

Welcome to E-XFL.COM

What is "[Embedded - Microcontrollers](#)"?

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

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	13
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	16-DIP (0.300", 7.62mm)
Supplier Device Package	16-PDIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908qb4cpe

Table of Contents

15.3.3.3	Transmission Format When CPHA = 1	160
15.3.3.4	Transmission Initiation Latency	161
15.3.4	Queuing Transmission Data	163
15.3.5	Resetting the SPI	164
15.3.6	Error Conditions	164
15.3.6.1	Overflow Error	164
15.3.6.2	Mode Fault Error	166
15.4	Interrupts	167
15.5	Low-Power Modes	168
15.5.1	Wait Mode	168
15.5.2	Stop Mode	168
15.6	SPI During Break Interrupts	168
15.7	I/O Signals	169
15.7.1	MISO (Master In/Slave Out)	169
15.7.2	MOSI (Master Out/Slave In)	169
15.7.3	SPSCK (Serial Clock)	169
15.7.4	\overline{SS} (Slave Select)	169
15.8	Registers	170
15.8.1	SPI Control Register	171
15.8.2	SPI Status and Control Register	172
15.8.3	SPI Data Register	174

Chapter 16 Timer Interface Module (TIM)

16.1	Introduction	175
16.2	Features	175
16.3	Functional Description	175
16.3.1	TIM Counter Prescaler	175
16.3.2	Input Capture	176
16.3.3	Output Compare	177
16.3.3.1	Unbuffered Output Compare	178
16.3.3.2	Buffered Output Compare	178
16.3.4	Pulse Width Modulation (PWM)	179
16.3.4.1	Unbuffered PWM Signal Generation	179
16.3.4.2	Buffered PWM Signal Generation	180
16.3.4.3	PWM Initialization	181
16.4	Interrupts	182
16.5	Low-Power Modes	182
16.5.1	Wait Mode	182
16.5.2	Stop Mode	182
16.6	TIM During Break Interrupts	182
16.7	I/O Signals	183
16.7.1	TIM Channel I/O Pins (TCH3:TCH0)	183
16.7.2	TIM Clock Pin (TCLK)	183
16.8	Registers	183
16.8.1	TIM Status and Control Register	183

General Description

- On-chip random-access memory (RAM)
- Enhanced serial communications interface (ESCI) module
- Serial peripheral interface (SPI) module
- 4-channel, 16-bit timer interface (TIM) module
- 10-channel, 10-bit analog-to-digital converter (ADC) with internal bandgap reference channel (ADC10)
- Up to 13 bidirectional input/output (I/O) lines and one input only:
 - Six shared with KBI
 - Ten shared with ADC
 - Four shared with TIM
 - Two shared with ESCI
 - Four shared with SPI
 - One input only shared with IRQ
 - High current sink/source capability on all port pins
 - Selectable pullups on all ports, selectable on an individual bit basis
 - Three-state ability on all port pins
- 6-bit keyboard interrupt with wakeup feature (KBI)
 - Programmable for rising/falling or high/low level detect
- Low-voltage inhibit (LVI) module features:
 - Software selectable trip point
- System protection features:
 - Computer operating properly (COP) watchdog
 - Low-voltage detection with reset
 - Illegal opcode detection with reset
 - Illegal address detection with reset
- External asynchronous interrupt pin with internal pullup ($\overline{\text{IRQ}}$) shared with general-purpose input pin
- Master asynchronous reset pin with internal pullup ($\overline{\text{RST}}$) shared with general-purpose input/output (I/O) pin
- Memory mapped I/O registers
- Power saving stop and wait modes
- MC68HC908QB8, MC68HC908QB4 and MC68HC908QY8 are available in these packages:
 - 16-pin plastic dual in-line package (PDIP)
 - 16-pin small outline integrated circuit (SOIC) package
 - 16-pin thin shrink small outline packages (TSSOP)

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

Memory

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0036	Oscillator Status and Control Register (OSCS) See page 100.	Read:	OSCOPT1	OSCOPT0	ICFS1	ICFS0	ECFS1	ECFS0	ECGON	ECGST
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0037	Reserved									
\$0038	Oscillator Trim Register (OSCTRIM) See page 101.	Read:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
		Write:								
		Reset:	1	0	0	0	0	0	0	0
\$0039 ↓ \$003B	Reserved									
\$003C	ADC10 Status and Control Register (ADSCR) See page 46.	Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
		Write:								
		Reset:	0	0	0	1	1	1	1	1
\$003D	ADC10 Data Register High (ADRH) See page 48.	Read:	0	0	0	0	0	0	AD9	AD8
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$003E	ADC10 Data Register Low (ADRL) See page 48.	Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$003F	ADC10 Clock Register (ADCLK) See page 49.	Read:	ADLPC	ADIV1	ADIV0	ADICLK	MODE1	MODE0	ADLSMP	ACLKEN
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE00	Break Status Register (BSR) See page 195.	Read:	R	R	R	R	R	R	SBSW	R
		Write:							0	
		Reset:							0	
\$FE01	SIM Reset Status Register (SRSR) See page 152.	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
\$FE02	Break Auxiliary Register (BRKAR) See page 195.	Read:	0	0	0	0	0	0	0	BDCOP
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE03	Break Flag Control Register (BFCR) See page 195.	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
\$FE04	Interrupt Status Register 1 (INT1) See page 149.	Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE05	Interrupt Status Register 2 (INT2) See page 149.	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE06	Interrupt Status Register 3 (INT3) See page 149.	Read:	IF22	IF21	IF20	IF19	IF18	IF17	IF16	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0

 = Unimplemented
 R = Reserved
 U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 5)

3.8.2 ADC10 Result High Register (ADRH)

This register holds the MSBs of the result and is updated each time a conversion completes. All other bits read as 0s. Reading ADRH prevents the ADC10 from transferring subsequent conversion results into the result registers until ADRL is read. If ADRL is not read until the after next conversion is completed, then the intermediate conversion result will be lost. In 8-bit mode, this register contains no interlocking with ADRL.

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

= Unimplemented

Figure 3-4. ADC10 Data Register High (ADRH), 8-Bit Mode

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	AD9	AD8
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 3-5. ADC10 Data Register High (ADRH), 10-Bit Mode

3.8.3 ADC10 Result Low Register (ADRL)

This register holds the LSBs of the result. This register is updated each time a conversion completes. Reading ADRH prevents the ADC10 from transferring subsequent conversion results into the result registers until ADRL is read. If ADRL is not read until the after next conversion is completed, then the intermediate conversion result will be lost. In 8-bit mode, there is no interlocking with ADRH.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 3-6. ADC10 Data Register Low (ADRL)

LVIRSTD — LVI Reset Disable Bit

LVIRSTD disables the reset signal from the LVI module.

1 = LVI module resets disabled

0 = LVI module resets enabled

LVIPWRD — LVI Power Disable Bit

LVIPWRD disables the LVI module.

1 = LVI module power disabled

0 = LVI module power enabled

LVITRIP — LVI Trip Point Selection Bit

LVITRIP selects the voltage operating mode of the LVI module. The voltage mode selected for the LVI should match the operating V_{DD} for the LVI's voltage trip points for each of the modes.

1 = LVI operates for a 5-V protection

0 = LVI operates for a 3-V protection

NOTE

The LVITRIP bit is cleared by a power-on reset (POR) only. Other resets will leave this bit unaffected.

SSREC — Short Stop Recovery Bit

SSREC enables the CPU to exit stop mode with a delay of 32 BUSCLKX4 cycles instead of a 4096 BUSCLKX4 cycle delay.

1 = Stop mode recovery after 32 BUSCLKX4 cycles

0 = Stop mode recovery after 4096 BUSCLKX4 cycles

NOTE

Exiting stop mode by an LVI reset will result in the long stop recovery.

When using the LVI during normal operation but disabling during stop mode, the LVI will have an enable time of t_{EN} . The system stabilization time for power-on reset and long stop recovery (both 4096 BUSCLKX4 cycles) gives a delay longer than the LVI enable time for these startup scenarios. There is no period where the MCU is not protected from a low-power condition. However, when using the short stop recovery configuration option, the 32 BUSCLKX4 delay must be greater than the LVI's turn on time to avoid a period in startup where the LVI is not protecting the MCU.

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.

1 = COP module disabled

0 = COP module enabled

7.3.3 Stack Pointer

The stack pointer is a 16-bit register that contains the address of the next location on the stack. During a reset, the stack pointer is preset to \$00FF. The reset stack pointer (RSP) instruction sets the least significant byte to \$FF and does not affect the most significant byte. The stack pointer decrements as data is pushed onto the stack and increments as data is pulled from the stack.

In the stack pointer 8-bit offset and 16-bit offset addressing modes, the stack pointer can function as an index register to access data on the stack. The CPU uses the contents of the stack pointer to determine the conditional address of the operand.

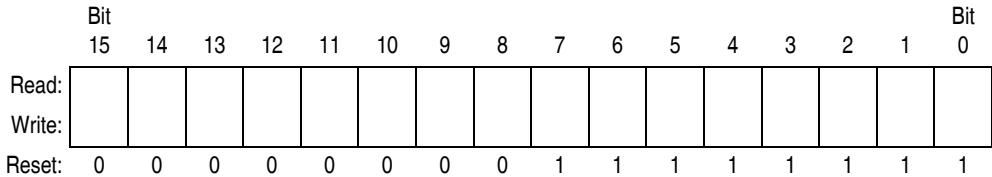


Figure 7-4. Stack Pointer (SP)

NOTE

The location of the stack is arbitrary and may be relocated anywhere in random-access memory (RAM). Moving the SP out of page 0 (\$0000 to \$00FF) frees direct address (page 0) space. For correct operation, the stack pointer must point only to RAM locations.

7.3.4 Program Counter

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched.

Normally, the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

During reset, the program counter is loaded with the reset vector address located at \$FFFE and \$FFFF. The vector address is the address of the first instruction to be executed after exiting the reset state.

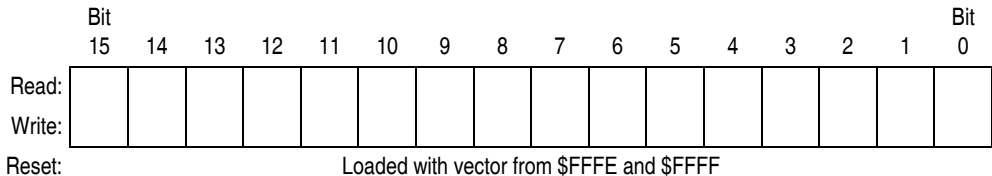


Figure 7-5. Program Counter (PC)

10.4 LVI Interrupts

The LVI module does not generate interrupt requests.

10.5 Low-Power Modes

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

10.5.1 Wait Mode

If enabled, the LVI module remains active in wait mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of wait mode.

10.5.2 Stop Mode

If the LVIPWRD bit in the configuration register is cleared and the LVISTOP bit in the configuration register is set, the LVI module remains active. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of stop mode.

10.6 Registers

The LVI status register (LVISR) contains a status bit that is useful when the LVI is enabled and LVI reset is disabled.

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

= Unimplemented
 R = Reserved

Figure 10-2. LVI Status Register (LVISR)

LVIOUT — LVI Output Bit

This read-only flag becomes set when the V_{DD} voltage falls below the V_{TRIPF} trip voltage and is cleared when V_{DD} voltage rises above V_{TRIPR} . (See [Table 10-1](#)).

Table 10-1. LVIOUT Bit Indication

V_{DD}	LVIOUT
$V_{DD} > V_{TRIPR}$	0
$V_{DD} < V_{TRIPF}$	1
$V_{TRIPF} < V_{DD} < V_{TRIPR}$	Previous value

11.3.5 RC Oscillator

The RC oscillator circuit is designed for use with an external resistor (R_{EXT}) to provide a clock source with a tolerance within 25% of the expected frequency. See [Figure 11-3](#).

The capacitor (C) for the RC oscillator is internal to the MCU. The R_{EXT} value must have a tolerance of 1% or less to minimize its effect on the frequency.

In this configuration, the OSC2 pin can be used as general-purpose input/output (I/O) port pins or other alternative pin function. The OSC2EN bit can be set to enable the OSC2 output function on the pin. Enabling the OSC2 output can affect the external RC oscillator frequency, f_{RCCLK} .

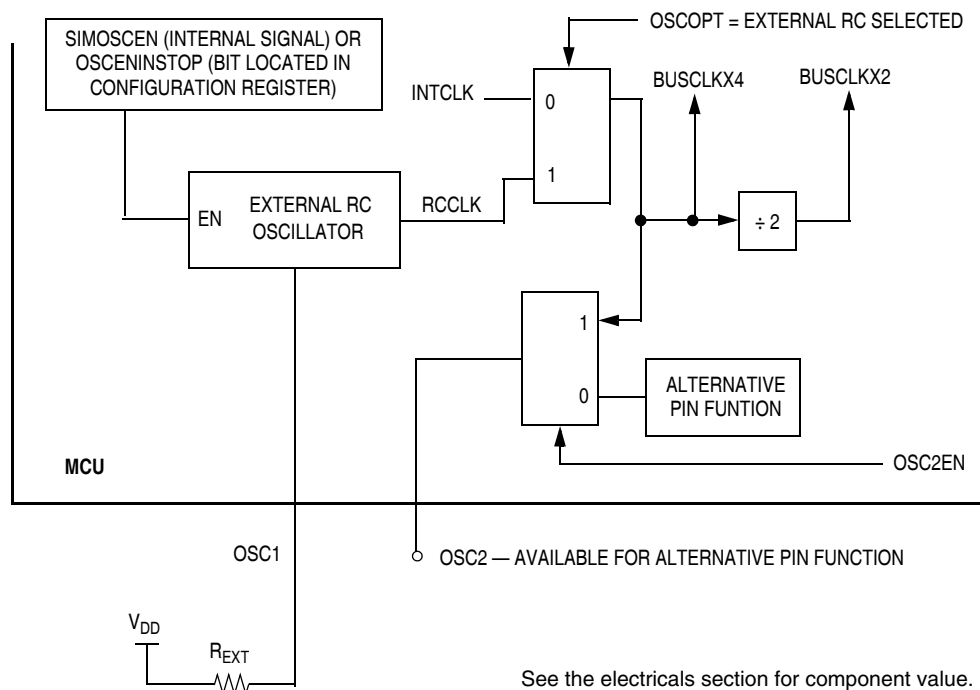


Figure 11-3. RC Oscillator External Connections

11.4 Interrupts

There are no interrupts associated with the OSC module.

11.5 Low-Power Modes

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

11.5.1 Wait Mode

The OSC module remains active in wait mode.

11.5.2 Stop Mode

The OSC module can be configured to remain active in stop mode by setting OSCENINSTOP located in a configuration register.

11.8 Registers

The oscillator module contains two registers:

- Oscillator status and control register (OSCSC)
- Oscillator trim register (OSCTRIM)

11.8.1 Oscillator Status and Control Register

The oscillator status and control register (OSCSC) contains the bits for switching between internal and external clock sources. If the application uses an external crystal, bits in this register are used to select the crystal oscillator amplifier necessary for the desired crystal. While running off the internal clock source, the user can use bits in this register to select the internal clock source frequency.

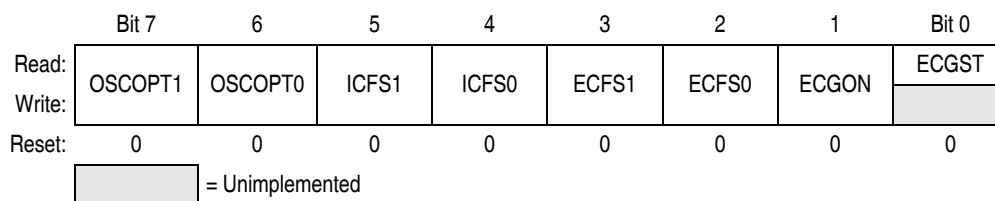


Figure 11-4. Oscillator Status and Control Register (OSCSC)

OSCOPT1:OSCOPT0 — OSC Option Bits

These read/write bits allow the user to change the clock source for the MCU. The default reset condition has the bus clock being derived from the internal oscillator. See [11.3.2.2 Internal to External Clock Switching](#) for information on changing clock sources.

OSCOPT1	OSCOPT0	Oscillator Modes
0	0	Internal oscillator (frequency selected using ICFSx bits)
0	1	External oscillator clock
1	0	External RC
1	1	External crystal (range selected using ECFSx bits)

ICFS1:ICFS0 — Internal Clock Frequency Select Bits

These read/write bits enable the frequency to be increased for applications requiring a faster bus clock when running off the internal oscillator. The WAIT instruction has no effect on the oscillator logic. BUSCLKX2 and BUSCLKX4 continue to drive to the SIM module.

ICFS1	ICFS0	Internal Clock Frequency
0	0	4.0 MHz — default reset condition
0	1	8.0 MHz
1	0	12.8 MHz
1	1	Reserved

14.4.2 Active Resets from Internal Sources

The $\overline{\text{RST}}$ pin is initially setup as a general-purpose input after a POR. Setting the RSTEN bit in the CONFIG2 register enables the pin for the reset function. This section assumes the RSTEN bit is set when describing activity on the $\overline{\text{RST}}$ pin.

NOTE

For POR and LVI resets, the SIM cycles through 4096 BUSCLKX4 cycles. The internal reset signal then follows the sequence from the falling edge of $\overline{\text{RST}}$ shown in Figure 14-4.

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.

All internal reset sources actively pull the $\overline{\text{RST}}$ pin low for 32 BUSCLKX4 cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles (see Figure 14-4). An internal reset can be caused by an illegal address, illegal opcode, COP time out, LVI, or POR (see Figure 14-5).

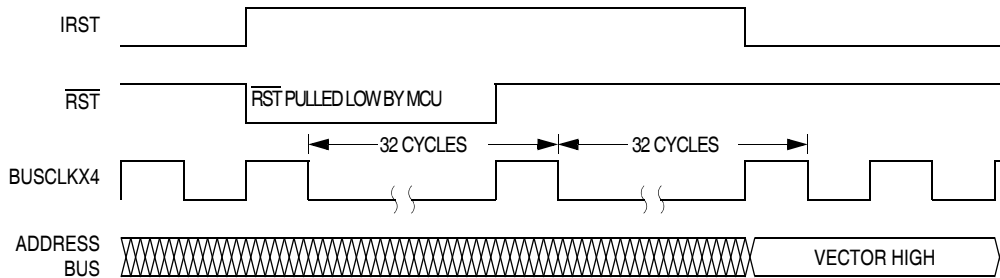


Figure 14-4. Internal Reset Timing

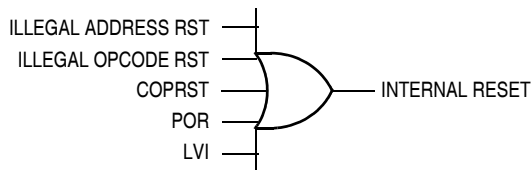


Figure 14-5. Sources of Internal Reset

Table 14-2. Reset Recovery Timing

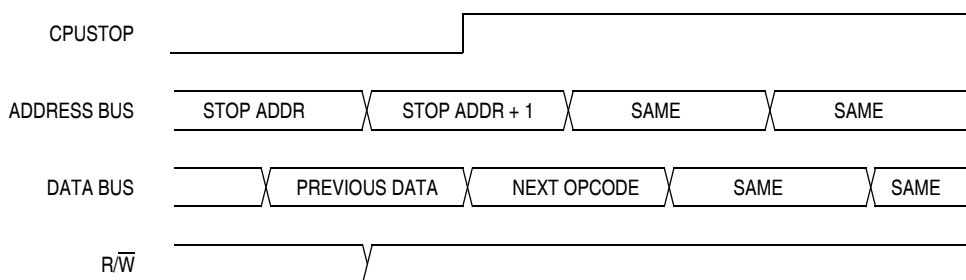
Reset Recovery Type	Actual Number of Cycles
POR/LVI	4163 (4096 + 64 + 3)
All others	67 (64 + 3)

System Integration Module (SIM)

The SIM counter is held in reset from the execution of the STOP instruction until the beginning of stop recovery. It is then used to time the recovery period. [Figure 14-17](#) shows stop mode entry timing and [Figure 14-18](#) shows the stop mode recovery time from interrupt or break

NOTE

To minimize stop current, all pins configured as inputs should be driven to a logic 1 or logic 0.



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

Figure 14-17. Stop Mode Entry Timing

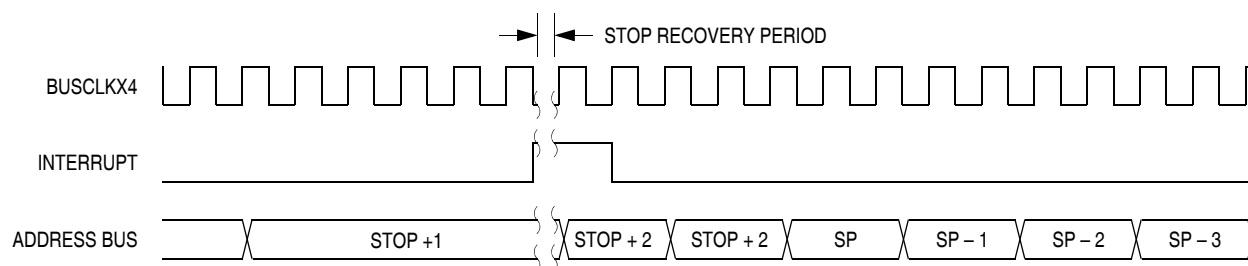


Figure 14-18. Stop Mode Recovery from Interrupt

14.8 SIM Registers

The SIM has two memory mapped registers.

14.8.1 SIM Reset Status Register

The SRSR register contains flags that show the source of the last reset. The status register will automatically clear after reading SRSR. A power-on reset sets the POR bit and clears all other bits in the register. All other reset sources set the individual flag bits but do not clear the register. More than one reset source can be flagged at any time depending on the conditions at the time of the internal or external reset. For example, the POR and LVI bit can both be set if the power supply has a slow rise time.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
Write:								
POR:	1	0	0	0	0	0	0	0

= Unimplemented

Figure 14-19. SIM Reset Status Register (SRSR)

POR — Power-On Reset Bit

- 1 = Last reset caused by POR circuit
- 0 = Read of SRSR

PIN — External Reset Bit

- 1 = Last reset caused by external reset pin (\overline{RST})
- 0 = POR or read of SRSR

COP — Computer Operating Properly Reset Bit

- 1 = Last reset caused by COP counter
- 0 = POR or read of SRSR

ILOP — Illegal Opcode Reset Bit

- 1 = Last reset caused by an illegal opcode
- 0 = POR or read of SRSR

ILAD — Illegal Address Reset Bit (illegal attempt to fetch an opcode from an unimplemented address)

- 1 = Last reset caused by an opcode fetch from an illegal address
- 0 = POR or read of SRSR

MODRST — Monitor Mode Entry Module Reset Bit

- 1 = Last reset caused by monitor mode entry when vector locations \$FFFE and \$FFFF are \$FF after POR while $IRQ \neq V_{TST}$
- 0 = POR or read of SRSR

LVI — Low Voltage Inhibit Reset bit

- 1 = Last reset caused by LVI circuit
- 0 = POR or read of SRSR

14.8.2 Break Flag Control Register

The break control register (BFCR) contains a bit that enables software to clear status bits while the MCU is in a break state.

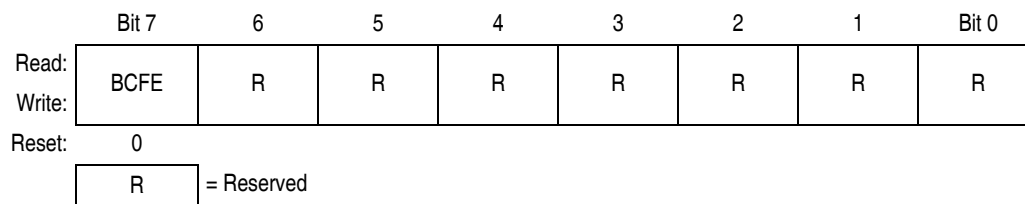


Figure 14-20. Break Flag Control Register (BFCR)

BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

- 1 = Status bits clearable during break
- 0 = Status bits not clearable during break

15.3 Functional Description

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.

The following paragraphs describe the operation of the SPI module.

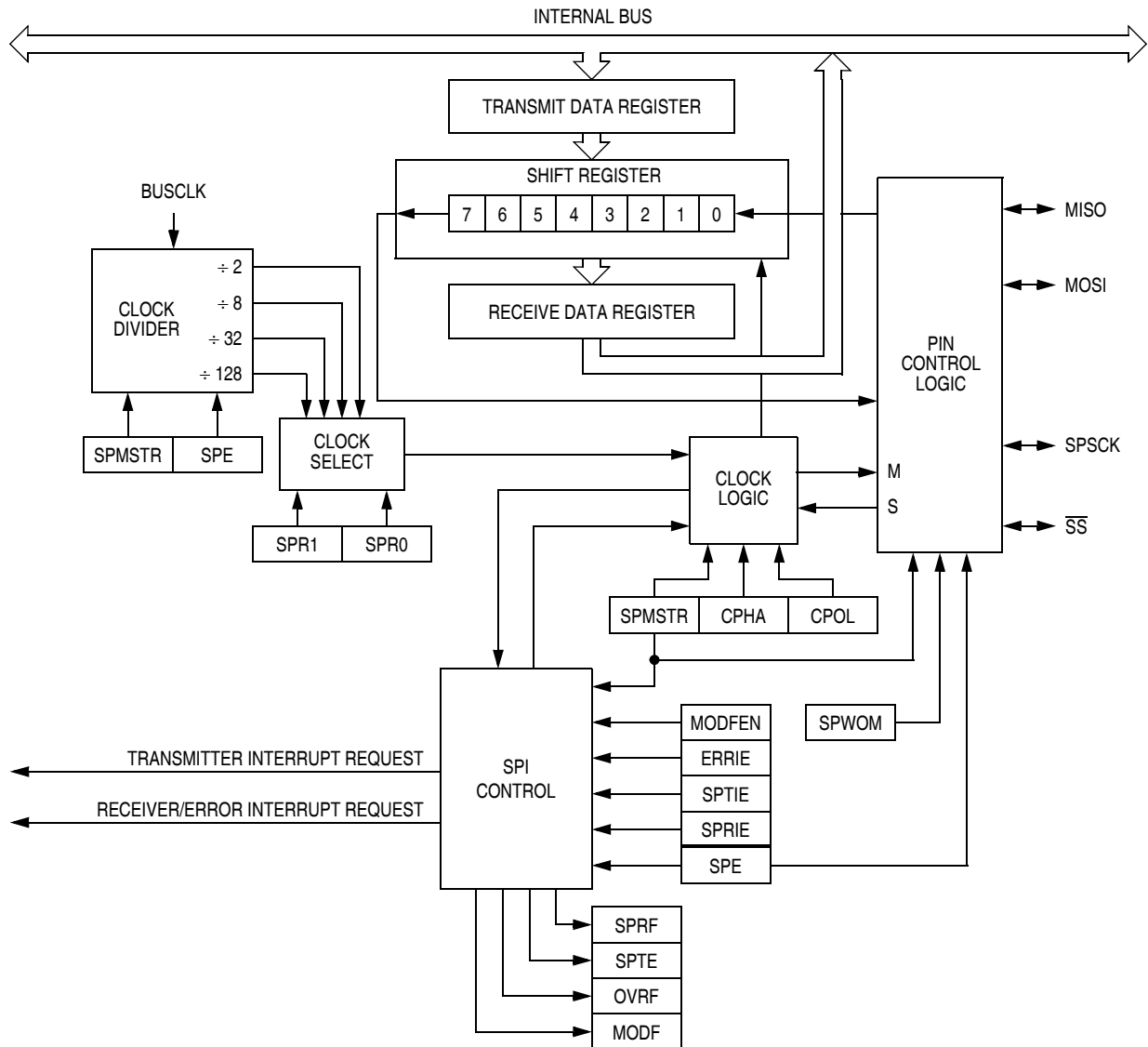


Figure 15-2. SPI Module Block Diagram

remain low between transmissions. This format may be preferable in systems having only one master and only one slave driving the MISO data line.

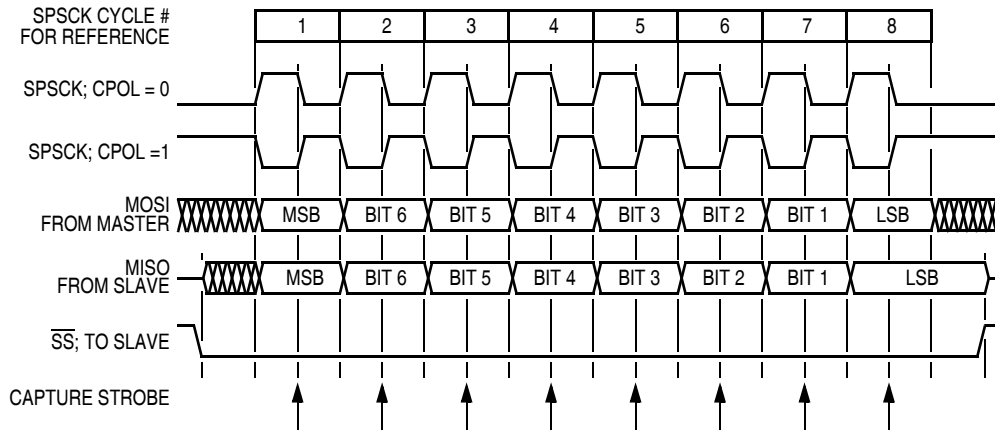


Figure 15-6. Transmission Format (CPHA = 1)

When CPHA = 1 for a slave, the first edge of the SPSCCK indicates the beginning of the transmission. This causes the SPI to leave its idle state and begin driving the MISO pin with the MSB of its data. After the transmission begins, no new data is allowed into the shift register from the transmit data register. Therefore, the SPI data register of the slave must be loaded with transmit data before the first edge of SPSCCK. Any data written after the first edge is stored in the transmit data register and transferred to the shift register after the current transmission.

15.3.3.4 Transmission Initiation Latency

When the SPI is configured as a master (SPMSTR = 1), writing to the SPDR starts a transmission. CPHA has no effect on the delay to the start of the transmission, but it does affect the initial state of the SPSCCK signal. When CPHA = 0, the SPSCCK signal remains inactive for the first half of the first SPSCCK cycle. When CPHA = 1, the first SPSCCK cycle begins with an edge on the SPSCCK 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 15-7.) The internal SPI clock in the master is a free-running derivative of the internal MCU clock. To conserve power, it is enabled only when both the SPE and SPMSTR bits are set. Because the SPI clock is free-running, it is uncertain where the write to the SPDR occurs relative to the slower SPSCCK. This uncertainty causes the variation in the initiation delay shown in Figure 15-7. This delay is no longer than a single SPI bit time. That is, the maximum delay is two MCU bus cycles for DIV2, eight MCU bus cycles for DIV8, 32 MCU bus cycles for DIV32, and 128 MCU bus cycles for DIV128.

15.8.1 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 SPSCCK, MOSI, and MISO pins as open-drain outputs
- Enables the SPI module

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

R = Reserved

Figure 15-13. SPI Control Register (SPCR)

SPRIE — SPI Receiver Interrupt Enable Bit

This read/write bit enables 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.

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

SPMSTR — SPI Master Bit

This read/write bit selects master mode operation or slave mode operation.

- 1 = Master mode
- 0 = Slave mode

CPOL — Clock Polarity Bit

This read/write bit determines the logic state of the SPSCCK pin between transmissions. (See [Figure 15-4](#) and [Figure 15-6](#).) To transmit data between SPI modules, the SPI modules must have identical CPOL values.

CPHA — Clock Phase Bit

This read/write bit controls the timing relationship between the serial clock and SPI data. (See [Figure 15-4](#) and [Figure 15-6](#).) To transmit data between SPI modules, the SPI modules must have identical CPHA values. When CPHA = 0, the \overline{SS} pin of the slave SPI module must be high between bytes. (See [Figure 15-12](#).)

SPWOM — SPI Wired-OR Mode Bit

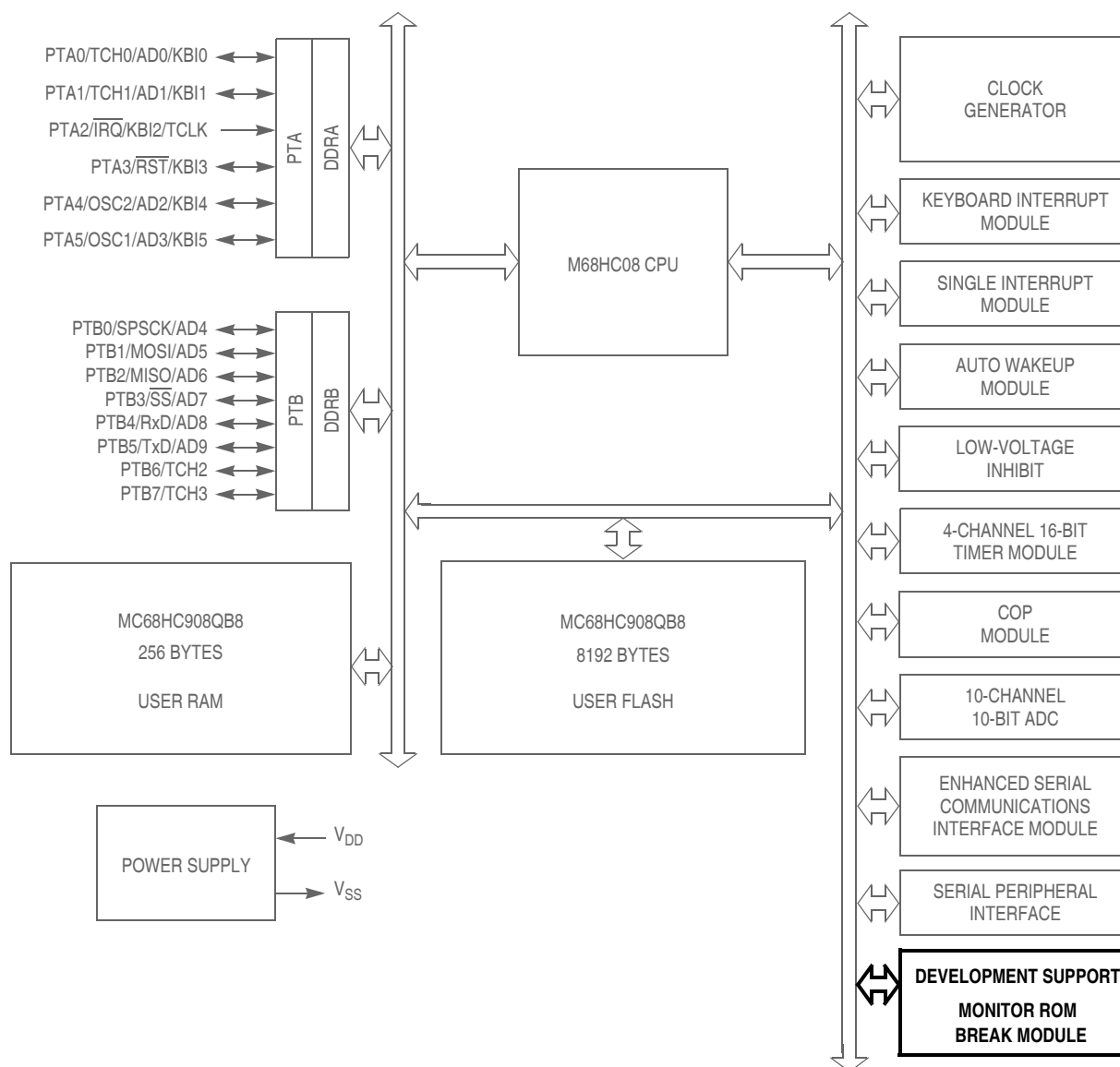
This read/write bit configures pins SPSCCK, MOSI, and MISO so that these pins become open-drain outputs.

- 1 = Wired-OR SPSCCK, MOSI, and MISO pins
- 0 = Normal push-pull SPSCCK, MOSI, and MISO pins

SPE — SPI Enable

This read/write bit enables the SPI module. Clearing SPE causes a partial reset of the SPI. (See [15.3.5 Resetting the SPI](#).)

- 1 = SPI module enabled
- 0 = SPI module disabled



\overline{RST} , \overline{IRQ} : Pins have internal pull up device
 All port pins have programmable pull up device
 PTA[0:5]: Higher current sink and source capability

Figure 17-1. Block Diagram Highlighting BRK and MON Blocks



How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 that may be provided in Freescale Semiconductor 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 Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor 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 Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor 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 Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2005–2010. All rights reserved.