

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, SPI
Peripherals	POR, PWM
Number of I/O	51
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x8b
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/mchc908ab32cfur2

4.8.1 FLASH Block Protect Register

The FLASH block protect register (FLBPR) is implemented as a byte within the FLASH memory, and therefore can only be written during a programming sequence of the FLASH memory. The value in this register determines the starting location of the protected range within the FLASH memory.

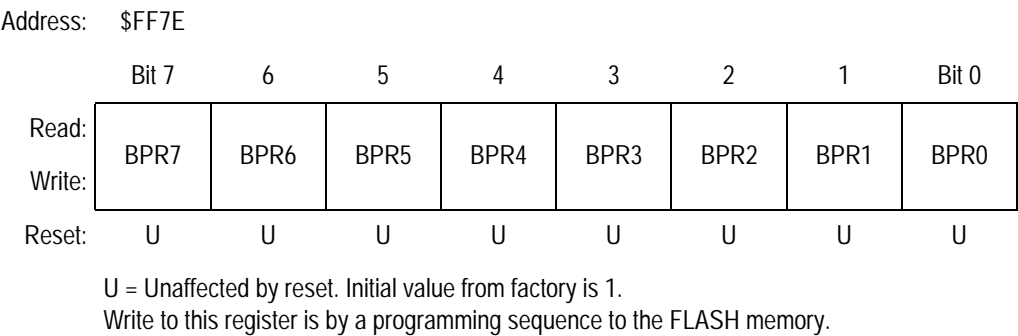


Figure 4-3. FLASH Block Protect Register (FLBPR)

BPR[7:0] — FLASH Block Protect Bits

These eight bits represent bits [14:7] of a 16-bit memory address. Bit-15 is logic 1 and bits [6:0] are logic 0s. The resultant 16-bit address is used for specifying the start address of the FLASH memory for block protection. The FLASH is protected from this start address to the end of FLASH memory, at \$FFFF. With this mechanism, the protect start address can be XX00 and XX80 (128 bytes page boundaries) within the FLASH memory.

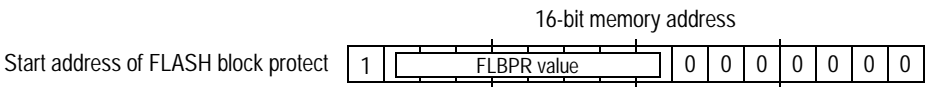


Figure 4-4. FLASH Block Protect Start Address

EEPGM — EEPROM Program/Erase Enable

This read/write bit enables the internal charge pump and applies the programming/erasing voltage to the EEPROM array if the EELAT bit is set and a write to a valid EEPROM location has occurred. Reset clears the EEPGM bit.

- 1 = EEPROM programming/erasing power switched on
- 0 = EEPROM programming/erasing power switched off

NOTE: Writing 0s to both the EELAT and EEPGM bits with a single instruction will only clear EEPGM. This is to allow time for the removal of high voltage.

5.11.2 EEPROM Array Configuration Register

The EEPROM array configuration register configures EEPROM security and EEPROM block protection.

This read-only register is loaded with the contents of the EEPROM non-volatile register (EENVR) after a reset.

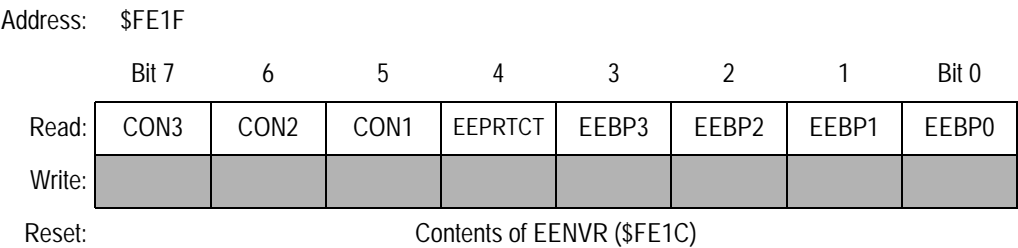


Figure 5-3. EEPROM Array Configuration Register (EEACR)

CON[3:1] — Unused

EEPRTCT — EEPROM Protection Bit

The EEPRTCT bit is used to enable the security feature in the EEPROM (see [5.7 EEPROM Security Options](#)).

- 1 = EEPROM security disabled
- 0 = EEPROM security enabled

Section 7. Central Processor Unit (CPU)

7.1 Contents

7.2	Introduction	89
7.3	Features	90
7.4	CPU Registers	90
7.4.1	Accumulator	91
7.4.2	Index Register	92
7.4.3	Stack Pointer	92
7.4.4	Program Counter	93
7.4.5	Condition Code Register	93
7.5	Arithmetic/Logic Unit (ALU)	96
7.6	Low-Power Modes	96
7.6.1	Wait Mode	96
7.6.2	Stop Mode	97
7.7	CPU During Break Interrupts	97
7.8	Instruction Set Summary	97
7.9	Opcode Map	97

7.2 Introduction

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

7.3 Features

- 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
- 64K-byte 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 64K-bytes
- Low-power stop and wait modes

7.4 CPU Registers

Figure 7-1 shows the five CPU registers. CPU registers are not part of the memory map.

8.5.1 SIM Counter during Power-On Reset

The power-on reset (POR) module detects power applied to the MCU. At power-on, the POR circuit asserts the signal PORRST. Once the SIM is initialized, it enables the clock generation module (CGM) to drive the bus clock state machine.

8.5.2 SIM Counter during Stop Mode Recovery

The SIM counter is also used for stop mode recovery. The STOP instruction clears the SIM counter. After an interrupt, break, or reset, the SIM senses the state of the short STOP recovery bit, SSREC, in the configuration register 1 (CONFIG1). If the SSREC bit is a logic 1, then the stop recovery is reduced from the normal delay of 4096 CGMXCLK cycles down to 32 CGMXCLK cycles. This is ideal for applications using canned oscillators that do not require long start-up times from stop mode. External crystal applications should use the full stop recovery time, that is, with SSREC cleared.

8.5.3 SIM Counter and Reset States

External reset has no effect on the SIM counter. (See [8.7.2 Stop Mode](#) for details.) The SIM counter is free-running after all reset states. (See [8.4.2 Active Resets from Internal Sources](#) for counter control and internal reset recovery sequences.)

8.6 Exception Control

Normal, sequential program execution can be changed in three different ways:

- Interrupts
 - Maskable hardware CPU interrupts
 - Non-maskable software interrupt instruction (SWI)
- Reset
- Break interrupts

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$001C	PLL Control Register (PCTL)	Read:	PLLIE	PLL F	PLL ON	BCS	1	1	1
		Write:							
		Reset:	0	0	1	0	1	1	1
\$001D	PLL Bandwidth Control Register (PBWC)	Read:	AUTO	LOCK	ACQ	XLD	0	0	0
		Write:							
		Reset:	0	0	0	0	0	0	0
\$001E	PLL Programming Register (PPG)	Read:	MUL7	MUL6	MUL5	MUL4	VRS7	VRS6	VRS5
		Write:							
		Reset:	0	1	1	0	0	1	1
									0

= Unimplemented

Figure 9-2. CGM I/O Register Summary

9.4.1 Crystal Oscillator Circuit

The crystal oscillator circuit consists of an inverting amplifier and an external crystal. The OSC1 pin is the input to the amplifier and the OSC2 pin is the output. The SIMOSCEN signal from the system integration module (SIM) enables the crystal oscillator circuit.

The CGMXCLK signal is the output of the crystal oscillator circuit and runs at a rate equal to the crystal frequency. CGMXCLK is then buffered to produce CGMRCLK, the PLL reference clock.

CGMXCLK can be used by other modules which require precise timing for operation. The duty cycle of CGMXCLK is not guaranteed to be 50% and depends on external factors, including the crystal and related external components.

An externally generated clock can also feed the OSC1 pin of the crystal oscillator circuit. For this configuration, the external clock should be connected to the OSC1 pin and the OSC2 pin allowed to float.

and require fast start-up. The following conditions apply when in manual mode:

- \overline{ACQ} is a writable control bit that controls the mode of the filter. Before turning on the PLL in manual mode, the \overline{ACQ} bit must be clear.
- Before entering tracking mode ($\overline{ACQ} = 1$), software must wait a given time, t_{ACQ} (see [9.10 Acquisition/Lock Time Specifications](#)), after turning on the PLL by setting PLLON in the PLL control register (PCTL).
- Software must wait a given time, t_{AL} , after entering tracking mode before selecting the PLL as the clock source to CGMOUT ($BCS = 1$).
- The LOCK bit is disabled.
- CPU interrupts from the CGM are disabled.

9.4.2.4 Programming the PLL

The following procedure shows how to program the PLL.

NOTE: *The round function in the following equations means that the real number should be rounded to the nearest integer number.*

1. Choose the desired bus frequency, f_{BUSDES} .
2. Calculate the desired VCO frequency (four times the desired bus frequency).

$$f_{VCLKDES} = 4 \times f_{BUSDES}$$

3. Choose a practical PLL reference frequency, f_{RCLK} .
4. Select a VCO frequency multiplier, N.

$$N = \text{round}\left(\frac{f_{VCLKDES}}{f_{RCLK}}\right)$$

The crystal loss detect function works only when the BCS bit is set, selecting CGMVCLK to drive CGMOUT. When BCS is clear, XLD always reads as 0.

Bits [3:0] — Reserved for test

These bits enable test functions not available in user mode. To ensure software portability from development systems to user applications, software should write zeros to Bits [3:0] whenever writing to PBWC.

9.6.3 PLL Programming Register (PPG)

The PLL programming register contains the programming information for the modulo feedback divider and the programming information for the hardware configuration of the VCO.

Address: \$001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MUL7	MUL6	MUL5	MUL4	VRS7	VRS6	VRS5	VRS4
Write:								
Reset:	0	1	1	0	0	1	1	0

Figure 9-8. PLL Programming Register (PPG)

MUL[7:4] — Multiplier Select Bits

These read/write bits control the modulo feedback divider that selects the VCO frequency multiplier, N. (See [9.4.2.1 PLL Circuits](#) and [9.4.2.4 Programming the PLL](#)). A value of \$0 in the multiplier select bits configures the modulo feedback divider the same as a value of \$1. Reset initializes these bits to \$6 to give a default multiply value of 6.

10.4.5 Commands

The monitor ROM uses these commands:

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

A sequence of IREAD or IWRITE commands can access a block of memory sequentially over the full 64k-byte memory map.

Table 10-3. READ (Read Memory) Command

Description	Read byte from memory
Operand	Specifies 2-byte address in high byte:low byte order
Data Returned	Returns contents of specified address
Opcode	\$4A
<div><p>Command Sequence</p><pre>graph LR S[SENT TO MONITOR] --> R1[READ] R1 --> R2[READ] R2 --> AH1[ADDRESS HIGH] AH1 --> AH2[ADDRESS HIGH] AH2 --> AL1[ADDRESS LOW] AL1 --> AL2[ADDRESS LOW] AL2 --> D[DATA] E[ECHO] --> R2 R2 --> R1 D --> R[RETURN]</pre></div>	

Table 10-8. RUN (Run User Program) Command

Description	Executes RTI instruction
Operand	None
Data Returned	None
Opcode	\$28
<p style="text-align: center;">Command Sequence</p>	

10.4.6 Baud Rate

With a 4.9152-MHz crystal and the PTC3 pin at logic 1 during reset, data is transferred between the monitor and host at 4800 baud. If the PTC3 pin is at logic 0 during reset, the monitor baud rate is 9600. When the CGM output, CGMOUT, is driven by the PLL, the baud rate is determined by the MUL[7:4] bits in the PLL programming register (PPG). (See [Section 9. Clock Generator Module \(CGM\)](#).)

Table 10-9. Monitor Baud Rate Selection

Monitor Baud Rate	VCO Frequency Multiplier (N)					
	1	2	3	4	5	6
4.9152 MHz	4800	9600	14,400	19,200	24,000	28,800
4.194 MHz	4096	8192	12,288	16,384	20,480	24,576

10.5 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.

During monitor mode entry, the MCU waits after the power-on reset for the host to send the eight security bytes on pin PTA0. If the received bytes match those at locations \$FFF6–\$FFFD, the host bypasses the security feature and can read all FLASH locations and execute code from FLASH. Security remains bypassed until a power-on reset occurs. If the reset was not a power-on reset, security remains bypassed and security code entry is not required. (See [Figure 10-6](#).)

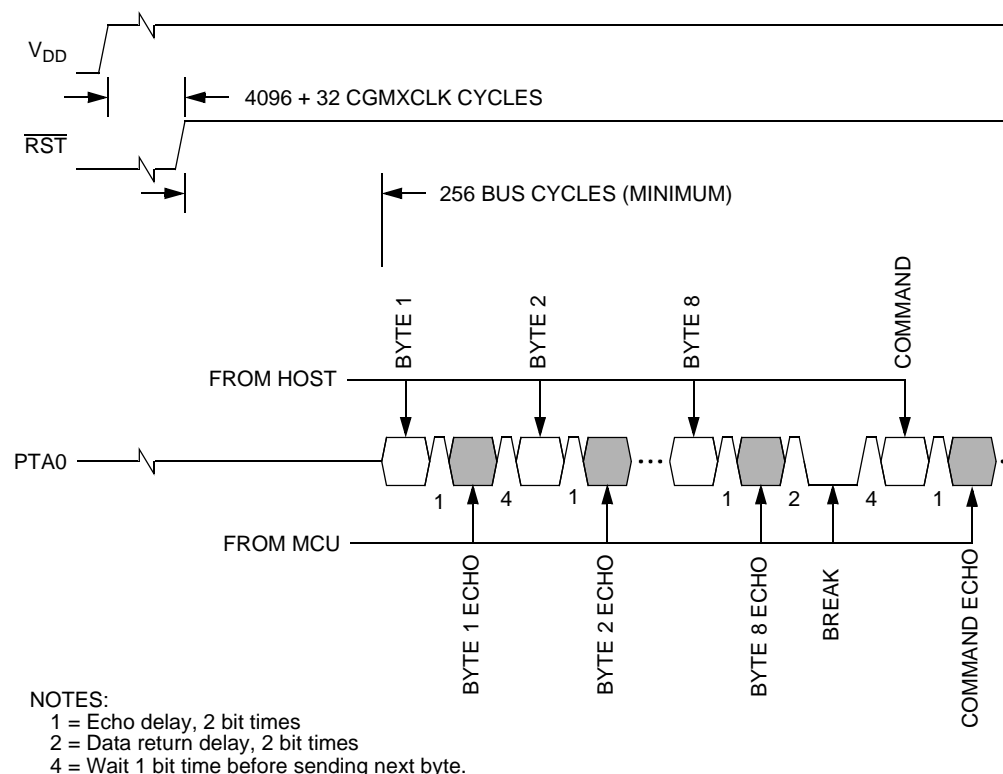


Figure 10-6. Monitor Mode Entry Timing

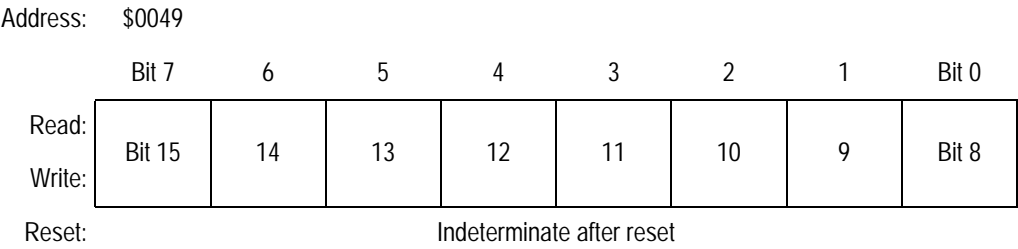


Figure 12-16. TIMB Channel 1 Register High (TBCH1H)

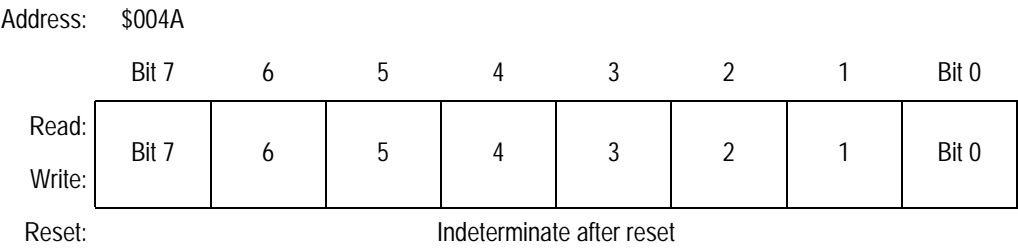


Figure 12-17. TIMB Channel 1 Register Low (TBCH1L)

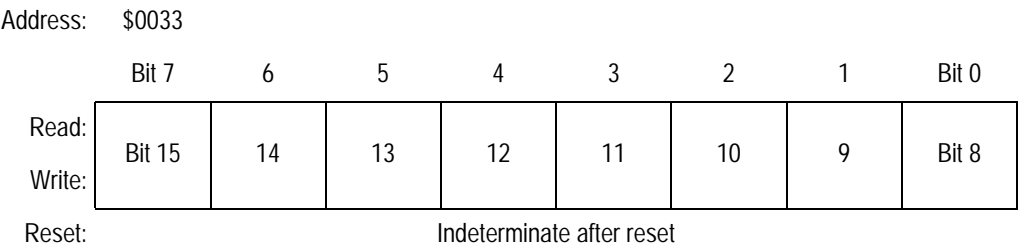


Figure 12-18. TIMB Channel 2 Register High (TBCH2H)

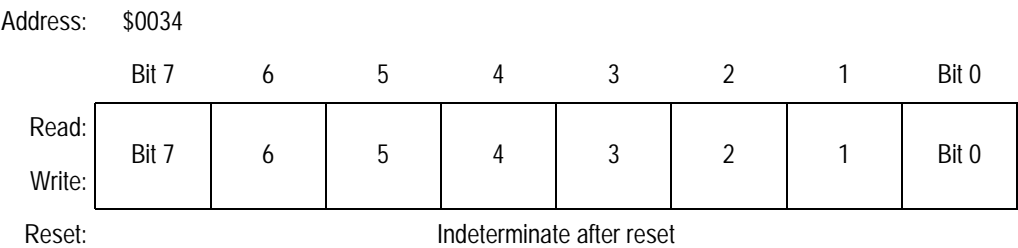


Figure 12-19. TIMB Channel 2 Register Low (TBCH2L)

PPS[2:0] — PIT Prescaler Select Bits

These read/write bits select one of the seven prescaler outputs as the input to the PIT counter as [Table 13-1](#) shows. Reset clears the PPS[2:0] bits.

Table 13-1. PIT Prescaler Selection

PPS2	PPS1	PPS0	PIT Clock Source
0	0	0	Internal Bus Clock ÷ 1
0	0	1	Internal Bus Clock ÷ 2
0	1	0	Internal Bus Clock ÷ 4
0	1	1	Internal Bus Clock ÷ 8
1	0	0	Internal Bus Clock ÷ 16
1	0	1	Internal Bus Clock ÷ 32
1	1	0	Internal Bus Clock ÷ 64
1	1	1	Internal Bus Clock ÷ 64

13.7.2 PIT Counter Registers

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

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

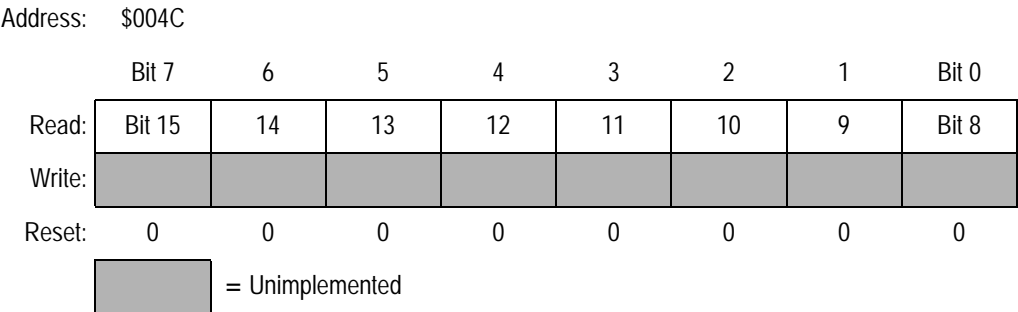


Figure 13-4. PIT Counter Register High (PCNTH)

14.8.2 ADC Data Register (ADR)

One 8-bit result register, ADC data register (ADR), is provided. This register is updated each time an ADC conversion completes.

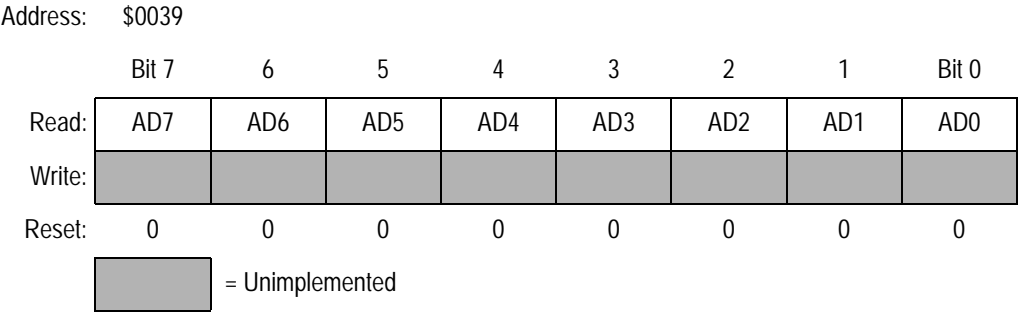


Figure 14-4. ADC Data Register (ADR)

14.8.3 ADC Clock Register (ADCLK)

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

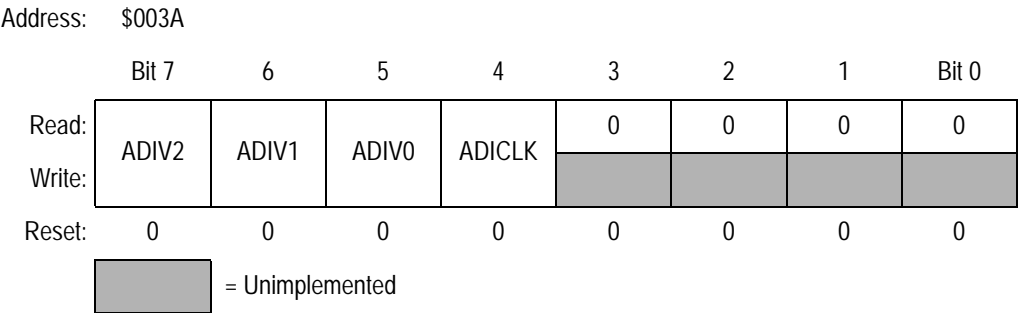


Figure 14-5. ADC Clock Register (ADCLK)

ADIV[2:0] — ADC Clock Prescaler Bits

ADIV[2:0] form a 3-bit field which selects the divide ratio used by the ADC to generate the internal ADC clock. [Table 14-2](#) shows the available clock configurations. The ADC clock should be set to approximately 1 MHz.

15.8.2	PTE1/RxD (Receive Data)	260
15.9	I/O Registers	261
15.9.1	SCI Control Register 1	261
15.9.2	SCI Control Register 2	264
15.9.3	SCI Control Register 3	267
15.9.4	SCI Status Register 1	269
15.9.5	SCI Status Register 2	273
15.9.6	SCI Data Register	274
15.9.7	SCI Baud Rate Register	275

15.2 Introduction

This section describes the serial communications interface (SCI) module, which allows high-speed asynchronous communications with peripheral devices and other MCUs.

NOTE: *References to DMA (direct-memory access) and associated functions are only valid if the MCU has a DMA module. This MCU does not have the DMA function. Any DMA-related register bits should be left in their reset state for normal MCU operation.*

15.3 Features

Features of the SCI module include the following:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- 32 programmable baud rates
- Programmable 8-bit or 9-bit character length
- Separately enabled transmitter and receiver
- Separate receiver and transmitter CPU interrupt requests
- Programmable transmitter output polarity

16.7 Queuing Transmission Data

The double-buffered transmit data register allows a data byte to be queued and transmitted. For an SPI configured as a master, a queued data byte is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag (SPTE) indicates when the transmit data buffer is ready to accept new data. Write to the transmit data register only when the SPTE bit is high. **Figure 16-8** shows the timing associated with doing back-to-back transmissions with the SPI (SPSCK has CPHA: CPOL = 1:0).

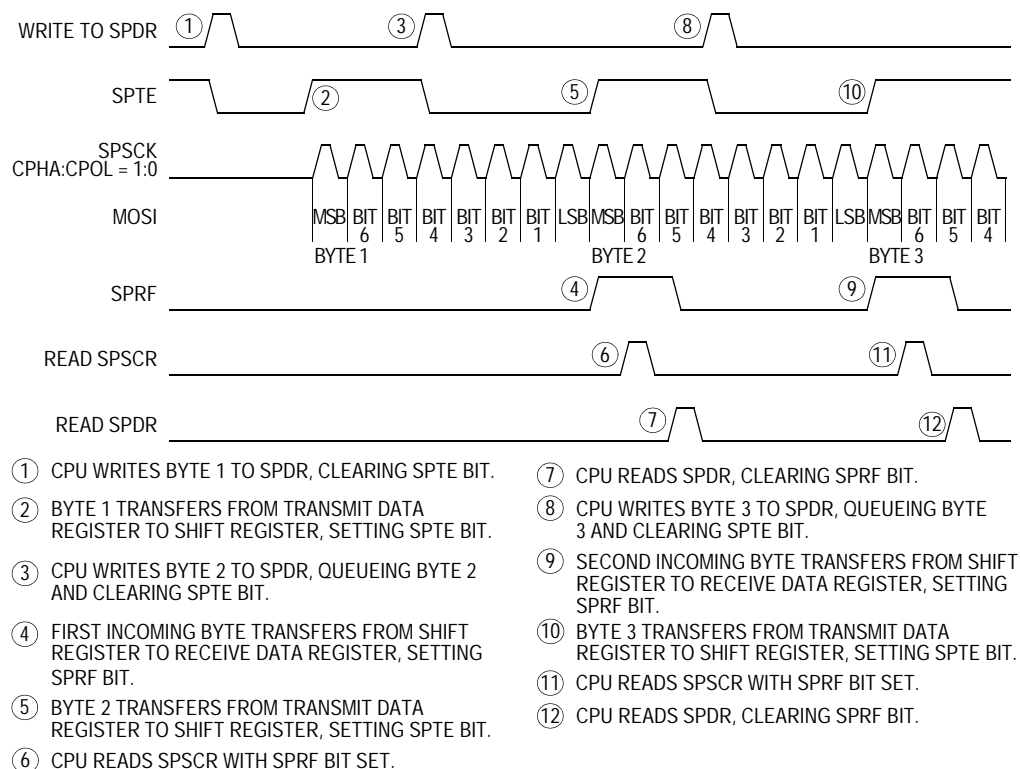


Figure 16-8. SPRF/SPTE CPU Interrupt Timing

The transmit data buffer allows back-to-back transmissions without the slave precisely timing its writes between transmissions as in a system with a single data buffer. Also, if no new data is written to the data buffer, the last value contained in the shift register is the next data word to be transmitted.

Section 17. Input/Output (I/O) Ports

17.1 Contents

17.2	Introduction	312
17.3	Port A	316
17.3.1	Port A Data Register (PTA)	316
17.3.2	Data Direction Register A (DDRA)	316
17.4	Port B	318
17.4.1	Port B Data Register (PTB)	318
17.4.2	Data Direction Register B (DDRB)	319
17.5	Port C	320
17.5.1	Port C Data Register (PTC)	320
17.5.2	Data Direction Register C (DDRC)	321
17.6	Port D	323
17.6.1	Port D Data Register (PTD)	323
17.6.2	Data Direction Register D (DDRD)	324
17.6.3	Port D Input Pullup Enable Register (PTDPUE)	325
17.7	Port E	326
17.7.1	Port E Data Register (PTE)	326
17.7.2	Data Direction Register E (DDRE)	328
17.8	Port F	329
17.8.1	Port F Data Register (PTF)	329
17.8.2	Data Direction Register F (DDRF)	330
17.8.3	Port F Input Pullup Enable Register (PTFPUE)	332
17.9	Port G	332
17.9.1	Port G Data Register (PTG)	332
17.9.2	Data Direction Register G (DDRG)	333
17.10	Port H	335
17.10.1	Port H Data Register (PTH)	335
17.10.2	Data Direction Register H (DDRH)	335

When DDRD_x is a logic 1, reading address \$0003 reads the PTD_x data latch. When DDRD_x is a logic 0, reading address \$0003 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit.

Table 17-5 summarizes the operation of the port D pins.

Table 17-5. Port D Pin Functions

DDRD Bit	PTD Bit	I/O Pin Mode	Accesses to DDRD	Accesses to PTD	
			Read/Write	Read	Write
0	X ⁽¹⁾	Input, Hi-Z ⁽²⁾	DDRD[7:0]	Pin	PTD[7:0] ⁽³⁾
1	X	Output	DDRD[7:0]	PTD[7:0]	PTD[7:0]

Notes:

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

17.6.3 Port D Input Pullup Enable Register (PTDPUE)

The port D input pullup enable register (PTDPUE) controls the input pullup device for each of the eight port D pins. Each bit is individually configurable and requires that the data direction register, DDRD, bit be configured as an input. Each pullup is automatically and dynamically disabled when a port bit's DDRD is configured for output mode.

Address: \$003D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTDPUE7	PTDPUE6	PTDPUE5	PTDPUE4	PTDPUE3	PTDPUE2	PTDPUE1	PTDPUE0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 17-14. Port D Input Pullup Enable Register (PTDPUE)

PTDPUE[7:0] — Port D Input Pullup Enable Bits

These writable bits are software programmable to enable pullup devices on an input port pin.

- 1 = Corresponding port D pin configured to have internal pullup
- 0 = Corresponding port D pin internal pullup disconnected

Section 20. Computer Operating Properly (COP)

20.1 Contents

20.2	Introduction	353
20.3	Functional Description	354
20.4	I/O Signals	355
20.4.1	CGMXCLK	355
20.4.2	STOP Instruction	355
20.4.3	COPCTL Write	355
20.4.4	Power-On Reset	355
20.4.5	Internal Reset	356
20.4.6	Reset Vector Fetch	356
20.4.7	COPD (COP Disable)	356
20.4.8	COPRS (COP Rate Select)	356
20.5	COP Control Register	357
20.6	Interrupts	357
20.7	Monitor Mode	357
20.8	Low-Power Modes	357
20.8.1	Wait Mode	358
20.8.2	Stop Mode	358
20.9	COP Module During Break Mode	358

20.2 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 clearing the COP counter periodically. The COP module can be disabled through the COPD bit in configuration register 1 (CONFIG1).

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a logic 1 to BRKA generates a break interrupt. Clear BRKA by writing a logic 0 to it before exiting the break routine. Reset clears the BRKA bit.

- 1 = (When read) Break address match
- 0 = (When read) No break address match

22.6.2 Break Address Registers

The break address registers (BRKH and BRKL) contain the high and low bytes of the desired breakpoint address. Reset clears the break address registers.

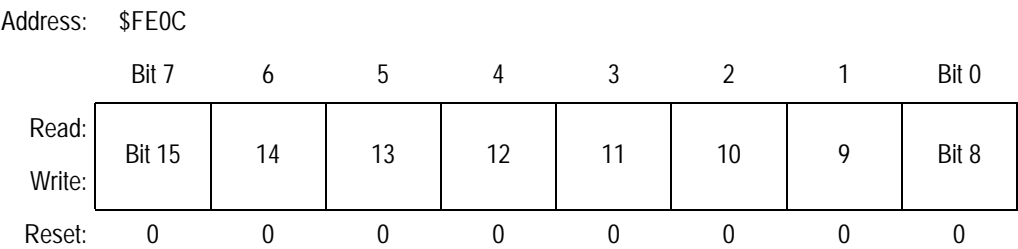


Figure 22-4. Break Address Register High (BRKH)

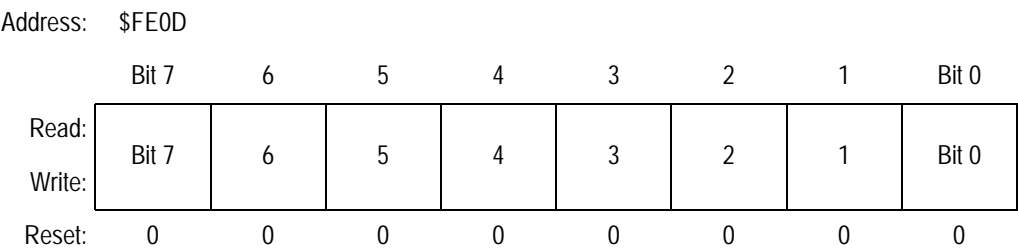


Figure 22-5. Break Address Register Low (BRKL)

22.6.3 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.