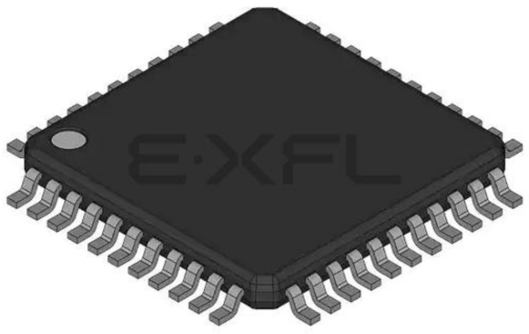
Freescale Semiconductor - MC68HC908AP8CB Datasheet



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	8MHz
Connectivity	I²C, IRSCI, SCI, SPI
Peripherals	LED, LVD, POR, PWM
Number of I/O	32
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	42-SDIP (0.600", 15.24mm)
Supplier Device Package	42-PDIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68hc908ap8cb

6. Clear the ERASE bit.
7. Wait for a time, t_{nvh} (5 µs).
8. Clear the HVEN bit.
9. After time, t_{rcv} (1 µs), the memory can be accessed in read mode again.

NOTE

Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps.

2.5.4 FLASH Mass Erase Operation

Use the following procedure to erase the entire FLASH memory:

1. Set both the ERASE bit and the MASS bit in the FLASH control register.
2. Write any data to any FLASH location within the FLASH memory address range.
3. Wait for a time, t_{nvs} (5 µs).
4. Set the HVEN bit.
5. Wait for a time t_{me} (200 ms). (See ***NOTE*** below.)
6. Clear the ERASE bit.
7. Wait for a time, t_{nvh1} (100 µs).
8. Clear the HVEN bit.
9. After time, t_{rcv} (1 µs), the memory can be accessed in read mode again.

NOTE

Due to the relatively long mass erase time, user should take care in the code to prevent a COP reset from happening while the HVEN bit is set.

Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps.

2.5.5 FLASH Program Operation

Programming of the FLASH memory is done on a row basis. A row consists of 64 consecutive bytes starting from addresses $XX00, $XX40, $XX80 or $XXC0. Use the following procedure to program a row of FLASH memory. (Figure 2-4 shows a flowchart of the programming algorithm.)

1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
2. Write any data to any FLASH location within the address range of the row to be programmed.
3. Wait for a time, t_{nvs} (5 µs).
4. Set the HVEN bit.
5. Wait for a time, t_{pgs} (10 µs).
6. Write data to the FLASH location to be programmed.
7. Wait for time, t_{prog} (20 µs to 40 µs).
8. Repeat steps 6 and 7 until all bytes within the row are programmed.
9. Clear the PGM bit.

10. Wait for time, t_{nvh} (5 μs).
11. Clear the HVEN bit.
12. After time, t_{rcv} (1 μs), the memory can be accessed in read mode again.

This program sequence is repeated throughout the memory until all data is programmed.

NOTE

The time between each FLASH address change (step 6 to step 6), or the time between the last FLASH addressed programmed to clearing the PGM bit (step 6 to step 9), must not exceed the maximum programming time, t_{prog} max.

NOTE

Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order shown, other unrelated operations may occur between the steps.

2.5.6 FLASH Protection

Due to the ability of the on-board charge pump to erase and program the FLASH memory in the target application, provision is made to protect pages of memory from unintentional erase or program operations due to system malfunction. This protection is done by use of a FLASH block protect register (FLBPR). The FLBPR determines the range of the FLASH memory which is to be protected. The range of the protected area starts from a location defined by FLBPR and ends to the bottom of the FLASH memory ($FFFF). When the memory is protected, the HVEN bit cannot be set in either erase or program operations.

NOTE

The mask option register ($FFCF) and the 48 bytes of user interrupt vectors ($FFD0–$FFFF) are always protected, regardless of the value in the FLASH block protect register. A mass erase is required to erase these locations.

3.4 Configuration Register 2 (CONFIG2)

Address: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	STOP_ ICLKDIS	STOP_ RCLKEN	STOP_ XCLKEN	OSCCLK1	OSCCLK0	0	0	SCIBD- SRC
Write:						▨	▨	
Reset:	0	0	0	0	0	0	0	0

Figure 3-3. Configuration Register 2 (CONFIG2)

STOP_ICLKDIS — Internal Oscillator Stop Mode Disable

STOP_ICLKDIS disables the internal oscillator during stop mode. Setting the STOP_ICLKDIS bit disables the oscillator during stop mode. (See Chapter 5 Oscillator (OSC).)

Reset clears this bit.

1 = Internal oscillator disabled during stop mode
0 = Internal oscillator enabled to operate during stop mode

STOP_RCLKEN — RC Oscillator Stop Mode Enable Bit

STOP_RCLKEN enables the RC oscillator to continue operating during stop mode. Setting the STOP_RCLKEN bit allows the oscillator to operate continuously even during stop mode. This is useful for driving the timebase module to allow it to generate periodic wake up while in stop mode. (See Chapter 5 Oscillator (OSC).)

Reset clears this bit.

1 = RC oscillator enabled to operate during stop mode
0 = RC oscillator disabled during stop mode

STOP_XCLKEN — X-tal Oscillator Stop Mode Enable Bit

STOP_XCLKEN enables the crystal (x-tal) oscillator to continue operating during stop mode. Setting the STOP_XCLKEN bit allows the x-tal oscillator to operate continuously even during stop mode. This is useful for driving the timebase module to allow it to generate periodic wake up while in stop mode. (See Chapter 5 Oscillator (OSC).) Reset clears this bit.

1 = X-tal oscillator enabled to operate during stop mode
0 = X-tal oscillator disabled during stop mode

OSCCLK1, OSCCLK0 — Oscillator Output Control Bits

OSCCLK1 and OSCCLK0 select which oscillator output to be driven out as OSCCLK to the timebase module (TBM). Reset clears these two bits.

OSCCLK1	OSCCLK0	Timebase Clock Source
0	0	Internal oscillator (ICLK)
0	1	RC oscillator (RCCLK)
1	0	X-tal oscillator (XTAL)
1	1	Not used

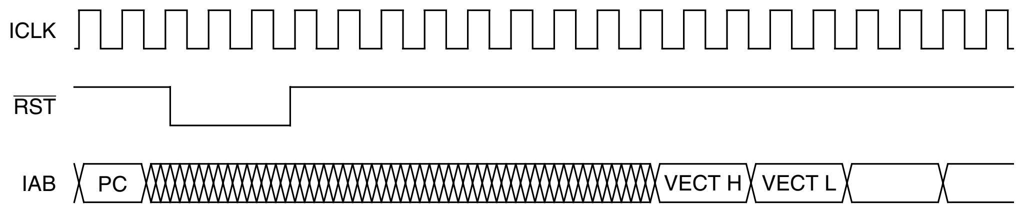


Figure 7-4. External Reset Timing

7.3.2 Active Resets from Internal Sources

All internal reset sources actively pull the $\overline{\text{RST}}$ pin low for 32 ICLK cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles (see Figure 7-5). An internal reset can be caused by an illegal address, illegal opcode, COP timeout, LVI, or POR (see Figure 7-6).

NOTE

For LVI or POR resets, the SIM cycles through 4096 + 32 ICLK 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 7-5.

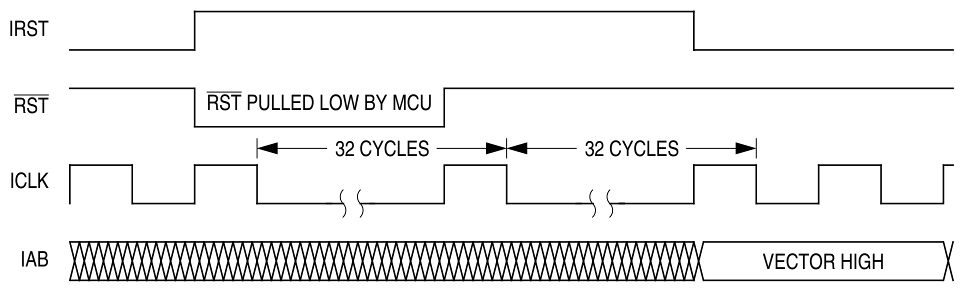


Figure 7-5. Internal Reset Timing

The COP reset is asynchronous to the bus clock.

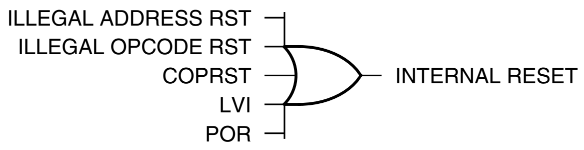


Figure 7-6. Sources of Internal Reset

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

7.3.2.1 Power-On Reset

When power is first applied to the MCU, the power-on reset module (POR) generates a pulse to indicate that power-on has occurred. The external reset pin ($\overline{\text{RST}}$) is held low while the SIM counter counts out 4096 + 32 ICLK cycles. Thirty-two ICLK cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur.

8.5.4 MON_PRGRNGE

In monitor mode, MON_PRGRNGE is used to program a range of FLASH locations with data loaded into the data array.

Table 8-14. MON_PRGRNGE Routine

Routine Name	MON_PRGRNGE
Routine Description	Program a range of locations, in monitor mode
Calling Address	$FF24
Stack Used	17 bytes
Data Block Format	Bus speed Data size Starting address (high byte) Starting address (low byte) Data 1 : Data N

The MON_PRGRNGE routine is designed to be used in monitor mode. It performs the same function as the PRGRNGE routine (see 8.5.1 PRGRNGE), except that MON_PRGRNGE returns to the main program via an SWI instruction. After a MON_PRGRNGE call, the SWI instruction will return the control back to the monitor code.

8.5.5 MON_ERARNGE

In monitor mode, ERARNGE is used to erase a range of locations in FLASH.

Table 8-15. MON_ERARNGE Routine

Routine Name	MON_ERARNGE
Routine Description	Erase a page or the entire array, in monitor mode
Calling Address	$FF28
Stack Used	11 bytes
Data Block Format	Bus speed Data size Starting address (high byte) Starting address (low byte)

The MON_ERARNGE routine is designed to be used in monitor mode. It performs the same function as the ERARNGE routine (see 8.5.2 ERARNGE), except that MON_ERARNGE returns to the main program via an SWI instruction. After a MON_ERARNGE call, the SWI instruction will return the control back to the monitor code.

9.4 Functional Description

Figure 9-1 shows the structure of the TIM. The central component of the TIM is the 16-bit TIM counter that can operate as a free-running counter or a modulo up-counter. The TIM counter provides the timing reference for the input capture and output compare functions. The TIM counter modulo registers, TMODH:TMODL, control the modulo value of the TIM counter. Software can read the TIM counter value at any time without affecting the counting sequence.

The two TIM channels (per timer) are programmable independently as input capture or output compare channels.

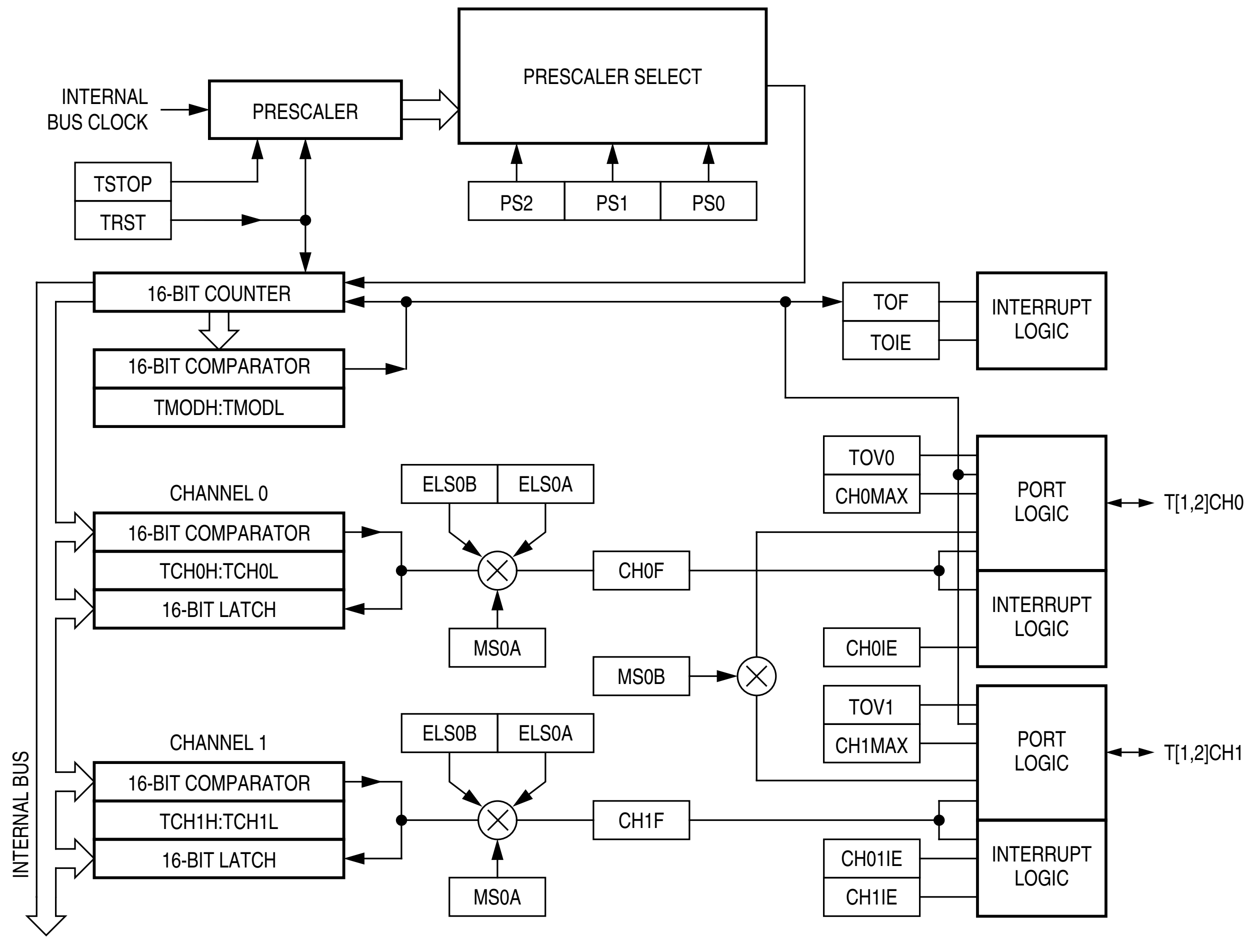


Figure 9-1. TIM Block Diagram

Figure 9-2 summarizes the timer registers.

NOTE

References to either timer 1 or timer 2 may be made in the following text by omitting the timer number. For example, TSC may generically refer to both T1SC and T2SC.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
	TIM1 Status and Control	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
$0020	Register	Write:	0			TRST	▨			
	(T1SC)	Reset:	0	0	1	0	0	0	0	0
	TIM1 Counter Register	Read:	Bit 15	14	13	12	11	10	9	Bit 8
$0021	High	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	(T1CNTH)	Reset:	0	0	0	0	0	0	0	0
	TIM1 Counter Register	Read:	Bit 7	6	5	4	3	2	1	Bit 0
$0022	Low	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	(T1CNTL)	Reset:	0	0	0	0	0	0	0	0
	TIM Counter Modulo	Read:	Bit 15	14	13	12	11	10	9	Bit 8
$0023	Register High	Write:								
	(TMODH)	Reset:	1	1	1	1	1	1	1	1
	TIM1 Counter Modulo	Read:	Bit 7	6	5	4	3	2	1	Bit 0
$0024	Register Low	Write:								
	(T1MODL)	Reset:	1	1	1	1	1	1	1	1
	TIM1 Channel 0 Status	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
$0025	and Control Register	Write:	0							
	(T1SC0)	Reset:	0	0	0	0	0	0	0	0
	TIM1 Channel 0	Read:	Bit 15	14	13	12	11	10	9	Bit 8
$0026	Register High	Write:								
	(T1CH0H)	Reset:	Indeterminate after reset							
	TIM1 Channel 0	Read:	Bit 7	6	5	4	3	2	1	Bit 0
$0027	Register Low	Write:								
	(T1CH0L)	Reset:	Indeterminate after reset							
	TIM1 Channel 1 Status	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
$0028	and Control Register	Write:	0		▨					
	(T1SC1)	Reset:	0	0	0	0	0	0	0	0
	TIM1 Channel 1	Read:	Bit 15	14	13	12	11	10	9	Bit 8
$0029	Register High	Write:								
	(T1CH1H)	Reset:	Indeterminate after reset							
	TIM1 Channel 1	Read:	Bit 7	6	5	4	3	2	1	Bit 0
$002A	Register Low	Write:								
	(T1CH1L)	Reset:	Indeterminate after reset							
	TIM2 Status and Control	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
$002B	Register	Write:	0			TRST	▨			
	(T2SC)	Reset:	0	0	1	0	0	0	0	0
	TIM2 Counter Register	Read:	Bit 15	14	13	12	11	10	9	Bit 8
$002C	High	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	(T2CNTH)	Reset:	0	0	0	0	0	0	0	0
	TIM2 Counter Register	Read:	Bit 7	6	5	4	3	2	1	Bit 0
$002D	Low	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	(T2CNTL)	Reset:	0	0	0	0	0	0	0	0
	TIM2 Counter Modulo	Read:	Bit 15	14	13	12	11	10	9	Bit 8
$002E	Register High	Write:								
	(T2MODH)	Reset:	1	1	1	1	1	1	1	1

▨ = Unimplemented

Figure 9-2. TIM I/O Register Summary (Sheet 1 of 2)

11.8.1 SCI Control Register 1

SCI control register 1:

- Enables loop mode operation
- Enables the SCI
- Controls output polarity
- Controls character length
- Controls SCI wakeup method
- Controls idle character detection
- Enables parity function
- Controls parity type

Address: $0013

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
Reset:	0	0	0	0	0	0	0	0

Figure 11-9. SCI Control Register 1 (SCC1)

LOOPS — Loop Mode Select Bit

This read/write bit enables loop mode operation. In loop mode the RxD pin is disconnected from the SCI, and the transmitter output goes into the receiver input. Both the transmitter and the receiver must be enabled to use loop mode. Reset clears the LOOPS bit.

1 = Loop mode enabled
0 = Normal operation enabled

ENSCI — Enable SCI Bit

This read/write bit enables the SCI and the SCI baud rate generator. Clearing ENSCI sets the SCTE and TC bits in SCI status register 1 and disables transmitter interrupts. Reset clears the ENSCI bit.

1 = SCI enabled
0 = SCI disabled

TXINV — Transmit Inversion Bit

This read/write bit reverses the polarity of transmitted data. Reset clears the TXINV bit.

1 = Transmitter output inverted
0 = Transmitter output not inverted

NOTE

Setting the TXINV bit inverts all transmitted values, including idle, break, start, and stop bits.

M — Mode (Character Length) Bit

This read/write bit determines whether SCI characters are eight or nine bits long. (See Table 11-5.) The ninth bit can serve as an extra stop bit, as a receiver wakeup signal, or as a parity bit. Reset clears the M bit.

1 = 9-bit SCI characters
0 = 8-bit SCI characters

TE — Transmitter Enable Bit

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

1 = Transmitter enabled
0 = Transmitter disabled

NOTE

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

RE — Receiver Enable Bit

Setting this read/write bit enables the receiver. Clearing the RE bit disables the receiver but does not affect receiver interrupt flag bits. Reset clears the RE bit.

1 = Receiver enabled
0 = Receiver disabled

NOTE

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

RWU — Receiver Wakeup Bit

This read/write bit puts the receiver in a standby state during which receiver interrupts are disabled. The WAKE bit in SCC1 determines whether an idle input or an address mark brings the receiver out of the standby state and clears the RWU bit. Reset clears the RWU bit.

1 = Standby state
0 = Normal operation

SBK — Send Break Bit

Setting and then clearing this read/write bit transmits a break character followed by a logic 1. The logic 1 after the break character guarantees recognition of a valid start bit. If SBK remains set, the transmitter continuously transmits break characters with no logic 1s between them. Reset clears the SBK bit.

1 = Transmit break characters
0 = No break characters being transmitted

NOTE

Do not toggle the SBK bit immediately after setting the SCTE bit. Toggling SBK before the preamble begins causes the SCI to send a break character instead of a preamble.

12.2 Pin Name Conventions

The generic names of the IRSCI I/O pins are:

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

IRSCI I/O (input/output) lines are implemented by sharing parallel I/O port pins. The full name of an IRSCI input or output reflects the name of the shared port pin. Table 12-1 shows the full names and the generic names of the IRSCI I/O pins. The generic pin names appear in the text of this section.

Table 12-1. Pin Name Conventions

Generic Pin Names:	RxD	TxD
Full Pin Names:	PTC7/SCRxD	PTC6/SCTxD

NOTE
When the IRSCI is enabled, the SCTxD pin is an open-drain output and requires a pullup resistor to be connected for proper SCI operation.

Addr.	**Register Name**		**Bit 7**	**6**	**5**	**4**	**3**	**2**	**1**	**Bit 0**
$0040	IRSCI Control Register 1 (IRSCC1)	Read:	LOOPS	ENSCI	0	M	WAKE	ILTY	PEN	PTY
		Write:			▨					
		Reset:	0	0	0	0	0	0	0	0
$0041	IRSCI Control Register 2 (IRSCC2)	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0042	IRSCI Control Register 3 (IRSCC3)	Read:	R8	T8	DMARE	DMATE	ORIE	NEIE	FEIE	PEIE
		Write:	▨							
		Reset:	U	U	0	0	0	0	0	0
$0043	IRSCI Status Register 1 (IRSCS1)	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	1	1	0	0	0	0	0	0
$0044	IRSCI Status Register 2 (IRSCS2)	Read:	▨	▨	▨	▨	▨	▨	BKF	RPF
		Write:							▨	▨
		Reset:	0	0	0	0	0	0	0	0
$0045	IRSCI Data Register (IRSCDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							
$0046	IRSCI Baud Rate Register (IRSCBR)	Read:	CKS	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:		▨						
		Reset:	0	0	0	0	0	0	0	0
$0047	IRSCI Infrared Control Register (IRSCIRCR)	Read:	R	0	0	0	R	TNP1	TNP0	IREN
		Write:		▨	▨	▨				
		Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented R = Reserved U = Unaffected

Figure 12-1. IRSCI I/O Registers Summary

- SCI transmitter empty (SCTE) — The SCTE bit in IRSCS1 indicates that the IRSCDR has transferred a character to the transmit shift register. SCTE can generate a transmitter CPU interrupt request. Setting the SCI transmit interrupt enable bit, SCTIE, in IRSCC2 enables the SCTE bit to generate transmitter CPU interrupt requests.
- Transmission complete (TC) — The TC bit in IRSCS1 indicates that the transmit shift register and the IRSCDR are empty and that no break or idle character has been generated. The transmission complete interrupt enable bit, TCIE, in IRSCC2 enables the TC bit to generate transmitter CPU interrupt requests.

12.5.3 Receiver

Figure 12-8 shows the structure of the SCI receiver.

Reading the SPI status and control register with SPRF set and then reading the receive data register clears SPRF. The clearing mechanism for the SPTE flag is always just a write to the transmit data register.

The SPI transmitter interrupt enable bit (SPTIE) enables the SPTE flag to generate transmitter CPU interrupt requests, provided that the SPI is enabled (SPE = 1).

The SPI receiver interrupt enable bit (SPRIE) enables the SPRF bit to generate receiver CPU interrupt requests, regardless of the state of the SPE bit. (See Figure 13-11.)

The error interrupt enable bit (ERRIE) enables both the MODF and OVRF bits to generate a receiver/error CPU interrupt request.

The mode fault enable bit (MODFEN) can prevent the MODF flag from being set so that only the OVRF bit is enabled by the ERRIE bit to generate receiver/error CPU interrupt requests.

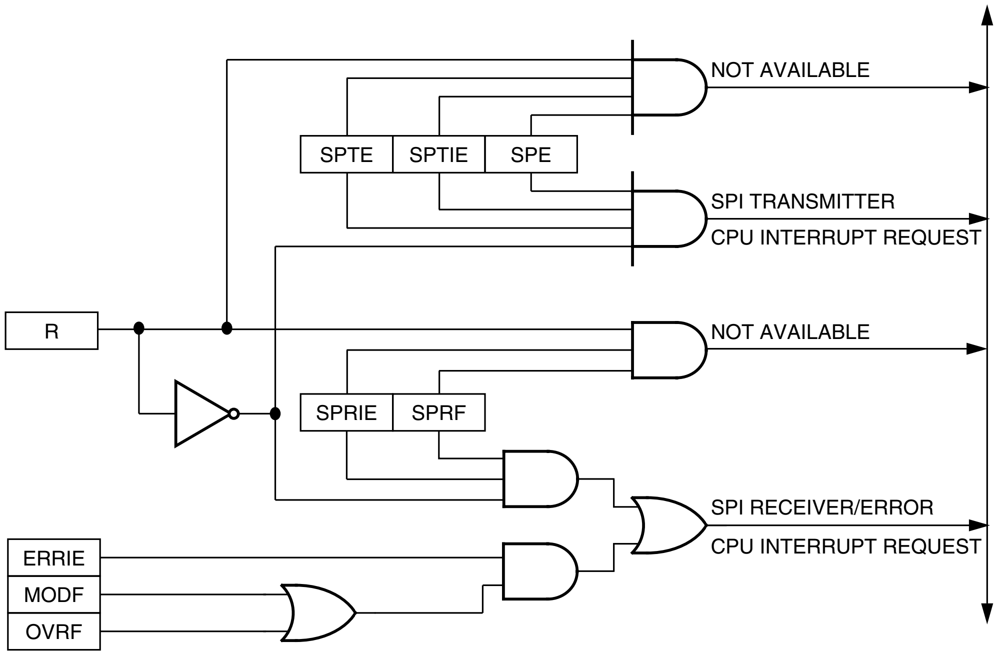


Figure 13-11. SPI Interrupt Request Generation

The following sources in the SPI status and control register can generate CPU interrupt requests:

- SPI receiver full bit (SPRF) — The SPRF bit becomes set every time a byte transfers from the shift register to the receive data register. If the SPI receiver interrupt enable bit, SPRIE, is also set, SPRF generates an SPI receiver/error CPU interrupt request.
- SPI transmitter empty (SPTE) — The SPTE bit becomes set every time a byte transfers from the transmit data register to the shift register. If the SPI transmit interrupt enable bit, SPTIE, is also set, SPTE generates an SPTE CPU interrupt request.

13.9 Resetting the SPI

Any system reset completely resets the SPI. Partial resets occur whenever the SPI enable bit (SPE) is low. Whenever SPE is low, the following occurs:

- The SPTE flag is set.
- Any transmission currently in progress is aborted.

SPWOM — SPI Wired-OR Mode Bit

This read/write bit disables the pullup devices on pins SPSCK, MOSI, and MISO so that those pins become open-drain outputs.

1 = Wired-OR SPSCK, MOSI, and MISO pins
0 = Normal push-pull SPSCK, 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 13.9 Resetting the SPI.) Reset clears the SPE bit.

1 = SPI module enabled
0 = SPI module disabled

SPTIE— SPI Transmit Interrupt Enable

This read/write bit enables CPU interrupt requests generated by the SPTE bit. SPTE is set when a byte transfers from the transmit data register to the shift register. Reset clears the SPTIE bit.

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

13.13.2 SPI Status and Control Register

The SPI status and control register contains flags to signal these conditions:

- Receive data register full
- Failure to clear SPRF bit before next byte is received (overflow error)
- Inconsistent logic level on $\overline{\text{SS}}$ pin (mode fault error)
- Transmit data register empty

The SPI status and control register also contains bits that perform these functions:

- Enable error interrupts
- Enable mode fault error detection
- Select master SPI baud rate

Address $0011

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SPRF	ERRIE	OVRF	MODF	SPTE	MODFEN	SPR1	SPR0
Write:	▨		▨	▨	▨			
Reset:	0	0	0	0	1	0	0	0

▨ = Unimplemented

Figure 13-14. SPI Status and Control Register (SPSCR)

SPRF — SPI Receiver Full Bit

This clearable, read-only flag is set each time a byte transfers from the shift register to the receive data register. SPRF generates a CPU interrupt request if the SPRIE bit in the SPI control register is set also. During an SPRF CPU interrupt, the CPU clears SPRF by reading the SPI status and control register with SPRF set and then reading the SPI data register. Reset clears the SPRF bit.

1 = Receive data register full
0 = Receive data register not full

14.8.6 Process Call

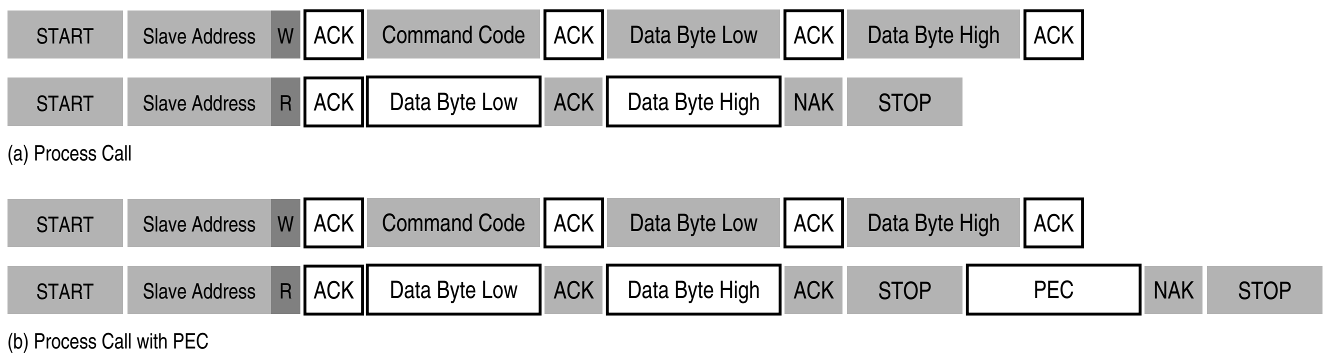


Figure 14-18. Process Call

14.8.7 Block Read/Write

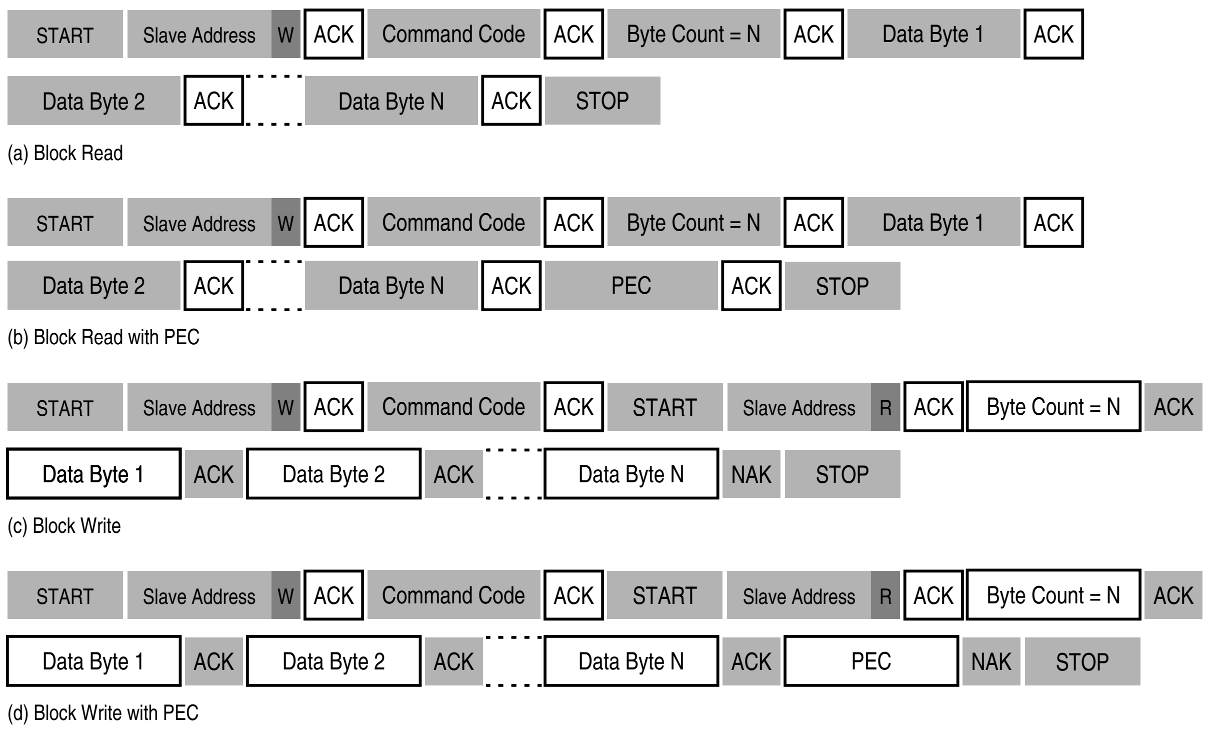


Figure 14-19. Block Read/Write

Chapter 15
Analog-to-Digital Converter (ADC)

15.1 Introduction

This section describes the analog-to-digital converter (ADC). The ADC is a 8-channel 10-bit linear successive approximation ADC.

15.2 Features

Features of the ADC module include:

- Eight channels with multiplexed input
- High impedance buffered input
- Linear successive approximation with monotonicity
- 10-bit resolution
- Single or continuous conversion
- Auto-scan conversion on four channels
- Conversion complete flag or conversion complete interrupt
- Selectable ADC clock
- Conversion result justification
  - 8-bit truncated mode
  - Right justified mode
  - Left justified mode
  - Left justified sign mode

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0057	ADC Status and Control Register (ADSCR)	Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
		Write:	▨							
		Reset:	0	0	0	1	1	1	1	1
$0058	ADC Clock Control Register (ADICLK)	Read:	ADIV2	ADIV1	ADIV0	ADICLK	MODE1	MODE0	0	0
		Write:							▨	R
		Reset:	0	0	0	0	0	1	0	0
$0059	ADC Data Register High 0 (ADRH0)	Read:	ADx	ADx	ADx	ADx	ADx	ADx	ADx	ADx
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005A	ADC Data Register Low 0 (ADRL0)	Read:	ADx	ADx	ADx	ADx	ADx	ADx	ADx	ADx
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005B	ADC Data Register Low 1 (ADRL1)	Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0

Figure 15-1. ADC I/O Register Summary

16.2.3 Port-A LED Control Register (LEDA)

The port-A LED control register (LEDA) controls the direct LED drive capability on PTA7–PTA0 pins. Each bit is individually configurable and requires that the data direction register, DDRA, bit be configured as an output.

Address: $000C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LEDA7	LEDA6	LEDA5	LEDA4	LEDA3	LEDA2	LEDA1	LEDA0
Reset:	0	0	0	0	0	0	0	0

Figure 16-5. Port A LED Control Register (LEDA)

LEDA[7:0] — Port A LED Drive Enable Bits

These read/write bits are software programmable to enable the direct LED drive on an output port pin.

1 = Corresponding port A pin is configured for direct LED drive, with 15 mA current sinking capability

0 = Corresponding port A pin is configured for standard drive

16.3 Port B

Port B is an 8-bit special-function port that shares two of its pins with the multi-master IIC (MMIIC) module, two of its pins with SCI module, and four of its pins with two timer interface (TIM1 and TIM2) modules.

NOTE

PTB3–PTB0 are open-drain pins when configured as outputs regardless whether the pins are used as general purpose I/O pins, MMIIC pins, or SCI pins. Therefore, when configured as general purpose output pins, MMIIC pins, or SCI pins (the TxD pin), pullup resistors must be connected to these pins.

16.3.1 Port B Data Register (PTB)

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 Function:	T2CH1	T2CH0	T1CH1	T1CH0	RxD	TxD	SCL	SDA

Figure 16-6. 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.

Chapter 21
Break Module (BRK)

21.1 Introduction

This section describes the break module. The break module can generate a break interrupt that stops normal program flow at a defined address to enter a background program.

21.2 Features

Features of the break module include:

- Accessible input/output (I/O) registers during the break interrupt
- CPU-generated break interrupts
- Software-generated break interrupts
- COP disabling during break interrupts

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE00	SIM Break Status Register (SBSR)	Read:	R	R	R	R	R	R	SBSW	R
		Write:							Note	
		Reset:							0	
$FE03	SIM Break Flag Control Register (SBFCR)	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE0C	Break Address Register High (BRKH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0D	Break Address Register Low (BRKL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0E	Break Status and Control Register (BRKSCR)	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:			▨	▨	▨	▨	▨	▨
		Reset:	0	0	0	0	0	0	0	0

Note: Writing a logic 0 clears BW. ▨ = Unimplemented R = Reserved

Figure 21-1. Break Module I/O Register Summary

21.3 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal to the CPU. The CPU then loads the instruction register with a software interrupt instruction (SWI) after completion of the current CPU instruction. The program counter vectors to $FFFC and $FFFD ($FEFC and $FEFD in monitor mode).