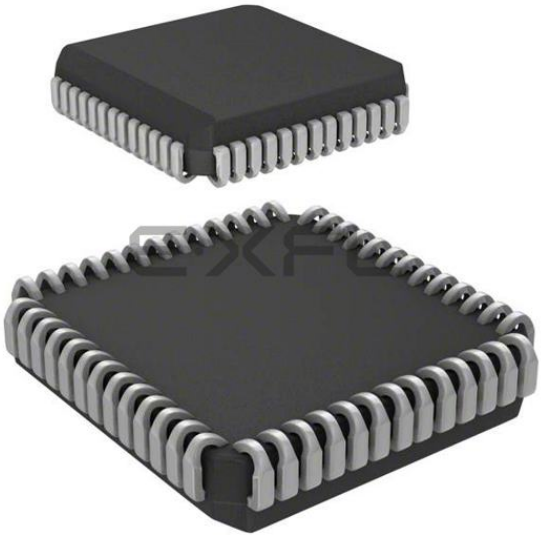


Welcome to E-XFL.COM

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	42
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	52-LCC (J-Lead)
Supplier Device Package	52-PLCC (19.1x19.1)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908as60acfn

Chapter 14
Monitor ROM (MON)

Chapter 15
Computer Operating Properly (COP)

Chapter 16
Low-Voltage Inhibit (LVI)

4.8 Low-Power Modes

The WAIT and STOP instructions will place the MCU in low power consumption standby modes.

4.8.1 WAIT Mode

Putting the MCU into wait mode while the FLASH is in read mode does not affect the operation of the FLASH memory directly; however, no memory activity will take place since the CPU is inactive.

The WAIT instruction should not be executed while performing a program or erase operation on the FLASH. Wait mode will suspend any FLASH program/erase operations and leave the memory in a Standby Mode.

4.8.2 STOP Mode

Putting the MCU into stop mode while the FLASH is in read mode does not affect the operation of the FLASH memory directly; however, no memory activity will take place since the CPU is inactive.

The STOP instruction should not be executed while performing a program or erase operation on the FLASH. Stop mode will suspend any FLASH program/erase operations and leave the memory in a Standby Mode.

NOTE

Standby Mode is the power saving mode of the FLASH module, in which all internal control signals to the FLASH are inactive and the current consumption of the FLASH is minimum.

These two registers are protected from erase and program operations if the EEDIVSECD is set to logic 1 in the EE1DIVH (see EEPROM-1 Timebase Divider Register) or programmed to a logic 1 in the EE1DIVHNVR.

NOTE

Once EEDIVSECD in the EE1DIVHNVR is programmed to 0 and after a system reset, the EE1DIV security feature is permanently enabled because the EEDIVSECD bit in the EE1DIVH is always loaded with 0 thereafter. Once this security feature is armed, erase and program mode are disabled for EE1DIVHNVR and EE1DIVLNVR. Modifications to the EE1DIVH and EE1DIVL registers are also disabled. Therefore, care should be taken before programming a value into the EE1DIVHNVR.

6.6 Low-Power Modes

The WAIT and STOP instructions can put the MCU in low power-consumption standby modes.

6.6.1 Wait Mode

The WAIT instruction does not affect the EEPROM. It is possible to start the program or erase sequence on the EEPROM and put the MCU in wait mode.

6.6.2 Stop Mode

The STOP instruction reduces the EEPROM power consumption to a minimum. The STOP instruction should not be executed while a programming or erasing sequence is in progress.

If stop mode is entered while EELAT and EEPGM are set, the programming sequence will be stopped and the programming voltage to the EEPROM array removed. The programming sequence will be restarted after leaving stop mode; access to the EEPROM is only possible after the programming sequence has completed.

If stop mode is entered while EELAT and EEPGM is cleared, the programming sequence will be terminated abruptly.

In either case, the data integrity of the EEPROM is not guaranteed.

AUTO — Automatic Termination of Program/Erase Cycle

When AUTO is set, EEPGM is cleared automatically after the program/erase cycle is terminated by the internal timer.

(See note D for 7.4.5.2 EEPROM-2 Programming, 7.4.5.3 EEPROM-2 Erasing, and 28.1.13 EEPROM Memory Characteristics)

1 = Automatic clear of EEPGM is enabled
0 = Automatic clear of EEPGM is disabled

EEPGM — EEPROM-2 Program/Erase Enable

This read/write bit enables the internal charge pump and applies the programming/erasing voltage to the EEPROM-2 array if the EELAT bit is set and a write to a valid EEPROM-2 location has occurred. Reset clears the EEPGM bit.

1 = EEPROM-2 programming/erasing power switched on
0 = EEPROM-2 programming/erasing power switched off

NOTE

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

7.5.2 EEPROM-2 Array Configuration Register

The EEPROM-2 array configuration register configures EEPROM-2 security and EEPROM-2 block protection.

This read-only register is loaded with the contents of the EEPROM-2 nonvolatile register (EE2NVR) after a reset.

Address: $FF7F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	UNUSED	UNUSED	UNUSED	EEPRTCT	EEBP3	EEBP2	EEBP1	EEBP0
Write:								
Reset:	Contents of EE2NVR ($FF7C)							

= Unimplemented

Figure 7-3. EEPROM-2 Array Configuration Register (EE2ACR)

Bit 7:5 — Unused Bits

These read/write bits are software programmable but have no functionality.

EEPRTCT — EEPROM-2 Protection Bit

The EEPRTCT bit is used to enable the security feature in the EEPROM (see EEPROM-2 Program/Erase Protection).

1 = EEPROM-2 security disabled
0 = EEPROM-2 security enabled

This feature is a write-once feature. Once the protection is enabled it may not be disabled.

EEDIVSECD — EEPROM-2 Divider Security Disable

This bit enables/disables the security feature of the EE2DIV registers. When EE2DIV security feature is enabled, the state of the registers EE2DIVH and EE2DIVL are locked (including EEDIVSECD bit). The EE2DIVHNVR and EE2DIVLNVR nonvolatile memory registers are also protected from being erased/programmed.

1 = EE2DIV security feature disabled
0 = EE2DIV security feature enabled

EEDIV[10:0] — EEPROM-2 timebase prescaler

These prescaler bits store the value of EE2DIV which is used as the divisor to derive a timebase of 35μs from the selected reference clock source (CGMXCLK or bus block in the CONFIG-2 register) for the EEPROM-2 related internal timer and circuits. EEDIV[10:0] bits are readable at any time. They are writable when EELAT = 0 and EEDIVSECD = 1.

The EE2DIV value is calculated by the following formula:

$$EE2DIV = INT[Reference\ Frequency(Hz) \times 35 \times 10^{-6} + 0.5]$$

Where the result inside the bracket is rounded down to the nearest integer value

For example, if the reference frequency is 4.9152MHz, the EE2DIV value is 172

NOTE

Programming/erasing the EEPROM with an improper EE2DIV value may result in data lost and reduce endurance of the EEPROM device.

7.5.5 EEPROM-2 Timebase Divider Nonvolatile Register

The 16-bit EEPROM-2 timebase divider nonvolatile register consists of two 8-bit registers: EE2DIVHNVR and EE2DIVLNVR. The contents of these two registers are respectively loaded into the EEPROM-2 timebase divider registers, EE2DIVH and EE2DIVL, after a reset.

These two registers are erased and programmed in the same way as an EEPROM-2 byte.

Address: $FF70

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	EEDIVS-ECD	R	R	R	R	EEDIV10	EEDIV9	EEDIV8
Reset:	Unaffected by reset; $FF when blank							

R = Reserved

Figure 7-7. EEPROM-2 Divider Nonvolatile Register High (EE2DIVHNVR))

Address: $FF71

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	EEDIV7	EEDIV6	EEDIV5	EEDIV4	EEDIV3	EEDIV2	EEDIV1	EEDIV0
Reset:	Unaffected by reset; $FF when blank							

Figure 7-8. EEPROM-2 Divider Nonvolatile Register Low (EE2DIVLNVR)

Table 14-5. IREAD (Indexed Read) Command

Description	Read next 2 bytes in memory from last address accessed
Operand	Specifies 2-byte address in high byte:low byte order
Data Returned	Returns contents of next two addresses
Opcode	$1A
Command Sequence	

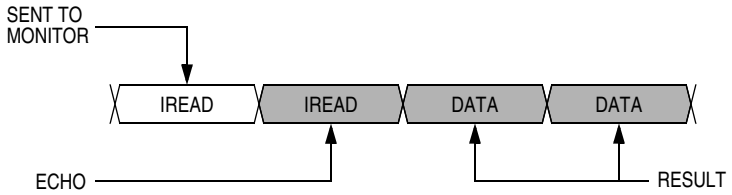


Table 14-6. IWRITE (Indexed Write) Command

Description	Write to last address accessed + 1
Operand	Specifies single data byte
Data Returned	None
Opcode	$19
Command Sequence	

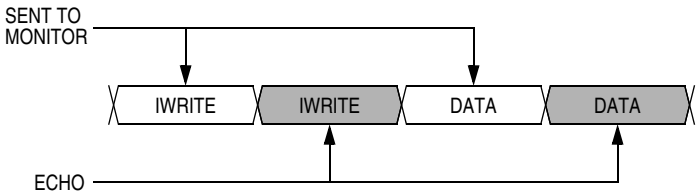
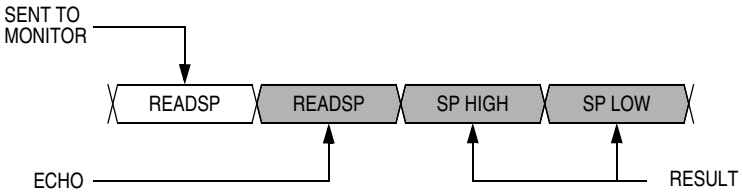


Table 14-7. READSP (Read Stack Pointer) Command

Description	Reads stack pointer
Operand	None
Data Returned	Returns stack pointer in high byte:low byte order
Opcode	$0C
Command Sequence	



Chapter 18
Serial Communications Interface (SCI)

18.1 Introduction

The SCI allows asynchronous communications with peripheral devices and other MCUs.

18.2 Features

The SCI module's features include:

- 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
- Two Receiver Wakeup Methods:
  - Idle Line Wakeup
  - Address Mark Wakeup
- Interrupt-Driven Operation with Eight Interrupt Flags:
  - Transmitter Empty
  - Transmission Complete
  - Receiver Full
  - Idle Receiver Input
  - Receiver Overrun
  - Noise Error
  - Framing Error
  - Parity Error
- Receiver Framing Error Detection
- Hardware Parity Checking
- 1/16 Bit-Time Noise Detection

18.3 Pin Name Conventions

The generic names of the SCI input/output (I/O) pins are:

- RxD (receive data)
- TxD (transmit data)

SCI I/O lines are implemented by sharing parallel I/O port pins. The full name of an SCI input or output reflects the name of the shared port pin. Table 18-1 shows the full names and the generic names of the SCI I/O pins.The generic pin names appear in the text of this subsection.

Table 18-1. Pin Name Conventions

Generic Pin Names	RxD	TxD
Full Pin Names	PTE1/SCRxD	PTE0/SCTxD

18.4 Functional Description

Figure 18-1 shows the structure of the SCI module. The SCI allows full-duplex, asynchronous, NRZ serial communication between the MCU and remote devices, including other MCUs. The transmitter and receiver of the SCI operate independently, although they use the same baud rate generator. During normal operation, the CPU monitors the status of the SCI, writes the data to be transmitted, and processes received data.

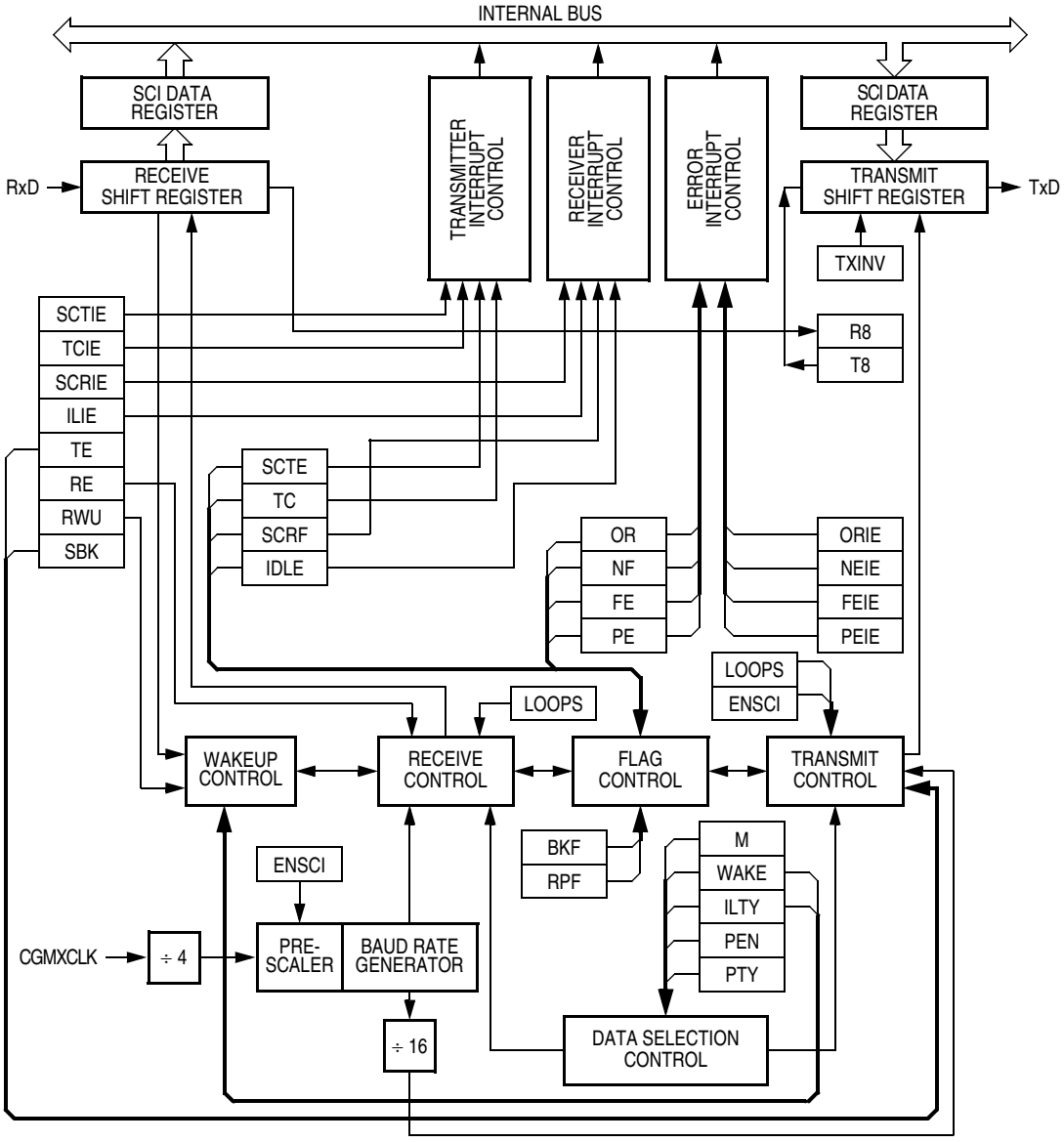


Figure 18-1. SCI Module Block Diagram

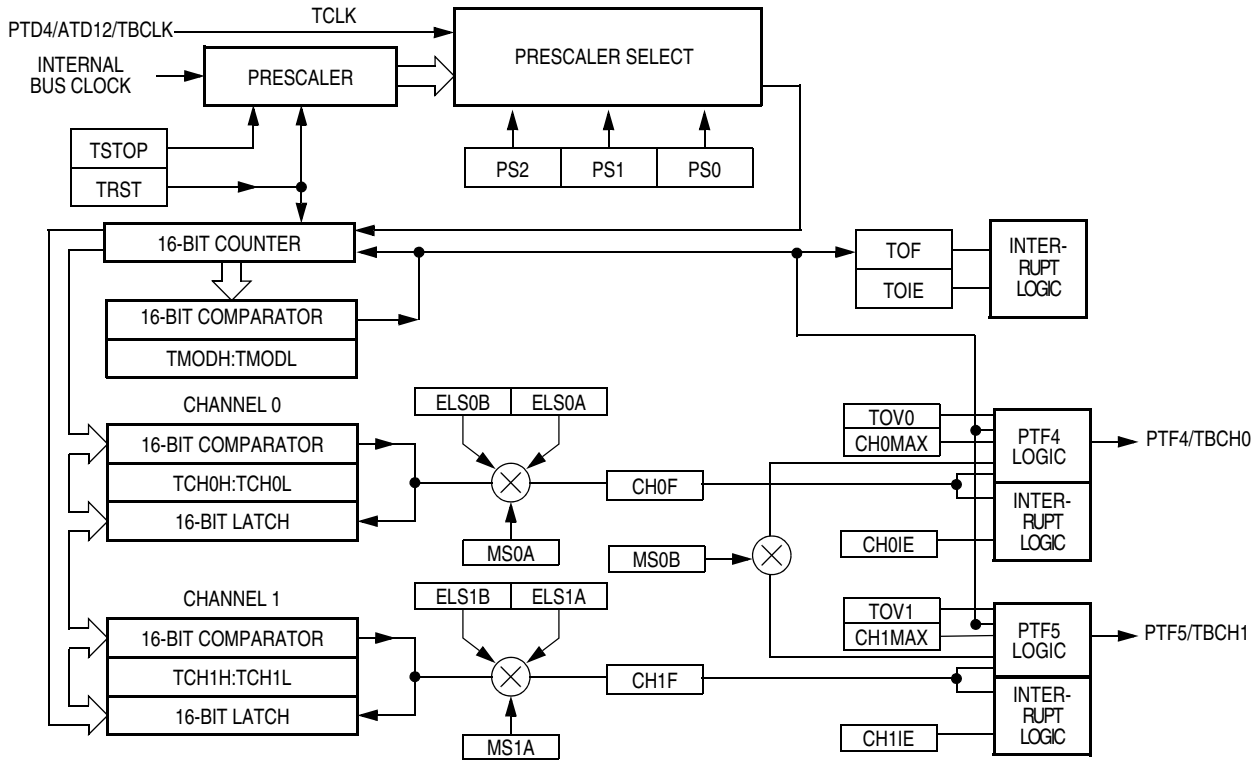


Figure 20-1. TIMB Block Diagram

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$0040	TIMB Status/Control Register (TBSC)	TOF	TOIE	TSTOP	TRST	0	PS2	PS1	PS0
$0041	TIMB Counter Register High (TBCNTH)	Bit 15	14	13	12	11	10	9	Bit 8
$0042	TIMB Counter Register Low (TBCNTL)	Bit 7	6	5	4	3	2	1	Bit 0
$0043	TIMB Counter Modulo Reg. High (TBMODH)	Bit 15	14	13	12	11	10	9	Bit 8
$0044	TIMB Counter Modulo Reg. Low (TBMODL)	Bit 7	6	5	4	3	2	1	Bit 0
$0045	TIMB Ch. 0 Status/Control Register (TBSC0)	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
$0046	TIMB Ch. 0 Register High (TBCH0H)	Bit 15	14	13	12	11	10	9	Bit 8
$0047	TIMB Ch. 0 Register Low (TBCH0L)	Bit 7	6	5	4	3	2	1	Bit 0
$0048	TIMB Ch. 1 Status/Control Register (TBSC1)	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
$0049	TIMB Ch. 1 Register High (TBCH1H)	Bit 15	14	13	12	11	10	9	Bit 8
$004A	TIMB Ch. 1 Register Low (TBCH1L)	Bit 7	6	5	4	3	2	1	Bit 0

R = Reserved

Figure 20-2. TIMB I/O Register Summary

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.

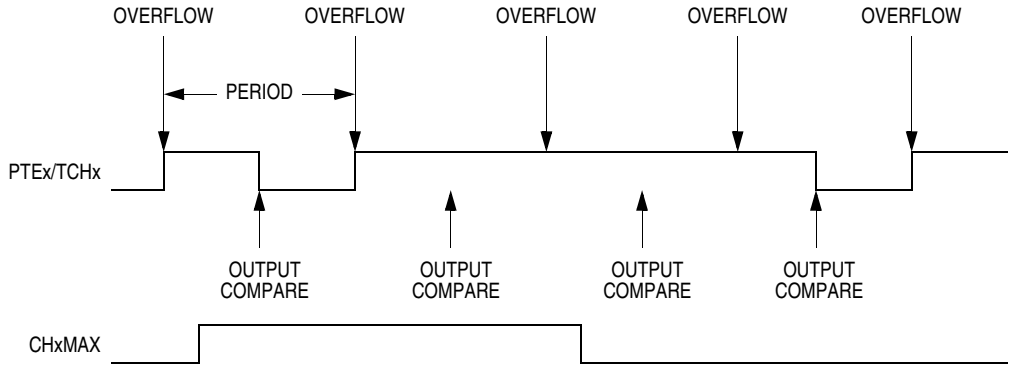


Figure 20-8. CHxMAX Latency

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.

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

21.5.2 Stop Mode

The PIT is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions or the state of the PIT counter. PIT operation resumes when the MCU exits stop mode after an external interrupt.

21.6 PIT During Break Interrupts

A break interrupt stops the PIT counter.

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state (see 9.7.3 SIM Break Flag Control Register).

To allow software to clear status bits during a break interrupt, write a 1 to the BCFE bit. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

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

21.7 I/O Registers

The following I/O registers control and monitor operation of the PIT:

- PIT status and control register (PSC)
- PIT counter registers (PCNTH–PCNTL)
- PIT counter modulo registers (PMODH–PMODL)

21.7.1 PIT Status and Control Register

The PIT status and control register:

- Enables PIT interrupt
- Flags PIT overflows
- Stops the PIT counter
- Resets the PIT counter
- Prescales the PIT counter clock

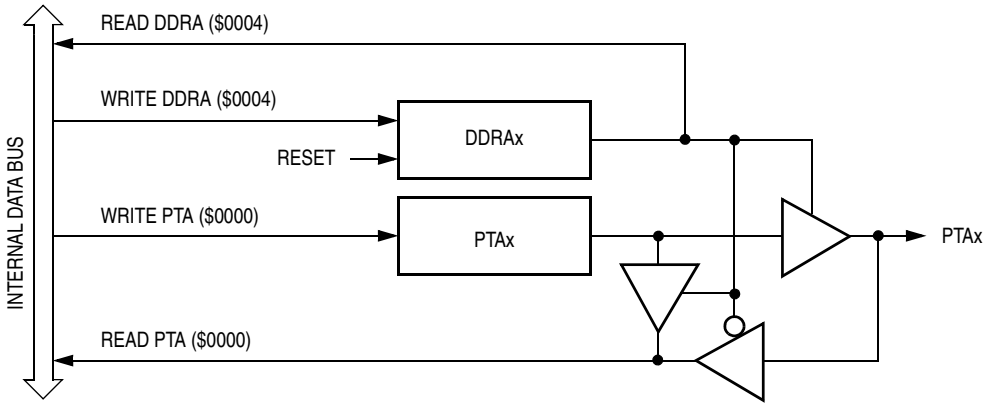


Figure 22-4. Port A I/O Circuit

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.

23.6.2 Interrupt Vectors

The MSCAN08 supports four interrupt vectors as shown in Table 23-1. The vector addresses and the relative interrupt priority are dependent on the chip integration and to be defined.

Table 23-1. MSCAN08 Interrupt Vector Addresses

Function	Source	Local Mask	Global Mask
Wakeup	WUPIF	WUPIE	I Bit
Error Interrupts	RWRNIF	RWRNIE	
	TWRNIF	TWRNIE	
	RERRIF	RERRIE	
	TERRIF	TERRIE	
	BOFFIF	BOFFIE	
	OVRIF	OVRIE	
Receive	RXF	RXFIE	
Transmit	TXE0	TXEIE0	
	TXE1	TXEIE1	
	TXE2	TXEIE2	

23.7 Protocol Violation Protection

The MSCAN08 will protect the user from accidentally violating the CAN protocol through programming errors. The protection logic implements the following features:

- The receive and transmit error counters cannot be written or otherwise manipulated.
- All registers which control the configuration of the MSCAN08 can not be modified while the MSCAN08 is on-line. The SFTRES bit in the MSCAN08 module control register (see 23.13.1 MSCAN08 Module Control Register 0) serves as a lock to protect the following registers:
  - MSCAN08 module control register 1 (CMCR1)
  - MSCAN08 bus timing register 0 and 1 (CBTR0 and CBTR1)
  - MSCAN08 identifier acceptance control register (CIDAC)
  - MSCAN08 identifier acceptance registers (CIDAR0–3)
  - MSCAN08 identifier mask registers (CIDMR0–3)
- The TxCAN pin is forced to recessive when the MSCAN08 is in any of the Low Power Modes.

23.8 Low Power Modes

In addition to normal mode, the MSCAN08 has three modes with reduced power consumption: Sleep, Soft Reset and Power Down modes. In Sleep and Soft Reset mode, power consumption is reduced by stopping all clocks except those to access the registers. In Power Down mode, all clocks are stopped and no power is consumed.

The WAIT and STOP instructions put the MCU in low power consumption stand-by modes. summarizes the combinations of MSCAN08 and CPU modes. A particular combination of modes is entered for the given settings of the bits SLPAK and SFTRES. For all modes, an MSCAN wake-up interrupt can occur only if SLPAK=WUPIE=1.

Addr	Register		Bit 7	6	5	4	3	2	1	Bit 0
$05b0	IDR0	Read: Write:	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
$05b1	IDR1	Read: Write:	ID20	ID19	ID18	SRR (=1)	IDE (=1)	ID17	ID16	ID15
$05b2	IDR2	Read: Write:	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
$05b3	IDR3	Read: Write:	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
$05b4	DSR0	Read: Write:	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
$05b5	DSR1	Read: Write:	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
$05b6	DSR2	Read: Write:	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
$05b7	DSR3	Read: Write:	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
$05b8	DSR4	Read: Write:	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
$05b9	DSR5	Read: Write:	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
$05bA	DSR6	Read: Write:	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
$05bB	DSR7	Read: Write:	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
$05bC	DLR	Read: Write:					DLC3	DLC2	DLC1	DLC0

= Unimplemented

Figure 23-11. Receive/Transmit Message Buffer Extended Identifier (IDRn)

Addr	Register		Bit 7	6	5	4	3	2	1	Bit 0
$05b0	IDR0	Read: Write:	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
$05b1	IDR1	Read: Write:	ID2	ID1	ID0	RTR	IDE(=0)			
$05b2	IDR2	Read: Write:								
$05b3	IDR3	Read: Write:								

= Unimplemented

Figure 23-12. Standard Identifier Mapping

Register Name and Address TASC5 — $0035

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH5F	CH5IE	0	MS5A	ELS5B	ELS5A	TOV5	CH5MAX
Write:	0		R					
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 25-7. TIMA Channel Status and Control Registers (TASC0–TASC5) (Continued)

CHxF — Channel x Flag Bit

When channel x is an input capture channel, this read/write bit is set when an active edge occurs on the channel x pin. When channel x is an output compare channel, CHxF is set when the value in the TIMA counter registers matches the value in the TIMA channel x registers.

When CHxIE = 1, clear CHxF by reading TIMA channel x status and control register with CHxF set and then writing a logic 0 to CHxF. If another interrupt request occurs before the clearing sequence is complete, then writing logic 0 to CHxF has no effect. Therefore, an interrupt request cannot be lost due to inadvertent clearing of CHxF.

Reset clears the CHxF bit. Writing a logic 1 to CHxF has no effect.

1 = Input capture or output compare on channel x
0 = No input capture or output compare on channel x

CHxIE — Channel x Interrupt Enable Bit

This read/write bit enables TIMA CPU interrupts on channel x.
Reset clears the CHxIE bit.

1 = Channel x CPU interrupt requests enabled
0 = Channel x CPU interrupt requests disabled

MSxB — Mode Select Bit B

This read/write bit selects buffered output compare/PWM operation. MSxB exists only in the TIMA channel 0, TIMA channel 2 and TIMA channel 4 status and control registers.

Setting MS0B disables the channel 1 status and control register and reverts TACH1 pin to general-purpose I/O.

Setting MS2B disables the channel 3 status and control register and reverts TACH3 pin to general-purpose I/O.

Setting MS4B disables the channel 5 status and control register and reverts TACH5 pin to general-purpose I/O.

Reset clears the MSxB bit.

1 = Buffered output compare/PWM operation enabled
0 = Buffered output compare/PWM operation disabled

MSxA — Mode Select Bit A

When ELSxB:A ≠ 00, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation. See Table 25-2.

1 = Unbuffered output compare/PWM operation
0 = Input capture operation

Idle

An idle is defined as a passive period greater than 300 μs in length.

27.4.4 J1850 VPW Valid/Invalid Bits and Symbols

The timing tolerances for **receiving** data bits and symbols from the J1850 bus have been defined to allow for variations in oscillator frequencies. In many cases the maximum time allowed to define a data bit or symbol is equal to the minimum time allowed to define another data bit or symbol.

Since the minimum resolution of the BDLC for determining what symbol is being received is equal to a single period of the MUX interface clock (t_{BDLC}), an apparent separation in these maximum time/minimum time concurrences equal to one cycle of t_{BDLC} occurs.

This one clock resolution allows the BDLC to differentiate properly between the different bits and symbols. This is done without reducing the valid window for receiving bits and symbols from transmitters onto the J1850 bus which have varying oscillator frequencies.

In Huntsinger's' variable pulse width (VPW) modulation bit encoding, the tolerances for both the passive and active data bits received and the symbols received are defined with no gaps between definitions. For example, the maximum length of a passive logic 0 is equal to the minimum length of a passive logic 1, and the maximum length of an active logic 0 is equal to the minimum length of a valid SOF symbol.

Invalid Passive Bit

See Figure 27-7 (1). If the passive-to-active received transition beginning the next data bit or symbol occurs between the active-to-passive transition beginning the current data bit (or symbol) and **a**, the current bit would be invalid.

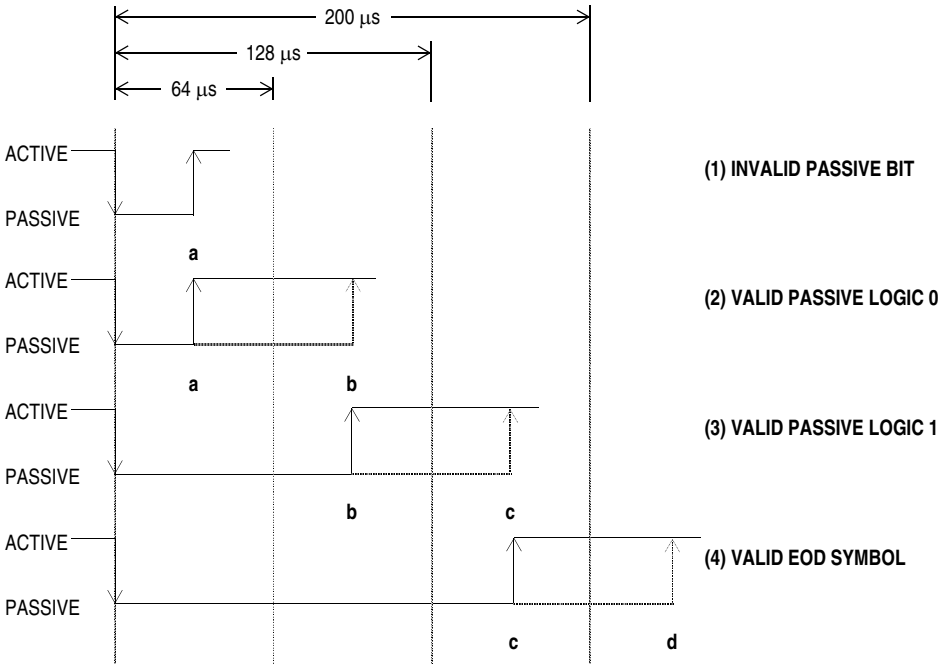


Figure 27-7. J1850 VPW Received Passive Symbol Times

TSIFR — Transmit Single Byte IFR with No CRC (Type 1 or 2) Bit

The TSIFR bit is used to request the BDLC to transmit the byte in the BDLC data register (BDR, $003F) as a single byte IFR with no CRC. Typically, the byte transmitted is a unique identifier or address of the transmitting (responding) node. See Figure 27-18.

1 = If this bit is set prior to a valid EOD being received with no CRC error, once the EOD symbol has been received the BDLC will attempt to transmit the appropriate normalization bit followed by the byte in the BDR.

0 = The TSIFR bit will be cleared automatically, once the BDLC has successfully transmitted the byte in the BDR onto the bus, or TEOD is set, or an error is detected on the bus.

If the programmer attempts to set the TSIFR bit immediately after the EOD symbol has been received from the bus, the TSIFR bit will remain in the reset state and no attempt will be made to transmit the IFR byte.

If a loss of arbitration occurs when the BDLC attempts to transmit and after the IFR byte winning arbitration completes transmission, the BDLC will again attempt to transmit the BDR (with no normalization bit). The BDLC will continue transmission attempts until an error is detected on the bus, or TEOD is set, or the BDLC transmission is successful.

If loss or arbitration occurs in the last two bits of the IFR byte, two additional 1 bits **will not** be sent out because the BDLC will attempt to retransmit the byte in the transmit shift register after the IRF byte winning arbitration completes transmission.

TMIFR1 — Transmit Multiple Byte IFR with CRC (Type 3) Bit

The TMIFR1 bit requests the BDLC to transmit the byte in the BDLC data register (BDR) as the first byte of a multiple byte IFR with CRC or as a single byte IFR with CRC. Response IFR bytes are still subject to J1850 message length maximums (see J1850 Frame Format and Figure 27-18).

1 = If this bit is set prior to a valid EOD being received with no CRC error, once the EOD symbol has been received the BDLC will attempt to transmit the appropriate normalization bit followed by IFR bytes. The programmer should set TEOD after the last IFR byte has been written into the BDR register. After TEOD has been set and the last IFR byte has been transmitted, the CRC byte is transmitted.

0 = The TMIFR1 bit will be cleared automatically – once the BDLC has successfully transmitted the CRC byte and EOD symbol – by the detection of an error on the multiplex bus or by a transmitter underrun caused when the programmer does not write another byte to the BDR after the TDRE interrupt.

If the TMIFR1 bit is set, the BDLC will attempt to transmit the normalization symbol followed by the byte in the BDR. After the byte in the BDR has been loaded into the transmit shift register, a TDRE interrupt (see 27.6.4 BDLC State Vector Register) will occur similar to the main message transmit sequence. The programmer should then load the next byte of the IFR into the BDR for transmission. When the last byte of the IFR has been loaded into the BDR, the programmer should set the TEOD bit in the BDLC control register 2 (BCR2). This will instruct the BDLC to transmit a CRC byte once the byte in the BDR is transmitted and then transmit an EOD symbol, indicating the end of the IFR portion of the message frame.

However, if the programmer wishes to transmit a single byte followed by a CRC byte, the programmer should load the byte into the BDR before the EOD symbol has been received, and then set the TMIFR1 bit. Once the TDRE interrupt occurs, the programmer should then set the TEOD bit in the BCR2. This will result in the byte in the BDR being the only byte transmitted before the IFR CRC byte, and no TDRE interrupt will be generated.

The BDR is double buffered via a transmit shadow register and a receive shadow register. After the byte in the transmit shift register has been transmitted, the byte currently stored in the transmit shadow register is loaded into the transmit shift register. Once the transmit shift register has shifted the first bit out, the TDRE flag is set, and the shadow register is ready to accept the next data byte. The receive shadow register works similarly. Once a complete byte has been received, the receive shift register stores the newly received byte into the receive shadow register. The RDRF flag is set to indicate that a new byte of data has been received. The programmer has one BDLC byte reception time to read the shadow register and clear the RDRF flag before the shadow register is overwritten by the newly received byte.

To abort an in-progress transmission, the programmer should stop loading data into the BDR. This will cause a transmitter underrun error and the BDLC automatically will disable the transmitter on the next non-byte boundary. This means that the earliest a transmission can be halted is after at least one byte plus two extra logic 1s have been transmitted. The receiver will pick this up as an error and relay it in the state vector register as an invalid symbol error.

NOTE

The extra logic 1s are an enhancement to the J1850 protocol which forces a byte boundary condition fault. This is helpful in preventing noise from going onto the J1850 bus from a corrupted message.

27.7 Low-Power Modes

The following information concerns wait mode and stop mode.

27.7.1 Wait Mode

This power-conserving mode is entered automatically from run mode whenever the CPU executes a WAIT instruction and the WCM bit in BDLC control register 1 (BCR1) is previously clear. In BDLC wait mode, the BDLC cannot drive any data.

A subsequent successfully received message, including one that is in progress at the time that this mode is entered, will cause the BDLC to wake up and generate a CPU interrupt request if the interrupt enable (IE) bit in the BDLC control register 1 (BCR1) is previously set. (See 27.6.2 BDLC Control Register 1 for a better understanding of IE.) This results in less of a power saving, but the BDLC is guaranteed to receive correctly the message which woke it up, since the BDLC internal operating clocks are kept running.

NOTE

Ensuring that all transmissions are complete or aborted before putting the BDLC into wait mode is important.

27.7.2 Stop Mode

This power-conserving mode is entered automatically from run mode whenever the CPU executes a STOP instruction or if the CPU executes a WAIT instruction and the WCM bit in the BDLC control register 1 (BCR1) is previously set. This is the lowest power mode that the BDLC can enter.

A subsequent passive-to-active transition on the J1850 bus will cause the BDLC to wake up and generate a non-maskable CPU interrupt request. When a STOP instruction is used to put the BDLC in stop mode, the BDLC is not guaranteed to correctly receive the message which woke it up, since it may take some time for the BDLC internal operating clocks to restart and stabilize. If a WAIT instruction is used to put the BDLC in stop mode, the BDLC is guaranteed to correctly receive the byte which woke it up, if and only if an end-of-frame (EOF) has been detected prior to issuing the WAIT instruction by the CPU. Otherwise, the BDLC will not correctly receive the byte that woke it up.

Appendix B
MC68HC908AZ60E

B.1 Introduction

The MC68HC908AZ60E is a reduced EMC version of the MC68HC908AZ60A. Every care has been taken to insure compatibility with the MC68HC908AZ60A. Some additional features are available, however the default state of all affected modules match the MC68HC908AZ60A functionality. The reset state of all MC68HC908AZ60E registers match the MC68HC908AZ60A except for some reserved memory locations. Although significant design changes have been made to improve the radiated RF emissions from the MCU, all electrical specifications are equal to or better than the MC68HC908AZ60A.

Slew rate controlled outputs have been added to all the general purpose I/O pins as well as the PTC2/MCLK, PTE5/MISO, PTE6/MOSI, and PTE7/SPSCK pins.

Table B-1. External Pins Summary (Sheet 1 of 3)

Pin Name	Function	Driver Type	Hysteresis[1]	Reset State
PTA7–PTA0	General-Purpose I/O	Dual State	No	Input Hi-Z
PTB7/ATD7–PTB0/ATD0	General-Purpose I/O ADC Channels	Dual State	No	Input Hi-Z
PTC5–PTC3	General-Purpose I/O	Dual State	No	Input Hi-Z
PTC2/MCLK	General-Purpose I/O MCLK output	Dual State	No	Input Hi-Z
PTC1–PTC0	General-Purpose I/O	Dual State	No	Input Hi-Z
PTD7	General Purpose I/O	Dual State	No	Input Hi-Z
PTD6/ATD14/TACLK ADC Channel	General-Purpose I/O ADC Channel/Timer External Input Clock	Dual State	Yes, TACLK	Input Hi-Z
PTD5/ATD13 ADC Channel	General-Purpose I/O ADC Channel	Dual State	No	Input Hi-Z
PTD4/ATD12/TBCLK ADC Channel	General-Purpose I/O ADC Channel/Timer External Input Clock	Dual State	Yes, TBCLK	Input Hi-Z
PTD3/ATD11–PTD0/ATD8 ADC Channels	General-Purpose I/O ADC Channel	Dual State	No	Input Hi-Z
PTE7/SPSCK	General-Purpose I/O SPI Clock	Dual State Open Drain	Yes, SPSCK	Input Hi-Z
PTE6/MOSI	General-Purpose I/O SPI Data Path	Dual State Open Drain	Yes, MOSI	Input Hi-Z
PTE5/MISO	General-Purpose I/O SPI Data Path	Dual State Open Drain	Yes, MISO	Input Hi-Z