

Welcome to [E-XFL.COM](https://www.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](#)"

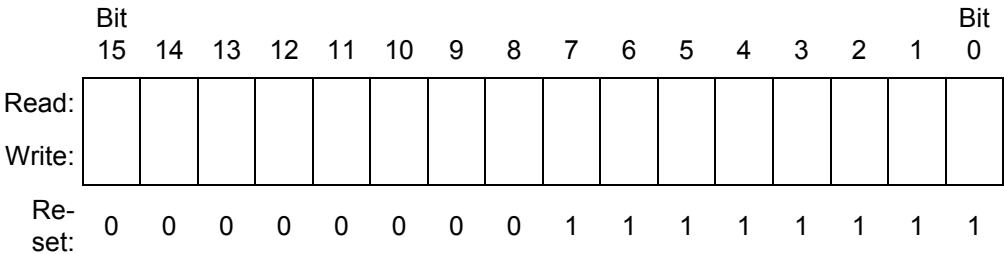
#### Details

Product Status	Active
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	SCI
Peripherals	LVD, POR, PWM
Number of I/O	12
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.600", 15.24mm)
Supplier Device Package	28-PDIP
Purchase URL	<a href="https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mc68hc908mr8cp">https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mc68hc908mr8cp</a>

## 6.4.3 Stack Pointer

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

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



**Figure 6-4. Stack Pointer (SP)**

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

SBSW can be read within the break state SWI routine. The user can modify the return address on the stack by subtracting one from it. The following code is an example of this. Writing 0 to the SBSW bit clears it.

```
; This code works if the H register has been pushed onto the stack in the break
; service routine software. This code should be executed at the end of the break
; service routine software.

HIBYTE    EQU    5
LOBYTE    EQU    6
;          If not SBSW, do RTI

          BRCLR   SBSW,SBSR, RETURN    ; See if wait mode was exited by break.
                                           ;

          TST     LOBYTE,SP            ; If RETURNLO is not zero,
          BNE     DOLO                  ; then just decrement low byte.
          DEC     HIBYTE,SP             ; Else deal with high byte, too.
DOLO      DEC     LOBYTE,SP            ; Point to WAIT opcode.
RETURN    PULH     RTI                 ; Restore H register.
```

## 8.4.2.2 Acquisition and Tracking Modes

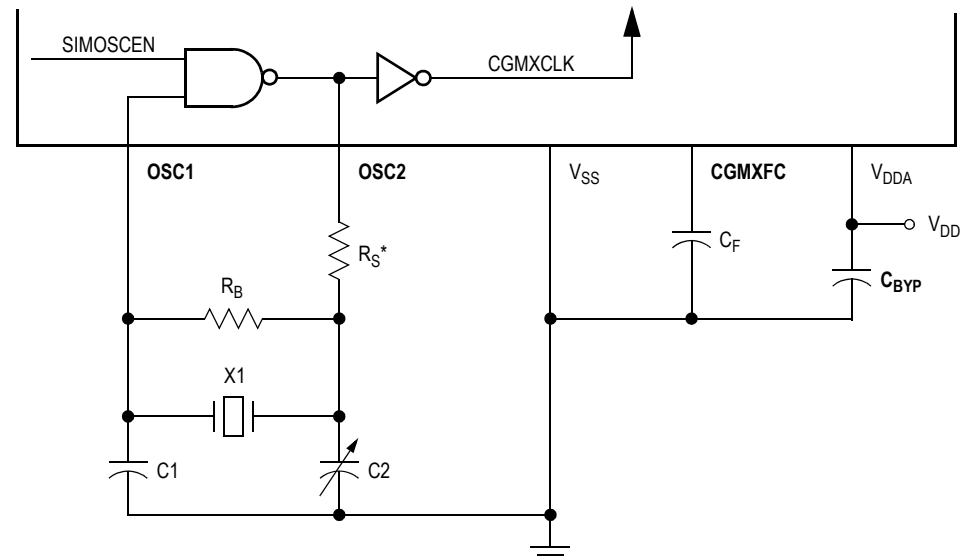
The PLL filter is manually or automatically configurable into one of two operating modes:

- Acquisition mode — In acquisition mode, the filter can make large frequency corrections to the VCO. This mode is used at PLL startup or when the PLL has suffered a severe noise hit and the VCO frequency is far off the desired frequency. When in acquisition mode, the ACQ bit is clear in the PLL bandwidth control register. See [8.6.2 PLL Bandwidth Control Register](#).
- Tracking mode — In tracking mode, the filter makes only small corrections to the frequency of the VCO. PLL jitter is much lower in tracking mode, but the response to noise is also slower. The PLL enters tracking mode when the VCO frequency is nearly correct, such as when the PLL is selected as the base clock source. The PLL is automatically in tracking mode when not in acquisition mode or when the ACQ bit is set. See [8.4.3 Base Clock Selector Circuit](#).

## 8.4.2.3 Manual and Automatic PLL Bandwidth Modes

The PLL can change the bandwidth or operational mode of the loop filter manually or automatically.

In automatic bandwidth control mode (AUTO = 1), the lock detector automatically switches between acquisition and tracking modes. Automatic bandwidth control mode also is used to determine when the VCO clock, CGMVCLK, is safe to use as the source for the base clock, CGMOUT ([8.6.2 PLL Bandwidth Control Register](#)). If PLL interrupts are enabled, the software can wait for a PLL interrupt request and then check the LOCK bit. If interrupts are disabled, software can poll the LOCK bit continuously (during PLL startup, usually) or at periodic intervals. In either case, when the LOCK bit is set, the VCO clock is safe to use as the source for the base clock (see [8.4.3 Base Clock Selector Circuit](#)). If the VCO is selected as the source for the base clock and the LOCK bit is clear, the PLL has suffered a severe noise hit and the software must take appropriate action, depending on the application. See [8.7 Interrupts](#) for information and precautions on using interrupts.



\* $R_S$  can be 0 (shorted) when used with higher-frequency crystals.  
Refer to manufacturer's data.

**Figure 8-3. CGM External Connections**

## 8.5 I/O Signals

The following paragraphs describe the CGM input/output (I/O) signals.

### 8.5.1 Crystal Amplifier Input Pin (OSC1)

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

### 8.5.2 Crystal Amplifier Output Pin (OSC2)

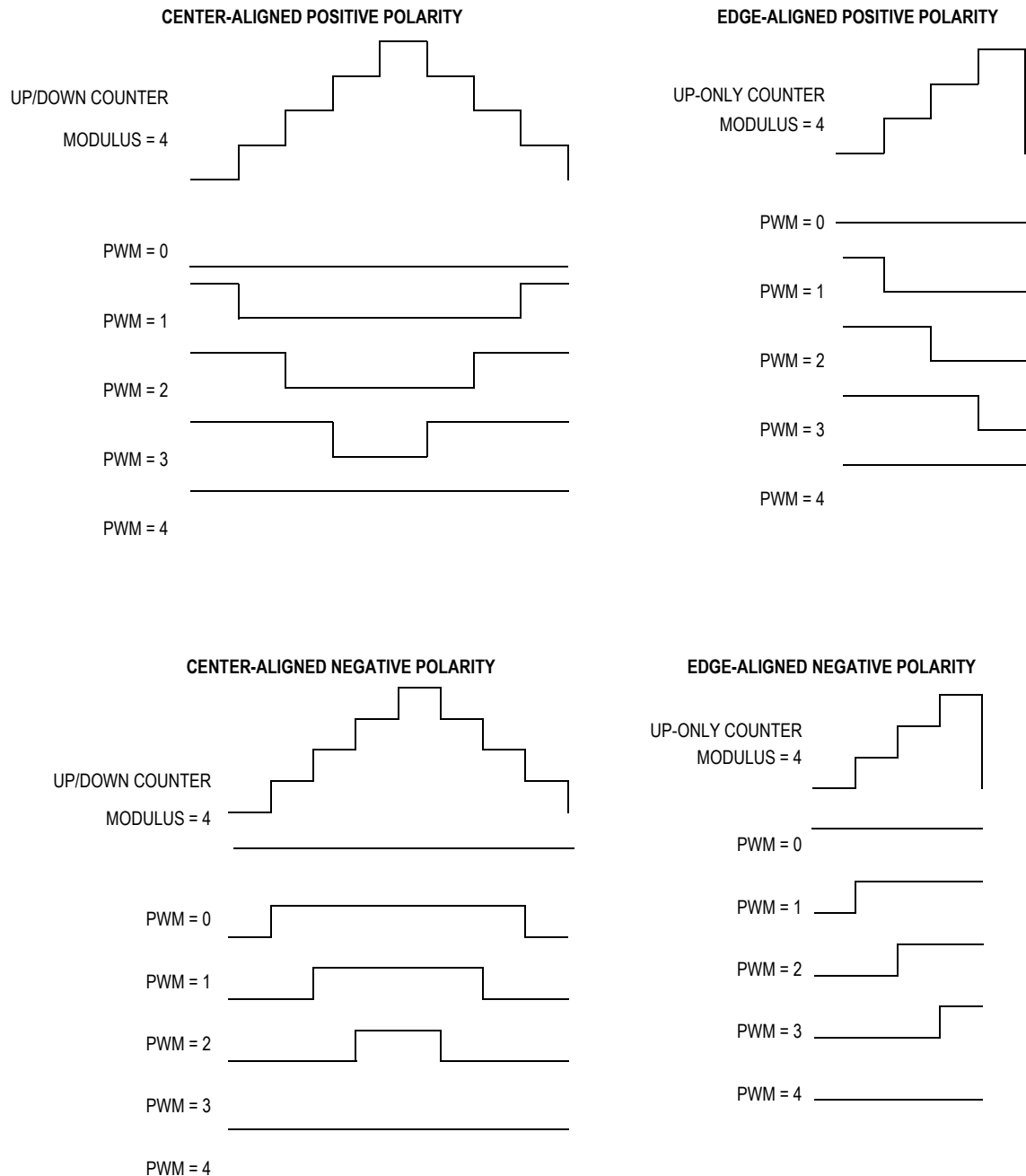
The OSC2 pin is the output of the crystal oscillator inverting amplifier.

### 8.5.3 External Filter Capacitor Pin (CGMXFC)

The CGMXFC pin is required by the loop filter to filter out phase corrections. A small external capacitor is connected to this pin.

**NOTE:** To prevent noise problems,  $C_F$  should be placed as close to the CGMXFC pin as possible, with minimum routing distances and no routing of other signals across the  $C_F$  connection.

# Pulse-Width Modulator for Motor Control



**Figure 9-17. PWM Polarity**

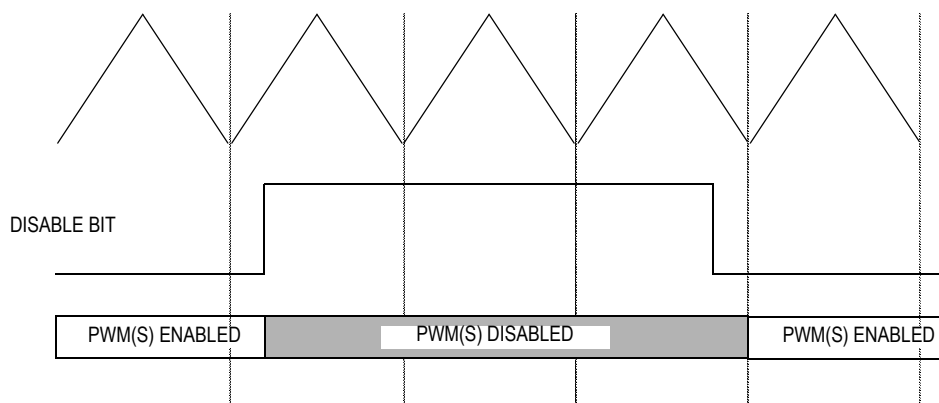
The function of the fault control and event bits is the same as in automatic mode except that the PWMs are not re-enabled until the FFLAGx event bit is cleared by writing to the FTACKx bit and the filtered fault condition is cleared (logic low).

## 9.7.2 Software Output Disable

Setting PWM disable bit DISX or DISY in PWM control register 1 immediately disables the corresponding PWM pins as determined by the bank and disable mapping register. The PWM pin(s) remain disabled until the PWM disable bit is cleared and a new PWM cycle begins as shown in [Figure 9-27](#). Setting a PWM disable bit does not latch a CPU interrupt request, and there are no event flags associated with the PWM disable bits.

## 9.7.3 Output Port Control

When operating the PWMs using the OUTx bits (OUTCTL = 1), fault protection applies as described in this section. Due to the absence of periodic PWM cycles, fault conditions are cleared upon each CPU cycle and the PWM outputs are re-enabled, provided all fault clearing conditions are satisfied.



**Figure 9-27. PWM Software Disable**

**NOTE:** When PWMINT is cleared, pending CPU interrupts are inhibited.

## PWMF— PWM Reload Flag

This read/write bit is set at the beginning of every reload cycle regardless of the state of the LDOK bit. This bit is cleared by reading PWM control register 1 with the PWMF flag set, then writing a logic 0 to PWMF. If another reload occurs before the clearing sequence is complete, then writing logic 0 to PWMF has no effect.

1 = New reload cycle began.

0 = New reload cycle has not begun.

**NOTE:** When PWMF is cleared, pending PWM CPU interrupts are cleared (excluding fault interrupts).

**CAUTION:** Bits 2 and/or 3 of PCTL1 are reserved and must never be set to a 1. Setting these bits to a 1 will affect the active PWM value registers. Undesirable results will occur.

## LDOK— Load OK Bit

This read/write bit allows the counter modulus, counter prescaler, and PWM values in the buffered registers to be used by the PWM generator. These values will not be used until the LDOK bit is set and a new PWM load cycle begins. LDOK may be cleared, if it is set, by writing a logic 0 to it prior to the beginning of a new PWM load cycle. Internally this bit is automatically cleared after the new values are loaded.

1 = Okay to load new modulus, prescaler, and PWM values at beginning of next PWM load cycle

0 = Not okay to load new modulus, prescaler, and PWM values

**NOTE:** The user should initialize the PWM registers and set the LDOK bit before enabling the PWM.

## PWMEN — PWM Module Enable Bit

This read/write bit enables and disables the PWM generator and the PWM pins. When PWMEN is clear, the PWM generator is disabled and the PWM pins are in the high-impedance state (unless OUTCTL = 1). When the PWMEN bit is set, the PWM generator and PWM pins are activated. For more information, see [9.8 Initialization and the PWMEN Bit](#).

1 = PWM generator and PWM pins enabled

0 = PWM generator and PWM pins disabled

Setting MS0B links channels 0 and 1 and configures them for buffered PWM operation. The TIMA channel 0 registers (TACH0H–TACH0L) initially control the buffered PWM output. TIMA status control register 0 (TASC0) controls and monitors the PWM signal from the linked channels. MS0B takes priority over MS0A.

Clearing the toggle-on-overflow bit, TOVx, inhibits output toggles on TIMA overflows. Subsequent output compares try to force the output to a state it is already in and have no effect. The result is a 0 percent duty cycle output.

Setting the channel x maximum duty cycle bit (CHxMAX) and setting the TOVx bit generates a 100 percent duty cycle output. See [11.10.4 TIMA Channel Status and Control Registers](#).

## 11.5 Interrupts

These TIMA sources can generate interrupt requests:

- TIMA overflow flag (TOF) — The TOF bit is set when the TIMA counter reaches the modulo value programmed in the TIMA counter modulo registers. The TIMA overflow interrupt enable bit, TOIE, enables TIMA overflow CPU interrupt requests. TOF and TOIE are in the TIMA status and control register.
- TIMA channel flags (CH1F–CH0F) — The CHxF bit is set when an input capture or output compare occurs on channel x. Channel x TIMA CPU interrupt requests are controlled by the channel x interrupt enable bit, CHxIE.

## 11.6 Wait Mode

The WAIT instruction puts the MCU in low-power standby mode.

The TIMA remains active after the execution of a WAIT instruction. In wait mode, the TIMA registers are not accessible by the CPU. Any enabled CPU interrupt request from the TIMA can bring the MCU out of wait mode.

# Timer Interface A (TIMA)

## 11.10.1 TIMA Status and Control Register

The TIMA status and control register:

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

Address \$000E  
:

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

R = Reserved

**Figure 11-4. TIMA Status and Control Register (TASC)**

### TOF — TIMA Overflow Flag Bit

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

- 1 = TIMA counter has reached modulo value.
- 0 = TIMA counter has not reached modulo value.

### TOIE — TIMA Overflow Interrupt Enable Bit

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

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

## 11.10.4 TIMA Channel Status and Control Registers

Each of the TIMA channel status and control registers:

- 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 TIMA overflow
- Selects 0% and 100% PWM duty cycle
- Selects buffered or unbuffered output compare/PWM operation

Register Name and Address: TASC0 — \$0013

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MA X
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Register Name and Address: TASC1 — \$0016

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MA X
Write:	0		R					
Reset:	0	0	0	0	0	0	0	0

R = Reserved

**Figure 11-7. TIMA Channel Status and Control Registers (TASC0–TASC1)**

**NOTE:** Before enabling a TIMA channel register for input capture operation, make sure that the PTBx/TACHx pin is stable for at least two bus clocks.

#### TOVx — Toggle-On-Overflow Bit

When channel x is an output compare channel, this read/write bit controls the behavior of the channel x output when the TIMA counter overflows. When channel x is an input capture channel, TOVx has no effect. Reset clears the TOVx bit.

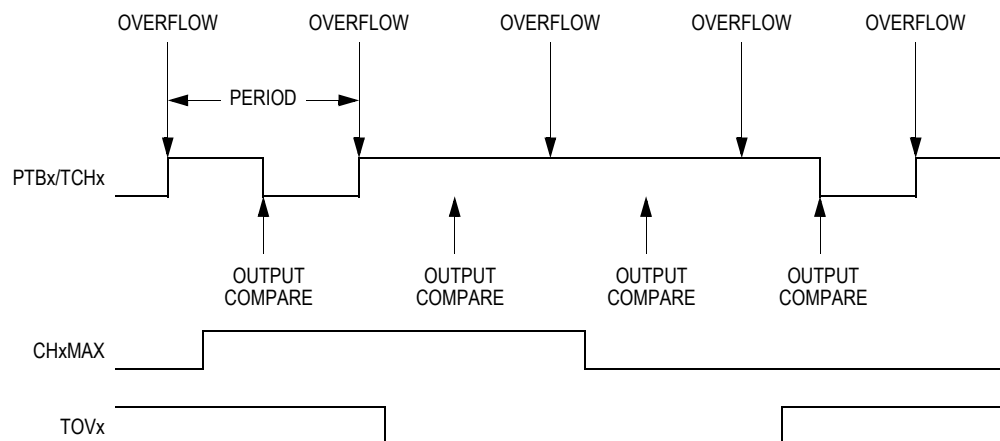
1 = Channel x pin toggles on TIMA counter overflow.

0 = Channel x pin does not toggle on TIMA counter overflow.

**NOTE:** When TOVx is set, a TIMA 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 percent. As [Figure 11-8](#) shows, the CHxMAX bit takes effect in the cycle after it is set or cleared. The output stays at the 100 percent duty cycle level until the cycle after CHxMAX is cleared.



**Figure 11-8. CHxMAX Latency**

# Timer Interface B (TIMB)

## 12.10.2 TIMB Counter Registers

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

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

Register Name and Address: TBCNTH — \$0052

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

Register Name and Address: TBCNTL — \$0053

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

R = Reserved

**Figure 12-5. TIMB Counter Registers (TBCNTH and TBCNTL)**

When CHxIE = 1, clear CHxF by reading TIMB channel x status and control register with CHxF set, and then writing a logic 0 to CHxF. If another interrupt request occurs before the clearing sequence is complete, then writing logic 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 logic 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 TIMB CPU interrupts 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 TIMB channel 0. Reset clears the MSxB bit.

1 = Buffered output compare/PWM operation enabled

0 = Buffered output compare/PWM operation disabled

#### MSxA — Mode Select Bit A

When ELSxB:A ≠ 00, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation. See [Table 12-2](#).

1 = Unbuffered output compare/PWM operation

0 = Input capture operation

When ELSxB:A = 00, this read/write bit selects the initial output level of the TCHx pin once PWM, input capture, or output compare operation is enabled. Reset clears the MSxA bit. See [Table 12-2](#).

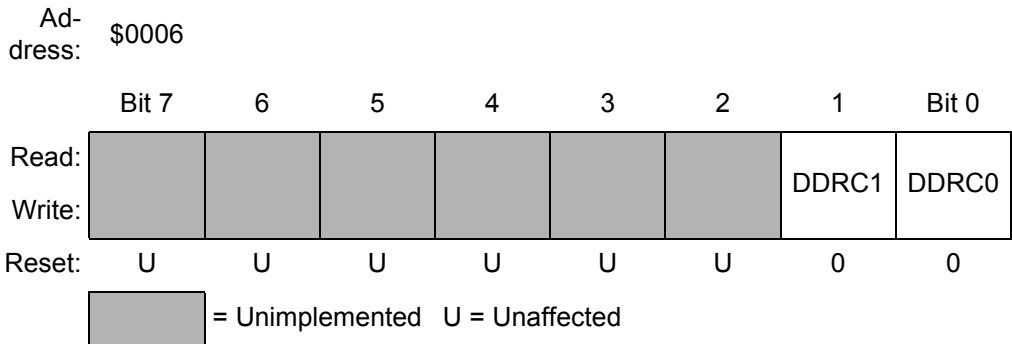
1 = Initial output level low

0 = Initial output level high

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

## 14.5.2 Data Direction Register C

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



**Figure 14-9. Data Direction Register C (DDRC)**

### DDRC[1:0] — Data Direction Register C Bits

These read/write bits control port C data direction. Reset clears DDRC[1:0], configuring all port C pins as inputs.

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

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

**Figure 14-10** shows the port C I/O logic.

## AIEN — ADC Interrupt Enable Bit

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

- 1 = ADC interrupt enabled
- 0 = ADC interrupt disabled

## ADCO — ADC Continuous Conversion Bit

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

- 1 = Continuous ADC conversion
- 0 = One ADC conversion

## ADCH[4:0] — ADC Channel Select Bits

ADCH4, ADCH3, ADCH2, ADCH1, and ADCH0 form a 5-bit field which is used to select one of the ADC channels detailed in [Table 18-1](#). Take care to prevent switching noise from corrupting the analog signal when simultaneously using a port pin as both an analog and digital input.

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

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

The voltage levels supplied from internal reference nodes as specified in [Table 18-1](#) are used to verify the operation of the ADC converter both in production test and for user applications.

# Analog-to-Digital Converter (ADC)

In 8-bit mode, this 8-bit result register holds the eight MSBs of the 10-bit result. This register is updated each time an ADC conversion completes. In 8-bit mode, this register contains no interlocking with ADRH.

Ad- dress:	\$0042							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Write:	R	R	R	R	R	R	R	R
Reset:	Unaffected by Reset							
	<div>R</div> = Reserved							

**Figure 18-8. ADC Data Register Low (ADRL)  
8-Bit Mode**

## 18.9.4 ADC Clock Register

This register selects the clock frequency for the ADC, selecting between modes of operation.

Ad- dress:	\$0043							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ADIV2	ADIV1	ADIV0	ADICLK	MODE1	MODE0	0	0
Write:							0	R
Reset:	0	0	0	0	0	1	0	0
	<div>R</div> = Reserved							

**Figure 18-9. ADC Clock Register (ADCLK)**

### ADIV2:ADIV0 — ADC Clock Prescaler Bits

ADIV2, ADIV1, and ADIV0 form a 3-bit field which selects the divide ratio used by the ADC to generate the internal ADC clock. [Table 18-2](#) shows the available clock configurations. The ADC clock should be set to between 500 kHz and 1 MHz.

## 21.4 Functional Operating Range

Characteristic	Symbol	Value	Unit
Operating temperature range <sup>(1)</sup> MC68HC908MR8CP MC68HC908MR8CFA MC68HC908MR8CDW MC68HC908MR8VFA MC68HC908MR8VP MC68HC908MR8VDW MC68HC908MR8MFA MC68HC908MR8MP MC68HC908MR8MDW	$T_A$	–40 to +85 –40 to +85 –40 to +85 –40 to +105 –40 to +105 –40 to +105 –40 to +105 –40 to +125 –40 to +125 –40 to +125	°C
Operating voltage range	$V_{DD}$	5.0 ± 10%	V

1. Contact a Freescale representative for temperature availability.  
 C = Extended temperature range (–40 to +85°C)  
 V = Industrial temperature range (–40 to +105°C)  
 M = Automotive temperature range (–40 to +125°C)

**register** — A circuit that stores a group of bits.

**reserved memory location** — A memory location that is used only in special factory test modes. Writing to a reserved location has no effect. Reading a reserved location returns an unpredictable value.

**reset** — To force a device to a known condition.

**ROM** — Read-only memory. A type of memory that can be read but cannot be changed (written). The contents of ROM must be specified before manufacturing the MCU.

**SCI** — See serial communication interface module (SCI).

**serial** — Pertaining to sequential transmission over a single line.

**serial communication interface module (SCI)** — A module in the M68HC08 Family that supports asynchronous communication.

**serial peripheral interface module (SPI)** — A module in the M68HC08 Family that supports synchronous communication.

**set** — To change a bit from logic 0 to logic 1; opposite of clear.

**shift register** — A chain of circuits that can retain the logic levels (logic 1 or logic 0) written to them and that can shift the logic levels to the right or left through adjacent circuits in the chain.

**signed** — A binary number notation that accommodates both positive and negative numbers. The most significant bit is used to indicate whether the number is positive or negative, normally logic 0 for positive and logic 1 for negative. The other seven bits indicate the magnitude of the number.

**SIM** — See system integration module (SIM).

**software** — Instructions and data that control the operation of a microcontroller.

**software interrupt (SWI)** — An instruction that causes an interrupt and its associated vector fetch.

**SPI** — See serial peripheral interface module (SPI).

**stack** — A portion of RAM reserved for storage of CPU register contents and subroutine return addresses.

**stack pointer (SP)** — A 16-bit register in the CPU08 containing the address of the next available storage location on the stack.

**start bit** — A bit that signals the beginning of an asynchronous serial transmission.

**variable** — A value that changes during the course of program execution.

**VCO** — See voltage-controlled oscillator.

**vector** — A memory location that contains the address of the beginning of a subroutine written to service an interrupt or reset.

**voltage-controlled oscillator (VCO)** — A circuit that produces an oscillating output signal of a frequency that is controlled by a dc voltage applied to a control input.

**waveform** — A graphical representation in which the amplitude of a wave is plotted against time.

**wired-OR** — Connection of circuit outputs so that if any output is high, the connection point is high.

**word** — A set of two bytes (16 bits).

**write** — The transfer of a byte of data from the CPU to a memory location.

**X** — The lower byte of the index register (H:X) in the CPU08.

**Z** — The zero bit in the condition code register of the CPU08. The CPU08 sets the zero bit when an arithmetic operation, logical operation, or data manipulation produces a result of \$00.

