Data Sheet Summary

MC68HC908JL8SM/D
3/2003

MC68HC908JL8
MC68HC908JK8

Introduction

This document provides an overview of the MC68HC908JL8 and MC68HC908JK8 devices. For complete details refer to the *MC68HC908JL8 Data Sheet* (Motorola document order number MC68HC908JL8/D).

General Description

The MC68HC908JL8 is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). The M68HC08 Family is a Complex Instruction Set Computer (CISC) with a Von Neumann architecture. 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.

Table 1. MC Order Numbers

MC Order Number	Operating Temperature Range	Package
MC68HC908JK8CP	–40 °C to +85 °C	20-pin PDIP
MC68HC908JK8MP	–40 °C to +125 °C	
MC68HC908JK8CDW	–40 °C to +85 °C	20-pin SOIC
MC68HC908JK8MDW	–40 °C to +125 °C	
MC68HC908JL8CP	–40 °C to +85 °C	28-pin PDIP
MC68HC908JL8MP	–40 °C to +125 °C	
MC68HC908JL8CDW	–40 °C to +85 °C	28-pin SOIC
MC68HC908JL8MDW	–40 °C to +125 °C	
MC68HC908JL8CSP	–40 °C to +85 °C	32-pin SDIP
MC68HC908JL8MSP	–40 °C to +125 °C	
MC68HC908JL8CFA	–40 °C to +85 °C	32-pin LQFP
MC68HC908JL8MFA	–40 °C to +125 °C	

Features

Features include:

- High-performance M68HC08 architecture
- Fully upward-compatible object code with M6805, M146805, and M68HC05 Families
- Low-power design; fully static with stop and wait modes
- Maximum internal bus frequency:
  - 8-MHz at 5V operating voltage
  - 4-MHz at 3V operating voltage
- Oscillator options:
  - Crystal or resonator
  - RC oscillator
- 8,192 bytes user program FLASH memory with security[1] feature
- 256 bytes of on-chip RAM
- Two 16-bit, 2-channel timer interface modules (TIM1 and TIM2) with selectable input capture, output compare, and PWM capability on each channel; external clock input option on TIM2
- 13-channel, 8-bit analog-to-digital converter (ADC)
- Serial communications interface module (SCI)
- 26 general-purpose input/output (I/O) ports:
  - 8 keyboard interrupt with internal pull-up
  - 11 LED drivers (sink)
  - 2 × 25mA open-drain I/O with pull-up
- System protection features:
  - Optional computer operating properly (COP) reset, driven by internal 64-kHz RC oscillator
  - Optional low-voltage detection with reset and selectable trip points for 3V and 5V operation
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- Master reset pin with internal pull-up and power-on reset
- $\overline{\text{IRQ}}$ with schmitt-trigger input and programmable pull-up

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

- MC68HC908JL8 is available in these packages:
  - 20-pin PDIP (MC68HC908JK8)
  - 20-pin SOIC (MC68HC908JK8)
  - 28-pin PDIP
  - 28-pin SOIC
  - 32-pin SDIP
  - 32-pin LQFP
- Specific features of the MC68HC908JL8 in 28-pin packages are:
  - 23 general-purpose I/Os only
  - 7 keyboard interrupt with internal pull-up
  - 10 LED drivers (sink)
  - 12-channel ADC
  - Timer I/O pins on TIM1 only
- Specific features of the MC68HC908JK8 are: (MC68HC908JL8 in 20-pin packages)
  - 15 general-purpose I/Os only
  - 1 keyboard interrupt with internal pull-up
  - 4 LED drivers (sink)
  - 10-channel ADC
  - Timer I/O pins on TIM1 only

MCU Block Diagram

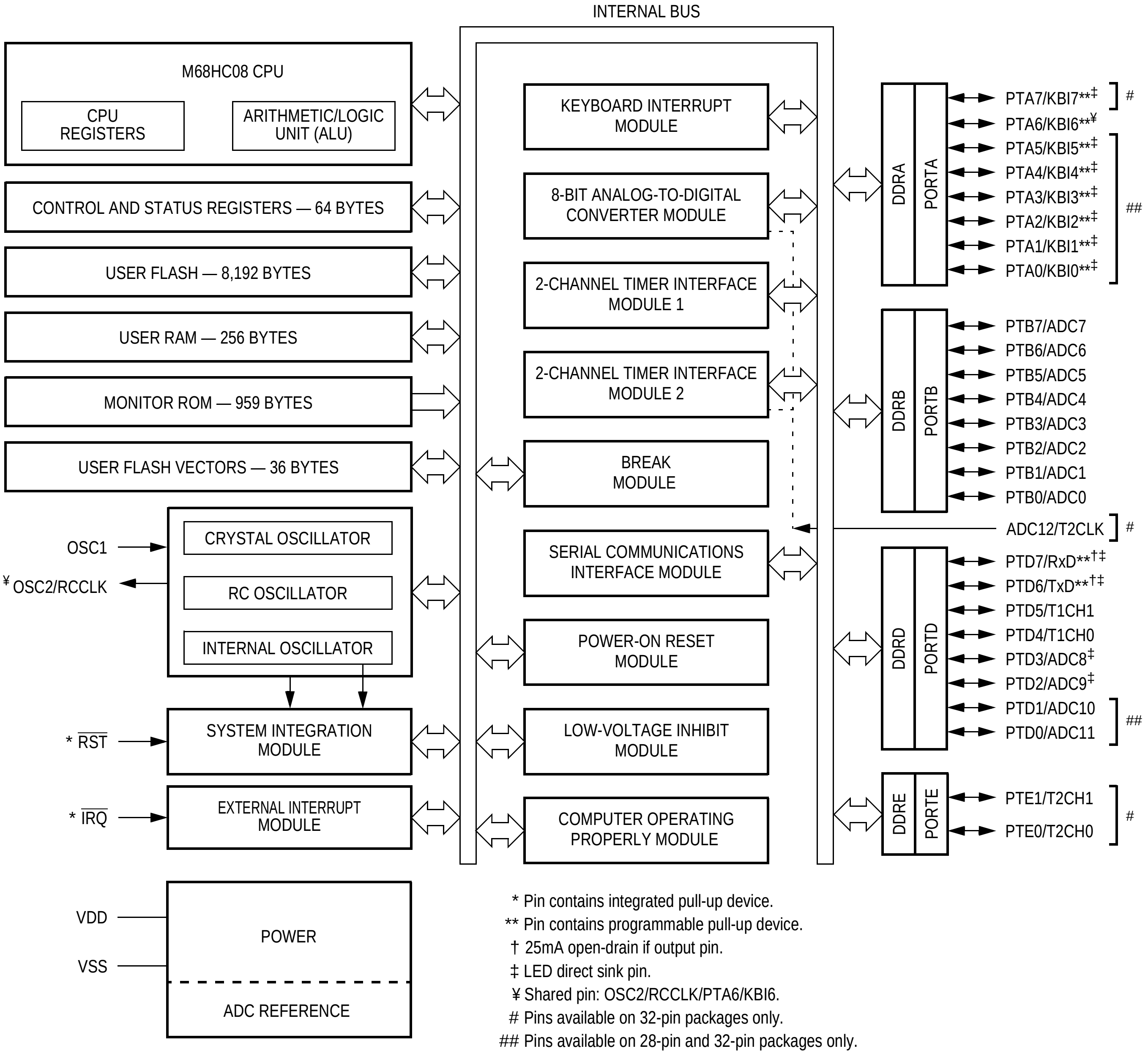
See **Figure 1**.

Memory

The central processor unit (CPU08) can address 64 Kbytes of memory space. The memory map is shown in **Figure 3**.

Addresses $0000–$003F, shown in **Figure 4**, contain most of the control, status, and data registers.

The vector addresses are shown in **Table 3**.



* Pin contains integrated pull-up device.
** Pin contains programmable pull-up device.
† 25mA open-drain if output pin.
‡ LED direct sink pin.
¥ Shared pin: OSC2/RCCLK/PTA6/KBI6.
# Pins available on 32-pin packages only.
## Pins available on 28-pin and 32-pin packages only.

Figure 1. Block Diagram

Pin Assignments

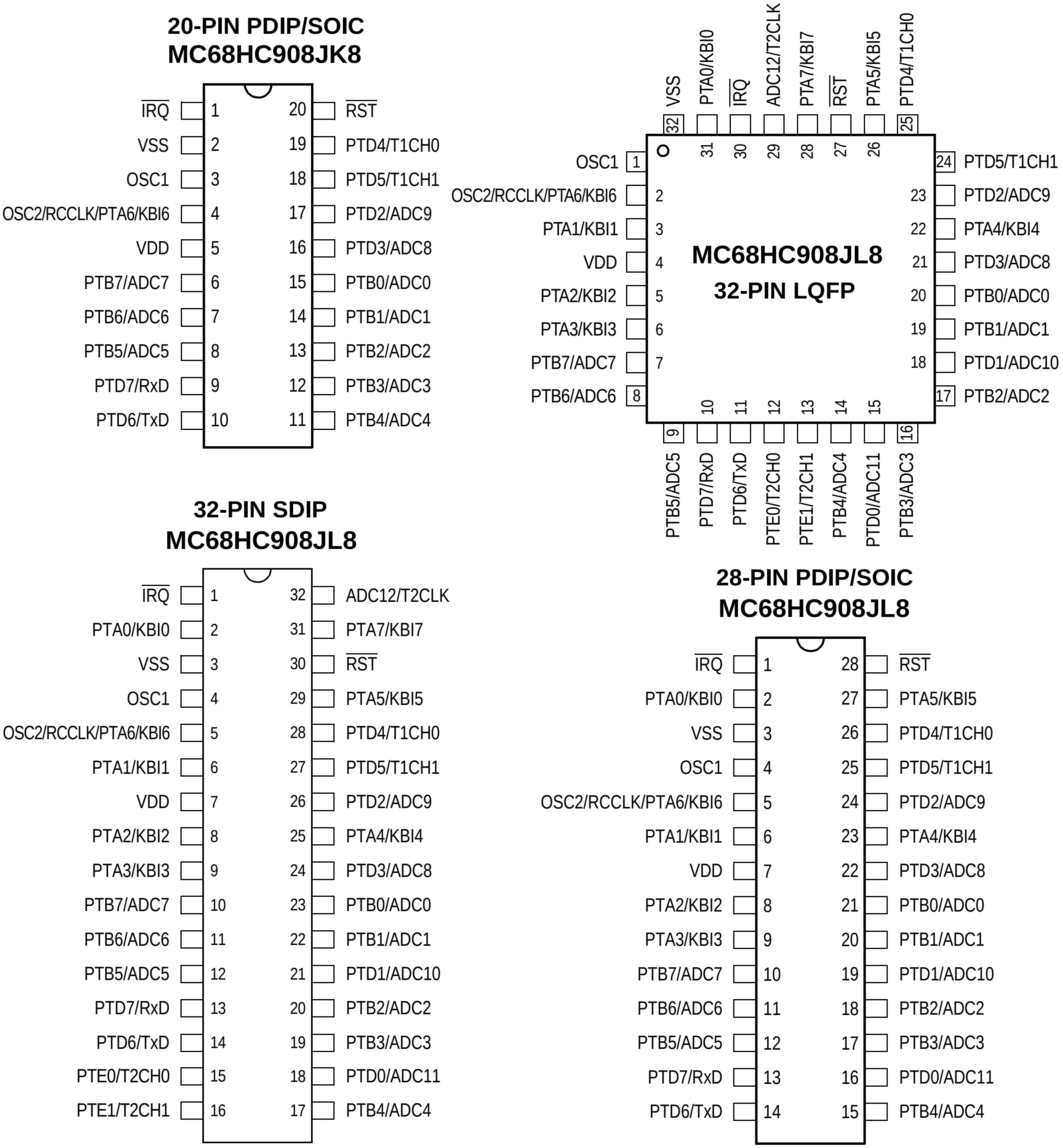


Figure 2. MCU Pin Assignments

Pin Functions

Table 2 provides a description of the pin functions.

Table 2. Pin Functions

PIN NAME	PIN DESCRIPTION	IN/OUT	VOLTAGE LEVEL
VDD	Power supply.	In	5V or 3V
VSS	Power supply ground.	Out	0V
$\overline{\text{RST}}$	Reset input, active low; with internal pull-up and schmitt trigger input.	In/Out	VDD
$\overline{\text{IRQ}}$	External IRQ pin; with programmable internal pull-up and schmitt trigger input.	In	VDD
	Used for monitor mode entry.	In	VDD to V_{TST}
OSC1	Crystal or RC oscillator input.	In	VDD
OSC2/RCCLK	OSC2: crystal oscillator output; inverted OSC1 signal.	Out	VDD
	RCCLK: RC oscillator clock output.	Out	VDD
	Pin as PTA6/KBI6 (see PTA0–PTA7).	In/Out	VDD
ADC12/T2CLK	ADC12: channel-12 input of ADC.	In	VSS to VDD
	T2CLK: external input clock for TIM2.	In	VDD
PTA0–PTA7	8-bit general purpose I/O port.	In/Out	VDD
	Each pin has programmable internal pull-up when configured as input.	In	VDD
	Pins as keyboard interrupts, KBI0–KBI7.	In	VDD
	PTA0–PTA5 and PTA7 have LED direct sink capability.	Out	VSS
	PTA6 as OSC2/RCCLK.	Out	VDD
PTB0–PTB7	8-bit general purpose I/O port.	In/Out	VDD
	Pins as ADC input channels, ADC0–ADC7.	In	VSS to VDD
PTD0–PTD7	8-bit general purpose I/O port; with programmable internal pull-ups on PTD6–PTD7.	In/Out	VDD
	PTD0–PTD3 as ADC input channels, ADC11–ADC8.	Input	VSS to VDD
	PTD2–PTD3 and PTD6–PTD7 have LED direct sink capability.	Out	VSS
	PTD4 as T1CH0 of TIM1.	In/Out	VDD
	PTD5 as T1CH1 of TIM1.	In/Out	VDD
	PTD6–PTD7 have configurable 25mA open-drain output.	Out	VSS
	PTD6 as TxD of SCI.	Out	VDD
	PTD7 as RxD of SCI.	In	VDD
PTE0–PTE1	2-bit general purpose I/O port.	In/Out	VDD
	PTE0 as T2CH0 of TIM2.	In/Out	VDD
	PTE1 as T2CH1 of TIM2.	In/Out	VDD

$0000 ↓ $003F	I/O REGISTERS 64 BYTES
$0040 ↓ $005F	RESERVED 32 BYTES
$0060 ↓ $015F	RAM 256 BYTES
$0160 ↓ $DBFF	UNIMPLEMENTED 55,968 BYTES
$DC00 ↓ $FBFF	FLASH MEMORY 8,192 BYTES
$FC00 ↓ $FDFF	MONITOR ROM 512 BYTES
$FE00	BREAK STATUS REGISTER (BSR)
$FE01	RESET STATUS REGISTER (RSR)
$FE02	RESERVED
$FE03	BREAK FLAG CONTROL REGISTER (BFCR)
$FE04	INTERRUPT STATUS REGISTER 1 (INT1)
$FE05	INTERRUPT STATUS REGISTER 2 (INT2)
$FE06	INTERRUPT STATUS REGISTER 3 (INT3)
$FE07	RESERVED
$FE08	FLASH CONTROL REGISTER (FLCR)
$FE09 ↓ $FF0B	RESERVED
$FE0C	BREAK ADDRESS HIGH REGISTER (BRKH)
$FE0D	BREAK ADDRESS LOW REGISTER (BRKL)
$FE0E	BREAK STATUS AND CONTROL REGISTER (BRKSCR)
$FE0F	RESERVED
$FE10 ↓ $FFCE	MONITOR ROM 447 BYTES
$FFCF	FLASH BLOCK PROTECT REGISTER (FLBPR)
$FFD0	MASK OPTION REGISTER (MOR)
$FFD1 ↓ $FFDB	RESERVED 11 BYTES
$FFDC ↓ $FFFF	USER FLASH VECTORS 36 BYTES

Figure 3. Memory Map

Addr.	Register	Bit 7	6	5	4	3	2	1	Bit 0
$0000	PTA	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
$0001	PTB	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
$0002	Unimplemented								
$0003	PTD	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
$0004	DDRA	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
$0005	DDRB	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
$0006	Unimplemented								
$0007	DDRD	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
$0008	PTE							PTE1	PTE0
$0009	Unimplemented								
$000A	PDCR					SLOWD7	SLOWD6	PTDPU7	PTDPU6
$000B	Unimplemented								
$000C	DDRE							DDRE1	DDRE0
$000D	PTAPUE	PTA6EN	PTAPUE6	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
$000E	PTA7PUE	PTAPUE7							
$000F– $0012	Unimplemented								
	Unimplemented								
$0013	SCC1	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
$0014	SCC2	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
$0015	SCC3	R8	T8	DMARE	DMATE	ORIE	NEIE	FEIE	PEIE
$0016	SCS1	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
$0017	SCS2							BKF	RPF
$0018	SCDR	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
$0019	SCBR			SCP1	SCP0		SCR2	SCR1	SCR0
$001A	KBSCR					KEYF	ACKK	IMASKK	MODEK
$001B	KBIER	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
$001C	Unimplemented								
$001D	INTSCR					IRQF	ACK	IMASK	MODE
$001E	CONFIG2	IRQPUD			LVIT1	LVIT0			STOP_ICLKDIS
$001F	CONFIG1	COPRS			LVID	R	SSREC	STOP	COPD
$0020	T1SC	TOF	TOIE	TSTOP	TRST	0	PS2	PS1	PS0
$0021	T1CNTH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
$0022	T1CNTL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0023	T1MODH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
$0024	T1MODL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0025	T1SC0	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
$0026	T1CH0H	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
$0027	T1CH0L	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0028	T1SC1	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
$0029	T1CH1H	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
$002A	T1CH1L	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

▨ = Unimplemented or Reserved

Figure 4. Control, Status, and Data Registers (Sheet 1 of 2)

Addr.	Register	Bit 7	6	5	4	3	2	1	Bit 0
$002B–	Unimplemented	▨	▨	▨	▨	▨	▨	▨	▨
$002F	Unimplemented	▨	▨	▨	▨	▨	▨	▨	▨
$0030	T2SC	TOF	TOIE	TSTOP	TRST	0	PS2	PS1	PS0
$0031	T2CNTH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
$0032	T2CNTL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0033	T2MODH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
$0034	T2MODL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0035	T2SC0	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
$0036	T2CH0H	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
$0037	T2CH0L	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0038	T2SC1	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
$0039	T2CH1H	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
$003A	T2CH1L	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$003B	Unimplemented	▨	▨	▨	▨	▨	▨	▨	▨
$0038	OSCTRIM	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
$003C	ADSCR	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
$003D	ADR	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
$003E	ADICLK	ADIV2	ADIV1	ADIV0	0	0	0	0	0
$003F	Unimplemented	▨	▨	▨	▨	▨	▨	▨	▨
$FE00	BSR	▨	▨	▨	▨	▨	▨	SBSW	▨
$FE01	RSR	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
$FE02	Reserved	▨	▨	▨	▨	▨	▨	▨	▨
$FE03	BFCR	BCFE	▨	▨	▨	▨	▨	▨	▨
$FE04	INT1	IF6	IF5	IF4	IF3	0	IF1	0	0
$FE05	INT2	IF14	IF13	IF12	IF11	0	0	IF8	IF7
$FE06	INT3	0	0	0	0	0	0	0	IF15
$FE07	Reserved	▨	▨	▨	▨	▨	▨	▨	▨
$FE08	FLCR	0	0	0	0	HVEN	MASS	ERASE	PGM
$FE09–	Reserved	▨	▨	▨	▨	▨	▨	▨	▨
$FE0B	Reserved	▨	▨	▨	▨	▨	▨	▨	▨
$FE0C	BRKH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
$FE0D	BRKL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$FE0E	BRKSCR	BRKE	BRKA	0	0	0	0	0	0
$FFCF	FLBPR (non-volatile register)	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
$FFD0	MOR (non-volatile register)	OSCSEL	▨	▨	▨	▨	▨	▨	▨
$FFFF	COPCTL	WRITE ANY VALUE TO RESET COP WATCHDOG							

▨ = Unimplemented or Reserved

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

Table 3. Vector Addresses

Vector Priority	Vector	Address	Vector
Lowest	IF16	—	Not used
↑	IF15	$FFDE	ADC conversion complete vector (high)
		$FFDF	ADC conversion complete vector (low)
	IF14	$FFE0	Keyboard vector (high)
		$FFE1	Keyboard vector (low)
	IF13	$FFE2	SCI transmit vector (high)
		$FFE3	SCI transmit vector (low)
	IF12	$FFE4	SCI receive vector (high)
		$FFE5	SCI receive vector (low)
	IF11	$FFE6	SCI error vector (high)
		$FFE7	SCI error vector (low)
	IF10 and IF9	—	Not used
	IF8	$FFEC	TIM2 overflow vector (high)
		$FFED	TIM2 overflow vector (low)
	IF7	$FFEE	TIM2 channel 1 vector (high)
		$FFEF	TIM2 channel 1 vector (low)
	IF6	$FFF0	TIM2 channel 0 vector (high)
		$FFF1	TIM2 channel 0 vector (low)
	IF5	$FFF2	TIM1 overflow vector (high)
		$FFF3	TIM1 overflow vector (low)
	IF4	$FFF4	TIM1 channel 1 vector (high)
		$FFF5	TIM1 channel 1 vector (low)
	IF3	$FFF6	TIM1 channel 0 vector (high)
		$FFF7	TIM1 channel 0 vector (low)
	IF2	—	Not used
	IF1	$FFFA	$\overline{\text{IRQ}}$ vector (high)
		$FFFB	$\overline{\text{IRQ}}$ vector (low)
	—	$FFFC	SWI vector (high)
↓		$FFFD	SWI vector (low)
Highest	—	$FFFE	Reset vector (high)
		$FFFF	Reset vector (low)

FLASH Module

The FLASH memory consists of an array of 8,192 bytes with an additional 36 bytes for user vectors, 2 bytes for non-volatile registers. The minimum size of FLASH memory that can be erased is 64 bytes; and the maximum size of FLASH memory that can be programmed in a program cycle is 32 bytes (a row). Program and erase operations are facilitated through control bits in the FLASH control register (FLCR). Details for these operations appear later in this section. The address ranges for the user memory and vectors are:

- $DC00 – $FBFF; user memory, 8,192 bytes
- $FFCF and $FFD0; non-volatile registers, 2 bytes
- $FFDC – $FFFF; user interrupt vectors, 36 bytes

NOTE: *An erased bit reads as logic 1 and a programmed bit reads as logic 0. A security feature prevents unauthorized viewing of the FLASH contents.*

FLASH Control Register

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

$FE08	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	HVEN	MASS	ERASE	PGM
Reset:	0	0	0	0	0	0	0	0

Figure 5. FLASH Control Register (FLCR)

HVEN — High Voltage Enable Bit
1 = High voltage enabled to array and charge pump on

MASS — Mass Erase Control Bit
1 = Mass Erase operation selected

ERASE — Erase Control Bit
1 = Erase operation selected

PGM — Program Control Bit
1 = Program operation selected

FLASH Page Erase Operation

Use the following procedure to erase a page of FLASH memory. A page consists of 64 consecutive bytes starting from addresses $XX00, $XX40, $XX80, or $XXC0. Any FLASH memory page can be erased alone. The 36-byte user interrupt vector area also forms a page, but only be erased by a mass erase operation.

1. Set the ERASE bit and clear the MASS bit in the FLASH control register.
2. Read the FLASH block protect register ($FFCF).
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 4 ms).
7. Clear the ERASE and MASS bits.
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: *Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. These operations must be performed in the order as shown, but other unrelated operations may occur between the steps.*

FLASH Program Operation

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

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

1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
2. Read from the FLASH block protect register ($FFCF).
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.
   (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.)
8. Wait for time, t_{prog} (minimum 30 µs).
9. Repeat step 6 and 7 until 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.

NOTE: *Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. These operations must be performed in the order shown, other unrelated operations may occur between the steps. Do not exceed t_{prog} maximum.*

FLASH Block Protect Register

The FLASH block protect register is implemented as a byte within the FLASH memory, and therefore it is programmed using a FLASH memory byte-programming operation. The value in this register determines the starting address of the protected range within the FLASH memory. The FLASH is protected from this address to the end of FLASH memory at $FFFF.

$FFCF	Bit 7	6	5	4	3	2	1	Bit 0
	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
Reset:	Unaffected by reset. Initial value from factory is all 1’s.							

Figure 6. FLASH Block Protect Register (FLBPR)

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

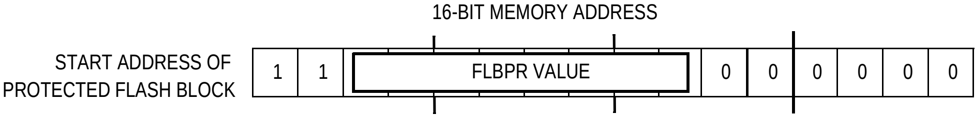


Figure 7. FLASH Block Protect Start Address

Table 4. Examples of Protect Start Address

BPR[7:0]	Start of Address of Protect Range
$00–$70	The entire FLASH memory is protected.
$71 (**0111 0001**)	$DC40 (11**01 1100 01**00 0000)
$72 (**0111 0010**)	$DC80 (11**01 1100 10**00 0000)
$73 (**0111 0011**)	$DCC0 (11**01 1100 11**00 0000)
and so on...	
$FD (**1111 1101**)	$FF40 (11**11 1111 01**00 0000)
$FE (**1111 1110**)	$FF80 (11**11 1111 10**00 0000)
$FF	The entire FLASH memory is not protected.

Mask Option Register (MOR)

The mask option register (MOR) is implemented as a byte within the FLASH memory, and therefore it is programmed using a FLASH memory byte-programming operation. This register is read after a power-on reset to determine the type of oscillator selected.

$FFD0	Bit 7	6	5	4	3	2	1	Bit 0
	OSCSEL	R	R	R	R	R	R	R
Reset:	U	U	U	U	U	U	U	U
Erased:	1	1	1	1	1	1	1	1

R = Reserved U = Unaffected

Figure 8 Mask Option Register (MOR)

OSCSEL — Oscillator Select Bit
1 = Crystal oscillator
0 = RC oscillator

Configuration Registers (CONFIG1, CONFIG2)

The configuration registers are used to initialize various options. The configuration registers can each be written once after each reset. Most of the configuration register bits are cleared during reset. Since the various options affect the operation of the microcontroller unit (MCU) it is recommended that these registers be written immediately after reset. The configuration registers are located at $001E and $001F, and may be read at anytime.

$001E	Bit 7	6	5	4	3	2	1	Bit 0
	IRQPUD	R	R	LVIT1	LVIT0	R	R	STOP_ ICLKDIS
Reset:	0	0	0	U	U	0	0	0
POR:	0	0	0	0	0	0	0	0

R = Reserved U = Unaffected

Figure 9 Configuration Register 2 (CONFIG2)

IRQPUD — $\overline{\text{IRQ}}$ Pin Pullup Disable Bit
0 = Internal pullup is connected between $\overline{\text{IRQ}}$ pin and V_{DD}

LVIT1, LVIT0 — LVI Trip Voltage Selection Bits
0:X = For 3-V operation
1:0 = For 5-V operation

STOP_ICLKDIS — Internal Oscillator Stop Mode Disable Bit
1 = Internal oscillator disabled during stop mode

$001F	Bit 7	6	5	4	3	2	1	Bit 0
	COPRS	R	R	LVID	R	SSREC	STOP	COPD
Reset:	0	0	0	0	0	0	0	0
POR:	0	0	0	0	0	0	0	0

U = Unaffected

Figure 10 Configuration Register 1 (CONFIG1)

COPRS — COP Reset Period Selection Bit
1 = COP reset short cycle = $(2^{13} - 2^{4})$ x ICLK
0 = COP reset long cycle = $(2^{18} - 2^{4})$ x ICLK
To prevent a reset due to a COP watchdog timeout, write any value to COPCTL ($FFFF) before the COP timer reaches the selected timeout.

LVID — LVI Disable Bit
1 = LVI disabled

SSREC — Short Stop Recovery Bit
1 = Stop mode recovery after 32 ICLK cycles
0 = Stop mode recovery after 4096 ICLK cycles

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

STOP — STOP Instruction Enable Bit
1 = STOP instruction enabled
0 = STOP instruction treated as illegal opcode

COPD — COP Disable Bit
1 = COP module disabled

SIM Reset Status Register

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

$FE01	Bit 7	6	5	4	3	2	1	Bit 0
	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
POR:	1	0	0	0	0	0	0	0

Figure 11. SIM Reset Status Register (SRSR)

POR — Power-On Reset Bit
  1 = Last reset caused by POR circuit

PIN — External Reset Bit
  1 = Last reset caused by external reset pin ($\overline{\text{RST}}$)

COP — Computer Operating Properly Reset Bit
  1 = Last reset caused by COP timeout

ILOP — Illegal Opcode Reset Bit
  1 = Last reset caused by an illegal opcode

ILAD — Illegal Address Reset Bit (opcode fetches only)
  1 = Last reset caused by an opcode fetch from an illegal address

MODRST — Monitor Mode Entry Module Reset Bit
  1 = Last reset caused by monitor mode entry when vector locations $FFFE and $FFFF are $FF after POR while $\overline{\text{IRQ}} = V_{DD}$

LVI — Low Voltage Inhibit Reset Bit
  1 = Last reset caused by LVI circuit

Interrupt Status Registers (INT1, INT2, INT3)

These three registers include status flags which indicate which interrupt sources currently have pending requests. See **Table 3**.

$FE04	Bit 7	6	5	4	3	2	1	Bit 0
	IF6	IF5	IF4	IF3	0	IF1	0	0
Reset:	0	0	0	0	0	0	0	0
Source:	T2TCH0	T1TOF	T1TCH1	T1TCH0		$\overline{\text{IRQ}}$		

Figure 12. Interrupt Status Register 1 (INT1)

$FE05	Bit 7	6	5	4	3	2	1	Bit 0
	IF14	IF13	IF12	IF11	0	0	IF8	IF7
Reset:	0	0	0	0	0	0	0	0
Source:	KBI	SCI Tx	SCI Rx	SCI Error			T2TOF	T2TCH1

Figure 13. Interrupt Status Register 2 (INT2)

$FE06	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	0	0	0	IF15
Reset:	0	0	0	0	0	0	0	0
Source:								ADC

Figure 14. Interrupt Status Register 3 (INT3)

IFxx — Interrupt Flags

These flags indicate the presence of interrupt requests from the sources shown below the corresponding IFxx bit.

1 = Interrupt request pending
0 = No interrupt request present

Central Processor Unit (CPU)

Figure 15 shows the five CPU registers. CPU registers are not part of the memory map.

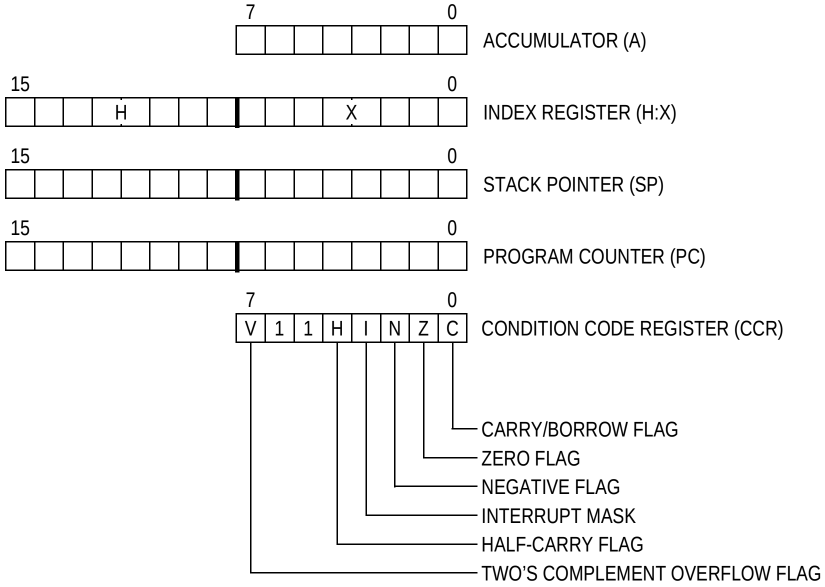


Figure 15. CPU Registers

Instruction Set Summary

Table 5 provides a summary of the M68HC08 instruction set.

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

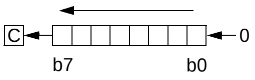
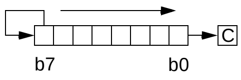
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X ADC *opr*,SP ADC *opr*,SP	Add with Carry	A ← (A) + (M) + (C)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A9 B9 C9 D9 E9 F9 9EE9 9ED9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

Table 5. Instruction Set Summary (Sheet 2 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
BGT *opr*	Branch if Greater Than (Signed Operands)	PC ← (PC) + 2 + *rel* ? (Z) | (N ⊕ V) = 0	–	–	–	–	–	–	REL	92	rr	3
BHCC *rel*	Branch if Half Carry Bit Clear	PC ← (PC) + 2 + *rel* ? (H) = 0	–	–	–	–	–	–	REL	28	rr	3
BHCS *rel*	Branch if Half Carry Bit Set	PC ← (PC) + 2 + *rel* ? (H) = 1	–	–	–	–	–	–	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? (C) | (Z) = 0	–	–	–	–	–	–	REL	22	rr	3
BHS *rel*	Branch if Higher or Same (Same as BCC)	PC ← (PC) + 2 + *rel* ? (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BIH *rel*	Branch if $\overline{\text{IRQ}}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 1	–	–	–	–	–	–	REL	2F	rr	3
BIL *rel*	Branch if $\overline{\text{IRQ}}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 0	–	–	–	–	–	–	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X BIT *opr*,SP BIT *opr*,SP	Bit Test	(A) & (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A5 B5 C5 D5 E5 F5 9EE5 9ED5	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
BLE *opr*	Branch if Less Than or Equal To (Signed Operands)	PC ← (PC) + 2 + *rel* ? (Z) | (N ⊕ V) = 1	–	–	–	–	–	–	REL	93	rr	3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? (C) = 1	–	–	–	–	–	–	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? (C) | (Z) = 1	–	–	–	–	–	–	REL	23	rr	3
BLT *opr*	Branch if Less Than (Signed Operands)	PC ← (PC) + 2 + *rel* ? (N ⊕ V) =1	–	–	–	–	–	–	REL	91	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? (I) = 0	–	–	–	–	–	–	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? (N) = 1	–	–	–	–	–	–	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? (I) = 1	–	–	–	–	–	–	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? (Z) = 0	–	–	–	–	–	–	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? (N) = 0	–	–	–	–	–	–	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel*	–	–	–	–	–	–	REL	20	rr	3
BRCLR *n*,*opr*,*rel*	Branch if Bit *n* in M Clear	PC ← (PC) + 3 + *rel* ? (Mn) = 0	–	–	–	–	–	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2	–	–	–	–	–	–	REL	21	rr	3
BRSET *n*,*opr*,*rel*	Branch if Bit *n* in M Set	PC ← (PC) + 3 + *rel* ? (Mn) = 1	–	–	–	–	–	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5

Table 5. Instruction Set Summary (Sheet 3 of 7)

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

Table 5. Instruction Set Summary (Sheet 4 of 7)

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

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

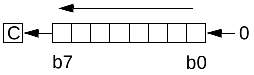
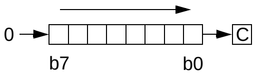
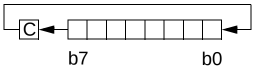
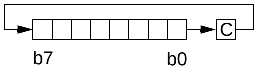
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X LSL *opr*,SP	Logical Shift Left (Same as ASL)		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
LSR *opr* LSRA LSR*X* LSR *opr*,X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

Table 5. Instruction Set Summary (Sheet 6 of 7)

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

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

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
TPA	Transfer CCR to A	A ← (CCR)	–	–	–	–	–	–	INH	85		1
TST *opr* TSTA TSTX TST *opr*,X TST ,X TST *opr*,SP	Test for Negative or Zero	(A) – $00 or (X) – $00 or (M) – $00	0	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3D 4D 5D 6D 7D 9E6D	dd ff ff	3 1 1 3 2 4
TSX	Transfer SP to H:X	H:X ← (SP) + 1	–	–	–	–	–	–	INH	95		2
TXA	Transfer X to A	A ← (X)	–	–	–	–	–	–	INH	9F		1
TXS	Transfer H:X to SP	(SP) ← (H:X) – 1	–	–	–	–	–	–	INH	94		2

A	Accumulator	*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

Oscillator Module (OSC)

The oscillator type is determined by the MOR at $FFD0, which is implemented as a byte within the FLASH memory, and therefore it is programmed using a FLASH memory byte-programming operation.

$FFD0	Bit 7	6	5	4	3	2	1	Bit 0
	OSCSEL	R	R	R	R	R	R	R
Reset:	U	U	U	U	U	U	U	U
Erased:	1	1	1	1	1	1	1	1

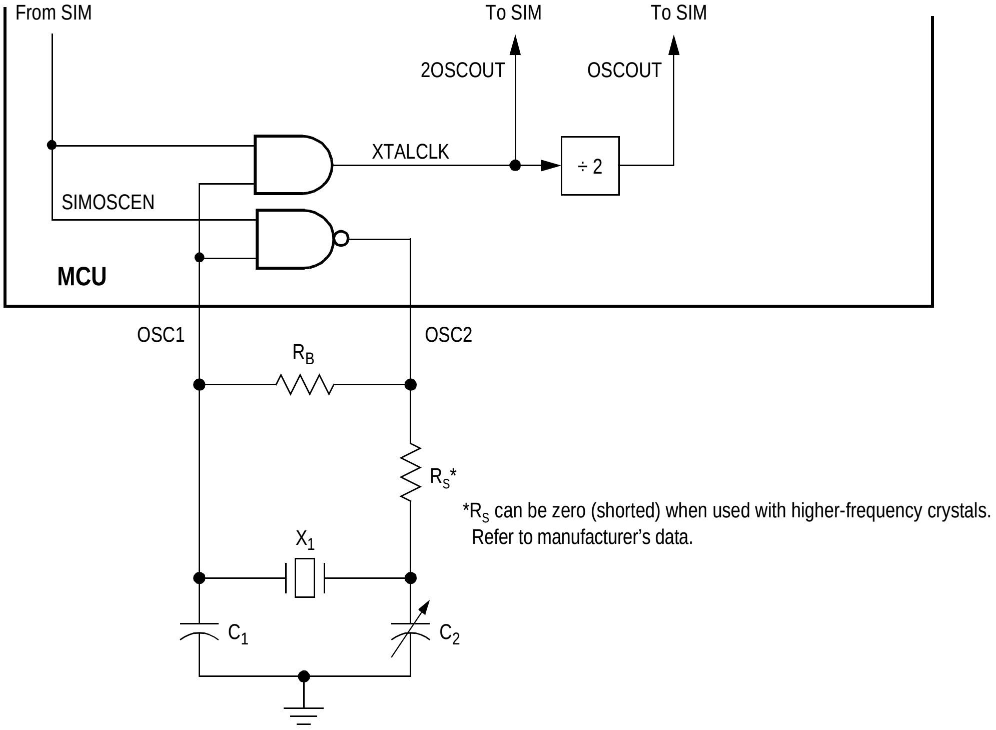
R = Reserved U = Unaffected

Figure 16 Mask Option Register (MOR)

OSCSEL — Oscillator Select Bit

1 = Crystal oscillator (see **Figure 17**)

0 = RC oscillator (see **Figure 18**)



*R_S can be zero (shorted) when used with higher-frequency crystals. Refer to manufacturer's data.

Figure 17. XTAL Oscillator External Connections

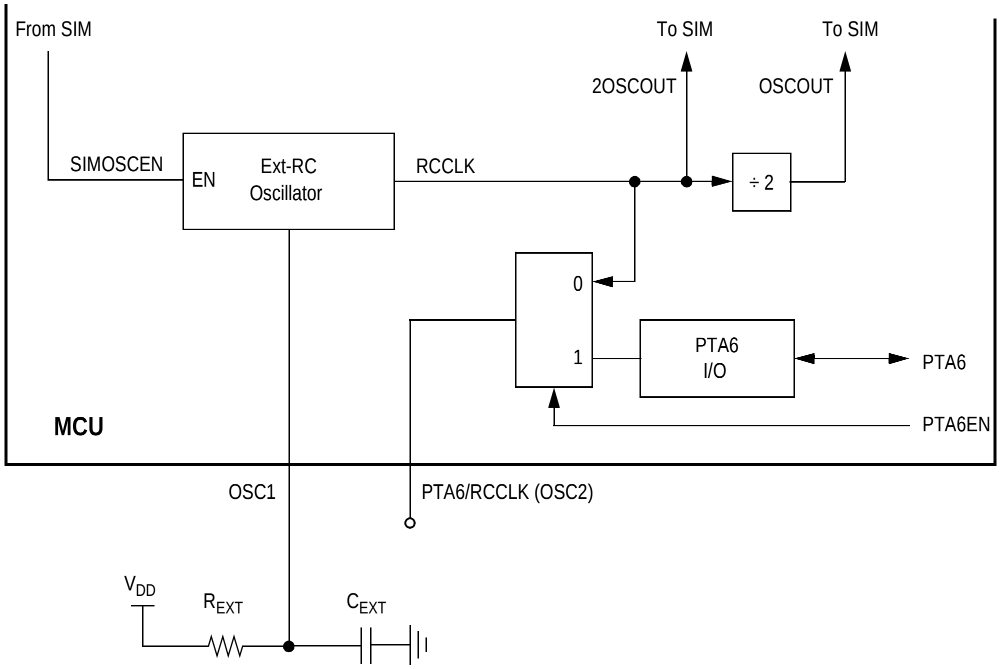


Figure 18. RC Oscillator External Connections

Features of each TIM include the following:

- Two input capture/output compare channels
  - Rising-edge, falling-edge, or any-edge input capture trigger
  - Set, clear, or toggle output compare action
- Buffered and unbuffered pulse width modulation (PWM) signal generation
- Programmable TIM clock input with 7-frequency internal bus clock prescaler selection (external clock input option on TIM2)
- Free-running or modulo up-count operation
- Optional toggle of any channel pin on overflow
- TIM counter stop and reset bits

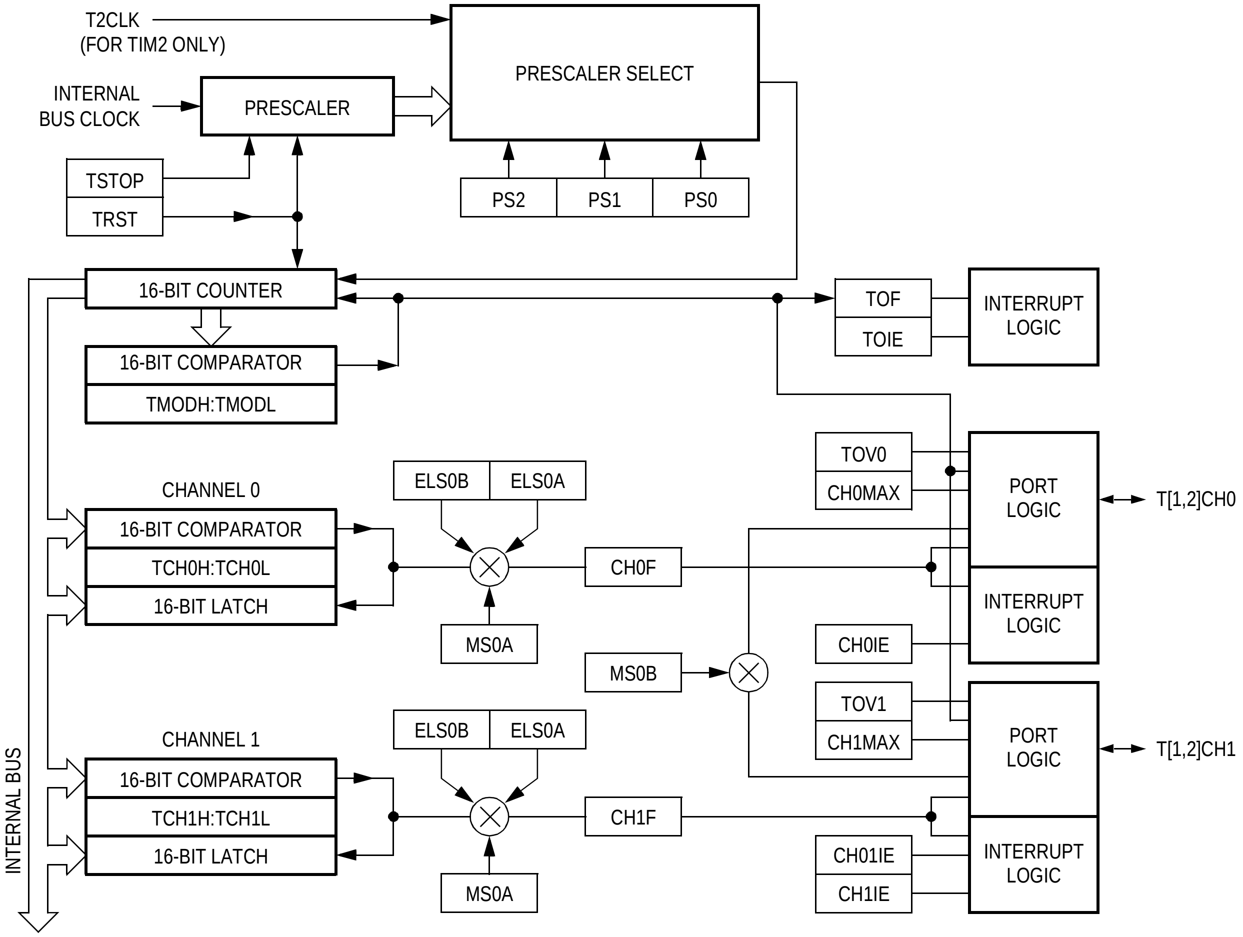


Figure 19. TIM Block Diagram

PWM Initialization

Recommended initialization procedure for unbuffered or buffered PWM signals.

1. In TSC:
   a. Stop the TIM counter by setting TSTOP.
   b. Reset the TIM counter and prescaler by setting TRST.
2. Write TMODH:TMODL to set the required PWM period.
3. Write TCHxH:TCHxL to set the required pulse width.
4. Write TIM channel x status and control register (TSCx) to select the desired function:
   a. Write 0:1 (for unbuffered output compare or PWM signals) or 1:0 (for buffered output compare or PWM signals) to the mode select bits, MSxB:MSxA. See **Table 7**.
   b. Write 1 to the toggle-on-overflow bit, TOVx.
   c. Write 1:0 (to clear output on compare) or 1:1 (to set output on compare) to the edge/level select bits, ELSxB:ELSxA. The output action on compare must force the output to the complement of the pulse width level. See **Table 7**.
5. Clear TSTOP in the TIM status control register (TSC).

TIM Status and Control Register

T1SC: $0020; T2SC: $0030

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

Figure 20. TIM Status and Control Register (TSC)

TOF — TIM Overflow Flag Bit

TOF is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. Clear TOF by reading the TIM status and control register when TOF is set and then writing a logic 0 to TOF.

1 = TIM counter has reached modulo value

TOIE — TIM Overflow Interrupt Enable Bit

1 = TIM overflow interrupts enabled

TSTOP — TIM Stop Bit

1 = TIM counter stopped

TRST — TIM Reset Bit

Setting this write-only bit resets the TIM counter and the TIM prescaler. TRST is cleared automatically after the TIM counter is reset and always reads as logic 0.

1 = Prescaler and TIM counter cleared

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

PS[2:0] — Prescaler Select Bits

Table 6. Prescaler Selection

PS2	PS1	PS0	TIM Clock Source
0	0	0	Internal bus clock ÷ 1
0	0	1	Internal bus clock ÷ 2
0	1	0	Internal bus clock ÷ 4
0	1	1	Internal bus clock ÷ 8
1	0	0	Internal bus clock ÷ 16
1	0	1	Internal bus clock ÷ 32
1	1	0	Internal bus clock ÷ 64
1	1	1	T2CLK (for TIM2 only)

TIM Counter Registers

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

T1CNTH: $0021; T2CNTH: $0031

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

T1CNTL: $0022; T2CNTL: $0032

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

Figure 21. TIM Counter Registers (TCNTH:TCNTL)

TIM Counter Modulo Registers

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.

T1MODH: $0023; T2MODH: $0033

	Bit 7	6	5	4	3	2	1	Bit 0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Reset:	1	1	1	1	1	1	1	1

T1MODL: $0024; T2MODL: $0034

	Bit 7	6	5	4	3	2	1	Bit 0
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reset:	1	1	1	1	1	1	1	1

Figure 22. TIM Counter Modulo Registers (TMODH:TMODL)

TIM Channel Status and Control Registers

T1SC0: $0025; T2SC0: $0035

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

T1SC1: $0028; T2SC1: $0038

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

Figure 23. TIM Channel Status and Control Registers (TSC0, TSC1)

CHxF — Channel x Flag Bit

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

Clear CHxF by reading the TIM channel x status and control register with CHxF set and then writing a logic 0 to CHxF.

1 = Input capture or output compare on channel x

CHxIE — Channel x Interrupt Enable Bit

1 = Channel x CPU interrupt requests enabled

MSxB, MSxA, ELSxB, and ELSxA

Table 7. Mode, Edge, and Level Selection

MSxB	MSxA	ELSxB	ELSxA	Mode	Configuration
X	0	0	0	Output preset	Pin under port control; initial output level high
X	1	0	0		Pin under port control; initial output level low
0	0	0	1	Input capture	Capture on rising edge only
0	0	1	0		Capture on falling edge only
0	0	1	1		Capture on rising or falling edge
0	1	0	1	Output compare or PWM	Toggle output on compare
0	1	1	0		Clear output on compare
0	1	1	1		Set output on compare
1	X	0	1	Buffered output compare or buffered PWM	Toggle output on compare
1	X	1	0		Clear output on compare
1	X	1	1		Set output on compare

TOVx — Toggle-On-Overflow Bit
1 = Channel x pin toggles on TIM counter overflow.

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

CHxMAX — Channel x Maximum Duty Cycle Bit

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

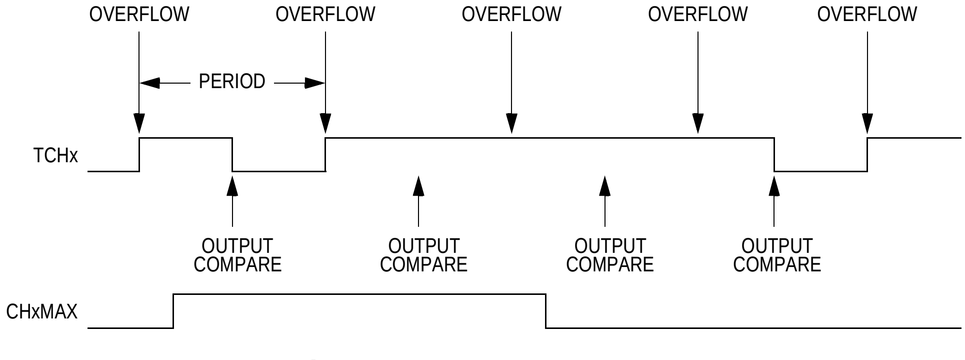


Figure 24. CHxMAX Latency

TIM Channel Registers

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

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

T1CH0H: $0026; T2CH0H: $0036

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

T1CH0L: $0027; T2CH0L: $0037

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

T1CH1H: $0029; T2CH1H: $0039

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

T1CH1L: $002A; T2CH1L: $003A

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

Figure 25. TIM Channel Registers (TCH0H:L, TCH1H:L)

Serial Communications Interface Module (SCI)

Features of the SCI module include the following:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- 32 programmable baud rates
- Programmable 8-bit or 9-bit character length
- Separately enabled transmitter and receiver
- Separate receiver and transmitter CPU interrupt requests
- Programmable transmitter output polarity
- Two receiver wakeup methods:
  - Idle line wakeup
  - Address mark wakeup
- Interrupt-driven operation with eight interrupt flags:
  - Transmitter empty
  - Transmission complete
  - Receiver full
  - Idle receiver input
  - Receiver overrun
  - Noise error
  - Framing error
  - Parity error
- Receiver framing error detection
- Hardware parity checking
- 1/16 bit-time noise detection
- OSCOUT as baud rate clock source

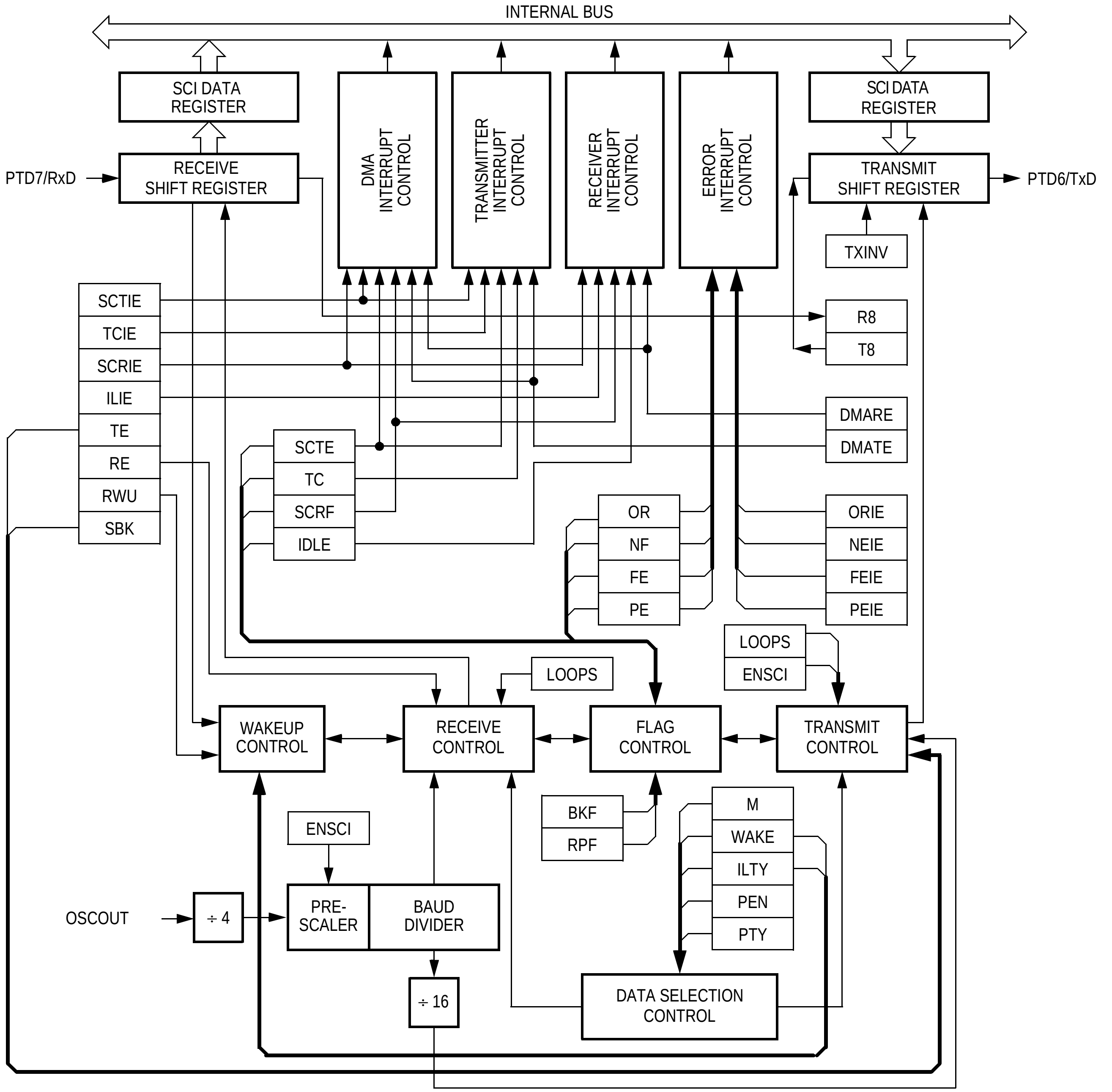


Figure 26. SCI Block Diagram

SCI Control Register 1

$0013	Bit 7	6	5	4	3	2	1	Bit 0
	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
Reset:	0	0	0	0	0	0	0	0

Figure 27. 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
1 = SCI enabled

TXINV — Transmit Inversion Bit
1 = Transmitter output inverted

M — Mode (Character Length) Bit
1 = 9-bit SCI characters
0 = 8-bit SCI characters

WAKE — Wakeup Condition Bit

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

ILTY — Idle Line Type Bit

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

PEN — Parity Enable Bit
1 = Parity function enabled

PTY — Parity Bit
1 = Odd parity
0 = Even parity

Table 8. Character Format Selection

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

SCI Control Register 2

$0014	Bit 7	6	5	4	3	2	1	Bit 0
	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0

Figure 28. SCI Control Register 1 (SCC1)

SCTIE — SCI Transmit Interrupt Enable Bit
1 = SCTE enabled to generate CPU interrupt

TCIE — Transmission Complete Interrupt Enable Bit
1 = TC bit enabled to generate CPU interrupt requests

SCRIE — SCI Receive Interrupt Enable Bit
1 = SCRF bit enabled to generate CPU interrupt

ILIE — Idle Line Interrupt Enable Bit
1 = IDLE bit enabled to generate CPU interrupt requests

TE — Transmitter Enable Bit
1 = Transmitter enabled

RE — Receiver Enable Bit
1 = Receiver enabled

RWU — Receiver Wakeup Bit
1 = Standby state
0 = Normal operation

SBK — Send Break Bit
1 = Transmit break characters
0 = No break characters being transmitted

SCI Control Register 3

$0015	Bit 7	6	5	4	3	2	1	Bit 0
	R8	T8	DMARE	DMATE	ORIE	NEIE	FEIE	PEIE
Reset:	U	U	0	0	0	0	0	0

Figure 29. SCI Control Register 3 (SCC3)

R8 — Received Bit 8

When the SCI is receiving 9-bit characters, R8 is the read-only ninth bit (bit 8) of the received character. R8 is received at the same time that the SCDR receives the other 8 bits.
When the SCI is receiving 8-bit characters, R8 is a copy of the eighth bit (bit 7).

T8 — Transmitted Bit 8

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

DMARE — DMA Receive Enable Bit

This bit should always remain as logic 0.

DMATE — DMA Transfer Enable Bit

This bit should always remain as logic 0.

ORIE — Receiver Overrun Interrupt Enable Bit
1 = SCI error CPU interrupt requests from OR bit enabled

NEIE — Receiver Noise Error Interrupt Enable Bit
1 = SCI error CPU interrupt requests from NE bit enabled

FEIE — Receiver Framing Error Interrupt Enable Bit
1 = SCI error CPU interrupt requests from FE bit enabled

PEIE — Receiver Parity Error Interrupt Enable Bit
1 = SCI error CPU interrupt requests from PE bit enabled

SCI Status Register 1

$0016	Bit 7	6	5	4	3	2	1	Bit 0
	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
Reset:	1	1	0	0	0	0	0	0

Figure 30. SCI Status Register 1 (SCS1)

SCTE — SCI Transmitter Empty Bit

This clearable, read-only bit is set when the SCDR transfers a character to the transmit shift register. SCTE can generate an SCI transmitter CPU interrupt request. When the SCTIE bit in SCC2 is set, SCTE generates an

SCI transmitter CPU interrupt request. In normal operation, clear the SCTE bit by reading SCS1 with SCTE set and then writing to SCDR. Reset sets the SCTE bit.

1 = SCDR data transferred to transmit shift register
0 = SCDR data not transferred to transmit shift register

TC — Transmission Complete Bit

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

1 = No transmission in progress
0 = Transmission in progress

SCRF — SCI Receiver Full Bit

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

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

IDLE — Receiver Idle Bit

This clearable, read-only bit is set when 10 or 11 consecutive logic 1s appear on the receiver input. IDLE generates an SCI receiver 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 31 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
1 = Noise detected

FE — Receiver Framing Error Bit
1 = Framing error detected

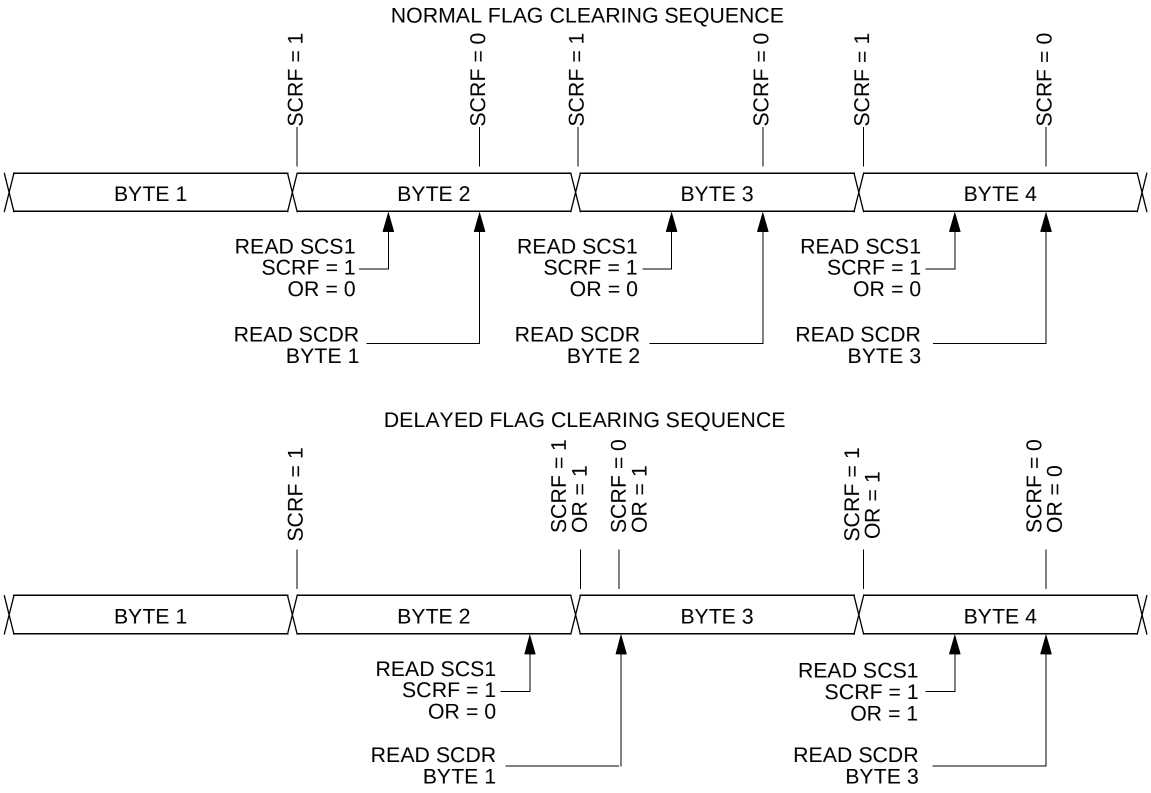


Figure 31. Flag Clearing Sequence

PE — Receiver Parity Error Bit
1 = Parity error detected

SCI Status Register 2

$0017	Bit 7	6	5	4	3	2	1	Bit 0
							BKF	RPF
Reset:	0	0	0	0	0	0	0	0

Figure 32. SCI Status Register 2 (SCS2)

BKF — Break Flag Bit

This clearable, read-only bit is set when the SCI detects a break character on the RxD pin. In SCS1, the FE and SCRF bits are also set. In 9-bit character transmissions, the R8 bit in SCC3 is cleared. BKF does not generate a CPU interrupt request. Clear BKF by reading SCS2 with BKF set and then reading the SCDR. Once cleared, BKF can become set again only after logic 1s again appear on the RxD pin followed by another break character. Reset clears the BKF bit.

1 = Break character detected

RPF — Reception in Progress Flag Bit

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

1 = Reception in progress

SCI Data Register

$0018	Bit 7	6	5	4	3	2	1	Bit 0
	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
Reset:	Unaffected by reset							

Figure 33. SCI Data Register (SCDR)

R7/T7–R0/T0 — Receive/Transmit Data Bits

Reading the SCDR accesses the read-only received data bits, R[7:0]. Writing to the SCDR writes the data to be transmitted, T[7:0]. Reset has no effect on the SCDR.

NOTE: *Do not use read/modify/write instructions on the SCI data register.*

SCI Baud Rate Register

$0019	Bit 7	6	5	4	3	2	1	Bit 0
			SCP1	SCP0	R	SCR2	SCR1	SCR0
Reset:	0	0	0	0	0	0	0	0

Figure 34. 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 9**. Reset clears SCP1 and SCP0.

SCR2–SCR0 — SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate divisor as shown in **Table 9**. Reset clears SCR2–SCR0.

Table 9. SCI Baud Rate Selection

SCR2, SCR1, SCR0	Baud Rate Divisor (BD)
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

Use this formula to calculate the SCI baud rate:

$$\text{baud rate} = \frac{\text{OSCOUT}}{64 \times \text{PD} \times \text{BD}}$$

where: PD = prescaler divisor, BD = baud rate divisor

Table 10 shows the SCI baud rates that can be generated with a 4.9152 MHz OSCOUT clock.

Table 10. SCI Baud Rate Selection Examples

SCP1 and SCP0	Prescaler Divisor (PD)	SCR2, SCR1, and SCR0	Baud Rate Divisor (BD)	Baud Rate (OSCOUT = 4.9152 MHz)
00	1	011	8	9,600
00	1	100	16	4,800
00	1	101	32	2,400
00	1	110	64	1,200
10	4	001	2	9,600
10	4	010	4	4,800
10	4	011	8	2,400
10	4	100	16	1,200

Analog-to-Digital Converter (ADC)

The ADC is an 8-bit, 13-channel analog-to-digital converter.

Features of the ADC module include:

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

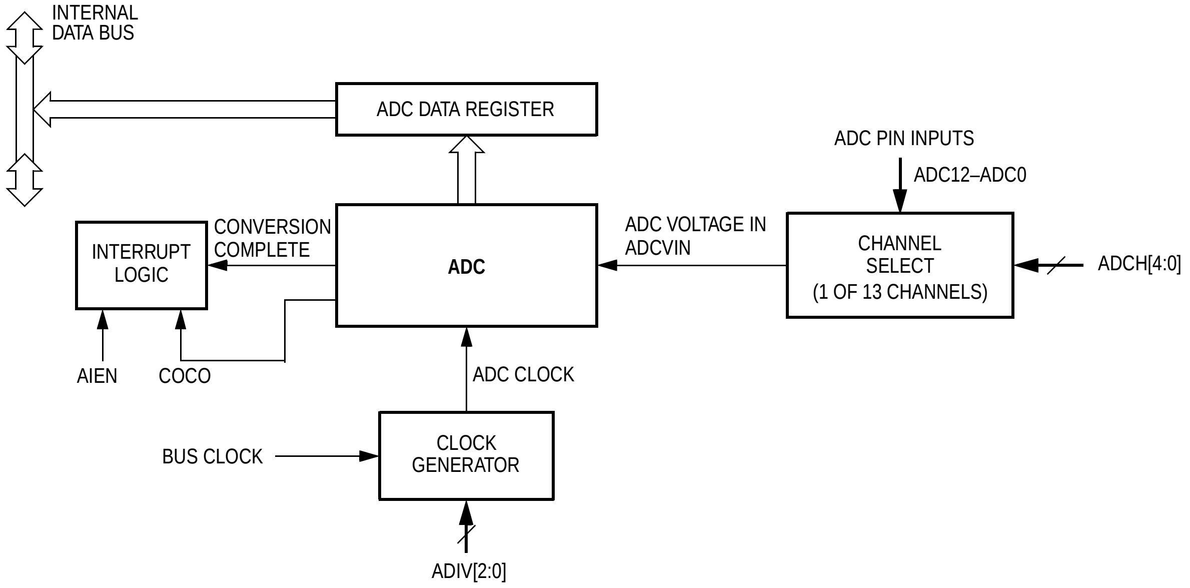


Figure 35. ADC Block Diagram

Conversion Time

$$\text{Conversion Time} = \frac{\text{14 ADC Clock Cycles}}{\text{ADC Clock Frequency}}$$

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

ADC Status and Control Register

$003C	Bit 7	6	5	4	3	2	1	Bit 0
	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
Reset:	0	0	0	1	1	1	1	1

Figure 36. ADC Status and Control Register (ADSCR)

COCO — Conversions Complete Bit

When the AIEN bit is a logic 0, the COCO is a read-only bit which is set each time a conversion is completed. This bit is cleared whenever ADSCR is written or whenever the ADR is read.

When the AIEN bit is a logic 1 (CPU interrupt enabled), COCO will always be logic 0 when read.
1 = Conversion completed (AIEN = 0)

AIEN — ADC Interrupt Enable Bit
1 = ADC interrupt enabled

ADCO — ADC Continuous Conversion Bit
1 = Continuous ADC conversion
0 = Single ADC conversion

ADCH[4:0] — ADC Channel Select Bits

NOTE: *Startup from the ADC power off state requires one conversion cycle to stabilize.*

Table 11. MUX Channel Select

ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	ADC Channel	Input Select
0	0	0	0	0	ADC0	PTB0
0	0	0	0	1	ADC1	PTB1
0	0	0	1	0	ADC2	PTB2
0	0	0	1	1	ADC3	PTB3
0	0	1	0	0	ADC4	PTB4
0	0	1	0	1	ADC5	PTB5
0	0	1	1	0	ADC6	PTB6
0	0	1	1	1	ADC7	PTB7
0	1	0	0	0	ADC8	PTD3
0	1	0	0	1	ADC9	PTD2
0	1	0	1	0	ADC10	PTD1
0	1	0	1	1	ADC11	PTD0
0	1	1	0	0	ADC12	ADC12

Table 11. MUX Channel Select (Continued)

ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	ADC Channel	Input Select
0 ↓ 1	1 ↓ 1	1 ↓ 0	0 ↓ 1	1 ↓ 0	— — —	Unused[1]
1	1	0	1	1	—	Reserved
1	1	1	0	0	—	Reserved
1	1	1	0	1	—	V_{DDA}[2]
1	1	1	1	0	—	V_{SSA}[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.

ADC Data Register

This register is updated each time an ADC conversion completes.

$003D	Bit 7	6	5	4	3	2	1	Bit 0
	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Reset:	Indeterminate after reset							

Figure 37. ADC Data Register (ADR)

ADC Input Clock Register

$003E	Bit 7	6	5	4	3	2	1	Bit 0
	ADIV2	ADIV1	ADIV0	0	0	0	0	0
Reset:	0	0	0	0	0	0	0	0

Figure 38. ADC Input Clock Register (ADICLK)

ADIV2–ADIV0 — ADC Clock Prescaler Bits

Table 12. ADC Clock Divide Ratio

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

X = don’t care

Input/Output (I/O) Ports

Port A

Port A is an 8-bit special function port that shares all of its pins with the keyboard interrupt (KBI) module. Each port A pin also has a software configurable pullup device if the corresponding port pin is configured as a general-purpose input port pin. PTA0–PTA5 and PTA7 has direct LED drive capability.

NOTE: *PTA0–PTA5 pins are available on 28-pin and 32-pin packages only. PTA7 pin is available on 32-pin packages only.*

Port A Data Register

$0000	Bit 7	6	5	4	3	2	1	Bit 0
	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Reset:	Unaffected by reset							
Additional Function:	KBI6	KBI6	KBI5	KBI4	KBI3	KBI2	KBI1	KBI0

Figure 39. Port A Data Register (PTA)

PTA[7:0] — Port A Data Bits

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

Data Direction Register A

$0004	Bit 7	6	5	4	3	2	1	Bit 0
	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset:	0	0	0	0	0	0	0	0

Figure 40. Data Direction Register A (DDRA)

DDRA[7:0] — Data Direction Register A Bits

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

Port A Input Pullup Enable Registers

$000D	Bit 7	6	5	4	3	2	1	Bit 0
	PTA6EN	PTAPUE6	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE2	PTAPUE0
Reset:	0	0	0	0	0	0	0	0

Figure 41. Port A Input Pull-up Enable Register (PTAPUE)

$000E	Bit 7	6	5	4	3	2	1	Bit 0
	PTAPUE7							
Reset:	0	0	0	0	0	0	0	0

Figure 42. PTA7 Input Pull-up Enable Register (PTA7PUE)

PTA6EN — Enable PTA6 on OSC2

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

1 = OSC2 pin configured for PTA6 I/O, and has all the interrupt and pull-up functions
0 = OSC2 pin outputs the RC oscillator clock (RCCLK)

PTAPUE[7:0] — Port A Input Pull-up Enable Bits

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

1 = Corresponding port A pin configured to have internal pull-up if its DDRA bit is set to 0

Port B

Port B is an 8-bit special function port that shares all of its port pins with the analog-to-digital converter (ADC) module.

Port B Data Register

$0001	Bit 7	6	5	4	3	2	1	Bit 0
	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Reset:	Unaffected by reset							
Additional Function:	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0

Figure 43. Port B Data Register (PTB)

PTB[7:0] — Port B Data Bits

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

Data Direction Register B

$0005	Bit 7	6	5	4	3	2	1	Bit 0
	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Reset:	0	0	0	0	0	0	0	0

Figure 44. Data Direction Register B (DDRB)

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

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

Port D

Port D is an 8-bit special function port that shares two of its pins with the serial communications interface module, two of its pins with the timer 1 interface module, and four of its pins with the analog-to-digital converter module. PTD6 and PTD7 each has high current sink (25mA) and programmable pull-up. PTD2, PTD3, PTD6 and PTD7 each has LED sink capability.

NOTE: *PTD0–PTD1 are available on 28-pin and 32-pin packages only.*

Port D Data Register

$0003	Bit 7	6	5	4	3	2	1	Bit 0
	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
Reset:	Unaffected by reset							
Additional Function:	RxD	TxD	T1CH1	T1CH0	ADC8	ADC7	ADC6	ADC5

Figure 45. Port D Data Register (PTD)

PTD[7:0] — Port D Data Bits

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

Data Direction Register D

$0007	Bit 7	6	5	4	3	2	1	Bit 0
	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Reset:	0	0	0	0	0	0	0	0

Figure 46. Data Direction Register B (DDRB)

DDRD[7:0] — Data Direction Register D Bits

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

NOTE: *For MC68HC908JK8 (devices packaged in a 20-pin package), PTD0–PTD1 and are not connected. DDRD0–DDRD1 should be set to a 1 to configure PTD0–PTD1 as outputs.*

Port D Control Register

$000A	Bit 7	6	5	4	3	2	1	Bit 0
	▨	▨	▨	▨	SLOWD7	SLOWD6	PTDPU7	PTDPU6
Reset:	0	0	0	0	0	0	0	0

Figure 47. Port D Control Register (PDCR)

SLOWDx — Slow Edge Enable

The SLOWD6 and SLOWD7 bits enable the slow-edge, open-drain, high current output (25mA sink) of port pins PTD6 and PTD7 respectively. DDRDx bit is not affected by SLOWDx.

1 = Slow edge enabled; pin is open-drain output

PTDPUx — Pull-up Enable

The PTDPU6 and PTDPU7 bits enable the 5kΩ pull-up on PTD6 and PTD7 respectively, regardless the status of DDRDx bit.

1 = Enable 5kΩ pull-up

Port E

Port D is a 2-bit special function port that shares its pins with the timer 2 interface module.

NOTE: *PTE0–PTE1 are available on 32-pin packages only.*

Port E Data Register

$0008	Bit 7	6	5	4	3	2	1	Bit 0
							PTE1	PTE0
Reset:	Unaffected by reset							
Additional Function:							T2CH1	T2CH0

Figure 48. Port E Data Register (PTE)

PTE[1:0] — Port E Data Bits

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

Data Direction Register E

$0007	Bit 7	6	5	4	3	2	1	Bit 0
							DDRE1	DDRE0
Reset:	0	0	0	0	0	0	0	0

Figure 49. Data Direction Register B (DDRB)

DDRE[1:0] — Data Direction Register E Bits

1 = Corresponding port E pin configured as output

0 = Corresponding port E pin configured as input

NOTE: *For those devices packaged in a 20-pin package and 28-pin package, PTE0–PTE1 are not connected. DDRE0–DDRE1 should be set to a 1 to configure PTE0–PTE1 as outputs.*

IRQ Status and Control Register

$001D	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	IRQF	ACK	IMASK	MODE
Reset:	0	0	0	0	0	0	0	0

Figure 50. IRQ Status and Control Register (INTSCR)

IRQF — IRQ Flag

This read-only status bit is high when the IRQ interrupt is pending.
1 = $\overline{\text{IRQ}}$ interrupt pending

ACK — IRQ Interrupt Request Acknowledge Bit

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

IMASK — IRQ Interrupt Mask Bit
1 = IRQ interrupt requests disabled

MODE — IRQ Edge/Level Select Bit

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

Keyboard Interrupt Module (KBI)

Features of the keyboard interrupt module include:

- Eight keyboard interrupt pins with pull-up devices
- Separate keyboard interrupt enable bits and one keyboard interrupt mask
- Programmable edge-only or edge- and level- interrupt sensitivity
- Exit from low-power modes

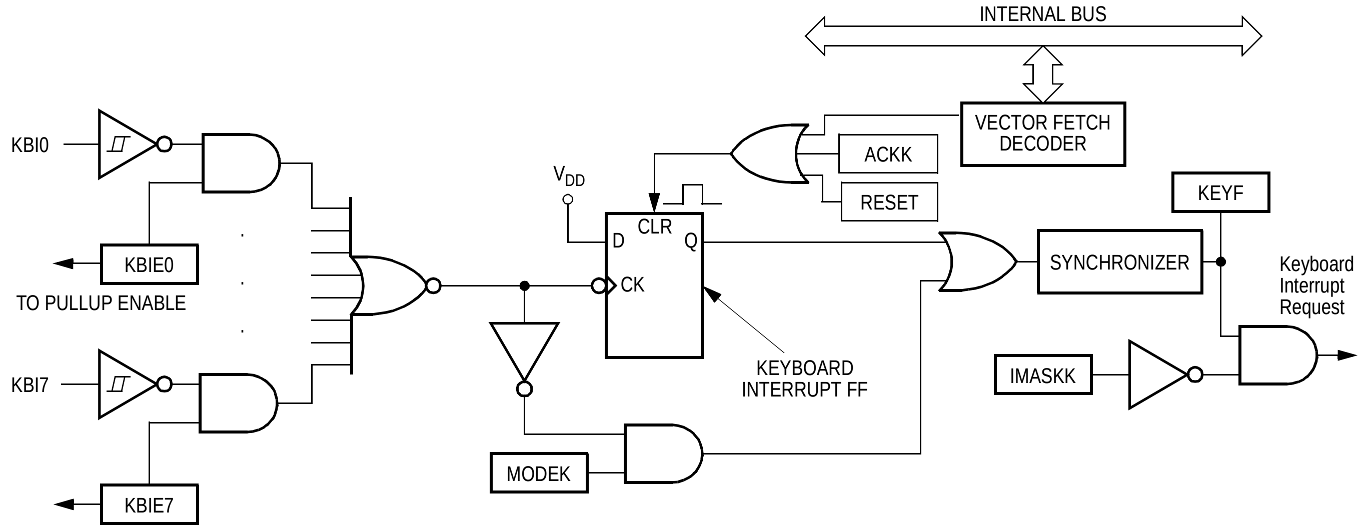


Figure 51. Keyboard Interrupt Block Diagram

Keyboard Status and Control Register

$001A	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	KEYF	ACKK	IMASKK	MODEK
Reset:	0	0	0	0	0	0	0	0

Figure 52. Keyboard Status and Control Register (KBSCR)

KEYF — Keyboard Flag Bit
1 = Keyboard interrupt pending

ACKK — Keyboard Acknowledge Bit

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

IMASKK— Keyboard Interrupt Mask Bit
1 = Keyboard interrupt requests masked (disabled)

MODEK — Keyboard Triggering Sensitivity Bit
1 = Keyboard interrupt requests on falling edges and low levels
0 = Keyboard interrupt requests on falling edges only

Keyboard Interrupt Enable Register

$001B	Bit 7	6	5	4	3	2	1	Bit 0
Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Reset:	0	0	0	0	0	0	0	0

Figure 53. Keyboard Interrupt Enable Register (KBIER)

KBIE7–KBIE0 — Port A Keyboard Interrupt Enable Bits
1 = KBIx pin enabled as keyboard interrupt pin

Condensed Electrical Characteristics

For more detailed information refer to the *MC68HC908JL8 Data Sheet* (Motorola document order number MC68HC908JL8/D).

5-Volt DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
V_{DD} supply current Run, f_{OP} = 8 MHz[3], XTAL option RC option Wait, f_{OP} = 8 MHz[4], XTAL option RC option Stop[5], –40°C to 125°C, XTAL option RC option	I_{DD}	 — — — — — —	 7.5 11 3 3.5 1.5 0.5	 10 13 5.5 6 3 3	 mA mA mA mA µA µA
POR rearm voltage[6]	V_{POR}	0	—	100	mV
POR rise time ramp rate[7]	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{TST}	$1.5 \times V_{DD}$	—	8.5	V
Pullup resistors[8] PTD6, PTD7 $\overline{RST}$, $\overline{IRQ}$, PTA0–PTA7	R_{PU}	 1.8 16	 3.3 26	 4.8 36	 kΩ kΩ
Low-voltage inhibit reset, trip falling voltage	V_{TRIPF}	3.60	4.25	4.48	V
Low-voltage inhibit reset, trip rising voltage	V_{TRIPR}	3.75	4.40	4.63	V

1. V_{DD} = 4.5 to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25°C only.
3. Run (operating) I_{DD} measured using external square wave clock source (f_{OP} = 8 MHz). 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. Wait I_{DD} measured using external square wave clock source (f_{OP} = 8 MHz). All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. All ports configured as inputs.
5. Stop I_{DD} measured with OSC1 grounded; no port pins sourcing current. LVI is disabled.
6. Maximum is highest voltage that POR is guaranteed.
7. If minimum V_{DD} is not reached before the internal POR reset is released, $\overline{RST}$ must be driven low externally until minimum V_{DD} is reached.
8. R_{PU1} and R_{PU2} are measured at V_{DD} = 5.0 V.

5-Volt Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency[2]	f_{OP}	—	8	MHz
$\overline{RST}$ input pulse width low[3]	t_{IRL}	750	—	ns

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

5-Volt Oscillator Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Internal oscillator frequency	f_{ICLK}		50k[1]		Hz
Crystal frequency, XTALCLK	$f_{XTALCLK}$		—	32M	Hz
RC oscillator frequency, RCCLK	f_{RCCLK}	2M	—	12M	Hz
External clock reference frequency[2]	f_{OSC}	dc	—	32M	Hz

[1] See Figure 55 for plot.

[2] No more than 10% duty cycle deviation from 50%.

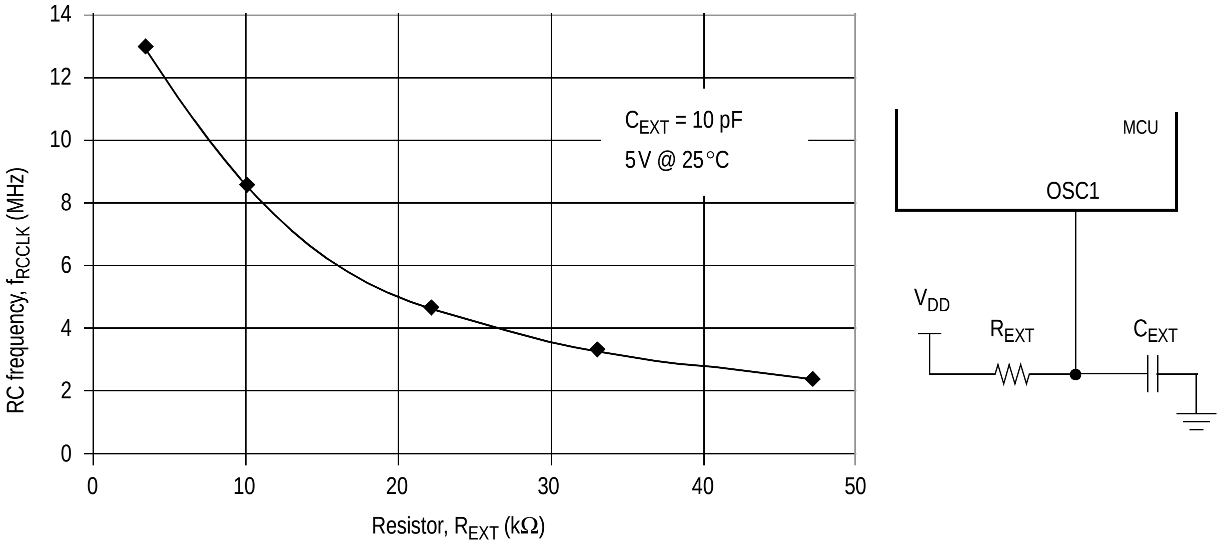


Figure 54. RC versus Frequency (5 Volts @ 25°C)

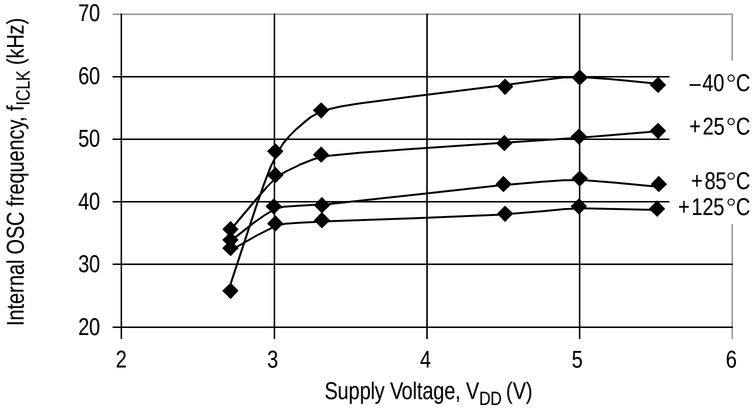


Figure 55. Internal Oscillator Frequency

3-Volt DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
V_{DD} supply current Run, f_{OP} = 4 MHz[3], XTAL option RC option Wait, f_{OP} = 4 MHz[4], XTAL option RC option Stop[5], –40°C to 85°C, XTAL option RC option	I_{DD}	 — — — — — —	 3 4 1 2 0.5 0.3	 8 10 4.5 6 5 2	 mA mA mA mA µA µA
POR rearm voltage[6]	V_{POR}	0	—	100	mV
POR rise time ramp rate[7]	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{TST}	$1.5 \times V_{DD}$	—	8.5	V
Pullup resistors[8] PTD6, PTD7 $\overline{\text{RST}}$, $\overline{\text{IRQ}}$, PTA0–PTA7	R_{PU}	 1.8 16	 3.3 26	 4.8 36	 kΩ kΩ
Low-voltage inhibit reset, trip voltage (no hysteresis)	V_{LVI3}	2.18	2.49	2.68	V

1. V_{DD} = 2.7 to 3.3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25°C only.
3. Run (operating) I_{DD} measured using external square wave clock source (f_{OP} = 4 MHz). 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. Wait I_{DD} measured using external square wave clock source (f_{OP} = 4 MHz). All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. All ports configured as inputs.
5. Stop I_{DD} measured with OSC1 grounded; no port pins sourcing current. LVI is disabled.
6. Maximum is highest voltage that POR is guaranteed.
7. If minimum V_{DD} is not reached before the internal POR reset is released, $\overline{\text{RST}}$ must be driven low externally until minimum V_{DD} is reached.
8. R_{PU1} and R_{PU2} are measured at V_{DD} = 5.0 V.

3-Volt Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency[2]	f_{OP}	—	4	MHz
$\overline{\text{RST}}$ input pulse width low[3]	t_{IRL}	1.5	—	µs

1. V_{DD} = 2.7 to 3.3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H; timing shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Some modules may require a minimum frequency greater than dc for proper operation; see appropriate table for this information.
3. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.

3-Volt Oscillator Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Internal oscillator frequency	f_{INTCLK}		45k[1]		Hz
Crystal frequency, XTALCLK	$f_{OSCXCLK}$		—	16M	Hz
RC oscillator frequency, RCCLK	f_{RCCLK}	2M	—	10M	Hz
External clock reference frequency[2]	$f_{OSCXCLK}$	dc	—	16M	Hz

1. See **Figure 55** for plot.
2. No more than 10% duty cycle deviation from 50%

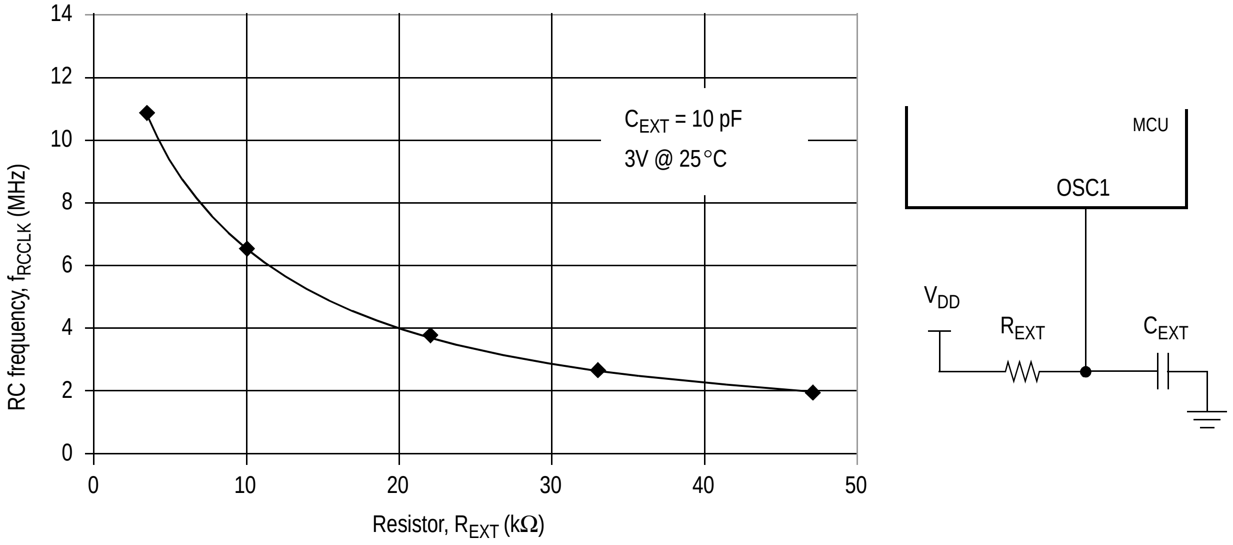


Figure 56. RC versus Frequency (3 Volts @ 25°C)

Typical Supply Currents

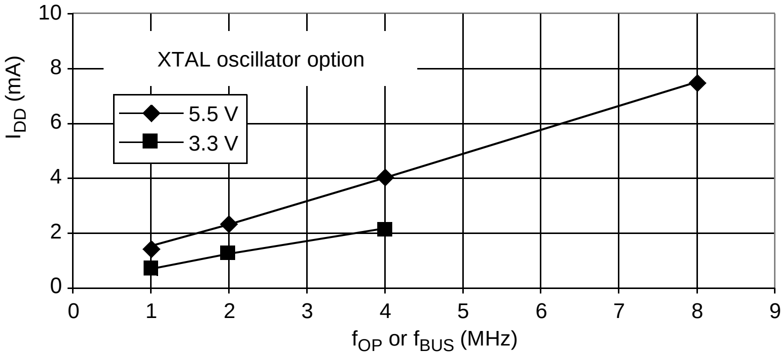


Figure 57. Typical Operating I_{DD} (XTAL osc), with All Modules Turned On (25°C)

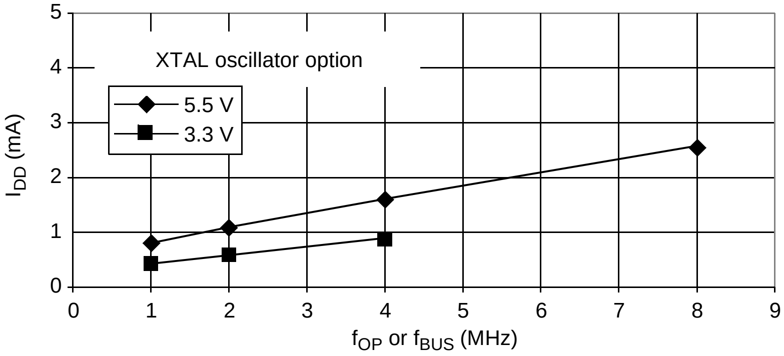


Figure 58. Typical Wait Mode I_{DD} (XTAL osc), with All Modules Turned Off (25°C)

Analog-to-Digital Converter Characteristics

Characteristic	Symbol	Min	Max	Unit	Comments
Input voltages	V_{ADIN}	V_{SS}	V_{DD}	V	—
Resolution	B_{AD}	8	8	Bits	—
Absolute accuracy	A_{AD}	± 0.5	± 1.5	LSB	Includes quantization
ADC internal clock	f_{ADIC}	0.5	1.048	MHz	$t_{ADIC} = 1/f_{ADIC}$, tested only at 1 MHz
Conversion range	R_{AD}	V_{SS}	V_{DD}	V	—
Power-up time	t_{ADPU}	16		t_{ADIC} cycles	$t_{ADIC} = 1/f_{ADIC}$
Conversion time	t_{ADC}	14	15	t_{ADIC} cycles	$t_{ADIC} = 1/f_{ADIC}$
Sample time[1]	t_{ADS}	5	—	t_{ADIC} cycles	$t_{ADIC} = 1/f_{ADIC}$
Zero input reading[2]	Z_{ADI}	00	01	Hex	$V_{IN} = V_{SS}$
Full-scale reading[3]	F_{ADI}	FE	FF	Hex	$V_{IN} = V_{DD}$
Input capacitance	C_{ADI}	—	8	pF	Not tested
Input leakage[3]	—	—	± 1	µA	—

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

Memory Characteristics

Characteristic	Symbol	Min	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]	32 k	8M	Hz
FLASH page erase time	t_{erase}[2]	4	—	ms
FLASH mass erase time	t_{merase}[3]	4	—	ms
FLASH PGM/ERASE to HVEN set up 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}[4]	1	—	µs
FLASH cumulative program hv period	t_{HV}[5]	—	4	ms
FLASH row erase/program endurance[6]	—	10 k	—	cycles
FLASH data retention time[7]	—	10	—	years

[1] f_{Read} is defined as the frequency range for which the FLASH memory can be read.
[2] If the page erase time is longer than t_{erase} min, there is no erase disturb, but it reduces the endurance of the FLASH memory.
[3] If the mass erase time is longer than t_{merase} min, there is no erase disturb, but it reduces the endurance of the FLASH memory.
[4] t_{rcv} is defined as the time it needs before the FLASH can be read after turning off the high voltage charge pump, by clearing HVEN to logic 0.
[5] t_{HV} is defined as the cumulative high voltage programming time to the same row before next erase.
t_{HV} must satisfy this condition: $t_{nvs} + t_{nvh} + t_{pgs} + (t_{prog} \times 32) \leq t_{HV}$ max.
[6] The minimum row endurance value specifies each row of the FLASH memory is guaranteed to work for at least this many erase/program cycles.
[7] The FLASH is guaranteed to retain data over the entire operating temperature range for at least the minimum time specified.

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
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
+81 2666 8080
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
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

RoHS-compliant and/or Pb- free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb- free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale.s Environmental Products program, go to http://www.freescale.com/epp.

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.
Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

freescale™
semiconductor

MC68HC908JL8SM/D
Rev. 0
3/2003