enable Register

The port A input pullup enable register (PTAPUE) contains a software configurable pullup device for each of the five port A pins. Each bit is individually configurable and requires that the data direction register, DDRA, bit be configured as an input. Each pullup is automatically disabled when a port bit's DDRA is configured for output mode.

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Address: $000D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
Write:								
Reset:	0	0	0	0	0	0	0	0

☐ = Unimplemented

Figure 11-5. Port A Input Pullup Enable Register (PTAPUE)

PTAPUE4–PTAPUE0 — Port A Input Pullup Enable Bits

These writable bits are software programmable to enable pullup devices on an input port bit.

1 = Corresponding port A pin configured to have internal pullup
0 = Corresponding port A pin has internal pullup disconnected

11.3 Port B

Port B is an 8-bit special-function port that shares four of its pins with the analog-to-digital converter module (ADC), two with the serial communication interface module (SCI) and two with an optional external clock source.

11.3.1 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				
Alternate Function:	OSC2	OSC1	TxD	RxD	AD3	AD2	AD1	AD0

Figure 11-6. 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.

OSC2 and OSC1 — OSC2 and OSC1 Bits

Under software control, PTB7 and PTB6 can be configured as external clock inputs and outputs. PTB7 will become an output clock, OSC2, if selected in the configuration registers and enabled in the ICG registers. PTB6 will become an external input clock source, OSC1, if selected in the configuration registers and enabled in the ICG registers. See Chapter 7 Internal Clock Generator Module (ICG) and Chapter 4 Configuration Register (CONFIG).

RxD — SCI Receive Data Input Bit

The PTB1/RxD pin is the receive data input for the SCI module. When the enable SCI bit, ENSCI, is clear, the SCI module is disabled, and the PTB1/RxD pin is available for general-purpose I/O. See Chapter 12 Serial Communications Interface Module (SCI).

TxD — SCI Transmit Data Output Bit
The PTB0/TxD pin is the transmit data output for the SCI module. When the enable SCI bit, ENSCI, is clear, the SCI module is disabled, and the PTB0/TxD pin is available for general-purpose I/O. See Chapter 12 Serial Communications Interface Module (SCI).

AD3–AD0 — Analog-to-Digital Input Bits
AD3–AD0 are pins used for the input channels to the analog-to-digital converter (ADC) module. The channel select bits in the ADC status and control register define which port B pin will be used as an ADC input and overrides any control from the port I/O logic by forcing that pin as the input to the analog circuitry. See Chapter 3 Analog-to-Digital Converter (ADC).

11.3.2 Data Direction Register B

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

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
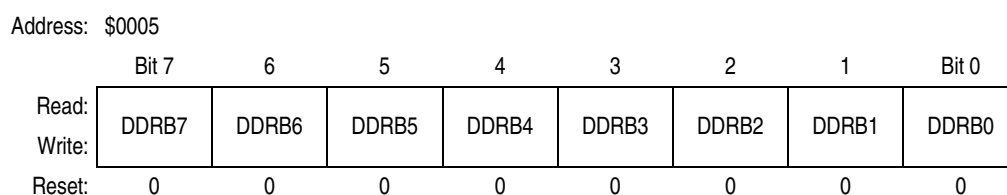
Figure 11-7. Data Direction Register B (DDRB)

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 11-8 shows the port B I/O logic.
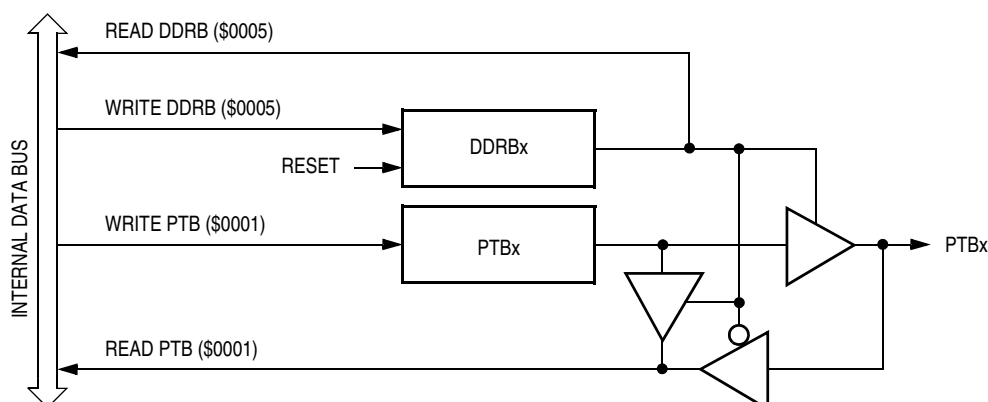
Figure 11-8. Port B I/O Circuit

When bit DDRBx is a 1, reading address $0001 reads the PTBx data latch. When bit DDRBx is a 0, reading address $0001 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 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	Input, Hi-Z	DDRB7–DDRB0	Pin	PTB7–PTB0[1]
1	X	Output	DDRB7–DDRB0	PTB7–PTB0	PTB7–PTB0

X = Don't care
Hi-Z = High impedance

1. Writing affects data register, but does not affect input.

Chapter 12
Serial Communications Interface Module (SCI)

12.1 Introduction

The serial communications interface (SCI) allows asynchronous communications with peripheral devices and other microcontroller unit (MCU).

12.2 Features

The SCI module's features include:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- Choice of baud rate clock source:
  – Internal bus clock
  – CGMXCLK
- 32 programmable baud rates
- Programmable 8-bit or 9-bit character length
- Separately enabled transmitter and receiver
- Separate receiver and transmitter central processor unit (CPU) interrupt requests
- Programmable transmitter output polarity
- Two receiver wakeup methods:
  – Idle line wakeup
  – Address mark wakeup
- Interrupt-driven operation with eight interrupt flags:
  – Transmitter empty
  – Transmission complete
  – Receiver full
  – Idle receiver input
  – Receiver overrun
  – Noise error
  – Framing error
  – Parity error
- Receiver framing error detection
- Hardware parity checking
- 1/16 bit-time noise detection
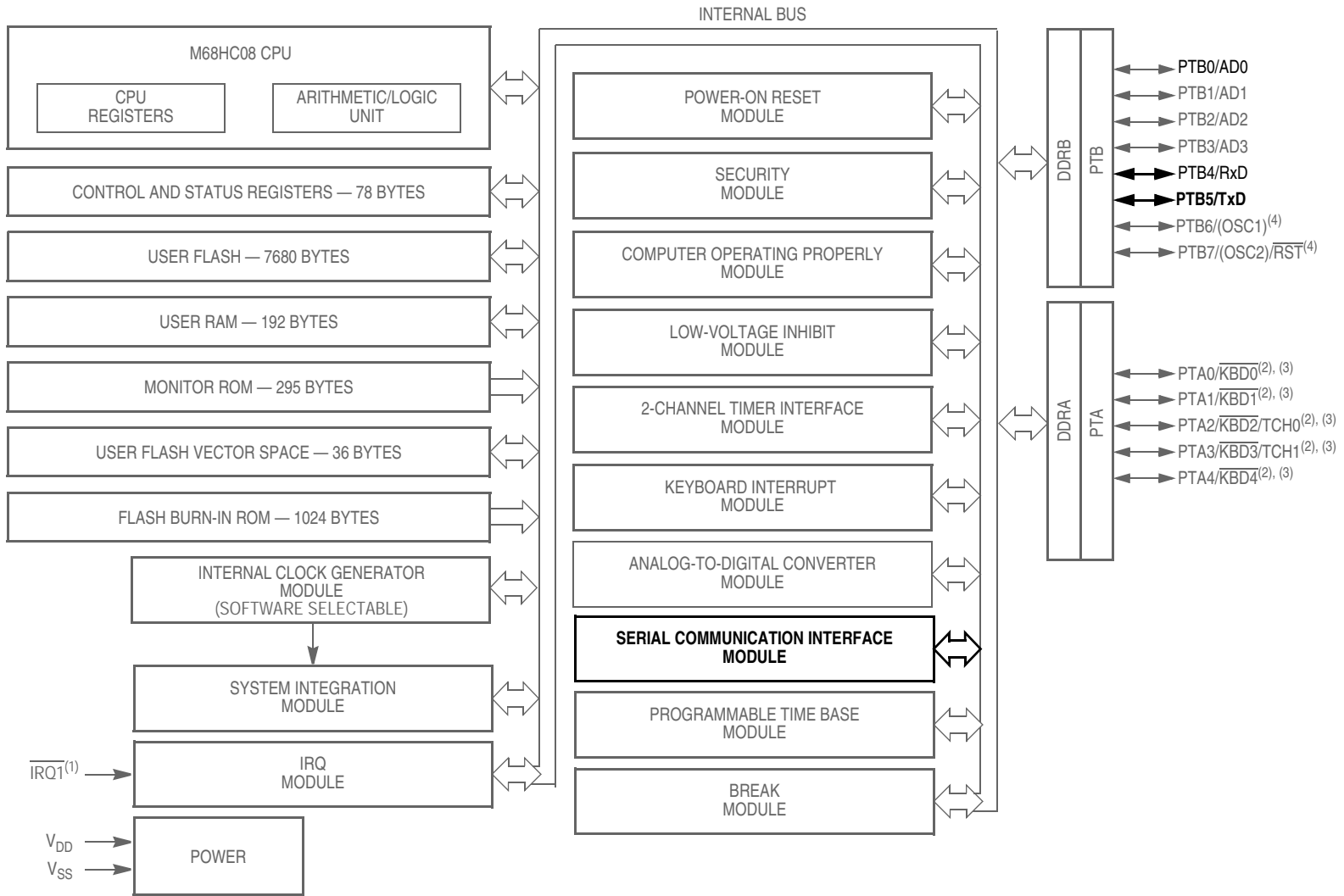
Notes:
 1. Pin contains integrated pullup resistor
 2. High-current source/sink pin
 3. Pin contains software selectable pullup resistor if general function I/O pin is configured as input.

Figure 12-1. Block Diagram Highlighting SCI Block and Pins

12.3 Pin Name Conventions

The generic names of the SCI input/output (I/O) pins are:

- RxD, receive data
- TxD, transmit data

SCI I/O lines are implemented by sharing parallel I/O port pins. The full name of an SCI input or output reflects the name of the shared port pin. Table 12-1 shows the full names and the generic names of the SCI I/O pins.The generic pin names appear in the text of this section.

Table 12-1. Pin Name Conventions

Generic Pin Names	RxD	TxD
Full Pin Names	PTB4/RxD	PTB5/TxD

12.4 Functional Description

Figure 12-3 shows the structure of the SCI module. The SCI allows full-duplex, asynchronous, NRZ serial communication between the MCU and remote devices, including other MCUs. The transmitter and receiver of the SCI operate independently, although they use the same baud rate generator.

The source of the baud rate clock is determined by the configuration register 2 bit, SCIBDSRC. If SCIBDSRC is set then the source of the SCI is the internal data bus clock. If SCIBDSRC is cleared, the source of the SCI is oscillator output CGMXCLK.

During normal operation, the CPU monitors the status of the SCI, writes the data to be transmitted, and processes received data.

12.4.1 Data Format

The SCI uses the standard non-return-to-zero mark/space data format illustrated in Figure 12-2.

Figure 12-2. SCI Data Formats

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Figure 12-3. SCI Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0013	SCI Control Register 1 (SCC1)	Read: Write:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
	See page 125.	Reset:	0	0	0	0	0	0	0	0
$0014	SCI Control Register 2 (SCC2)	Read: Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
	See page 127.	Reset:	0	0	0	0	0	0	0	0
$0015	SCI Control Register 3 (SCC3)	Read: Write:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
	See page 129.	Reset:	U	U	0	0	0	0	0	0
$0016	SCI Status Register 1 (SCS1)	Read: Write:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
	See page 130.	Reset:	1	1	0	0	0	0	0	0
$0017	SCI Status Register 2 (SCS2)	Read: Write:	0	0	0	0	0	0	BKF	RPF
	See page 132.	Reset:	0	0	0	0	0	0	0	0
$0018	SCI Data Register (SCDR)	Read: Write:	R7 T7	R6 T6	R5 T5	R4 T4	R3 T3	R2 T2	R1 T1	R0 T0
	See page 133.	Reset:				Unaffected by reset				
$0019	SCI Baud Rate Register (SCBR)	Read: Write:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
	See page 133.	Reset:	0	0	0	0	0	0	0	0

☐ = Unimplemented   R = Reserved   U = Unaffected

Figure 12-4. SCI I/O Register Summary

12.4.2 Transmitter

Figure 12-5 shows the structure of the SCI transmitter.

12.4.2.1 Character Length

The transmitter can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCC1) determines character length. When transmitting 9-bit data, bit T8 in SCI control register 3 (SCC3) is the ninth bit (bit 8).

12.4.2.2 Character Transmission

During an SCI transmission, the transmit shift register shifts a character out to the TxD pin. The SCI data register (SCDR) is the write-only buffer between the internal data bus and the transmit shift register. To initiate an SCI transmission:

1. Enable the SCI by writing a 1 to the enable SCI bit (ENSCI) in SCI control register 1 (SCC1).
2. Enable the transmitter by writing a 1 to the transmitter enable bit (TE) in SCI control register 2 (SCC2).
3. Clear the SCI transmitter empty bit by first reading SCI status register 1 (SCS1) and then writing to the SCDR.
4. Repeat step 3 for each subsequent transmission.
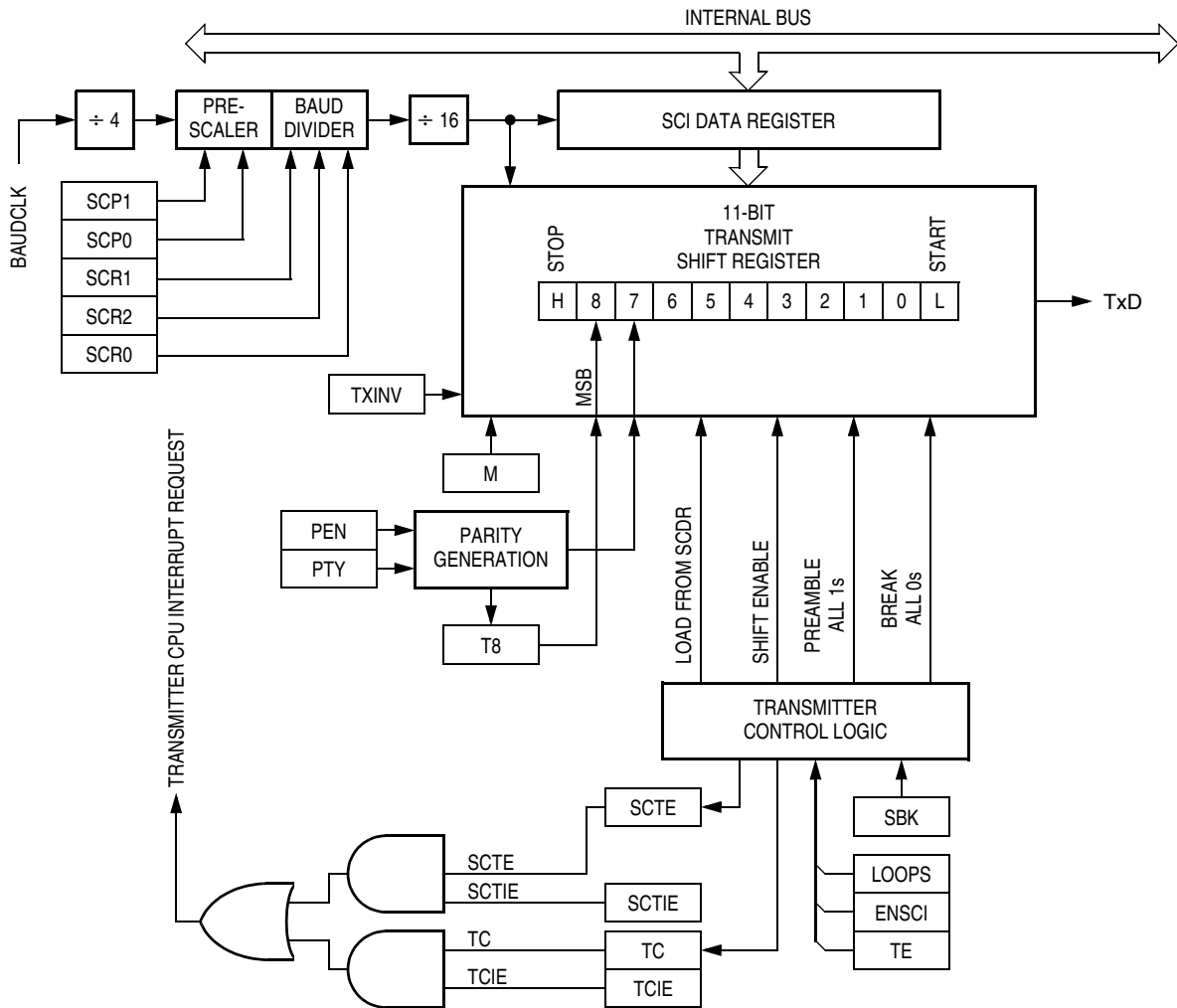
Figure 12-5. SCI Transmitter Break Characters

At the start of a transmission, transmitter control logic automatically loads the transmit shift register with a preamble of logic 1s. After the preamble shifts out, control logic transfers the SCDR data into the transmit shift register. A logic 0 start bit automatically goes into the least significant bit position of the transmit shift register. A logic 1 stop bit goes into the most significant bit position.

The SCI transmitter empty bit, SCTE, in SCS1 becomes set when the SCDR transfers a byte to the transmit shift register. The SCTE bit indicates that the SCDR can accept new data from the internal data bus. If the SCI transmit interrupt enable bit, SCTIE, in SCC2 is also set, the SCTE bit generates a transmitter CPU interrupt request.

When the transmit shift register is not transmitting a character, the TxD pin goes to the idle condition, logic 1. If at any time software clears the ENSCI bit in SCI control register 1 (SCC1), the transmitter and receiver relinquish control of the port B pins.

Writing a 1 to the send break bit, SBK, in SCC2 loads the transmit shift register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on the M bit in SCC1. As long as SBK is at 1, transmitter logic continuously loads break characters into the transmit shift register. After software clears the SBK bit, the shift register finishes transmitting the last

break character and then transmits at least one 1. The automatic 1 at the end of a break character guarantees the recognition of the start bit of the next character.

12.4.2.3 Break Characters

The SCI recognizes a break character when a start bit is followed by eight or nine 0 data bits and a 0 where the stop bit should be. Receiving a break character has these effects on SCI registers:

- Sets the framing error bit (FE) in SCS1
- Sets the SCI receiver full bit (SCRF) in SCS1
- Clears the SCI data register (SCDR)
- Clears the R8 bit in SCC3
- Sets the break flag bit (BKF) in SCS2
- May set the overrun (OR), noise flag (NF), parity error (PE), or reception-in-progress flag (RPF) bits

12.4.2.4 Idle Characters

An idle character contains all 1s and has no start, stop, or parity bit. Idle character length depends on the M bit in SCC1. The preamble is a synchronizing idle character that begins every transmission.

If the TE bit is cleared during a transmission, the TxD pin becomes idle after completion of the transmission in progress. Clearing and then setting the TE bit during a transmission queues an idle character to be sent after the character currently being transmitted.

> **NOTE**
> *When queueing an idle character, return the TE bit to 1 before the stop bit of the current character shifts out to the TxD pin. Setting TE after the stop bit appears on TxD causes data previously written to the SCDR to be lost.*
>
> *A good time to toggle the TE bit for a queued idle character is when the SCTE bit becomes set and just before writing the next byte to the SCDR.*

12.4.2.5 Inversion of Transmitted Output

The transmit inversion bit (TXINV) in SCI control register 1 (SCC1) reverses the polarity of transmitted data. All transmitted values, including idle, break, start, and stop bits, are inverted when TXINV is at 1. See 12.7.1 SCI Control Register 1.

12.4.2.6 Transmitter Interrupts

These conditions can generate CPU interrupt requests from the SCI transmitter:

- SCI transmitter empty (SCTE) — The SCTE bit in SCS1 indicates that the SCDR has transferred a character to the transmit shift register. SCTE can generate a transmitter CPU interrupt request. Setting the SCI transmit interrupt enable bit, SCTIE, in SCC2 enables the SCTE bit to generate transmitter CPU interrupt requests.
- Transmission complete (TC) — The TC bit in SCS1 indicates that the transmit shift register and the SCDR are empty and that no break or idle character has been generated. The transmission complete interrupt enable bit, TCIE, in SCC2 enables the TC bit to generate transmitter CPU interrupt requests.

12.4.3 Receiver

Figure 12-6 shows the structure of the SCI receiver.
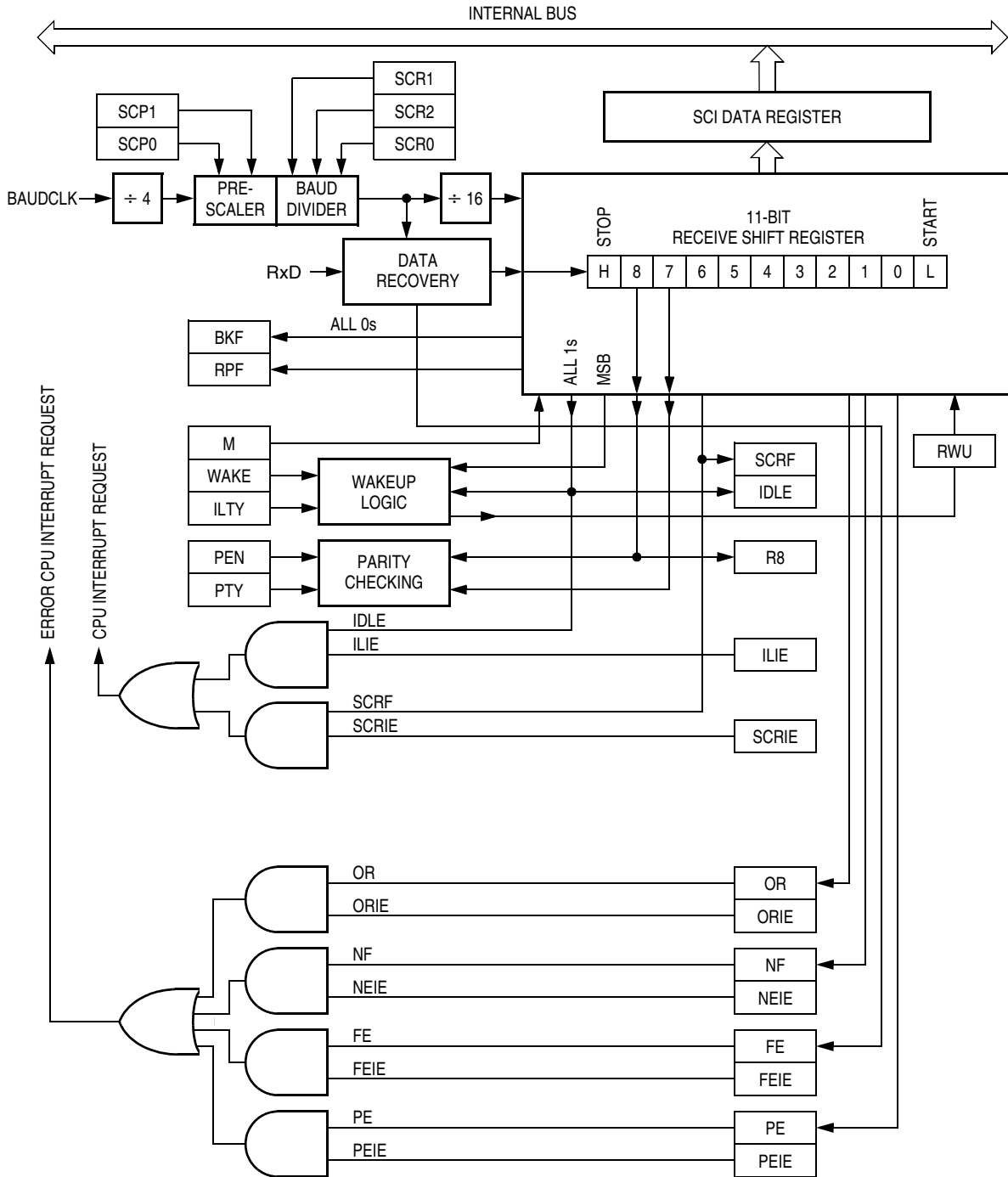
Figure 12-6. SCI Receiver Block Diagram

12.4.3.1 Character Length

The receiver can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCC1) determines character length. When receiving 9-bit data, bit R8 in SCI control register 2 (SCC2) is the ninth bit (bit 8). When receiving 8-bit data, bit R8 is a copy of the eighth bit (bit 7).

12.4.3.2 Character Reception

During an SCI reception, the receive shift register shifts characters in from the RxD pin. The SCI data register (SCDR) is the read-only buffer between the internal data bus and the receive shift register.

After a complete character shifts into the receive shift register, the data portion of the character transfers to the SCDR. The SCI receiver full bit, SCRF, in SCI status register 1 (SCS1) becomes set, indicating that the received byte can be read. If the SCI receive interrupt enable bit, SCRIE, in SCC2 is also set, the SCRF bit generates a receiver CPU interrupt request.

12.4.3.3 Data Sampling

The receiver samples the RxD pin at the RT clock rate. The RT clock is an internal signal with a frequency 16 times the baud rate. To adjust for baud rate mismatch, the RT clock is resynchronized at these times (see Figure 12-7):

- After every start bit
- After the receiver detects a data bit change from 1 to 0 (after the majority of data bit samples at RT8, RT9, and RT10 returns a valid 1 and the majority of the next RT8, RT9, and RT10 samples returns a valid 0)

To locate the start bit, data recovery logic does an asynchronous search for a 0 preceded by three 1s. When the falling edge of a possible start bit occurs, the RT clock begins to count to 16.
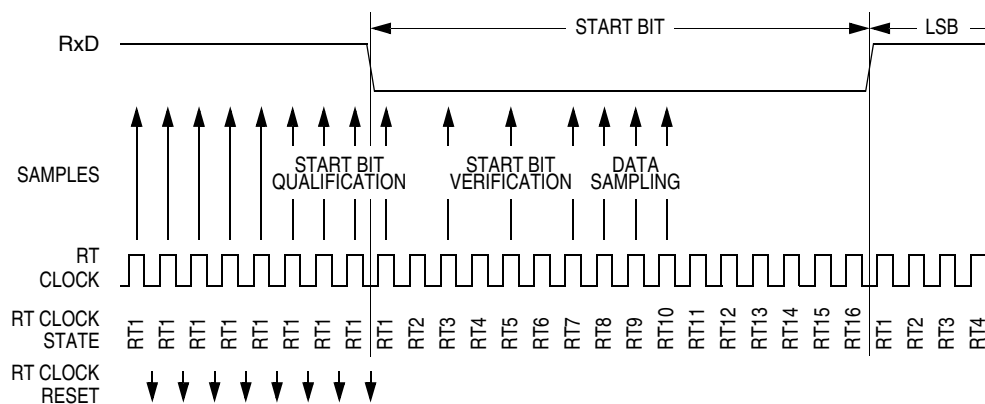
Figure 12-7. Receiver Data Sampling

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7.

Table 12-2 summarizes the results of the start bit verification samples.

Table 12-2. Start Bit Verification

RT3, RT5, and RT7 Samples	Start Bit Verification	Noise Flag
000	Yes	0
001	Yes	1
010	Yes	1
011	No	0
100	Yes	1
101	No	0
110	No	0
111	No	0

If start bit verification is not successful, the RT clock is reset and a new search for a start bit begins.

To determine the value of a data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 12-3 summarizes the results of the data bit samples.

Table 12-3. Data Bit Recovery

RT8, RT9, and RT10 Samples	Data Bit Determination	Noise Flag
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

NOTE
The RT8, RT9, and RT10 samples do not affect start bit verification. If any or all of the RT8, RT9, and RT10 start bit samples are 1s following a successful start bit verification, the noise flag (NF) is set and the receiver assumes that the bit is a start bit.

To verify a stop bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 12-4 summarizes the results of the stop bit samples.

Table 12-4. Stop Bit Recovery

RT8, RT9, and RT10 Samples	Framing Error Flag	Noise Flag
000	1	0
001	1	1
010	1	1
011	0	1
100	1	1
101	0	1
110	0	1
111	0	0

12.4.3.4 Framing Errors

If the data recovery logic does not detect a 1 where the stop bit should be in an incoming character, it sets the framing error bit, FE, in SCS1. A break character also sets the FE bit because a break character has no stop bit. The FE bit is set at the same time that the SCRF bit is set.

12.4.3.5 Baud Rate Tolerance

A transmitting device may be operating at a baud rate below or above the receiver baud rate. Accumulated bit time misalignment can cause one of the three stop bit data samples to fall outside the actual stop bit. Then a noise error occurs. If more than one of the samples is outside the stop bit, a framing error occurs. In most applications, the baud rate tolerance is much more than the degree of misalignment that is likely to occur.

As the receiver samples an incoming character, it resynchronizes the RT clock on any valid falling edge within the character. Resynchronization within characters corrects misalignments between transmitter bit times and receiver bit times.

Slow Data Tolerance

Figure 12-8 shows how much a slow received character can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.
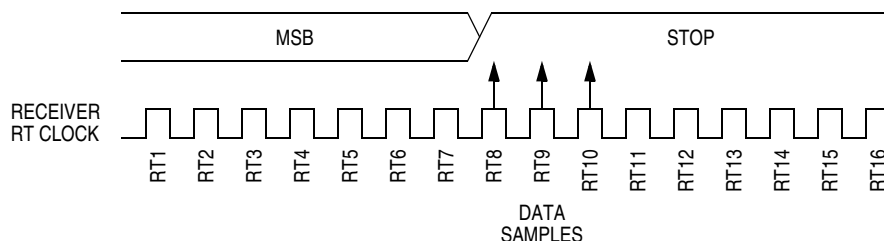
Figure 12-8. Slow Data

For an 8-bit character, data sampling of the stop bit takes the receiver
9 bit times $\times$ 16 RT cycles + 10 RT cycles = 154 RT cycles.

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

With the misaligned character shown in Figure 12-8, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 9 bit times × 16 RT cycles + 3 RT cycles = 147 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit character with no errors is

$$\left| \frac{154 - 147}{154} \right| \times 100 = 4.54\%$$

For a 9-bit character, data sampling of the stop bit takes the receiver
10 bit times × 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in Figure 12-8, the receiver counts 170 RT cycles at the point when the count of the transmitting device is
10 bit times × 16 RT cycles + 3 RT cycles = 163 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is

$$\left| \frac{170 - 163}{170} \right| \times 100 = 4.12\%$$

Fast Data Tolerance

Figure 12-9 shows how much a fast received character can be misaligned without causing a noise error or a framing error. The fast stop bit ends at RT10 instead of RT16 but is still there for the stop bit data samples at RT8, RT9, and RT10.

Figure 12-9. Fast Data

For an 8-bit character, data sampling of the stop bit takes the receiver
9 bit times × 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in Figure 12-9, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 10 bit times × 16 RT cycles = 160 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is

$$\left| \frac{154 - 160}{154} \right| \times 100 = 3.90\%.$$
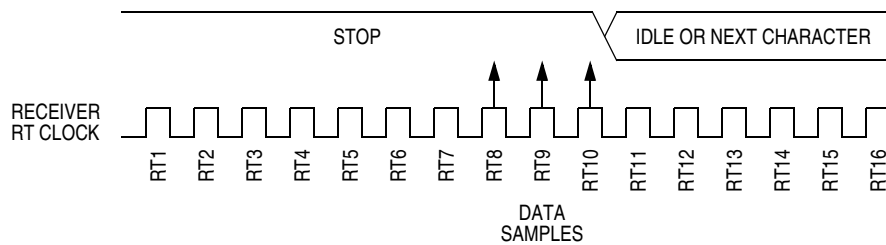
For a 9-bit character, data sampling of the stop bit takes the receiver
10 bit times × 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in Figure 12-9, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 11 bit times × 16 RT cycles = 176 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is

$$\left| \frac{170 - 176}{170} \right| \times 100 = 3.53\%.$$

12.4.3.6 Receiver Wakeup

So that the MCU can ignore transmissions intended only for other receivers in multiple-receiver systems, the receiver can be put into a standby state. Setting the receiver wakeup bit, RWU, in SCC2 puts the receiver into a standby state during which receiver interrupts are disabled.

Depending on the state of the WAKE bit in SCC1, either of two conditions on the RxD pin can bring the receiver out of the standby state:

1. Address mark — An address mark is a 1 in the most significant bit position of a received character. When the WAKE bit is set, an address mark wakes the receiver from the standby state by clearing the RWU bit. The address mark also sets the SCI receiver full bit, SCRF. Software can then compare the character containing the address mark to the user-defined address of the receiver. If they are the same, the receiver remains awake and processes the characters that follow. If they are not the same, software can set the RWU bit and put the receiver back into the standby state.

2. Idle input line condition — When the WAKE bit is clear, an idle character on the RxD pin wakes the receiver from the standby state by clearing the RWU bit. The idle character that wakes the receiver does not set the receiver idle bit, IDLE, or the SCI receiver full bit, SCRF. The idle line type bit, ILTY, determines whether the receiver begins counting 1s as idle character bits after the start bit or after the stop bit.

> ***NOTE***
> *With the WAKE bit clear, setting the RWU bit after the RxD pin has been idle may cause the receiver to wake up immediately.*

12.4.3.7 Receiver Interrupts

These sources can generate CPU interrupt requests from the SCI receiver:

- SCI receiver full (SCRF) — The SCRF bit in SCS1 indicates that the receive shift register has transferred a character to the SCDR. SCRF can generate a receiver CPU interrupt request. Setting the SCI receive interrupt enable bit, SCRIE, in SCC2 enables the SCRF bit to generate receiver CPU interrupts.

- Idle input (IDLE) — The IDLE bit in SCS1 indicates that 10 or 11 consecutive 1s shifted in from the RxD pin. The idle line interrupt enable bit, ILIE, in SCC2 enables the IDLE bit to generate CPU interrupt requests.

12.4.3.8 Error Interrupts

These receiver error flags in SCS1 can generate CPU interrupt requests:

- Receiver overrun (OR) — The OR bit indicates that the receive shift register shifted in a new character before the previous character was read from the SCDR. The previous character remains in the SCDR, and the new character is lost. The overrun interrupt enable bit, ORIE, in SCC3 enables OR to generate SCI error CPU interrupt requests.

- Noise flag (NF) — The NF bit is set when the SCI detects noise on incoming data or break characters, including start, data, and stop bits. The noise error interrupt enable bit, NEIE, in SCC3 enables NF to generate SCI error CPU interrupt requests.

- Framing error (FE) — The FE bit in SCS1 is set when a 0 occurs where the receiver expects a stop bit. The framing error interrupt enable bit, FEIE, in SCC3 enables FE to generate SCI error CPU interrupt requests.

- Parity error (PE) — The PE bit in SCS1 is set when the SCI detects a parity error in incoming data. The parity error interrupt enable bit, PEIE, in SCC3 enables PE to generate SCI error CPU interrupt requests.

12.5 Low-Power Modes

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

12.5.1 Wait Mode

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

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

12.5.2 Stop Mode

The SCI module is inactive in stop mode. The STOP instruction does not affect SCI register states. SCI module operation resumes after the MCU exits stop mode.

Because the internal clock is inactive during stop mode, entering stop mode during an SCI transmission or reception results in invalid data.

12.6 I/O Signals

Port B shares two of its pins with the SCI module. The two SCI I/O pins are:
- TxD — Transmit data
- RxD — Receive data

12.6.1 TxD (Transmit Data)

The TxD pin is the serial data output from the SCI transmitter. The SCI shares the TxD pin with port B. When the SCI is enabled, the TxD pin is an output regardless of the state of the DDRB5 bit in data direction register B (DDRB).

12.6.2 RxD (Receive Data)

The RxD pin is the serial data input to the SCI receiver. The SCI shares the RxD pin with port B. When the SCI is enabled, the RxD pin is an input regardless of the state of the DDRB4 bit in data direction register B (DDRB).

12.7 I/O Registers

These I/O registers control and monitor SCI operation:

- SCI control register 1 (SCC1)
- SCI control register 2 (SCC2)
- SCI control register 3 (SCC3)
- SCI status register 1 (SCS1)
- SCI status register 2 (SCS2)
- SCI data register (SCDR)
- SCI baud rate register (SCBR)

12.7.1 SCI Control Register 1

SCI control register 1 (SCC1):

- Enables loop mode operation
- Enables the SCI
- Controls output polarity
- Controls character length
- Controls SCI wakeup method
- Controls idle character detection
- Enables parity function
- Controls parity type

Address: $0013

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
Reset:	0	0	0	0	0	0	0	0

Figure 12-10. SCI Control Register 1 (SCC1)

LOOPS — Loop Mode Select Bit
This read/write bit enables loop mode operation. In loop mode the RxD pin is disconnected from the SCI, and the transmitter output goes into the receiver input. Both the transmitter and the receiver must be enabled to use loop mode. Reset clears the LOOPS bit.
   1 = Loop mode enabled
   0 = Normal operation enabled

ENSCI — Enable SCI Bit
This read/write bit enables the SCI and the SCI baud rate generator. Clearing ENSCI sets the SCTE and TC bits in SCI status register 1 and disables transmitter interrupts. Reset clears the ENSCI bit.
   1 = SCI enabled
   0 = SCI disabled

TXINV — Transmit Inversion Bit

This read/write bit reverses the polarity of transmitted data. Reset clears the TXINV bit.

1 = Transmitter output inverted
0 = Transmitter output not inverted

NOTE

Setting the TXINV bit inverts all transmitted values, including idle, break, start, and stop bits.

M — Mode (Character Length) Bit

This read/write bit determines whether SCI characters are eight or nine bits long. (See Table 12-5.) The ninth bit can serve as an extra stop bit, as a receiver wakeup signal, or as a parity bit. Reset clears the M bit.

1 = 9-bit SCI characters
0 = 8-bit SCI characters

WAKE — Wakeup Condition Bit

This read/write bit determines which condition wakes up the SCI: a 1 (address mark) in the most significant bit position of a received character or an idle condition on the RxD pin. Reset clears the WAKE bit.

1 = Address mark wakeup
0 = Idle line wakeup

ILTY — Idle Line Type Bit

This read/write bit determines when the SCI starts counting 1s as idle character bits. The counting begins either after the start bit or after the stop bit. If the count begins after the start bit, then a string of 1s preceding the stop bit may cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. Reset clears the ILTY bit.

1 = Idle character bit count begins after stop bit.
0 = Idle character bit count begins after start bit.

PEN — Parity Enable Bit

This read/write bit enables the SCI parity function. (See Table 12-5.) When enabled, the parity function inserts a parity bit in the most significant bit position. (See Figure 12-2.) Reset clears the PEN bit.

1 = Parity function enabled
0 = Parity function disabled

PTY — Parity Bit

This read/write bit determines whether the SCI generates and checks for odd parity or even parity. (See Table 12-5.) Reset clears the PTY bit.

1 = Odd parity
0 = Even parity

NOTE

Changing the PTY bit in the middle of a transmission or reception can generate a parity error.

Table 12-5. Character Format Selection

Control Bits		Character Format				
M	PEN–PTY	Start Bits	Data Bits	Parity	Stop Bits	Character Length
0	0X	1	8	None	1	10 Bits
1	0X	1	9	None	1	11 Bits
0	10	1	7	Even	1	10 Bits
0	11	1	7	Odd	1	10 Bits
1	10	1	8	Even	1	11 Bits
1	11	1	8	Odd	1	11 Bits

12.7.2 SCI Control Register 2

SCI control register 2 (SCC2):
- Enables these CPU interrupt requests:
  - Enables the SCTE bit to generate transmitter CPU interrupt requests
  - Enables the TC bit to generate transmitter CPU interrupt requests
  - Enables the SCRF bit to generate receiver CPU interrupt requests
  - Enables the IDLE bit to generate receiver CPU interrupt requests
- Enables the transmitter
- Enables the receiver
- Enables SCI wakeup
- Transmits SCI break characters

Address: $0014

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0
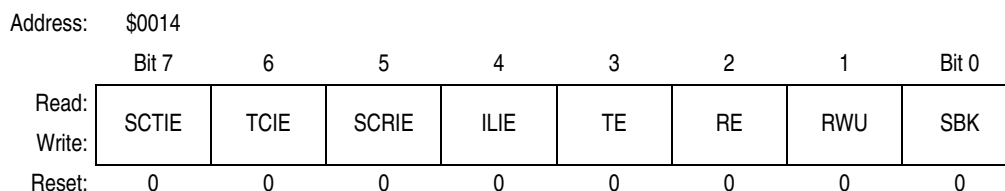
Figure 12-11. SCI Control Register 2 (SCC2)

SCTIE — SCI Transmit Interrupt Enable Bit
This read/write bit enables the SCTE bit to generate SCI transmitter CPU interrupt requests. Setting the SCTIE bit in SCC3 enables the SCTE bit to generate CPU interrupt requests. Reset clears the SCTIE bit.
   1 = SCTE enabled to generate CPU interrupt
   0 = SCTE not enabled to generate CPU interrupt

TCIE — Transmission Complete Interrupt Enable Bit
This read/write bit enables the TC bit to generate SCI transmitter CPU interrupt requests. Reset clears the TCIE bit.
   1 = TC enabled to generate CPU interrupt requests
   0 = TC not enabled to generate CPU interrupt requests

SCRIE — SCI Receive Interrupt Enable Bit

This read/write bit enables the SCRF bit to generate SCI receiver CPU interrupt requests. Setting the SCRIE bit in SCC3 enables the SCRF bit to generate CPU interrupt requests. Reset clears the SCRIE bit.

    1 = SCRF enabled to generate CPU interrupt
    0 = SCRF not enabled to generate CPU interrupt

ILIE — Idle Line Interrupt Enable Bit

This read/write bit enables the IDLE bit to generate SCI receiver CPU interrupt requests. Reset clears the ILIE bit.

    1 = IDLE enabled to generate CPU interrupt requests
    0 = IDLE not enabled to generate CPU interrupt requests

TE — Transmitter Enable Bit

Setting this read/write bit begins the transmission by sending a preamble of 10 or 11 1s from the transmit shift register to the TxD pin. If software clears the TE bit, the transmitter completes any transmission in progress before the TxD returns to the idle condition (logic 1). Clearing and then setting TE during a transmission queues an idle character to be sent after the character currently being transmitted. Reset clears the TE bit.

    1 = Transmitter enabled
    0 = Transmitter disabled

NOTE

*Writing to the TE bit is not allowed when the enable SCI bit (ENSCI) is clear.
ENSCI is in SCI control register 1.*

RE — Receiver Enable Bit

Setting this read/write bit enables the receiver. Clearing the RE bit disables the receiver but does not affect receiver interrupt flag bits. Reset clears the RE bit.

    1 = Receiver enabled
    0 = Receiver disabled

NOTE

*Writing to the RE bit is not allowed when the enable SCI bit (ENSCI) is
clear. ENSCI is in SCI control register 1.*

RWU — Receiver Wakeup Bit

This read/write bit puts the receiver in a standby state during which receiver interrupts are disabled. The WAKE bit in SCC1 determines whether an idle input or an address mark brings the receiver out of the standby state and clears the RWU bit. Reset clears the RWU bit.

    1 = Standby state
    0 = Normal operation

SBK — Send Break Bit

Setting and then clearing this read/write bit transmits a break character followed by a 1. The 1 after the break character guarantees recognition of a valid start bit. If SBK remains set, the transmitter continuously transmits break characters with no 1s between them. Reset clears the SBK bit.

    1 = Transmit break characters
    0 = No break characters being transmitted

NOTE

*Do not toggle the SBK bit immediately after setting the SCTE bit. Toggling
SBK before the preamble begins causes the SCI to send a break character
instead of a preamble.*

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

12.7.3 SCI Control Register 3

SCI control register 3 (SCC3):

- Stores the ninth SCI data bit received and the ninth SCI data bit to be transmitted.
- Enables these interrupts:
    - Receiver overrun interrupts
    - Noise error interrupts
    - Framing error interrupts
    - Parity error interrupts

Address: $0015

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
Write:		T8	R	R	ORIE	NEIE	FEIE	PEIE
Reset:	U	U	0	0	0	0	0	0

☐ = Unimplemented   [R] = Reserved   U = Unaffected
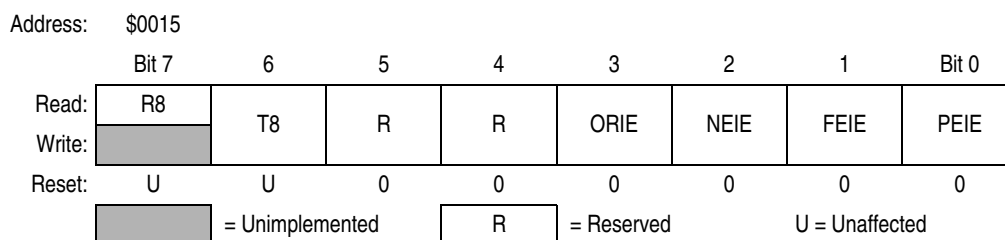
Figure 12-12. SCI Control Register 3 (SCC3)

R8 — Received Bit 8
When the SCI is receiving 9-bit characters, R8 is the read-only ninth bit (bit 8) of the received character. R8 is received at the same time that the SCDR receives the other eight bits.

When the SCI is receiving 8-bit characters, R8 is a copy of the eighth bit (bit 7). Reset has no effect on the R8 bit.

T8 — Transmitted Bit 8
When the SCI is transmitting 9-bit characters, T8 is the read/write ninth bit (bit 8) of the transmitted character. T8 is loaded into the transmit shift register at the same time that the SCDR is loaded into the transmit shift register. Reset has no effect on the T8 bit.

ORIE — Receiver Overrun Interrupt Enable Bit
This read/write bit enables SCI error CPU interrupt requests generated by the receiver overrun bit, OR.
    1 = SCI error CPU interrupt requests from OR bit enabled
    0 = SCI error CPU interrupt requests from OR bit disabled

NEIE — Receiver Noise Error Interrupt Enable Bit
This read/write bit enables SCI error CPU interrupt requests generated by the noise error bit, NE. Reset clears NEIE.
    1 = SCI error CPU interrupt requests from NE bit enabled
    0 = SCI error CPU interrupt requests from NE bit disabled

FEIE — Receiver Framing Error Interrupt Enable Bit
This read/write bit enables SCI error CPU interrupt requests generated by the framing error bit, FE. Reset clears FEIE.
    1 = SCI error CPU interrupt requests from FE bit enabled
    0 = SCI error CPU interrupt requests from FE bit disabled

PEIE — Receiver Parity Error Interrupt Enable Bit

This read/write bit enables SCI receiver CPU interrupt requests generated by the parity error bit, PE. Reset clears PEIE.

    1 = SCI error CPU interrupt requests from PE bit enabled

    0 = SCI error CPU interrupt requests from PE bit disabled

NOTE

Bits 5 and 4 are reserved for MCUs with a direct-memory access (DMA) module. Because the MC68HC908KX8 does not have a DMA module, these bits should not be set.

12.7.4 SCI Status Register 1

SCI status register 1 (SCS1) contains flags to signal these conditions:

- Transfer of SCDR data to transmit shift register complete
- Transmission complete
- Transfer of receive shift register data to SCDR complete
- Receiver input idle
- Receiver overrun
- Noisy data
- Framing error
- Parity error

Address: $0016

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
Write:								
Reset:	1	1	0	0	0	0	0	0

    = Unimplemented
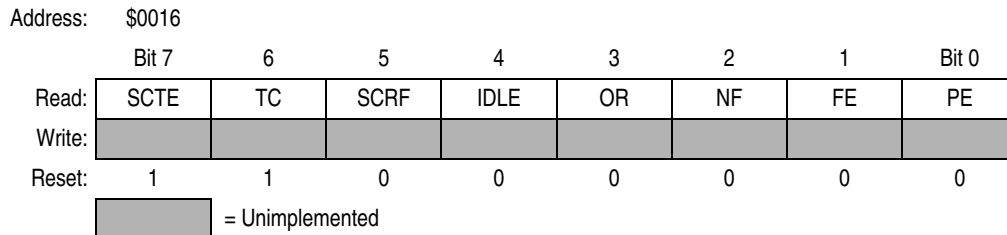
Figure 12-13. SCI Status Register 1 (SCS1)

SCTE — SCI Transmitter Empty Bit

This clearable, read-only bit is set when the SCDR transfers a character to the transmit shift register. SCTE can generate an SCI transmitter CPU interrupt request. When the SCTIE bit in SCC2 is set, SCTE generates an SCI transmitter CPU interrupt request. In normal operation, clear the SCTE bit by reading SCS1 with SCTE set and then writing to SCDR. Reset sets the SCTE bit.

    1 = SCDR data transferred to transmit shift register

    0 = SCDR data not transferred to transmit shift register

TC — Transmission Complete Bit

This read-only bit is set when the SCTE bit is set, and no data, preamble, or break character is being transmitted. TC generates an SCI transmitter CPU interrupt request if the TCIE bit in SCC2 is also set. TC is cleared automatically when data, preamble, or break is queued and ready to be sent. There may be up to 1.5 transmitter clocks of latency between queueing data, preamble, and break and the transmission actually starting. Reset sets the TC bit.

    1 = No transmission in progress

    0 = Transmission in progress

SCRF — SCI Receiver Full Bit

This clearable, read-only bit is set when the data in the receive shift register transfers to the SCI data register. SCRF can generate an SCI receiver CPU interrupt request. When the SCRIE bit in SCC2 is set the SCRF generates a CPU interrupt request. In normal operation, clear the SCRF bit by reading SCS1 with SCRF set and then reading the SCDR. Reset clears SCRF.

    1 = Received data available in SCDR
    0 = Data not available in SCDR

IDLE — Receiver Idle Bit

This clearable, read-only bit is set when 10 or 11 consecutive 1s appear on the receiver input. IDLE generates an SCI error CPU interrupt request if the ILIE bit in SCC2 is also set. Clear the IDLE bit by reading SCS1 with IDLE set and then reading the SCDR. After the receiver is enabled, it must receive a valid character that sets the SCRF bit before an idle condition can set the IDLE bit. Also, after the IDLE bit has been cleared, a valid character must again set the SCRF bit before an idle condition can set the IDLE bit. Reset clears the IDLE bit.

    1 = Receiver input idle
    0 = Receiver input active (or idle since the IDLE bit was cleared)

OR — Receiver Overrun Bit

This clearable, read-only bit is set when software fails to read the SCDR before the receive shift register receives the next character. The OR bit generates an SCI error CPU interrupt request if the ORIE bit in SCC3 is also set. The data in the shift register is lost, but the data already in the SCDR is not affected. Clear the OR bit by reading SCS1 with OR set and then reading the SCDR. Reset clears the OR bit.

    1 = Receive shift register full and SCRF = 1
    0 = No receiver overrun

Software latency may allow an overrun to occur between reads of SCS1 and SCDR in the flag-clearing sequence. Figure 12-14 shows the normal flag-clearing sequence and an example of an overrun caused by a delayed flag-clearing sequence. The delayed read of SCDR does not clear the OR bit because OR was not set when SCS1 was read. Byte 2 caused the overrun and is lost. The next flag-clearing sequence reads byte 3 in the SCDR instead of byte 2.

In applications that are subject to software latency or in which it is important to know which byte is lost due to an overrun, the flag-clearing routine can check the OR bit in a second read of SCS1 after reading the data register.

NF — Receiver Noise Flag Bit

This clearable, read-only bit is set when the SCI detects noise on the RxD pin. NF generates an NF CPU interrupt request if the NEIE bit in SCC3 is also set. Clear the NF bit by reading SCS1 and then reading the SCDR. Reset clears the NF bit.

    1 = Noise detected
    0 = No noise detected

FE — Receiver Framing Error Bit

This clearable, read-only bit is set when a 0 is accepted as the stop bit. FE generates an SCI error CPU interrupt request if the FEIE bit in SCC3 also is set. Clear the FE bit by reading SCS1 with FE set and then reading the SCDR. Reset clears the FE bit.

    1 = Framing error detected
    0 = No framing error detected

NORMAL FLAG CLEARING SEQUENCE

DELAYED FLAG CLEARING SEQUENCE
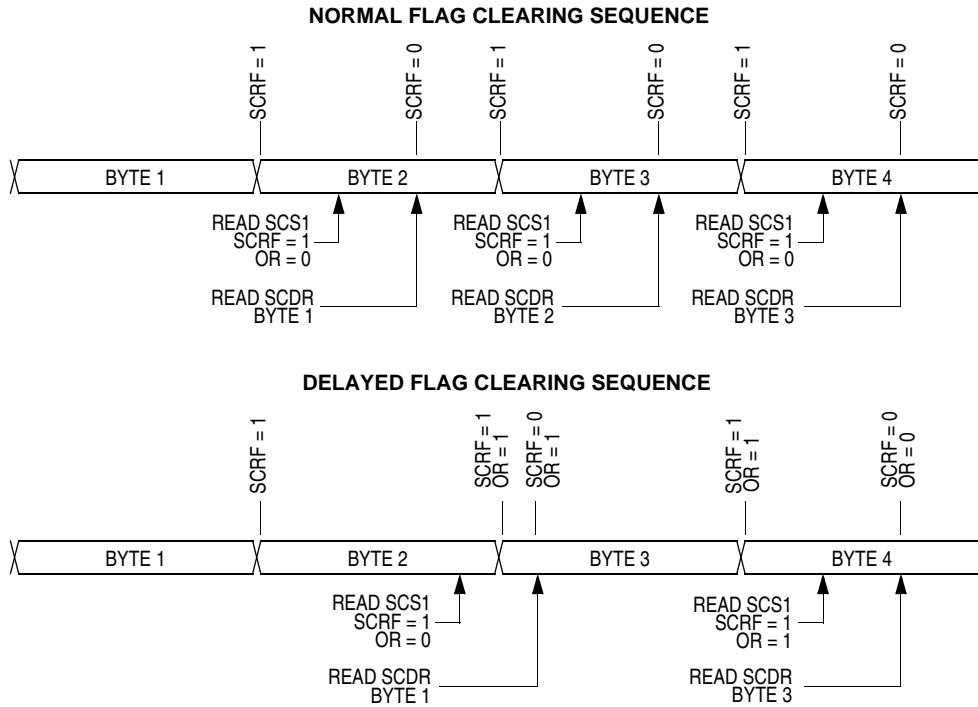
Figure 12-14. Flag Clearing Sequence

PE — Receiver Parity Error Bit

This clearable, read-only bit is set when the SCI detects a parity error in incoming data. PE generates a PE CPU interrupt request if the PEIE bit in SCC3 is also set. Clear the PE bit by reading SCS1 with PE set and then reading the SCDR. Reset clears the PE bit.

    1 = Parity error detected
    0 = No parity error detected

12.7.5 SCI Status Register 2

SCI status register 2 (SCS2) contains flags to signal these conditions:

- Break character detected
- Incoming data

Address: $0017

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

    = Unimplemented

Figure 12-15. SCI Status Register 2 (SCS2)

BKF — Break Flag Bit

This clearable, read-only bit is set when the SCI detects a break character on the RxD pin. In SCS1, the FE and SCRF bits are also set. In 9-bit character transmissions, the R8 bit in SCC3 is cleared. BKF does not generate a CPU interrupt request. Clear BKF by reading SCS2 with BKF set and then reading

the SCDR. Once cleared, BKF can become set again only after 1s again appear on the RxD pin followed by another break character. Reset clears the BKF bit.

    1 = Break character detected
    0 = No break character detected

RPF — Reception-in-Progress Flag Bit
This read-only bit is set when the receiver detects a 0 during the RT1 time period of the start bit search. RPF does not generate an interrupt request. RPF is reset after the receiver detects false start bits (usually from noise or a baud rate mismatch), or when the receiver detects an idle character. Polling RPF before disabling the SCI module or entering stop mode can show whether a reception is in progress.

    1 = Reception in progress
    0 = No reception in progress

12.7.6 SCI Data Register

The SCI data register (SCDR) is the buffer between the internal data bus and the receive and transmit shift registers. Reset has no effect on data in the SCI data register.

Address: $0018

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R7	R6	R5	R4	R3	R2	R1	R0
Write:	T7	T6	T5	T4	T3	T2	T1	T0
Reset:				Unaffected by reset				

Figure 12-16. SCI Data Register (SCDR)

R7/T7–R0/T0 — Receive/Transmit Data Bits
Reading address $0018 accesses the read-only received data bits, R7–R0. Writing to address $0018 writes the data to be transmitted, T7–T0. Reset has no effect on the SCI data register.

NOTE
Do not use read-modify-write instructions on the SCI data register.

12.7.7 SCI Baud Rate Register

The baud rate register (SCBR) selects the baud rate for both the receiver and the transmitter.

Address: $0019

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
Write:			SCP1	SCP0	R	SCR2	SCR1	SCR0
Reset:	0	0	0	0	0	0	0	0

    = Unimplemented   R = Reserved

Figure 12-17. SCI Baud Rate Register (SCBR)

SCP1 and SCP0 — SCI Baud Rate Prescaler Bits

These read/write bits select the baud rate prescaler divisor as shown in Table 12-6. Reset clears SCP1 and SCP0.

Table 12-6. SCI Baud Rate Prescaling

SCP[1:0]	Prescaler Divisor (PD)
00	1
01	3
10	4
11	13

SCR2–SCR0 — SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate divisor as shown in Table 12-7. Reset clears SCR2–SCR0.

Table 12-7. SCI Baud Rate Selection

SCR[2:1:0]	Baud Rate Divisor (BD)
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

Use this formula to calculate the SCI baud rate:

$$\text{Baud rate} = \frac{f_{BAUDCLK}}{64 \times PD \times BD}$$

where:

$f_{BAUDCLK}$ = baud clock frequency
PD = prescaler divisor
BD = baud rate divisor

Table 12-8 shows the SCI baud rates that can be generated with a 4.9152-MHz CGMXCLK frequency.

Table 12-8. SCI Baud Rate Selection Examples

SCP[1:0]	Prescaler Divisor (PD)	SCR[2:1:0]	Baud Rate Divisor (BD)	Baud Rate ($f_{BAUDCLK}$ = 4.9152 MHz)
00	1	000	1	76,800
00	1	001	2	38,400
00	1	010	4	19,200
00	1	011	8	9600
00	1	100	16	4800
00	1	101	32	2400
00	1	110	64	1200
00	1	111	128	600
01	3	000	1	25,600
01	3	001	2	12,800
01	3	010	4	6400
01	3	011	8	3200
01	3	100	16	1600
01	3	101	32	800
01	3	110	64	400
01	3	111	128	200
10	4	000	1	19,200
10	4	001	2	9600
10	4	010	4	4800
10	4	011	8	2400
10	4	100	16	1200
10	4	101	32	600
10	4	110	64	300
10	4	111	128	150
11	13	000	1	5908
11	13	001	2	2954
11	13	010	4	1477
11	13	011	8	739
11	13	100	16	369
11	13	101	32	185
11	13	110	64	92
11	13	111	128	46

Chapter 13
System Integration Module (SIM)

13.1 Introduction

This section describes the system integration module (SIM), which supports up to 24 external and/or internal interrupts. The SIM is a system state controller that coordinates the central processor unit (CPU) and exception timing. Together with the CPU, the SIM controls all microcontroller unit (MCU) activities.

Figure 13-1 is a summary of the SIM input/output (I/O) registers. A block diagram of the SIM is shown in Figure 13-2.

The SIM is responsible for:

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

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE01	SIM Reset Status Register (SRSR) See page 148.	Read:	POR	0	COP	ILOP	ILAD	MENRST	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
$FE04	Interrupt Status Register 1 (INT1) See page 149.	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 150.	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE06	Interrupt Status Register 3 (INT3) See page 150.	Read:	IF22	IF21	IF20	IF19	IF18	IF17	IF16	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0

	= Unimplemented	R	= Reserved

Figure 13-1. SIM I/O Register Summary

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1
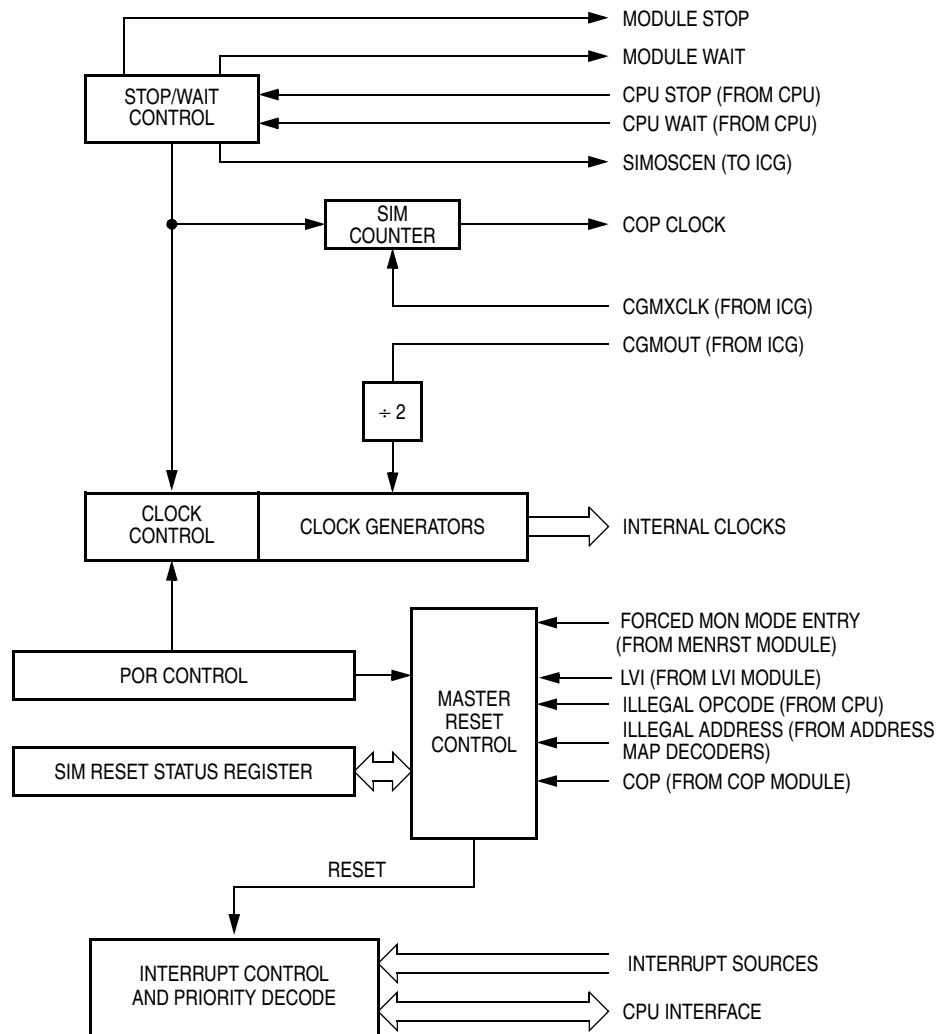
Figure 13-2. SIM Block Diagram

Table 13-1 shows the internal signal names used in this section.

Table 13-1. Signal Name Conventions

Signal Name	Description
CGMXCLK	Selected clock source from internal clock generator module (ICG)
CGMOUT	Clock output from ICG module (bus clock = CGMOUT divided by two)
IAB	Internal address bus
IDB	Internal data bus
PORRST	Signal from the power-on reset (POR) module to the SIM
IRST	Internal reset signal
R/$\overline{W}$	Read/write signal

13.2 SIM Bus Clock Control and Generation

The bus clock generator provides system clock signals for the CPU and peripherals on the MCU. The system clocks are generated from an incoming clock, CGMOUT, as shown in Figure 13-3. This clock originates from either an external oscillator or from the internal clock generator.
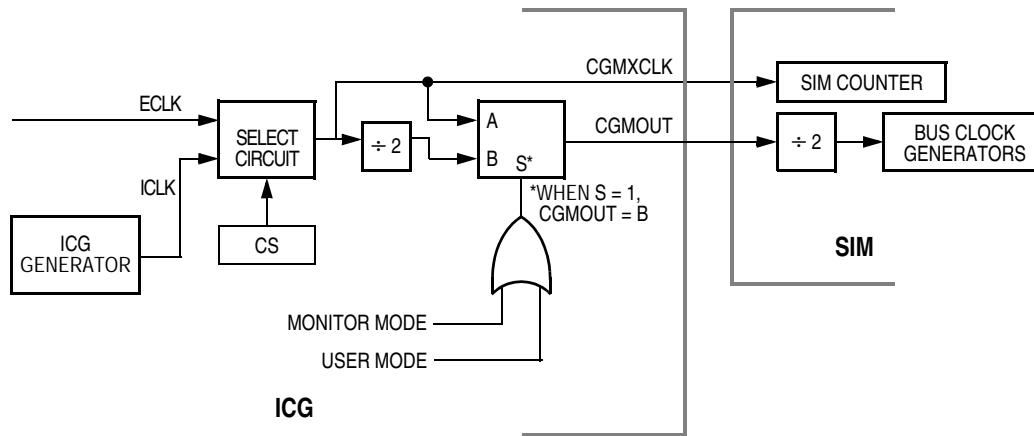
Figure 13-3. System Clock Signals

13.2.1 Bus Timing

In user mode, the internal bus frequency is the internal clock generator output (CGMXCLK) divided by four.

13.2.2 Clock Startup from POR or LVI Reset

When the power-on reset (POR) module or the low-voltage inhibit (LVI) module generates a reset, the clocks to the CPU and peripherals are inactive and held in an inactive phase until after 4096 CGMXCLK cycles. The MCU is held in reset by the SIM during this entire period. The bus clocks start upon completion of the timeout.

13.2.3 Clocks in Stop Mode and Wait Mode

Upon exit from stop mode by an interrupt or reset, the SIM allows CGMXCLK to clock the SIM counter. The CPU and peripheral clocks do not become active until after the stop delay timeout. Stop mode recovery timing is discussed in detail in 13.6.2 Stop Mode.

In wait mode, the CPU clocks are inactive. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

13.3 Reset and System Initialization

The MCU has these internal reset sources:

- Power-on reset (POR) module
- Computer operating properly (COP) module
- Low-voltage inhibit (LVI) module
- Illegal opcode
- Illegal address
- Forced monitor mode entry reset (MENRST) module

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

These internal resets clear the SIM counter and set a corresponding bit in the SIM reset status register (SRSR). See 13.4 SIM Counter and 13.7.1 SIM Reset Status Register.

13.3.1 Active Resets from Internal Sources

An internal reset can be caused by an illegal address, illegal opcode, COP timeout, LVI, POR, or MENRST as shown in Figure 13-4.

NOTE
For LVI or POR resets, the SIM cycles through 4096 CGMXCLK cycles during which the SIM asserts IRST. The internal reset signal then follows with the 64-cycle phase as shown in Figure 13-5.
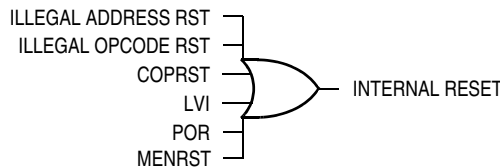
The COP reset is asynchronous to the bus clock.

Figure 13-4. Sources of Internal Reset
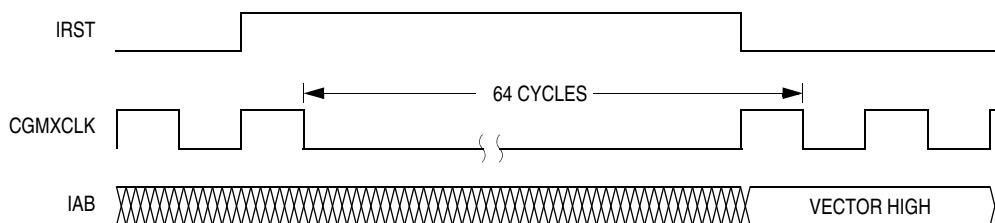
Figure 13-5. Internal Reset Timing

13.3.1.1 Power-On Reset

When power is first applied to the MCU, the power-on reset (POR) module generates a pulse to indicate that power-on has occurred. The MCU is held in reset while the SIM counter counts out 4096 CGMXCLK cycles. Another 64 CGMXCLK cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur.

At power-on, these events occur:

- A POR pulse is generated.
- The internal reset signal is asserted.
- The SIM enables CGMOUT.
- Internal clocks to the CPU and modules are held inactive for 4096 CGMXCLK cycles to allow stabilization of the internal clock generator.
- The POR bit of the SIM reset status register (SRSR) is set and all other bits in the register are cleared.
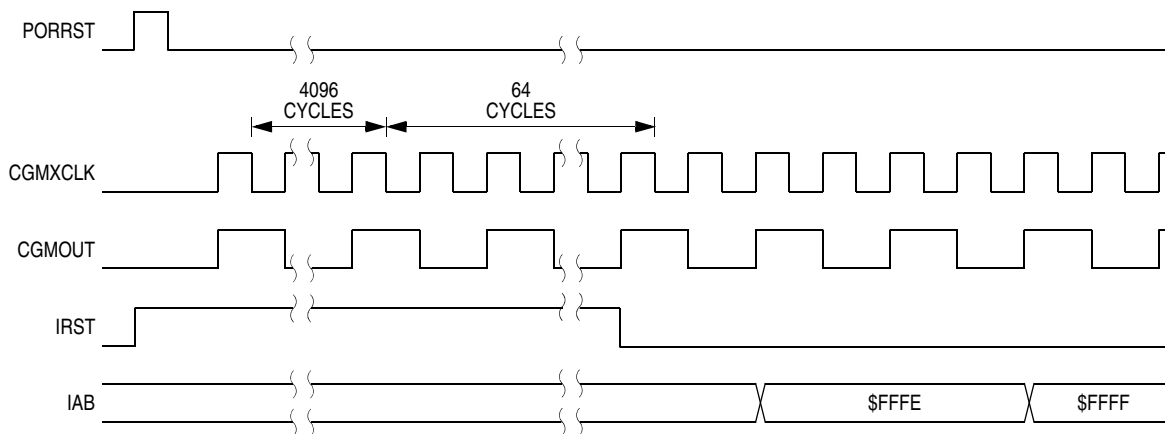
Figure 13-6. POR Recovery

13.3.1.2 Computer Operating Properly (COP) Reset

An input to the SIM is reserved for the COP reset signal. The overflow of the COP counter causes an internal reset and sets the COP bit in the reset status register (SRSR).

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

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

13.3.1.3 Illegal Opcode Reset

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

If the stop enable bit, STOP, in the configuration register (CONFIG1) is 0, the SIM treats the STOP instruction as an illegal opcode and causes an illegal opcode reset.

13.3.1.4 Illegal Address Reset

An opcode fetch from an unmapped address generates an illegal address reset. The SIM verifies that the CPU is fetching an opcode prior to asserting the ILAD bit in the SIM reset status register (SRSR) and resetting the MCU. A data fetch from an unmapped address does not generate a reset.

13.3.1.5 Forced Monitor Mode Entry Reset (MENRST)

The MENRST module is monitoring the reset vector fetches and will assert an internal reset if it detects that the reset vectors are erased ($FF). When the MCU comes out of reset, it is forced into monitor mode. See 16.3 Monitor ROM (MON).

13.3.1.6 Low-Voltage Inhibit (LVI) Reset

The low-voltage inhibit module (LVI) asserts its output to the SIM when the V_{DD} voltage falls to the V_{TRIPF} voltage. The LVI bit in the SIM reset status register (SRSR) is set and a chip reset is asserted if the LVIPWRD and LVIRSTD bits in the CONFIG register are at 0. The MCU is held in reset until V_{DD} rises above V_{TRIPR}. The MCU remains in reset until the SIM counts 4096 CGMXCLK to begin a reset recovery. Another 64 CGMXCLK cycles later, the CPU is released from reset to allow the reset vector sequence to occur. See Chapter 10 Low-Voltage Inhibit (LVI).

13.4 SIM Counter

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

13.4.1 SIM Counter During Power-On Reset

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

13.4.2 SIM Counter During Stop Mode Recovery

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

13.4.3 SIM Counter and Reset States

The SIM counter is free-running after all reset states. See 13.3.1 Active Resets from Internal Sources for counter control and internal reset recovery sequences.

13.5 Program Exception Control

Normal, sequential program execution can be changed in two ways:

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

2. Reset

13.5.1 Interrupts

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the return-from-interrupt (RTI) instruction recovers the CPU register contents from the stack so that normal processing can resume. Figure 13-7 shows interrupt entry timing. Figure 13-8 shows interrupt recovery timing.
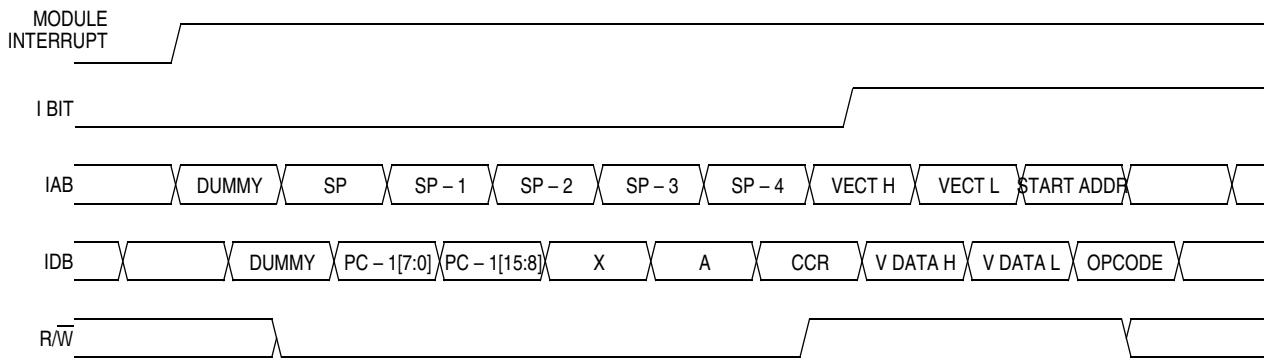
Figure 13-7. Interrupt Entry
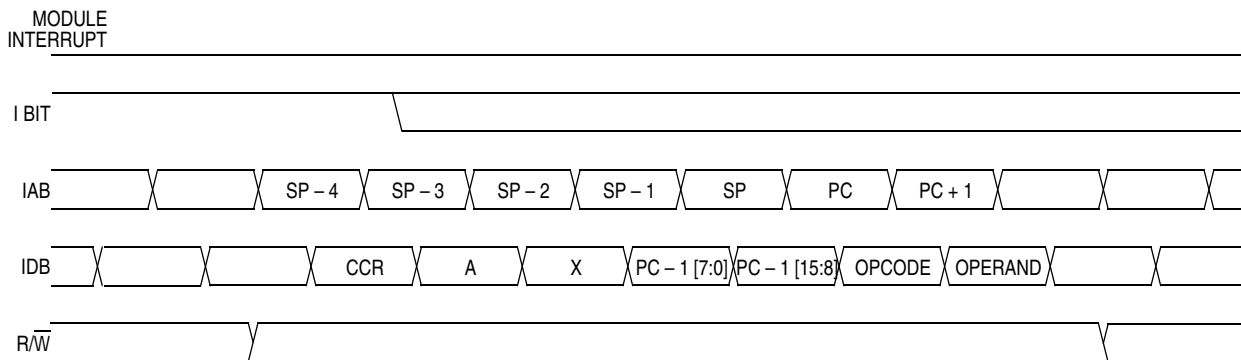
Figure 13-8. Interrupt Recovery

Interrupts are latched, and arbitration is performed in the SIM at the start of interrupt processing. The arbitration result is a constant that the CPU uses to determine which vector to fetch. As shown in Figure 13-9, once an interrupt is latched by the SIM, no other interrupt can take precedence, regardless of priority, until the latched interrupt is serviced or the I bit is cleared.
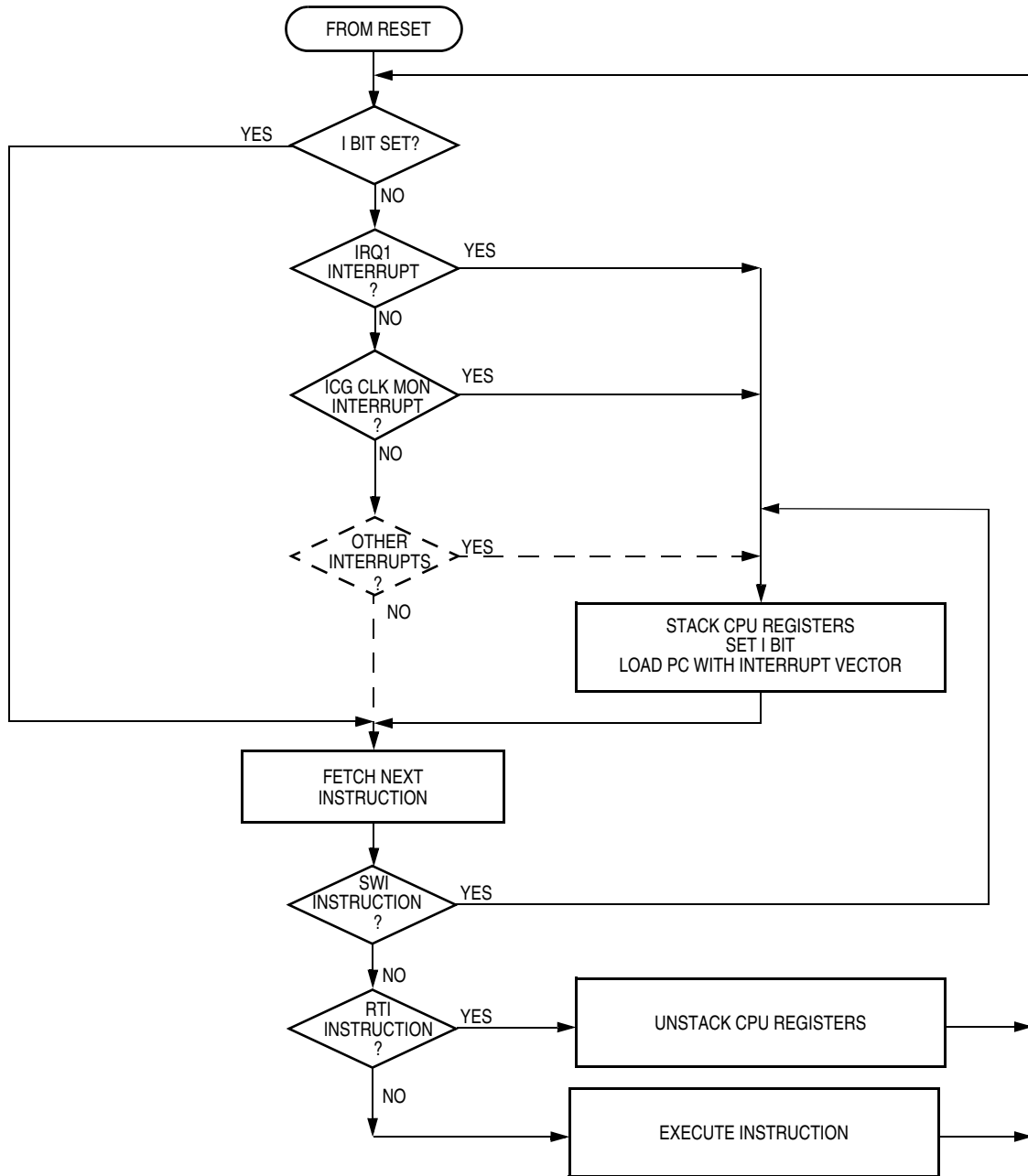
Figure 13-9. Interrupt Processing

13.5.1.1 Hardware Interrupts

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

If more than one interrupt is pending at the end of an instruction execution, the highest priority interrupt is serviced first. Figure 13-10 demonstrates what happens when two interrupts are pending. If an interrupt

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Freescale Semiconductor

is pending upon exit from the original interrupt service routine, the pending interrupt is serviced before the load-accumulator- from-memory (LDA) instruction is executed.

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

> *NOTE*
> *To maintain compatibility with the M68HC05, M6805, and M146805 Families the H register is not pushed on the stack during interrupt entry. If the interrupt service routine modifies the H register or uses the indexed addressing mode, software should save the H register and then restore it prior to exiting the routine.*
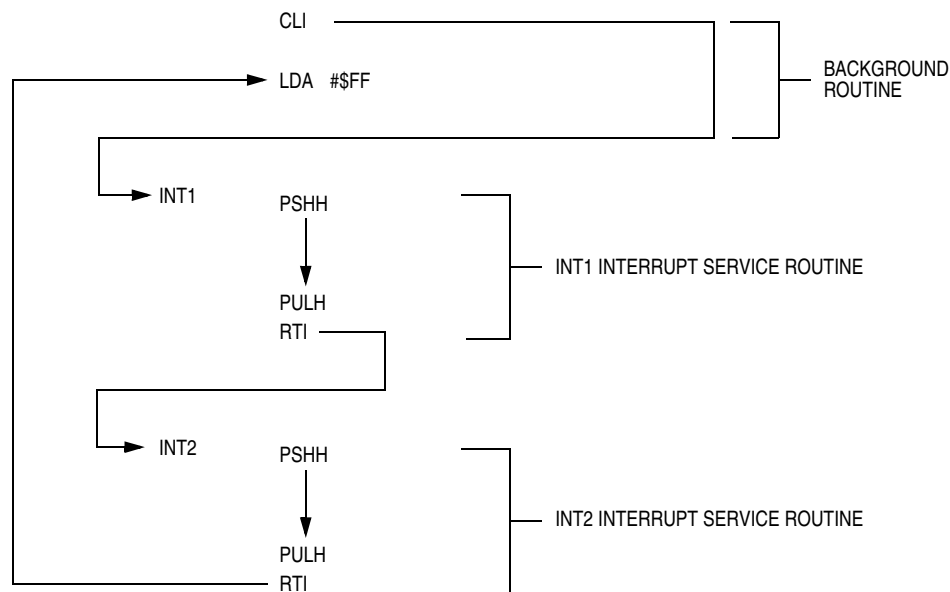
Figure 13-10. Interrupt Recognition Example

13.5.1.2 SWI Instruction

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

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

13.5.2 Reset

All reset sources always have higher priority than interrupts and cannot be arbitrated.

13.6 Low-Power Modes

Executing the WAIT or STOP instruction puts the MCU in a low power- consumption mode for standby situations. The SIM holds the CPU in a non-clocked state. Both STOP and WAIT clear the interrupt mask (I) in the condition code register, allowing interrupts to occur. Low-power modes are exited via an interrupt or reset.

13.6.1 Wait Mode

In wait mode, the CPU clocks are inactive while one set of peripheral clocks continues to run. Figure 13-11 shows the timing for wait mode entry.
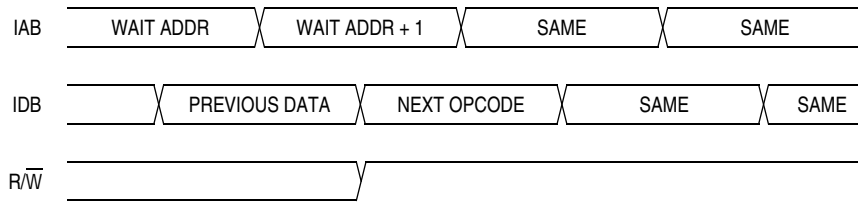
Note: Previous data can be operand data or the WAIT opcode, depending on the last instruction.

Figure 13-11. Wait Mode Entry Timing

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

Wait mode can also be exited by a reset. If the COP disable bit, COPD, in the configuration register is a 0, then the computer operating properly module (COP) is enabled and remains active in wait mode.

Figure 13-12 and Figure 13-13 show the timing for WAIT recovery.
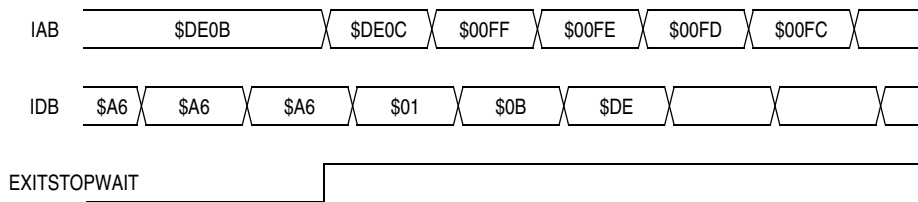
Note: EXITSTOPWAIT = CPU interrupt

Figure 13-12. Wait Recovery from Interrupt
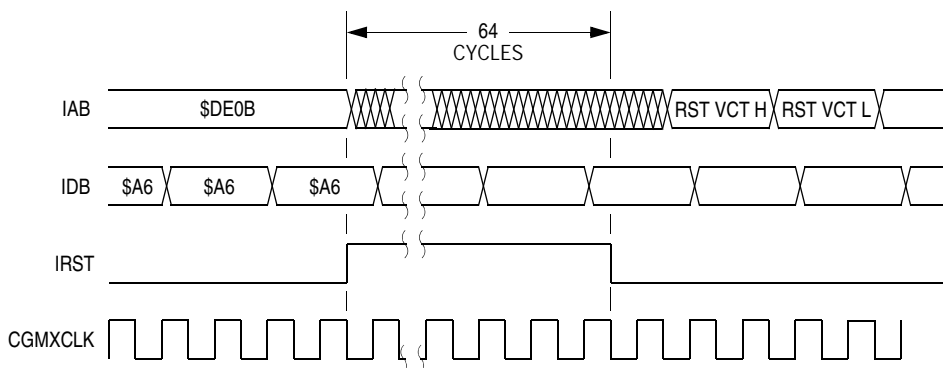
Figure 13-13. Wait Recovery from Internal Reset

13.6.2 Stop Mode

In stop mode, the SIM counter is held in reset and the CPU and peripheral clocks are held inactive. If the OSCENINSTOP bit in the configuration register is not enabled, the SIM also disables the internal clock generator module outputs (CGMOUT and CGMXCLK).

The CPU and peripheral clocks do not become active until after the stop delay timeout. Stop mode is exited via an interrupt request from a module that is still active in stop mode or from a system reset.

An interrupt request from a module that is still active in stop mode can cause an exit from stop mode. Stop recovery time is selectable using the SSREC bit in the configuration register. If SSREC is set, stop recovery is reduced from the normal delay of 4096 CGMXCLK cycles down to 32. Stacking for interrupts begins after the selected stop recovery time has elapsed.

When stop mode is exited due to a reset condition, the SIM forces a long stop recovery time of 4096 CGMXCLK cycles.

NOTE
Short stop recovery is ideal for applications using canned oscillators that do not require long startup times for stop mode. External crystal applications should use the full stop recovery time by clearing the SSREC bit.

The SIM counter is held in reset from the execution of the STOP instruction until the beginning of stop recovery. It is then used to time the recovery period. Figure 13-14 shows stop mode entry timing.
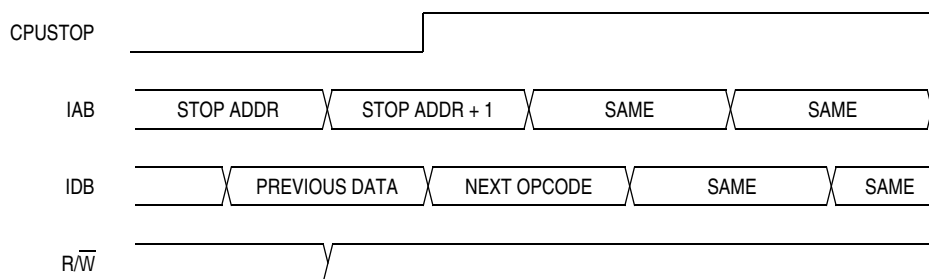
Note: Previous data can be operand data or the STOP opcode, depending on the last instruction.

Figure 13-14. Stop Mode Entry Timing
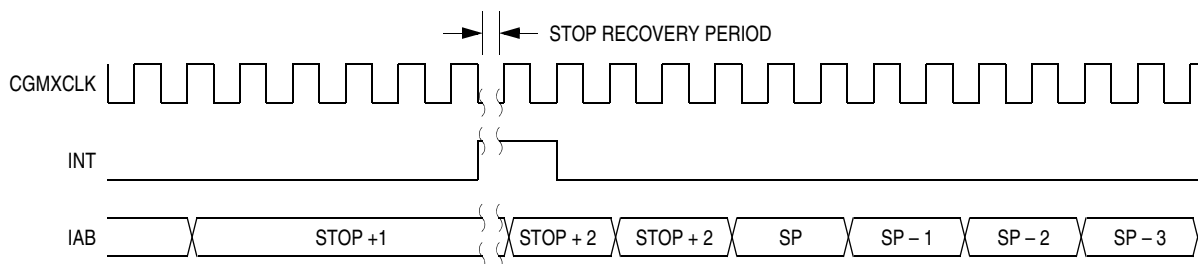
Figure 13-15. Stop Mode Recovery from Interrupt

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

13.7 SIM Registers

The SIM has four memory mapped registers described here.

1.   SIM reset status register (SRSR)
2.   Interrupt status register 1 (INT1)
3.   Interrupt status register 2 (INT2)
4.   Interrupt status register 2 (INT3)

13.7.1 SIM Reset Status Register

This register contains five bits that show the source of the last reset. The status register will clear automatically after reading it. A power-on reset sets the POR bit and clears all other bits in the register.

Address: $FE01

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

= Unimplemented

Figure 13-16. SIM Reset Status Register (SRSR)

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

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

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

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

MENRST — Forced Monitor Mode Entry Reset Bit
   1 = Last reset was caused by the MENRST circuit
   0 = POR or read of SRSR

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

13.7.2 Interrupt Status Registers

The flags in the interrupt status registers identify maskable interrupt sources. The interrupt sources and the interrupt status register flags that they set are summarized in Table 13-2. The interrupt status registers can be useful for debugging.

Table 13-2. Interrupt Sources

Source	Flag	Mask[1]	INT Register Flag	Priority[2]	Vector Address
SWI instruction	—	—	—	0	$FFFC–$FFFD
IRQ1 pin	IRQF1	IMASK1	IF1	1	$FFFA–$FFFB
ICG clock monitor	CMF	CMIE	IF2	2	$FFF8–$FFF9
TIM channel 0	CH0F	CH0IE	IF3	3	$FFF6–$FFF7
TIM channel 1	CH1F	CH1IE	IF4	4	$FFF4–$FFF5
TIM overflow	TOF	TOIE	IF5	5	$FFF2–$FFF3
SCI receiver overrun error	OR	ORIE	IF11	6	$FFE6–$FFE7
SCI receiver noise error	NF	NEIE			
SCI receiver framing error	FE	FEIE			
SCI receiver parity error	PE	PEIE			
SCI receiver full	SCRF	SCRIE	IF12	7	$FFE4–$FFE5
SCI receiver idle	IDLE	ILIE			
SCI transmitter empty	SCTE	SCTIE	IF13	8	$FFE2–$FFE3
SCI transmission complete	TC	TCIE			
Keyboard pins	KEYF	IMASKK	IF14	9	$FFE0–$FFE1
ADC conversion complete	—	AIEN	IF15	10	$FFDE–$FFDF
Timebase module	TBIE	TBF	IF16	11	$FFDC–$FFDD

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

13.7.2.1 Interrupt Status Register 1

Address:	$FE04							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R	= Reserved

Figure 13-17. Interrupt Status Register 1 (INT1)

IF5–IF1 — Interrupt Flags 5, 4, 3, 2, and 1
These flags indicate the presence of interrupt requests from the sources shown in Table 13-2.
1 = Interrupt request present
0 = No interrupt request present

IF6 — Interrupt Flag 6
   Since the MC68HC908KX8 parts do not use this interrupt flag, this bit will always read 0.

Bit 0 and Bit 1 — Always read 0

13.7.2.2 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 13-18. Interrupt Status Register 2 (INT2)

IF14–IF11 — Interrupt Flags 14–11
   These flags indicate the presence of interrupt requests from the sources shown in Table 13-2.
      1 = Interrupt request present
      0 = No interrupt request present

IF10–IF7 — Interrupt Flags 10–7
   Since the MC68HC908KX8 parts do not use these interrupt flags, these bits will always read 0.

13.7.2.3 Interrupt Status Register 3

Address: $FE06

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF22	IF21	IF20	IF19	IF18	IF17	IF16	IF15
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R	= Reserved

Figure 13-19. Interrupt Status Register 3 (INT3)

IF22–IF17 — Interrupt Flags 22–17
   Since the MC68HC908KX8 parts do not use these interrupt flags, these bits will always read 0.

IF16–IF15 — Interrupt Flags 16–15
   These flags indicate the presence of interrupt requests from the sources shown in Table 13-2.
      1 = Interrupt request present
      0 = No interrupt request present

Chapter 14
Timebase Module (TBM)

14.1 Introduction

This section describes the timebase module (TBM). The TBM will generate periodic interrupts at user selectable rates using a counter clocked by either the internal or external clock sources. This TBM version uses 15 divider stages, eight of which are user selectable.

14.2 Features

Features of the TBM module include:

- Software configurable periodic interrupts with divide by 8, 16, 32, 64, 128, 2048, 8192, and 32,768 taps of the selected clock source
- Configurable for operation during stop mode to allow periodic wake up from stop

14.3 Functional Description

This module can generate a periodic interrupt by dividing the clock source supplied from the internal clock generator module, TBMCLK. Note that this clock source is the external clock ECLK when the ECGON bit in the ICG control register (ICGCR) is set. Otherwise, TBMCLK is driven at the internally generated clock frequency (ICLK). In other words, if the external clock is enabled it will be used as the TBMCLK, even if the MCU bus clock is based on the internal clock.

The counter is initialized to all 0s when TBON bit is cleared. The counter, shown in Figure 14-1, starts counting when the TBON bit is set. When the counter overflows at the tap selected by TBR2–TBR0, the TBIF bit gets set. If the TBIE bit is set, an interrupt request is sent to the CPU. The TBIF flag is cleared by writing a 1 to the TACK bit. The first time the TBIF flag is set after enabling the timebase module, the interrupt is generated at approximately half of the overflow period. Subsequent events occur at the exact period.

The timebase module may remain active after execution of the STOP instruction if the internal clock generator has been enabled to operate during stop mode through the OSCENINSTOP bit in the configuration register. The timebase module can be used in this mode to generate a periodic wakeup from stop mode.

14.4 Interrupts

The timebase module can periodically interrupt the CPU with a rate defined by the selected TBMCLK and the select bits TBR2–TBR0. When the timebase counter chain rolls over, the TBIF flag is set. If the TBIE bit is set, enabling the timebase interrupt, the counter chain overflow will generate a CPU interrupt request.

Interrupts must be acknowledged by writing a 1 to the TACK bit.
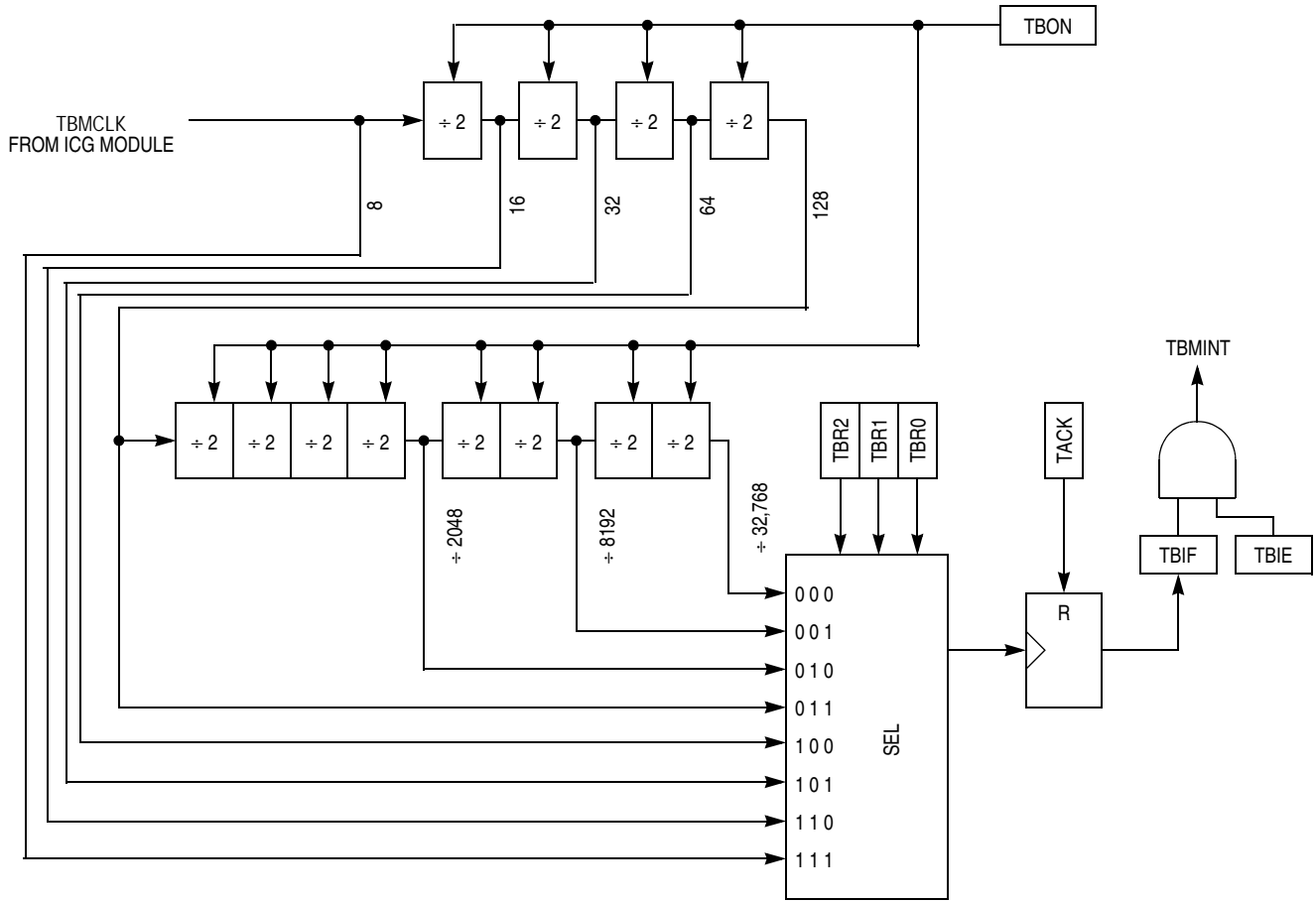
Figure 14-1. Timebase Block Diagram

14.5 TBM Interrupt Rate

The interrupt rate is determined by the equation:

$$t_{TBMRATE} = \frac{1}{f_{TBMRATE}} = \frac{Divider}{f_{TBMCLK}}$$

where:

f_{TBMCLK} = Frequency supplied from the internal clock generator (ICG) module

Divider = Divider value as determined by TBR2–TBR0 settings. See Table 14-1.

As an example, a clock source of 4.9152 MHz and the TBR2–TBR0 set to {011}, the divider tap is 128 and the interrupt rate calculates to $128/4.9152 \times 10^6 = 26$ μs.

Table 14-1. Timebase Divider Selection

TBR2	TBR1	TBR0	Divider Tap
0	0	0	32768
0	0	1	8192
0	1	0	2048
0	1	1	128
1	0	0	64
1	0	1	32
1	1	0	16
1	1	1	8

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

14.6 Low-Power Modes

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

14.6.1 Wait Mode

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

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

14.6.2 Stop Mode

The timebase module may remain active after execution of the STOP instruction if the internal clock generator has been enabled to operate during stop mode through the OSCENINSTOP bit in the configuration register. The timebase module can be used in this mode to generate a periodic wake up from stop mode.

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

If the timebase functions are not required during stop mode, reduce power consumption by disabling the timebase module before executing the STOP instruction.

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

14.7 Timebase Control Register

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

Address: $001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TBIF	TBR2	TBR1	TBR0	0	TBIE	TBON	R
Write:					TACK			
Reset:	0	0	0	0	0	0	0	0

[ ] = Unimplemented   [R] = Reserved
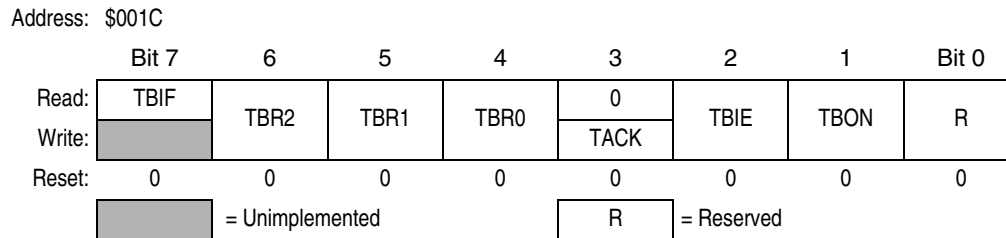
Figure 14-2. Timebase Control Register (TBCR)

TBIF — Timebase Interrupt Flag
 This read-only flag bit is set when the timebase counter has rolled over.
  1 = Timebase interrupt pending
  0 = Timebase interrupt not pending

TBR2–TBR0 — Timebase Divider Selection Bits
 These read/write bits select the tap in the counter to be used for timebase interrupts as shown in Table 14-1.

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

TACK— Timebase ACKnowledge Bit
 The TACK bit is a write-only bit and always reads as 0. Writing a 1 to this bit clears TBIF, the timebase interrupt flag bit. Writing a 0 to this bit has no effect.
  1 = Clear timebase interrupt flag
  0 = No effect

TBIE — Timebase Interrupt Enabled Bit
 This read/write bit enables the timebase interrupt when the TBIF bit becomes set. Reset clears the TBIE bit.
  1 = Timebase interrupt is enabled.
  0 = Timebase interrupt is disabled.

TBON — Timebase Enabled Bit
 This read/write bit enables the timebase. Timebase may be turned off to reduce power consumption when its function is not necessary. The counter can be initialized by clearing and then setting this bit. Reset clears the TBON bit.
  1 = Timebase is enabled.
  0 = Timebase is disabled and the counter initialized to 0s.

Chapter 15
Timer Interface Module (TIM)

15.1 Introduction

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

15.2 Features

Features include:

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

15.3 Pin Name Conventions

The TIM shares two input/output (I/O) pins with two port A I/O pins. The full names of the TIM I/O pins are listed in Table 15-1. The generic pin names appear in the text that follows.

Table 15-1. Pin Name Conventions

TIM Generic Pin Names:	**TCH0**	**TCH1**
Full TIM Pin Names:	PTA2/KBD2/TCH0	PTA3/KBD3/TCH1

15.4 Functional Description

Figure 15-2 shows the structure of the TIM. The central component of the TIM is the 16-bit TIM counter that can operate as a free-running counter or a modulo up-counter. The TIM counter provides the timing reference for the input capture and output compare functions. The TIM counter modulo registers, TMODH and TMODL, control the modulo value of the TIM counter. Software can read the TIM counter value at any time without affecting the counting sequence.

The two TIM channels are programmable independently as input capture or output compare channels.

Figure 15-3 summarizes the timer registers.
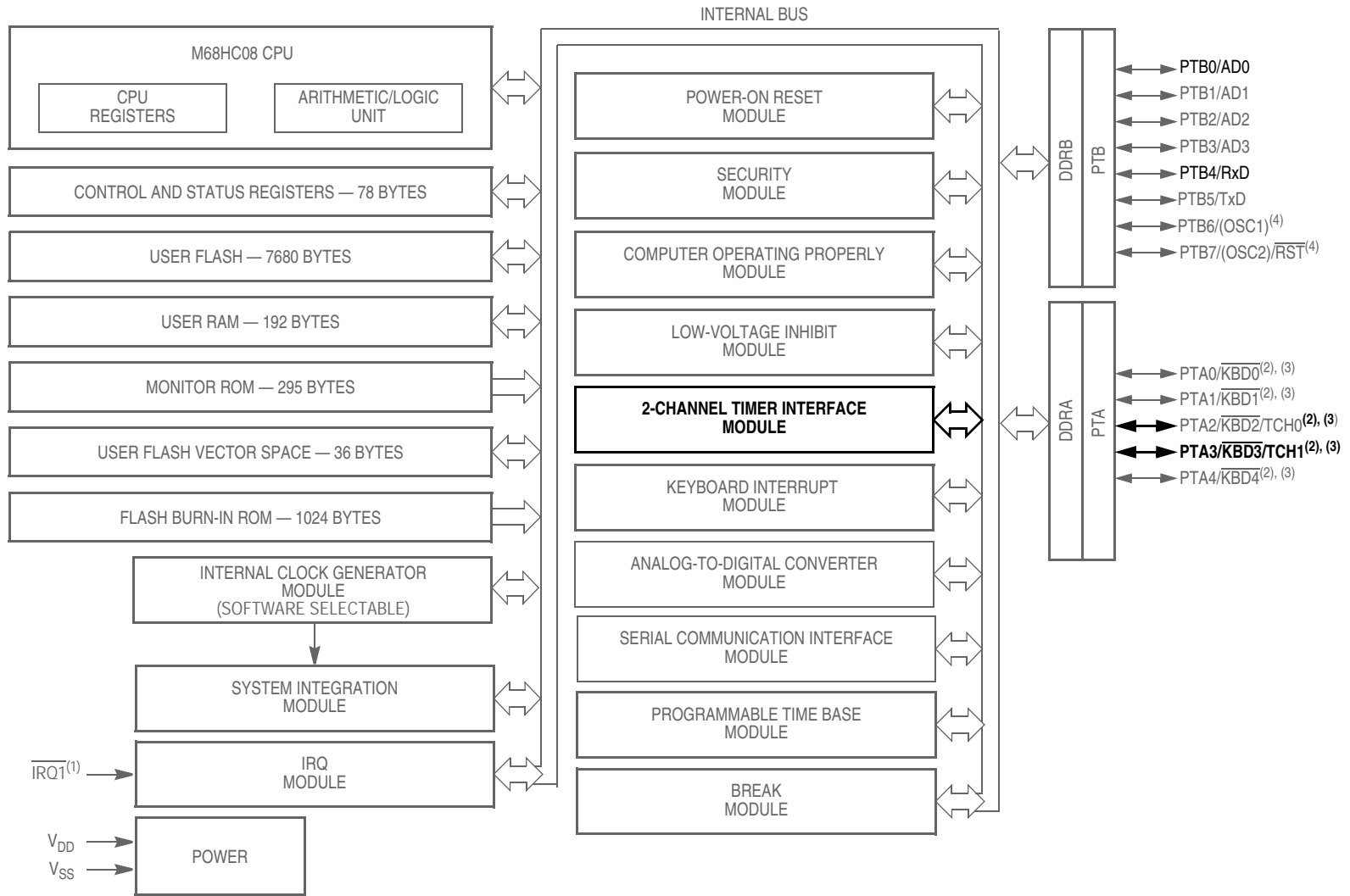
Notes:
1. Pin contains integrated pullup resistor
2. High-current source/sink pin
3. Pin contains software selectable pullup resistor if general function I/O pin is configured as input.

Figure 15-1. Block Diagram Highlighting TIM Block and Pins
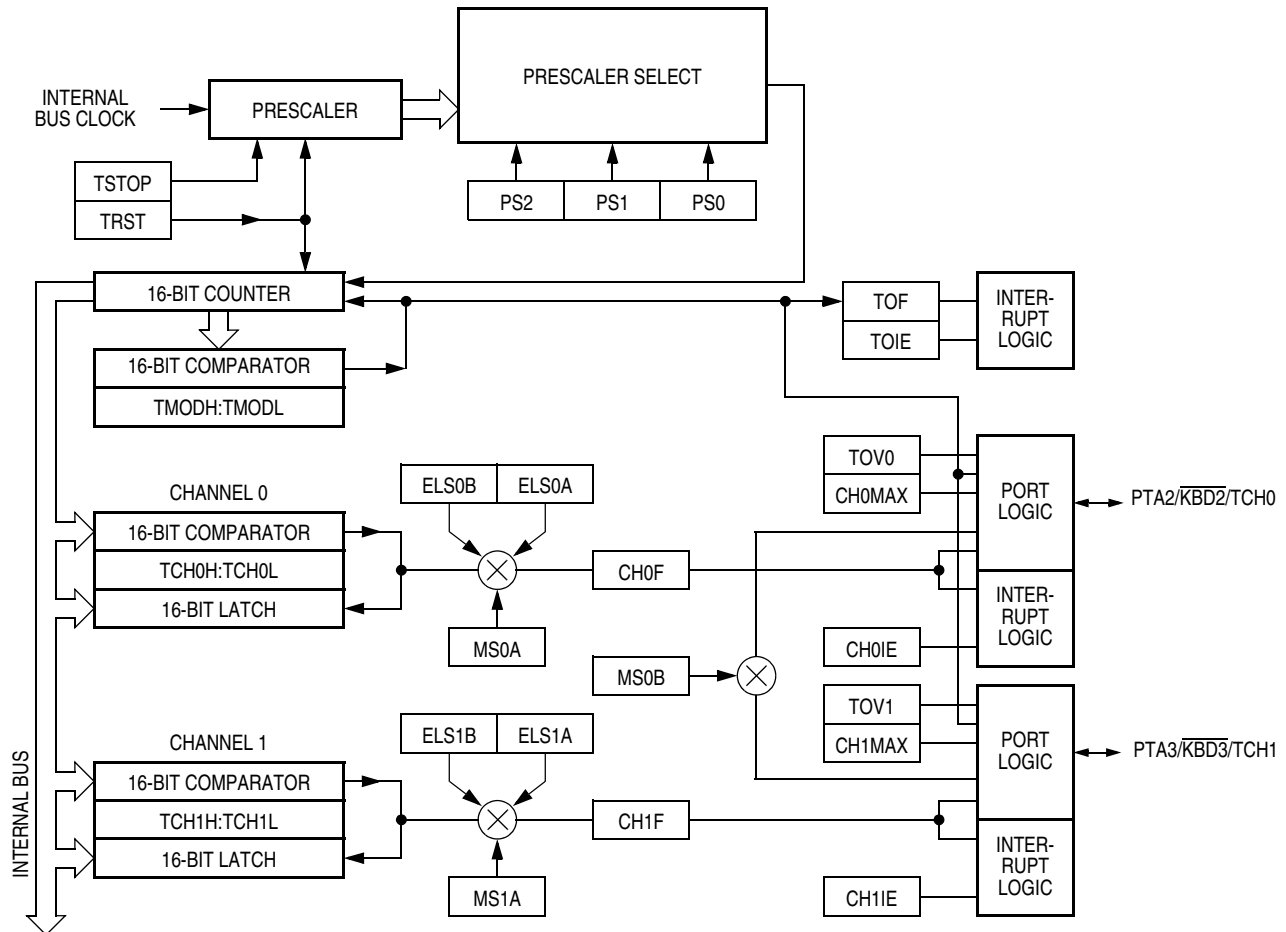
Figure 15-2. TIM Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0020	Timer Status and Control Register (TSC) See page 163.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0021	Timer Counter Register High (TCNTH) See page 164.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0022	Timer Counter Register Low (TCNTL) See page 164.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0023	Timer Counter Modulo Register High (TMODH) See page 165.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1

= Unimplemented

Figure 15-3. TIM I/O Register Summary

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0024	Timer Counter Modulo Register Low (TMODL) See page 165.	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$0025	Timer Channel 0 Status and Control Register (TSC0) See page 165.	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0026	Timer Channel 0 Register High (TCH0H) See page 168.	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							
$0027	Timer Channel 0 Register Low (TCH0L) See page 168.	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							
$0028	Timer Channel 1 Status and Control Register (TSC1) See page 165.	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0029	Timer Channel 1 Register High (TCH1H) See page 168.	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							
$002A	Timer Channel 1 Register Low (TCH1L) See page 168.	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							

= Unimplemented

Figure 15-3. TIM I/O Register Summary (Continued)

15.4.1 TIM Counter Prescaler

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

15.4.2 Input Capture

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

15.4.3 Output Compare

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

15.4.4 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in 15.4.3 Output Compare. The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIM channel registers.

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

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

  * When changing to a smaller value, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current output compare pulse. The interrupt routine has until the end of the counter overflow period to write the new value.

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

15.4.5 Buffered Output Compare

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

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

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

15.4.6 Pulse-Width Modulation (PWM)

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

As Figure 15-4 shows, the output compare value in the TIM channel registers determines the pulse width of the PWM signal. The time between overflow and output compare is the pulse width. Program the TIM

to clear the channel pin on output compare if the state of the PWM pulse is logic 1. Program the TIM to set the pin on overflow if the state of the PWM pulse is logic 0.

The value in the TIM counter modulo registers and the selected prescaler output determines the frequency of the PWM output. The frequency of an 8-bit PWM signal is variable in 256 increments. Writing $00FF (255) to the TIM counter modulo registers produces a PWM period of 256 times the internal bus clock period if the prescaler select value is $000. See 15.8.1 TIM Status and Control Register.

The value in the TIM channel registers determines the pulse width of the PWM output. The pulse width of an 8-bit PWM signal is variable in 256 increments. Writing $0080 (128) to the TIM channel registers produces a duty cycle of 128/256 or 50 percent.
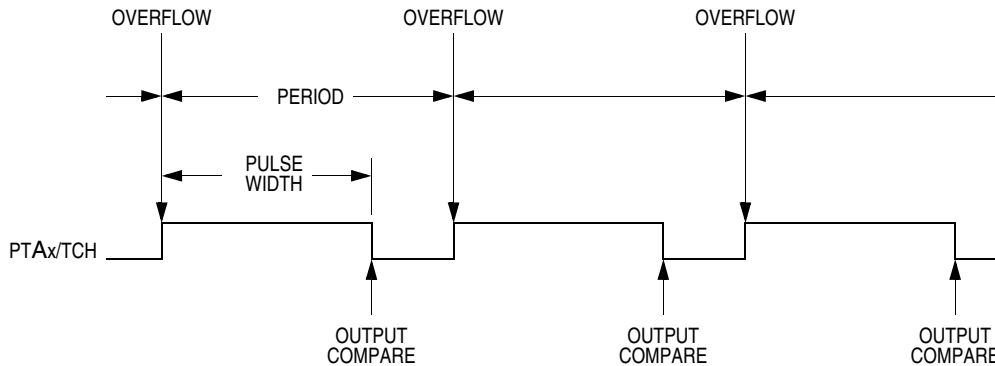
Figure 15-4. PWM Period and Pulse Width

15.4.7 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in 15.4.6 Pulse-Width Modulation (PWM). The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the old value currently in the TIM channel registers.

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

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

*   When changing to a shorter pulse width, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current pulse. The interrupt routine has until the end of the PWM period to write the new value.

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

NOTE
In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0 percent duty cycle generation and removes the ability of the channel to self-correct

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

15.4.8 Buffered PWM Signal Generation

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

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

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

15.4.9 PWM Initialization

To ensure correct operation when generating unbuffered or buffered PWM signals, use this initialization
procedure:
1. In the TIM status and control register (TSC):
    a. Stop the TIM counter by setting the TIM stop bit, TSTOP.
    b. Reset the TIM counter and prescaler by setting the TIM reset bit, TRST.
2. In the TIM counter modulo registers (TMODH and TMODL), write the value for the required PWM
   period.
3. In the TIM channel x registers (TCHxH and TCHxL), write the value for the required pulse width.
4. In TIM channel x status and control register (TSCx):
    a. Write 0:1 (for unbuffered output compare or PWM signals) or 1:0 (for buffered output compare
       or PWM signals) to the mode select bits, MSxB and MSxA. See Table 15-2.
    b. Write 1 to the toggle-on-overflow bit, TOVx.
    c. Write 1:0 (to clear output on compare) or 1:1 (to set output on compare) to the edge/level
       select bits, ELSxB and ELSxA. The output action on compare must force the output to the
       complement of the pulse width level. See Table 15-2.

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

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

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

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

Setting the channel x maximum duty cycle bit (CHxMAX) and setting the TOVx bit generates a 100 percent duty cycle output. See 15.8.4 TIM Channel Status and Control Registers.

15.5 Interrupts

These TIM sources can generate interrupt requests:

- TIM overflow flag (TOF) — The timer overflow flag (TOF) bit is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. The TIM overflow interrupt enable bit, TOIE, enables TIM overflow interrupt requests. TOF and TOIE are in the TIM status and control registers.

- TIM channel flags (CH1F and CH0F) — The CHxF bit is set when an input capture or output compare occurs on channel x. Channel x TIM CPU interrupt requests are controlled by the channel x interrupt enable bit, CHxIE. Channel x TIM CPU interrupt requests are enabled when CHxIE = 1. CHxF and CHxIE are in the TIM channel x status and control register.

15.6 Low-Power Modes

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

15.6.1 Wait Mode

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

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

15.6.2 Stop Mode

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

15.7 I/O Signals

Port A shares two of its pins with the TIM, PTA3/$\overline{\text{KBD3}}$/TCH1 and PTA2/$\overline{\text{KBD2}}$/TCH0. Each channel input/output (I/O) pin is programmable independently as an input capture pin or an output compare pin. TCH0 can be configured as buffered output compare or buffered PWM pins.

15.8 I/O Registers

These I/O registers control and monitor operation of the TIM:
- TIM status and control register (TSC)
- TIM control registers (TCNTH and TCNTL)
- TIM counter modulo registers (TMODH and TMODL)
- TIM channel status and control registers (TSC0 and TSC1)
- TIM channel registers (TCH0H and TCH0L, TCH1H and TCH1L)

15.8.1 TIM Status and Control Register

The TIM status and control register (TSC):
- Enables TIM overflow interrupts
- Flags TIM overflows
- Stops the TIM counter
- Resets the TIM counter
- Prescales the TIM counter clock

Address: $0020

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

　　　　= Unimplemented

Figure 15-5. TIM Status and Control Register (TSC)

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

TOIE — TIM Overflow Interrupt Enable Bit
　　This read/write bit enables TIM overflow interrupts when the TOF bit becomes set. Reset clears the TOIE bit.
　　　　1 = TIM overflow interrupts enabled
　　　　0 = TIM overflow interrupts disabled

TSTOP — TIM Stop Bit
　　This read/write bit stops the TIM counter. Counting resumes when TSTOP is cleared. Reset sets the TSTOP bit, stopping the TIM counter until software clears the TSTOP bit.
　　　　1 = TIM counter stopped
　　　　0 = TIM counter active

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

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

TRST — TIM Reset Bit

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

    1 = Prescaler and TIM counter cleared

    0 = No effect

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

PS2–PS0 — Prescaler Select Bits

These read/write bits select one of the seven prescaler outputs as the input to the TIM counter as Table 15-2 shows. Reset clears the PS2–PS0 bits.

Table 15-2. Prescaler Selection

PS2–PS0	TIM 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	Not available

15.8.2 TIM Counter Registers

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

Register name and address: TCNTH — $0021

	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 — $0022

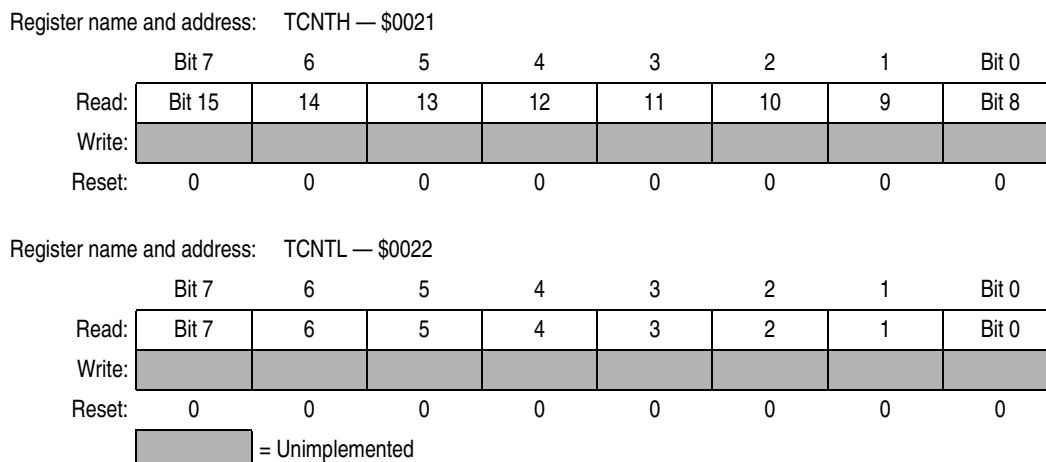|	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	0	0	0	0	0	0	0	0

    = Unimplemented

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

15.8.3 TIM Counter Modulo Registers

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

Register name and address: TMODH — $0023

	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 — $0024

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

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

15.8.4 TIM Channel Status and Control Registers

Each of the TIM channel status and control registers (TSC0 and TSC1):
- Flags input captures and output compares
- Enables input capture and output compare interrupts
- Selects input capture, output compare, or PWM operation
- Selects high, low, or toggling output on output compare
- Selects rising edge, falling edge, or any edge as the active input capture trigger
- Selects output toggling on TIM overflow
- Selects 0 percent and100 percent PWM duty cycle
- Selects buffered or unbuffered output compare/PWM operation

Register name and address: TSC0 — $0025

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

Register name and address: TSC1 — $0028

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

= Unimplemented

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

CHxF — Channel x Flag Bit

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

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

Reset clears the CHxF bit. Writing a 1 to CHxF has no effect.
  1 = Input capture or output compare on channel x
  0 = No input capture or output compare on channel x

CHxIE — Channel x Interrupt Enable Bit

This read/write bit enables TIM CPU interrupts on channel x. Reset clears the CHxIE bit.
  1 = Channel x CPU interrupt requests
  0 = Channel x CPU interrupt requests disabled

MS0B — Mode Select Bit B

This read/write bit selects buffered output compare/PWM operation. MS0B exists only in the TIM channel 0 status and control register.

Setting MS0B disables the channel 1 status and control register and reverts TCH1 to general-purpose I/O. Reset clears the MSxB bit.
  1 = Buffered output compare/PWM operation enabled
  0 = Buffered output compare/PWM operation disabled

MSxA — Mode Select Bit A

When ELSxB:A ≠ 00, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation. See Table 15-3.
  1 = Unbuffered output compare/PWM operation
  0 = Input capture operation

When ELSxB:A = 00, this read/write bit selects the initial output level of the TCHx pin. See Table 15-3. Reset clears the MSxA bit.
  1 = Initial output level low
  0 = Initial output level high

NOTE
Before changing a channel function by writing to the MS0B or MSxA bit, set the TSTOP and TRST bits in the TIM status and control register (TSC).

ELSxB and ELSxA — Edge/Level Select Bits

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

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

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

Table 15-3. Mode, Edge, and Level Selection

MSxB	MSxA	ELSxB	ELSxA	Mode	Configuration
X	0	0	0	Output preset	Pin under port control; initial output level high
X	1	0	0		Pin under port control; initial output level low
0	0	0	1	Input capture	Capture on rising edge only
0	0	1	0		Capture on falling edge only
0	0	1	1		Capture on rising or falling edge
0	1	0	0	Output compare or PWM	Software compare only
0	1	0	1		Toggle output on compare
0	1	1	0		Clear output on compare
0	1	1	1		Set output on compare
1	X	0	1	Buffered output compare or buffered PWM	Toggle output on compare
1	X	1	0		Clear output on compare
1	X	1	1		Set output on compare

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

TOVx — Toggle On Overflow Bit

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

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

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

CHxMAX — Channel x Maximum Duty Cycle Bit

When the TOVx bit is at 1 and clear output on compare is selected, setting the CHxMAX bit forces the
duty cycle of buffered and unbuffered PWM signals to 100 percent. As Figure 15-9 shows, the
CHxMAX bit takes effect in the cycle after it is set or cleared. The output stays at 100 percent duty
cycle level until the cycle after CHxMAX is cleared.

NOTE
*The PWM 0 percent duty cycle is defined as output low all of the time. To
generate the 0 percent duty cycle, select clear output on compare and then
clear the TOVx bit (CHxMAX = 0). The PWM 100 percent duty cycle is
defined as output high all of the time. To generate the 100 percent duty
cycle, use the CHxMAX bit in the TSCx register.*

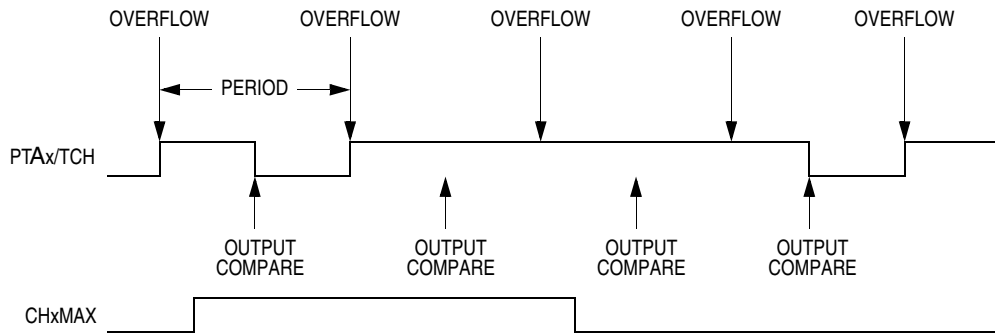**MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1**

Figure 15-9. CHxMAX Latency

15.8.5 TIM Channel Registers

These read/write registers (TCH0H/L and TCH1H/L) contain the captured TIM counter value of the input capture function or the output compare value of the output compare function. The state of the TIM channel registers after reset is unknown.

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

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

Register name and address: TCH0H — $0026

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

Register name and address: TCH0L — $0027

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

Register name and address: TCH1H — $0029

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

Register name and address: TCH1L — $002A

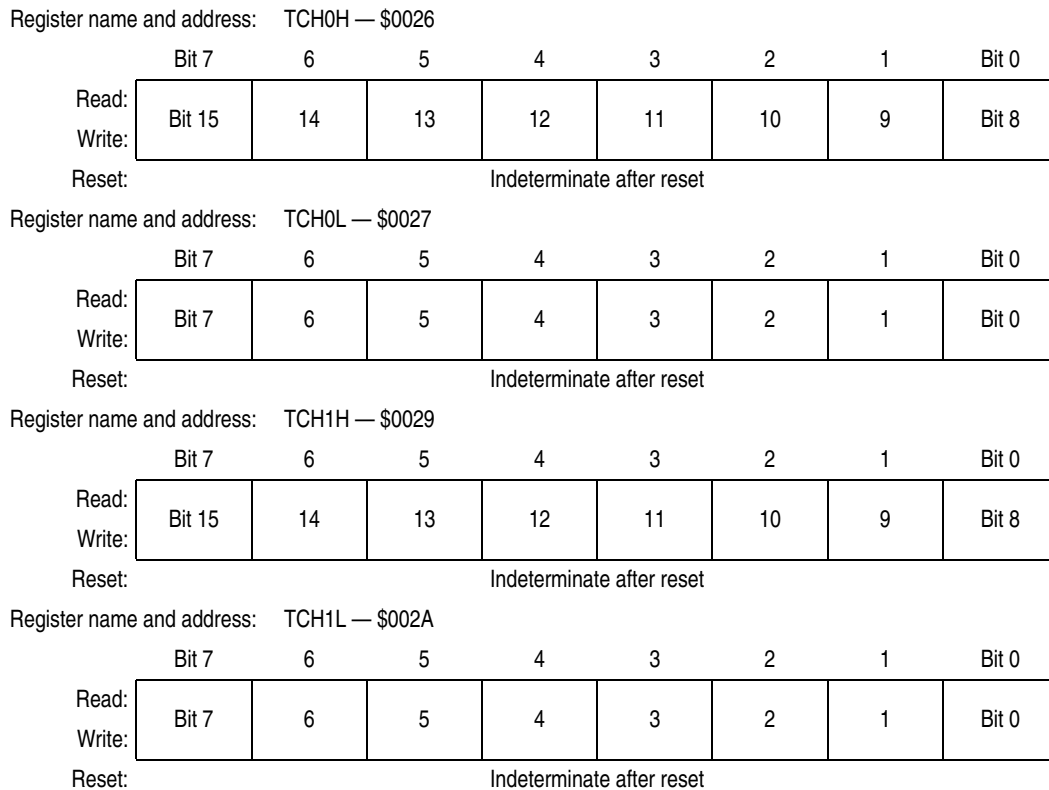|	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:				Indeterminate after reset				

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

Chapter 16
Development Support

16.1 Introduction

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

16.2 Break Module (BRK)

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

Features include:

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

16.2.1 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal 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 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 16-1 shows the structure of the break module.

16.2.1.1 Flag Protection During Break Interrupts

The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state.
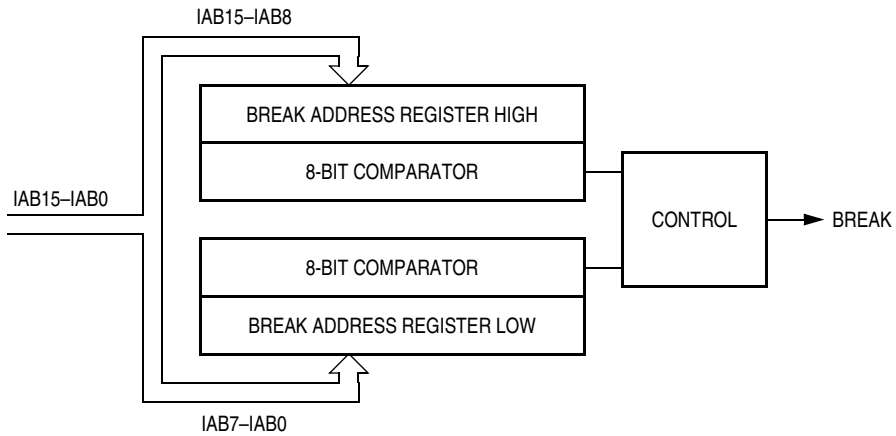
Figure 16-1. Break Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE00	SIM Break Status Register (SBSR) See page 172.	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	SIM Break Flag Control Register (SBFCR) See page 173.	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE09	Break Address Register High (BRKH) See page 172.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0A	Break Address Register Low (BRKL) See page 172.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0B	Break Status and Control Register (BRKSCR) See page 171.	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE02	Break Auxiliary Register (BRKAR) See page 173.	Read:	0	0	0	0	0	0	0	BDCOP
		Write:								
		Reset:	0	0	0	0	0	0	0	0

Note: Writing a 0 clears BW.   ▓ = Unimplemented   R = Reserved

Figure 16-2. I/O Register Summary

16.2.1.2 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.

16.2.1.3 TIM1 and TIM2 During Break Interrupts

A break interrupt stops the timer counters.

16.2.1.4 COP During Break Interrupts

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

16.2.2 Break Module Registers

These registers control and monitor operation of the break module:

- Break status and control register (BRKSCR)
- Break address register high (BRKH)
- Break address register low (BRKL)
- SIM break status register (SBSR)
- SIM break flag control register (SBFCR)

16.2.2.1 Break Status and Control Register

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

Address: $FE0B

	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 16-3. Break Status and Control Register (BRKSCR)

BRKE — Break Enable Bit
    This read/write bit enables breaks on break address register matches. Clear BRKE by writing a 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 1 to BRKA generates a break interrupt. Clear BRKA by writing a 0 to it before exiting the break routine. Reset clears the BRKA bit.
        1 = When read, break address match
        0 = When read, no break address match

16.2.2.2 Break Address Registers

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

Address:	$FE09							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
Reset:	0	0	0	0	0	0	0	0

Figure 16-4. Break Address Register High (BRKH)

Address:	$FE0A							
	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 16-5. Break Address Register Low (BRKL)

16.2.2.3 Break Status Register

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

Address:	$FE00							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	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 0 clears BW. R = Reserved

Figure 16-6. SIM Break Status Register (SBSR)

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 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.

16.2.2.4 Break Flag Control Register

The break flag control register (SBFCR) contains a bit that enables software to clear status bits while the MCU is in a break state.
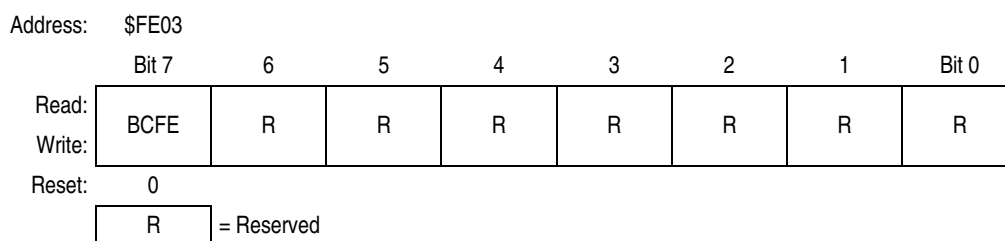
Address: $FE03

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

R	= Reserved

Figure 16-7. SIM Break Flag Control Register (SBFCR)

BCFE — Break Clear Flag Enable Bit
This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.
1 = Status bits clearable during break
0 = Status bits not clearable during break

16.2.2.5 Break Auxiliary Register

The break auxiliary register (BRKAR) contains a bit that enables software to disable the COP while the MCU is in a state of break interrupt with monitor mode.
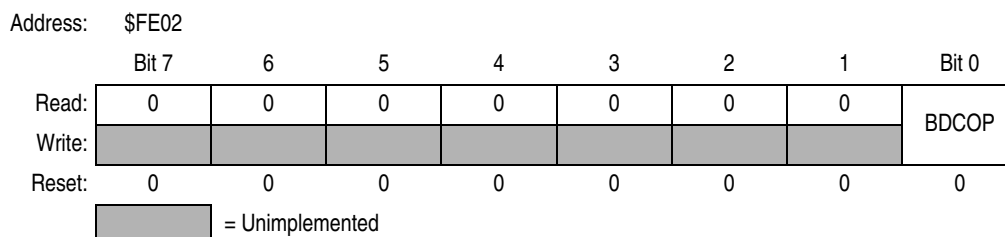
Address: $FE02

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

	= Unimplemented

Figure 16-8. Break Auxiliary Register (BRKAR)

BDCOP — Break Disable COP Bit
This read/write bit disables the COP during a break interrupt. Reset clears the BDCOP bit.
1 = COP disabled during break interrupt
0 = COP enabled during break interrupt

16.2.3 Low-Power Modes

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

16.2.3.1 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. Clear the BW bit by writing 0 to it.

16.2.3.2 Stop Mode

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

16.3 Monitor ROM (MON)

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

Features 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 FLASH
- FLASH memory security[1]
- FLASH memory programming interface
- Monitor mode entry without high voltage, V_{TST}, if reset vector is blank ($FFFE and $FFFF contain $FF)
- Standard monitor mode entry if high voltage, V_{TST}, is applied to $\overline{IRQ}$

16.3.1 Functional Description

The monitor ROM receives and executes commands from a host computer via a standard RS-232 interface. Simple monitor commands can access any memory address. In monitor mode, the microcontroller unit (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.

16.3.1.1 Monitor Mode Entry

There are two methods for entering monitor mode. The first is the traditional M68HC08 method where V_{TST} is applied to $\overline{IRQ1}$ and the mode pins are configured appropriately. A second method, intended for in-circuit programming applications, will force entry into monitor mode without requiring high voltage on the $\overline{IRQ1}$ pin when the reset vector locations of the FLASH are erased ($FF).

Both of these methods require that the PTA1 pin be pulled low for the first 24 CGMXCLK cycles after the part comes out of reset. This check is used by the monitor code to configure the MCU for serial communication.

16.3.1.2 Normal Monitor Mode

Normal monitor mode is useful for MCU evaluation, factory testing, and development tool programming operation. Figure 16-9 shows an example circuit used for normal monitor mode. Table 16-1 shows the pin conditions for entering this mode.

---

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.
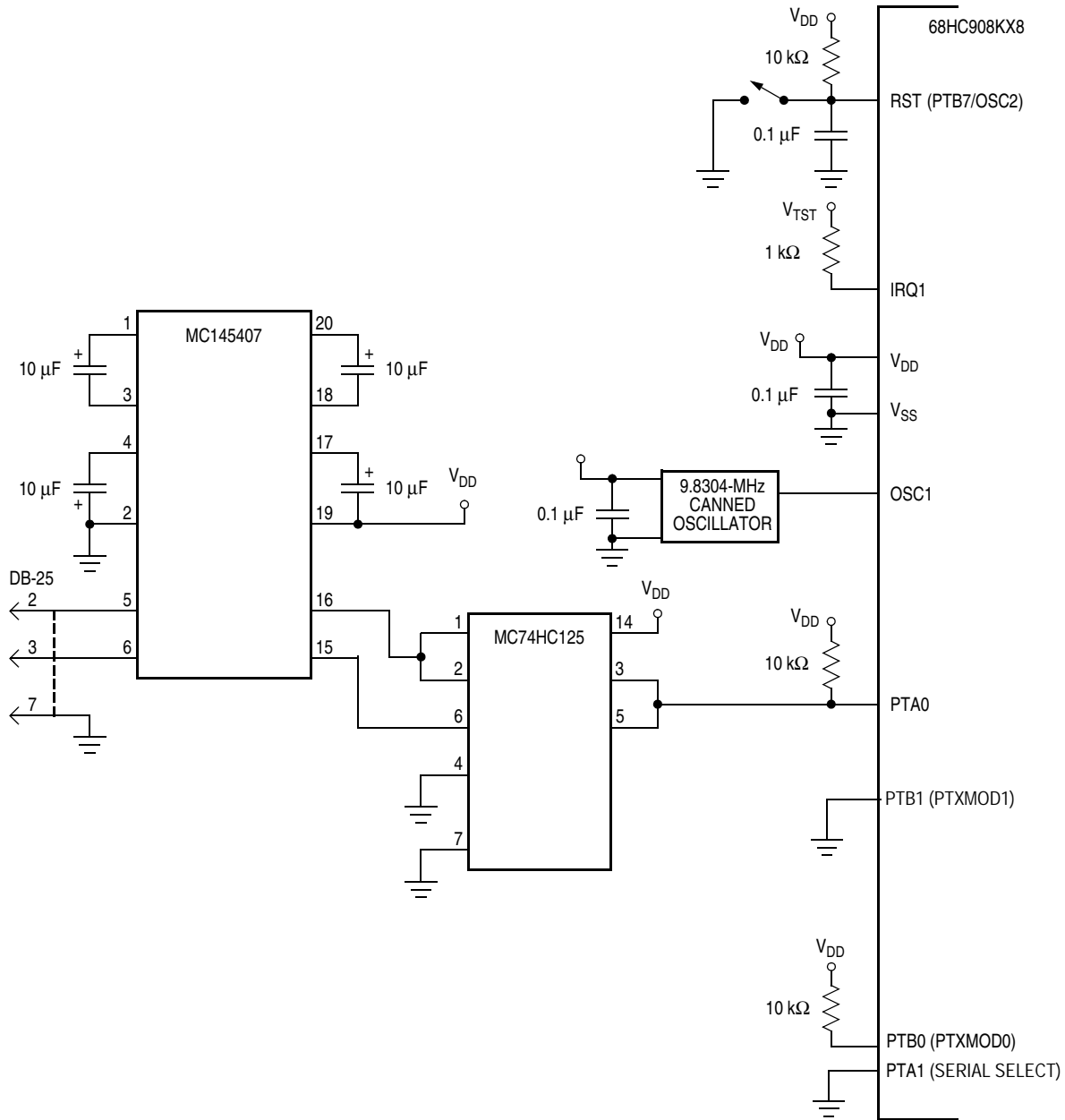
Figure 16-9. Normal Monitor Mode Circuit

Table 16-1. Monitor Mode Entry

$FFFE/ $FFFF	$\overline{\text{IRQ1}}$ Pin	PTB1 Pin (PTXMOD1)	PTB0 Pin (PTXMOD0)	PTA1 Pin	PTA0 Pin	CGMOUT	Bus Frequency (f_{OP})
X	V_{TST}	0	1	0	1	$\dfrac{\text{CGMXCLK}}{2}$	$\dfrac{\text{CGMOUT}}{2}$
$FF blank	V_{DD}	X	X	0	1	$\dfrac{\text{CGMXCLK}}{2}$	$\dfrac{\text{CGMOUT}}{2}$

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

NOTE

*PTA1 = 0 and PTA0 = 1 allow normal serial communications. PTA1 = 1
allows parallel communications during security code entry. (For parallel
communications, configure PTA0 = 0 or PTA0 = 1.)*

The MCU initially comes out of reset using the external clock for its clock source. This overrides the user
mode operation of the oscillator circuits where the part comes up using the internally generated oscillator.
Running from an external clock allows the MCU, using an appropriate frequency clock source, to
communicate with host software at standard baud rates.

NOTE

*While the voltage on $\overline{IRQ1}$ is at V_{TST}, the ICG module is bypassed and the
external square-wave clock becomes the clock source. Dropping $\overline{IRQ1}$ to
below V_{TST} will remove the bypass and the MCU will revert to the clock
source selected by the ICG (as determined by the settings in the ICG
registers).*

*In normal monitor mode with V_{TST} on $\overline{IRQ1}$, the MCU alters
PTB7/(OSC2)/$\overline{RST}$ to function as a $\overline{RST}$ pin. This is useful for testing the
MCU. Dropping $\overline{IRQ1}$ voltage to below V_{TST} will revert PTB7/(OSC2)/$\overline{RST}$
to its user mode function.*

The computer operating properly (COP) module is disabled in normal monitor mode whenever V_{TST} is
applied to the $\overline{IRQ1}$ pin. If the voltage on $\overline{IRQ1}$ is less than V_{TST}, the COP module is controlled by the
COPD configuration bit.

16.3.1.3 Forced Monitor Mode

If the voltage applied to the $\overline{IRQ1}$ is less than V_{TST}, the MCU will come out of reset in user mode. The
MENRST module is monitoring the reset vector fetches and will assert an internal reset if it detects that
the reset vectors are erased ($FF). When the MCU comes out of reset, it is forced into monitor mode
without requiring high voltage on the $\overline{IRQ1}$ pin.

Once out of reset, the monitor code is initially executing off the internal clock at its default frequency. The
monitor code reconfigures the ICG module to use the external square-wave clock source. Switching to an
external clock source allows the MCU, using an appropriate clock frequency, to communicate with host
software at standard baud rates.

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

16.3.1.4 Monitor Mode Vectors

Monitor mode uses alternate vectors for reset and SWI interrupts. 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. Table 16-2 shows vector differences between user mode and monitor mode.

Table 16-2. Monitor Mode Vector Relocation

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

16.3.1.5 Data Format

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

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

Figure 16-10. Monitor Data Format

16.3.1.6 Break Signal

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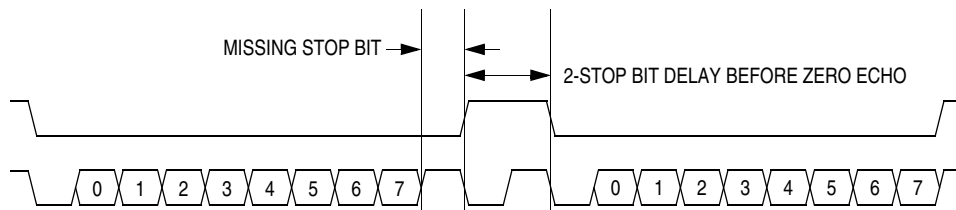

Figure 16-11. Break Transaction

16.3.1.7 Baud Rate

The communication baud rate is controlled by the CGMXCLK frequency output of the internal clock generator module.

16.3.1.8 Force Monitor Mode

In forced monitor mode, the baud rate is fixed at CGMXCLK/1024. A CMGXCLK frequency of 4.9152 MHz results in a 4800 baud rate. A 9.8304-MHz frequency produces a 9600 baud rate.

16.3.1.9 Normal Monitor Mode

In normal monitor mode, the communication baud rate is controlled by the CGMXCLK frequency output of the internal clock generator module. Table 16-3 lists CGMXCLK frequencies required to achieve standard baud rates. Other standard baud rates can be accomplished using other clock frequencies. The internal clock can be used as the clock source by programming the internal clock generator registers however, monitor mode will always be entered using the external clock as the clock source.

Table 16-3. Normal Monitor Mode Baud Rate Selection

CGMXCLK Frequency (MHz)	Baud Rate
9.8304	9600

16.3.1.10 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
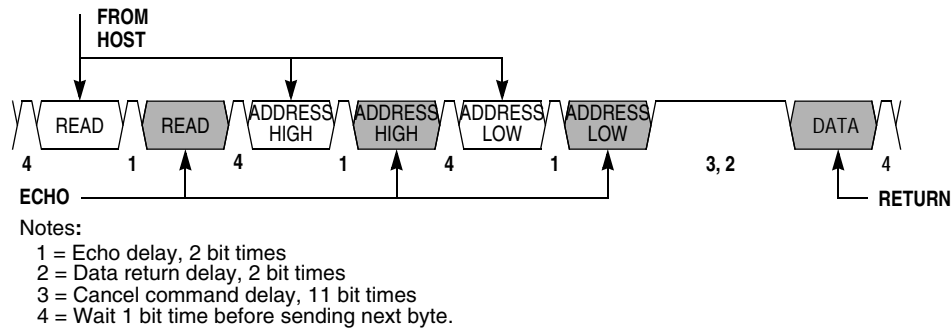*Wait one bit time after each echo before sending the next byte.*

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 16-12. Read Transaction

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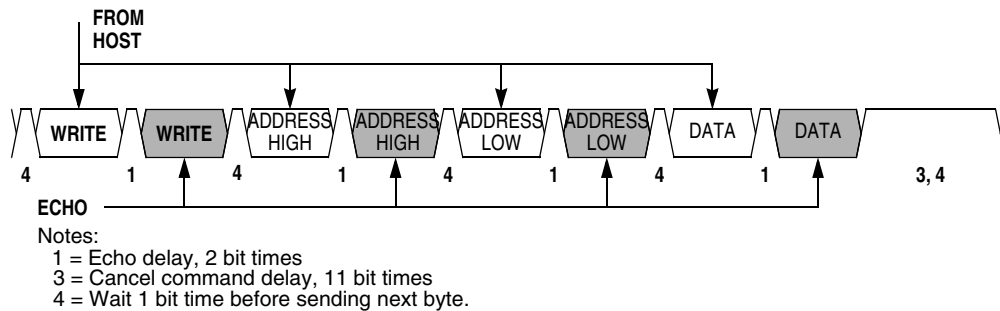4 = Wait 1 bit time before sending next byte.

Figure 16-13. Write Transaction

A brief description of each monitor mode command is given in Table 16-4 through Table 16-9.

Table 16-4. 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 16-5. 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 16-6. 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 16-7. IWRITE (Indexed Write) Command

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

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

Table 16-8. 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 16-9. 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.
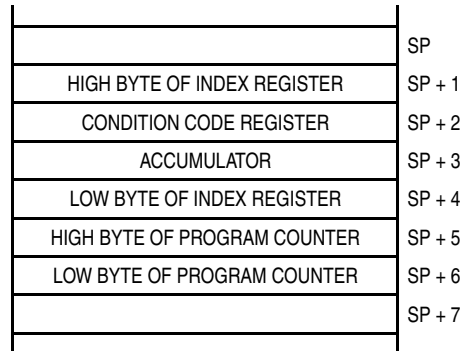
	SP
HIGH BYTE OF INDEX REGISTER	SP + 1
CONDITION CODE REGISTER	SP + 2
ACCUMULATOR	SP + 3
LOW BYTE OF INDEX REGISTER	SP + 4
HIGH BYTE OF PROGRAM COUNTER	SP + 5
LOW BYTE OF PROGRAM COUNTER	SP + 6
	SP + 7

Figure 16-14. Stack Pointer at Monitor Mode Entry

16.3.2 Security

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

> ### *NOTE*
> *Do not leave locations $FFF6–$FFFD blank. For security reasons, program locations $FFF6–$FFFD even if they are not used for vectors. If FLASH is erased, the eight security byte values to be sent to the MCU are $FF, the unprogrammed state of the FLASH.*

During monitor mode entry, a reset must be asserted. PTA1 must be held low during the reset and 24 CGMXCLK cycles after the end of the reset. Then the MCU will wait for eight security bytes on PTA0. Each byte will be echoed back to the host. See Figure 16-15.

If the received bytes match those at locations $FFF6–$FFFD, the host bypasses the security feature and can read all FLASH locations and execute code from FLASH. Security remains bypassed until a reset occurs. After any reset, security will be locked. To bypass security again, the host must resend the eight security bytes on PTA0.

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 FLASH locations returns undefined data, and trying to execute code from FLASH causes an illegal address reset.
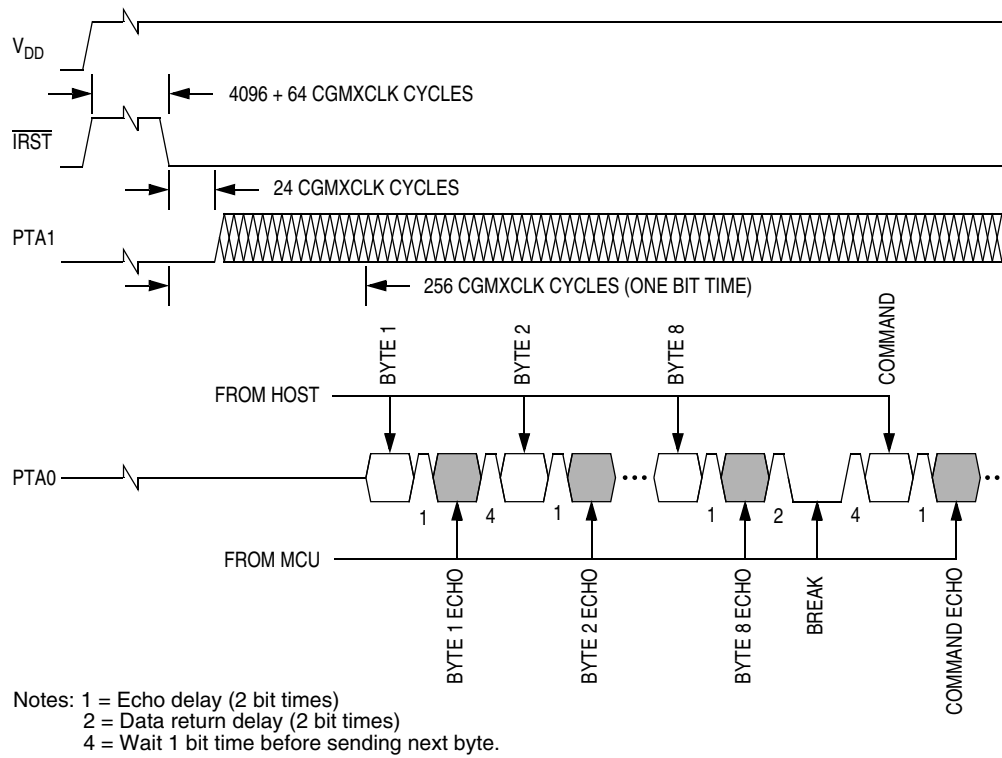
MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Notes: 1 = Echo delay (2 bit times)
2 = Data return delay (2 bit times)
4 = Wait 1 bit time before sending next byte.

Figure 16-15. Monitor Mode Entry Timing

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.

Chapter 17
Electrical Specifications

17.1 Introduction

This section contains electrical and timing specifications.

17.2 Absolute Maximum Ratings

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

> **NOTE**
>
> *This device is not guaranteed to operate properly at the maximum ratings. Refer to 17.5 5.0-Vdc DC Electrical Characteristics, and   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}, V_{SS}, and PTA0–PTA4	I	±15	mA
Maximum current for pins PTA0–PTA4	I_{PTA0}–I_{PTA4}	±25	mA
Maximum current out of V_{SS}	I_{MVSS}	100	mA
Maximum current into V_{DD}	I_{MVDD}	100	mA
Storage temperature	T_{STG}	–55 to +150	°C

1. Voltages referenced to V_{SS}

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

17.3 Functional Operating Range

Characteristic	Symbol	Value	Unit
Operating temperature range	T_A	−40 to 125	°C
Operating voltage range	V_{DD}	$3.0 \pm 10\%$ $5.0 \pm 10\%$	V

17.4 Thermal Characteristics

Characteristic	Symbol	Value	Unit

Thermal resistance
    PDIP (16 pins)
    SOIC (16 pins)
	θ_{JA}	66
		95
			°C/W

I/O pin power dissipation	$P_{I/O}$	User determined	W
Power dissipation[1]	P_D	$P_D = (I_{DD} \times V_{DD}) + P_{I/O} =$ $K/(T_J + 273°C)$	W
Constant[2]	K	$P_D \times (T_A + 273°C)$ $+ P_D^2 \times \theta_{JA}$	W/°C
Average junction temperature	T_J	$T_A + (P_D \times \theta_{JA})$	°C
Maximum junction temperature	T_{JM}	135	°C

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

17.5 5.0-Vdc DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage I_{Load} = –2.0 mA, all I/O pins I_{Load} = –10.0 mA, all I/O pins I_{Load} = –15.0 mA, PTA0–PTA4 only	V_{OH}	V_{DD} –0.4 V_{DD} –1.5 V_{DD} –0.8	— — —	— — —	V
Output low voltage I_{Load} = 1.6 mA, all I/O pins I_{Load} = 10.0 mA, all I/O pins I_{Load} = 15.0 mA, PTA0–PTA4 only	V_{OL}	— — —	— — —	0.4 1.5 0.8	V
Input high voltage — all ports, $\overline{IRQ1}$	V_{IH}	0.7 x V_{DD}	—	V_{DD} + 0.3	V
Input low voltage — all ports, $\overline{IRQ1}$	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
V_{DD} supply current Run[3], [4] Wait[4], [5] Stop, 25°C[6]	I_{DD}	— — —	15 2.2 0.8	25 5 1.75	mA mA μA
I/O ports Hi-Z leakage current[7]	I_{IL}	–10	—	+10	μA
Input current	I_{In}	–10	—	+10	μA
Capacitance Ports (as input or output)	C_{Out} C_{In}	— —	— —	12 8	pF
POR rearm voltage[8]	V_{POR}	0	—	100	mV
POR reset voltage[9]	V_{POR}	0	700	800	mV
POR rise time ramp rate	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{TST}	V_{DD}+ 2.5		V_{DD}+ 4.0	V
Low-voltage inhibit reset, trip falling voltage	V_{TRIPF}	3.90	4.25	4.50	V
Low-voltage inhibit reset, trip rising voltage	V_{TRIPR}	4.20	4.35	4.60	V
Low-voltage inhibit reset/recover hysteresis	V_{HYS}	—	100	—	mV
Pullup resistor PTA0–PTA4, $\overline{IRQ1}$	R_{PU}	24	—	48	kΩ

1. V_{DD} = 5.5 Vdc to 4.5 Vdc, V_{SS} = 0 Vdc, T_A = –40°C to +125°C, unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range, 25°C only.
3. Run (operating) I_{DD} measured using internal oscillator at its 32-MHz rate. V_{DD} = 5.5 Vdc. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. All ports configured as inputs. Measured with all modules enabled.
4. All measurements taken with LVI enabled.
5. Wait I_{DD} measured using internal oscillator at its 1-MHz rate. All inputs 0.2 V from rail; no dc loads; less than 100 pF on all outputs. All ports configured as inputs.
6. Stop I_{DD} is measured with no port pin sourcing current; all modules are disabled. OSCSTOPEN option is not selected.
7. Pullups and pulldowns are disabled.
8. Maximum is highest voltage that POR is guaranteed.
9. Maximum is highest voltage that POR is possible.

17.6 3.0-Vdc DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage I_{Load} = −0.6 mA, all I/O pins I_{Load} = −4.0 mA, all I/O pins I_{Load} = −10 mA, PTA0–PTA4 only	V_{OH}	V_{DD} −0.3 V_{DD} −1.0 V_{DD} −0.6	— — —	— — —	V
Output low voltage I_{Load} = 0.5 mA, all I/O pins I_{Load} = 6.0 mA, all I/O pins I_{Load} = 10 mA, PTA0–PTA4 only	V_{OL}	— — —	— — —	0.3 1.0 0.6	V V V
Input high voltage — all ports, $\overline{IRQ1}$	V_{IH}	0.7 x V_{DD}	—	V_{DD} + 0.3	V
Input low voltage — all ports, $\overline{IRQ1}$	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
V_{DD} supply current Run[3], [4] Wait[4], [5] Stop, 25°C[6]	I_{DD}	— — —	5 1 0.65	10 2.5 1.25	mA mA μA
I/O ports Hi-Z leakage current[7]	I_{IL}	−10	—	+10	μA
Input current	I_{In}	−10	—	+10	μA
Capacitance Ports (as input or output)	C_{Out} C_{In}	— —	— —	12 8	pF
POR rearm voltage[8]	V_{POR}	0	—	100	mV
POR reset voltage[9]	V_{POR}	0	700	800	mV
POR rise time ramp rate	R_{POR}	0.02	—	—	V/ms
Monitor mode entry voltage	V_{TST}	V_{DD}+ 2.5	—	V_{DD}+ 4.0	V
Low-voltage inhibit reset, trip falling voltage	V_{TRIPF}	2.45	2.60	2.70	V
Low-voltage inhibit reset, trip rising voltage	V_{TRIPR}	2.55	2.66	2.80	V
Low-voltage inhibit reset/recover hysteresis	V_{HYS}	—	60	—	mV
Pullup resistor PTA0–PTA4, $\overline{IRQ1}$	R_{PU}	24	—	48	kΩ

1. V_{DD} = 3.3 to 2.7 Vdc, V_{SS} = 0 Vdc, T_A = −40°C to +125°C, unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range, 25°C only.
3. Run (operating) I_{DD} measured using internal oscillator at its 16-MHz rate. V_{DD} = 3.3 Vdc. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. All ports configured as inputs. Measured with all modules enabled.
4. All measurements taken with LVI enabled.
5. Wait I_{DD} measured using internal oscillator at its 1 MHz rate. All inputs 0.2 V from rail; no dc loads; less than 100 pF on all outputs. All ports configured as inputs.
6. Stop I_{DD} is measured with no port pins sourcing current; all modules are disabled.
7. Pullups and pulldowns are disabled.
8. Maximum is highest voltage that POR is guaranteed.
9. Maximum is highest voltage that POR is possible.

17.7 Internal Oscillator Characteristics

Characteristic[1]	Symbol	Min	Typ	Max	Unit
Internal oscillator base frequency[2], [3]	f_{INTOSC}	230.4	307.2	384	kHz
Internal oscillator tolerance	f_{OSC_TOL}	−25	—	+25	%
Internal oscillator multiplier[4]	N	1	—	127	—

1. V_{DD} = 5.5 Vdc to 2.7 Vdc, V_{SS} = 0 Vdc, T_A = −40°C to +125°C, unless otherwise noted
2. Internal oscillator is selectable through software for a maximum frequency. Actual frequency will be multiplier (N) x base frequency.
3. f_{Bus} = (f_{INTOSC} / 4) x N when internal clock source selected
4. Multiplier must be chosen to limit the maximum bus frequency of 4 MHz for 2.7-V operation and 8 MHz for 4.5-V operation.

17.8 External Oscillator Characteristics

Characteristic[1]	Symbol	Min	Typ	Max	Unit

External clock option[2], [3]
  With ICG clock disabled
  With ICG clock enabled
  EXTSLOW = 1[4]
  EXTSLOW = 0[4]
	f_{EXTOSC}	dc[5]
		
		60
		307.2 k
			—
			
			—
			—
				32 M[6]
				
				307.2 k
				32 M[6]
					Hz

External crystal options[7], [8]
  EXTSLOW = 1[4]
  EXTSLOW = 0[4]
	f_{EXTOSC}	30 k
		1 M
			—
			—
				100 k
				8 M
					Hz

Crystal load capacitance[9]	C_L	—	—	—	pF
Crystal fixed capacitance[9]	C_1	—	2 x C_L	—	pF
Crystal tuning capacitance[9]	C_2	—	2 x C_L	—	pF
Feedback bias resistor[9]	R_B	—	10	—	MΩ
Series resistor [9], [10]	R_S	—	—	—	MΩ

1. V_{DD} = 5.5 to 2.7 Vdc, V_{SS} = 0 Vdc, T_A = −40°C to +125°C, unless otherwise noted
2. Setting EXTCLKEN configuration option enables OSC1 pin for external clock square-wave input.
3. No more than 10% duty cycle deviation from 50%
4. EXTSLOW configuration option configures external oscillator for a slow speed crystal and sets the clock monitor circuits of the ICG module to expect an external clock frequency that is higher/lower than the internal oscillator base frequency, f_{INTOSC}.
5. Some modules may require a minimum frequency greater than dc for proper operation. See appropriate table for this information.
6. MCU speed derates from 32 MHz at V_{DD} = 4.5 Vdc to 16 MHz at V_{DD} = 2.7 Vdc.
7. Setting EXTCLKEN and EXTXTALEN configuration options enables OSC1 and OSC2 pins for external crystal   option.
8. f_{Bus} = (f_{EXTOSC} / 4) when external clock source is selected.
9. Consult crystal vendor data sheet, see Figure 7-3. External Clock Generator Block Diagram.
10. Not required for high-frequency crystals

17.9 Trimmed Accuracy of the Internal Clock Generator

The unadjusted frequency of the low-frequency base clock (IBASE), when the comparators in the frequency comparator indicate zero error, can vary as much as ±25% due to process, temperature, and voltage. The trimming capability exists to compensate for process affects. The remaining variation in frequency is due to temperature, voltage, and change in target frequency (multiply register setting). These affects are designed to be minimal, however variation does occur. Better performance is seen at 3 V and lower settings of N.

17.9.1 2.7-Volt to 3.3-Volt Trimmed Internal Clock Generator Characteristics

Characteristic[1]	Symbol	Min	Typ	Max	Unit

Absolute trimmed internal oscillator tolerance[2], [3]
  −40°C to 85°C
  −40°C to 125°C
	F_{abs_tol}	—
		—
			2.5
			4.0
				5.0
				5.7
					%

Variation over temperature[3], [4]	V_{ar_temp}	—	0.03	0.05	%/C

Variation over voltage[3], [5]
  25°C
  −40°C to 85°C
  −40°C to 125°C
	V_{ar_volt}	—
		—
		—
			0.5
			0.7
			0.7
				2.0
				2.0
				2.0
					%/V

1. These specifications concern long-term frequency variation. Each measurement is taken over a 1-ms period.
2. Absolute value of variation in ICG output frequency, trimmed at nominal V_{DD} and temperature, as temperature and V_{DD} are allowed to vary for a single given setting of N.
3. Specification is characterized but not tested.
4. Variation in ICG output frequency for a fixed N and voltage
5. Variation in ICG output frequency for a fixed N

17.9.2 4.5-Volt to 5.5-Volt Trimmed Internal Clock Generator Characteristics

Characteristic[1]	Symbol	Min	Typ	Max	Unit

Absolute trimmed internal oscillator tolerance[2], [3]
  −40°C to 85°C
  −40°C to 125°C
	F_{abs_tol}	—
		—
			4.0
			5.0
				7.0
				10.0
					%

Variation over temperature[3], [4]	V_{ar_temp}	—	0.05	0.08	%/C

Variation over voltage[3], [5]
  25°C
  −40°C to 85°C
  −40°C to 125°C
	V_{ar_volt}	—
		—
		—
			1.0
			1.0
			1.0
				2.0
				2.0
				2.0
					%/V

1. These specifications concern long-term frequency variation. Each measurement is taken over a 1-ms period.
2. Absolute value of variation in ICG output frequency, trimmed at nominal V_{DD} and temperature, as temperature and V_{DD} are allowed to vary for a single given setting of N.
3. Specification is characterized but not tested.
4. Variation in ICG output frequency for a fixed N and voltage
5. Variation in ICG output frequency for a fixed N

Figure 17-1 through Figure 17-4 illustrate typical performance. The formula for this variation of frequency is (measured-nominal)/nominal. Figure 17-1 shows the variation in ICG frequency for a part trimmed at nominal voltage and temperature across V_{DD} and temperature for a 3-V application with multiply register (N) set to 1. Figure 17-2 shows 5 V.
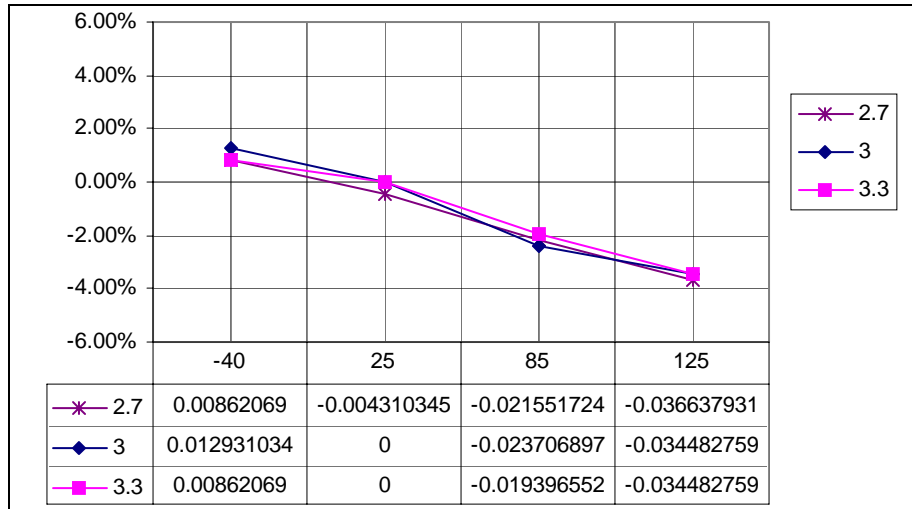

	-40	25	85	125
2.7	0.00862069	-0.004310345	-0.021551724	-0.036637931
3	0.012931034	0	-0.023706897	-0.034482759
3.3	0.00862069	0	-0.019396552	-0.034482759

**Figure 17-1. Example of Frequency Variation Across Temperature,
Trimmed at Nominal 3 Volts, 25°C, and N = 1**

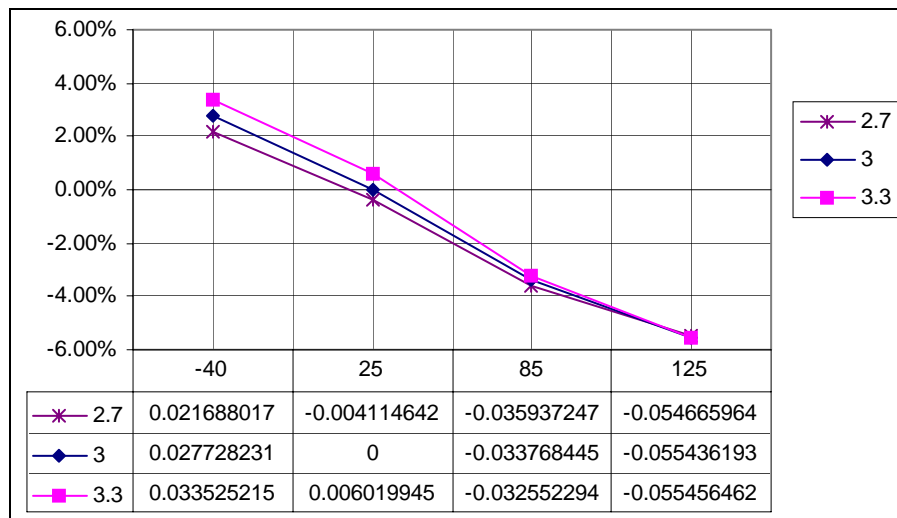|	-40	25	85	125
2.7	0.021688017	-0.004114642	-0.035937247	-0.054665964
3	0.027728231	0	-0.033768445	-0.055436193
3.3	0.033525215	0.006019945	-0.032552294	-0.055456462

**Figure 17-2. Example of Frequency Variation Across Temperature,
Trimmed at Nominal 3 Volts, 25°C, and N = 104**

Figure 17-3 and Figure 17-4 shows N set to 104, hex 68, which corresponds to an ICG frequency of 31.9 MHz or 7.9 MHz bus.

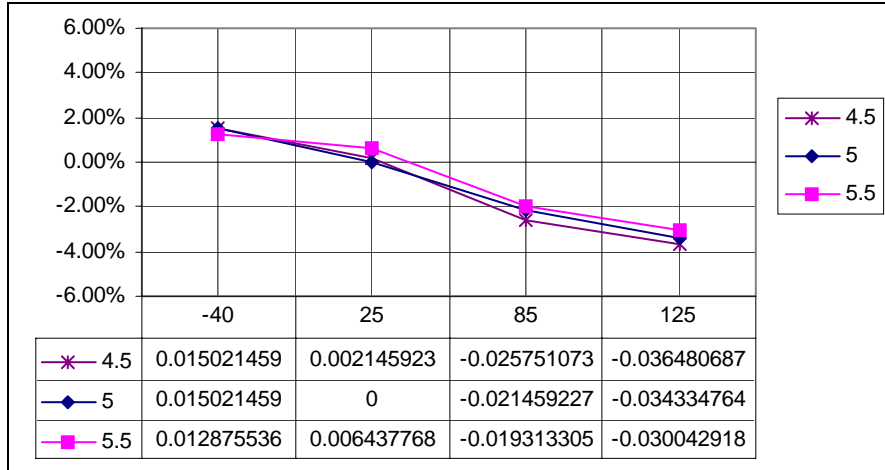|	-40	25	85	125
4.5	0.015021459	0.002145923	-0.025751073	-0.036480687
5	0.015021459	0	-0.021459227	-0.034334764
5.5	0.012875536	0.006437768	-0.019313305	-0.030042918

Figure 17-3. Example of Frequency Variation Across Temperature, Trimmed at Nominal 5 Volts, 25°C, and N = 1

	-40	25	85	125
4.5	0.031595514	-0.001573799	-0.045660099	-0.073709584
5	0.045460884	0	-0.052393569	-0.076379066
5.5	0.042572265	-0.009064287	-0.047532721	-0.077255613
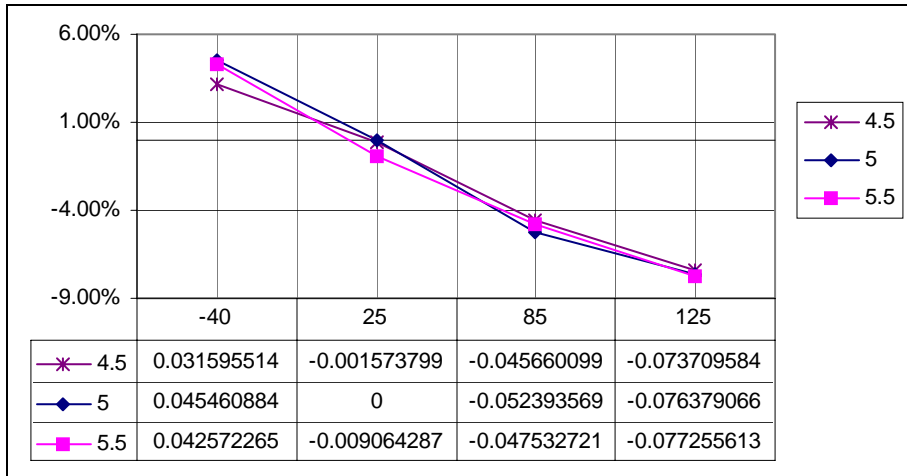
Figure 17-4. Example of Frequency Variation Across Temperature, Trimmed at Nominal 5 Volts, 25°C, and N = 104

17.10 Analog-to-Digital Converter (ADC) Characteristics

Characteristic	Symbol	Min	Max	Unit	Notes
Supply voltage	V_{DD}	2.7	5.5	V	
Input voltages	V_{ADIN}	0	V_{DD}	V	
Resolution	B_{AD}	8	8	Bits	
Absolute accuracy[1], [2]	A_{AD}	−2.5	+2.5	Counts	8 bits = 256 counts
ADC clock rate	f_{ADIC}	500 k	1.048 M	Hz	$t_{AIC} = 1/f_{ADIC}$, Tested only at 1 MHz
Conversion range	R_{AD}	V_{SS}	V_{DD}	V	
Power-up time	t_{ADPU}	16	—	t_{AIC} cycles	
Conversion time	t_{ADC}	16	17	t_{AIC} cycles	
Sample time	t_{ADS}	5	—	t_{AIC} cycles	
Monotocity	M_{AD}	Guaranteed			
Zero input reading	Z_{ADI}	00	—	Hex	$V_{In} = V_{SS}$
Full-scale reading	F_{ADI}	—	FF	Hex	$V_{In} = V_{DD}$
Input capacitance	C_{ADI}	—	20	pF	Not tested

1. One count is 1/256 of V_{DD}.
2. V_{REFH} is shared with V_{DD}. V_{REFL} is shared with V_{SS}.

17.11 Memory Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
RAM data retention voltage	V_{RDR}	1.3	—	—	V
FLASH program bus clock frequency	—	1	—	—	MHz
FLASH read bus clock frequency	f_{Read}[1]	0	—	8 M	Hz
FLASH page erase time <1 K cycles >1 K cycles	t_{Erase}	0.9 3.6	1 4	1.1 5.5	ms
FLASH mass erase time	t_{MErase}	4	—	—	ms
FLASH PGM/ERASE to HVEN setup time	t_{NVS}	10	—	—	μs
FLASH high-voltage hold time	t_{NVH}	5	—	—	μs
FLASH high-voltage hold time (mass erase)	t_{NVHL}	100	—	—	μs
FLASH program hold time	t_{PGS}	5	—	—	μs
FLASH program time	t_{PROG}	30	—	40	μs
FLASH return to read time	t_{RCV}[2]	1	—	—	μs
FLASH cumulative program HV period	t_{HV}[3]	—	—	4	ms
FLASH endurance[4]	—	10 k	100 k	—	Cycles
FLASH data retention time[5]	—	15	100	—	Years

1. f_{Read} is defined as the frequency range for which the FLASH memory can be read.
2. t_{RCV} is defined as the time it needs before the FLASH can be read after turning off the high voltage charge pump, by clearing HVEN to 0.
3. t_{HV} is defined as the cumulative high voltage programming time to the same row before next erase.
   t_{HV} must satisfy this condition: $t_{NVS} + t_{NVH} + t_{PGS} + (t_{PROG} \times 64) \leq t_{HV}$ maximum.
4. Typical endurance was evaluated for this product family. For additional information on how Freescale defines *Typical Endurance*, please refer to Engineering Bulletin EB619.
5. Typical data retention values are based on intrinsic capability of the technology measured at high temperature and de-rated to 25°C using the Arrhenius equation. For additional information on how Freescale defines *Typical Data Retention*, please refer to Engineering Bulletin EB618.

Chapter 18
Ordering Information and Mechanical Specifications

18.1 Introduction

This section contains ordering numbers for MC68HC908KX8 and MC68HC908KX2. Refer to Figure 18-1 for an example of the device numbering system.

In addition, this section gives the package dimensions for:

- 16-pin plastic dual in-line package (case number 648D)
- 16-pin small outline package (case number 751G)

18.2 MC Order Numbers

Table 18-1. MC Order Numbers

MC Order Number[1]	Operating Temperature Range
MC68HC908KX8CP MC68HC908KX8CDW	−40°C to +85°C
MC68HC908KX8VP MC68HC908KX8VDW	−40°C to +105°C
MC68HC908KX8MP MC68HC908KX8MDW	−40°C to +125°C
MC68HC908KX2CP MC68HC908KX2CDW	−40°C to +85°C
MC68HC908KX2VP MC68HC908KX2VDW	−40°C to +105°C
MC68HC908KX2MP MC68HC908KX2MDW	−40°C to +125°C

1. P = Plastic dual in-line package
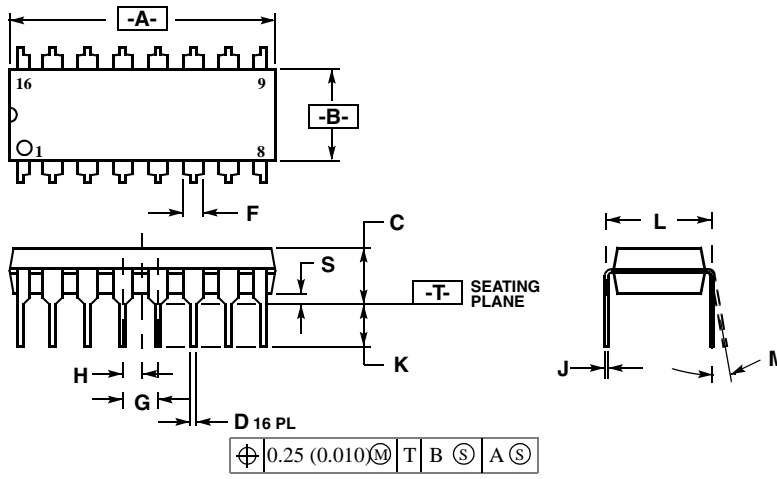   DW = Small outline package

M C 6 8 H C 9 0 8 K X 8 X X X

FAMILY ← → PACKAGE DESIGNATOR
→ TEMPERATURE RANGE
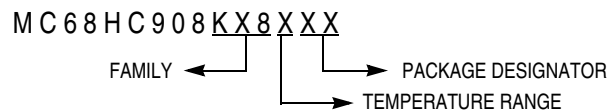
Figure 18-1. Device Numbering System

# 18.3 16-Pin Plastic Dual In-Line Package (PDIP)

NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
4. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION.
5. MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED 0.25 (0.010).
6. ROUNDED CORNERS OPTIONAL.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.740	0.760	18.80	19.30
B	0.245	0.260	6.23	6.60
C	0.145	0.175	3.69	4.44
D	0.015	0.021	0.39	0.53
F	0.050	0.070	1.27	1.77
G	0.100 BSC		2.54 BSC	
H	0.050 BSC		1.27 BSC	
J	0.008	0.015	0.21	0.38
K	0.120	0.140	3.05	3.55
L	0.295	0.305	7.50	7.74
M	0 °	10 °	0 °	10 °
S	0.015	0.035	0.39	0.88

18.4 16-Pin Small Outline Package (SOIC)
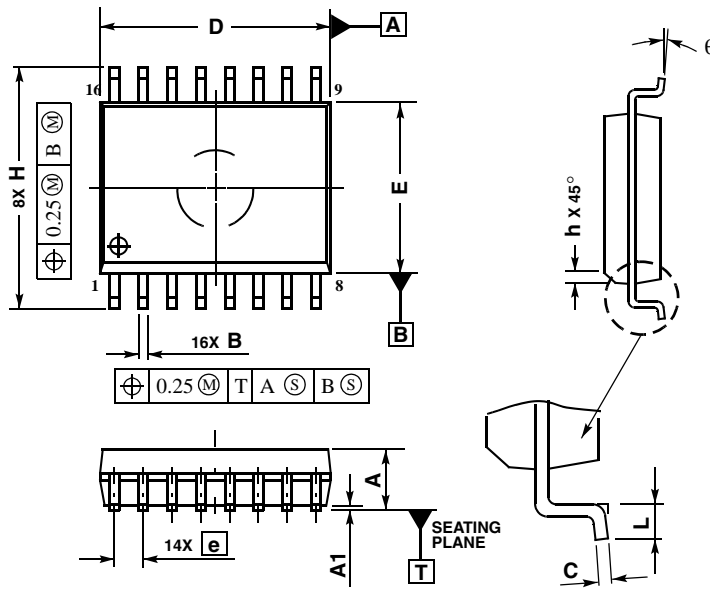
NOTES:
1. DIMENSIONS ARE IN MILLIMETERS.
2. INTERPRET DIMENSIONS AND TOLERANCES PER ASME Y14.5M, 1994.
3. DIMENSIONS D AND E DO NOT INLCUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 PER SIDE.
5. DIMENSION B DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 TOTAL IN EXCESS OF THE B DIMENSION AT MAXIMUM MATERIAL CONDITION.

DIM	MILLIMETERS	
	MIN	MAX
A	2.35	2.65
A1	0.10	0.25
B	0.35	0.49
C	0.23	0.32
D	10.15	10.45
E	7.40	7.60
e	1.27 BSC	
H	10.05	10.55
h	0.25	0.75
L	0.50	0.90
θ	0 °	7 °

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Appendix A
MC68HC908KX2

A.1 Introduction

This appendix describes the differences between the MC68HC908KX8 and the MC68HC908KX2.

A.2 Functional Description

The MC68HC908KX2 FLASH memory is an array of 2,048 bytes with an additional 36 bytes of user vectors and one byte used for block protection. See Figure A-1.

NOTE
An erased bit reads as a 1 and a programmed bit reads as a 0.

The program and erase operations are facilitated through control bits in the FLASH control register (FLCR). See 2.6 FLASH Control Register.

The FLASH is organized internally as an 8-word by 8-bit complementary metal-oxide semiconductor (CMOS) page erase, byte (8-bit) program embedded FLASH memory. Each page consists of 64 bytes. The page erase operation erases all words within a page. A page is composed of two adjacent rows.

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

See 2.6 FLASH Control Register for a complete description of FLASH operation.

---

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.
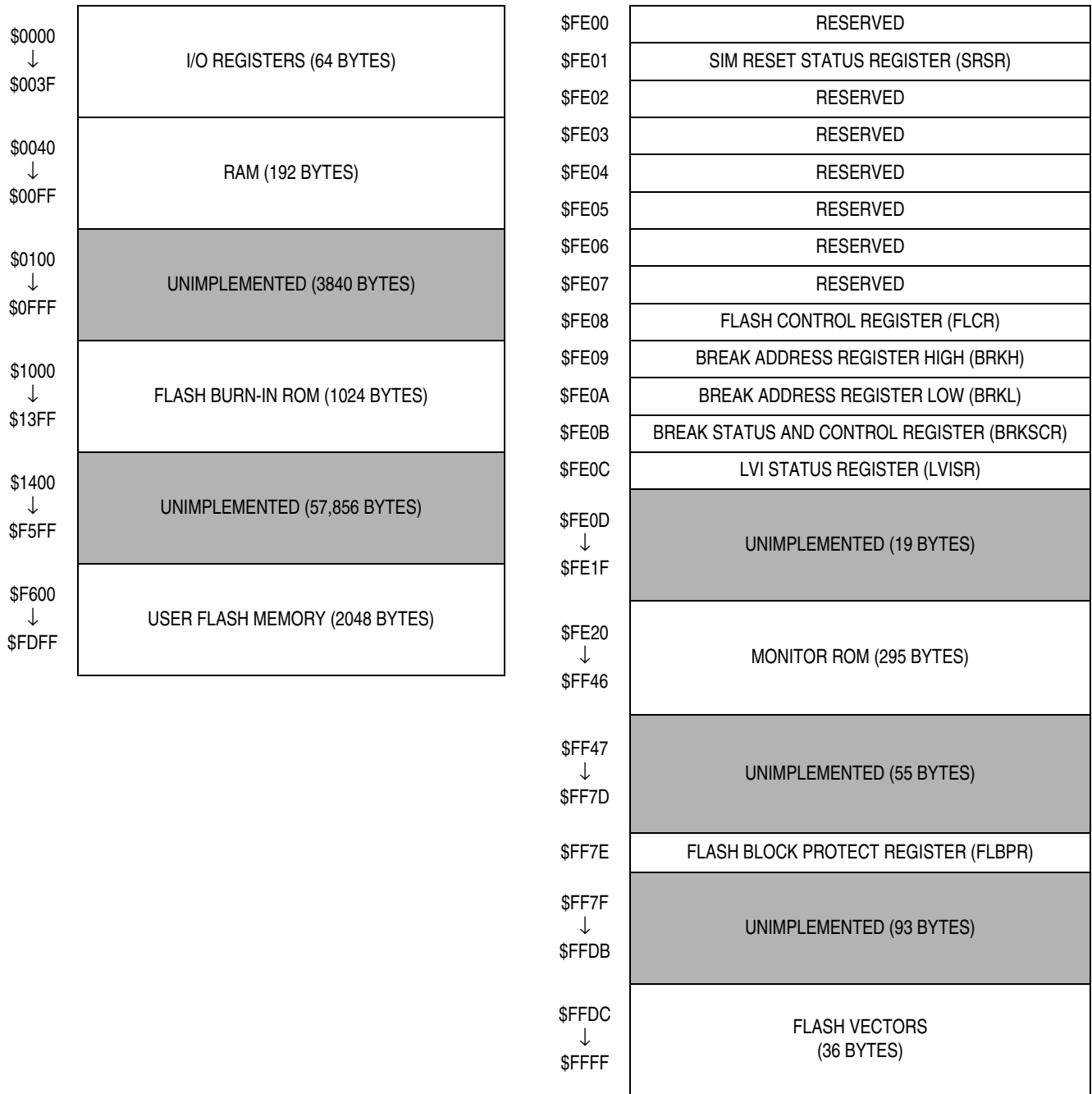
$0000 ↓ $003F	I/O REGISTERS (64 BYTES)
$0040 ↓ $00FF	RAM (192 BYTES)
$0100 ↓ $0FFF	UNIMPLEMENTED (3840 BYTES)
$1000 ↓ $13FF	FLASH BURN-IN ROM (1024 BYTES)
$1400 ↓ $F5FF	UNIMPLEMENTED (57,856 BYTES)
$F600 ↓ $FDFF	USER FLASH MEMORY (2048 BYTES)

$FE00	RESERVED
$FE01	SIM RESET STATUS REGISTER (SRSR)
$FE02	RESERVED
$FE03	RESERVED
$FE04	RESERVED
$FE05	RESERVED
$FE06	RESERVED
$FE07	RESERVED
$FE08	FLASH CONTROL REGISTER (FLCR)
$FE09	BREAK ADDRESS REGISTER HIGH (BRKH)
$FE0A	BREAK ADDRESS REGISTER LOW (BRKL)
$FE0B	BREAK STATUS AND CONTROL REGISTER (BRKSCR)
$FE0C	LVI STATUS REGISTER (LVISR)
$FE0D ↓ $FE1F	UNIMPLEMENTED (19 BYTES)
$FE20 ↓ $FF46	MONITOR ROM (295 BYTES)
$FF47 ↓ $FF7D	UNIMPLEMENTED (55 BYTES)
$FF7E	FLASH BLOCK PROTECT REGISTER (FLBPR)
$FF7F ↓ $FFDB	UNIMPLEMENTED (93 BYTES)
$FFDC ↓ $FFFF	FLASH VECTORS (36 BYTES)

Figure A-1. MC68HC908KX2 Memory Map

Appendix B
MC68HC08KX8

B.1 Introduction

This appendix describes the differences between the read-only memory (ROM) version (MC68HC08KX8) and the FLASH version (MC68HC908KX8) of the microcontroller.

Basically, the differences are:

- FLASH x ROM module changes
  - FLASH for ROM substitution
  - Partial use of FLASH-related module
- Configuration register programming
- Wider range of operating voltage

B.2 FLASH x ROM Module Changes

This subsection describes changes between the FLASH and ROM modules.

B.2.1 FLASH for ROM Substitution

FLASH memory and FLASH supporting modules are replaced by ROM memory, see Figure B-1. In Figure B-1, the user FLASH and user FLASH vector space are respectively substituted by user ROM and user ROM vector space.

Additionally, these modules and registers have been eliminated in the ROM version:

- FLASH burn-in ROM module — Auxiliary FLASH routine codes
- FLASH charge pump module — High-voltage for FLASH programming
- MENRST module — Helps erased FLASH parts programming, see 13.3.1.5 Forced Monitor Mode Entry Reset (MENRST)
- SIM reset status register, bit 2 — Refers to MENRST. See 13.7.1 SIM Reset Status Register. MENRST has no function in the ROM version and reading this bit will return 0.
- FLASH test control register, FLTCR
- FLASH control register, FLCR
- FLASH block protect register, FLBPR
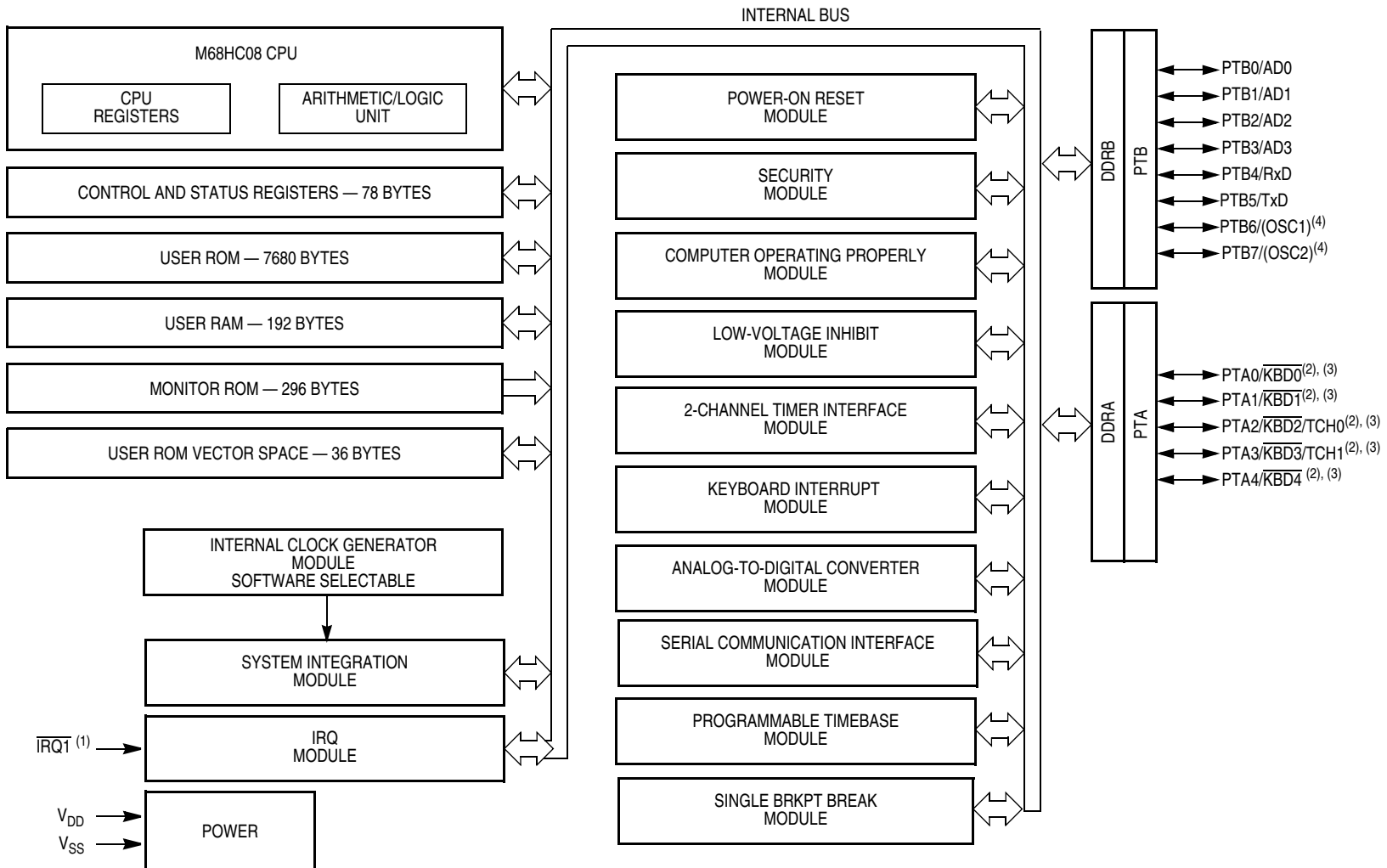
Notes:
1. Pin contains integrated pullup resistor.
2. High-current source/sink pin
3. Pin contains software selectable pullup resistor if general function I/O pin is configured as input.
4. Pins are used for external clock source or crystal/ceramic resonator option.

Figure B-1. M68HC08KX8 MCU Block Diagram

B.2.2 Partial Use of FLASH-Related Module

16.3 Monitor ROM (MON) was written having FLASH as user memory and user vector space.

MON functions are maintained for the ROM version. MON will allow execution of code in random-access memory (RAM) or ROM and provide ROM memory security[1]. The memory programming interface, though, will have no effect in ROM version.

An assumption that must be made for the ROM version is that the reset vector will always have a value different from $0000, corresponding to the user code start address. For this reason, force entry into monitor mode, described in 16.3.1.1 Monitor Mode Entry and in 16.3.1.3 Forced Monitor Mode, is not applicable to the ROM version. The MENRST module has been eliminated from the ROM version.

The security function described in 16.3.2 Security also applies to the user ROM memory for the ROM version.

B.3 Configuration Register Programming

Functionally, the terms MOR (mask option register) and CONFIG (configuration register) can be used interchangeably. MOR and CONFIG are equivalent since both define the same module functionality options through the registers bits. As a naming convention, though, configuration registers are named MOR for a ROM version and CONFIG for a FLASH version.

Some modules affected by the configuration register bits make reference to default values of these bits and have recommendation notes on programming them.

For specific information see:

- FLASH — 2.5 FLASH Memory (FLASH)
- ICG — 7.6 CONFIG (or MOR) Register Options
- LVI — 10.3 Functional Description
- PORT — 11.3.1 Port B Data Register
- COP — 5.4.7 COPD (COP Disable) and 5.4.8 COPRS (COP Rate Select)
- CONFIG — 4.2 Functional Description

NOTE
The user must keep in mind that these notes are not entirely applicable to the MOR found in the ROM version. The MOR bits can neither assume the described CONFIG default values after reset nor can they be modified later under user code control. While the MOR is mask defined, and consequently unwritable, CONFIG can be written once after each reset.

---

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

MC68HC908KX8 • MC68HC908KX2 • MC68HC08KX8 Data Sheet, Rev. 2.1

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	LVI2	EXTXTALEN	EXTSLOW	EXTCLKEN	0	OSCEINSTOP	SCIBDSRC
Write:								
Reset:				Unaffected by reset				

R	= Reserved

Figure B-2. Mask Option Register 2 (MOR2)

Address: $001F

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

Figure B-3. Mask Option Register 1 (MOR1)

NOTE
With the FLASH charge pump eliminated, MOR2 bit 2 (originally PMPREGD in CONFIG) has no effect. Reading this bit will return 0. For a complete description of other configuration bits, refer to 4.2 Functional Description.

B.4 Electrical Specifications

This subsection contains electrical and timing specifications for the MC68HC08KX8.

B.4.1 Absolute Maximum Ratings

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

> *NOTE*
> *This device is not guaranteed to operate properly at the maximum ratings. Refer to B.4.4 5.0-Vdc DC Electrical Characteristics, and   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}, V_{SS}, and PTA0–PTA4	I	±15	mA
Maximum current for pins PTA0–PTA4	I_{PTA0}−I_{PTA4}	±25	mA
Maximum current out of V_{SS}	I_{MVSS}	100	mA
Maximum current into V_{DD}	I_{MVDD}	100	mA
Storage temperature	T_{STG}	−55 to +150	°C

1. Voltages referenced to V_{SS}

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

B.4.2 Functional Operating Range

Characteristic	Symbol	Value	Unit
Operating temperature range	T_A	–40 to 105	°C
Operating voltage range	V_{DD}	$3.0 \pm 10\%$ $5.0 \pm 10\%$	V

B.4.3 Thermal Characteristics

Characteristic	Symbol	Value	Unit

Thermal resistance
  PDIP (16 pins)
  SOIC (16 pins)
	θ_{JA}	66
		95
			°C/W

I/O pin power dissipation	$P_{I/O}$	User determined	W
Power dissipation[1]	P_D	$P_D = (I_{DD} \times V_{DD}) + P_{I/O} = K/(T_J + 273°C)$	W
Constant[2]	K	$P_D \times (T_A + 273°C) + P_D^2 \times \theta_{JA}$	W/°C
Average junction temperature	T_J	$T_A + (P_D \times \theta_{JA})$	°C
Maximum junction temperature	T_{JM}	125	°C

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

B.4.4 5.0-Vdc DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit

Output high voltage
  I_{Load} = –2.0 mA, all I/O pins
  I_{Load} = –10.0 mA, all I/O pins
  I_{Load} = –15.0 mA, PTA0–PTA4 only
	V_{OH}	V_{DD} –0.4
		V_{DD} –1.5
		V_{DD} –0.8
			—
			—
			—
				—
				—
				—
					V

Output low voltage
  I_{Load} = 1.6 mA, all I/O pins
  I_{Load} = 10.0 mA, all I/O pins
  I_{Load} = 15.0 mA, PTA0–PTA4 only
	V_{OL}	—
		—
		—
			—
			—
			—
				0.4
				1.5
				0.8
					V

Input high voltage — all ports, $\overline{IRQ1}$	V_{IH}	0.7 x V_{DD}	—	V_{DD} + 0.3	V
Input low voltage — all ports, $\overline{IRQ1}$	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V

V_{DD} supply current
  Run[3], [4]
  Wait[4], [5]
  Stop, 25°C[4], [6]
	I_{DD}	—
		—
		—
			16.6
			1.9
			0.8
				20
				5
				1.75
					mA
					mA
					μA

I/O ports Hi-Z leakage current[7]	I_{IL}	–10	—	+10	μA
Input leakage current	I_{In}	–1.0	—	+1.0	μA
Capacitance Ports (as input or output)	C_{Out} C_{In}	— —	— —	12 8	pF
POR rearm voltage[8]	V_{POR}	0	—	100	mV
POR reset voltage[9]	V_{POR}	0	700	800	mV
POR rise time ramp rate	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{TST}	V_{DD}+ 2.5		V_{DD}+ 4.0	V
Low-voltage inhibit reset, trip falling voltage	V_{TRIPF}	3.90	4.3	4.50	V
Low-voltage inhibit reset, trip rising voltage	V_{TRIPR}	4.00	4.4	4.60	V
Low-voltage inhibit reset/recover hysteresis	V_{HYS}	—	100	—	mV
Pullup resistor PTA0–PTA4, $\overline{IRQ1}$	R_{PU}	24	—	48	kΩ

1. V_{DD} = 5.5 Vdc to 4.5 Vdc, V_{SS} = 0 Vdc, T_A = –40°C to +85°C, unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range, 25°C only.
3. Run (operating) I_{DD} measured using internal oscillator at its 32-MHz rate. V_{DD} = 5.5 Vdc. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. All ports configured as inputs. Measured with all modules enabled.
4. All measurements taken with LVI enabled.
5. Wait I_{DD} measured using internal oscillator at its 1-MHz rate. All inputs 0.2 V from rail; no dc loads; less than 100 pF on all outputs. All ports configured as inputs.
6. Stop I_{DD} is measured with no port pin sourcing current; all modules are disabled. OSCSTOPEN option is not selected.
7. Pullups and pulldowns are disabled.
8. Maximum is highest voltage that POR is guaranteed.
9. Maximum is highest voltage that POR is possible.

B.4.5 3.0-Vdc DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage I_{Load} = –0.6 mA, all I/O pins I_{Load} = –4.0 mA, all I/O pins I_{Load} = –10 mA, PTA0–PTA4 only	V_{OH}	V_{DD} –0.3 V_{DD} –1.0 V_{DD} –0.6	— — —	— — —	V
Output low voltage I_{Load} = 0.5 mA, all I/O pins I_{Load} = 6.0 mA, all I/O pins I_{Load} = 10 mA, PTA0–PTA4 only	V_{OL}	— — —	— — —	0.3 1.0 0.6	V V V
Input high voltage — all ports, $\overline{IRQ1}$	V_{IH}	0.7 x V_{DD}	—	V_{DD} + 0.3	V
Input low voltage — all ports, $\overline{IRQ1}$	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
V_{DD} supply current Run[3], [4] Wait[4], [5] Stop, 25°C[4], [6]	I_{DD}	— — —	4.4 1 0.65	10 2.5 1.25	mA mA μA
I/O ports Hi-Z leakage current[7]	I_{IL}	–10	—	+10	μA
Input leakage current	I_{In}	–1.0	—	+1.0	μA
Capacitance Ports (as input or output)	C_{Out} C_{In}	— —	— —	12 8	pF
POR rearm voltage[8]	V_{POR}	0	—	100	mV
POR reset voltage[9]	V_{POR}	0	700	800	mV
POR rise time ramp rate	R_{POR}	0.02	—	—	V/ms
Monitor mode entry voltage	V_{TST}	V_{DD}+ 2.5	—	V_{DD}+ 4.0	V
Low-voltage inhibit reset, trip falling voltage	V_{TRIPF}	2.4	2.60	2.70	V
Low-voltage inhibit reset, trip rising voltage	V_{TRIPR}	2.5	2.68	2.80	V
Low-voltage inhibit reset/recover hysteresis	V_{HYS}	—	80	—	mV
Pullup resistor PTA0–PTA4, $\overline{IRQ1}$	R_{PU}	24	—	48	kΩ

1. V_{DD} = 3.3 to 2.7 Vdc, V_{SS} = 0 Vdc, T_A = –40°C to +85°C, unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range, 25°C only.
3. Run (operating) I_{DD} measured using internal oscillator at its 16-MHz rate. V_{DD} = 3.3 Vdc. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. All ports configured as inputs. Measured with all modules enabled.
4. All measurements taken with LVI enabled.
5. Wait I_{DD} measured using internal oscillator at its 1 MHz rate. All inputs 0.2 V from rail; no dc loads; less than 100 pF on all outputs. All ports configured as inputs.
6. Stop I_{DD} is measured with no port pins sourcing current; all modules are disabled.
7. Pullups and pulldowns are disabled.
8. Maximum is highest voltage that POR is guaranteed.
9. Maximum is highest voltage that POR is possible.

B.4.6 Internal Oscillator Characteristics

Characteristic[1]	Symbol	Min	Typ	Max	Unit
Internal oscillator base frequency[2], [3]	f_{INTOSC}	230.4	320	384	kHz
Internal oscillator tolerance	f_{OSC_TOL}	−25	—	+25	%
Internal oscillator multiplier[4]	N	1	—	127	—

1. V_{DD} = 5.5 Vdc to 2.7 Vdc, V_{SS} = 0 Vdc, T_A = −40°C to +85°C, unless otherwise noted
2. Internal oscillator is selectable through software for a maximum frequency. Actual frequency will be multiplier (N) x base frequency.
3. f_{Bus} = (f_{INTOSC} / 4) x N when internal clock source selected
4. Multiplier must be chosen to limit the maximum bus frequency of 4 MHz for 2.7-V operation and 8 MHz for 4.5-V operation.

B.4.7 External Oscillator Characteristics

Characteristic[1]	Symbol	Min	Typ	Max	Unit

External clock option[2], [3]
  With ICG clock disabled
  With ICG clock enabled
  EXTSLOW = 1[4]
  EXTSLOW = 0[4]
	f_{EXTOSC}	dc[5]
		
		60
		307.2 k
			—
			
			—
			—
				32 M[6]
				
				307.2 k
				32 M[6]
					Hz

External crystal options[7], [8]
  EXTSLOW = 1[4]
  EXTSLOW = 0[4]
	f_{EXTOSC}	30 k
		1 M
			—
			—
				100 k
				8 M
					Hz

Crystal load capacitance[9]	C_L	—	—	—	pF
Crystal fixed capacitance[9]	C_1	—	2 x C_L	—	pF
Crystal tuning capacitance[9]	C_2	—	2 x C_L	—	pF
Feedback bias resistor[9]	R_B	—	10	—	MΩ
Series resistor [9], [10]	R_S	—	—	—	MΩ

1. V_{DD} = 5.5 to 2.7 Vdc, V_{SS} = 0 Vdc, T_A = −40°C to +85°C, unless otherwise noted
2. Setting EXTCLKEN configuration option enables OSC1 pin for external clock square-wave input.
3. No more than 10% duty cycle deviation from 50%
4. EXTSLOW configuration option configures external oscillator for a slow speed crystal and sets the clock monitor circuits of the ICG module to expect an external clock frequency that is higher/lower than the internal oscillator base frequency, f_{INTOSC}.
5. Some modules may require a minimum frequency greater than dc for proper operation. See appropriate table for this information.
6. MCU speed derates from 32 MHz at V_{DD} = 4.5 Vdc to 16 MHz at V_{DD} = 2.7 Vdc.
7. Setting EXTCLKEN and EXTXTALEN configuration options enables OSC1 and OSC2 pins for external crystal option.
8. f_{Bus} = (f_{EXTOSC} / 4) when external clock source is selected.
9. Consult crystal vendor data sheet, see Figure 7-3. External Clock Generator Block Diagram.
10. Not required for high-frequency crystals

B.4.8 Trimmed Accuracy of the Internal Clock Generator

The unadjusted frequency of the low-frequency base clock (IBASE), when the comparators in the frequency comparator indicate zero error, can vary as much as ±25% due to process, temperature, and voltage. The trimming capability exists to compensate for process affects. The remaining variation in frequency is due to temperature, voltage, and change in target frequency (multiply register setting). These affects are designed to be minimal, however variation does occur. Better performance is seen at 3 V and lower settings of N.

B.4.8.1 2.7-Volt to 3.3-Volt Trimmed Internal Clock Generator Characteristics

Characteristic[1]	Symbol	Min	Typ	Max	Unit
Absolute trimmed internal oscillator tolerance[2], [3] −40°C to 85°C	F_{abs_tol}	—	1.5	5.0	%
Variation over temperature[3], [4]	V_{ar_temp}	—	0.03	0.05	%/C
Variation over voltage[3], [5] 25°C −40°C to 85°C	V_{ar_volt}	— —	0.5 0.7	2.0 2.0	%/V

1. These specifications concern long-term frequency variation. Each measurement is taken over a 1-ms period.
2. Absolute value of variation in ICG output frequency, trimmed at nominal V_{DD} and temperature, as temperature and V_{DD} are allowed to vary for a single given setting of N.
3. Specification is characterized but not tested.
4. Variation in ICG output frequency for a fixed N and voltage
5. Variation in ICG output frequency for a fixed N

B.4.8.2 4.5-Volt to 5.5-Volt Trimmed Internal Clock Generator Characteristics

Characteristic[1]	Symbol	Min	Typ	Max	Unit
Absolute trimmed internal oscillator tolerance[2], [3] −40°C to 85°C	F_{abs_tol}	—	4.0	7.0	%
Variation over temperature[3], [4]	V_{ar_temp}	—	0.05	0.08	%/C
Variation over voltage[3], [5] 25°C −40°C to 85°C	V_{ar_volt}	— —	1.0 1.0	2.0 2.0	%/V

1. These specifications concern long-term frequency variation. Each measurement is taken over a 1-ms period.
2. Absolute value of variation in ICG output frequency, trimmed at nominal V_{DD} and temperature, as temperature and V_{DD} are allowed to vary for a single given setting of N.
3. Specification is characterized but not tested.
4. Variation in ICG output frequency for a fixed N and voltage
5. Variation in ICG output frequency for a fixed N

B.4.9 Analog-to-Digital Converter (ADC) Characteristics

Characteristic	Symbol	Min	Max	Unit	Notes
Supply voltage	V_{DD}	2.7	5.5	V	
Input voltages	V_{ADIN}	0	V_{DD}	V	
Resolution	B_{AD}	8	8	Bits	
Absolute accuracy[1], [2]	A_{AD}	−2.5	+2.5	Counts	8 bits = 256 counts
ADC clock rate	f_{ADIC}	500 k	1.048 M	Hz	$t_{AIC} = 1/f_{ADIC}$, Tested only at 1 MHz
Conversion range	R_{AD}	V_{SS}	V_{DD}	V	
Power-up time	t_{ADPU}	16	—	t_{AIC} cycles	
Conversion time	t_{ADC}	16	17	t_{AIC} cycles	
Sample time	t_{ADS}	5	—	t_{AIC} cycles	
Monotocity	M_{AD}	Guaranteed			
Zero input reading	Z_{ADI}	00	—	Hex	$V_{In} = V_{SS}$
Full-scale reading	F_{ADI}	—	FF	Hex	$V_{In} = V_{DD}$
Input capacitance	C_{ADI}	—	20	pF	Not tested

1. One count is 1/256 of V_{DD}.
2. V_{REFH} is shared with V_{DD}. V_{REFL} is shared with V_{SS}.

B.4.10 Memory Characteristics

Characteristic	Symbol	Min	Max	Units
RAM data retention voltage[1]	V_{RDR}	1.3	—	V

1. Specification is characterized but not tested.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

MC68HC908KX8
Rev. 2.1, 07/2005

freescale™
semiconductor