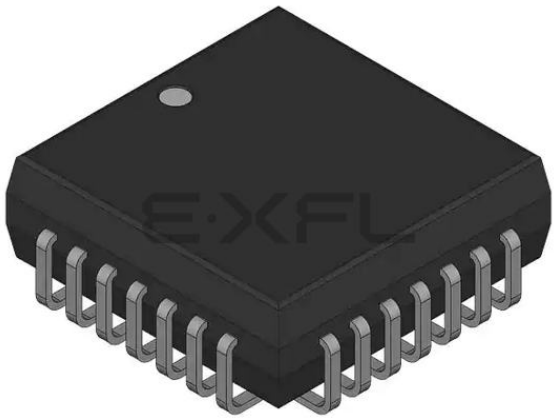


Welcome to E-XFL.COM

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	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K 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	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908ap32cfae

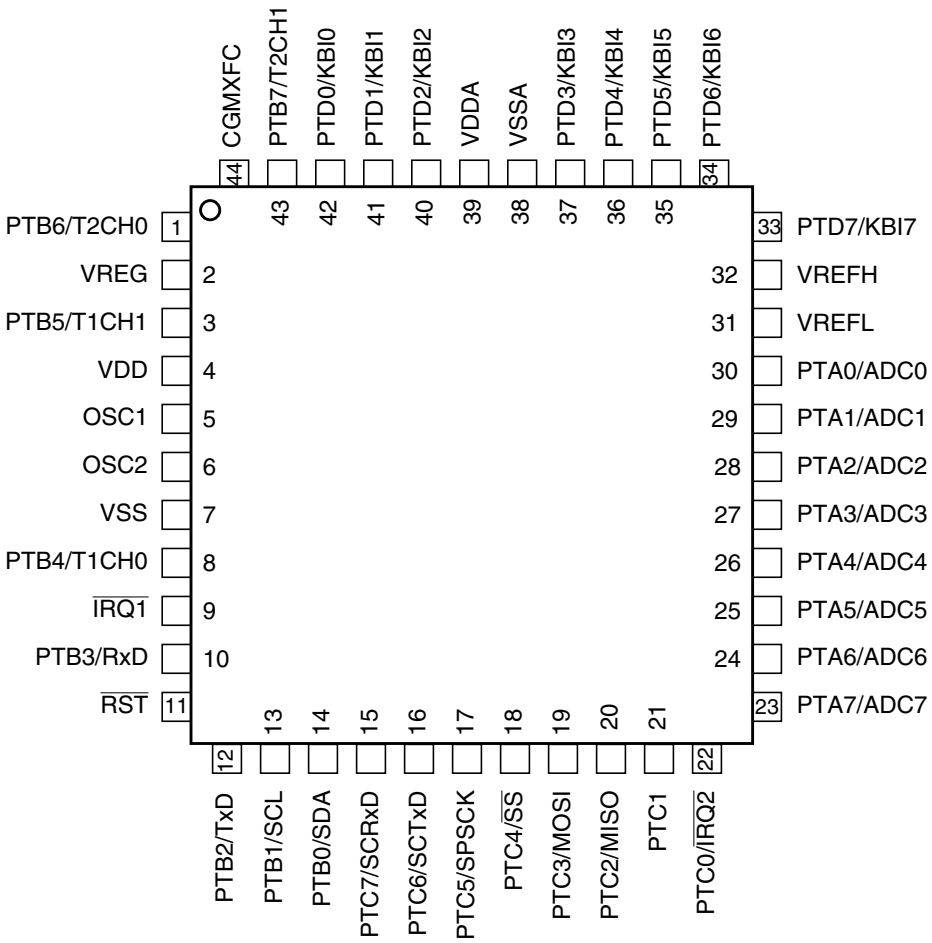


Figure 1-3. 44-Pin QFP Pin Assignments

Chapter 2
Memory

2.1 Introduction

The CPU08 can address 64k-bytes of memory space. The memory map, shown in Figure 2-1, includes:

- 62,368 bytes of user FLASH — MC68HC908AP64
  32,768 bytes of user FLASH — MC68HC908AP32
  16,384 bytes of user FLASH — MC68HC908AP16
  8,192 bytes of user FLASH — MC68HC908AP8
- 2,048 bytes of RAM — MC68HC908AP64 and MC68HC908AP32
  1,024 bytes of RAM — MC68HC908AP16 and MC68HC908AP8
- 48 bytes of user-defined vectors
- 959 bytes of monitor ROM

2.2 Input/Output (I/O) Section

Most of the control, status, and data registers are in the zero page area of $0000–$005F. Additional I/O registers have these addresses:

- $FE00; SIM break status register, SBSR
- $FE01; SIM reset status register, SRSR
- $FE02; Reserved
- $FE03; SIM break flag control register, SBFCR
- $FE04; interrupt status register 1, INT1
- $FE05; interrupt status register 2, INT2
- $FE06; interrupt status register 3, INT3
- $FE07; Reserved
- $FE08; FLASH control register, FLCR
- $FE09; FLASH block protect register, FLBPR
- $FE0A; Reserved
- $FE0B; Reserved
- $FE0C; Break address register high, BRKH
- $FE0D; Break address register low, BRKL
- $FE0E; Break status and control register, BRKSCR
- $FE0F; LVI Status register, LVISR
- $FFCF; Mask option register, MOR (FLASH register)
- $FFFF; COP control register, COPCTL

2.3 Monitor ROM

The 959 bytes at addresses $FC00–$FDFF and $FE10–$FFCE are reserved ROM addresses that contain the instructions for the monitor functions. (See Chapter 8 Monitor ROM (MON).)

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

- Phase detector
- Loop filter
- Lock detector

The operating range of the VCO is programmable for a wide range of frequencies and for maximum immunity to external noise, including supply and CGMXFC noise. The VCO frequency is bound to a range from roughly one-half to twice the center-of-range frequency, f_{VRS}. Modulating the voltage on the CGMXFC pin changes the frequency within this range. By design, f_{VRS} is equal to the nominal center-of-range frequency, f_{NOM}, (125 kHz) times a linear factor, L, and a power-of-two factor, E, or $(L \times 2^E)f_{NOM}$.

CGMRCLK is the PLL reference clock, a buffered version of CGMXCLK. CGMRCLK runs at a frequency, f_{RCLK}, and is fed to the PLL through a programmable modulo reference divider, which divides f_{RCLK} by a factor, R. The divider's output is the final reference clock, CGMRDV, running at a frequency, $f_{RDV} = f_{RCLK}/R$. With an external crystal (30kHz–100kHz), always set R = 1 for specified performance. With an external high-frequency clock source, use R to divide the external frequency to between 30kHz and 100kHz.

The VCO's output clock, CGMVCLK, running at a frequency, f_{VCLK}, is fed back through a programmable pre-scaler divider and a programmable modulo divider. The pre-scaler divides the VCO clock by a power-of-two factor P (the CGMPCLK) and the modulo divider reduces the VCO clock by a factor, N. The dividers' output is the VCO feedback clock, CGMVDV, running at a frequency, $f_{VDV} = f_{VCLK}/(N \times 2^P)$. (See 6.3.6 Programming the PLL for more information.)

The phase detector then compares the VCO feedback clock, CGMVDV, with the final reference clock, CGMRDV. A correction pulse is generated based on the phase difference between the two signals. The loop filter then slightly alters the DC voltage on the external capacitor connected to CGMXFC based on the width and direction of the correction pulse. The filter can make fast or slow corrections depending on its mode, described in 6.3.4 Acquisition and Tracking Modes. The value of the external capacitor and the reference frequency determines the speed of the corrections and the stability of the PLL.

The lock detector compares the frequencies of the VCO feedback clock, CGMVDV, and the final reference clock, CGMRDV. Therefore, the speed of the lock detector is directly proportional to the final reference frequency, f_{RDV}. The circuit determines the mode of the PLL and the lock condition based on this comparison.

6.3.4 Acquisition and Tracking Modes

The PLL filter is manually or automatically configurable into one of two operating modes:

- Acquisition mode — In acquisition mode, the filter can make large frequency corrections to the VCO. This mode is used at PLL start up or when the PLL has suffered a severe noise hit and the VCO frequency is far off the desired frequency. When in acquisition mode, the $\overline{ACQ}$ bit is clear in the PLL bandwidth control register. (See 6.5.2 PLL Bandwidth Control Register.)
- Tracking mode — In tracking mode, the filter makes only small corrections to the frequency of the VCO. PLL jitter is much lower in tracking mode, but the response to noise is also slower. The PLL enters tracking mode when the VCO frequency is nearly correct, such as when the PLL is selected as the base clock source. (See 6.3.8 Base Clock Selector Circuit.) The PLL is automatically in tracking mode when not in acquisition mode or when the $\overline{ACQ}$ bit is set.

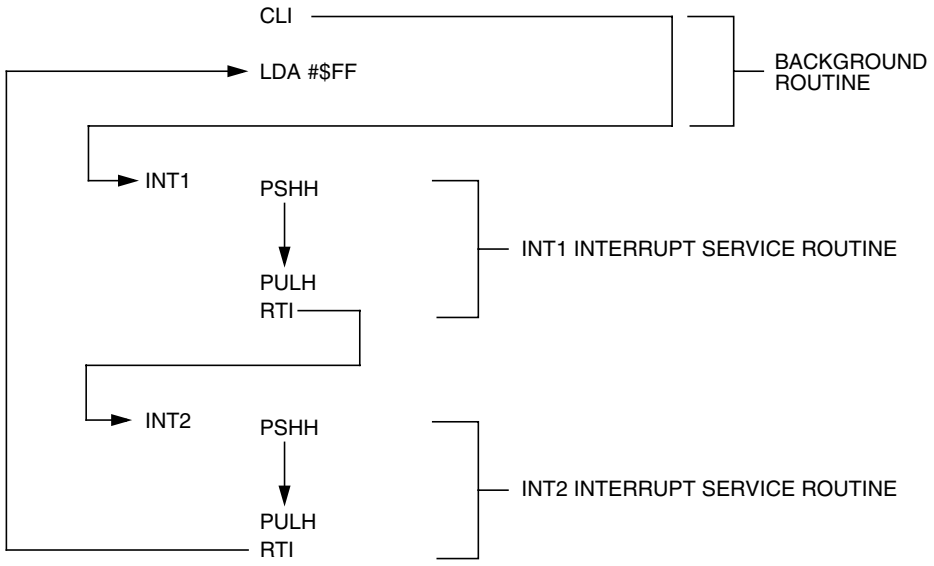


Figure 7-11. Interrupt Recognition Example

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.

7.5.1.2 SWI Instruction

The SWI instruction is a non-maskable instruction that causes an interrupt regardless of the state of the interrupt mask (I bit) in the condition code register.

NOTE

A software interrupt pushes PC onto the stack. A software interrupt does not push PC – 1, as a hardware interrupt does.

7.5.2 Interrupt Status Registers

The flags in the interrupt status registers identify maskable interrupt sources. Table 7-3 summarizes the interrupt sources and the interrupt status register flags that they set. The interrupt status registers can be useful for debugging.

7.7.1 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from stop mode or wait mode.

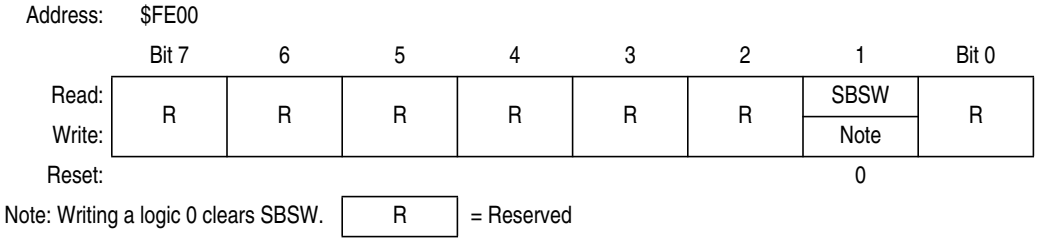


Figure 7-20. SIM Break Status Register (SBSR)

SBSW — Break Wait Bit

This status bit is set when a break interrupt causes an exit from wait mode or stop mode. Clear SBSW by writing a logic 0 to it. Reset clears SBSW.

1 = Stop mode or wait mode was exited by break interrupt

0 = Stop mode or wait mode was not exited by break interrupt

SBSW can be read within the break interrupt routine. The user can modify the return address on the stack by subtracting 1 from it. The following code is an example.

```
This code works if the H register has been pushed onto the stack in the break
service routine software. This code should be executed at the end of the break
service routine software.
HIBYTE   EQU    5
LOBYTE   EQU    6
         If not SBSW, do RTI
         BRCLR   SBSW,SBSR, RETURN    ;See if wait mode or stop mode was exited by
                                      ;break.
         TST     LOBYTE,SP            ;If RETURNLO is not zero,
         BNE     DOLO                 ;then just decrement low byte.
         DEC     HIBYTE,SP            ;Else deal with high byte, too.
DOLO     DEC     LOBYTE,SP            ;Point to WAIT/STOP opcode.
RETURN   PULH                         ;Restore H register.
         RTI
```

7.7.3 SIM Break Flag Control Register

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

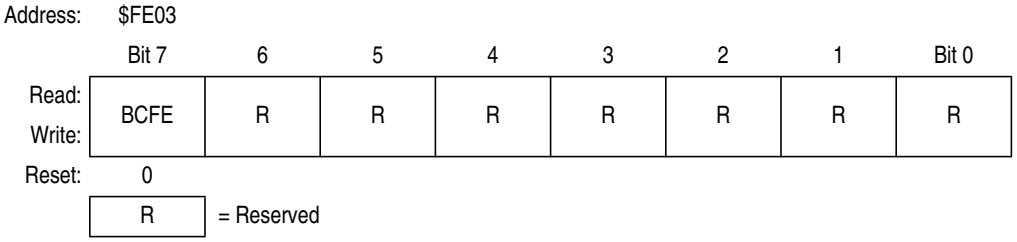


Figure 7-22. SIM Break Flag Control Register (SBFCR)

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

8.3.5 Commands

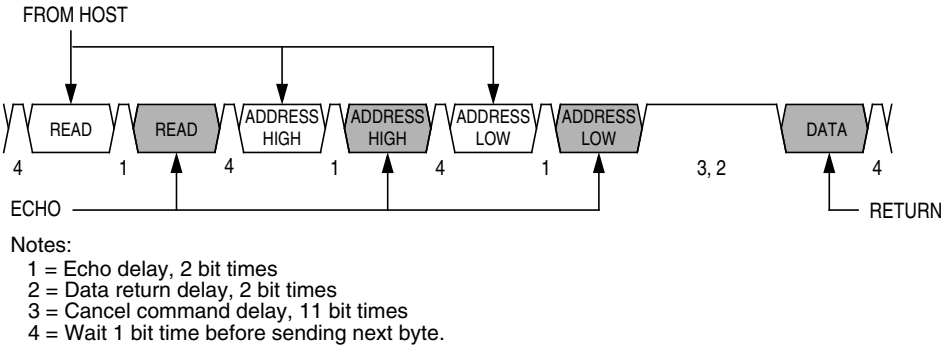
The monitor ROM firmware uses these commands:

- READ (read memory)
- WRITE (write memory)
- IREAD (indexed read)
- IWRITE (indexed write)
- READSP (read stack pointer)
- RUN (run user program)

The monitor ROM firmware echoes each received byte back to the PTA0 pin for error checking. An 11-bit delay at the end of each command allows the host to send a break character to cancel the command. A delay of two bit times occurs before each echo and before READ, IREAD, or READSP data is returned. The data returned by a read command appears after the echo of the last byte of the command.

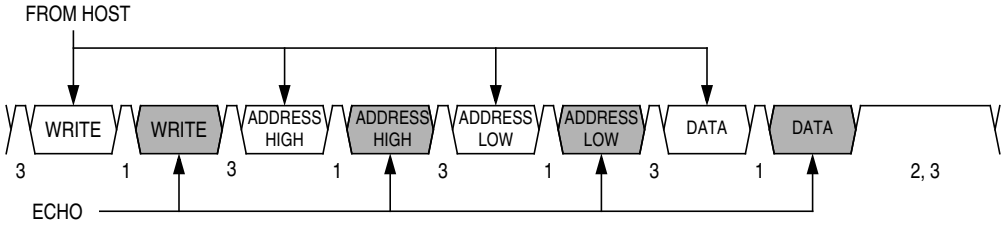
NOTE

Wait one bit time after each echo before sending the next byte.



Notes:
1 = Echo delay, 2 bit times
2 = Data return delay, 2 bit times
3 = Cancel command delay, 11 bit times
4 = Wait 1 bit time before sending next byte.

Figure 8-5. Read Transaction



Notes:
1 = Echo delay, 2 bit times
2 = Cancel command delay, 11 bit times
3 = Wait 1 bit time before sending next byte.

Figure 8-6. Write Transaction

A brief description of each monitor mode command is given in Table 8-4 through Table 8-9.

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.

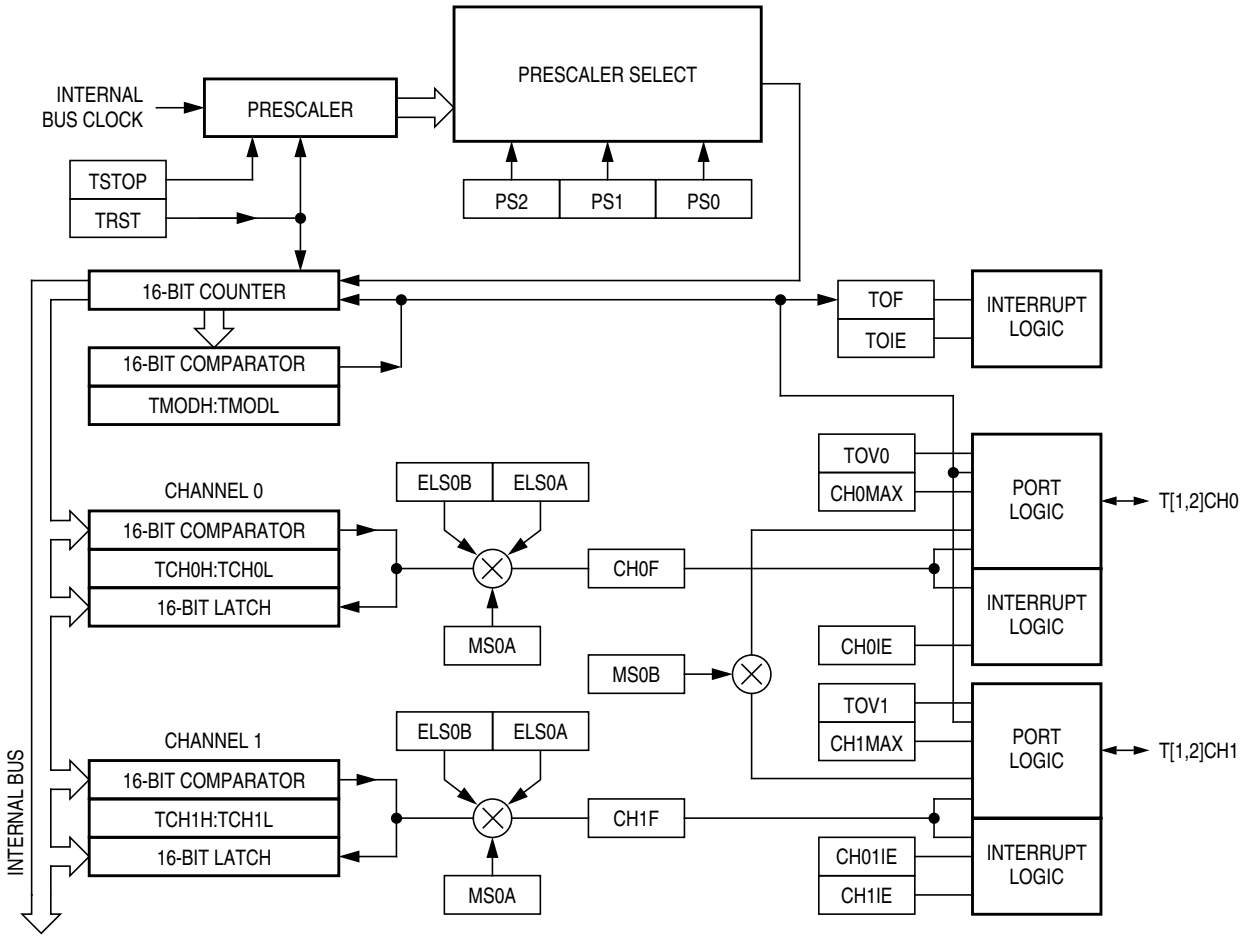


Figure 9-1. TIM Block Diagram

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.

the received byte can be read. If the SCI receive interrupt enable bit, SCRIE, in SCC2 is also set, the SCRF bit generates a receiver CPU interrupt request.

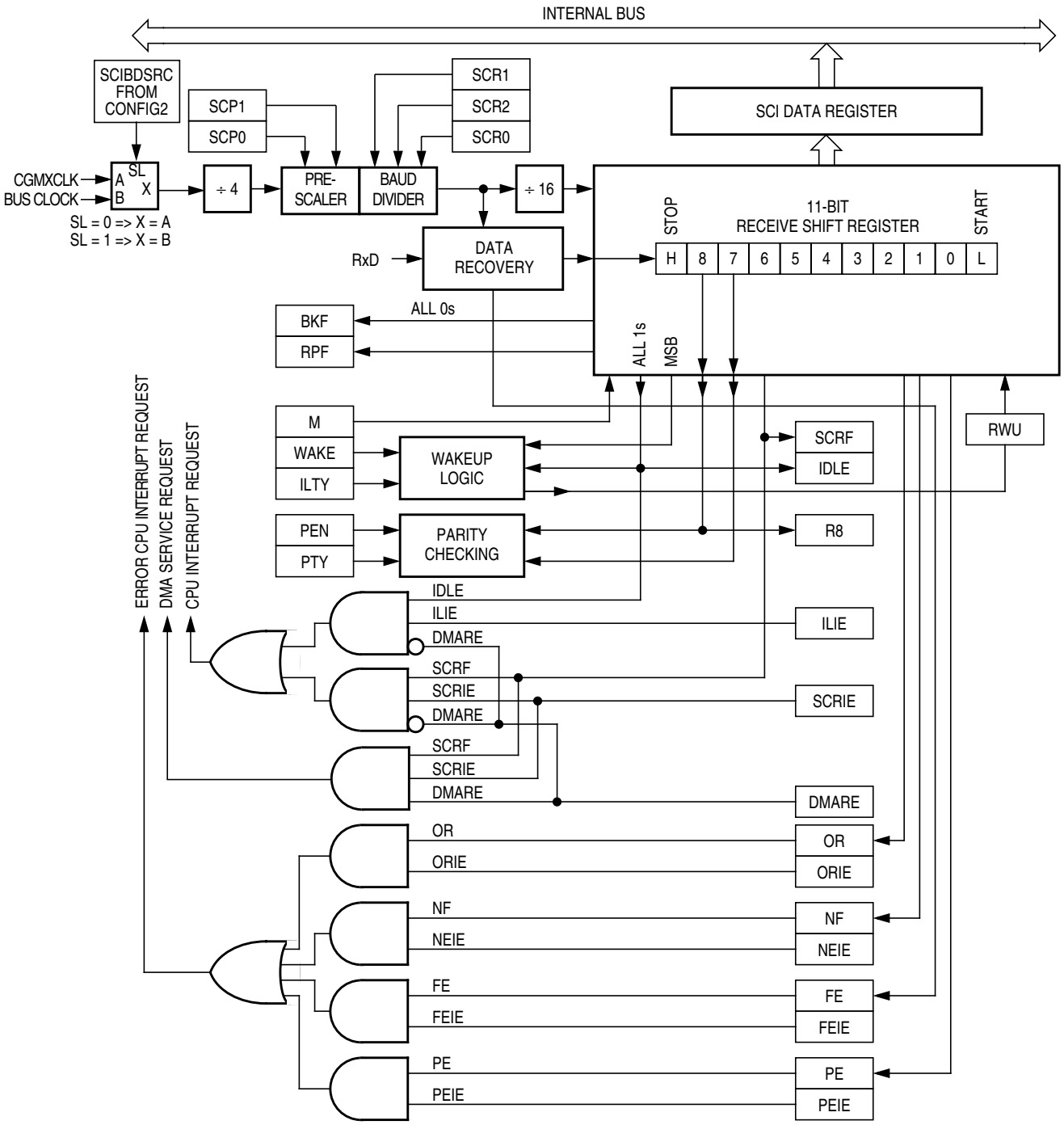


Figure 11-5. SCI Receiver Block Diagram

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

NOTE

For SCI operations, the IR sub-module is transparent to the SCI module. Data at going out of the SCI transmitter and data going into the SCI receiver is always in SCI format. It makes no difference to the SCI module whether the IR sub-module is enabled or disabled.

NOTE

This SCI module is a standard HC08 SCI module with the following modifications:

- *A control bit, CKS, is added to the SCI baud rate control register to select between two input clocks for baud rate clock generation*
- *The TXINV bit is removed from the SCI control register 1*

12.5.1 Data Format

The SCI uses the standard non-return-to-zero mark/space data format illustrated in Figure 12-6.

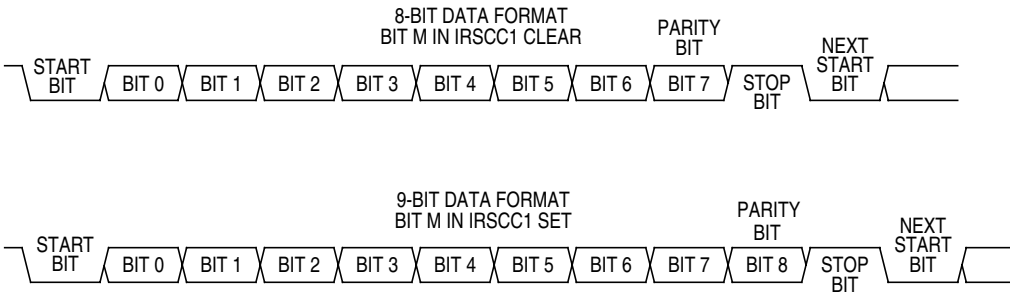


Figure 12-6. SCI Data Formats

12.5.2 Transmitter

Figure 12-7 shows the structure of the SCI transmitter.

The baud rate clock source for the SCI can be selected by the CKS bit, in the SCI baud rate register (see 12.9.7 IRSCI Baud Rate Register).

12.5.2.1 Character Length

The transmitter can accommodate either 8-bit or 9-bit data. The state of the M bit in IRSCI control register 1 (IRSCC1) determines character length. When transmitting 9-bit data, bit T8 in IRSCI control register 3 (IRSCC3) is the ninth bit (bit 8).

- 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.

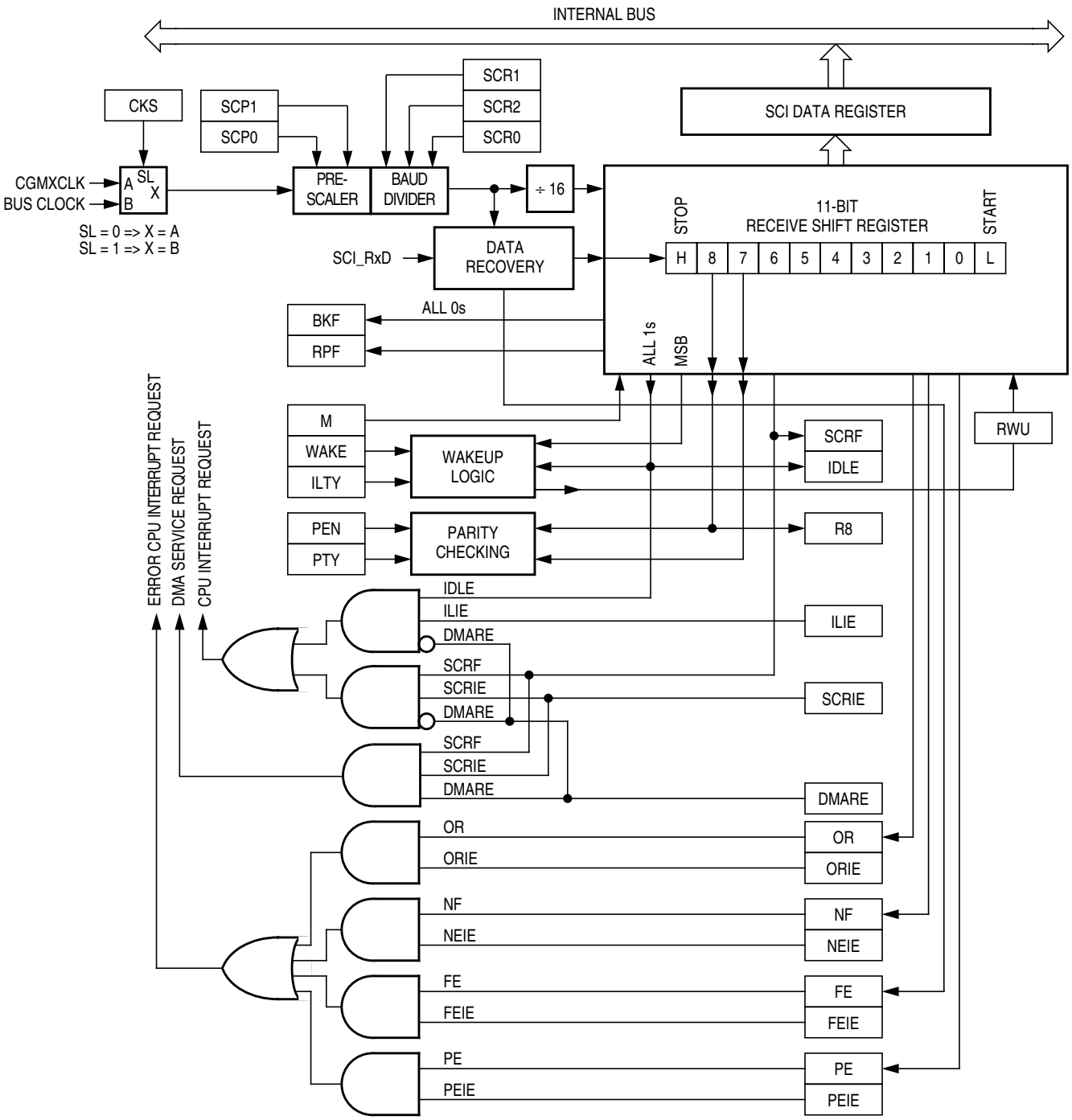


Figure 12-8. SCI Receiver Block Diagram

12.5.3.1 Character Length

The receiver can accommodate either 8-bit or 9-bit data. The state of the M bit in IRSCI control register 1 (IRSCC1) determines character length. When receiving 9-bit data, bit R8 in IRSCI control register 2 (IRSCC2) is the ninth bit (bit 8). When receiving 8-bit data, bit R8 is a copy of the eighth bit (bit 7).

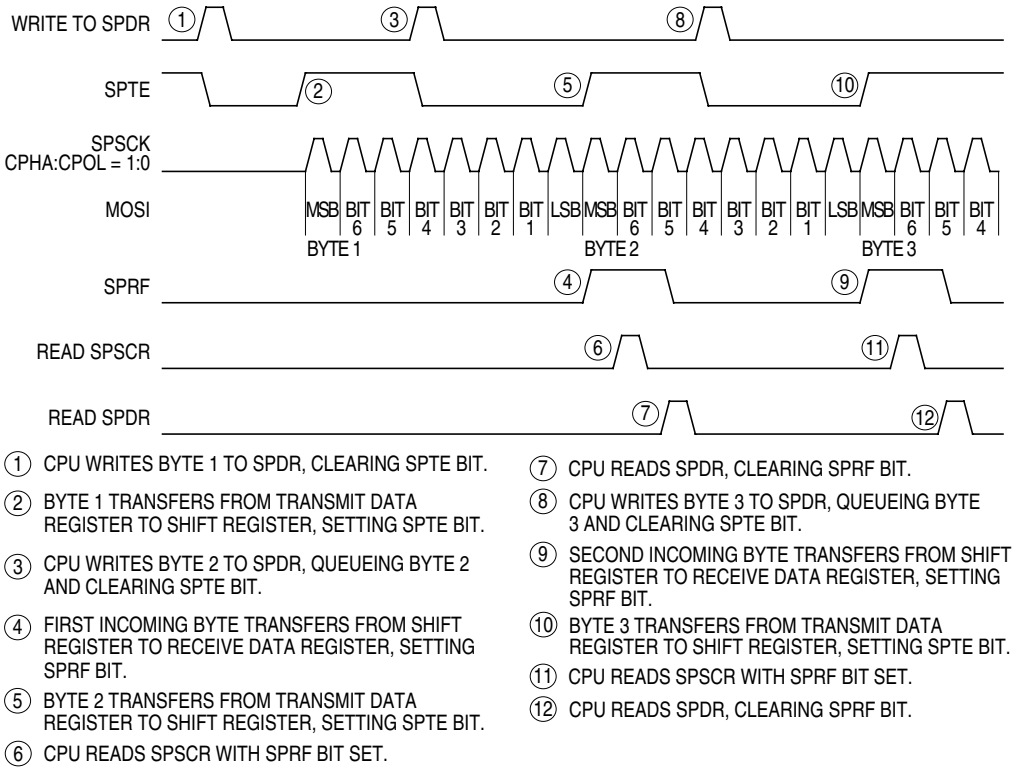


Figure 13-8. SPRF/SPTE CPU Interrupt Timing

The transmit data buffer allows back-to-back transmissions without the slave precisely timing its writes between transmissions as in a system with a single data buffer. Also, if no new data is written to the data buffer, the last value contained in the shift register is the next data word to be transmitted.

For an idle master or idle slave that has no data loaded into its transmit buffer, the SPTE is set again no more than two bus cycles after the transmit buffer empties into the shift register. This allows the user to queue up a 16-bit value to send. For an already active slave, the load of the shift register cannot occur until the transmission is completed. This implies that a back-to-back write to the transmit data register is not possible. The SPTE indicates when the next write can occur.

13.7 Error Conditions

The following flags signal SPI error conditions:

- Overflow (OVRF) — Failing to read the SPI data register before the next full byte enters the shift register sets the OVRF bit. The new byte does not transfer to the receive data register, and the unread byte still can be read. OVRF is in the SPI status and control register.
- Mode fault error (MODF) — The MODF bit indicates that the voltage on the slave select pin ($\overline{SS}$) is inconsistent with the mode of the SPI. MODF is in the SPI status and control register.

Table 15-1. MUX Channel Select

ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	ADC Channel	Input Select
0	0	0	0	0	ADC0	PTA0
0	0	0	0	1	ADC1	PTA1
0	0	0	1	0	ADC2	PTA2
0	0	0	1	1	ADC3	PTA3
0	0	1	0	0	ADC4	PTA4
0	0	1	0	1	ADC5	PTA5
0	0	1	1	0	ADC6	PTA6
0	0	1	1	1	ADC7	PTA7
0 ↓ 1	1 ↓ 1	0 ↓ 1	0 ↓ 0	0 ↓ 0	ADC8 ↓ ADC28	Reserved
1	1	1	0	1	ADC29	V_{REFH} (see Note 2)
1	1	1	1	0	ADC30	V_{REFL} (see Note 2)
1	1	1	1	1	ADC powered-off	—

NOTES:
1. If any unused channels are selected, the resulting ADC conversion will be unknown.
2. The voltage levels supplied from internal reference nodes as specified in the table are used to verify the operation of the ADC converter both in production test and for user applications.

15.7.2 ADC Clock Control Register

The ADC clock control register (ADICLK) selects the clock frequency for the ADC.

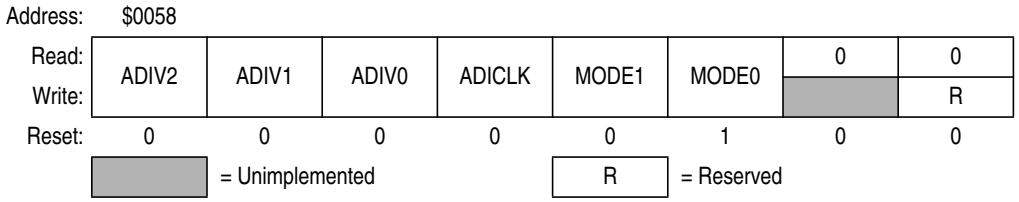


Figure 15-4. ADC Clock Control Register (ADICLK)

ADIV[2:0] — ADC Clock Prescaler Bits

ADIV2, ADIV1, and ADIV0 form a 3-bit field which selects the divide ratio used by the ADC to generate the internal ADC clock.

Table 15-2 shows the available clock configurations. The ADC clock should be set to between 500 kHz and 1MHz.

NOTE

When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine.

The $\overline{\text{IRQ1}}$ pin has a permanent internal pullup device connected, while the $\overline{\text{IRQ2}}$ pin has an optional pullup device that can be enabled or disabled by the PUC0ENB bit in the INTSCR2 register.

17.5 IRQ Module During Break Interrupts

The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear the latch during the break state. (See Chapter 21 Break Module (BRK).)

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

To protect CPU interrupt flags during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), writing to the ACK bit in the IRQ status and control register during the break state has no effect on the IRQ interrupt flags.

17.6 IRQ Registers

Each IRQ is controlled and monitored by an status and control register.

- IRQ1 Status and Control Register — $001E
- IRQ2 Status and Control Register — $001C

17.6.1 IRQ1 Status and Control Register

The IRQ1 status and control register (INTSCR1) controls and monitors operation of IRQ1. The INTSCR1 has the following functions:

- Shows the state of the IRQ1 flag
- Clears the IRQ1 latch
- Masks IRQ1 interrupt request
- Controls triggering sensitivity of the $\overline{\text{IRQ1}}$ interrupt pin

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	IRQ1F	0	IMASK1	MODE1
Write:						ACK1		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 17-4. IRQ1 Status and Control Register (INTSCR1)

IRQ1F — IRQ1 Flag Bit

This read-only status bit is high when the IRQ1 interrupt is pending.

1 = $\overline{\text{IRQ1}}$ interrupt pending
0 = $\overline{\text{IRQ1}}$ interrupt not pending

ACK1 — IRQ1 Interrupt Request Acknowledge Bit

Writing a logic 1 to this write-only bit clears the IRQ1 latch. ACK1 always reads as logic 0. Reset clears ACK1.

IMASK1 — IRQ1 Interrupt Mask Bit

Writing a logic 1 to this read/write bit disables IRQ1 interrupt requests. Reset clears IMASK1.

1 = $\overline{\text{IRQ1}}$ interrupt requests disabled
0 = $\overline{\text{IRQ1}}$ interrupt requests enabled

MODE1 — IRQ1 Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the $\overline{\text{IRQ1}}$ pin. Reset clears MODE1.

1 = $\overline{\text{IRQ1}}$ interrupt requests on falling edges and low levels
0 = $\overline{\text{IRQ1}}$ interrupt requests on falling edges only

17.6.2 IRQ2 Status and Control Register

The IRQ2 status and control register (INTSCR2) controls and monitors operation of IRQ2. The INTSCR2 has the following functions:

- Enables/disables the internal pullup device on $\overline{\text{IRQ2}}$ pin
- Shows the state of the IRQ2 flag
- Clears the IRQ2 latch
- Masks IRQ2 interrupt request
- Controls triggering sensitivity of the $\overline{\text{IRQ2}}$ interrupt pin

Address: $001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PUC0ENB	0	0	IRQ2F	0	IMASK2	MODE2
Write:						ACK2		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 17-5. IRQ2 Status and Control Register (INTSCR2)

PUC0ENB — $\overline{\text{IRQ2}}$ Pin Pullup Enable Bit.

Setting this bit to logic 1 disables the pullup on PTC0/$\overline{\text{IRQ2}}$ pin.
Reset clears this bit.

1 = $\overline{\text{IRQ2}}$ pin internal pullup is disabled
0 = $\overline{\text{IRQ2}}$ pin internal pullup is enabled

IRQ2F — IRQ2 Flag Bit

This read-only status bit is high when the IRQ2 interrupt is pending.

1 = $\overline{\text{IRQ2}}$ interrupt pending
0 = $\overline{\text{IRQ2}}$ interrupt not pending

ACK2 — IRQ2 Interrupt Request Acknowledge Bit

Writing a logic 1 to this write-only bit clears the IRQ2 latch. ACK2 always reads as logic 0. Reset clears ACK2.

IMASK2 — IRQ2 Interrupt Mask Bit

Writing a logic 1 to this read/write bit disables IRQ2 interrupt requests. Reset clears IMASK2.

1 = IRQ2 interrupt requests disabled
0 = IRQ2 interrupt requests enabled

MODE2 — IRQ2 Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the $\overline{\text{IRQ2}}$ pin. Reset clears MODE2.

1 = $\overline{\text{IRQ2}}$ interrupt requests on falling edges and low levels
0 = $\overline{\text{IRQ2}}$ interrupt requests on falling edges only

21.4.1 Wait Mode

If enabled, the break module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if SBSW is set. (see Chapter 7 System Integration Module (SIM)) Clear the BW bit by writing logic 0 to it.

21.4.2 Stop Mode

A break interrupt causes exit from stop mode and sets the SBSW bit in the break status register.

21.5 Break Module Registers

These registers control and monitor operation of the break module:

- Break status and control register (BRKSCR)
- Break address register high (BRKH)
- Break address register low (BRKL)
- SIM break status register (SBSR)
- SIM break flag control register (SBFCR)

21.5.1 Break Status and Control Register

The break status and control register (BRKSCR) contains break module enable and status bits.

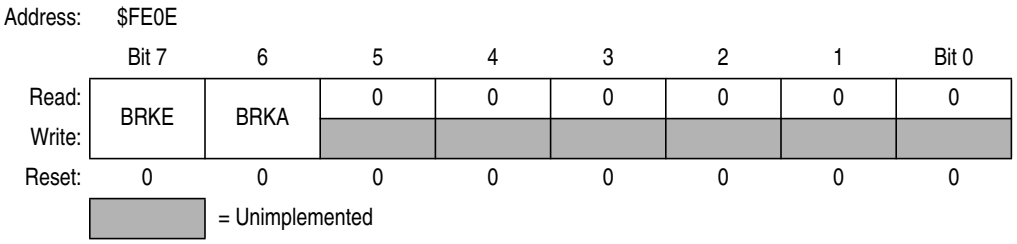


Figure 21-3. Break Status and Control Register (BRKSCR)

BRKE — Break Enable Bit

This read/write bit enables breaks on break address register matches. Clear BRKE by writing a logic 0 to bit 7. Reset clears the BRKE bit.

1 = Breaks enabled on 16-bit address match
0 = Breaks disabled on 16-bit address match

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a logic 1 to BRKA generates a break interrupt. Clear BRKA by writing a logic 0 to it before exiting the break routine. Reset clears the BRKA bit.

1 = (When read) Break address match
0 = (When read) No break address match

22.3 Functional Operating Range

Table 22-2. Operating Range

Characteristic	Symbol	Value	Unit
Operating temperature range	T_A	−40 to +85	°C
Operating voltage range	V_{DD}	2.7 to 5.5	V

22.4 Thermal Characteristics

Table 22-3. Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance 42-Pin SDIP 44-Pin QFP 48-Pin LQFP	θ_{JA}	 60 95 80	 °C/W °C/W °C/W
I/O pin power dissipation	$P_{I/O}$	User determined	W
Power dissipation(1)	P_D	$P_D = (I_{DD} \times V_{DD}) + P_{I/O} = K/(T_J + 273\ °C)$	W
Constant(2)	K	$P_D \times (T_A + 273\ °C) + P_D^2 \times \theta_{JA}$	W/°C
Average junction temperature	T_J	$T_A + (P_D \times \theta_{JA})$	°C
Maximum junction temperature	T_{JM}	100	°C

1. Power dissipation is a function of temperature.
2. K constant unique to the device. K can be determined for a known T_A and measured P_D. With this value of K, P_D and T_J can be determined for any value of T_A.