

Welcome to [E-XFL.COM](#)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	2MHz
Connectivity	-
Peripherals	LVD, POR, PWM
Number of I/O	13
Program Memory Size	1.5KB (1.5K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	16-DIP (0.300", 7.62mm)
Supplier Device Package	16-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mchlc908qy1pe

Table of Contents

Chapter 1 General Description

1.1	Introduction	15
1.2	Features	15
1.3	MCU Block Diagram	17
1.4	Pin Assignments	17
1.5	Pin Functions	20
1.6	Pin Function Priority	21

Chapter 2 Memory

2.1	Introduction	23
2.2	Unimplemented Memory Locations	23
2.3	Reserved Memory Locations	23
2.4	Input/Output (I/O) Section	25
2.5	Random-Access Memory (RAM)	31
2.6	FLASH Memory (FLASH)	32
2.6.1	FLASH Control Register	32
2.6.2	FLASH Page Erase Operation	33
2.6.3	FLASH Mass Erase Operation	34
2.6.4	FLASH Program Operation	34
2.6.5	FLASH Protection	35
2.6.6	FLASH Block Protect Register	37
2.6.7	Wait Mode	38
2.6.8	Stop Mode	38

Chapter 3 Analog-to-Digital Converter (ADC)

3.1	Introduction	39
3.2	Features	39
3.3	Functional Description	39
3.3.1	ADC Port I/O Pins	39
3.3.2	Voltage Conversion	41
3.3.3	Conversion Time	42
3.3.4	Continuous Conversion	42
3.3.5	Accuracy and Precision	42
3.4	Interrupts	42
3.5	Low-Power Modes	42
3.5.1	Wait Mode	42
3.5.2	Stop Mode	42

Chapter 1

General Description

1.1 Introduction

The MC68HLC908QY4 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 1-1. Summary of Device Variations

Device	FLASH Memory Size	Analog-to-Digital Converter	Pin Count
MC68HLC908QT1	1536 bytes	—	8 pins
MC68HLC908QT2	1536 bytes	4 ch, 8 bit	8 pins
MC68HLC908QT4	4096 bytes	4 ch, 8 bit	8 pins
MC68HLC908QY1	1536 bytes	—	16 pins
MC68HLC908QY2	1536 bytes	4 ch, 8 bit	16 pins
MC68HLC908QY4	4096 bytes	4 ch, 8 bit	16 pins

1.2 Features

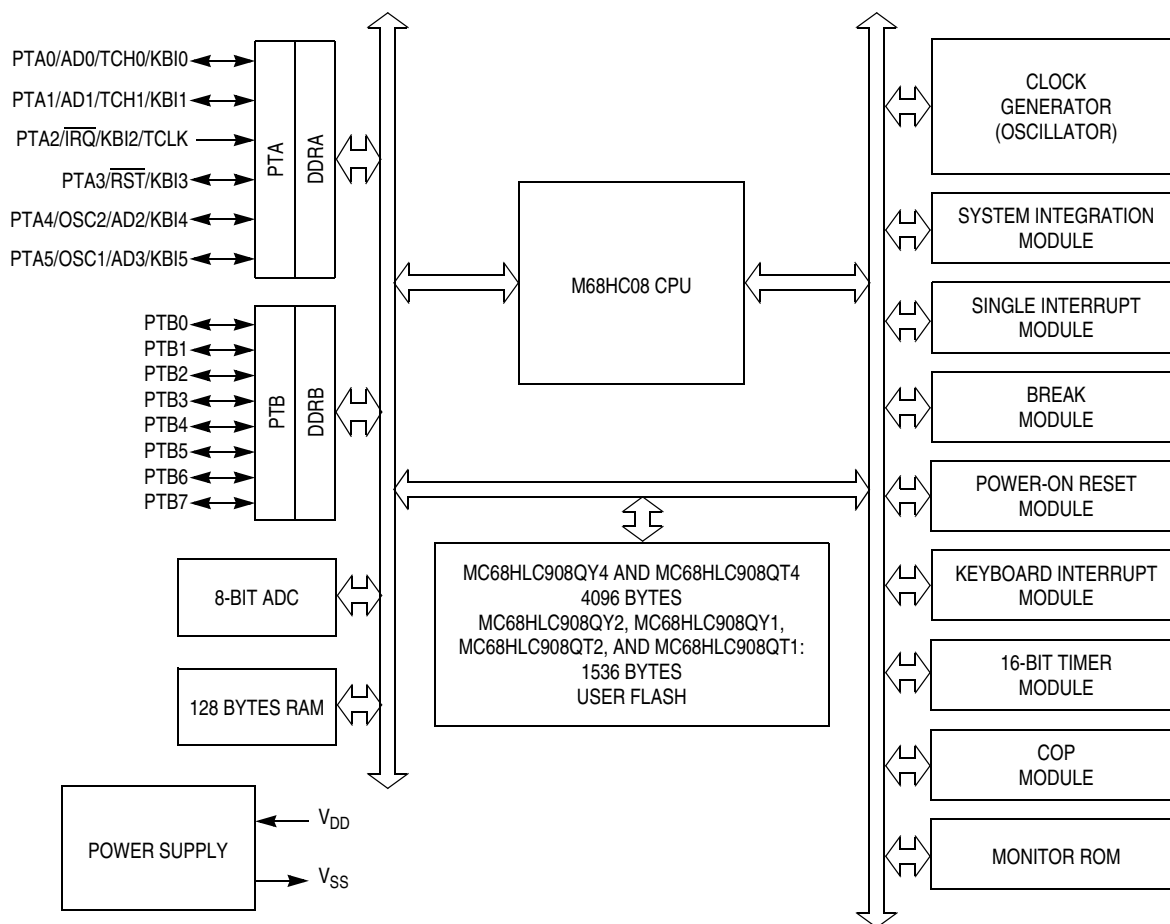
Features include:

- High-performance M68HC08 CPU core
- Fully upward-compatible object code with M68HC05 Family
- Operating voltage range of 2.2 V to 3.6 V
- 2-MHz internal bus operation
- Trimmable internal oscillator
 - 1.0 MHz internal bus operation
 - 8-bit trim capability allows 0.4% accuracy⁽¹⁾
 - $\pm 25\%$ untrimmed
- Auto wakeup from STOP capability
- Configuration (CONFIG) register for MCU configuration options, including:
 - Low-voltage inhibit (LVI) trip point
- In-system FLASH programming
- FLASH security⁽²⁾

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

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

General Description



\overline{RST} , \overline{IRQ} : Pins have internal (about 30K Ohms) pull up

PTA[0:5]: High current sink and source capability

PTA[0:5]: Pins have programmable keyboard interrupt and pull up

PTB[0:7]: Not available on 8-pin devices – MC68HLC908QT1, MC68HLC908QT2, and MC68HLC908QT4 (see note in [12.1 Introduction](#))

ADC: Not available on the MC68HLC908QY1 and MC68HLC908QT1

Figure 1-1. Block Diagram

Table 2-1. Vector Addresses

Vector Priority	Vector	Address	Vector
<div style="text-align: center;"> Lowest Highest </div>	IF15	\$FFDE	ADC conversion complete vector (high)
		\$FFDF	ADC conversion complete vector (low)
	IF14	\$FFE0	Keyboard vector (high)
		\$FFE1	Keyboard vector (low)
	IF13 ↓ IF6	—	Not used
	IF5	\$FFF2	TIM overflow vector (high)
		\$FFF3	TIM overflow vector (low)
	IF4	\$FFF4	TIM Channel 1 vector (high)
		\$FFF5	TIM Channel 1 vector (low)
	IF3	\$FFF6	TIM Channel 0 vector (high)
		\$FFF7	TIM Channel 0 vector (low)
	IF2	—	Not used
	IF1	\$FFFA	$\overline{\text{IRQ}}$ vector (high)
		\$FFFFB	$\overline{\text{IRQ}}$ vector (low)
	—	\$FFFC	SWI vector (high)
		\$FFFD	SWI vector (low)
	—	\$FFFE	Reset vector (high)
		\$FFFF	Reset vector (low)

2.5 Random-Access Memory (RAM)

The 128 bytes of random-access memory (RAM) are located at addresses \$0080–\$00FF. The location of the stack RAM is programmable. The 16-bit stack pointer allows the stack to be anywhere in the 64-Kbyte memory space.

NOTE

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

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

NOTE

For M6805, M146805, and M68HC05 compatibility, the H register is not stacked.

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

NOTE

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

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.

NOTE

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

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

NOTE

Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order shown, other unrelated operations may occur between the steps. Do not exceed t_{PROG} maximum, see [16.12 Memory Characteristics](#).

2.6.5 FLASH Protection

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

NOTE

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

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

Chapter 3

Analog-to-Digital Converter (ADC)

3.1 Introduction

This section describes the analog-to-digital converter (ADC). The ADC is an 8-bit, 4-channel analog-to-digital converter. The ADC module is only available on the MC68HLC908QY2, MC68HLC908QT2, MC68HLC908QY4, and MC68HLC908QT4.

3.2 Features

Features of the ADC module include:

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

3.3 Functional Description

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

Figure 3-2 shows a block diagram of the ADC.

3.3.1 ADC Port I/O Pins

PTA0, PTA1, PTA4, and PTA5 are general-purpose I/O pins that are shared with the ADC channels. The channel select bits (ADC status and control register (ADSCR), \$003C), define which ADC channel/port pin will be used as the input signal. The ADC overrides the port I/O logic by forcing that pin as input to the ADC. The remaining ADC channels/port pins are controlled by the port I/O logic and can be used as general-purpose I/O. Writes to the port register or data direction register (DDR) will not have any 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 0 if the corresponding DDR bit is at 0. If the DDR bit is 1, the value in the port data latch is read.

6.3.7 COPRS (COP Rate Select)

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

6.4 COP Control Register

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

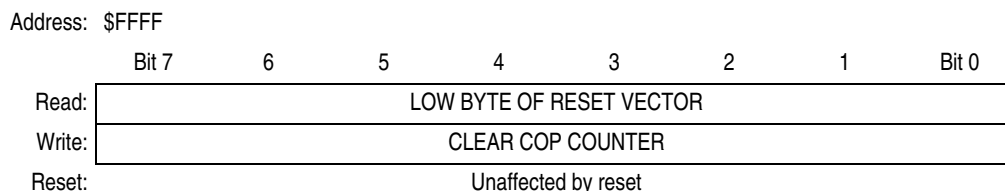


Figure 6-2. COP Control Register (COPCTL)

6.5 Interrupts

The COP does not generate CPU interrupt requests.

6.6 Monitor Mode

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

6.7 Low-Power Modes

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

6.7.1 Wait Mode

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

6.7.2 Stop Mode

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

6.8 COP Module During Break Mode

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

Z — Zero Flag

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

1 = Zero result

0 = Non-zero result

C — Carry/Borrow Flag

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

1 = Carry out of bit 7

0 = No carry out of bit 7

7.4 Arithmetic/Logic Unit (ALU)

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

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

7.5 Low-Power Modes

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

7.5.1 Wait Mode

The WAIT instruction:

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

7.5.2 Stop Mode

The STOP instruction:

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

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

7.6 CPU During Break Interrupts

If a break module is present on the MCU, the CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with \$FFFC:\$FFFD or with \$FEFC:\$FEFD in monitor mode

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

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

Table 7-1. Instruction Set Summary (Sheet 4 of 6)

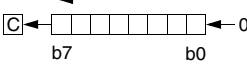
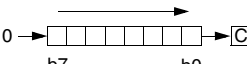
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z				
JMP <i>opr</i> JMP <i>opr</i> JMP <i>opr,X</i> JMP <i>opr,X</i> JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff	2 3 4 3 2
JSR <i>opr</i> JSR <i>opr</i> JSR <i>opr,X</i> JSR <i>opr,X</i> JSR ,X	Jump to Subroutine	PC ← (PC) + <i>n</i> (<i>n</i> = 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 # <i>opr</i> LDA <i>opr</i> LDA <i>opr</i> LDA <i>opr,X</i> LDA <i>opr,X</i> LDA ,X LDA <i>opr</i> ,SP LDA <i>opr</i> ,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 # <i>opr</i> LDHX <i>opr</i>	Load H:X from M	H:X ← (M:M + 1)	0	–	–	↑	↑	IMM DIR	45 55	ii jj dd	3 4
LDX # <i>opr</i> LDX <i>opr</i> LDX <i>opr</i> LDX <i>opr,X</i> LDX <i>opr,X</i> LDX ,X LDX <i>opr</i> ,SP LDX <i>opr</i> ,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 ff ee ff	2 3 4 4 3 2 4 5
LSL <i>opr</i> LSLA LSLX LSL <i>opr,X</i> LSL ,X LSL <i>opr</i> ,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 <i>opr</i> LSRA LSRX LSR <i>opr,X</i> LSR ,X LSR <i>opr</i> ,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 <i>opr,opr</i> MOV <i>opr,X</i> + MOV # <i>opr,opr</i> MOV X+, <i>opr</i>	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	–	–	–	INH	42		5
NEG <i>opr</i> NEGA NEGX NEG <i>opr,X</i> NEG ,X NEG <i>opr</i> ,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 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 # <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ORA <i>opr,X</i> ORA <i>opr,X</i> ORA ,X ORA <i>opr</i> ,SP ORA <i>opr</i> ,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 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

Table 7-1. 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 \leftarrow (PC) + 1; Push (PCL) SP \leftarrow (SP) - 1; Push (PCH) SP \leftarrow (SP) - 1; Push (X) SP \leftarrow (SP) - 1; Push (A) SP \leftarrow (SP) - 1; Push (CCR) SP \leftarrow (SP) - 1; I \leftarrow 1 PCH \leftarrow Interrupt Vector High Byte PCL \leftarrow Interrupt Vector Low Byte	-	-	1	-	-	-	INH	83		9
TAP	Transfer A to CCR	CCR \leftarrow (A)	\uparrow	\uparrow	\uparrow	\uparrow	\uparrow	\uparrow	INH	84		2
TAX	Transfer A to X	X \leftarrow (A)	-	-	-	-	-	-	INH	97		1
TPA	Transfer CCR to A	A \leftarrow (CCR)	-	-	-	-	-	-	INH	85		1
TST <i>opr</i> TSTA TSTX TST <i>opr</i> ,X TST ,X TST <i>opr</i> ,SP	Test for Negative or Zero	(A) - \$00 or (X) - \$00 or (M) - \$00	0	-	-	\uparrow	\uparrow	-	DIR INH INH IX1 IX SP1	3D 4D 5D 6D 7D 9E6D	dd ff ff	3 1 1 3 2 4
TSX	Transfer SP to H:X	H:X \leftarrow (SP) + 1	-	-	-	-	-	-	INH	95		2
TXA	Transfer X to A	A \leftarrow (X)	-	-	-	-	-	-	INH	9F		1
TXS	Transfer H:X to SP	(SP) \leftarrow (H:X) - 1	-	-	-	-	-	-	INH	94		2
WAIT	Enable Interrupts; Wait for Interrupt	I bit \leftarrow 0; Inhibit CPU clocking until interrupted	-	-	0	-	-	-	INH	8F		1

A	Accumulator	<i>n</i>	Any bit
C	Carry/borrow bit	<i>opr</i>	Operand (one or two bytes)
CCR	Condition code register	PC	Program counter
dd	Direct address of operand	PCH	Program counter high byte
dd rr	Direct address of operand and relative offset of branch instruction	PCL	Program counter low byte
DD	Direct to direct addressing mode	REL	Relative addressing mode
DIR	Direct addressing mode	<i>rel</i>	Relative program counter offset byte
DIX+	Direct to indexed with post increment addressing mode	rr	Relative program counter offset byte
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	SP1	Stack pointer, 8-bit offset addressing mode
EXT	Extended addressing mode	SP2	Stack pointer 16-bit offset addressing mode
ff	Offset byte in indexed, 8-bit offset addressing	SP	Stack pointer
H	Half-carry bit	U	Undefined
H	Index register high byte	V	Overflow bit
hh ll	High and low bytes of operand address in extended addressing	X	Index register low byte
I	Interrupt mask	Z	Zero bit
ii	Immediate operand byte	&	Logical AND
IMD	Immediate source to direct destination addressing mode		Logical OR
IMM	Immediate addressing mode	\oplus	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	\leftarrow	Loaded with
IX1+	Indexed, 8-bit offset, post increment addressing mode	?	If
IX2	Indexed, 16-bit offset addressing mode	:	Concatenated with
M	Memory location	\uparrow	Set or cleared
N	Negative bit	—	Not affected

7.8 Opcode Map

See [Table 7-2](#).

Low-Voltage Inhibit (LVI)

The LVI is enabled out of reset. The LVI module contains a bandgap reference circuit and comparator. Clearing the LVI power disable bit (LVIPWRD) enables the LVI to monitor V_{DD} voltage. Clearing the LVI reset disable bit (LVIRSTD) enables the LVI module to generate a reset when V_{DD} falls below a voltage, V_{TRIPF} or V_{DTRIPF} . Setting the LVI enable in stop mode bit (LVISTOP) enables the LVI to operate in stop mode. Setting the LVD or LVR trip point bit (LVDLVR) selects the LVD trip point voltage. The actual trip thresholds are specified in [16.5 DC Electrical Characteristics](#). Either trip level can be used as a detect or reset.

NOTE

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

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

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

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

10.3.1 Polled LVI Operation

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

10.3.2 Forced Reset Operation

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

10.3.3 Voltage Hysteresis Protection

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

10.3.4 LVI Trip Selection

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

13.4.2.3 Illegal Opcode Reset

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

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

13.4.2.4 Illegal Address Reset

An opcode fetch from an unmapped address generates an illegal address reset. The SIM verifies that the CPU is fetching an opcode prior to asserting the ILAD bit in the SIM reset status register (SRSR) and resetting the MCU. A data fetch from an unmapped address does not generate a reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources. See [Figure 2-1. Memory Map](#) for memory ranges.

13.4.2.5 Low-Voltage Inhibit (LVI) Reset

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

13.5 SIM Counter

The SIM counter is used by the power-on reset module (POR) and in stop mode recovery to allow the oscillator time to stabilize before enabling the internal bus (IBUS) clocks. The SIM counter also serves as a prescaler for the computer operating properly module (COP). The SIM counter uses 12 stages for counting, followed by a 13th stage that triggers a reset of SIM counters and supplies the clock for the COP module. The SIM counter is clocked by the falling edge of BUSCLKX4.

13.5.1 SIM Counter During Power-On Reset

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

13.5.2 SIM Counter During Stop Mode Recovery

The SIM counter also is used for stop mode recovery. The STOP instruction clears the SIM counter. After an interrupt, break, or reset, the SIM senses the state of the short stop recovery bit, SSREC, in the configuration register 1 (CONFIG1). If the SSREC bit is a 1, then the stop recovery is reduced from the normal delay of 4096 BUSCLKX4 cycles down to 32 BUSCLKX4 cycles. This is ideal for applications using canned oscillators that do not require long start-up times from stop mode. External crystal applications should use the full stop recovery time, that is, with SSREC cleared in the configuration register 1 (CONFIG1).

13.8.2 Break Flag Control Register

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

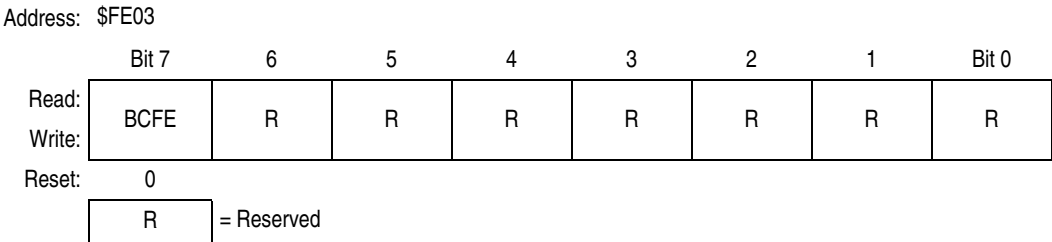


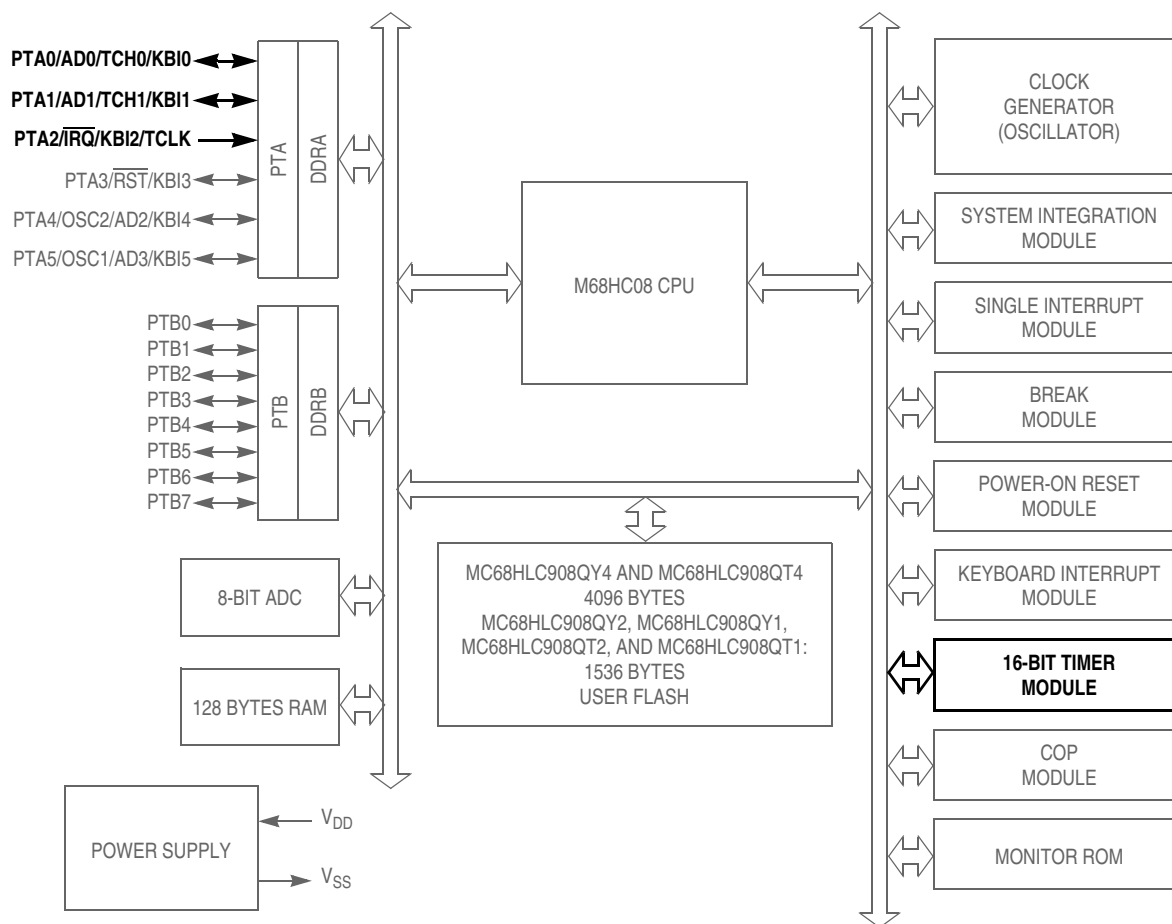
Figure 13-20. Break Flag Control Register (BFCR)

BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

- 1 = Status bits clearable during break
- 0 = Status bits not clearable during break

Timer Interface Module (TIM)



$\overline{\text{RST}}$, $\overline{\text{IRQ}}$: Pins have internal (about 30K Ohms) pull up

PTA[0:5]: High current sink and source capability

PTA[0:5]: Pins have programmable keyboard interrupt and pull up

PTB[0:7]: Not available on 8-pin devices – MC68HLC908QT1, MC68HLC908QT2, and MC68HLC908QT4 (see note in [12.1 Introduction](#))

ADC: Not available on the MC68HLC908QY1 and MC68HLC908QT1

Figure 14-1. Block Diagram Highlighting TIM Block and Pins

Figure 14-2 shows the structure of the TIM. The central component of the TIM is the 16-bit TIM counter that can operate as a free-running counter or a modulo up-counter. The TIM counter provides the timing reference for the input capture and output compare functions. The TIM counter modulo registers, TMODH:TMODL, control the modulo value of the TIM counter. Software can read the TIM counter value at any time without affecting the counting sequence.

[illegible]

Figure 14-2. TIM Block Diagram

14.9.4 TIM Channel Status and Control Registers

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

- Flags input captures and output compares
- Enables input capture and output compare interrupts
- Selects input capture, output compare, or PWM operation
- Selects high, low, or toggling output on output compare
- Selects rising edge, falling edge, or any edge as the active input capture trigger
- Selects output toggling on TIM overflow
- Selects 0% and 100% PWM duty cycle
- Selects buffered or unbuffered output compare/PWM operation

Address:	\$0025 TSC0							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Address:	\$0028 TSC1							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figure 14-7. TIM Channel Status and Control Registers (TSC0:TSC1)

CHxF — Channel x Flag Bit

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

Clear CHxF by reading the TIM channel x status and control register with CHxF set and then writing a 0 to CHxF. If another interrupt request occurs before the clearing sequence is complete, then writing a 0 to CHxF has no effect. Therefore, an interrupt request cannot be lost due to inadvertent clearing of CHxF.

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

- 1 = Input capture or output compare on channel x
- 0 = No input capture or output compare on channel x

CHxIE — Channel x Interrupt Enable Bit

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

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

MSxB — Mode Select Bit B

This read/write bit selects buffered output compare/PWM operation. MSxB exists only in the TIM channel 0 status and control register.

16.11 Timer Interface Module Characteristics

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

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

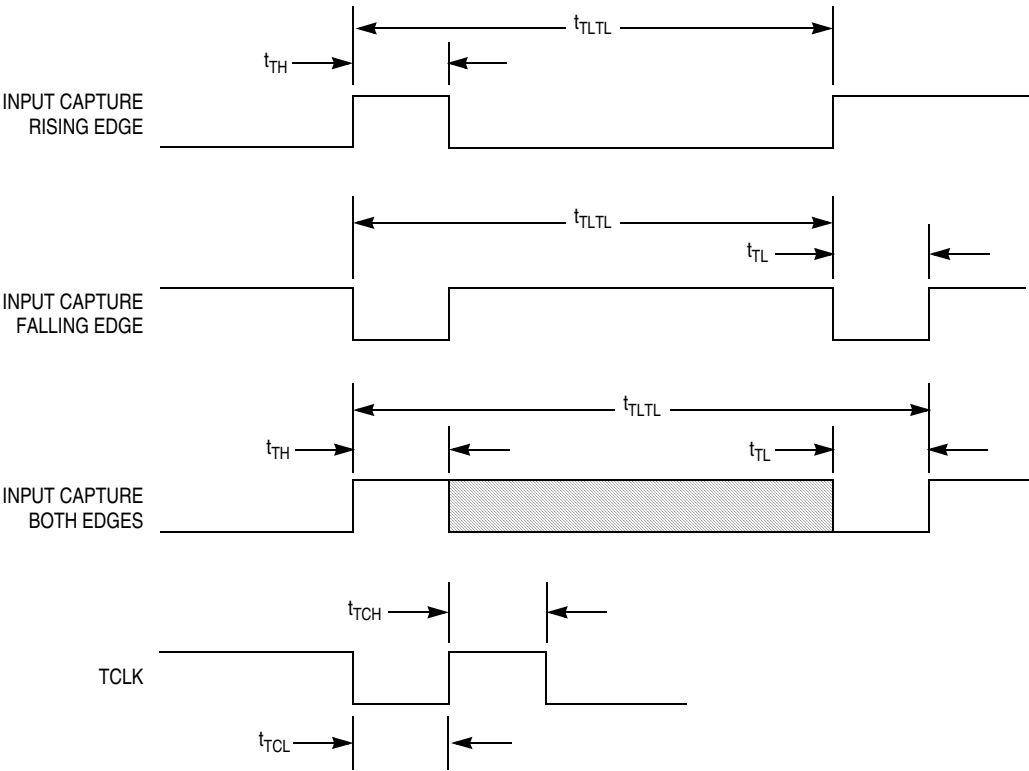
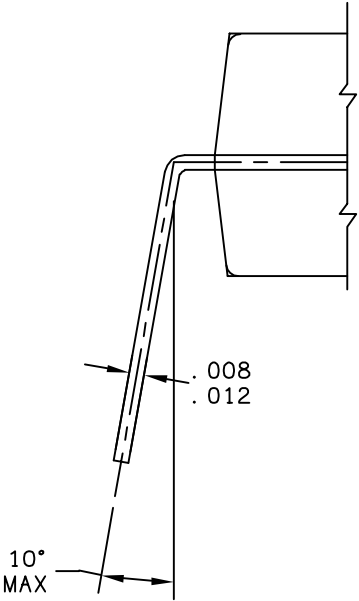


Figure 16-8. Timer Input Timing



DETAIL "D"

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.		MECHANICAL OUTLINE		PRINT VERSION NOT TO SCALE	
TITLE: 8 LD PDIP			DOCUMENT NO: 98ASB42420B		REV: N
			CASE NUMBER: 626-06		19 MAY 2005
			STANDARD: NON-JEDEC		



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.
ELECTRONIC VERSIONS ARE UNCONTROLLED EXCEPT WHEN ACCESSED
DIRECTLY FROM THE DOCUMENT CONTROL REPOSITORY. PRINTED
VERSIONS ARE UNCONTROLLED EXCEPT WHEN STAMPED "CONTROLLED
COPY" IN RED.

MECHANICAL OUTLINES
DICTIONARY

DOCUMENT NO: 98ASB42431B

PAGE: 648

DO NOT SCALE THIS DRAWING

REV: T

NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1994.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION TO CENTER OF LEADS WHEN FORMED PARALLEL.
4. DIMENSIONS DOES NOT INCLUDE MOLD FLASH.
5. ROUNDED CORNERS OPTIONAL.
6. 648-01 THRU -08 OBSOLETE, NEW STANDARD 648-09.

DIM	MILLIMETERS		INCHES		DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX		MIN	MAX	MIN	MAX
A	18.80	19.55	0.740	0.770					
B	6.35	6.85	0.250	0.270					
C	3.69	4.44	0.145	0.175					
D	0.39	0.53	0.015	0.021					
F	1.02	1.77	0.040	0.070					
G	2.54 BSC		0.100 BSC						
H	1.27 BSC		0.050 BSC						
J	0.21	0.38	0.008	0.015					
K	2.80	3.30	0.110	0.130					
L	7.50	7.74	0.295	0.305					
M	0°	10°	0°	10°					
S	0.51	1.01	0.020	0.040					

TITLE:

16 LD PDIP

CASE NUMBER: 648-08

STANDARD: NON-JEDEC

PACKAGE CODE: 0006

SHEET: 2 OF 4

How to Reach Us:

USA/Europe/Locations not listed:

Freescale Semiconductor Literature Distribution
P.O. Box 5405, Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

Japan:

Freescale Semiconductor Japan Ltd.
SPS, Technical Information Center
3-20-1, Minami-Azabu
Minato-ku
Tokyo 106-8573, Japan
81-3-3440-3569

Asia/Pacific:

Freescale Semiconductor H.K. Ltd.
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T. Hong Kong
852-26668334

Learn More:

For more information about Freescale Semiconductor products, please visit
<http://www.freescale.com>

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™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.
© Freescale Semiconductor, Inc. 2004.