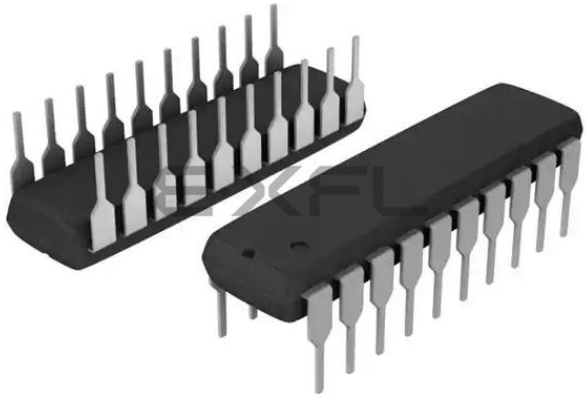


Welcome to E-XFL.COM

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	-
Peripherals	LVR, POR, PWM
Number of I/O	18
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 7x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-DIP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908lb8vpe

MC68HC908LB8

Data Sheet

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://www.freescale.com

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Revision History

Date	Revision Level	Description	Page Number(s)
1/2005	0	First release	N/A
8/2005	1	Section 4.7 Application Information added. Minor changes to the second and third paragraphs in the note in Section 10.4.9 Deadtime Insertion.	56 101

  - Three shared with op amp/comparator
  - Seven shared with ADC module (AD[0:6])
  - One shared with timer channel 0
  - Two shared with OSC1 and OSC2
  - One shared with reset
  - Seven shared with keyboard interrupt
  - One input-only pin shared with external interrupt (IRQ)
- Available packages:
  - 20-pin small outline integrated chip (SOIC) package
  - 20-pin plastic dual in-line package (PDIP)
- On-chip programming firmware for use with host personal computer which does not require high voltage for entry
- System protection features:
  - Optional computer operating properly (COP) reset
  - Low-voltage reset
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- Low-power design; fully static with stop and wait modes
- Standard low-power modes of operation:
  - Wait mode
  - Stop mode
- Master reset pin and power-on reset (POR)
- 674 bytes of FLASH programming routines read-only memory (ROM)
- Break module (BRK) to allow single breakpoint setting during in-circuit debugging
- Internal pullup on $\overline{\text{RST}}$ pin to reduce customer system cost

Data registers are shown in Figure 2-2. Table 2-1 is a list of vector locations.

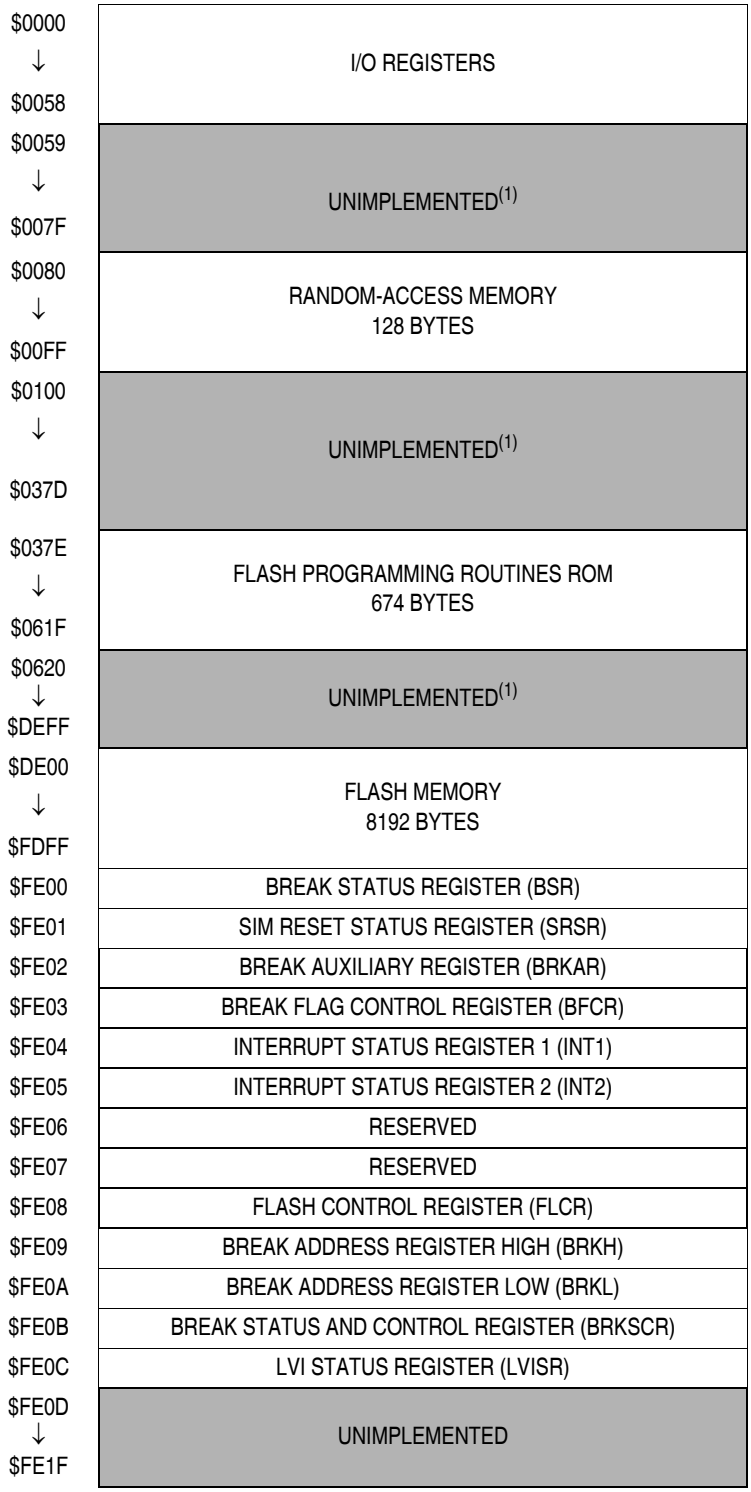


Figure 2-1. Memory Map

9. Clear the HVEN bit.
10. After time, t_{RCV} (typical 1 μs), the memory can be accessed in read mode again.

NOTE

Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps.

CAUTION

A mass erase will erase the internal oscillator trim value at $FFC0.

2.6.4 FLASH Program/Read Operation

Programming of the FLASH memory is done on a row basis. A row consists of 32 consecutive bytes starting from addresses $XX00, $XX20, $XX40, $XX60, $XX80, $XXA0, $XXC0, and $XXE0.

During the programming cycle, make sure that all addresses being written to fit within one of the ranges specified above. Attempts to program addresses in different row ranges in one programming cycle will fail. Use this step-by-step procedure to program a row of FLASH memory (Figure 2-4 is a flowchart representation).

NOTE

In order to avoid program disturbs, the row must be erased before any byte on that row is programmed.

1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
2. Read from the FLASH block protect register.
3. Write any data to any FLASH address within the row address range desired.
4. Wait for a time, t_{NVS} (minimum 10 μs).
5. Set the HVEN bit.
6. Wait for a time, t_{PGS} (minimum 5 μs).
7. Write data to the FLASH address to be programmed.
8. Wait for a time, t_{PROG} (minimum 30 μs).
9. Repeat step 7 and 8 until all the bytes within the row are programmed.
10. Clear the PGM bit.(1)
11. Wait for a time, t_{NVH} (minimum 5 μs).
12. Clear the HVEN bit.
13. After time, t_{RCV} (minimum 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.

1. The time between each FLASH address change, or the time between the last FLASH address programmed to clearing PGM bit, must not exceed the maximum programming time, t_{PROG} maximum.

3.2 Features

Features of the ADC module include:

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

3.3 Functional Description

Seven ADC channels are available for sampling external sources at pins PTB7/ADC6, PTA6/ADC5, PTA4/ADC4–PTA0/ADC0. An analog multiplexer allows a single ADC converter to select one of seven ADC channels as ADC voltage in (ADCVIN). ADCVIN is converted by the successive approximation register based counters. When the conversion is complete, ADC places the result in the ADC data register and sets a flag or generates an interrupt.
See Figure 3-2.

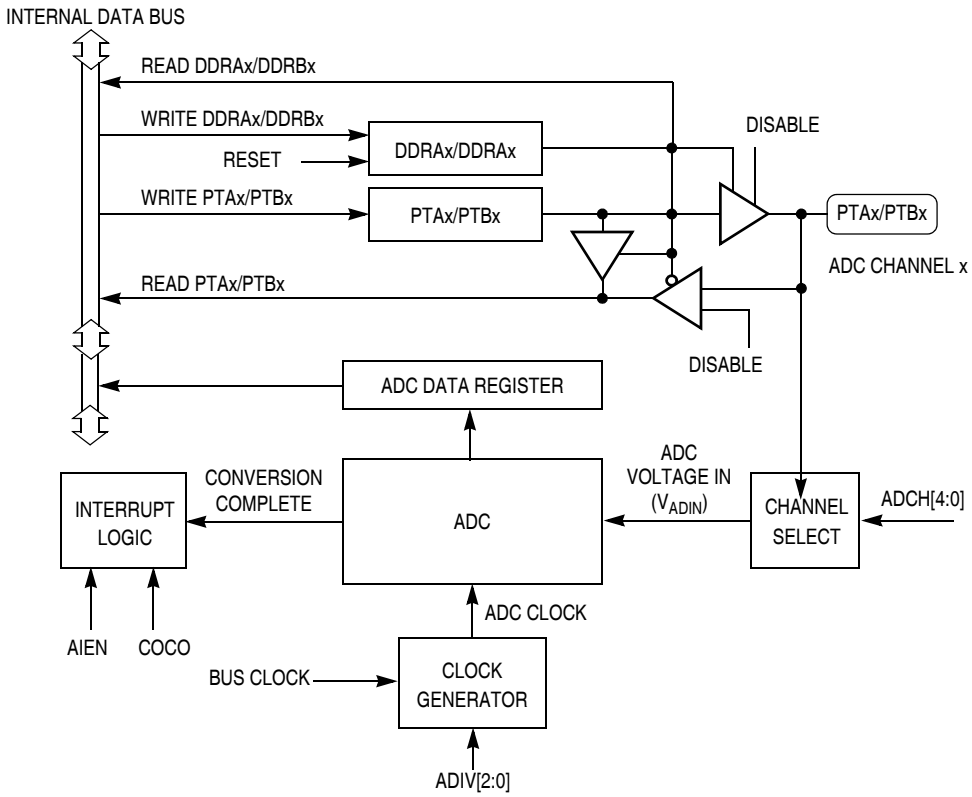


Figure 3-2. ADC Block Diagram

3.3.1 ADC Port I/O Pins

PTB7/ADC6, PTA6/ADC5, PTA4/ADC4–PTA0/ADC0 are general-purpose I/O (input/output) pins that share with the ADC channels. The channel select bits define which ADC channel/port pin will be used as

NOTES:
1. If any unused channels are selected, the resulting ADC conversion will be unknown or reserved.
2. V_{REFH} and V_{REFL} are internally connected to V_{DD} and V_{SS} respectively.

3.8.2 ADC Data Register

One 8-bit result register is provided. This register is updated each time an ADC conversion completes.

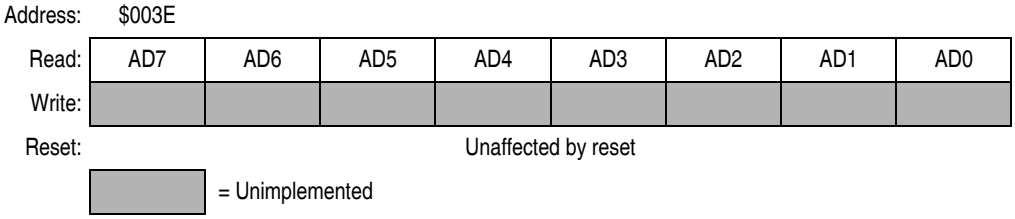


Figure 3-4. ADC Data Register (ADR)

3.8.3 ADC Clock Register

The ADC clock register (ADCLK) selects the clock frequency for the ADC.

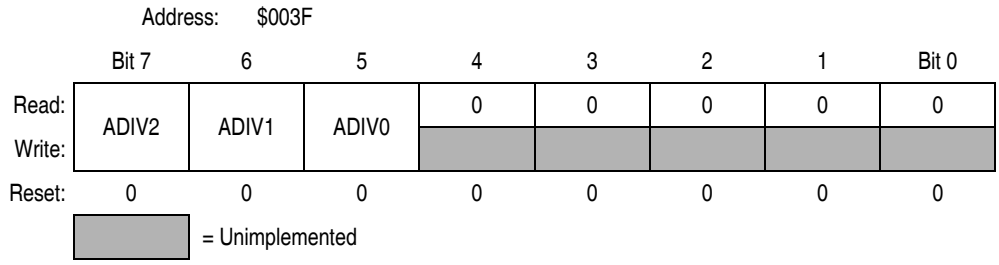


Figure 3-5. ADC Clock Register (ADCLK)

ADIV2–ADIV0 — ADC Clock Prescaler Bits

ADIV2–ADIV0 form a 3-bit field which selects the divide ratio used by the ADC to generate the internal ADC clock. Table 3-2 shows the available clock configurations. The ADC clock should be set to approximately 1 MHz.

Table 3-2. ADC Clock Divide Ratio

ADIV2	ADIV1	ADIV0	ADC Clock Rate
0	0	0	ADC input clock ÷ 1
0	0	1	ADC input clock ÷ 2
0	1	0	ADC input clock ÷ 4
0	1	1	ADC input clock ÷ 8
1	X(1)	X(1)	ADC input clock ÷ 16

NOTES:
1. X = Don't care

The ADC requires a clock rate of approximately 1 MHz for correct operation. If the selected clock source is not fast enough, the ADC will generate incorrect conversions. See 20.8 5.0-Volt ADC Characteristics.

$$f_{ADIC} = \frac{\text{Bus frequency}}{\text{ADIV[2:0]}} \cong 1 \text{ MHz}$$

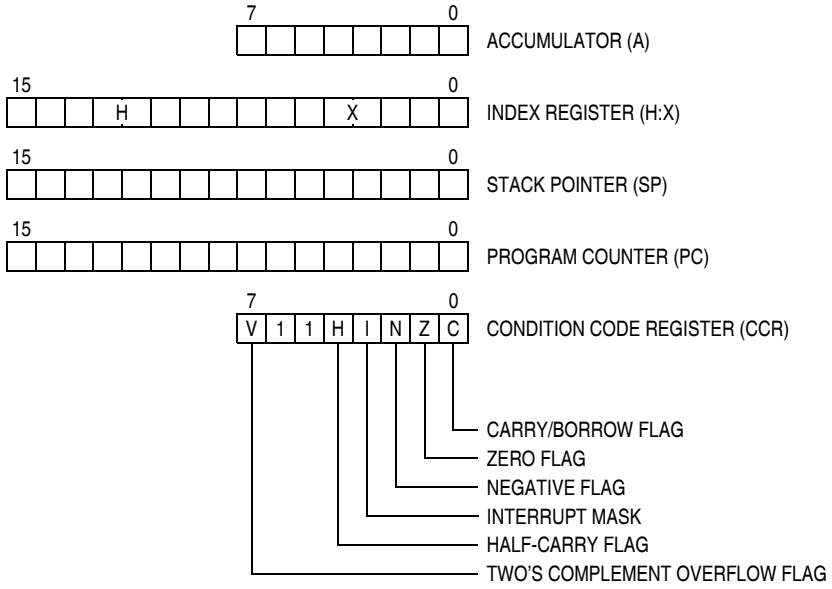


Figure 7-1. CPU Registers

7.3.1 Accumulator

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and the results of arithmetic/logic operations.

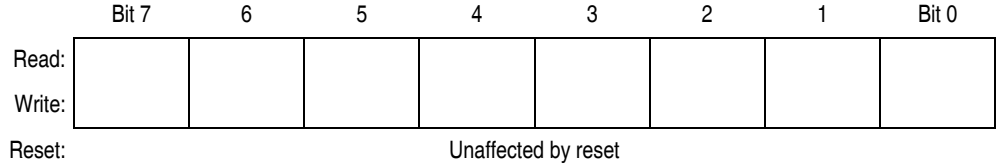


Figure 7-2. Accumulator (A)

7.3.2 Index Register

The 16-bit index register allows indexed addressing of a 64-Kbyte memory space. H is the upper byte of the index register, and X is the lower byte. H:X is the concatenated 16-bit index register.

In the indexed addressing modes, the CPU uses the contents of the index register to determine the conditional address of the operand.

The index register can serve also as a temporary data storage location.

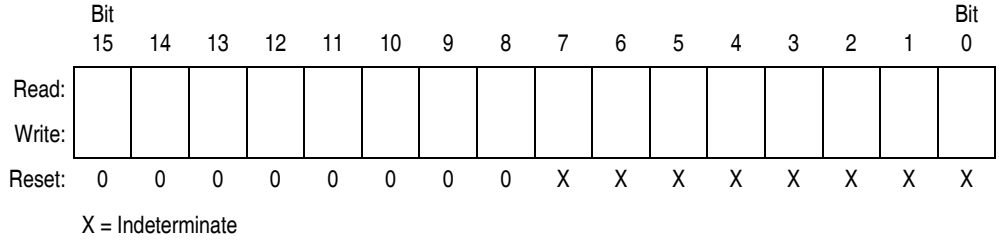


Figure 7-3. Index Register (H:X)

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* (Freescale Semiconductor 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 2 of 7)

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

- Controls triggering sensitivity of the $\overline{\text{IRQ}}$ interrupt pin

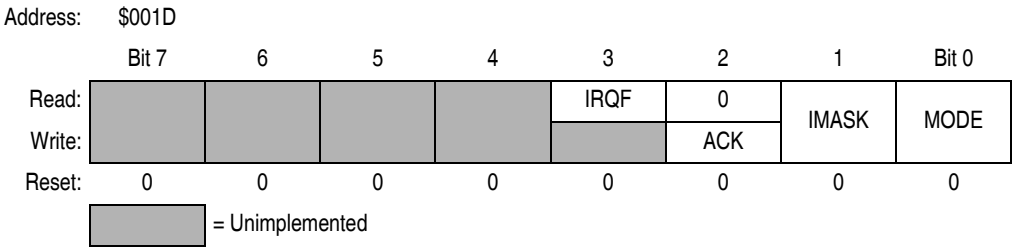


Figure 8-3. IRQ Status and Control Register (INTSCR)

IRQF — IRQ Flag Bit

This read-only status bit is high when the IRQ interrupt is pending.

1 = $\overline{\text{IRQ}}$ interrupt pending
0 = $\overline{\text{IRQ}}$ interrupt not pending

ACK — IRQ Interrupt Request Acknowledge Bit

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

IMASK — IRQ Interrupt Mask Bit

Writing a 1 to this read/write bit disables IRQ interrupt requests. Reset clears IMASK.

1 = IRQ interrupt requests disabled
0 = IRQ interrupt requests enabled

MODE — IRQ Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the $\overline{\text{IRQ}}$ pin. Reset clears MODE.

1 = $\overline{\text{IRQ}}$ interrupt requests on falling edges and low levels
0 = $\overline{\text{IRQ}}$ interrupt requests on falling edges only

The rate of switching is controlled by the dithering controller, and is dependent on the values of the CLKSRC bit and the SEL[2:0] bits in the HRPDCR register, the contents of the HRPTBH:HRPTBL registers, and, depending on the value of the HRPMODE bit, the five least significant bits in the HRPPERL or HRPDCL registers**.**

Table 10-2. HRPMODE Bit Options

HRPMODE	Mode	PERIOD1	PERIOD2	DUTY1	DUTY2
0	Variable Frequency	P[10:0]	P[10:0] +1	PERIOD1/2	PERIOD2/2
1	Variable Duty Cycle	P[10:0]	P[10:0]	DC[10:0]	DC[10:0] +1

For more detailed information, see 10.4.7 Dithering Controller.

11.10 Op Amp/Comparator

11.10.1 Wait Mode

While in WAIT the state of the op amp/comparator cannot be changed. If the op amp/comparator module is not needed during wait mode, reduce power consumption by disabling the op amp/comparator before executing the WAIT command.

11.10.2 Stop Mode

The op amp/comparator is inactive after execution of the STOP command. The op amp/comparator will be in a low-power state and will not drive its output pin. When the MCU exits stop mode after and external interrupt, the op amp/comparator continues operation.

11.11 Oscillator Module (OSC)

11.11.1 Wait Mode

The WAIT instruction has no effect on the oscillator logic. BUSCLKX2 and BUSCLKX4 continue to drive to the SIM module.

11.11.2 Stop Mode

The STOP instruction disables either the XTALCLK, the RCCLK, or INTCLK output, hence BUSCLKX2 and BUSCLKX4.

11.12 Pulse-Width Modulator Module (PWM)

11.12.1 Wait Mode

When the microcontroller is put in low-power wait mode via the WAIT instruction, all clocks to the PWM module will continue to run. If an interrupt is issued from the PWM module (via a reload or a fault), the microcontroller will exit wait mode.

Clearing the PWMEN bit before entering wait mode will reduce power consumption in wait mode because the counter, prescaler divider, and LDFQ divider will no longer be clocked. In addition, power will be reduced because the PWMs will no longer toggle.

11.12.2 Stop Mode

When the microcontroller is put into stop mode via the STOP instruction, the PWM will stop functioning. The PWM0 and PWM1 outputs are set to logic 0. The STOP instruction does not affect the register conditions or the state of the PWM counters. When the MCU exits stop mode after an external interrupt the PWM resumes operation.

13.4 Oscillator Module Signals

The following paragraphs describe the signals that are inputs to and outputs from the oscillator module.

13.4.1 Crystal Amplifier Input Pin (OSC1)

The OSC1 pin is either an input to the crystal oscillator amplifier, an input to the RC oscillator circuit, or an external clock source.

For the internal oscillator configuration, the OSC1 pin can assume other functions according to Table 13-1.

13.4.2 Crystal Amplifier Output Pin (OSC2/PTC1/BUSCLKX4)

For the XTAL oscillator device**,** the OSC2 pin is the crystal oscillator inverting amplifier output.

For the external clock option, the OSC2 pin is dedicated to the PTC1 I/O function. The OSC2EN bit has no effect.

For the internal oscillator or RC oscillator options, the OSC2 pin can assume other functions according to Table 13-1, or the output of the oscillator clock (BUSCLKX4).

Table 13-1. OSC2 Pin Function

Option	OSC2 Pin Function
XTAL oscillator	Inverting OSC1
External clock	PTC1 I/O
Internal oscillator or RC oscillator	Controlled by OSC2EN bit in PTCPUE register OSC2EN = 0: PTC1 I/O OSC2EN = 1: BUSCLKX4 output

13.4.3 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal comes from the system integration module (SIM) and enables/disables either the XTAL oscillator circuit, the RC oscillator, or the internal oscillator.

13.4.4 XTAL Oscillator Clock (XTALCLK)

XTALCLK is the XTAL oscillator output signal. It runs at the full speed of the crystal (f_{XCLK}) and comes directly from the crystal oscillator circuit. Figure 13-2 shows only the logical relation of XTALCLK to OSC1 and OSC2 and may not represent the actual circuitry. The duty cycle of XTALCLK is unknown and may depend on the crystal and other external factors. Also, the frequency and amplitude of XTALCLK can be unstable at start up.

13.4.5 RC Oscillator Clock (RCCLK)

RCCLK is the RC oscillator output signal. Its frequency is directly proportional to the time constant of external R and internal C. Figure 13-3 shows only the logical relation of RCCLK to OSC1 and may not represent the actual circuitry.

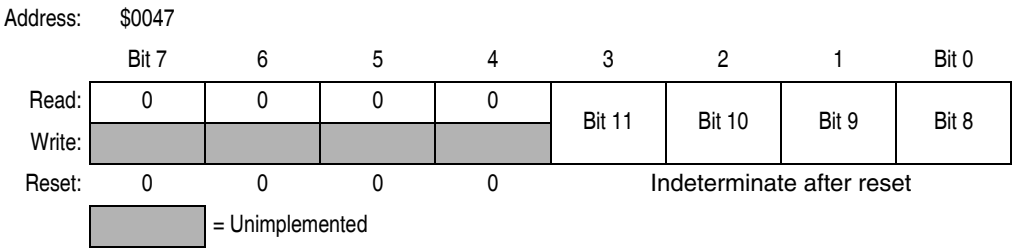


Figure 15-16. PWM Counter Modulo Register High (PMODH)

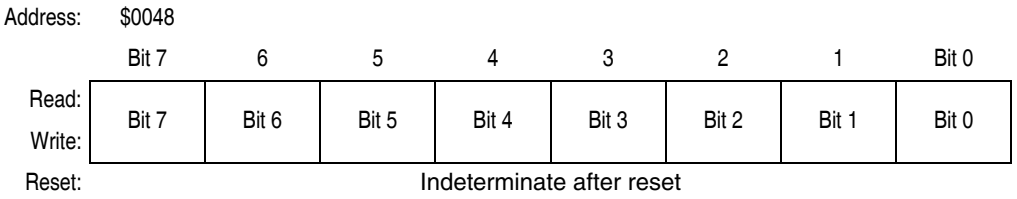


Figure 15-17. PWM Counter Modulo Register Low (PMODL)

To avoid erroneous PWM periods, this value is buffered and will not be used by the PWM generator until the LDOK bit has been set and the next PWM load cycle begins.

NOTE

When reading this register, the value read is the buffer (not necessarily the value the PWM generator is currently using).

Because of the equals-comparator architecture of this PWM, the modulus = 0 case is considered illegal. Therefore, the modulus register is not reset, and a modulus value of 0 will result in waveforms inconsistent with the other modulus waveforms. If a modulus of 0 is loaded, the counter will continually count down from $FFF. This operation will not be tested or guaranteed (the user should consider it illegal). However, the fault conditions will still be guaranteed.

15.8.3 PWMx Value Registers

Each of the two PWMs has a 16-bit PWM value register.

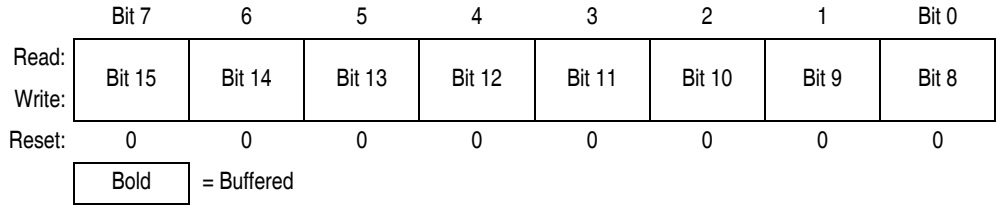


Figure 15-18. PWMx Value Registers High (PVALxH)

Address: $0041

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	**LDFQ1**	**LDFQ0**	DIS1	DIS0	POL1	POL0	**PRSC1**	**PRSC0**
Reset:	0	0	0	0	1	1	0	0

Bold = Buffered

Figure 15-21. PWM Control Register 2 (PCTL2)

LDFQ1 and LDFQ0 — PWM Load Frequency Bits

These buffered read/write bits select the PWM CPU load frequency according to Table 15-4.

NOTE

When reading these bits, the value read is the buffer value (not necessarily the value the PWM generator is currently using).

The LDFQx bits take effect when the current load cycle is complete regardless of the state of the load okay bit, LDOK.

Table 15-4. PWM Reload Frequency

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

NOTE

Reading the LDFQx bit reads the buffered values and not necessarily the values currently in effect.

DIS1 — Software Disable Bit for PWM1

This read/write bit allows the user to disable pin PWM1.

1 = Disable PWM1
0 = Re-enable PWM1

DIS0 — Software Disable Bit for PWM0

This read/write bit allows the user to disable pin PWM0.

1 = Disable PWM0
0 = Re-enable PWM0

POL1 — Polarity Bit for PWM1

This read/write bit selects the polarity of the PWM waveform of PWM1. Positive polarity means that when the PWM is active the PWM output is high. Conversely, negative polarity means that when the PWM is active the PWM output is low.

1 = PWM1 has positive polarity
0 = PWM1 has negative polarity

19.3.1.4 Data Format

Communication with the monitor module is in standard non-return-to-zero (NRZ) mark/space data format. Transmit and receive baud rates must be identical.

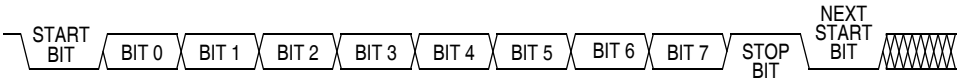


Figure 19-13. Monitor Data Format

19.3.1.5 Break Signal

A start bit (0) followed by nine 0 bits is a break signal. When the monitor receives a break signal, it drives the PTA0 pin high for the duration of two bits and then echoes back the break signal.

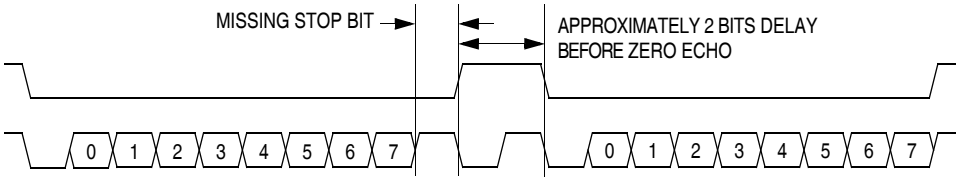


Figure 19-14. Break Transaction

19.3.1.6 Baud Rate

The communication baud rate is controlled by the external clock frequency or internal oscillator frequency.

Table 19-1 has the external frequency required to achieve a standard baud rate of 9600 bps. The effective baud rate is the bus frequency divided by 256 for the external oscillator and divided by 417 for the internal oscillator. If a crystal is used as the source, be aware of the upper frequency limit that the MCU can operate.

19.3.1.7 Commands

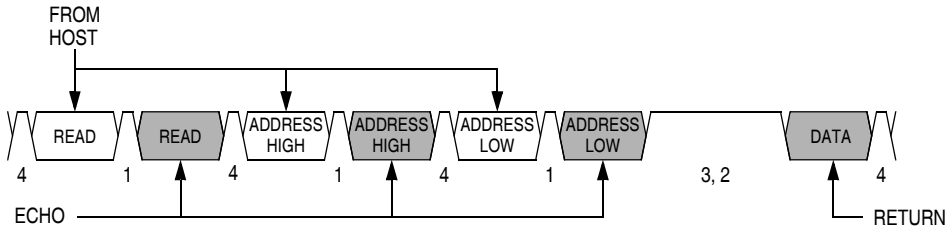
The monitor module firmware uses these commands:

- READ (read memory)
- WRITE (write memory)
- IREAD (indexed read)
- IWRITE (indexed write)
- READSP (read stack pointer)
- RUN (run user program)

The monitor module firmware echoes each received byte back to the PTA0 pin for error checking. An 11-bit delay at the end of each command allows the host to send a break character to cancel the command. A delay of two bit times occurs before each echo and before READ, IREAD, or READSP data is returned. The data returned by a read command appears after the echo of the last byte of the command.

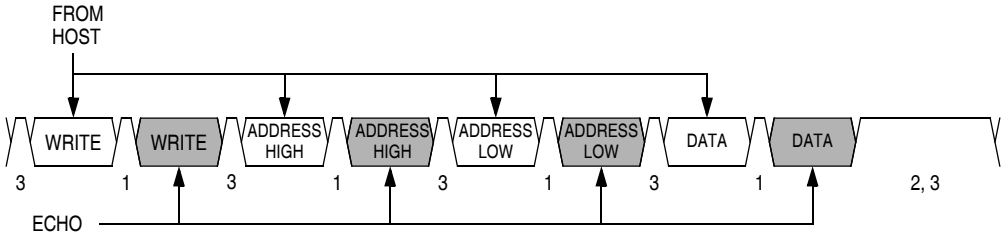
NOTE

Wait one bit time after each echo before sending the next byte.



Notes:
1 = Echo delay, approximately 2 bit times
2 = Data return delay, approximately 2 bit times
3 = Cancel command delay, 11 bit times
4 = Wait 1 bit time before sending next byte.

Figure 19-15. Read Transaction



Notes:
1 = Echo delay, approximately 2 bit times
2 = Cancel command delay, 11 bit times
3 = Wait 1 bit time before sending next byte.

Figure 19-16. Write Transaction

Chapter 21
Ordering Information and Mechanical Specifications

21.1 Introduction

This section provides ordering information for the MC68HC908GZ8 along with the dimensions for:

- 20-pin small outline intergrated circuit (SOIC) — case 751D
- 20-pin plastic dual in-line package (PDIP) — case 738

The following figures show the latest package drawings at the time of this publication. To make sure that you have the latest package specifications, contact your local Freescale Semiconductor Sales Office.

21.2 MC Order Numbers

Table 21-1. MC Order Numbers

MC Order Number	Operating Temperature Range	Package
MC68HC908LB8CDWE	−40°C to +85°C	20-pin Small outline integrated circuit (SOIC)
MC68HC908LB8VDWE	−40°C to +105°C	
MC68HC908LB8MDWE	−40°C to +125°C	
MC68HC908LB8CPE	−40°C to +85°C	20-pin Plastic dual In-line package (PDIP)
MC68HC908LB8VPE	−40°C to +105°C	
MC68HC908LB8MPE	−40°C to +125°C	

Temperature and package designators:
C = –40°C to +85°C
V = –40°C to +105°C
M = –40°C to +125°C
DW = Small outline integrated circuit package (SOIC)
E = Leadfree
P = Plastic dual in-line package (PDIP)