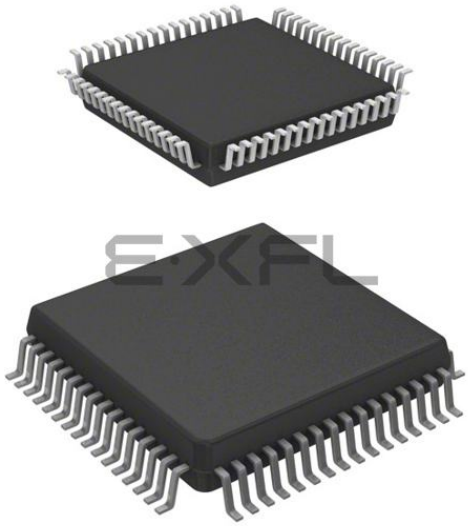

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	IRSCI, SPI
Peripherals	LCD, LVD, POR, PWM
Number of I/O	32
Program Memory Size	12KB (12K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc98lj12cfue

MC68HC908LJ12

Technical Data

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale was negligent regarding the design or manufacture of the part. Freescale, Inc. is an Equal Opportunity/Affirmative Action Employer.

Section 10. Monitor ROM (MON)

List of Figures

Features of the CPU08 include the following:

- Enhanced HC05 programming model
- Extensive loop control functions
- 16 addressing modes (eight more than the HC05)
- 16-bit Index register and stack pointer
- Memory-to-memory data transfers
- Fast 8 × 8 multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Efficient C language support

1.4 MCU Block Diagram

Figure 1-1 shows the structure of the MC68HC908LJ12.

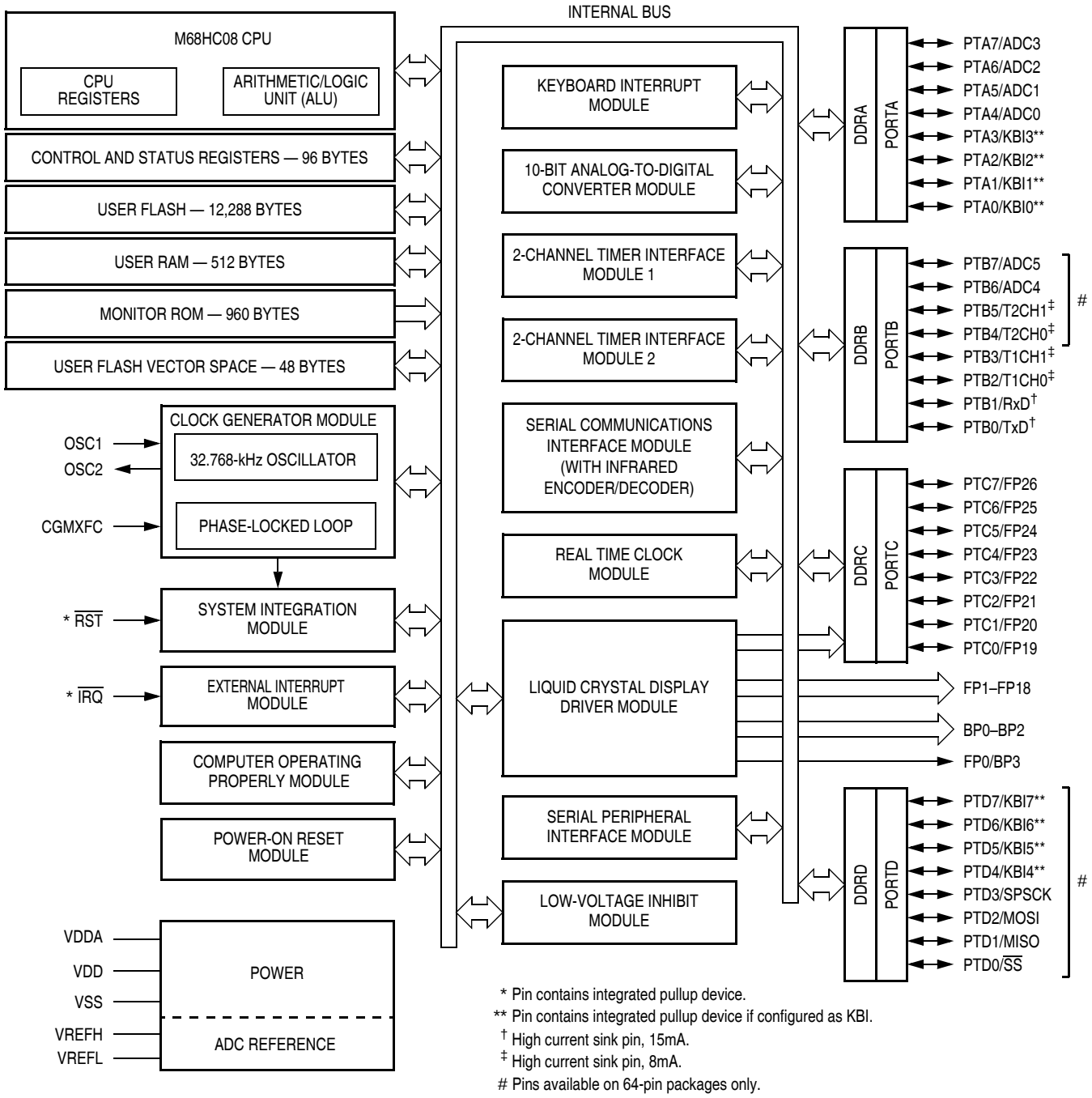

Figure 1-1. MC68HC908LJ12 Block Diagram

I — Interrupt Mask

When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the interrupt vector is fetched.

1 = Interrupts disabled
0 = Interrupts enabled

NOTE: *To maintain M6805 Family compatibility, the upper byte of the index register (H) is not stacked automatically. If the interrupt service routine modifies H, then the user must stack and unstack H using the PSHH and PULH instructions.*

After the I bit is cleared, the highest-priority interrupt request is serviced first.

A return-from-interrupt (RTI) instruction pulls the CPU registers from the stack and restores the interrupt mask from the stack. After any reset, the interrupt mask is set and can be cleared only by the clear interrupt mask software instruction (CLI).

N — Negative Flag

The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result.

1 = Negative result
0 = Non-negative result

Z — Zero Flag

The CPU sets the zero flag when an arithmetic operation, logic operation, or data manipulation produces a result of $00.

1 = Zero result
0 = Non-zero result

6.6.2 Stop Mode

The STOP instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling external interrupts. After exit from stop mode by external interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock.

After exiting stop mode, the CPU clock begins running after the oscillator stabilization delay.

6.7 CPU During Break Interrupts

If the break module is enabled, a break interrupt causes the CPU to execute the software interrupt instruction (SWI) at the completion of the current CPU instruction. (See **Section 22. Break Module (BRK)**.) The program counter vectors to $FFFC–$FFFD ($FEFC–$FEFD in monitor mode).

A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation if the break interrupt has been deasserted.

6.8 Instruction Set Summary

Table 6-1 provides a summary of the M68HC08 instruction set.

6.9 Opcode Map

The opcode map is provided in **Table 6-2**.

8.4.7 Special Programming Exceptions

The programming method described in **8.4.6 Programming the PLL** does not account for three possible exceptions. A value of 0 for R, N, or L is meaningless when used in the equations given. To account for these exceptions:

- A 0 value for R or N is interpreted exactly the same as a value of 1.
- A 0 value for L disables the PLL and prevents its selection as the source for the base clock.

(See **8.4.8 Base Clock Selector Circuit**.)

8.4.8 Base Clock Selector Circuit

This circuit is used to select either the oscillator clock, CGMXCLK, or the divided VCO clock, CGMPCLK, as the source of the base clock, CGMOUT. The two input clocks go through a transition control circuit that waits up to three CGMXCLK cycles and three CGMPCLK cycles to change from one clock source to the other. During this time, CGMOUT is held in stasis. The output of the transition control circuit is then divided by two to correct the duty cycle. Therefore, the bus clock frequency, which is one-half of CGMOUT, is one-fourth the frequency of the selected clock (CGMXCLK or CGMPCLK).

For the CGMXCLK, the divide-by-2 can be by-passed by setting the DIV2CLK bit in the CONFIG2 register. Therefore, the bus clock frequency can be one-half of CGMXCLK.

The BCS bit in the PLL control register (PCTL) selects which clock drives CGMOUT. The divided VCO clock cannot be selected as the base clock source if the PLL is not turned on. The PLL cannot be turned off if the divided VCO clock is selected. The PLL cannot be turned on or off simultaneously with the selection or deselection of the divided VCO clock. The divided VCO clock also cannot be selected as the base clock source if the factor L is programmed to a 0. This value would set up a condition inconsistent with the operation of the PLL, so that the PLL would be disabled and the oscillator clock would be forced as the source of the base clock.

9.6.1.1 Hardware Interrupts

A hardware interrupt does not stop the current instruction. Processing of a hardware interrupt begins after completion of the current instruction. When the current instruction is complete, the SIM checks all pending hardware interrupts. If interrupts are not masked (I bit clear in the condition code register) and if the corresponding interrupt enable bit is set, the SIM proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If more than one interrupt is pending at the end of an instruction execution, the highest priority interrupt is serviced first. **Figure 9-11** demonstrates what happens when two interrupts are pending. If an interrupt is pending upon exit from the original interrupt service routine, the pending interrupt is serviced before the LDA instruction is executed.

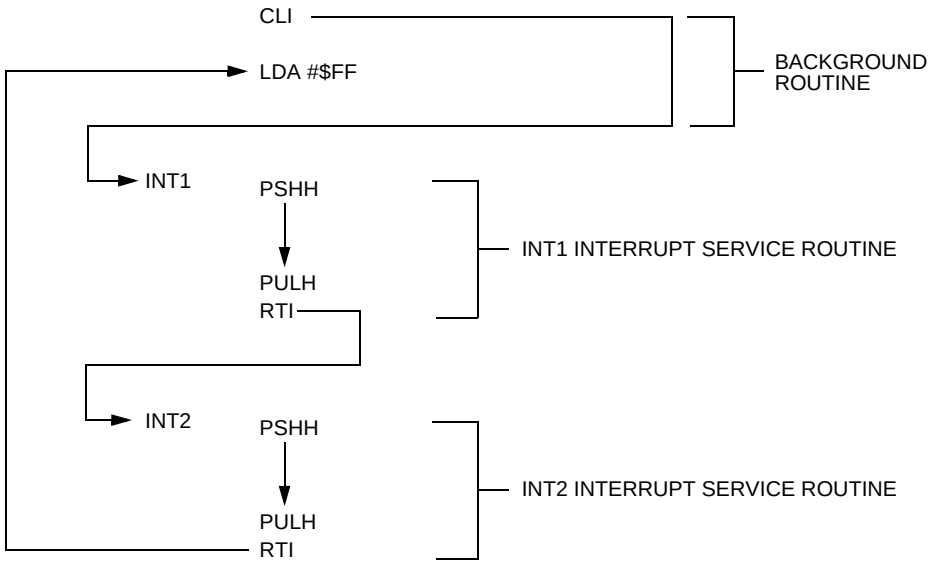

Figure 9-11. Interrupt Recognition Example

The LDA opcode is prefetched by both the INT1 and INT2 RTI instructions. However, in the case of the INT1 RTI prefetch, this is a redundant operation.

NOTE: *To maintain compatibility with the M6805 Family, the H register is not pushed on the stack during interrupt entry. If the interrupt service routine modifies the H register or uses the indexed addressing mode, software should save the H register and then restore it prior to exiting the routine.*

Table 10-1. Monitor Mode Signal Requirements and Options

$\overline{IRQ}$	$\overline{RST}$	Address $FFFE/ $FFFF	PTA2	PTA1	PTA0[1]	PTC1	External Clock[2]	Bus Frequency	PLL	COP	Baud Rate	Comment
X	GND	X	X	X	X	X	X	0	X	Disabled	0	No operation until reset goes high
V_{TST}[3]	V_{DD} or V_{TST}	X	0	1	1	0	4.9152 MHz	2.4576 MHz	OFF	Disabled	9600	PTA1 and PTA2 voltages only required if $\overline{IRQ}$ = V_{TST}; PTC1 determines frequency divider
V_{TST}[3]	V_{DD} or V_{TST}	X	0	1	1	1	9.8304 MHz	2.4576 MHz	OFF	Disabled	9600	PTA1 and PTA2 voltages only required if $\overline{IRQ}$ = V_{TST}; PTC1 determines frequency divider
V_{DD}	V_{DD}	Blank "$FFFF"	X	X	1	X	9.8304 MHz	2.4576 MHz	OFF	Disabled	9600	External frequency always divided by 4
GND	V_{DD}	Blank "$FFFF"	X	X	1	X	32.768 kHz	2.4576 MHz	ON	Disabled	9600	PLL enabled (BCS set) in monitor code
V_{DD} or GND	V_{TST}	Blank "$FFFF"	X	X	X	X	X	—	OFF	Enabled	—	Enters user mode — will encounter an illegal address reset
V_{DD} or GND	V_{DD} or V_{TST}	Not Blank	X	X	X	X	X	—	OFF	Enabled	—	Enters user mode

Notes:
1. PTA0 = 1 if serial communication; PTA0 = 0 if parallel communication
2. External clock is derived by a 32.768 kHz crystal or a 4.9152/9.8304 MHz off-chip oscillator
3. Monitor mode entry by $\overline{IRQ}$ = V_{TST}, a 4.9152/9.8304 MHz off-chip oscillator must be used. The MCU internal crystal oscillator circuit is bypassed.

10.6.1 PRGRNGE

PRGRNGE is used to program a range of FLASH locations with data loaded into the data array.

Table 10-11. PRGRNGE Routine

Routine Name	PRGRNGE
Routine Description	Program a range of locations
Calling Address	$FC06
Stack Used	14 bytes
Data Block Format	Bus speed (BUS_SPD) Data size (DATASIZE) Start address high (ADDRH) Start address (ADDRL) Data 1 (DATA1) : Data N (DATAN)

The start location of the FLASH to be programmed is specified by the address ADDRH:ADDRL and the number of bytes from this location is specified by DATASIZE. The maximum number of bytes that can be programmed in one routine call is 255 bytes (max. DATASIZE is 255).

ADDRH:ADDRL do not need to be at a page boundary, the routine handles any boundary misalignment during programming. A check to see that all bytes in the specified range are erased is not performed by this routine prior programming. Nor does this routine do a verification after programming, so there is no return confirmation that programming was successful. User must assure that the range specified is first erased.

The coding example below is to program 64 bytes of data starting at FLASH location $EF00, with a bus speed of 4.9152 MHz. The coding assumes the data block is already loaded in RAM, with the address pointer, FILE_PTR, pointing to the first byte of the data block.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0032	Timer 2 Channel 0 Register Low (T2CH0L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0033	Timer 2 Channel 1 Status and Control Register (T2SC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0034	Timer 2 Channel 1 Register High (T2CH1H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0035	Timer 2 Channel 1 Register Low (T2CH1L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							

= Unimplemented

Figure 11-2. TIM I/O Register Summary (Sheet 3 of 3)

11.5.1 TIM Counter Prescaler

The TIM clock source can be one of the seven prescaler outputs. The prescaler generates seven clock rates from the internal bus clock. The prescaler select bits, PS[2:0], in the TIM status and control register select the TIM clock source.

11.5.2 Input Capture

With the input capture function, the TIM can capture the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the TIM latches the contents of the TIM counter into the TIM channel registers, TCHxH:TCHxL. The polarity of the active edge is programmable. Input captures can generate TIM CPU interrupt requests.

12.4.1 Time Functions

Real time clock functions are provided by the second, minute, and hour counter registers. All three clock counters are able to generate interrupts on every counter increment, providing periodic interrupts for the second (SECF), minute (MINF), and hour (HRF). A CPU interrupt request is generated if the corresponding enable bit (SECIE, MINIE, and HRIE) is also set.

12.4.2 Calendar Functions

Calendar functions are provided by the day, day-of-week, month, and year counter registers. The roll over of the day counter is automatically adjusted for the month and leap years. The setting for the year counter ranges from 1901 to 2099.

The day flag (DAYF) is set on every increment of the day counter. A CPU interrupt request is generated if the day interrupt enable bit (DAYIE) is also set.

12.4.3 Alarm Functions

An alarm function is provided for the minute and hour counters. When minute counter matches the value stored in the alarm minute register, and the hour counter matches the value stored in the alarm hour register, the alarm flag (ALMF) will be set. A CPU interrupt request is generated if the alarm interrupt enable bit (ALMIE) is also set.

12.4.4 Timebase Interrupts

In addition to the second, minute, hour, and day periodic interrupts generated by the clock functions, the divider circuits generates a 2 Hz and a 4 Hz periodic interrupt. These are indicated by the TB1F and TB2F flags. A CPU interrupt request is generated if the corresponding enable bits (TB1IE and TB2IE) is also set.

12.6.2 RTC Control Register 2 (RTCCR2)

The RTC control register 2 (RTCCR2) contains control and clock selection bits for RTC operation.

Address: $0043

	0	0	CHRE	RTCE	0	XTL2	XTL1	XTL0
Read:								
Write:	R	CHRCLR						
Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved

Figure 12-4. RTC Control Register 2 (RTCCR2)

CHRCLR — Chronograph counter clear

Setting this write-only bit resets the chronograph counter. Setting CHRCLR has no effect on any other registers. Counting resumes from $00. CHRCLR is cleared automatically after the chronograph counter is reset and always reads as logic 0. Reset clears the CHRCLR bit.

1 = Chronograph counter cleared
0 = No effect

CHRE — Chronograph Enable

This read/write bit enables the chronograph counter, the value in the chronograph data register increments by 1 in every 1/100 seconds. When the chronograph counter is disabled (CHRE = 0), the value in the chronograph data register is held at the count value. Reset clears the CHRE bit.

1 = Chronograph counter enabled
0 = Chronograph counter disabled

RTCE — Real Time Clock Enable

This read/write bit enables the entire RTC module, allowing all RTC and chronograph operations. Disabling the RTC module does not affect the contents in the RTC registers. Reset clears the RTCE bit.

1 = RTC module enabled
0 = RTC module disabled

13.8 Low-Power Modes

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

13.8.1 Wait Mode

The SCI module remains active after the execution of a WAIT instruction. In wait mode, the SCI module registers are not accessible by the CPU. Any enabled CPU interrupt request from the SCI module can bring the MCU out of wait mode.

If SCI module functions are not required during wait mode, reduce power consumption by disabling the module before executing the WAIT instruction.

Refer to **9.7 Low-Power Modes** for information on exiting wait mode.

13.8.2 Stop Mode

The SCI module is inactive after the execution of a STOP instruction. The STOP instruction does not affect SCI register states. SCI module operation resumes after an external interrupt.

Because the internal clock is inactive during stop mode, entering stop mode during an SCI transmission or reception results in invalid data.

Refer to **9.7 Low-Power Modes** for information on exiting stop mode.

14.14 I/O Registers

Three registers control and monitor SPI operation:

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

14.14.1 SPI Control Register

The SPI control register:

- Enables SPI module interrupt requests
- Configures the SPI module as master or slave
- Selects serial clock polarity and phase
- Configures the SPSCK, MOSI, and MISO pins as open-drain outputs
- Enables the SPI module

Address: $0010

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
Reset:	0	0	1	0	1	0	0	0

= Unimplemented R = Reserved

Figure 14-13. SPI Control Register (SPCR)

SPRIE — SPI Receiver Interrupt Enable Bit

This read/write bit enables CPU interrupt requests generated by the SPRF bit. The SPRF bit is set when a byte transfers from the shift register to the receive data register. Reset clears the SPRIE bit.

1 = SPRF CPU interrupt requests enabled
0 = SPRF CPU interrupt requests disabled

18.4 Functional Description

A logic 0 applied to the external interrupt pin can latch a CPU interrupt request. **Figure 18-1** 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 IRQ latch.
- Software clear — Software can clear the interrupt latch by writing to the acknowledge bit in the interrupt status and control register (INTSCR). Writing a logic 1 to the ACK bit clears the IRQ latch.
- Reset — A reset automatically clears the interrupt latch.

The external interrupt pin is falling-edge-triggered and is software-configurable to be either falling-edge or low-level-triggered. The MODE bit in the INTSCR controls the triggering sensitivity of the $\overline{\text{IRQ}}$ pin.

When the interrupt pin is edge-triggered only, the CPU interrupt request remains set until a vector fetch, software clear, or reset occurs.

When the interrupt pin is both falling-edge and low-level-triggered, the CPU interrupt request remains set until both of the following occur:

- Vector fetch or software clear
- Return of the interrupt pin to logic 1

The vector fetch or software clear may occur before or after the interrupt pin returns to logic 1. As long as the pin is low, the interrupt request remains pending. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

When set, the IMASK bit in the INTSCR mask all external interrupt requests. A latched interrupt request is not presented to the interrupt priority logic unless the IMASK bit is clear.

NOTE: *The interrupt mask (I) in the condition code register (CCR) masks all interrupt requests, including external interrupt requests.*

Section 19. Keyboard Interrupt Module (KBI)

19.1 Contents

19.2 Introduction

The keyboard interrupt module (KBI) provides eight independently maskable external interrupts which are accessible via PTA0–PTA3 and PTD4–PTD7. When a port pin is enabled for keyboard interrupt function, an internal 30kΩ pullup device is also enabled on the pin.

20.4.5 Internal Reset

An internal reset clears the COP prescaler and the COP counter.

20.4.6 Reset Vector Fetch

A reset vector fetch occurs when the vector address appears on the data bus. A reset vector fetch clears the COP prescaler.

20.4.7 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the CONFIG1 register. (See **Figure 20-2** and **Section 5. Configuration Registers (CONFIG)**.)

20.4.8 COPRS (COP Rate Select)

The COPRS signal reflects the state of the COP rate select bit (COPRS) in the CONFIG1 register.

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	0	SSREC	STOP	COPD
Write:								
Reset:	0	0	0	1	0	0	0	0

= Unimplemented

Figure 20-2. Configuration Register 1 (CONFIG1)

COPRS — COP Rate Select

COPRS selects the COP time-out period. Reset clears COPRS.

1 = COP time out period = $2^{13} – 2^4$ ICLK cycles
0 = COP time out period = $2^{18} – 2^4$ ICLK cycles

COPD — COP Disable Bit

COPD disables the COP module.

1 = COP module disabled
0 = COP module enabled

21.4.1 Interrupt LVI Operation

In applications that can operate at V_{DD} levels below the V_{TRIPF} level, software can monitor V_{DD} by polling the LVIOUT bit, or by setting the LVI interrupt enable bit, LVIIE, to enable interrupt requests. In the configuration register 1 (CONFIG1), the LVIPWRD bit must be at logic 0 to enable the LVI module, and the LVIRSTD bit must be at logic 1 to disable LVI resets.

The LVI interrupt flag, LVIIF, is set whenever the LVIOUT bit changes state (toggles). When LVIF is set, a CPU interrupt request is generated if the LVIIE is also set. In the LVI interrupt service subroutine, LVIIF bit can be cleared by writing a logic 1 to the LVI interrupt acknowledge bit, LVIIAK.

21.4.2 Forced Reset Operation

In applications that require V_{DD} to remain above the V_{TRIPF} level, enabling LVI resets allows the LVI module to reset the MCU when V_{DD} falls below the V_{TRIPF} level. In the configuration register 1 (CONFIG1), the LVIPWRD and LVIRSTD bits must be at logic 0 to enable the LVI module and to enable LVI resets.

If LVIIE is set to enable LVI interrupts when LVIRSTD is cleared, LVI reset has a higher priority over LVI interrupt. In this case, when V_{DD} falls below the V_{TRIPF} level, an LVI reset will occur, and the LVIIE bit will be cleared.

21.4.3 Voltage Hysteresis Protection

Once the LVI has triggered (by having V_{DD} fall below V_{TRIPF}), the LVI will maintain a reset condition until V_{DD} rises above the rising trip point voltage, V_{TRIPR}. This prevents a condition in which the MCU is continually entering and exiting reset if V_{DD} is approximately equal to V_{TRIPF}. V_{TRIPR} is greater than V_{TRIPF} by the hysteresis voltage, V_{HYS}.