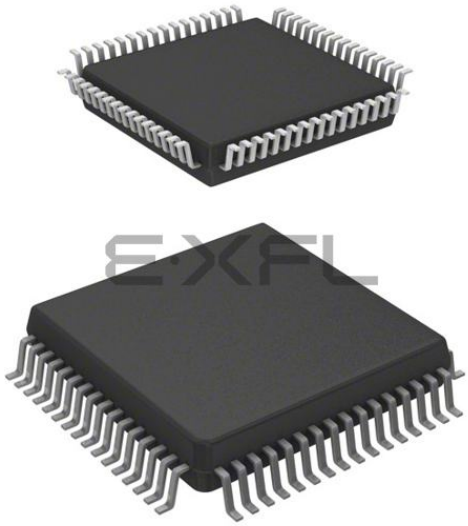

Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	HC08
Core Size	8-Bit
Speed	8.4MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	52
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 15x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908az60acfue

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0021	Keyboard Interrupt Enable Register (KBIER)	Read:	0	0	0	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
$0022	Timer A Counter Register High (TACNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$0023	Timer A Counter Register Low (TACNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
$0024	Timer A Modulo Register High (TAMODH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$0025	Timer A Modulo Register Low (TAMODL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
$0026	Timer A Channel 0 Status and Control Register (TASC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
$0027	Timer A Channel 0 Register High (TACH0H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$0028	Timer A Channel 0 Register Low (TACH0L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
$0029	Timer A Channel 1 Status and Control Register (TASC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0		R					
$002A	Timer A Channel 1 Register High (TACH1H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$002B	Timer A Channel 1 Register Low (TACH1L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
$002C	Timer A Channel 2 Status and Control Register (TASC2)	Read:	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	TOV2	CH2MAX
		Write:	0							
$002D	Timer A Channel 2 Register High (TACH2H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$002E	Timer A Channel 2 Register Low (TACH2L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
$002F	Timer A Channel 3 Status and Control Register (TASC3)	Read:	CH3F	CH3IE	0	MS3A	ELS3B	ELS3A	TOV3	CH3MAX
		Write:	0		R					
$0030	Timer A Channel 3 Register High (TACH3H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$0031	Timer A Channel 3 Register Low (TACH3L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								

= Unimplemented R = Reserved

Figure 2-2. I/O Data, Status and Control Registers (Sheet 3 of 5)

Chapter 3
Random-Access Memory (RAM)

3.1 Introduction

This chapter describes the 2048 bytes of random-access memory (RAM).

3.2 Functional Description

Addresses $0050 through $044F and $0A00 through $0DFF are RAM locations. The location of the stack RAM is programmable with the reset stack pointer instruction (RSP). The 16-bit stack pointer allows the stack RAM to be anywhere in the 64K-byte memory space.

NOTE

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

Within page zero are 176 bytes of RAM. Because the location of the stack RAM is programmable, all page zero RAM locations can be used for input/output (I/O) control and user data or code. When the stack pointer is moved from its reset location at $00FF, direct addressing mode instructions can access all page zero RAM locations efficiently. Page zero RAM, therefore, provides ideal locations for frequently accessed global variables.

Before processing an interrupt, the CPU uses five bytes of the stack to save the contents of the CPU registers.

NOTE

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

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

NOTE

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

EEBP[3:0] — EEPROM-1 Block Protection Bits

These bits prevent blocks of EEPROM-1 array from being programmed or erased.

1 = EEPROM-1 array block is protected

0 = EEPROM-1 array block is unprotected

Block Number (EEBPx)	Address Range
EEBP0	$0800–$087F
EEBP1	$0880–$08FF
EEBP2	$0900–$097F
EEBP3	$0980–$09FF

Table 6-4. EEPROM-1 Block Protect and Security Summary

Address Range	EEBPx	EEPRTCT = 1	EEPRTCT = 0
$0800 - $087F	EEBP0 = 0	Byte Programming Available Bulk, Block and Byte Erasing Available	Byte Programming Available Only Byte Erasing Available
	EEBP0 = 1	Protected	Protected
$0880 - $08EF	EEBP1 = 0	Byte Programming Available Bulk, Block and Byte Erasing Available	Byte Programming Available Only Byte Erasing Available
	EEBP1 = 1	Protected	Protected
$08F0 - $08FF	EEBP1 = 0	Byte Programming Available Bulk, Block and Byte Erasing Available	Secured (No Programming or Erasing)
	EEBP1 = 1	Protected	
$0900 - $097F	EEBP2 = 0	Byte Programming Available Bulk, Block and Byte Erasing Available	Byte Programming Available Only Byte Erasing Available
	EEBP2 = 1	Protected	Protected
$0980 - $09FF	EEBP3 = 0	Byte Programming Available Bulk, Block and Byte Available	Byte Programming Available Only Byte Erasing Available
	EEBP3 = 1	Protected	Protected

Chapter 8
Central Processor Unit (CPU)

8.1 Introduction

The M68HC08 CPU (central processor unit) is an enhanced and fully object-code-compatible version of the M68HC05 CPU. The *CPU08 Reference Manual* (document order number CPU08RM/AD) contains a description of the CPU instruction set, addressing modes, and architecture.

8.2 Features

Features of the CPU include:

- Object code fully upward-compatible with M68HC05 Family
- 16-bit stack pointer with stack manipulation instructions
- 16-bit index register with x-register manipulation instructions
- 8-MHz CPU internal bus frequency
- 64-Kbyte program/data memory space
- 16 addressing modes
- Memory-to-memory data moves without using accumulator
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- Enhanced binary-coded decimal (BCD) data handling
- Modular architecture with expandable internal bus definition for extension of addressing range beyond 64 Kbytes
- Low-power stop and wait modes

8.3 CPU Registers

Figure 8-1 shows the five CPU registers. CPU registers are not part of the memory map.

Table 8-1. Instruction Set Summary (Sheet 3 of 6)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
CLR *opr* CLRA CLRX CLRH CLR *opr*,X CLR ,X CLR *opr*,SP	Clear	M ← $00 A ← $00 X ← $00 H ← $00 M ← $00 M ← $00 M ← $00	0	–	–	0	1	–	DIR INH INH INH IX1 IX SP1	3F 4F 5F 8C 6F 7F 9E6F	dd ff ff	3 1 1 1 3 2 4
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X CMP *opr*,SP CMP *opr*,SP	Compare A with M	(A) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A1 B1 C1 D1 E1 F1 9EE1 9ED1	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
COM *opr* COMA COMX COM *opr*,X COM ,X COM *opr*,SP	Complement (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (M) X ← ($\overline{X}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	0	–	–	↕	↕	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd ff ff	4 1 1 4 3 5
CPHX #*opr* CPHX *opr*	Compare H:X with M	(H:X) – (M:M + 1)	↕	–	–	↕	↕	↕	IMM DIR	65 75	ii ii+1 dd	3 4
CPX #*opr* CPX *opr* CPX *opr* CPX ,X CPX *opr*,X CPX *opr*,X CPX *opr*,SP CPX *opr*,SP	Compare X with M	(X) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A3 B3 C3 D3 E3 F3 9EE3 9ED3	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
DAA	Decimal Adjust A	$(A)_{10}$	U	–	–	↕	↕	↕	INH	72		2
DBNZ *opr,rel* DBNZA *rel* DBNZX *rel* DBNZ *opr*,X,*rel* DBNZ X,*rel* DBNZ *opr*,SP,*rel*	Decrement and Branch if Not Zero	A ← (A) – 1 or M ← (M) – 1 or X ← (X) – 1 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 4 + *rel* ? (result) ≠ 0	–	–	–	–	–	–	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	5 3 3 5 4 6
DEC *opr* DECA DECX DEC *opr*,X DEC ,X DEC *opr*,SP	Decrement	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1 M ← (M) – 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff	4 1 1 4 3 5
DIV	Divide	A ← (H:A)/(X) H ← Remainder	–	–	–	–	↕	↕	INH	52		7
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X EOR *opr*,SP EOR *opr*,SP	Exclusive OR M with A	A ← (A ⊕ M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A8 B8 C8 D8 E8 F8 9EE8 9ED8	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
INC *opr* INCA INCX INC *opr*,X INC ,X INC *opr*,SP	Increment	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1 M ← (M) + 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff	4 1 1 4 3 5

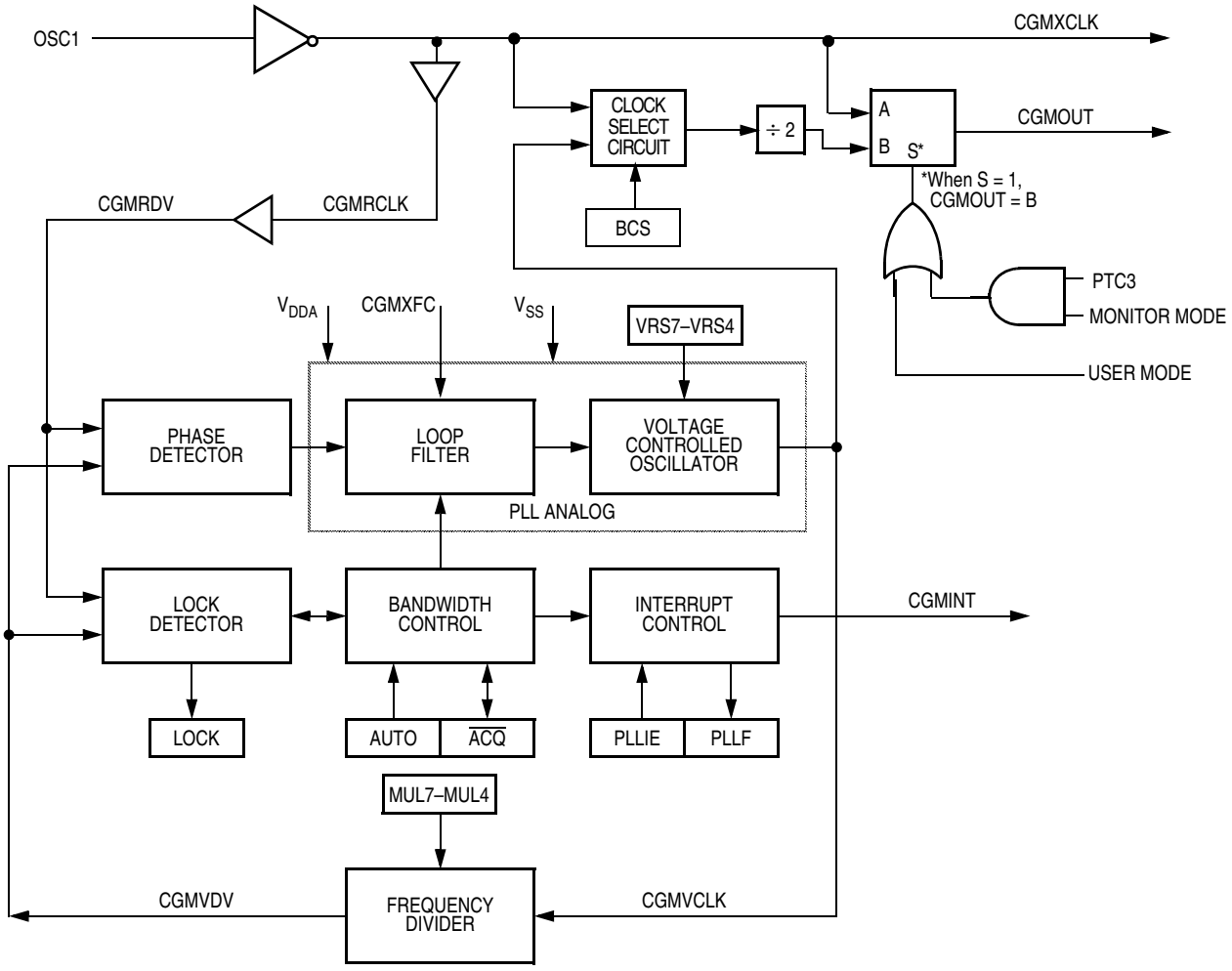

Figure 10-1. CGM Block Diagram

LVIRST — LVI Reset Enable Bit

LVIRST enables the reset signal from the LVI module. (See Chapter 16 Low-Voltage Inhibit (LVI)).

1 = LVI module resets enabled
0 = LVI module resets disabled

LVIPWR — LVI Power Enable Bit

LVIPWR enables the LVI module. (See Chapter 16 Low-Voltage Inhibit (LVI)).

1 = LVI module power enabled
0 = LVI module power disabled

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. (See 9.6.2 Stop Mode).

1 = Stop mode recovery after 32 CGMXCLK cycles
0 = Stop mode recovery after 4096 CGMXCLK cycles

NOTE

If using an external crystal oscillator, do not set the SSREC bit.

COPL — COP Long Timeout

COPL enables the shorter COP timeout period. (See Chapter 15 Computer Operating Properly (COP)).

1 = COP timeout period is $2^{13} – 2^4$ CGMXCLK cycles
0 = COP timeout period is $2^{18} – 2^4$ CGMXCLK cycles

STOP — STOP Instruction Enable Bit

STOP enables the STOP instruction.

1 = STOP instruction enabled
0 = STOP instruction treated as illegal opcode

COPD — COP Disable Bit

COPD disables the COP module. (See Chapter 15 Computer Operating Properly (COP)).

1 = COP module disabled
0 = COP module enabled

14.3.2 Data Format

Communication with the monitor ROM is in standard non-return-to-zero (NRZ) mark/space data format. (See Figure 14-2 and Figure 14-3.)

The data transmit and receive rate can be anywhere up to 28.8 kBaud. Transmit and receive baud rates must be identical.

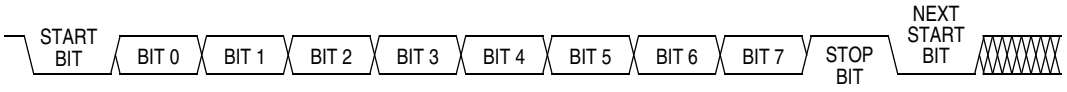

Figure 14-2. Monitor Data Format

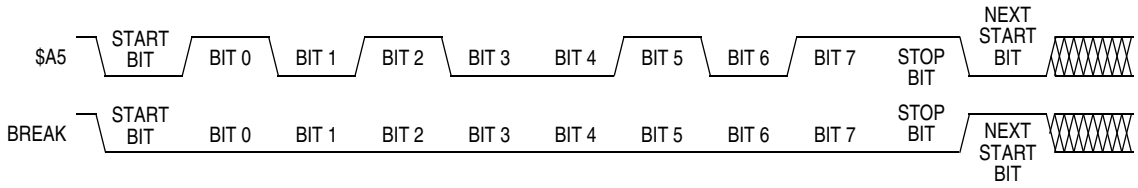

Figure 14-3. Sample Monitor Waveforms

14.3.3 Echoing

As shown in Figure 14-4, the monitor ROM immediately echoes each received byte back to the PTA0 pin for error checking.

Any result of a command appears after the echo of the last byte of the command.

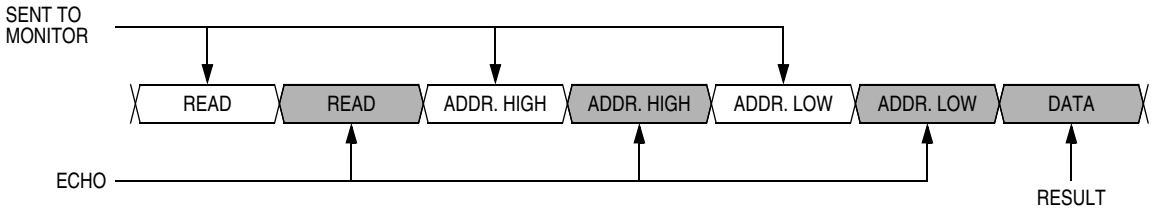

Figure 14-4. Read Transaction

14.3.4 Break Signal

A start bit followed by nine low bits is a break signal. (See Figure 14-5). When the monitor receives a break signal, it drives the PTA0 pin high for the duration of two bits before echoing the break signal.

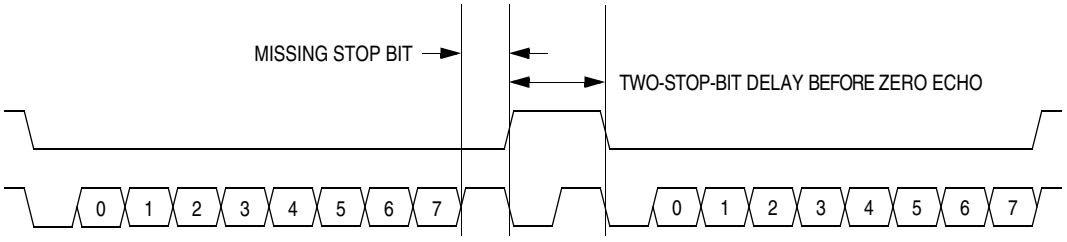

Figure 14-5. Break Transaction

14.3.7 MC68HC908AZ60A Baud Rate

The MC68HC908AZ60A features a monitor mode which is optimised to operate with either a 4.9152 MHz crystal clock source (or multiples of 4.9152 MHz) or a 4 MHz crystal (or multiples of 4 MHz). This supports designs which use the MSCAN module, which is generally clocked from a 4 MHz, 8 MHz or 16 MHZ internal reference clock. The table below outlines the available baud rates for a range of crystals and how they can match to a PC baud rate.

Table 14-10 MC68HC908AZ60A Monitor Baud Rate Selection

	Baud rate		Closest PC baud PC		Error %	
Clock freq	PTC3=0	PTC3=1	PTC3=0	PTC3=1	PTC3=0	PTC3=1
32kHz	57.97	28.98	57.6	28.8	0.64	0.63
1MHz	1811.59	905.80	1800	900	0.64	0.64
2MHz	3623.19	1811.59	3600	1800	0.64	0.64
4MHz	7246.37	3623.19	7200	3600	0.64	0.64
4.194MHz	7597.83	3798.91	7680	3840	1.08	1.08
4.9152MHz	8904.35	4452.17	8861	4430	0.49	0.50
8MHz	14492.72	7246.37	14400	7200	0.64	0.64
16MHz	28985.51	14492.75	28800	14400	0.64	0.64

14.3.8 Security

A security feature discourages unauthorized reading of FLASH locations while in monitor mode. The host can bypass the security feature at monitor mode entry by sending eight security bytes that match the bytes at locations $FFF6–$FFFD. Locations $FFF6–$FFFD contain user-defined data.

NOTE

Do not leave locations $FFF6–$FFFD blank. For security reasons, program locations $FFF6–$FFFD even if they are not used for vectors. If FLASH is unprogrammed, the eight security byte values to be sent are $FF, the unprogrammed state of FLASH.

During monitor mode entry, the MCU waits after the power-on reset for the host to send the eight security bytes on pin PA0.

18.4.1 Data Format

The SCI uses the standard non-return-to-zero mark/space data format illustrated in Figure 18-3.

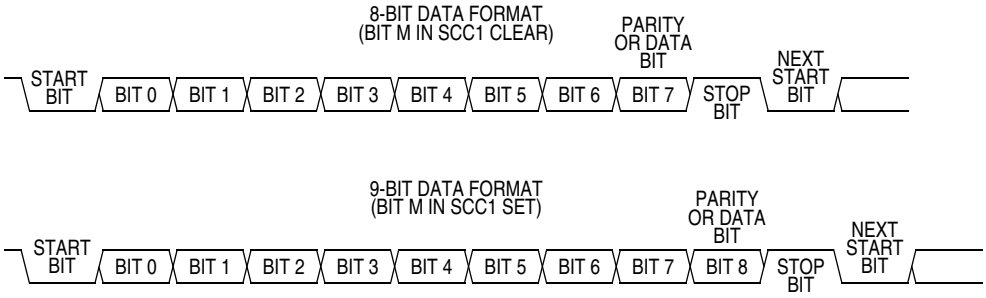

Figure 18-3. SCI Data Formats

18.4.2 Transmitter

Figure 18-4 shows the structure of the SCI transmitter.

18.4.2.1 Character Length

The transmitter can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCC1) determines character length. When transmitting 9-bit data, bit T8 in SCI control register 3 (SCC3) is the ninth bit (bit 8).

18.4.2.2 Character Transmission

During an SCI transmission, the transmit shift register shifts a character out to the TxD pin. The SCI data register (SCDR) is the write-only buffer between the internal data bus and the transmit shift register. To initiate an SCI transmission:

1. Enable the SCI by writing a 1 to the enable SCI bit (ENSCI) in SCI control register 1 (SCC1).
2. Enable the transmitter by writing a 1 to the transmitter enable bit (TE) in SCI control register 2 (SCC2).
3. Clear the SCI transmitter empty bit (SCTE) by first reading SCI status register 1 (SCS1) and then writing to the SCDR.
4. Repeat step 3 for each subsequent transmission.

At the start of a transmission, transmitter control logic automatically loads the transmit shift register with a preamble of 1s. After the preamble shifts out, control logic transfers the SCDR data into the transmit shift register. A 0 start bit automatically goes into the least significant bit position of the transmit shift register. A 1 stop bit goes into the most significant bit position.

The SCI transmitter empty bit, SCTE, in SCS1 becomes set when the SCDR transfers a byte to the transmit shift register. The SCTE bit indicates that the SCDR can accept new data from the internal data bus. If the SCI transmit interrupt enable bit, SCTIE, in SCC2 is also set, the SCTE bit generates a transmitter CPU interrupt request.

When the transmit shift register is not transmitting a character, the TxD pin goes to the idle condition, 1. If at any time software clears the ENSCI bit in SCI control register 1 (SCC1), the transmitter and receiver relinquish control of the port E pins.

The maximum frequency of the SPSCK for an SPI configured as a slave is the bus clock speed, which is twice as fast as the fastest master SPSCK clock that can be generated. The frequency of the SPSCK for an SPI configured as a slave does not have to correspond to any SPI baud rate. The baud rate only controls the speed of the SPSCK generated by an SPI configured as a master. Therefore, the frequency of the SPSCK for an SPI configured as a slave can be any frequency less than or equal to the bus speed.

When the master SPI starts a transmission, the data in the slave shift register begins shifting out on the MISO pin. The slave can load its shift register with a new byte for the next transmission by writing to its transmit data register. The slave must write to its transmit data register at least one bus cycle before the master starts the next transmission. Otherwise the byte already in the slave shift register shifts out on the MISO pin. Data written to the slave shift register during a a transmission remains in a buffer until the end of the transmission.

When the clock phase bit (CPHA) is set, the first edge of SPSCK starts a transmission. When CPHA is clear, the falling edge of $\overline{SS}$ starts a transmission. See 19.5 Transmission Formats.

If the write to the data register is late, the SPI transmits the data already in the shift register from the previous transmission.

NOTE

To prevent SPSCK from appearing as a clock edge, SPSCK must be in the proper idle state before the slave is enabled.

19.5 Transmission Formats

During an SPI transmission, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). A serial clock line synchronizes shifting and sampling on the two serial data lines. A slave select line allows individual selection of a slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. On a master SPI device, the slave select line can be used optionally to indicate a multiple-master bus contention.

19.5.1 Clock Phase and Polarity Controls

Software can select any of four combinations of serial clock (SCK) phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, which selects an active high or low clock and has no significant effect on the transmission format.

The clock phase (CPHA) control bit (SPCR) selects one of two fundamentally different transmission formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

NOTE

Before writing to the CPOL bit or the CPHA bit (SPCR), disable the SPI by clearing the SPI enable bit (SPE).

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 two-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.

Since the SPTE bit cannot be cleared during a break with the BCFE bit cleared, a write to the data register in break mode will not initiate a transmission nor will this data be transferred into the shift register. Therefore, a write to the SPDR in break mode with the BCFE bit cleared has no effect.

19.12 I/O Signals

The SPI module has four I/O pins and shares three of them with a parallel I/O port.

- MISO — Data received
- MOSI — Data transmitted
- SPSCK — Serial clock
- $\overline{SS}$ — Slave select
- V_{SS} — Clock ground

The SPI has limited inter-integrated circuit (I^2C) capability (requiring software support) as a master in a single-master environment. To communicate with I^2C peripherals, MOSI becomes an open-drain output when the SPWOM bit in the SPI control register is set. In I^2C communication, the MOSI and MISO pins are connected to a bidirectional pin from the I^2C peripheral and through a pullup resistor to V_{DD}.

19.12.1 MISO (Master In/Slave Out)

MISO is one of the two SPI module pins that transmit serial data. In full duplex operation, the MISO pin of the master SPI module is connected to the MISO pin of the slave SPI module. The master SPI simultaneously receives data on its MISO pin and transmits data from its MOSI pin.

Slave output data on the MISO pin is enabled only when the SPI is configured as a slave. The SPI is configured as a slave when its SPMSTR bit is 0 and its $\overline{SS}$ pin is low. To support a multiple-slave system, a high on the $\overline{SS}$ pin puts the MISO pin in a high-impedance state.

When enabled, the SPI controls data direction of the MISO pin regardless of the state of the data direction register of the shared I/O port.

19.12.2 MOSI (Master Out/Slave In)

MOSI is one of the two SPI module pins that transmit serial data. In full duplex operation, the MOSI pin of the master SPI module is connected to the MOSI pin of the slave SPI module. The master SPI simultaneously transmits data from its MOSI pin and receives data on its MISO pin.

When enabled, the SPI controls data direction of the MOSI pin regardless of the state of the data direction register of the shared I/O port.

19.12.3 SPSCK (Serial Clock)

The serial clock synchronizes data transmission between master and slave devices. In a master MCU, the SPSCK pin is the clock output. In a slave MCU, the SPSCK pin is the clock input. In full duplex operation, the master and slave MCUs exchange a byte of data in eight serial clock cycles.

Use the following 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.

20.3.4.2 Buffered PWM Signal Generation

Channels 0 and 1 can be linked to form a buffered PWM channel whose output appears on the PTF4/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 PTF4/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, PTF5/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.

When ELSxB:A = 00, this read/write bit selects the initial output level of the TBCHx pin once PWM, input capture or output compare operation is enabled (see Table 20-2). Reset clears the MSxA bit.

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).

ELSxB and ELSxA — Edge/Level Select Bits

When channel x is an input capture channel, these read/write bits control the active edge-sensing logic on channel x.

When channel x is an output compare channel, ELSxB and ELSxA control the channel x output behavior when an output compare occurs.

When ELSxB and ELSxA are both clear, channel x is not connected to port F and pin PTFx/TBCHx is available as a general-purpose I/O pin. However, channel x is at a state determined by these bits and becomes transparent to the respective pin when PWM, input capture, or output compare mode is enabled. Table 20-2 shows how ELSxB and ELSxA work. Reset clears the ELSxB and ELSxA bits.

Table 20-2. Mode, Edge, and Level Selection

MSxB	MSxA	ELSxB	ELSxA	Mode	Configuration
X	0	0	0	Output preset	Pin under port control; initial output level high
X	1	0	0		Pin under port control; initial output level low
0	0	0	1	Input capture	Capture on rising edge only
0	0	1	0		Capture on falling edge only
0	0	1	1		Capture on rising or falling edge
0	1	0	0	Output compare or PWM	Software compare only
0	1	0	1		Toggle output on compare
0	1	1	0		Clear output on compare
0	1	1	1		Set output on compare
1	X	0	1	Buffered output compare or buffered PWM	Toggle output on compare
1	X	1	0		Clear output on compare
1	X	1	1		Set output on compare

NOTE

Before enabling a TIMB channel register for input capture operation, make sure that the PTFx/TBCHx 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 TIMB 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 TIMB counter overflow.
0 = Channel x pin does not toggle on TIMB counter overflow.

NOTE

When TOVx is set, a TIMB 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%. As Figure 20-8 shows, the CHxMAX bit takes effect in the cycle after it is set or cleared. The output stays at the 100% duty cycle level until the cycle after CHxMAX is cleared.

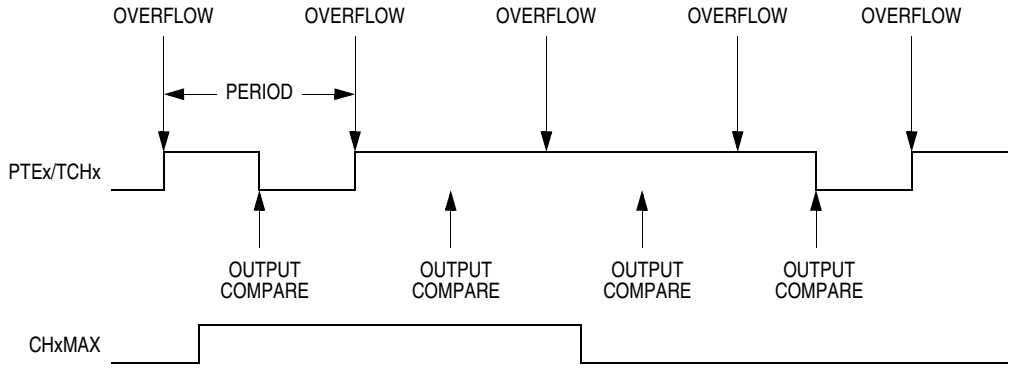

Figure 20-8. CHxMAX Latency

20.8.5 TIMB Channel Registers

These read/write registers contain the captured TIMB counter value of the input capture function or the output compare value of the output compare function. The state of the TIMB channel registers after reset is unknown.

In input capture mode (MSxB–MSxA = 0:0) reading the high byte of the TIMB channel x registers (TBCHxH) inhibits input captures until the low byte (TBCHxL) is read.

In output compare mode (MSxB–MSxA ≠ 0:0) writing to the high byte of the TIMB channel x registers (TBCHxH) inhibits output compares and the CHxF bit until the low byte (TBCHxL) is written.

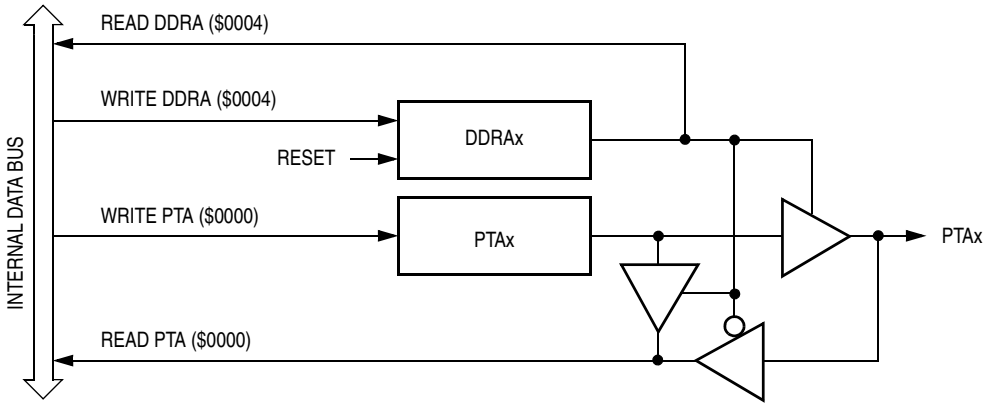

Figure 22-4. Port A I/O Circuit

When bit DDRAx is a logic 1, reading address $0000 reads the PTAx data latch. When bit DDRAx is a logic 0, reading address $0000 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 22-1 summarizes the operation of the port A pins.

Table 22-1. Port A Pin Functions

DDRA Bit	PTA Bit	I/O Pin Mode	Accesses to DDRA	Accesses to PTA	
			Read/Write	Read	Write
0	X	Input, Hi-Z	DDRA[7:0]	Pin	PTA[7:0](1)
1	X	Output	DDRA[7:0]	PTA[7:0]	PTA[7:0]

X = don't care
Hi-Z = high impedance
1. Writing affects data register, but does not affect input.

25.3.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in 25.3.4 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 TIMA channel registers.

An unsynchronized write to the TIMA 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 TIMA overflow interrupt routine to write a new, smaller pulse width value may cause the compare to be missed. The TIMA may pass the new value before it is written to the TIMA channel registers.

Use the following 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 TIMA overflow interrupts and write the new value in the TIMA overflow interrupt routine. The TIMA 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.

25.3.4.2 Buffered PWM Signal Generation

Channels 0 and 1 can be linked to form a buffered PWM channel whose output appears on the PTE2/TACH0 pin. The TIMA channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS0B bit in TIMA channel 0 status and control register (TASC0) links channel 0 and channel 1. The TIMA channel 0 registers initially control the pulse width on the PTE2/TACH0 pin. Writing to the TIMA channel 1 registers enables the TIMA channel 1 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIMA channel registers (0 or 1) that control the pulse width are the ones written to last. TASC0 controls and monitors the buffered PWM function and TIMA channel 1 status and control register (TASC1) is unused. While the MS0B bit is set, the channel 1 pin, PTE3/TACH1, is available as a general-purpose I/O pin.

Channels 2 and 3 can be linked to form a buffered PWM channel whose output appears on the PTF0/TACH2 pin. The TIMA channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS2B bit in TIMA channel 2 status and control register (TASC2) links channel 2 and channel 3. The TIMA channel 2 registers initially control the pulse width on the PTF0/TACH2 pin. Writing to the TIMA channel 3 registers enables the TIMA channel 3 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIMA channel registers

A.1.4 CONFIG-2

CONFIG-2 register $FE09 has 2 new bits activated. Bit 3 is now a silicon hard set bit, which identifies this new A-suffix silicon (1) from the previous non-A suffix silicon (0). Bit 7 is now an EEPROM time base divider clock select bit selecting the reference clock source for the EEPROM time base divider module (refer to EEPROM changes described above).

A.1.5 Keyboard Interrupt

The keyboard module is now a feature of the MC68HC908AS60A in 64-qfp package whereas previously it was only a feature of the AZ device. Vector addresses $FFD2 and $FFD3 are now in the AS memory map in support of this option.

A.1.6 Current Consumption

Current consumption will be significantly lower in many applications. Although maximum specifications are still very dependent upon fabrication process variation and configuration of the MCU in the target application, additional values have been added to the I_{DD} specifications to provide typical current consumption data. Please see Chapter 28 Electrical Specifications for further details.

A.1.7 Illegal Address Reset

Only an opcode fetch from an illegal address will generate an illegal address reset. Data fetches from unmapped addresses will not generate a reset.

A.1.8 Monitor Mode Entry and COP Disable Voltage

The monitor mode entry and COP disable voltage specifications (V_{HI}) have been increased. Please see Chapter 28 Electrical Specifications for details.

A.1.9 Low-Voltage Inhibit (LVI)

The Low-Voltage Inhibit (LVI) specifications for trip and recovery voltage (V_{LVI}) have been altered based upon module performance on silicon. Please see for Chapter 28 Electrical Specifications details.

$0A00 ↓ $0DFF	RAM-2 1024 BYTES
$0E00 ↓ $7FFF	FLASH-2 29,184 BYTES
$8000 ↓ $FDFF	FLASH-1 32,256 BYTES
$FE00	SIM BREAK STATUS REGISTER (SBSR)
$FE01	SIM RESET STATUS REGISTER (SRSR)
$FE02	RESERVED
$FE03	SIM BREAK FLAG CONTROL REGISTER (SBFCR)
$FE04	RESERVED
$FE05	RESERVED
$FE06	RESERVED
$FE07	RESERVED
$FE08	FLASH-2 CONTROL REGISTER (FL2CR)
$FE09	CONFIGURATION WRITE-ONCE REGISTER (CONFIG-2)
$FE0A	RESERVED
$FE0B	CONFIGURATION WRITE-ONCE REGISTER (CONFIG-3)
$FE0C	BREAK ADDRESS REGISTER HIGH (BRKH)
$FE0D	BREAK ADDRESS REGISTER LOW (BRKL)
$FE0E	BREAK STATUS AND CONTROL REGISTER (BSCR)
$FE0F	LVI STATUS REGISTER (LVISR)
$FE10	EEPROM-1EEDIVH NONVOLATILE REGISTER (EE1DIVHNVR)
$FE11	EEPROM-1EEDIVL NONVOLATILE REGISTER (EE1DIVLNVR)
$FE12	RESERVED
$FE13	RESERVED
$FE14	RESERVED
$FE15	RESERVED
$FE16	RESERVED
$FE17	RESERVED
$FE18	RESERVED
$FE19	RESERVED
$FE1A	EEPROM-1 EE DIVIDER HIGH REGISTER(EE1DIVH)
$FE1B	EEPROM-1 EE DIVIDER LOW REGISTER(EE1DIVL)
$FE1C	EEPROM-1 EEPROM NONVOLATILE REGISTER (EE1NVR)
$FE1D	EEPROM-1 EEPROM CONTROL REGISTER (EE1CR)
$FE1E	RESERVED
$FE1F	EEPROM-1 EEPROM ARRAY CONFIGURATION REGISTER (EE1ACR)

Figure B-1. MC68HC908AZ60E Memory Map (Sheet 2 of 3)

$FE20 ↓ $FF1F	MONITOR ROM 256BYTES
$FF20 ↓ $FF6F	UNIMPLEMENTED 80 BYTES
$FF70	EEPROM-2 EEDIVH NONVOLATILE REGISTER (EE2DIVHNVR)
$FF71	EEPROM-2 EEDIVL NONVOLATILE REGISTER (EE2DIVLNVR)
$FF72	RESERVED
$FF73	RESERVED
$FF74	RESERVED
$FF75	RESERVED
$FF76	RESERVED
$FF77	RESERVED
$FF78	RESERVED
$FF79	RESERVED
$FF7A	EEPROM-2 EE DIVIDER HIGH REGISTER (EE2DIVH)
$FF7B	EEPROM-2 EE DIVIDER LOW REGISTER (EE2DIVL)
$FF7C	EEPROM-2 EEPROM NONVOLATILE REGISTER (EE2NVR)
$FF7D	EEPROM-2 EEPROM CONTROL REGISTER (EE2CR)
$FF7E	RESERVED
$FF7F	EEPROM-2 EEPROM ARRAY CONFIGURATION REGISTER (EE2ACR)
$FF80	FLASH-1 BLOCK PROTECT REGISTER (FL1BPR)
$FF81	FLASH-2 BLOCK PROTECT REGISTER (FL2BPR)
$FF82 ↓ $FF87	RESERVED 6 BYTES
$FF88	FLASH-1 CONTROL REGISTER (FL1CR)
$FF89	RESERVED
$FF8A	RESERVED
$FF8B ↓ $FFCB	RESERVED 64 BYTES
$FFCC ↓ $FFFF	VECTORS 52 BYTES See Table B-2

Figure B-1. MC68HC908AZ60E Memory Map (Sheet 3 of 3)