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	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	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908mr8cdwe

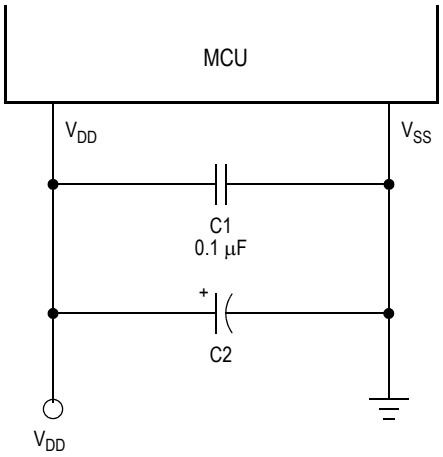
List of Figures

List of Tables

1.5.1 Power Supply Pins (V_{DD} and V_{SS})

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

Fast signal transitions on MCU pins place high, short-duration current demands on the power supply. To prevent noise problems, take special care to provide power supply bypassing at the MCU as **Figure 1-3** shows. Place the C1 bypass capacitor as close to the MCU as possible. Use a high-frequency-response ceramic capacitor for C1. C2 is an optional bulk current bypass capacitor for use in applications that require the port pins to source high current levels.



Note: Component values shown represent typical applications.

Figure 1-3. Power Supply Bypassing

1.5.2 Oscillator Pins (OSC1 and OSC2)

The OSC1 and OSC2 pins are the connections for the on-chip oscillator circuit. See **Section 8. Clock Generator Module (CGM)**.

1.5.3 External Reset Pin ($\overline{RST}$)

A logic 0 on the $\overline{RST}$ pin forces the MCU to a known startup state. $\overline{RST}$ is bidirectional, allowing a reset of the entire system. It is driven low when any internal reset source is asserted. See **Section 7. System Integration Module (SIM)**.

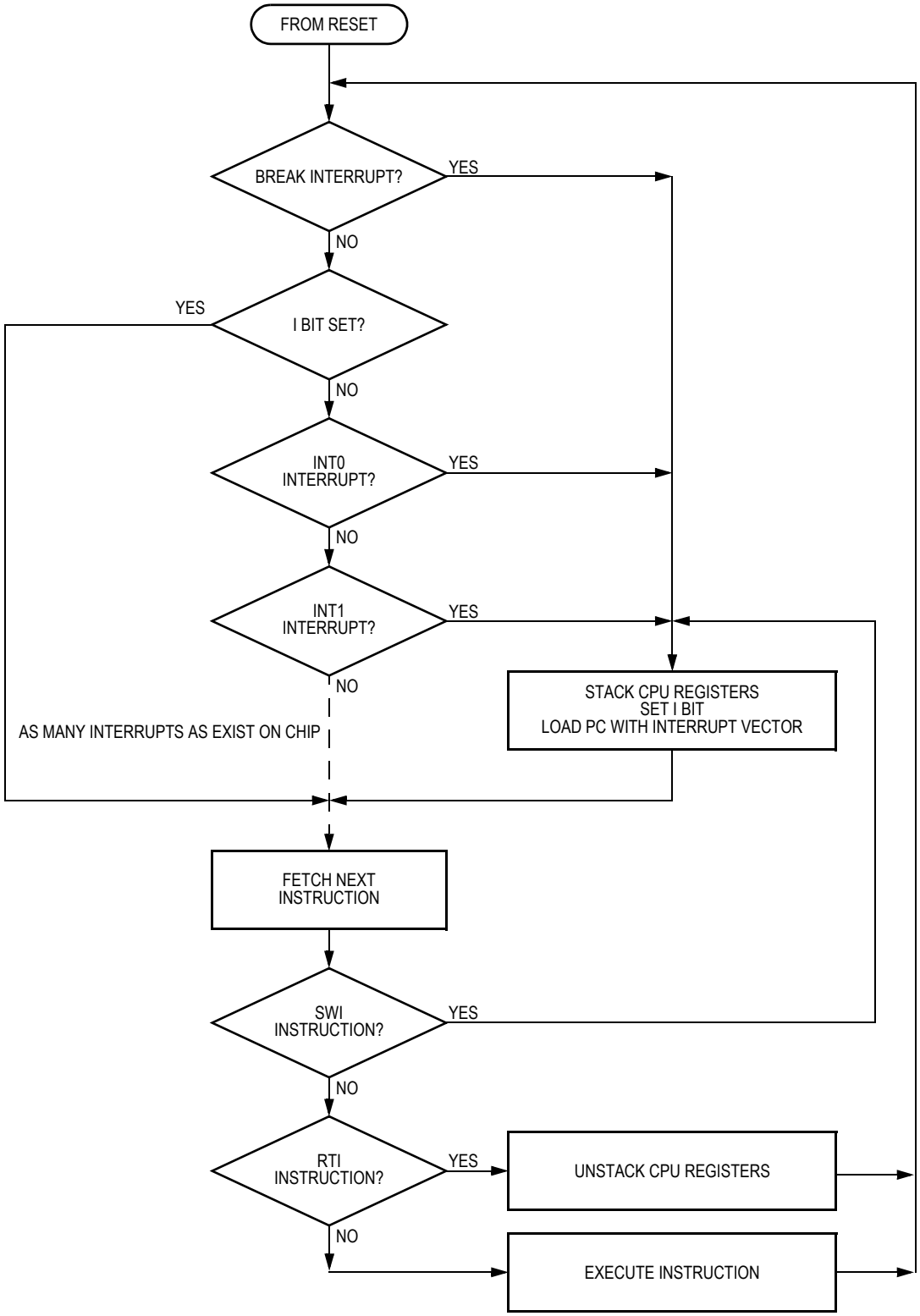


Figure 7-9. Interrupt Processing

Section 8. Clock Generator Module (CGM)

8.1 Contents

8.4.4 CGM External Connections

In its typical configuration, the CGM requires seven external components. Five of these are for the crystal oscillator and two are for the PLL.

The crystal oscillator is normally connected in a Pierce oscillator configuration, as shown in **Figure 8-3**. **Figure 8-3** shows only the logical representation of the internal components and may not represent actual circuitry.

The oscillator configuration uses five components:

1. Crystal, X_1
2. Fixed capacitor, C_1
3. Tuning capacitor, C_2 (can also be a fixed capacitor)
4. Feedback resistor, R_B
5. Series resistor, R_S (optional)

The series resistor (R_S) is included in the diagram to follow strict Pierce oscillator guidelines and may not be required for all ranges of operation, especially with high frequency crystals. Refer to the crystal manufacturer's data for more information.

Figure 8-3 also shows the external components for the PLL:

- Bypass capacitor, C_{BYP}
- Filter capacitor, C_F

Routing should be done with great care to minimize signal cross talk and noise. See **8.11 Acquisition/Lock Time Specifications** for routing information and more information on the filter capacitor's value and its effects on PLL performance.

8.6 CGM Registers

These registers control and monitor operation of the CGM:

- PLL control register (PCTL), see **8.6.1 PLL Control Register**
- PLL bandwidth control register (PBWC), see **8.6.2 PLL Bandwidth Control Register**
- PLL programming register (PPG), see **8.6.3 PLL Programming Register**

Figure 8-4 is a summary of the CGM registers.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$005C	PLL Control Register (PCTL) See page **126.**	Read:	PLLIE	PLLF	PLLON	BCS	1	1	1	1
		Write:		R			R	R	R	R
		Re-set:	0	0	1	0	1	1	1	1
$005D	PLL Bandwidth Control Register (PBWC) See page **129.**	Read:	AUTO	LOCK	$\overline{\text{ACQ}}$	XLD	0	0	0	0
		Write:		R			R	R	R	R
		Re-set:	0	0	0	0	0	0	0	0

Figure 8-4. CGM I/O Register Summary

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$005E	PLL Programming Register (PPG) See page **131.**	Read:	MUL7	MUL6	MUL5	MUL4	VRS7	VRS6	VRS5	VRS4
		Write:								
		Re-set:	0	1	1	0	0	1	1	0
		R	= Reserved							

Notes:
1. When AUTO = 0, PLLIE is forced to logic 0 and is read-only.
2. When AUTO = 0, PLLF and LOCK read as logic 0.
3. When AUTO = 1, ACQ is read-only.
4. When PLLON = 0 or VRS[7:4] = $0, BCS is forced to logic 0 and is read-only.
5. When PLLON = 1, the PLL programming register is read-only.
6. When BCS = 1, PLLON is forced set and is read-only.

Figure 8-4. CGM I/O Register Summary

8.6.1 PLL Control Register

The PLL control register (PCTL) contains the interrupt enable and flag bits, the on/off switch, and the base clock selector bit.

Address: $005C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PLLIE	PLLF	PLLON	BCS	1	1	1	1
Write:		R			R	R	R	R
Reset:	0	0	1	0	1	1	1	1
R	= Reserved							

Figure 8-5. PLL Control Register (PCTL)

Table 9-2. PWM Reload Frequency

Reload Frequency Bits LDFQ1:LDFQ0	PWM Reload Frequency
00	Every PWM cycle
01	Every 2 PWM cycles
10	Every 4 PWM cycles
11	Every 8 PWM cycles

For ease of software, the LDFQx bits are buffered. When the LDFQx bits are changed, the reload frequency will not change until the previous reload cycle is completed. See **Figure 9-5**.

NOTE: *When reading the LDFQx bits, the value is the buffered value (for example, not necessarily the value being acted upon).*

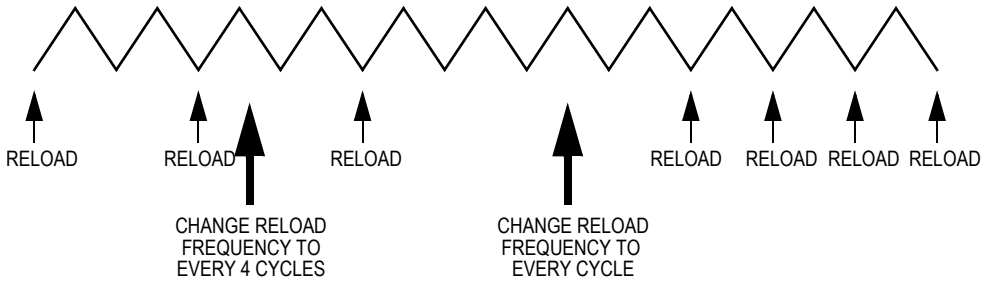


Figure 9-5. Reload Frequency Change

PWMINT enables CPU interrupt requests as shown in **Figure 9-6**. When this bit is set, CPU interrupt requests are generated when the PWMF bit is set. When the PWMINT bit is clear, PWM interrupt requests are inhibited. PWM reloads will still occur at the reload rate, but no interrupt requests will be generated.

9.6.4 Output Port Control Register

Conditions may arise in which the PWM pins need to be individually controlled. This is made possible by the PWM output control register (PWMOUT) shown in **Figure 9-18**.

Address:	$0025							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	OUTCTL	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 9-18. PWM Output Control Register (PWMOUT)

If the OUTCTL bit is set, the PWM pins can be controlled by the OUTx bits. These bits behave according to **Table 9-4**.

Table 9-4. OUTx Bits

OUTx Bit	Complementary Mode	Independent Mode
OUT1	1 — PWM1 is active. 0 — PWM1 is inactive.	1 — PWM1 is active 0 — PWM1 is inactive
OUT2	1 — PWM2 is complement of PWM1. 0 — PWM2 is inactive.	1 — PWM2 is active 0 — PWM2 is inactive
OUT3	1 — PWM3 is active. 0 — PWM3 is inactive.	1 — PWM3 is active 0 — PWM3 is inactive
OUT4	1 — PWM4 is complement of PWM3. 0 — PWM4 is inactive.	1 — PWM4 is active 0 — PWM4 is inactive
OUT5	1 — PWM5 is active. 0 — PWM5 is inactive.	1 — PWM5 is active 0 — PWM5 is inactive
OUT6	1 — PWM6 is complement of PWM5. 0 — PWM6 is inactive.	1 — PWM6 is active 0 — PWM6 is inactive

When OUTCTL is set, the polarity options TOPPOL and BOTPOL will still affect the outputs. In addition, if complementary operation is in use, the PWM pairs will not be allowed to be active simultaneously, and dead-time will still not be violated. When OUTCTL is set and

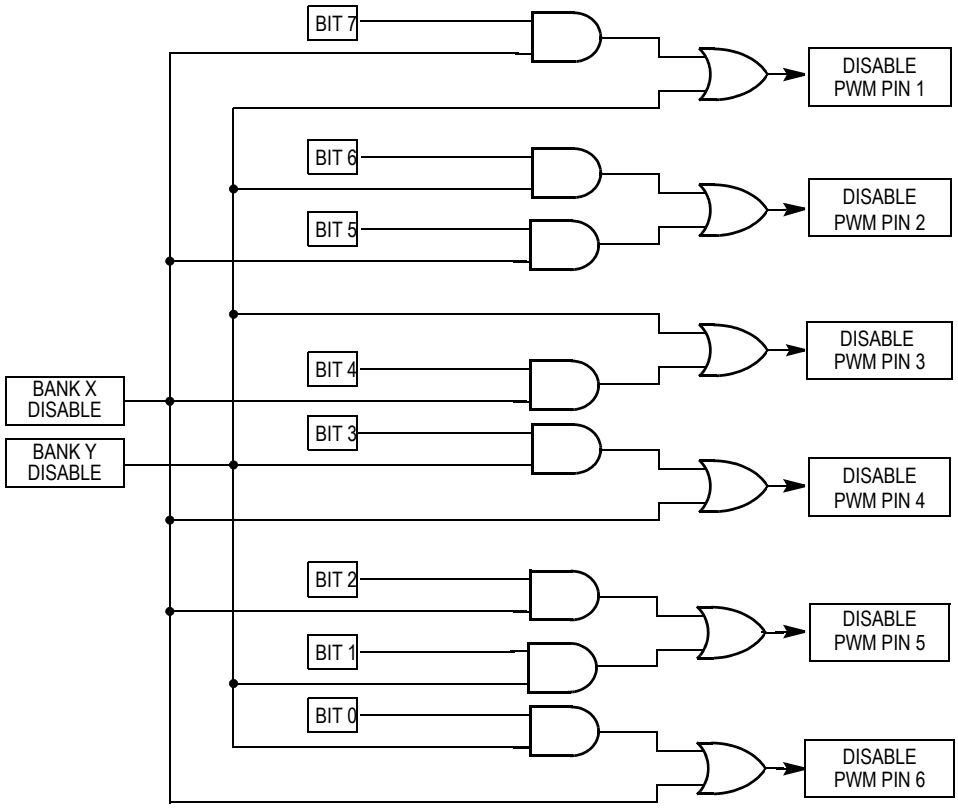


Figure 9-23. PWM Disabling Decode Scheme

9.7.1 Fault Condition Input Pins

A logic high level on a fault pin disables the respective PWM(s) determined by the bank and the disable mapping register. Each fault pin incorporates a filter to assist in rejecting spurious faults. All of the external fault pins are software-configurable to re-enable the PWMs either with the fault pin (automatic mode) or with software (manual mode). Each fault pin has an associated FMODE bit to control the PWM re-enabling method. Automatic mode is selected by setting the FMODEx bit in the fault control register. Manual mode is selected when FMODEx is clear.

NOTE: *PORTC, when used as an input port, mirrors the state of the fault input pins, as PORTC has the capability of being used as an output port. When either pin of PORTC is set as an output, by setting its respective PORTC data direction register bit, the respective fault pin logic is disconnected from that pin and the fault input will be defaulted to normal*

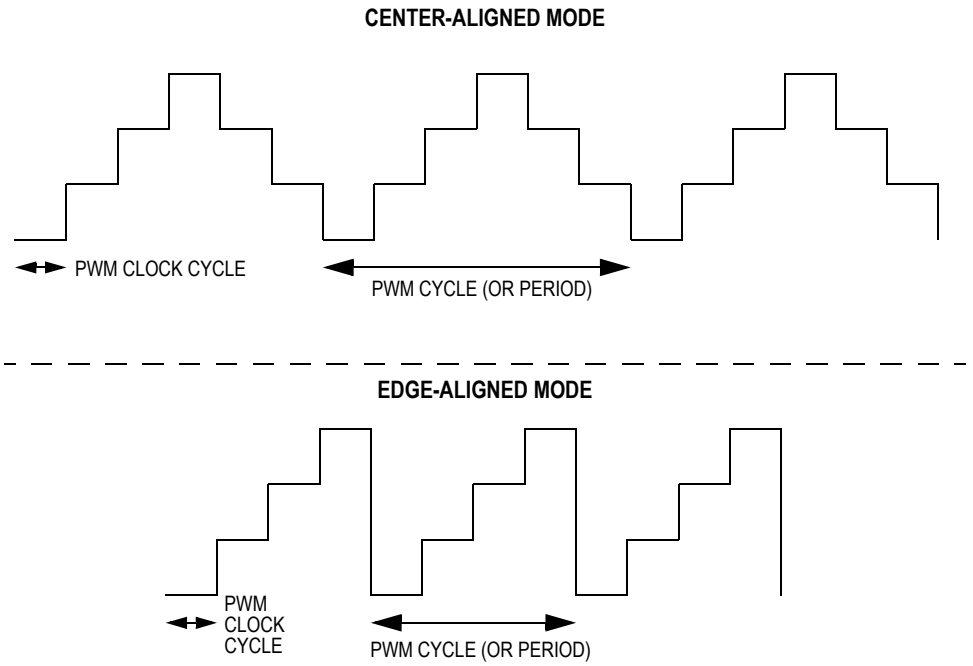


Figure 9-43. PWM Clock Cycle and PWM Cycle Definitions

PWM Load Frequency — The frequency at which new PWM parameters get loaded into the PWM. See **Figure 9-44**.

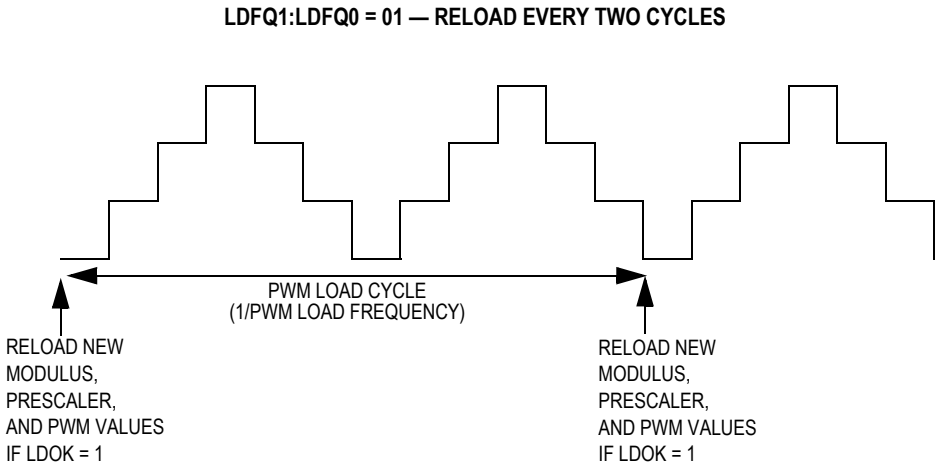


Figure 9-44. PWM Load Cycle/Frequency Definition

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

12.7 Stop Mode

TIMB is inactive after execution of a STOP instruction. The STOP instruction does not affect register conditions or the state of the TIMB counter.

TIMB operation resumes when the MCU exits stop mode after an external interrupt.

12.8 TIMB During Break Interrupts

A break interrupt stops the TIMB counter and inhibits input captures.

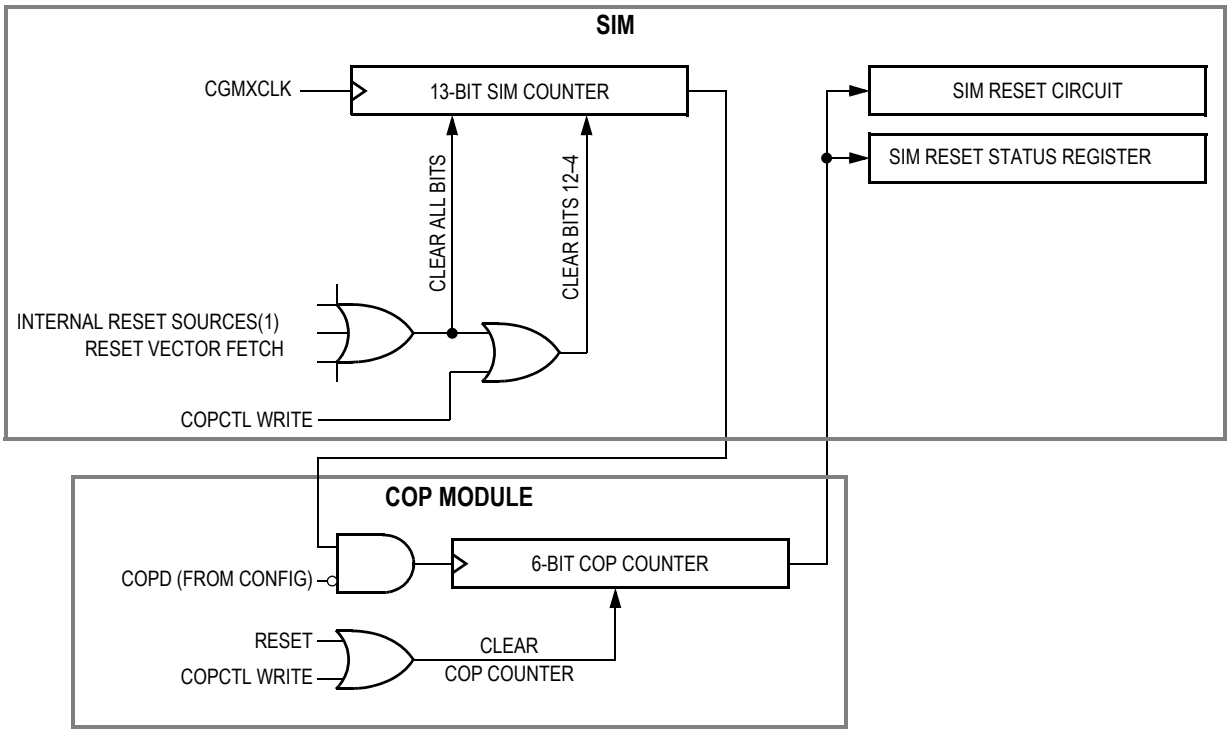
The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state. See **7.7.5 SIM Break Flag Control Register**.

To allow software to clear status bits during a break interrupt, write a logic 1 to the BCFE bit. 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 logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a 2-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is at logic 0. After the break, doing the second step clears the status bit.

15.3 Functional Description

Figure 15-1 shows the structure of the COP module.



Note: See 7.4.2 Active Resets from Internal Sources.

Figure 15-1. COP Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FFFF	COP Control Register (COPCTL) See page 294.	Read:	Low byte of reset vector							
		Write:	Clear COP counter							
		Reset:	Unaffected by reset							

Figure 15-2. COP I/O Register Summary

16.4 Functional Description

A logic 0 applied to any of the external interrupt pins can latch a CPU interrupt request. **Figure 16-1** shows the structure of the IRQ module.

Interrupt signals on the $\overline{\text{IRQ}}$ pin are latched into the IRQ latch. An interrupt latch remains set until one of the following actions occurs:

- Vector fetch — A vector fetch automatically generates an interrupt acknowledge signal that clears the latch that caused the vector fetch.
- Software clear — Software can clear an interrupt latch by writing to the appropriate acknowledge bit in the interrupt status and control register (ISCR). Writing a logic 1 to the ACK1 bit clears the IRQ latch.
- Reset — A reset automatically clears both interrupt latches.

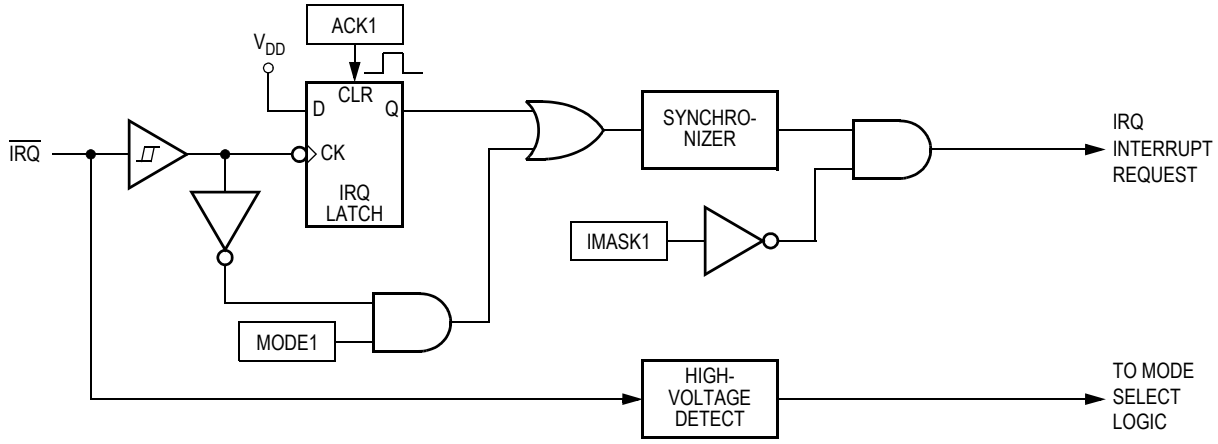


Figure 16-1. IRQ Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$003F	IRQ Status/Control Register (ISCR) See page **303.**	Read:	0	0	0	0	IRQF	0	IMASK1	MODE1
		Write:	R	R	R	R		ACK1		
		Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 16-2. IRQ I/O Register Summary

20.3 Features

Features of the break module include:

- Accessible input/output (I/O) registers during the break interrupt
- Central processor unit (CPU) generated break interrupts
- Software-generated break interrupts
- Computer operating properly (COP) disabling during break interrupts

20.4 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal to the CPU. The CPU then loads the instruction register with a software interrupt instruction (SWI) after completion of the current CPU instruction. The program counter vectors to $FFFC and $FFFD ($FEFC and $FEFD in monitor mode).

These events can cause a break interrupt to occur:

- A CPU-generated address (the address in the program counter) matches the contents of the break address registers.
- Software writes a logic 1 to the BRKA bit in the break status and control register.

When a CPU-generated address matches the contents of the break address registers, the break interrupt begins after the CPU completes its current instruction. A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the microcontroller unit (MCU) to normal operation. **Figure 20-1** shows the structure of the break module.

20.4.1 Flag Protection During Break Interrupts

The BCFE bit in the system integration module (SIM) break flag control register (SBFCR) enables software to clear status bits during the break state.

$FE0D	Break Address Register Low (BRKL) See page 334.	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	0	0	0	0	0	0	0	0
$FE0E	Break Status and Control Register (BRKSCR) See page 333.	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

Note: Writing a logic 0 clears BW.

= Unimplemented

R = Reserved

Figure 20-2. I/O Register Summary

20.4.2 CPU During Break Interrupts

The CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with $FFFC and $FFFD ($FEFC and $FEFD in monitor mode)

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

20.4.3 TIM During Break Interrupts

A break interrupt stops the timer counters.

20.4.4 COP During Break Interrupts

The COP is disabled during a break interrupt when V_{TST} is present on the $\overline{RST}$ pin.

20.6.3 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from wait mode. The flag is useful in applications requiring a return to wait mode after exiting from a break interrupt.

Address: $FE00

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

Note: Writing a logic 0 clears BW.

R = Reserved

Figure 20-6. SIM Break Status Register (SBSR)

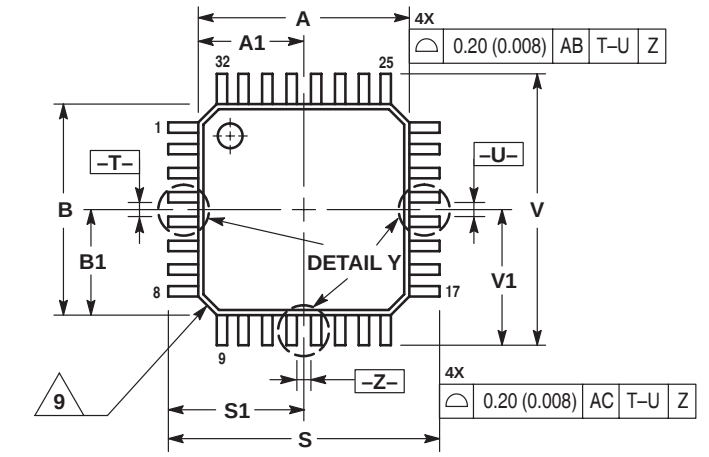
BW — Break Wait Bit

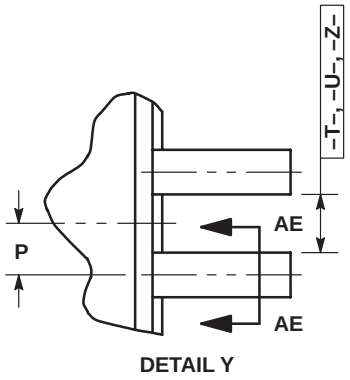
This read/write bit is set when a break interrupt causes an exit from wait mode. Clear BW by writing a logic 0 to it. Reset clears BW.

1 = Break interrupt during wait mode
0 = No break interrupt during wait mode

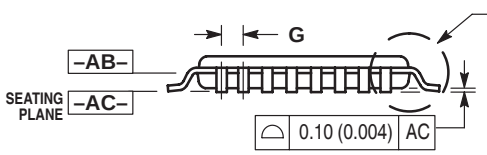
BW can be read within the break interrupt routine. The user can modify the return address on the stack by subtracting 1 from it. The example code shown in **Figure 20-7** works if the H register was stacked in the break interrupt routine. Execute this code at the end of the break interrupt routine.

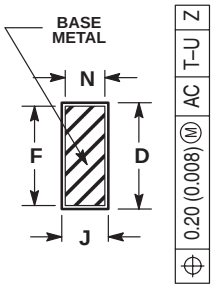
22.3 32-Pin LQFP (Case #873A)



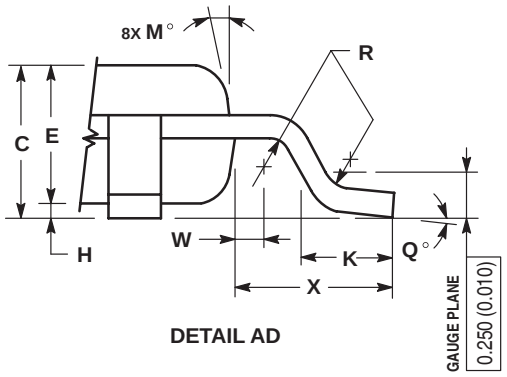


DETAIL Y





SECTION AE–AE



DETAIL AD

NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE –AB– IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS –T–, –U–, AND –Z– TO BE DETERMINED AT DATUM PLANE –AB–.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE –AC–.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.250 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE –AB–.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE D DIMENSION TO EXCEED 0.520 (0.020).
8. MINIMUM SOLDER PLATE THICKNESS SHALL BE 0.0076 (0.0003).
9. EXACT SHAPE OF EACH CORNER MAY VARY FROM DEPICTION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	7.000 BSC		0.276 BSC	
A1	3.500 BSC		0.138 BSC	
B	7.000 BSC		0.276 BSC	
B1	3.500 BSC		0.138 BSC	
C	1.400	1.600	0.055	0.063
D	0.300	0.450	0.012	0.018
E	1.350	1.450	0.053	0.057
F	0.300	0.400	0.012	0.016
G	0.800 BSC		0.031 BSC	
H	0.050	0.150	0.002	0.006
J	0.090	0.200	0.004	0.008
K	0.500	0.700	0.020	0.028
M	12° REF		12° REF	
N	0.090	0.160	0.004	0.006
P	0.400 BSC		0.016 BSC	
Q	1°	5°	1°	5°
R	0.150	0.250	0.006	0.010
S	9.000 BSC		0.354 BSC	
S1	4.500 BSC		0.177 BSC	
V	9.000 BSC		0.354 BSC	
V1	4.500 BSC		0.177 BSC	
W	0.200 REF		0.008 REF	
X	1.000 REF		0.039 REF	

comparator — A device that compares the magnitude of two inputs. A digital comparator defines the equality or relative differences between two binary numbers.

computer operating properly module (COP) — A counter module in the M68HC08 Family that resets the MCU if allowed to overflow.

condition code register (CCR) — An 8-bit register in the CPU08 that contains the interrupt mask bit and five bits that indicate the results of the instruction just executed.

control bit — One bit of a register manipulated by software to control the operation of the module.

control unit — One of two major units of the CPU. The control unit contains logic functions that synchronize the machine and direct various operations. The control unit decodes instructions and generates the internal control signals that perform the requested operations. The outputs of the control unit drive the execution unit, which contains the arithmetic logic unit (ALU), CPU registers, and bus interface.

COP — See computer operating properly module (COP).

counter clock — The input clock to the TIM counter. This clock is an output of the prescaler sub-module. The frequency of the counter clock is f_{TCNT}, and the period is t_{TCNT}.

CPU — See central processor unit (CPU).

CPU08 — The central processor unit of the M68HC08 Family.

CPU cycles — A CPU clock cycle is one period of the internal bus-rate clock, f_{OP}, normally derived by dividing a crystal oscillator source by two or more so the high and low times will be equal. The length of time required to execute an instruction is measured in CPU clock cycles.

CPU registers — Memory locations that are wired directly into the CPU logic instead of being part of the addressable memory map. The CPU always has direct access to the information in these registers. The CPU registers in an M68HC08 are:

- A, 8-bit accumulator
- H:X, 16-bit index register
- SP, 16-bit stack pointer
- PC, 16-bit program counter
- CCR, condition code register containing the V, H, I, N, Z, and C bits