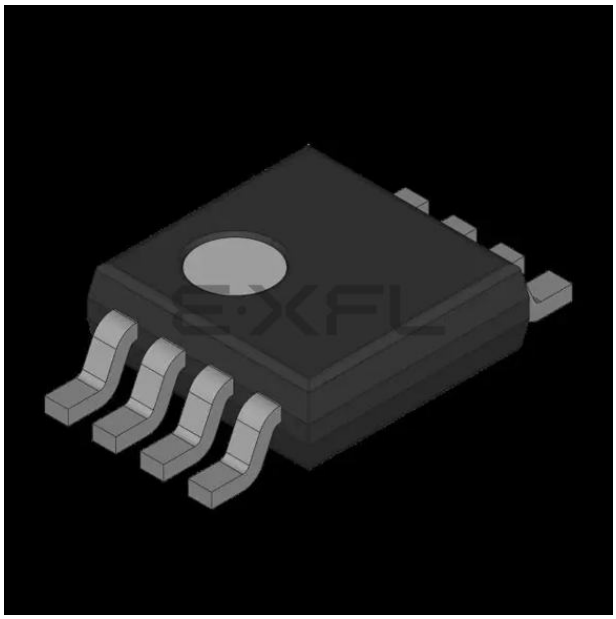


Welcome to E-XFL.COM

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	Active
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/pro/item?MUrl=&PartUrl=mc908az60acfu

Chapter 23
MSCAN Controller (MSCAN08)

Figure 1-5 shows MC68HC908AS60A 52-pin PLCC pin assignments.

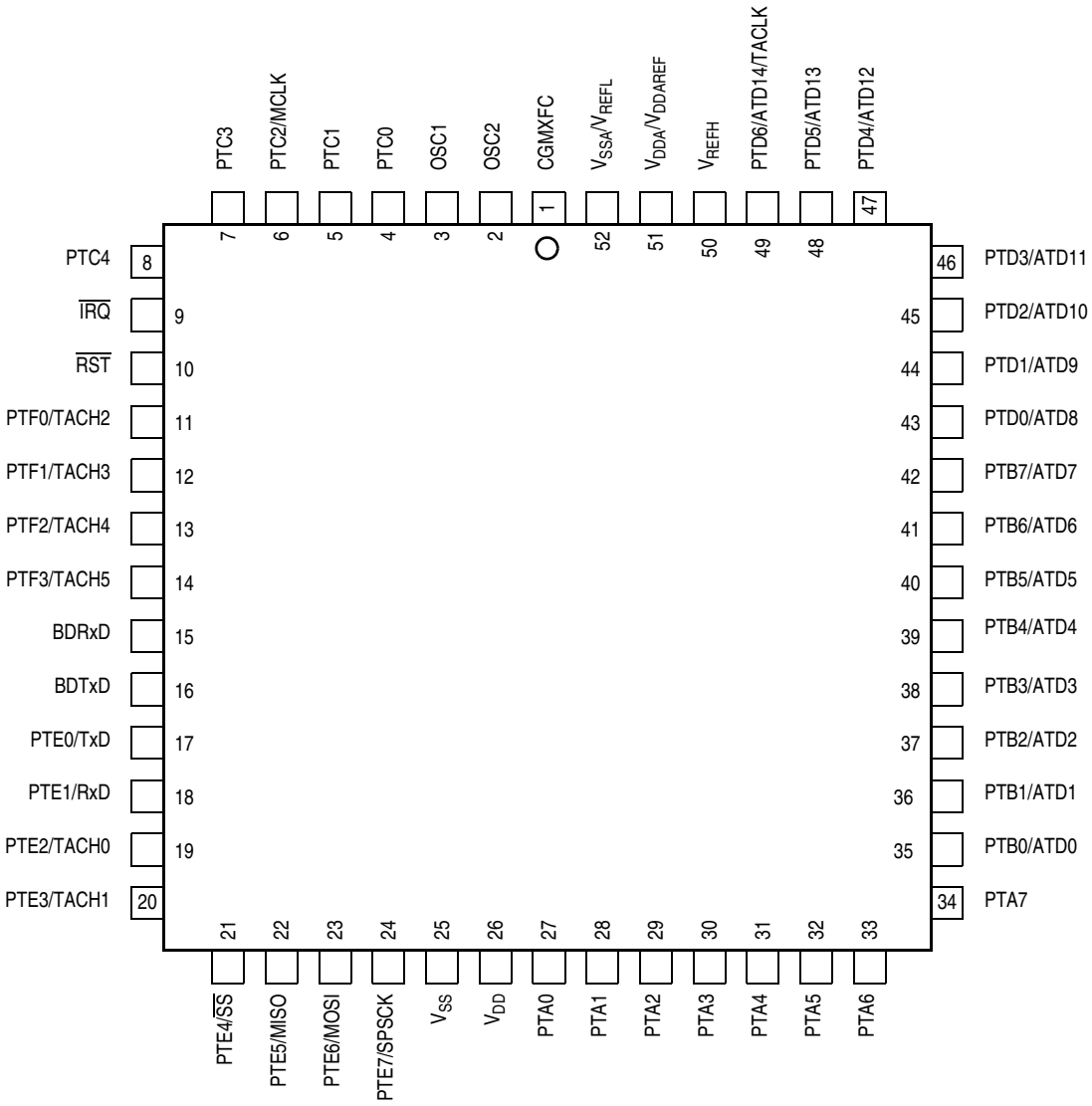


Figure 1-5. MC68HC908AS60A (52-Pin PLCC)

9.3.2 Active Resets from Internal Sources

All internal reset sources actively pull the $\overline{\text{RST}}$ pin low for 32 CGMXCLK cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles (see Figure 9-5). An internal reset can be caused by an illegal address, illegal opcode, COP timeout, LVI, or POR (see Figure 9-6). Note that for LVI or POR resets, the SIM cycles through 4096 CGMXCLK cycles during which the SIM forces the $\overline{\text{RST}}$ pin low. The internal reset signal then follows the sequence from the falling edge of $\overline{\text{RST}}$ shown in Figure 9-5.

The COP reset is asynchronous to the bus clock.

The active reset feature allows the part to issue a reset to peripherals and other chips within a system built around the MCU.

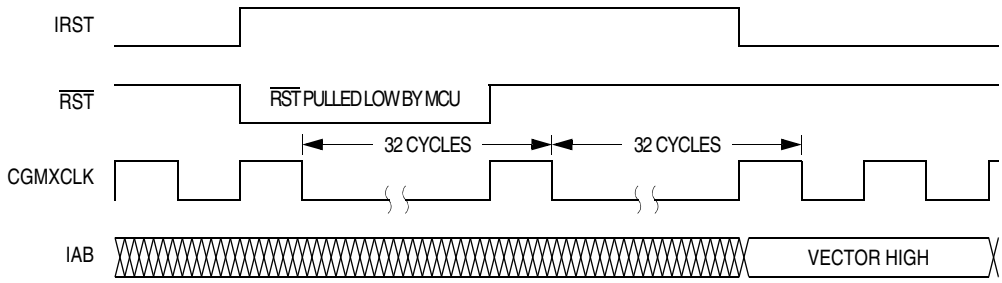


Figure 9-5. Internal Reset Timing

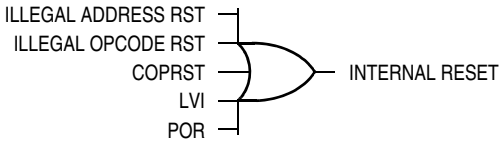


Figure 9-6. Sources of Internal Reset

Table 9-3. PIN Bit Set Timing

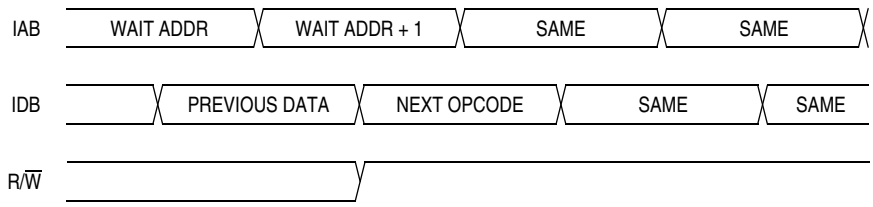
Reset Recovery Type	Actual Number of Cycles
POR/LVI	4163 (4096 + 64 + 3)
All others	67 (64 + 3)

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.

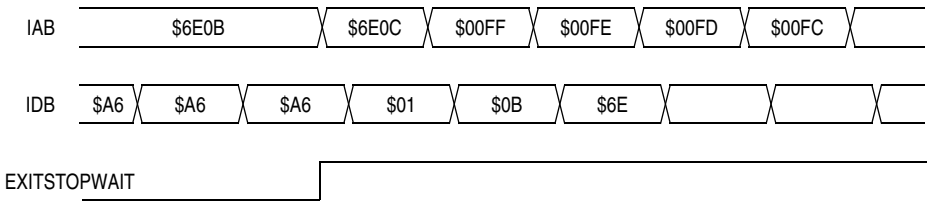
A module that is active during wait mode can wake up the CPU with an interrupt if the interrupt is enabled. Stacking for the interrupt begins one cycle after the WAIT instruction during which the interrupt occurred. Refer to the wait mode subsection of each module to see if the module is active or inactive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

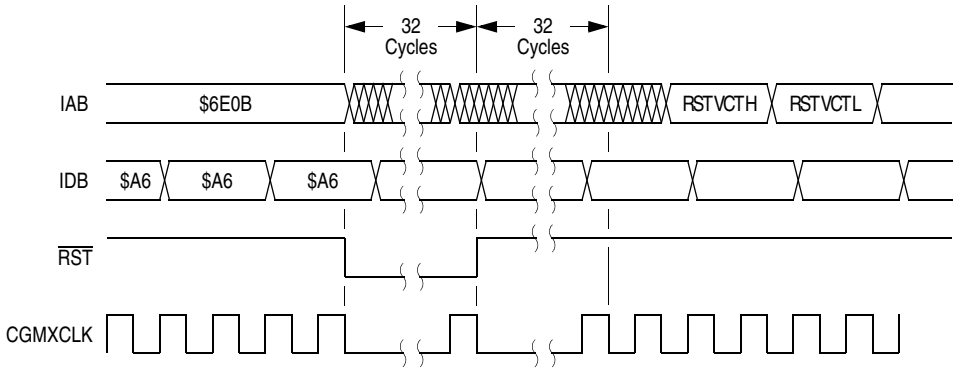


Figure 9-14. Wait Recovery from Internal Reset

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

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	

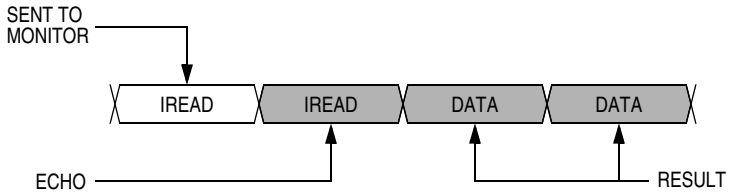


Table 14-6. IWRITE (Indexed Write) Command

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

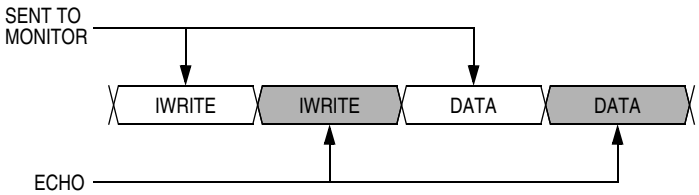
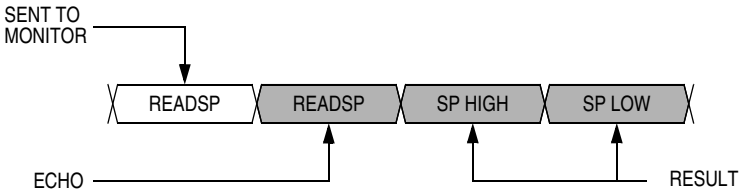


Table 14-7. READSP (Read Stack Pointer) Command

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



15.7.2 Stop Mode

Stop mode turns off the CGMXCLK input to the COP and clears the COP prescaler. Service the COP immediately before entering or after exiting stop mode to ensure a full COP timeout period after entering or exiting stop mode.

The STOP bit in the configuration register (CONFIG) enables the STOP instruction. To prevent inadvertently turning off the COP with a STOP instruction, disable the STOP instruction by clearing the STOP bit.

15.8 COP Module During Break Interrupts

The COP is disabled during a break interrupt when V_{Hi} is present on the $\overline{RST}$ pin.

Chapter 17
External Interrupt Module (IRQ)

17.1 Introduction

This chapter describes the nonmaskable external interrupt (IRQ) input.

17.2 Features

Features include:

- Dedicated External Interrupt Pin ($\overline{\text{IRQ}}$)
- Hysteresis Buffer
- Programmable Edge-Only or Edge- and Level-Interrupt Sensitivity
- Automatic Interrupt Acknowledge

17.3 Functional Description

A falling edge applied to the external interrupt pin can latch a CPU interrupt request. Figure 17-1 shows the structure of the IRQ module.

Interrupt signals on the $\overline{\text{IRQ}}$ pin are latched into the IRQ latch. An interrupt latch remains set until one of the following actions occurs:

- Vector fetch — A vector fetch automatically generates an interrupt acknowledge signal that clears the latch that caused the vector fetch.
- Software clear — Software can clear an interrupt latch by writing to the appropriate acknowledge bit in the interrupt status and control register (ISCR). Writing a logic 1 to the ACK bit clears the IRQ latch.
- Reset — A reset automatically clears both interrupt latches.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
SCI Control Register 1 (SCC1)	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
	Write:								
	Reset:	0	0	0	0	0	0	0	0
SCI Control Register 2 (SCC2)	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
	Write:								
	Reset:	0	0	0	0	0	0	0	0
SCI Control Register 3 (SCC3)	Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
	Write:								
	Reset:	U	U	0	0	0	0	0	0
SCI Status Register 1 (SCS1)	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
	Write:								
	Reset:	1	1	0	0	0	0	0	0
SCI Status Register 2 (SCS2)	Read:	0	0	0	0	0	0	BKF	RPF
	Write:								
	Reset:	0	0	0	0	0	0	0	0
SCI Data Register (SCDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
	Write:	T7	T6	T5	T4	T3	T2	T1	T0
	Reset:	Unaffected by Reset							
SCI Baud Rate Register (SCBR)	Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
	Write:								
	Reset:	0	0	0	0	0	0	0	0

= Unimplemented U = Unaffected R = Reserved

Figure 18-2. SCI I/O Register Summary

Table 18-2. SCI I/O Register Address Summary

Register	SCC1	SCC2	SCC3	SCS1	SCS2	SCDR	SCBR
Address	$0013	$0014	$0015	$0016	$0017	$0018	$0019

18.8.6 SCI Data Register

The SCI data register is the buffer between the internal data bus and the receive and transmit shift registers. Reset has no effect on data in the SCI data register.

Address: $0018

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R7	R6	R5	R4	R3	R2	R1	R0
Write:	T7	T6	T5	T4	T3	T2	T1	T0
Reset:	Unaffected by Reset							

Figure 18-17. SCI Data Register (SCDR)

R7/T7:R0/T0 — Receive/Transmit Data Bits

Reading address $0018 accesses the read-only received data bits, R7:R0. Writing to address $0018 writes the data to be transmitted, T7:T0. Reset has no effect on the SCI data register.

NOTE

Do not use read-modify-write instructions on the SCI data register.

18.8.7 SCI Baud Rate Register

The baud rate register selects the baud rate for both the receiver and the transmitter.

Address: $0019

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved

Figure 18-18. SCI Baud Rate Register (SCBR)

SCP1 and SCP0 — SCI Baud Rate Prescaler Bits

These read/write bits select the baud rate prescaler divisor as shown in Table 18-9. Reset clears SCP1 and SCP0.

Table 18-9. SCI Baud Rate Prescaling

SCP[1:0]	Prescaler Divisor (PD)
00	1
01	3
10	4
11	13

SCR2–SCR0 — SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate divisor as shown in Table 18-10. Reset clears SCR2–SCR0.

19.5.3 Transmission Format When CPHA = 1

Figure 19-5 shows an SPI transmission in which CPHA (SPCR) is logic 1. The figure should not be used as a replacement for data sheet parametric information. Two waveforms are shown for SCK: one for CPOL = 0 and another for CPOL = 1. The diagram may be interpreted as a master or slave timing diagram since the serial clock (SCK), master in/slave out (MISO), and master out/slave in (MOSI) pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the slave select input to the slave. The slave SPI drives its MISO output only when its slave select input ($\overline{SS}$) is low, so that only the selected slave drives to the master. The $\overline{SS}$ pin of the master is not shown but is assumed to be inactive. The $\overline{SS}$ pin of the master must be high or must be reconfigured as general-purpose I/O not affecting the SPI. (See 19.6.2 Mode Fault Error). When CPHA = 1, the master begins driving its MOSI pin on the first SPSCK edge. Therefore, the slave uses the first SPSCK edge as a start transmission signal. The $\overline{SS}$ pin can remain low between transmissions. This format may be preferable in systems having only one master and only one slave driving the MISO data line.

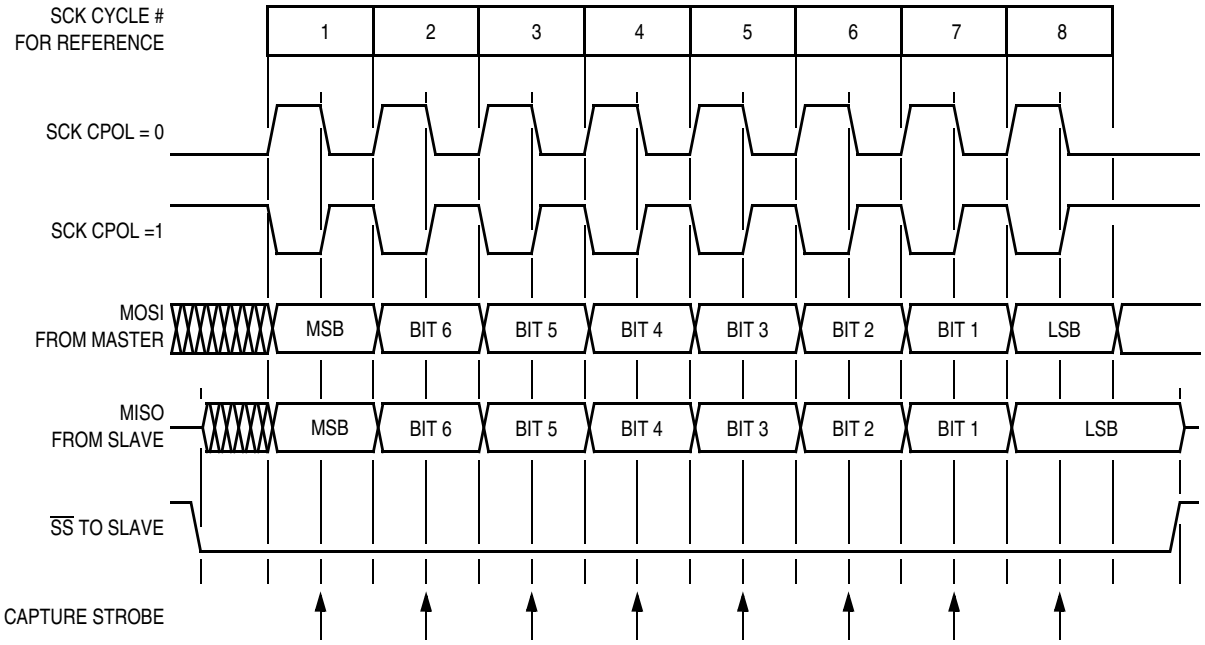


Figure 19-5. Transmission Format (CPHA = 1)

19.5.4 Transmission Initiation Latency

When the SPI is configured as a master (SPMSTR = 1), transmissions are started by a software write to the SPDR ($0012). CPHA has no effect on the delay to the start of the transmission, but it does affect the initial state of the SCK signal. When CPHA = 0, the SCK signal remains inactive for the first half of the first SCK cycle. When CPHA = 1, the first SCK cycle begins with an edge on the SCK line from its inactive to its active level. The SPI clock rate (selected by SPR1–SPR0) affects the delay from the write to SPDR and the start of the SPI transmission. (See Figure 19-6). The internal SPI clock in the master is a free-running derivative of the internal MCU clock. It is only enabled when both the SPE and SPMSTR bits (SPCR) are set to conserve power. SCK edges occur half way through the low time of the internal MCU clock. Since the SPI clock is free-running, it is uncertain where the write to the SPDR will occur relative to the slower SCK. This uncertainty causes the variation in the initiation delay shown in Figure 19-6. This

19.13.3 SPI Data Register

The SPI data register is the read/write buffer for the receive data register and the transmit data register. Writing to the SPI data register writes data into the transmit data register. Reading the SPI data register reads data from the receive data register. The transmit data and receive data registers are separate buffers that can contain different values. (See Figure 19-2.)

Address: $0012

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R7	R6	R5	R4	R3	R2	R1	R0
Write:	T7	T6	T5	T4	T3	T2	T1	T0
Reset:	Indeterminate after Reset							

Figure 19-14. SPI Data Register (SPDR)

R7–R0/T7–T0 — Receive/Transmit Data Bits

NOTE

Do not use read-modify-write instructions on the SPI data register since the buffer read is not the same as the buffer written.

Address: $0040

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
Write:	0			TRST	R			
Reset:	0	0	1	0	0	0	0	0

R = Reserved

Figure 20-4. TIMB Status and Control Register (TBSC)

TOF — TIMB Overflow Flag Bit

This read/write flag is set when the TIMB counter reaches the modulo value programmed in the TIMB counter modulo registers. Clear TOF by reading the TIMB status and control register when TOF is set and then writing a logic 0 to TOF. If another TIMB overflow occurs before the clearing sequence is complete, then writing logic 0 to TOF has no effect. Therefore, a TOF interrupt request cannot be lost due to inadvertent clearing of TOF. Reset clears the TOF bit. Writing a logic 1 to TOF has no effect.

1 = TIMB counter has reached modulo value
0 = TIMB counter has not reached modulo value

TOIE — TIMB Overflow Interrupt Enable Bit

This read/write bit enables TIMB overflow interrupts when the TOF bit becomes set. Reset clears the TOIE bit.

1 = TIMB overflow interrupts enabled
0 = TIMB overflow interrupts disabled

TSTOP — TIMB Stop Bit

This read/write bit stops the TIMB counter. Counting resumes when TSTOP is cleared. Reset sets the TSTOP bit, stopping the TIMB counter until software clears the TSTOP bit.

1 = TIMB counter stopped
0 = TIMB counter active

NOTE

Do not set the TSTOP bit before entering wait mode if the TIMB is required to exit wait mode. Also, when the TSTOP bit is set and the timer is configured for input capture operation, input captures are inhibited until TSTOP is cleared.

When using TSTOP to stop the timer counter, see if any timer flags are set. If a timer flag is set, it must be cleared by clearing TSTOP, then clearing the flag, then setting TSTOP again.

TRST — TIMB Reset Bit

Setting this write-only bit resets the TIMB counter and the TIMB prescaler. Setting TRST has no effect on any other registers. Counting resumes from $0000. TRST is cleared automatically after the TIMB counter is reset and always reads as logic 0. Reset clears the TRST bit.

1 = Prescaler and TIMB counter cleared
0 = No effect

NOTE

Setting the TSTOP and TRST bits simultaneously stops the TIMB counter at a value of $0000.

21.7.3 PIT Counter Modulo Registers

The read/write PIT modulo registers contain the modulo value for the PIT counter. When the PIT counter reaches the modulo value the overflow flag (POF) becomes set and the PIT counter resumes counting from $0000 at the next timer clock. Writing to the high byte (PMODH) inhibits the POF bit and overflow interrupts until the low byte (PMODL) is written. Reset sets the PIT counter modulo registers.

Address: $004E:$004F

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

Address: $004E:$004F

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

Figure 21-5. PIT Counter Modulo Registers (PMODH–PMODL)

NOTE

Reset the PIT counter before writing to the PIT counter modulo registers.

22.3 Port B

Port B is an 8-bit special function port that shares all of its pins with the analog-to-digital converter.

22.3.1 Port B Data Register

The port B data register contains a data latch for each of the eight port B pins.

Address:	$0001							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Reset:	Unaffected by Reset							
Alternative Functions:	ATD7	ATD6	ATD5	ATD4	ATD3	ATD2	ATD1	ATD0

Figure 22-5. Port B Data Register (PTB)

PTB[7:0] — Port B Data Bits

These read/write bits are software programmable. Data direction of each port B pin is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.

ATD[7:0] — ADC Channels

PTB7/ATD7–PTB0/ATD0 are eight of the analog-to-digital converter channels. The ADC channel select bits, CH[4:0], determine whether the PTB7/ATD7–PTB0/ATD0 pins are ADC channels or general-purpose I/O pins. If an ADC channel is selected and a read of this corresponding bit in the port B data register occurs, the data will be 0 if the data direction for this bit is programmed as an input. Otherwise, the data will reflect the value in the data latch. (See Chapter 26 Analog-to-Digital Converter (ADC)). Data direction register B (DDRB) does not affect the data direction of port B pins that are being used by the ADC. However, the DDRB bits always determine whether reading port B returns to the states of the latches or 0.

22.7.1 Port F Data Register

The port F data register contains a data latch for each of the seven port F pins.

Address:	$0009							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PTF6	PTF5	PTF4	PTF3	PTF2	PTF1	PTF0
Write:	R							
Reset:	Unaffected by Reset							
Alternative Function:			TBCH1	TBCH0	TACH5	TACH4	TACH3	TACH2

R = Reserved

Figure 22-17. Port F Data Register (PTF)

PTF[6:0] — Port F Data Bits

These read/write bits are software programmable. Data direction of each port F pin is under the control of the corresponding bit in data direction register F. Reset has no effect on PTF[6:0].

TACH[5:2] — Timer A Channel I/O Bits

The PTF3–PTF0/TACH2 pins are the TIM input capture/output compare pins. The edge/level select bits, ELSxB:ELSxA, determine whether the PTF3–PTF0/TACH2 pins are timer channel I/O pins or general-purpose I/O pins. (See 25.8.1 TIMA Status and Control Register).

TBCH[1:0] — Timer B Channel I/O Bits

The PTF5/TBCH1–PTF4/TBCH0 pins are the TIMB input capture/output compare pins. The edge/level select bits, ELSxB:ELSxA, determine whether the PTF5/TBCH1–PTF4/TBCH0 pins are timer channel I/O pins or general-purpose I/O pins. (See 20.8.1 TIMB Status and Control Register).

NOTE

Data direction register F (DDRF) does not affect the data direction of port F pins that are being used by the TIM. However, the DDRF bits always determine whether reading port F returns the states of the latches or the states of the pins. (See Table 22-6).

22.7.2 Data Direction Register F

Data direction register F determines whether each port F pin is an input or an output. Writing a logic 1 to a DDRF bit enables the output buffer for the corresponding port F pin; a logic 0 disables the output buffer.

Address:	$000D							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	DDRF6	DDRF5	DDRF4	DDRF3	DDRF2	DDRF1	DDRF0
Write:	R							
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 22-18. Data Direction Register F (DDRF)

DDRF[6:0] — Data Direction Register F Bits

These read/write bits control port F data direction. Reset clears DDRF[6:0], configuring all port F pins as inputs.

1 = Corresponding port F pin configured as output
0 = Corresponding port F pin configured as input

NOTE

Avoid glitches on port F pins by writing to the port F data register before changing data direction register F bits from 0 to 1.

Figure 22-19 shows the port F I/O logic.

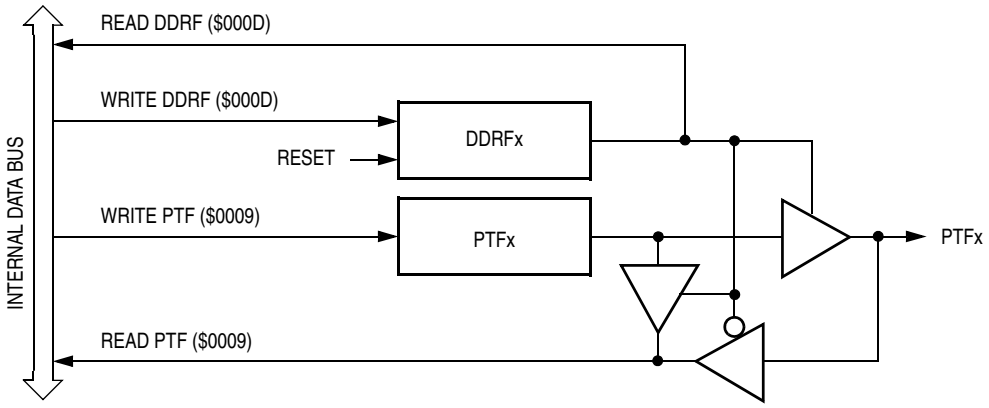


Figure 22-19. Port F I/O Circuit

When bit DDRFx is a logic 1, reading address $0009 reads the PTFx data latch. When bit DDRFx is a logic 0, reading address $0009 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 22-6 summarizes the operation of the port F pins.

Table 22-6. Port F Pin Functions

DDRF Bit	PTF Bit	I/O Pin Mode	Accesses to DDRF	Accesses to PTF	
			Read/Write	Read	Write
0	X	Input, Hi-Z	DDRF[6:0]	Pin	PTF[6:0](1)
1	X	Output	DDRF[6:0]	PTF[6:0]	PTF[6:0]

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

To transmit a message, the CPU08 has to identify an available transmit buffer which is indicated by a set transmit buffer empty (TXE) flag in the MSCAN08 transmitter flag register (CTFLG) (see 23.13.7 MSCAN08 Transmitter Flag Register).

The CPU08 then stores the identifier, the control bits and the data content into one of the transmit buffers. Finally, the buffer has to be flagged ready for transmission by clearing the TXE flag.

The MSCAN08 then will schedule the message for transmission and will signal the successful transmission of the buffer by setting the TXE flag. A transmit interrupt is generated[1] when TXE is set and can be used to drive the application software to re-load the buffer.

In case more than one buffer is scheduled for transmission when the CAN bus becomes available for arbitration, the MSCAN08 uses the local priority setting of the three buffers for prioritisation. For this purpose, every transmit buffer has an 8-bit local priority field (PRIO). The application software sets this field when the message is set up. The local priority reflects the priority of this particular message relative to the set of messages being emitted from this node. The lowest binary value of the PRIO field is defined as the highest priority.

The internal scheduling process takes place whenever the MSCAN08 arbitrates for the bus. This is also the case after the occurrence of a transmission error.

When a high priority message is scheduled by the application software, it may become necessary to abort a lower priority message being set up in one of the three transmit buffers. As messages that are already under transmission cannot be aborted, the user has to request the abort by setting the corresponding abort request flag (ABTRQ) in the transmission control register (CTCR). The MSCAN08 will then grant the request, if possible, by setting the corresponding abort request acknowledge (ABTAK) and the TXE flag in order to release the buffer and by generating a transmit interrupt. The transmit interrupt handler software can tell from the setting of the ABTAK flag whether the message was actually aborted (ABTAK = 1) or sent (ABTAK = 0).

23.5 Identifier Acceptance Filter

The Identifier Acceptance Registers (CIDAR0-3) define the acceptance patterns of the standard or extended identifier (ID10-ID0 or ID28-ID0). Any of these bits can be marked ‘don’t care’ in the Identifier Mask Registers (CIDMR0-3).

A filter hit is indicated to the application on software by a set RXF (Receive Buffer Full Flag, see 23.13.5 MSCAN08 Receiver Flag Register (CRFLG)) and two bits in the Identifier Acceptance Control Register (see 23.13.9 MSCAN08 Identifier Acceptance Control Register). These Identifier Hit Flags (IDHIT1-0) clearly identify the filter section that caused the acceptance. They simplify the application software’s task to identify the cause of the receiver interrupt. In case that more than one hit occurs (two or more filters match) the lower hit has priority.

A very flexible programmable generic identifier acceptance filter has been introduced to reduce the CPU interrupt loading. The filter is programmable to operate in four different modes:

- Single identifier acceptance filter, each to be applied to a) the full 29 bits of the extended identifier and to the following bits of the CAN frame: RTR, IDE, SRR or b) the 11 bits of the standard identifier plus the RTR and IDE bits of CAN 2.0A/B messages. This mode implements a single filter for a full length CAN 2.0B compliant extended identifier. Figure 23-3 shows how the 32-bit filter bank (CIDAR0-3, CIDMR0-3) produces a filter 0 hit.

1. The transmit interrupt will occur only if not masked. A polling scheme can be applied on TXE also.

23.13 Programmer's Model of Control Registers

The programmer's model has been laid out for maximum simplicity and efficiency. Figure 23-14 gives an overview on the control register block of the MSCAN08.

Addr	Register		Bit 7	6	5	4	3	2	1	Bit 0
$0500	CMCR0	Read:	0	0	0	SYNCH	TLNKEN	SLPAK	SLPRQ	SFTRES
		Write:								
$0501	CMCR1	Read:	0	0	0	0	0	LOOPB	WUPM	CLKSRC
		Write:								
$0502	CBTR0	Read:	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
		Write:								
$0503	CBTR1	Read:	SAMP	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
		Write:								
$0504	CRFLG	Read:	WUPIF	RWRNIF	TWRNIF	RERRIF	TERRIF	BOFFIF	OVRIF	RXF
		Write:								
$0505	CRIER	Read:	WUPIE	RWRNIE	TWRNIE	RERRIE	TERRIE	BOFFIE	OVRIE	RXFIE
		Write:								
$0506	CTFLG	Read:	0	ABTAK2	ABTAK1	ABTAK0	0	TXE2	TXE1	TXE0
		Write:								
$0507	CTCR	Read:	0	ABTRQ2	ABTRQ1	ABTRQ0	0	TXEIE2	TXEIE1	TXEIE0
		Write:								
$0508	CIDAC	Read:	0	0	IDAM1	IDAM0	0	0	IDHIT1	IDHIT0
		Write:								
$0509	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
$050E	CRXERR	Read:	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
		Write:								
$050F	CTXERR	Read:	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
		Write:								
$0510	CIDAR0	Read:	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		Write:								
$0511	CIDAR1	Read:	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		Write:								
$0512	CIDAR2	Read:	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		Write:								
$0513	CIDAR3	Read:	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		Write:								
$0514	CIDMR0	Read:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		Write:								
$0515	CIDMR1	Read:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		Write:								
$0516	CIDMR2	Read:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		Write:								
$0517	CIDMR3	Read:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		Write:								

= Unimplemented R = Reserved

Figure 23-14. MSCAN08 Control Register Structure

Valid Passive Logic 0

See Figure 27-7 (2). If the passive-to-active received transition beginning the next data bit (or symbol) occurs between **a** and **b**, the current bit would be considered a logic 0.

Valid Passive Logic 1

See Figure 27-7 (3). If the passive-to-active received transition beginning the next data bit (or symbol) occurs between **b** and **c**, the current bit would be considered a logic 1.

Valid EOD Symbol

See Figure 27-7 (4). If the passive-to-active received transition beginning the next data bit (or symbol) occurs between **c** and **d**, the current symbol would be considered a valid end-of-data symbol (EOD).

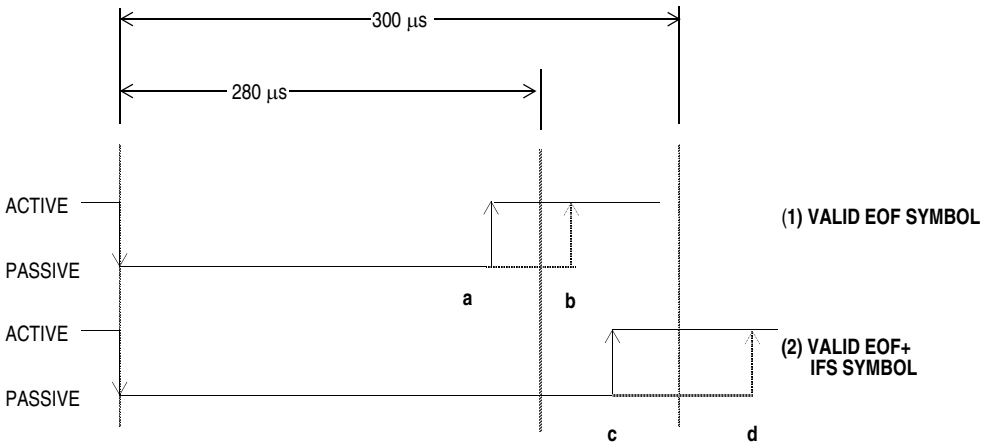


Figure 27-8. J1850 VPW Received Passive EOF and IFS Symbol Times

Valid EOF and IFS Symbol

In Figure 27-8 (1), if the passive-to-active received transition beginning the SOF symbol of the next message occurs between **a** and **b**, the current symbol will be considered a valid end-of-frame (EOF) symbol.

See Figure 27-8 (2). If the passive-to-active received transition beginning the SOF symbol of the next message occurs between c and d, the current symbol will be considered a valid EOF symbol followed by a valid inter-frame separation symbol (IFS). All nodes must wait until a valid IFS symbol time has expired before beginning transmission. However, due to variations in clock frequencies and bus loading, some nodes may recognize a valid IFS symbol before others and immediately begin transmitting. Therefore, any time a node waiting to transmit detects a passive-to-active transition once a valid EOF has been detected, it should immediately begin transmission, initiating the arbitration process.

Idle Bus

In Figure 27-8 (2), if the passive-to-active received transition beginning the start-of-frame (SOF) symbol of the next message does not occur before **d,** the bus is considered to be idle, and any node wishing to transmit a message may do so immediately.

Invalid Active Bit

In Figure 27-9 (1), if the active-to-passive received transition beginning the next data bit (or symbol) occurs between the passive-to-active transition beginning the current data bit (or symbol) and **a**, the current bit would be invalid.