



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	8MHz
Connectivity	LINbus, SCI, SPI
Peripherals	POR, PWM
Number of I/O	24
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908ey16avfje

MC68HC908EY16A MC68HC908EY8A

Data Sheet

***M68HC08
Microcontrollers***

MC68HC908EY16A
Rev. 2
09/2010

freescale.com

Table of Contents

Chapter 1 General Description

1.1	Introduction	19
1.2	Features	19
1.3	MCU Block Diagram	20
1.4	Pin Assignments	22
1.5	Pin Functions	22
1.5.1	Power Supply Pins (V_{DD} and V_{SS})	22
1.5.2	Oscillator Pins (PTC4/OSC1 and PTC3/OSC2)	23
1.5.3	External Reset Pin (\overline{RST})	23
1.5.4	External Interrupt Pin (\overline{IRQ})	23
1.5.5	Analog Power Supply/Reference Pins (V_{DDA} , V_{REFH} , V_{SSA} , and V_{REFL})	23
1.5.6	Port A I/O Pins (PTA6/ \overline{SS} , PTA5/SPSCK, PTA4/KBD4, PTA3/KBD3/RxD, PTA2/KBD2/TxD, PTA1/KBD1, and PTA0/KBD0)	24
1.5.7	Port B I/O Pins (PTB7/AD7/TBCH1, PTB6/AD6/TBCH0, PTB5/AD5/SPSCK, PTB4/AD4/MO-SI, PTB3/AD3/MISO, PTB2/AD2–PTB0/AD0)	24
1.5.8	Port C I/O Pins (PTC4/OSC1, PTC3/OSC2, PTC2/MCLK/ \overline{SS} , PTC1/MOSI, PTC0/MISO)	24
1.5.9	Port D I/O Pins (PTD1/TACH1–PTD0/TACH0)	24
1.5.10	Port E I/O Pins (PTE1/RxD–PTE0/TxD)	24
1.6	Pin Summary	25
1.7	Priority of Shared Pins	27

Chapter 2 Memory

2.1	Introduction	29
2.2	Unimplemented Memory Locations	29
2.3	Reserved Memory Locations	29
2.4	Input/Output (I/O) Section	29
2.5	Random Access Memory (RAM)	39
2.6	FLASH Memory (FLASH)	39
2.6.1	FLASH Control Register	40
2.6.2	FLASH Page Erase Operation	41
2.6.3	FLASH Mass Erase Operation	42
2.6.4	FLASH Program/Read Operation	43
2.6.5	FLASH Block Protection	45
2.6.6	FLASH Block Protect Register	45
2.6.7	Wait Mode	46
2.6.8	Stop Mode	46

Chapter 3

Chapter 2

Memory

2.1 Introduction

The M68HC08 central processor unit (CPU08) can address 64 Kbytes of memory space. The memory map, shown in [Figure 2-1](#), includes:

- 16 Kbytes of FLASH memory, 15,872 bytes of user space
- 512 bytes of random-access memory (RAM)
- 36 bytes of user-defined vectors
- 350 bytes of monitor routines in read-only memory (ROM)
- 674 bytes of integrated FLASH burn-in routines in ROM

2.2 Unimplemented Memory Locations

Accessing an unimplemented location can cause an illegal address reset. In the memory map ([Figure 2-1](#)) and in register figures in this document, unimplemented locations are shaded.

2.3 Reserved Memory Locations

Accessing a reserved location can have unpredictable effects on microcontroller unit (MCU) operation. In the [Figure 2-1](#) and in register figures in this document, reserved locations are marked with the word reserved or with the letter R.

2.4 Input/Output (I/O) Section

Most of the control, status, and data registers are in the zero page area of \$0000–\$003F. Additional I/O registers have these addresses:

- \$FE00; SIM break status register, SBSR
- \$FE01; SIM reset status register, SRSR
- \$FE03; SIM break flag control register, SBFCR
- \$FE04; interrupt status register 1, INT1
- \$FE05; interrupt status register 2, INT2
- \$FE06; interrupt status register 3, INT3
- \$FE08; FLASH control register, FLCR
- \$FE09; break address register high, BRKH
- \$FE0A; break address register low, BRKL
- \$FE0B; break status and control register, BRKSCR
- \$FE0C; LVI status register, LVISR
- \$FF7E; FLASH block protect register, FLBPR
- \$FF80; 5V internal oscillator trim value (optional), ICGT5V
- \$FF81; 3V internal oscillator trim value (optional), ICGT3V

Table 2-2. Examples of Protect Address Ranges

BPR[7:0]	Addresses of Protect Range
\$00	The entire FLASH memory is protected.
\$01 (0000 0001)	\$C040 (1100 0000 0100 0000) — \$FFFF
\$02 (0000 0010)	\$C080 (1100 0000 1000 0000) — \$FFFF
\$03 (0000 0011)	\$C0C0 (1100 0000 1100 0000) — \$FFFF
\$04 (0000 0100)	\$C100 (1100 0001 0000 0000) — \$FFFF
and so on...	
\$FC (1111 1100)	\$FF00 (1111 1111 0000 0000) — FFFF
\$FD (1111 1101)	\$FF40 (1111 1111 0100 0000) — \$FFFF FLBPR and vectors are protected
\$FE (1111 1110)	\$FF80 (1111 1111 1000 0000) — FFFF Vectors are protected
\$FF	The entire FLASH memory is not protected.

2.6.7 Wait Mode

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

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

2.6.8 Stop Mode

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

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

NOTE

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

When a conversion is aborted, the contents of the data registers, ADRH and ADRL, are not altered but continue to be the values transferred after the completion of the last successful conversion. In the case that the conversion was aborted by a reset, ADRH and ADRL return to their reset states.

Upon reset or when a conversion is otherwise aborted, the ADC10 module will enter a low power, inactive state. In this state, all internal clocks and references are disabled. This state is entered asynchronously and immediately upon aborting of a conversion.

3.3.3.4 Total Conversion Time

The total conversion time depends on many factors such as sample time, bus frequency, whether ACLKEN is set, and synchronization time. The total conversion time is summarized in [Table 3-1](#).

Table 3-1. Total Conversion Time versus Control Conditions

Conversion Mode	ACLKEN	Maximum Conversion Time
8-Bit Mode (short sample — ADLSMP = 0):		
Single or 1st continuous	0	18 ADCK + 3 bus clock
Single or 1st continuous	1	18 ADCK + 3 bus clock + 5 μ s
Subsequent continuous ($f_{Bus} \geq f_{ADCK}$)	x	16 ADCK
8-Bit Mode (long sample — ADLSMP = 1):		
Single or 1st continuous	0	38 ADCK + 3 bus clock
Single or 1st continuous	1	38 ADCK + 3 bus clock + 5 μ s
Subsequent continuous ($f_{Bus} \geq f_{ADCK}$)	x	36 ADCK
10-Bit Mode (short sample — ADLSMP = 0):		
Single or 1st continuous	0	21 ADCK + 3 bus clock
Single or 1st continuous	1	21 ADCK + 3 bus clock + 5 μ s
Subsequent continuous ($f_{Bus} \geq f_{ADCK}$)	x	19 ADCK
10-Bit Mode (long sample — ADLSMP = 1):		
Single or 1st continuous	0	41 ADCK + 3 bus clock
Single or 1st continuous	1	41 ADCK + 3 bus clock + 5 μ s
Subsequent continuous ($f_{Bus} \geq f_{ADCK}$)	x	39 ADCK

The maximum total conversion time for a single conversion or the first conversion in continuous conversion mode is determined by the clock source chosen and the divide ratio selected. The clock source is selectable by the ADICLK and ACLKEN bits, and the divide ratio is specified by the ADIV bits. For example, if the alternate clock source is 16 MHz and is selected as the input clock source, the input clock divide-by-8 ratio is selected and the bus frequency is 4 MHz, then the conversion time for a single 10-bit conversion is:

$$\text{Maximum Conversion time} = \frac{21 \text{ ADCK cycles}}{16 \text{ MHz}/8} + \frac{3 \text{ bus cycles}}{4 \text{ MHz}} = 11.25 \mu\text{s}$$

$$\text{Number of bus cycles} = 11.25 \mu\text{s} \times 4 \text{ MHz} = 45 \text{ cycles}$$

NOTE

The ADCK frequency must be between f_{ADCK} minimum and f_{ADCK} maximum to meet A/D specifications.

3.4 Interrupts

When AIEN is set, the ADC10 is capable of generating a CPU interrupt after each conversion. A CPU interrupt is generated when the conversion completes (indicated by COCO being set). COCO will set at the end of a conversion regardless of the state of AIEN.

3.5 Low-Power Modes

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

3.5.1 Wait Mode

The ADC10 will continue the conversion process and will generate an interrupt following a conversion if AIEN is set. If the ADC10 is not required to bring the MCU out of wait mode, ensure that the ADC10 is not in continuous conversion mode by clearing ADCO in the ADC10 status and control register before executing the WAIT instruction. In single conversion mode the ADC10 automatically enters a low-power state when the conversion is complete. It is not necessary to set the channel select bits (ADCH[4:0]) to all 1s to enter a low power state.

3.5.2 Stop Mode

If ACLKEN is clear, executing a STOP instruction will abort the current conversion and place the ADC10 in a low-power state. Upon return from stop mode, a write to ADCSC is required to resume conversions, and the result stored in ADRH and ADRL will represent the last completed conversion until the new conversion completes.

If ACLKEN is set, the ADC10 continues normal operation during stop mode. The ADC10 will continue the conversion process and will generate an interrupt following a conversion if AIEN is set. If the ADC10 is not required to bring the MCU out of stop mode, ensure that the ADC10 is not in continuous conversion mode by clearing ADCO in the ADC10 status and control register before executing the STOP instruction. In single conversion mode the ADC10 automatically enters a low-power state when the conversion is complete. It is not necessary to set the channel select bits (ADCH[4:0]) to all 1s to enter a low-power state.

If ACLKEN is set, a conversion can be initiated while in stop using the external hardware trigger ADEXTCO when in external convert mode. The ADC10 will operate in a low-power mode until the trigger is asserted, at which point it will perform a conversion and assert the interrupt when complete (if AIEN is set).

3.6 ADC10 During Break Interrupts

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. BCFE in the break flag control register (BFCR) enables software to clear status bits during the break state. See BFCR in the SIM section of this data sheet.

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

To protect status bits during the break state, write a 0 to BCFE. With BCFE cleared (its default state), software can read and write registers during the break state without affecting status bits. Some status bits have a two-step read/write clearing procedure. If software does the first step on such a bit before the

Chapter 4

BEMF Counter Module (BEMF)

4.1 Introduction

This section describes the BEMF module. The BEMF counter integrates over time, while the PTD0/TACH0 pin is active. This function is useful for measuring recirculation currents in motors occurring on switching of inductive loads.

BEMF is the abbreviation for **B**ack **E**lectro**M**agnetic **F**orce.

4.2 Functional Description

The 8-bit BEMF counter runs at the internal bus frequency divided by 64. Whenever PTD0/TACH0 is a logic 1, the counter increments by 1 with each period.

4.3 BEMF Register

The BEMF register contains the eight read-only bits of the BEMF counter, showing its actual value. A read access to the BEMF register resets all counter bits to 0.

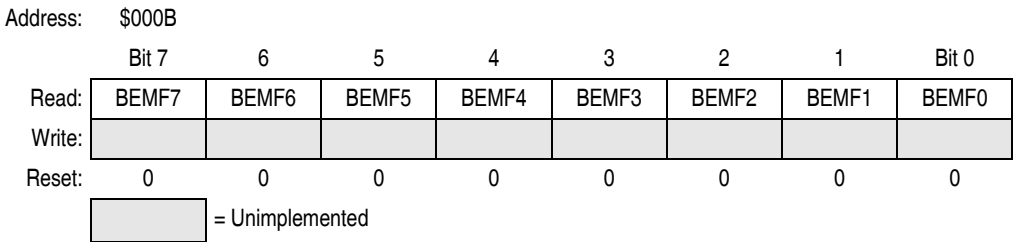


Figure 4-1. BEMF Register (BEMF)

4.4 Input Signal

Port D shares the PTD0/TACH0 pin with the BEMF module. To measure an external signal with the BEMF module, PTD0/TACH0 must be configured as an input (DDRD0 = 0).

4.5 Low Power Modes

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

4.5.1 Wait Mode

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

Address:	\$001F							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:								
Write:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3 ⁽¹⁾	SSREC	STOP	COPD
Reset:	0	0	0	0	0	0	0	0

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

Figure 5-1. Configuration Register 1 (CONFIG1)

COPRS — COP Rate Select Bit

COPRS selects the COP timeout period. Reset clears COPRS. See [Chapter 6 Computer Operating Properly](#).

1 = COP timeout period = 8176 CGMXCLK cycles

0 = COP timeout period = 262,128 CGMXCLK cycles

LVISTOP — LVI Enable in Stop Mode Bit

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

1 = LVI enabled during stop mode

0 = LVI disabled during stop mode

LVIRSTD — LVI Reset Disable Bit

LVIRSTD disables the reset signal from the LVI module. See [Chapter 11 Low-Voltage Inhibit \(LVI\) Module](#).

1 = LVI module resets disabled

0 = LVI module resets enabled

LVIPWRD — LVI Power Disable Bit

LVIPWRD disables the LVI module. See [Chapter 11 Low-Voltage Inhibit \(LVI\) Module](#).

1 = LVI module power disabled

0 = LVI module power enabled

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

LVI5OR3 selects the voltage operating mode of the LVI module. See [Chapter 11 Low-Voltage Inhibit \(LVI\) Module](#). The voltage mode selected for the LVI will typically be 5 V. However, users may choose to operate the LVI in 3-V mode if desired. See [Chapter 20 Electrical Specifications](#) for the LVI's voltage trip points for each of the modes.

1 = LVI operates in 5-V mode.

0 = LVI operates in 3-V mode.

NOTE

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

SSREC — Short Stop Recovery Bit

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

1 = Stop mode recovery after 32 CGMXCLK cycles

0 = Stop mode recovery after 4096 CGMXCLK cycles

NOTE

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

OSCENINSTOP — Oscillator Enable In Stop Mode Bit

OSCENINSTOP, when set, will enable the internal clock generator module to continue to generate clocks (either internal, ICLK, or external, ECLK) in stop mode. See [Chapter 8 Internal Clock Generator \(ICG\) Module](#). This function is used to keep the timebase running while the rest of the microcontroller stops. When clear, all clock generation will cease and both ICLK and ECLK will be forced low during stop mode. The default state for this option is clear, disabling the ICG in stop mode.

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

SSBPUENB — \overline{SS} Pullup Enable Bit

Clearing SSBPUENB enables the \overline{SS} pullup resistor.

- 1 = Disables \overline{SS} pullup resistor.
- 0 = Enables \overline{SS} pullup resistor.

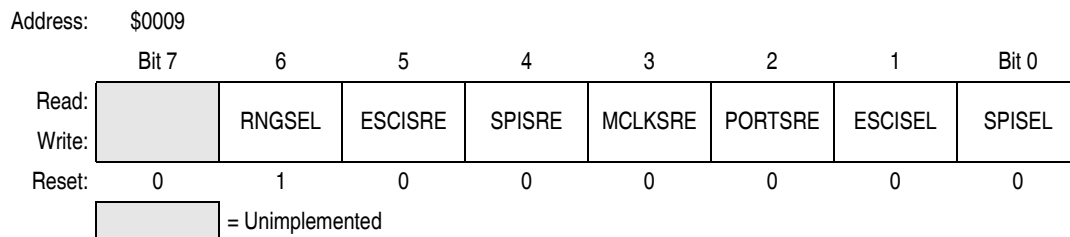


Figure 5-3. Configuration Register 3 (CONFIG3)

RNGSEL — External Oscillator Frequency Range Select

RNGSEL works in conjunction with EXTSLOW to enable the amplifiers for the crystal oscillator.

Table 5-2. External Crystal Frequency Range Selection

EXTSLOW	RNGSEL	Frequency Range
0	0	8–32 MHz
0	1	1–8 MHz
1	0	32–100 kHz
1	1	Reserved

Setting EXTSLOW will force RNGSEL to a 0. RNGSEL cannot be written if EXTSLOW = 1.

ESCISRE — Slew Rate Enable for ESCI

- 1 = Slew rate controlled output for TxD
- 0 = Normal output

SPIPRE — Slew Rate Enable for SPI

- 1 = Slew rate controlled output for MISO, MOSI, and SCK
- 0 = Normal outputs

MCLKSRE — Slew Rate Enable for MCLK

- 1 = Slew rate controlled output for MCLK
- 0 = Normal output

PORTSRE — Slew Rate Enable for Ports

- 1 = Slew rate controlled output for all ports
- 0 = Normal output

ESCISEL — ESCI Pin Selection Bit

ESCISEL is used to select the pins to be used as ESCI pins when the ESCI is enabled. For more information on the ESCI, see [Chapter 13 Enhanced Serial Communications Interface \(ESCI\) Module](#).

1 = TxD on PTA2 — RxD on PTA3

0 = TxD on PTE0 — RxD on PTE1

SPISEL — SPI Pin Selection Bit

SPISEL is used to select the pins to be used as SPI pins when the SPI is enabled. For more information on the SPI, see [Chapter 15 Serial Peripheral Interface \(SPI\) Module](#).

1 = MISO on PTB3 — MOSI on PTB4 — SPSCK on PTB5 — \overline{SS} on PTC2

0 = MISO on PTC0 — MOSI on PTC1 — SPSCK on PTA5 — \overline{SS} on PTA6

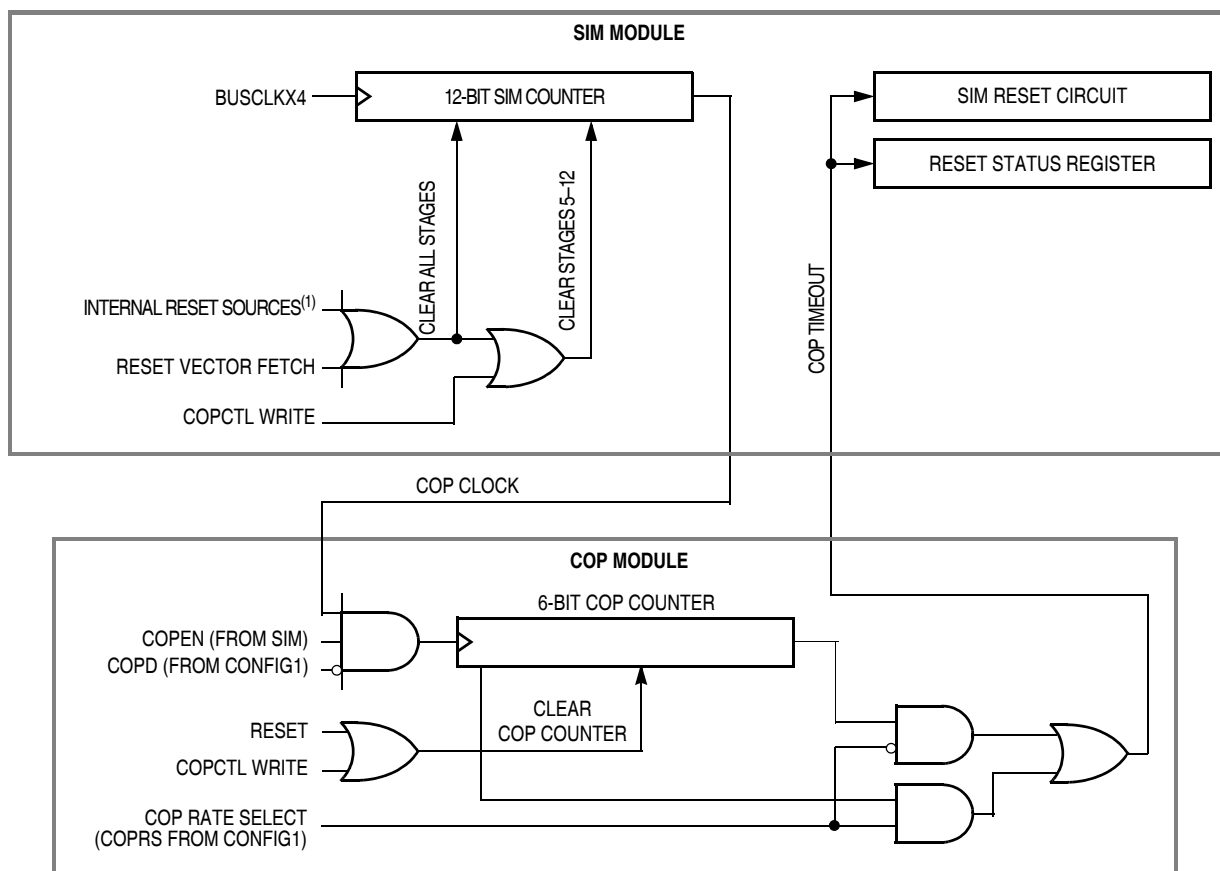
Chapter 6

Computer Operating Properly

6.1 Introduction

The computer operating properly (COP) module contains a free-running counter that generates a reset if allowed to overflow. The COP module helps software recover from runaway code. Prevent a COP reset by periodically clearing the COP counter.

6.2 Functional Description



1. See [Chapter 14 System Integration Module \(SIM\)](#) for more details.

Figure 6-1. COP Block Diagram

The COP counter is a free-running 6-bit counter preceded by a 12-bit prescaler. If not cleared by software, the COP counter overflows and generates an asynchronous reset after 8176 or 262,128 CGMXCLK cycles, depending on the state of the COP rate select bit, COPRS, in the CONFIG1. When COPRS = 0, a 4.9152-MHz crystal gives a COP timeout period of 53.3 ms. Writing any value to location \$FFFF before

7.7 Instruction Set Summary

Table 7-1 provides a summary of the M68HC08 instruction set.

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

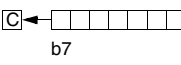
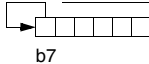
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ADC #opr ADC opr ADC opr ADC opr,X ADC opr,X ADC ,X ADC opr,SP ADC opr,SP	Add with Carry	$A \leftarrow (A) + (M) + (C)$	†	†	–	†	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	A9 B9 C9 D9 E9 F9 9EE9 9ED9	ii dd hh ll ee ff ff ff 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	$A \leftarrow (A) + (M)$	†	†	–	†	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	AB BB CB DB EB FB 9EEB 9EDB	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
AIS #opr	Add Immediate Value (Signed) to SP	$SP \leftarrow (SP) + (16 \ll M)$	–	–	–	–	–	–	IMM	A7	ii	2
AIX #opr	Add Immediate Value (Signed) to H:X	$H:X \leftarrow (H:X) + (16 \ll M)$	–	–	–	–	–	–	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	$A \leftarrow (A) \& (M)$	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 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 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 ff ff	4 1 1 4 3 5
BCC rel	Branch if Carry Bit Clear	$PC \leftarrow (PC) + 2 + rel ? (C) = 0$	–	–	–	–	–	–	REL	24	rr	3
BCLR n, opr	Clear Bit n in M	$M_n \leftarrow 0$	–	–	–	–	–	–	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)	$PC \leftarrow (PC) + 2 + rel ? (C) = 1$	–	–	–	–	–	–	REL	25	rr	3
BEQ rel	Branch if Equal	$PC \leftarrow (PC) + 2 + rel ? (Z) = 1$	–	–	–	–	–	–	REL	27	rr	3
BGE opr	Branch if Greater Than or Equal To (Signed Operands)	$PC \leftarrow (PC) + 2 + rel ? (N \oplus V) = 0$	–	–	–	–	–	–	REL	90	rr	3
BGT opr	Branch if Greater Than (Signed Operands)	$PC \leftarrow (PC) + 2 + rel ? (Z) \mid (N \oplus V) = 0$	–	–	–	–	–	–	REL	92	rr	3
BHCC rel	Branch if Half Carry Bit Clear	$PC \leftarrow (PC) + 2 + rel ? (H) = 0$	–	–	–	–	–	–	REL	28	rr	3
BHCS rel	Branch if Half Carry Bit Set	$PC \leftarrow (PC) + 2 + rel ? (H) = 1$	–	–	–	–	–	–	REL	29	rr	3
BHI rel	Branch if Higher	$PC \leftarrow (PC) + 2 + rel ? (C) \mid (Z) = 0$	–	–	–	–	–	–	REL	22	rr	3

Table 7-2. Opcode Map

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

INH Inherent
 IMM Immediate
 DIR Direct
 EXT Extended
 DD Direct-Direct
 IX+D Indexed-Direct
 REL Relative
 IX Indexed, No Offset
 IX1 Indexed, 8-Bit Offset
 IX2 Indexed, 16-Bit Offset
 IMM Immediate-Direct
 DIX+ Direct-Indexed

*Pre-byte for stack pointer indexed instructions

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

Low Byte of Opcode in Hexadecimal

MSB LSB	0	High Byte of Opcode in Hexadecimal
0	5 BRSET0 3 DIR	Cycles Opcode Mnemonic Number of Bytes / Addressing Mode

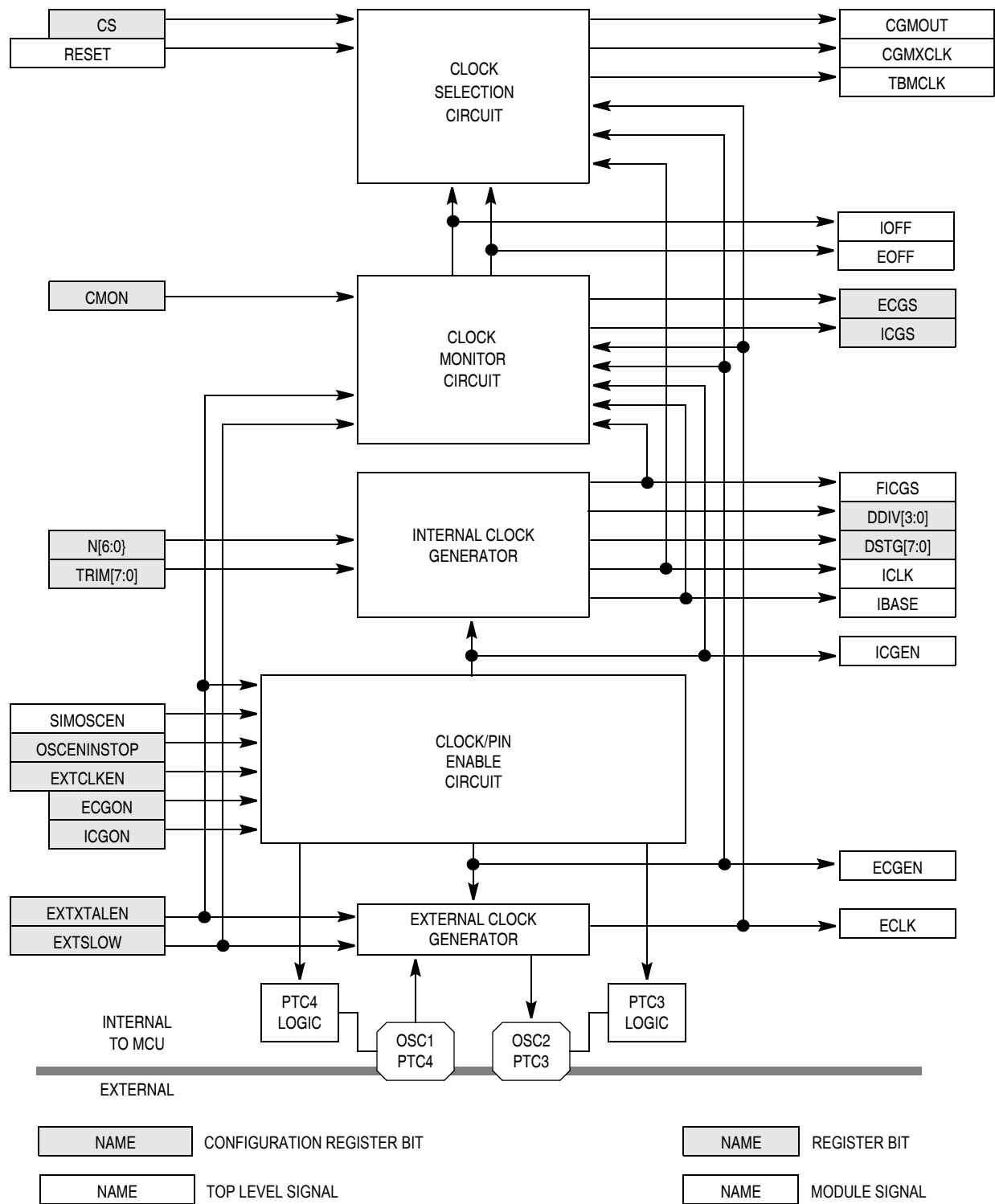


Figure 8-2. ICG Module Block Diagram

8.3.4.1 Clock Monitor Reference Generator

The clock monitor uses a reference based on one clock source to monitor the other clock source. The clock monitor reference generator generates the external reference clock (EREF) based on the external clock (ECLK) and the internal reference clock (IREF) based on the internal clock (ICLK). To simplify the circuit, the low-frequency base clock (IBASE) is used in place of ICLK because it always operates at or near 307.2 kHz. For proper operation, EREF must be at least twice as slow as IBASE and IREF must be at least twice as slow as ECLK.

To guarantee that IREF is slower than ECLK and EREF is slower than IBASE, one of the signals is divided down. Which signal is divided and by how much is determined by the external slow (EXTSLOW) and external crystal enable (EXTXTALEN) bits in the CONFIG, according to the rules in [Table 8-2](#).

NOTE

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

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

8.3.4.2 Internal Clock Activity Detector

The internal clock activity detector, shown in [Figure 8-6](#), looks for at least one falling edge on the low-frequency base clock (IBASE) every time the external reference (EREF) is low. Since EREF is less than half the frequency of IBASE, this should occur every time. If it does not occur two consecutive times, the internal clock inactivity indicator (IOFF) is set. IOFF will be cleared the next time there is a falling edge of IBASE while EREF is low.

The internal clock stable bit (ICGS) is also generated in the internal clock activity detector. ICGS is set when the internal clock generator's filter stable signal (FICGS) indicates that IBASE is within about 5 percent of the target 307.2 kHz \pm 25 percent for two consecutive measurements. ICGS is cleared when FICGS is clear, the internal clock generator is turned off or is in stop mode (ICGEN is clear), or when IOFF is set.

8.3.4.3 External Clock Activity Detector

The external clock activity detector, shown in [Figure 8-7](#), looks for at least one falling edge on the external clock (ECLK) every time the internal reference (IREF) is low. Since IREF is less than half the frequency of ECLK, this should occur every time. If it does not occur two consecutive times, the external clock inactivity indicator (EOFF) is set. EOFF will be cleared the next time there is a falling edge of ECLK while IREF is low.

18.3.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in [18.3.4 Pulse Width Modulation \(PWM\)](#). The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the value currently in the TIMB channel registers.

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

Use these methods to synchronize unbuffered changes in the PWM pulse width on channel x:

- When changing to a shorter pulse width, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current pulse. The interrupt routine has until the end of the PWM period to write the new value.
- When changing to a longer pulse width, enable TIMB overflow interrupts and write the new value in the TIMB overflow interrupt routine. The TIMB overflow interrupt occurs at the end of the current PWM period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same PWM period.

NOTE

In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare also can cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.

18.3.4.2 Buffered PWM Signal Generation

Channels 0 and 1 can be linked to form a buffered PWM channel whose output appears on the PTB6/AD6/TBCH0 pin. The TIMB channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS0B bit in TIMB channel 0 status and control register (TBSC0) links channel 0 and channel 1. The TIMB channel 0 registers initially control the pulse width on the PTB6/AD6/TBCH0 pin. Writing to the TIMB channel 1 registers enables the TIMB channel 1 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIMB channel registers (0 or 1) that control the pulse width are the ones written to last. TBSC0 controls and monitors the buffered PWM function, and TIMB channel 1 status and control register (TBSC1) is unused. While the MS0B bit is set, the channel 1 pin, PTB7/AD7/TBCH1, is available as a general-purpose I/O pin.

NOTE

In buffered PWM signal generation, do not write new pulse width values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered PWM signals.

18.3.4.3 PWM Initialization

To ensure correct operation when generating unbuffered or buffered PWM signals, use this initialization procedure:

1. In the TIMB status and control register (TBSC):
 - a. Stop the TIMB counter by setting the TIMB stop bit, TSTOP.
 - b. Reset the TIMB counter prescaler by setting the TIMB reset bit, TRST.
2. In the TIMB counter modulo registers (TBMODH–TBMODL), write the value for the required PWM period.
3. In the TIMB channel x registers (TBCHxH–TBCHxL), write the value for the required pulse width.
4. In TIMB channel x status and control register (TBSCx):
 - 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 18-2](#).
 - b. Write 1 to the toggle-on-overflow bit, TOVx.
 - c. Write 1:0 (to clear output on compare) or 1:1 (to set output on compare) to the edge/level select bits, ELSxB–ELSxA. The output action on compare must force the output to the complement of the pulse width level. See [Table 18-2](#).

NOTE

In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare can also cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.

5. In the TIMB status control register (TBSC), clear the TIMB stop bit, TSTOP.

Setting MS0B links channels 0 and 1 and configures them for buffered PWM operation. The TIMB channel 0 registers (TBCH0H–TBCH0L) initially control the buffered PWM output. TIMB status control register 0 (TBSC0) 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 TIMB 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% duty cycle output.

Setting the channel x maximum duty cycle bit (CHxMAX) and setting the TOVx bit generates a 100% duty cycle output. See [18.8.4 TIMB Channel Status and Control Registers](#).

18.4 Interrupts

These TIMB sources can generate interrupt requests:

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

18.8.2 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 — \$002C							
		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:									
Reset:		0	0	0	0	0	0	0	0

Register Name and Address		TBCNTL — \$002D							
		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:									
Reset:		0	0	0	0	0	0	0	0


 = Unimplemented

Figure 18-5. TIMB Counter Registers (TBCNTH and TBCNTL)

18.8.3 TIMB Counter Modulo Registers

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

Register Name and Address		TBMODH — \$002E							
		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:									
Reset:		1	1	1	1	1	1	1	1

Register Name and Address		TBMODL — \$002F							
		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:									
Reset:		1	1	1	1	1	1	1	1

Figure 18-6. TIMB Counter Modulo Registers (TMODH and TMODL)

NOTE

Reset the TIMB counter before writing to the TIMB counter modulo registers.

20.8 3V Control Timing

Characteristic ⁽¹⁾	Symbol	Min	Max	Unit
Frequency of operation ⁽²⁾				
Crystal option (EXTSLOW = 1, RNGSEL = 0)	f_{OSC}	32	100	kHz
Crystal option (EXTSLOW = 0, RNGSEL = 1)		1	8	MHz
Crystal option (EXTSLOW = 0, RNGSEL = 0)		8	16	MHz
External clock option (EXTSLOW = 1, RNGSEL = 0) ⁽³⁾		dc ⁽⁴⁾	100	kHz
External clock option (EXTSLOW = 0, RNGSEL = 1)		1	8	MHz
External clock option (EXTSLOW = 0, RNGSEL = 0)		8	16	MHz
Internal operating frequency	f_{OP}	—	4	MHz
Internal operating period ($1/f_{OP}$)	t_{CYC}	250	—	ns
\overline{RST} input pulse width low ⁽⁵⁾	t_{RL}	200	—	ns
\overline{IRQ} interrupt pulse width low ⁽⁶⁾ (edge-triggered)	t_{LIH}	200	—	ns
\overline{IRQ} interrupt pulse period	t_{LIL}	Note ⁽⁷⁾	—	t_{CYC}

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_{SS} , unless otherwise noted.
2. See [Chapter 8 Internal Clock Generator \(ICG\) Module](#) for more information.
3. No more than 10% duty cycle deviation from 50%
4. Some modules may require a minimum frequency greater than dc for proper operation. See appropriate table for this information.
5. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.
6. Minimum pulse width is for guaranteed interrupt. It is possible for a smaller pulse width to be recognized.
7. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1 t_{CYC} .

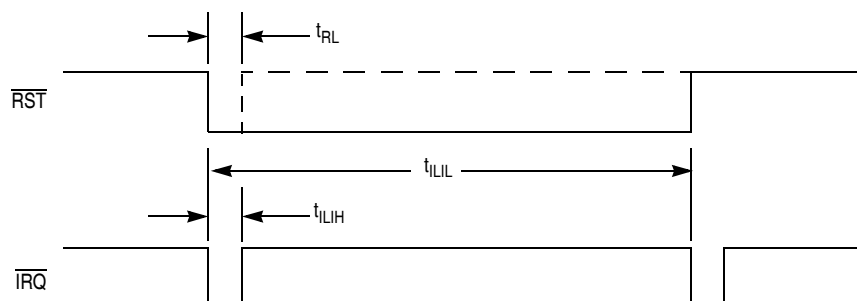


Figure 20-2. \overline{RST} and \overline{IRQ} Timing

20.9 Internal Oscillator Characteristics

Characteristic ⁽¹⁾	Symbol	Min	Typ	Max	Unit
Internal oscillator base frequency ^{(2), (3)}	f_{INTOSC}	230.4	307.2	384	kHz
Internal oscillator tolerance	f_{OSC_TOL}	-25	—	+25	%
Internal oscillator multiplier ⁽⁴⁾	N	1	—	127	—

1. $V_{DD} = 4.5$ to 5.5 Vdc, $V_{SS} = 0$ Vdc, $T_A = -40^\circ\text{C}$ to $+125^\circ\text{C}$, unless otherwise noted
2. Internal oscillator is selectable through software for a maximum frequency. Actual frequency will be multiplier (N) x base frequency.
3. $f_{BUS} = (f_{INTOSC} / 4) \times N$ when internal clock source selected
4. Multiplier must be chosen to limit the maximum bus frequency of 8 MHz for 4.5-V operation.