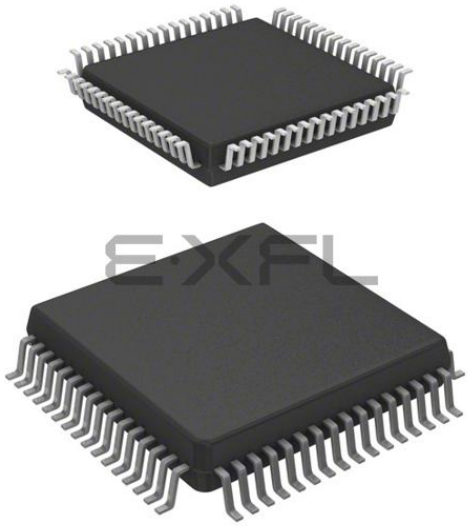


Welcome to E-XFL.COM

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	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 ~ 125°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/mc908az60amfuer

1.4.15 Port E I/O Pins (PTE7/SPSCK–PTE0/TxD)

Port E is an 8-bit special function port that shares two of its pins with the Timer Interface Module A (TIMA), four of its pins with the Serial Peripheral Interface module (SPI), and two of its pins with the Serial Communication Interface module (SCI). See Chapter 18 Serial Communications Interface (SCI), Chapter 19 Serial Peripheral Interface (SPI), Chapter 25 Timer Interface Module A (TIMA), and Chapter 22 Input/Output Ports.

1.4.16 Port F I/O Pins (PTF6–PTF0/TACH2)

Port F is a 7-bit special function port that shares its pins with the Timer Interface Module B (TIMB). Six of its pins are shared with the Timer Interface Module A (TIMA-6). See Chapter 25 Timer Interface Module A (TIMA), Chapter 20 Timer Interface Module B (TIMB), and Chapter 22 Input/Output Ports.

1.4.17 Port G I/O Pins (PTG2/KBD2–PTG0/KBD0)

Port G is a 3-bit special function port that shares all of its pins with the Keyboard Module (KBD). See Chapter 24 Keyboard Module (KBI) and Chapter 22 Input/Output Ports.

1.4.18 Port H I/O Pins (PTH1/KBD4–PTH0/KBD3)

Port H is a 2-bit special-function port that shares all of its pins with the Keyboard Module (KBD). See Chapter 24 Keyboard Module (KBI) and Chapter 22 Input/Output Ports.

1.4.19 CAN Transmit Pin (CANTx)

This pin is the digital output from the CAN module (CANTx). See Chapter 23 MSCAN Controller (MSCAN08).

1.4.20 CAN Receive Pin (CANRx)

This pin is the digital input to the CAN module (CANRx). See Chapter 23 MSCAN Controller (MSCAN08).

1.4.21 BDLC Transmit Pin (BDTxD)

This pin is the digital output from the BDLC module (BDTxD). See Chapter 27 Byte Data Link Controller (BDLC).

1.4.22 BDLC Receive Pin (BDRxD)

This pin is the digital input to the CAN module (BDRxD). See Chapter 27 Byte Data Link Controller (BDLC).

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

6.5.3 EEPROM-1 Nonvolatile Register

The contents of this register is loaded into the EEPROM-1 array configuration register (EE1ACR) after a reset.

This register is erased and programmed in the same way as an EEPROM byte. (See 6.5.1 EEPROM-1 Control Register for individual bit descriptions).

Address: $FE1C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	UNUSED	UNUSED	UNUSED	EEPRTCT	EEBP3	EEBP2	EEBP1	EEBP0
Write:								
Reset:	PV							

PV = Programmed value or 1 in the erased state.

Figure 6-4. EEPROM-1 Nonvolatile Register (EE1NVR)

NOTE

The EE1NVR will leave the factory programmed with $F0 such that the full array is available and unprotected.

6.5.4 EEPROM-1 Timebase Divider Register

The 16-bit EEPROM-1 timebase divider register consists of two 8-bit registers: EE1DIVH and EE1DIVL. The 11-bit value in this register is used to configure the timebase divider circuit to obtain the 35 μs timebase for EEPROM-1 control.

These two read/write registers are respectively loaded with the contents of the EEPROM-1 timebase divider nonvolatile registers (EE1DIVHNVR and EE1DIVLNVR) after a reset.

Address: $FE1A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	EEDIVS-ECD	0	0	0	0	EEDIV10	EEDIV9	EEDIV8
Write:								
Reset:	Contents of EE1DIVHNVR ($FE10), Bits [6:3] = 0							

= Unimplemented

Figure 6-5. EE1DIV Divider High Register (EE1DIVH)

Address: $FE1B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	EEDIV7	EEDIV6	EEDIV5	EEDIV4	EEDIV3	EEDIV2	EEDIV1	EEDIV0
Write:								
Reset:	Contents of EE1DIVLNVR ($FE11)							

Figure 6-6. EE1DIV Divider Low Register (EE1DIVL)

7.4.5 EEPROM-2 Programming and Erasing

The unprogrammed or erase state of an EEPROM bit is a logic 1. The factory default for all bytes within the EEPROM-2 array is $FF.

The programming operation changes an EEPROM bit from logic 1 to logic 0 (programming cannot change a bit from logic 0 to a logic 1). In a single programming operation, the minimum EEPROM programming size is one bit; the maximum is eight bits (one byte).

The erase operation changes an EEPROM bit from logic 0 to logic 1. In a single erase operation, the minimum EEPROM erase size is one byte; the maximum is the entire EEPROM-2 array.

The EEPROM can be programmed such that one or multiple bits are programmed (written to a logic 0) at a time. However, the user may never program the same bit location more than once before erasing the entire byte. In other words, the user is not allowed to program a logic 0 to a bit that is already programmed (bit state is already logic 0).

For some applications it might be advantageous to track more than 10K events with a single byte of EEPROM by programming one bit at a time. For that purpose, a special selective bit programming technique is available. An example of this technique is illustrated in Table 7-2.

Table 7-2. Example Selective Bit Programming Description

Description	Program Data in Binary	Result in Binary
Original state of byte (erased)	n/a	1111:1111
First event is recorded by programming bit position 0	1111:1110	1111:1110
Second event is recorded by programming bit position 1	1111:1101	1111:1100
Third event is recorded by programming bit position 2	1111:1011	1111:1000
Fourth event is recorded by programming bit position 3	1111:0111	1111:0000
Events five through eight are recorded in a similar fashion		

NOTE

None of the bit locations are actually programmed more than once although the byte was programmed eight times.

When this technique is utilized, a program/erase cycle is defined as multiple program sequences (up to eight) to a unique location followed by a single erase operation.

7.4.5.1 Program/Erase Using AUTO Bit

An additional feature available for EEPROM-2 program and erase operations is the AUTO mode. When enabled, AUTO mode will activate an internal timer that will automatically terminate the program/erase cycle and clear the EEPGM bit. Please see 7.4.5.2 EEPROM-2 Programming, 7.4.5.3 EEPROM-2 Erasing, and 7.5.1 EEPROM-2 Control Register for more information.

9.4.3 SIM Counter and Reset States

External reset has no effect on the SIM counter. See 9.6.2 Stop Mode for details. The SIM counter is free-running after all reset states. See 9.3.2 Active Resets from Internal Sources for counter control and internal reset recovery sequences.

9.5 Program 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

9.5.1 Interrupts

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. Figure 9-8 shows interrupt entry timing. Figure 9-10 shows interrupt recovery timing.

Interrupts are latched, and arbitration is performed in the SIM at the start of interrupt processing. The arbitration result is a constant that the CPU uses to determine which vector to fetch. Once an interrupt is latched by the SIM, no other interrupt can take precedence, regardless of priority, until the latched interrupt is serviced (or the I bit is cleared), see Figure 9-9.

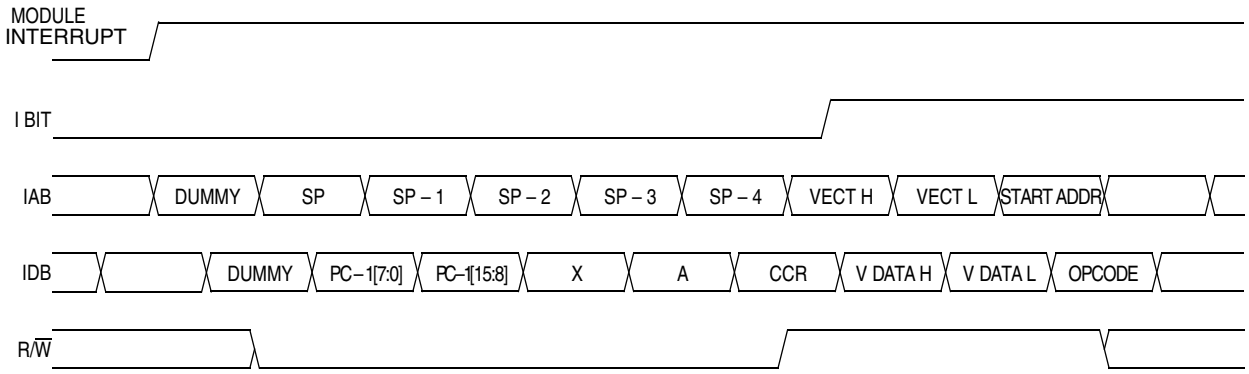


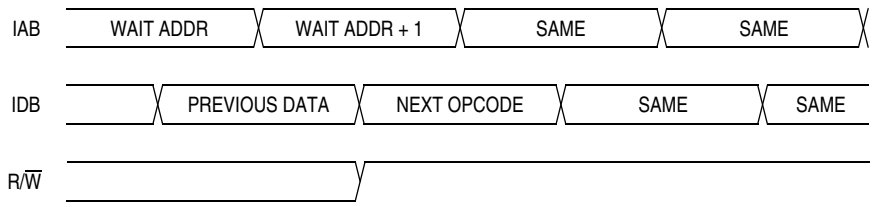
Figure 9-8. Interrupt Entry

9.6.1 Wait Mode

In wait mode, the CPU clocks are inactive while one set of peripheral clocks continue to run. Figure 9-12 shows the timing for wait mode entry.

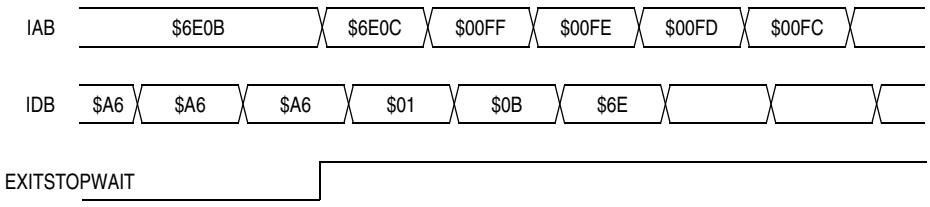
A module that is active during wait mode can wake up the CPU with an interrupt if the interrupt is enabled. Stacking for the interrupt begins one cycle after the WAIT instruction during which the interrupt occurred. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

Wait mode can also be exited by a reset or break. A break interrupt during wait mode sets the SIM break wait bit, BW, in the SIM break status register (SBSR). If the COP disable bit, COPD, in the configuration register is logic 0, then the computer operating properly module (COP) is enabled and remains active in wait mode.



NOTE: Previous data can be operand data or the WAIT opcode, depending on the last instruction.

Figure 9-12. Wait Mode Entry Timing



NOTE: EXITSTOPWAIT = $\overline{\text{RST}}$ pin OR CPU interrupt OR break interrupt

Figure 9-13. Wait Recovery from Interrupt or Break

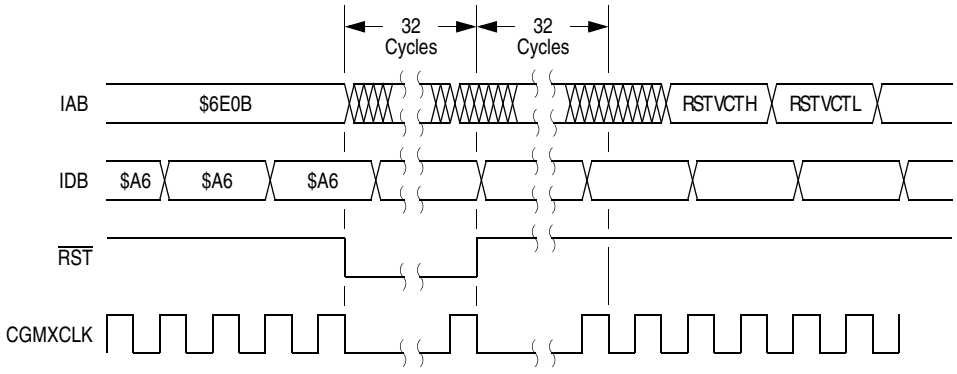


Figure 9-14. Wait Recovery from Internal Reset

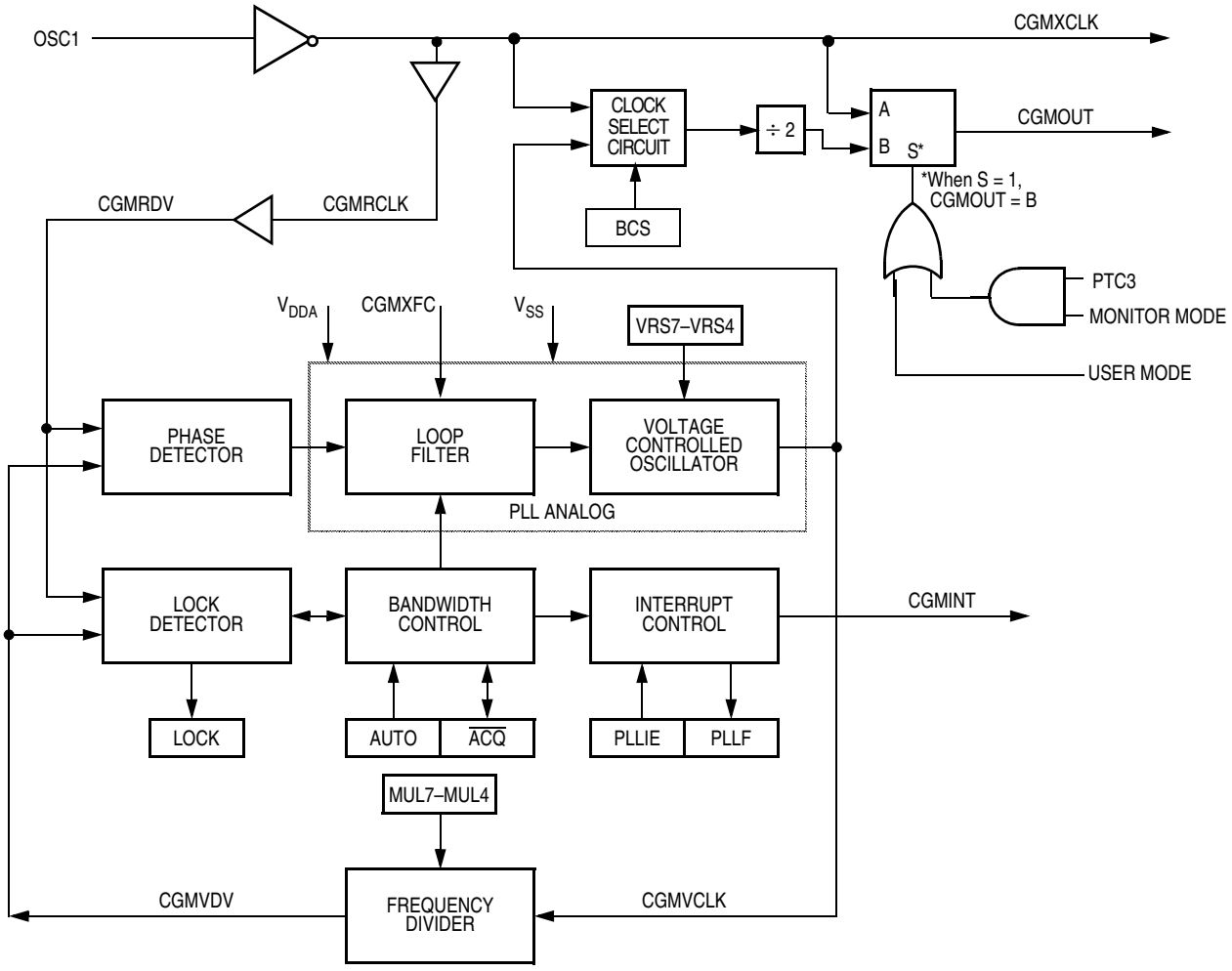


Figure 10-1. CGM Block Diagram

it may take up to three CGMXCLK and three CGMVCLK cycles to complete the transition from one source clock to the other. During the transition, CGMOUT is held in stasis. See 10.3.3 Base Clock Selector Circuit. Reset and the STOP instruction clear the BCS bit.

1 = CGMVCLK divided by two drives CGMOUT
0 = CGMXCLK divided by two drives CGMOUT

NOTE

PLLON and BCS have built-in protection that prevents the base clock selector circuit from selecting the VCO clock as the source of the base clock if the PLL is off. Therefore, PLLON cannot be cleared when BCS is set, and BCS cannot be set when PLLON is clear. If the PLL is off (PLLON = 0), selecting CGMVCLK requires two writes to the PLL control register. See 10.3.3 Base Clock Selector Circuit.

PCTL3–PCTL0 — Unimplemented

These bits provide no function and always read as logic 1s.

10.5.2 PLL Bandwidth Control Register

The PLL bandwidth control register:

- Selects automatic or manual (software-controlled) bandwidth control mode
- Indicates when the PLL is locked
- In automatic bandwidth control mode, indicates when the PLL is in acquisition or tracking mode
- In manual operation, forces the PLL into acquisition or tracking mode

Address: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AUTO	LOCK	$\overline{ACQ}$	XLD	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-5. PLL Bandwidth Control Register (PBWC)

AUTO — Automatic Bandwidth Control Bit

This read/write bit selects automatic or manual bandwidth control. When initializing the PLL for manual operation (AUTO = 0), clear the $\overline{ACQ}$ bit before turning on the PLL. Reset clears the AUTO bit.

1 = Automatic bandwidth control
0 = Manual bandwidth control

LOCK — Lock Indicator Bit

When the AUTO bit is set, LOCK is a read-only bit that becomes set when the VCO clock, CGMVCLK, is locked (running at the programmed frequency). When the AUTO bit is clear, LOCK reads as logic 0 and has no meaning. Reset clears the LOCK bit.

1 = VCO frequency correct or locked
0 = VCO frequency incorrect or unlocked

$\overline{ACQ}$ — Acquisition Mode Bit

When the AUTO bit is set, $\overline{ACQ}$ is a read-only bit that indicates whether the PLL is in acquisition mode or tracking mode. When the AUTO bit is clear, $\overline{ACQ}$ is a read/write bit that controls whether the PLL is in acquisition or tracking mode.

Address:	$0014							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0

Figure 18-12. SCI Control Register 2 (SCC2)

SCTIE — SCI Transmit Interrupt Enable Bit

This read/write bit enables the SCTE bit to generate SCI transmitter CPU interrupt requests. Setting the SCTIE bit in SCC3 enables the SCTE bit to generate CPU interrupt requests. Reset clears the SCTIE bit.

1 = SCTE enabled to generate CPU interrupt
0 = SCTE not enabled to generate CPU interrupt

TCIE — Transmission Complete Interrupt Enable Bit

This read/write bit enables the TC bit to generate SCI transmitter CPU interrupt requests. Reset clears the TCIE bit.

1 = TC enabled to generate CPU interrupt requests
0 = TC not enabled to generate CPU interrupt requests

SCRIE — SCI Receive Interrupt Enable Bit

This read/write bit enables the SCRF bit to generate SCI receiver CPU interrupt requests. Setting the SCRIE bit in SCC3 enables the SCRF bit to generate CPU interrupt requests. Reset clears the SCRIE bit.

1 = SCRF enabled to generate CPU interrupt
0 = SCRF not enabled to generate CPU interrupt

ILIE — Idle Line Interrupt Enable Bit

This read/write bit enables the IDLE bit to generate SCI receiver CPU interrupt requests. Reset clears the ILIE bit.

1 = IDLE enabled to generate CPU interrupt requests
0 = IDLE not enabled to generate CPU interrupt requests

TE — Transmitter Enable Bit

Setting this read/write bit begins the transmission by sending a preamble of 10 or 11 1s from the transmit shift register to the TxD pin. If software clears the TE bit, the transmitter completes any transmission in progress before the TxD returns to the idle condition (1). Clearing and then setting TE during a transmission queues an idle character to be sent after the character currently being transmitted. Reset clears the TE bit.

1 = Transmitter enabled
0 = Transmitter disabled

NOTE

Writing to the TE bit is not allowed when the enable SCI bit (ENSCI) is clear. ENSCI is in SCI control register 1.

R8 — Received Bit 8

When the SCI is receiving 9-bit characters, R8 is the read-only ninth bit (bit 8) of the received character. R8 is received at the same time that the SCDR receives the other 8 bits.

When the SCI is receiving 8-bit characters, R8 is a copy of the eighth bit (bit 7). Reset has no effect on the R8 bit.

T8 — Transmitted Bit 8

When the SCI is transmitting 9-bit characters, T8 is the read/write ninth bit (bit 8) of the transmitted character. T8 is loaded into the transmit shift register at the same time that the SCDR is loaded into the transmit shift register. Reset has no effect on the T8 bit.

ORIE — Receiver Overrun Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the receiver overrun bit, OR.

1 = SCI error CPU interrupt requests from OR bit enabled
0 = SCI error CPU interrupt requests from OR bit disabled

NEIE — Receiver Noise Error Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the noise error bit, NE. Reset clears NEIE.

1 = SCI error CPU interrupt requests from NE bit enabled
0 = SCI error CPU interrupt requests from NE bit disabled

FEIE — Receiver Framing Error Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the framing error bit, FE. Reset clears FEIE.

1 = SCI error CPU interrupt requests from FE bit enabled
0 = SCI error CPU interrupt requests from FE bit disabled

PEIE — Receiver Parity Error Interrupt Enable Bit

This read/write bit enables SCI receiver CPU interrupt requests generated by the parity error bit, PE. Reset clears PEIE.

1 = SCI error CPU interrupt requests from PE bit enabled
0 = SCI error CPU interrupt requests from PE bit disabled

18.8.4 SCI Status Register 1

SCI status register 1 contains flags to signal the following conditions:

- Transfer of SCDR data to transmit shift register complete
- Transmission complete
- Transfer of receive shift register data to SCDR complete
- Receiver input idle
- Receiver overrun
- Noisy data
- Framing error
- Parity error

FE — Receiver Framing Error Bit

This clearable, read-only bit is set when a logic 0 is accepted as the stop bit. FE generates an SCI error CPU interrupt request if the FEIE bit in SCC3 also is set. Clear the FE bit by reading SCS1 with FE set and then reading the SCDR. Reset clears the FE bit.

1 = Framing error detected
0 = No framing error detected

PE — Receiver Parity Error Bit

This clearable, read-only bit is set when the SCI detects a parity error in incoming data. PE generates a SCI receiver CPU interrupt request if the PEIE bit in SCC3 is also set. Clear the PE bit by reading SCS1 with PE set and then reading the SCDR. Reset clears the PE bit.

1 = Parity error detected
0 = No parity error detected

18.8.5 SCI Status Register 2

SCI status register 2 contains flags to signal the following conditions:

- Break character detected
- Incoming data

Address: $0017

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	BKF	RPF
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 18-16. SCI Status Register 2 (SCS2)

BKF — Break Flag Bit

This clearable, read-only bit is set when the SCI detects a break character on the RxD pin. In SCS1, the FE and SCRF bits are also set. In 9-bit character transmissions, the R8 bit in SCC3 is cleared. BKF does not generate a CPU interrupt request. Clear BKF by reading SCS2 with BKF set and then reading the SCDR. Once cleared, BKF can become set again only after 1s appear on the RxD pin followed by another break character. Reset clears the BKF bit.

1 = Break character detected
0 = No break character detected

RPF — Reception in Progress Flag Bit

This read-only bit is set when the receiver detects a logic 0 during the RT1 time period of the start bit search. RPF does not generate an interrupt request. RPF is reset after the receiver detects false start bits (usually from noise or a baud rate mismatch), or when the receiver detects an idle character. Polling RPF before disabling the SCI module or entering stop mode can show whether a reception is in progress.

1 = Reception in progress
0 = No reception in progress

The generic names of the SPI I/O registers are:

- SPI control register (SPCR)
- SPI status and control register (SPSCR)
- SPI data register (SPDR)

Table 19-2 shows the names and the addresses of the SPI I/O registers.

Table 19-2. I/O Register Addresses

Register Name	Address
SPI Control Register (SPCR)	$0010
SPI Status and Control Register (SPSCR)	$0011
SPI Data Register (SPDR)	$0012

19.4 Functional Description

Figure 19-1 summarizes the SPI I/O registers and Figure 19-2 shows the structure of the SPI module.

Addr	Register Name	R/W	Bit 7	6	5	4	3	2	1	Bit 0
$0010	SPI Control Register (SPCR)	Read:	SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
		Write:								
		Reset:	0	0	1	0	1	0	0	0
$0011	SPI Status and Control Register (SPSCR)	Read:	SPRF	ERRIE	OVRF	MODF	SPTE	MODFEN	SPR1	SPR0
		Write:								
		Reset:	0	0	0	0	1	0	0	0
$0012	SPI Data Register (SPDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by Reset							

R = Reserved = Unimplemented

Figure 19-1. SPI I/O Register Summary

The SPI module allows full-duplex, synchronous, serial communication between the MCU and peripheral devices, including other MCUs. Software can poll the SPI status flags or SPI operation can be interrupt driven. All SPI interrupts can be serviced by the CPU.

The following paragraphs describe the operation of the SPI module.

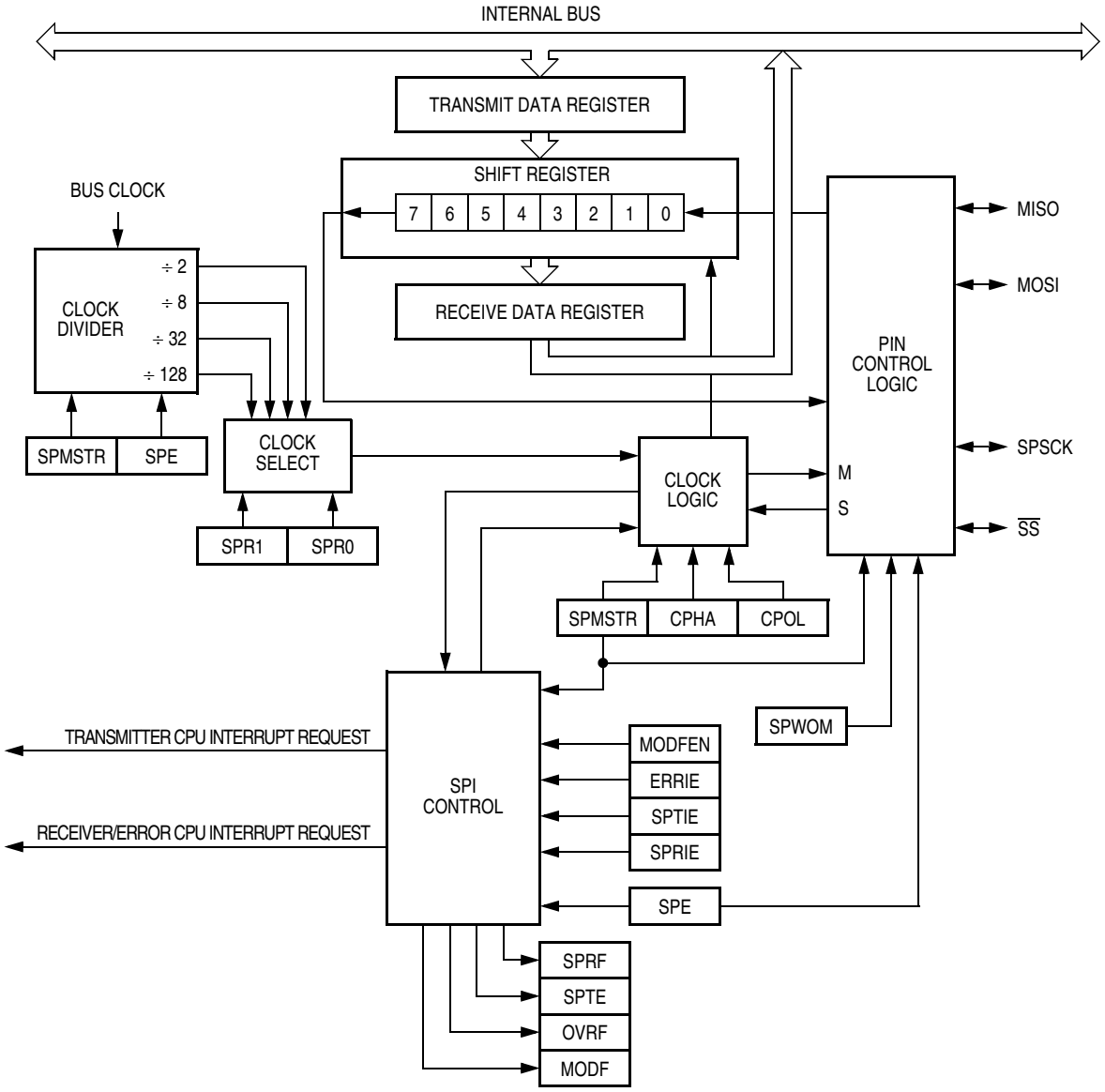


Figure 19-2. SPI Module Block Diagram

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.

Register Name and Address TBCH0H — $0046

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	Indeterminate after Reset							

Register Name and Address TBCH0L — $0047

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	Indeterminate after Reset							

Register Name and Address TBCH1H — $0049

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	Indeterminate after Reset							

Register Name and Address TBCH1L — $004A

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	Indeterminate after Reset							

Figure 20-9. TIMB Channel Registers (TBCH0H/L–TBCH1H/L)

Chapter 22
Input/Output Ports

22.1 Introduction

On the MC68HC908AZ60A and 64-pin MC68HC908AS60A, fifty bidirectional input/output (I/O) form seven parallel ports. On the52-pin MC68HC908AS60A, forty bidirectional input/output (I/O) form six parallel ports. All I/O pins are programmable as inputs or outputs.

NOTE

Connect any unused I/O pins to an appropriate logic level, either V_{DD} or V_{SS}. Although the I/O ports do not require termination for proper operation, termination reduces excess current consumption and the possibility of electrostatic damage.

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA)	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
$0001	Port B Data Register (PTB)	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
$0002	Port C Data Register (PTC)	0	0	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
$0003	Port D Data Register (PTD)	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
$0004	Data Direction Register A (DDRA)	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
$0005	Data Direction Register B (DDRB)	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
$0006	Data Direction Register C (DDRC)	MCLKEN	0	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
$0007	Data Direction Register D (DDRD)	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
$0008	Port E Data Register (PTE)	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0
$0009	Port F Data Register (PTF)	0	PTF6	PTF5	PTF4	PTF3	PTF2	PTF1	PTF0
$000A	Port G Data Register (PTG)	0	0	0	0	0	PTG2	PTG1	PTG0
$000B	Port H Data Register (PTH)	0	0	0	0	0	0	PTH1	PTH0
$000C	Data Direction Register E (DDRE)	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
$000D	Data Direction Register F (DDRF)	0	DDRF6	DDRF5	DDRF4	DDRF3	DDRF2	DDRF1	DDRF0
$000E	Data Direction Register G (DDRG)	0	0	0	0	0	DDRG2	DDRG1	DDRG0
$000F	Data Direction Register H (DDRH)	0	0	0	0	0	0	DDRH1	DDRH0

Figure 22-1. I/O Port Register Summary

23.12.4 Data Segment Registers (DSRn)

The eight data segment registers contain the data to be transmitted or received. The number of bytes to be transmitted or being received is determined by the data length code in the corresponding DLR.

23.12.5 Transmit Buffer Priority Registers

Address: $05bD

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PRIO7	PRIO6	PRIO5	PRIO4	PRIO3	PRIO2	PRIO1	PRIO0
Reset:	Unaffected by Reset							

Figure 23-13. Transmit Buffer Priority Register (TBPR)

PRIO7–PRIO0 — Local Priority

This field defines the local priority of the associated message buffer. The local priority is used for the internal prioritisation process of the MSCAN08 and is defined to be highest for the smallest binary number. The MSCAN08 implements the following internal prioritisation mechanism:

- All transmission buffers with a cleared TXE flag participate in the prioritisation right before the SOF is sent.
- The transmission buffer with the lowest local priority field wins the prioritisation.
- In case more than one buffer has the same lowest priority, the message buffer with the lower index number wins.

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.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0039	Analog-to-Digital Data Register (ADR)	Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		Write:								
$003A	Analog-to-Digital Input Clock Register (ADICLK)	Read:	ADIV2	ADIV1	ADIV0	ADICLK	0	0	0	0
		Write:					R	R	R	R
$0040	Timer B Status and Control Register (TBSCR)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST	R			
$0041	Timer B Counter Register High (TBCNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$0042	Timer B Counter Register Low (TBCNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
$0043	Timer B Modulo Register High (TBMODH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$0044	Timer B Modulo Register Low (TBMODL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
$0045	Timer B CH0 Status and Control Register (TBSC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
$0046	Timer B CH0 Register High (TBCH0H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$0047	Timer B CH0 Register Low (TBCH0L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
$0048	Timer B CH1 Status and Control Register (TBSC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
$0049	Timer B CH1 Register High (TBCH1H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$004A	Timer B CH1 Register Low (TBCH1L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
$004B	PIT Status and Control Register (PSC)	Read:	POF	POIE	PSTOP	0	0	PPS2	PPS1	PPS0
		Write:	0			PRST				
$004C	PIT Counter Register High (PCNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$004D	PIT Counter Register Low (PCNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
$004E	PIT Modulo Register High (PMODH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
$004F	PIT Modulo Register Low (PMODL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								

= Unimplemented R = Reserved

Figure B-2. I/O Data, Status and Control Registers (Sheet 4 of 4)

toggle — To change the state of an output from a logic 0 to a logic 1 or from a logic 1 to a logic 0.

tracking mode — Mode of low-jitter PLL operation during which the PLL is locked on a frequency. Also see "acquisition mode."

two's complement — A means of performing binary subtraction using addition techniques. The most significant bit of a two's complement number indicates the sign of the number (1 indicates negative). The two's complement negative of a number is obtained by inverting each bit in the number and then adding 1 to the result.

unbuffered — Utilizes only one register for data; new data overwrites current data.

unimplemented memory location — A memory location that is not used. Writing to an unimplemented location has no effect. Reading an unimplemented location returns an unpredictable value. Executing an opcode at an unimplemented location causes an illegal address reset.

V —The overflow bit in the condition code register of the CPU08. The CPU08 sets the V bit when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow bit.

variable — A value that changes during the course of program execution.

VCO — See "voltage-controlled oscillator."

vector — A memory location that contains the address of the beginning of a subroutine written to service an interrupt or reset.

voltage-controlled oscillator (VCO) — A circuit that produces an oscillating output signal of a frequency that is controlled by a dc voltage applied to a control input.

waveform — A graphical representation in which the amplitude of a wave is plotted against time.

wired-OR — Connection of circuit outputs so that if any output is high, the connection point is high.

word — A set of two bytes (16 bits).

write — The transfer of a byte of data from the CPU to a memory location.

X — The lower byte of the index register (H:X) in the CPU08.

Z — The zero bit in the condition code register of the CPU08. The CPU08 sets the zero bit when an arithmetic operation, logical operation, or data manipulation produces a result of $00.