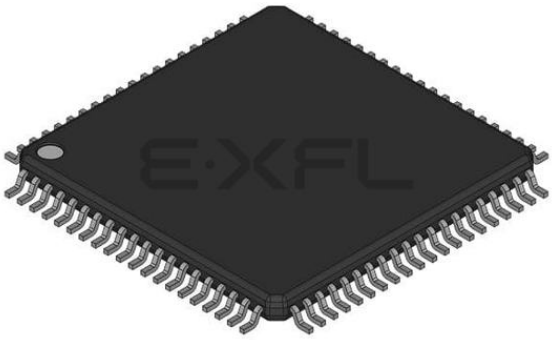


Welcome to E-XFL.COM

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, SPI
Peripherals	LCD, LVD, POR, PWM
Number of I/O	40
Program Memory Size	24KB (24K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 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	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mchc908lk24cfqe

4.7 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 4-3** 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. Read the FLASH block protect register.
3. Write any data to any FLASH address within the row address range desired.
4. Wait for a time, t_{nvs} (10 µs).
5. Set the HVEN bit.
6. Wait for a time, t_{pgs} (5 µs).
7. Write data to the FLASH address to be programmed.
8. Wait for time, t_{prog} (30 µs).
9. Repeat steps 7 and 8 until all bytes within the row are programmed.
10. Clear the PGM bit.
11. Wait for time, t_{nvh} (5 µs).
12. Clear the HVEN bit.
13. 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 7 to step 7), or the time between the last FLASH addressed programmed to clearing the PGM bit (step 7 to step 10), 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.*

8.2 Introduction

This section describes the clock generator module (CGM). The CGM generates the base clock signal, CGMOUT, which is based on either the oscillator clock divided by two or the divided phase-locked loop (PLL) clock, CGMPCLK, divided by two. CGMOUT is the clock from which the SIM derives the system clocks, including the bus clock, which is at a frequency of CGMOUT÷2.

The PLL is a frequency generator designed for use with a low frequency crystal (typically 32.768kHz) to generate a base frequency and dividing to a maximum bus frequency of 8MHz.

8.4.1 Oscillator Module

The oscillator module provides two clock outputs CGMXCLK and CGMRCLK to the CGM module. CGMXCLK or CGMXCLK divide-by-two can be selected to drive the SIM module to generate the system bus clocks. CGMRCLK is the reference clock for the phase-lock-loop, to generate a higher frequency clock. The oscillator module also provides the reference clock for the real time clock (RTC) module.
See **Section 7. Oscillator (OSC)** for detailed description on oscillator module. See **Section 12. Real Time Clock (RTC)** for detailed description on RTC.

8.4.2 Phase-Locked Loop Circuit (PLL)

The PLL is a frequency generator that can operate in either acquisition mode or tracking mode, depending on the accuracy of the output frequency. The PLL can change between acquisition and tracking modes either automatically or manually.

8.4.3 PLL Circuits

The PLL consists of these circuits:

- Voltage-controlled oscillator (VCO)
- Reference divider
- Frequency pre-scaler
- Modulo VCO frequency divider
- Phase detector
- Loop filter
- Lock detector

The most critical parameter which affects the reaction times of the PLL is the reference frequency, f_{RDV}. This frequency is the input to the phase detector and controls how often the PLL makes corrections. For stability, the corrections must be small compared to the desired frequency, so several corrections are required to reduce the frequency error. Therefore, the slower the reference the longer it takes to make these corrections. This parameter is under user control via the choice of crystal frequency f_{XCLK} and the R value programmed in the reference divider. (See **8.4.3 PLL Circuits**, **8.4.6 Programming the PLL**, and **8.6.5 PLL Reference Divider Select Register**.)

Another critical parameter is the external filter network. The PLL modifies the voltage on the VCO by adding or subtracting charge from capacitors in this network. Therefore, the rate at which the voltage changes for a given frequency error (thus change in charge) is proportional to the capacitance. The size of the capacitor also is related to the stability of the PLL. If the capacitor is too small, the PLL cannot make small enough adjustments to the voltage and the system cannot lock. If the capacitor is too large, the PLL may not be able to adjust the voltage in a reasonable time. (See **8.9.3 Choosing a Filter**.)

Also important is the operating voltage potential applied to V_{DDA}. The power supply potential alters the characteristics of the PLL. A fixed value is best. Variable supplies, such as batteries, are acceptable if they vary within a known range at very slow speeds. Noise on the power supply is not acceptable, because it causes small frequency errors which continually change the acquisition time of the PLL.

Temperature and processing also can affect acquisition time because the electrical characteristics of the PLL change. The part operates as specified as long as these influences stay within the specified limits. External factors, however, can cause drastic changes in the operation of the PLL. These factors include noise injected into the PLL through the filter capacitor, filter capacitor leakage, stray impedances on the circuit board, and even humidity or circuit board contamination.

9.6 Exception Control

Normal, sequential program execution can be changed in three different ways:

- Interrupts:
  - Maskable hardware CPU interrupts
  - Non-maskable software interrupt instruction (SWI)
- Reset
- Break interrupts

9.6.1 Interrupts

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. **Figure 9-8** shows interrupt entry timing, and **Figure 9-9** shows interrupt recovery timing.

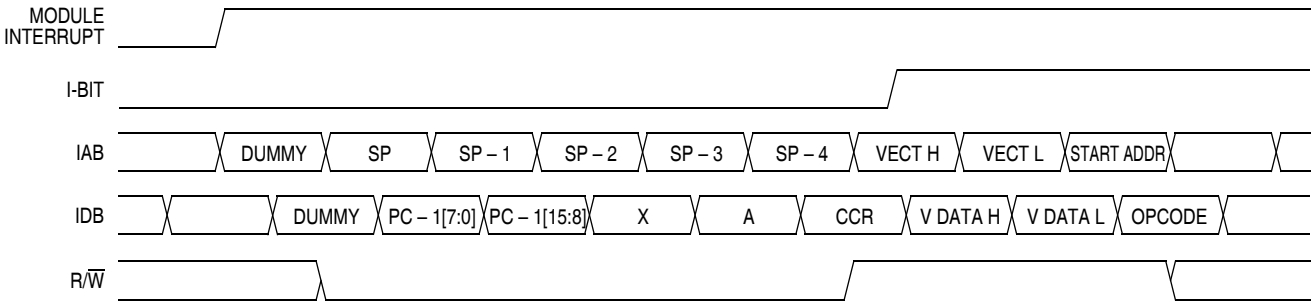


Figure 9-8. Interrupt Entry Timing

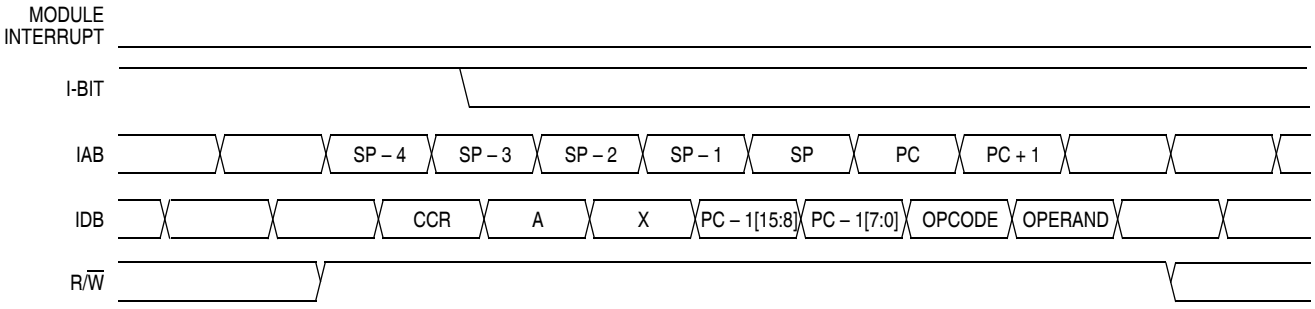
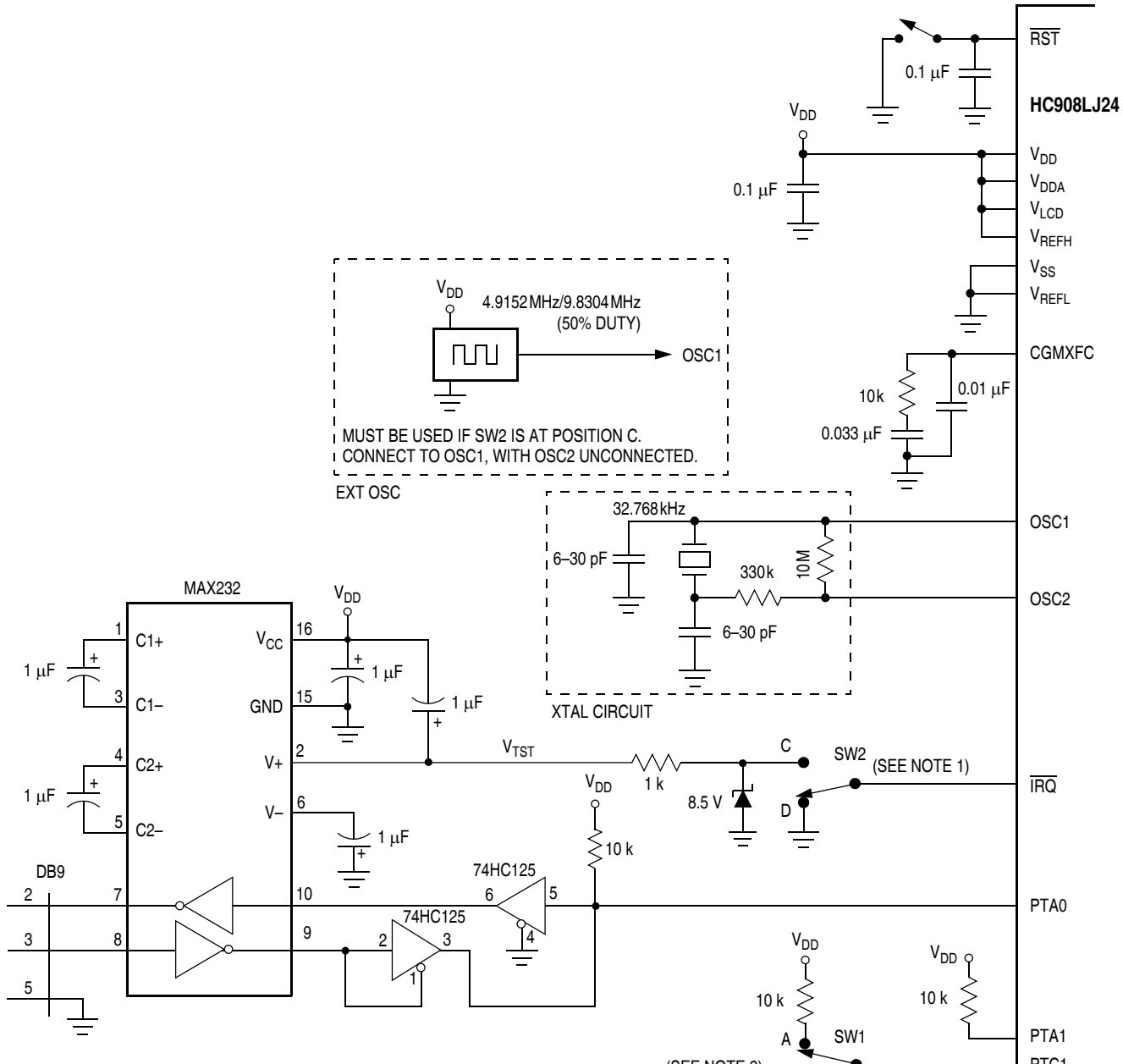
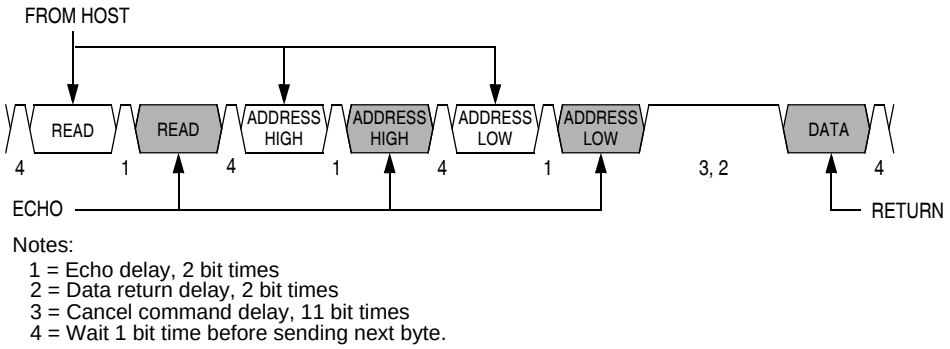


Figure 9-9. Interrupt Recovery Timing



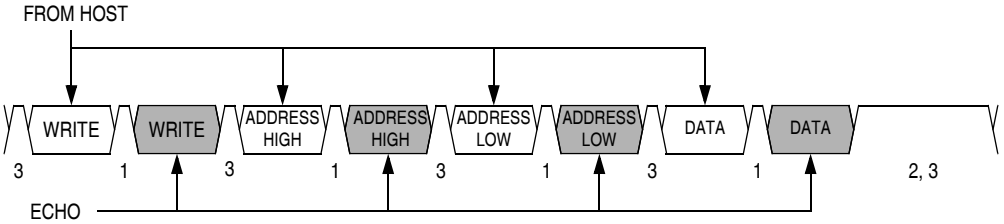
NOTES:
1. Monitor mode entry method:
   SW2: Position C — High voltage entry (V_{TST}); must use external OSC
   Bus clock depends on SW1 (note 2).
   SW2: Position D — Reset vector must be blank ($FFFE:$FFFF = $FF)
   Bus clock = 2.4576 MHz.
2. Affects high voltage entry to monitor mode only (SW2 at position C):
   SW1: Position A — Bus clock = OSC1 ÷ 4
   SW1: Position B — Bus clock = OSC1 ÷ 2
5. See **Table 24-4** for V_{TST} voltage level requirements.

Figure 10-1. Monitor Mode Circuit



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 10-5. 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 10-6. Write Transaction

A brief description of each monitor mode command is given in **Table 10-4** through **Table 10-9**.

Table 10-4. READ (Read Memory) Command

Description	Read byte from memory
Operand	2-byte address in high-byte:low-byte order
Data Returned	Returns contents of specified address
Opcode	$4A
Command Sequence	

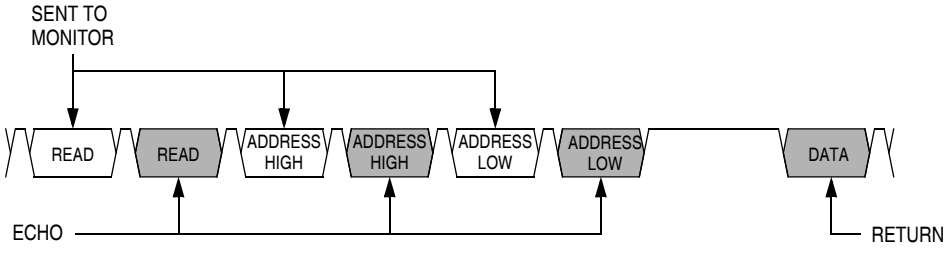
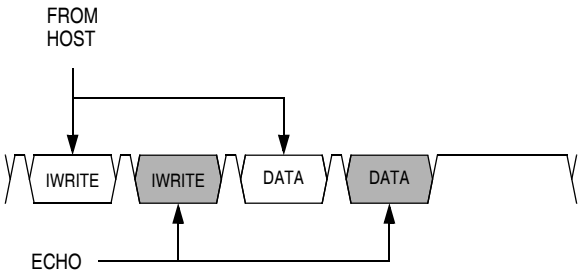
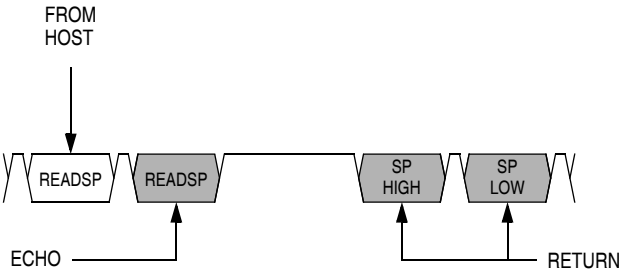


Table 10-7. IWRITE (Indexed Write) Command

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

A sequence of IREAD or IWRITE commands can access a block of memory sequentially over the full 64-Kbyte memory map.

Table 10-8. READSP (Read Stack Pointer) Command

Description	Reads stack pointer
Operand	None
Data Returned	Returns incremented stack pointer value (SP + 1) in high-byte:low-byte order
Opcode	$0C
Command Sequence	

10.6.6 MON_LDRNGE

In monitor mode, LDRNGE is used to load the data array in RAM with data from a range of FLASH locations.

Table 10-16. ICP_LDRNGE Routine

Routine Name	MON_LDRNGE
Routine Description	Loads data from a range of locations, in monitor mode
Calling Address	$FF24
Stack Used	11 bytes
Data Block Format	Bus speed Data size Starting address (high byte) Starting address (low byte) Data 1 : Data N

The MON_LDRNGE routine is designed to be used in monitor mode. It performs the same function as the LDRNGE routine (see **10.6.3 LDRNGE**), except that MON_LDRNGE returns to the main program via an SWI instruction. After a MON_LDRNGE call, the SWI instruction will return the control back to the monitor code.

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*

13.7.1 Data Format

The SCI uses the standard non-return-to-zero mark/space data format illustrated in **Figure 13-6**.

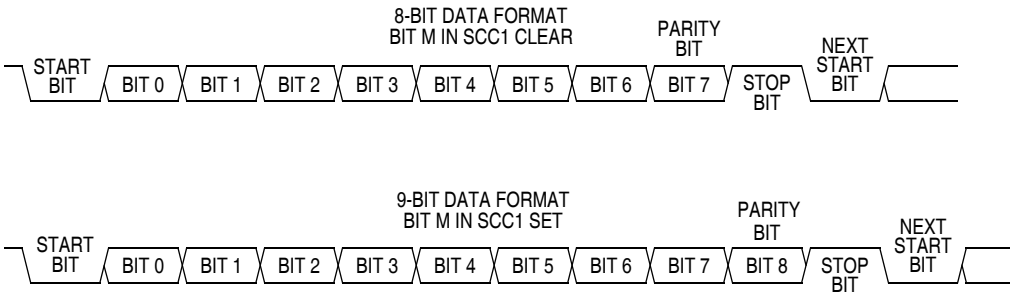


Figure 13-6. SCI Data Formats

13.7.3.7 Receiver Interrupts

The following sources can generate CPU interrupt requests from the SCI receiver:

- SCI receiver full (SCRF) — The SCRF bit in SCS1 indicates that the receive shift register has transferred a character to the SCDR. SCRF can generate a receiver interrupt request. Setting the SCI receive interrupt enable bit, SCRIE, in SCC2 enables the SCRF bit to generate receiver CPU interrupts.
- Idle input (IDLE) — The IDLE bit in SCS1 indicates that 10 or 11 consecutive logic 1s shifted in from the RxD pin. The idle line interrupt enable bit, ILIE, in SCC2 enables the IDLE bit to generate CPU interrupt requests.

13.7.3.8 Error Interrupts

The following receiver error flags in SCS1 can generate CPU interrupt requests:

- Receiver overrun (OR) — The OR bit indicates that the receive shift register shifted in a new character before the previous character was read from the SCDR. The previous character remains in the SCDR, and the new character is lost. The overrun interrupt enable bit, ORIE, in SCC3 enables OR to generate SCI error CPU interrupt requests.
- Noise flag (NF) — The NF bit is set when the SCI detects noise on incoming data or break characters, including start, data, and stop bits. The noise error interrupt enable bit, NEIE, in SCC3 enables NF to generate SCI error CPU interrupt requests.
- Framing error (FE) — The FE bit in SCS1 is set when a logic 0 occurs where the receiver expects a stop bit. The framing error interrupt enable bit, FEIE, in SCC3 enables FE to generate SCI error CPU interrupt requests.
- Parity error (PE) — The PE bit in SCS1 is set when the SCI detects a parity error in incoming data. The parity error interrupt enable bit, PEIE, in SCC3 enables PE to generate SCI error CPU interrupt requests.

13.11.5 SCI Status Register 2 (SCS2)

SCI status register 2 contains flags to signal the following conditions:

- Break character detected
- Incoming data

Address: $0017

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

= Unimplemented

Figure 13-17. SCI Status Register 2 (SCS2)

BKF — Break Flag Bit

This clearable, read-only bit is set when the SCI detects a break character on the RxD pin. In SCS1, the FE and SCRF bits are also set. In 9-bit character transmissions, the R8 bit in SCC3 is cleared. BKF does not generate a CPU interrupt request. Clear BKF by reading SCS2 with BKF set and then reading the SCDR. Once cleared, BKF can become set again only after logic 1s again appear on the RxD pin followed by another break character. Reset clears the BKF bit.

1 = Break character detected
0 = No break character detected

RPF — Reception in Progress Flag Bit

This read-only bit is set when the receiver detects a logic 0 during the RT1 time period of the start bit search. RPF does not generate an interrupt request. RPF is reset after the receiver detects false start bits (usually from noise or a baud rate mismatch) or when the receiver detects an idle character. Polling RPF before disabling the SCI module or entering stop mode can show whether a reception is in progress.

1 = Reception in progress
0 = No reception in progress

13.11.7 SCI Baud Rate Register

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

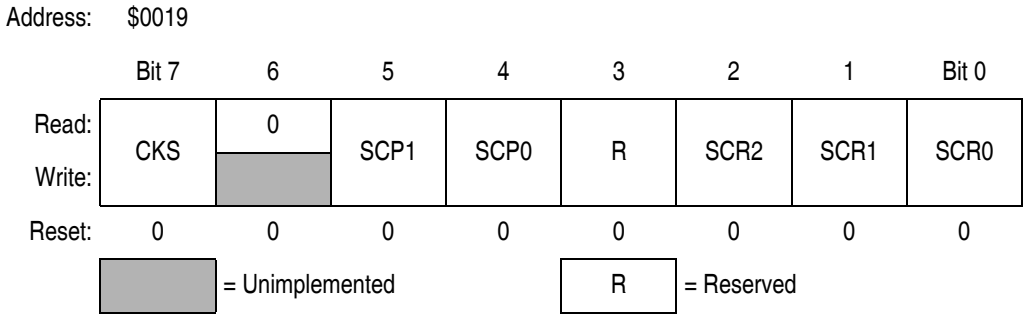


Figure 13-19. SCI Baud Rate Register (SCBR)

CKS — Baud Clock Input Select

This read/write bit selects the source clock for the baud rate generator. Reset clears the CKS bit, selecting CGMXCLK.

1 = Bus clock drives the baud rate generator
0 = CGMXCLK drives the baud rate generator

SCP1 and SCP0 — SCI Baud Rate Prescaler Bits

These read/write bits select the baud rate prescaler divisor as shown in **Table 13-7**. Reset clears SCP1 and SCP0.

Table 13-7. SCI Baud Rate Prescaling

SCP1 and SCP0	Prescaler Divisor (PD)
00	1
01	3
10	4
11	13

SCR2–SCR0 — SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate divisor as shown in **Table 13-8**. Reset clears SCR2–SCR0.

When CPHA = 1 for a slave, the first edge of the SPSCK 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. Once 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 SPSCK. Any data written after the first edge is stored in the transmit data register and transferred to the shift register after the current transmission.

14.6.4 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 SPSCK signal. When CPHA = 0, the SPSCK signal remains inactive for the first half of the first SPSCK cycle. When CPHA = 1, the first SPSCK cycle begins with an edge on the SPSCK 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 14-7**.) 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. SPSCK edges occur halfway through the low time of the internal MCU clock. Since the SPI clock is free-running, it is uncertain where the write to the SPDR occurs relative to the slower SPSCK. This uncertainty causes the variation in the initiation delay shown in **Figure 14-7**. 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.

16.5 Interrupts

When the AIEN bit is set, the ADC module is capable of generating a CPU interrupt after each ADC conversion. A CPU interrupt is generated if the COCO bit is at logic 0. The COCO bit is not used as a conversion complete flag when interrupts are enabled. The interrupt vector is defined in **Table 2-1 . Vector Addresses**.

16.6 Low-Power Modes

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

16.6.1 Wait Mode

The ADC continues normal operation in wait mode. Any enabled CPU interrupt request from the ADC can bring the MCU out of wait mode. If the ADC is not required to bring the MCU out of wait mode, power down the ADC by setting the ADCH[4:0] bits to logic 1's before executing the WAIT instruction.

16.6.2 Stop Mode

The ADC module is inactive after the execution of a STOP instruction. Any pending conversion is aborted. ADC conversions resume when the MCU exits stop mode. Allow one conversion cycle to stabilize the analog circuitry before attempting a new ADC conversion after exiting stop mode.

16.7 I/O Signals

The ADC module has ten channels, six channels are shared with port A and port B I/O pins; two channels are the ADC voltage reference inputs, V_{REFH} and V_{REFL}; one channel is the V_{LCD} input; and one channel is the 1.2V bandgap reference voltage.

If the external clock (CGMXCLK) is equal to or greater than 1MHz, CGMXCLK can be used as the clock source for the ADC. If CGMXCLK is less than 1MHz, use the PLL-generated bus clock as the clock source. As long as the internal ADC clock is at f_{ADIC}, correct operation can be guaranteed.

1 = Internal bus clock
0 = External clock, CGMXCLK

$$f_{ADIC} = \frac{\text{CGMXCLK or bus frequency}}{\text{ADIV[2:0]}}$$

MODE1 and MODE0 — Modes of Result Justification

MODE1 and MODE0 selects between four modes of operation. The manner in which the ADC conversion results will be placed in the ADC data registers is controlled by these modes of operation. Reset returns right-justified mode.

Table 16-3. ADC Mode Select

MODE1	MODE0	ADC Clock Rate
0	0	8-bit truncated mode
0	1	Right justified mode
1	0	Left justified mode
1	1	Left justified sign data mode

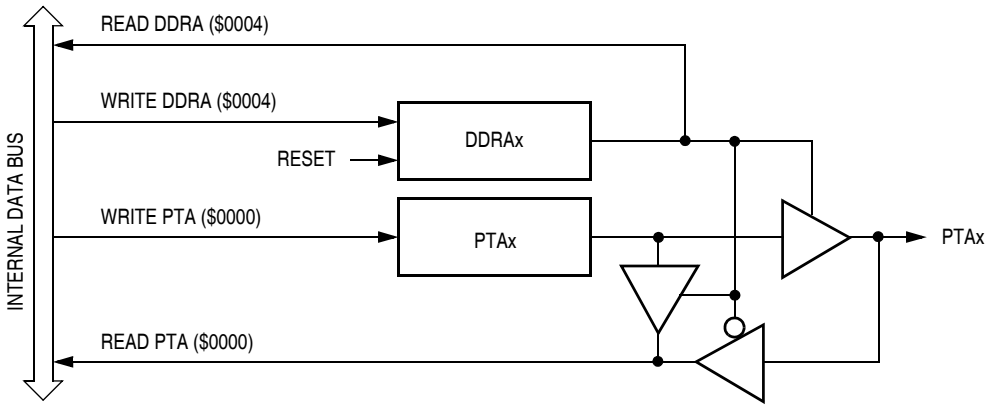


Figure 18-4. Port A I/O Circuit

When DDRAx is a logic 1, reading address $0000 reads the PTAx data latch. When DDRAx is a logic 0, reading address $0000 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit.

Table 18-2 summarizes the operation of the port A pins.

Table 18-2. Port A Pin Functions

DDRA Bit	PTA Bit	I/O Pin Mode	Accesses to DDRA	Accesses to PTA	
			Read/Write	Read	Write
0	X(1)	Input, Hi-Z(2)	DDRA[7:0]	Pin	PTA[7:0](3)
1	X	Output	DDRA[7:0]	PTA[7:0]	PTA[7:0]

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

When DDREx is a logic 1, reading address $0008 reads the PTEx data latch. When DDREx is a logic 0, reading address $0008 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit.

Table 18-6 summarizes the operation of the port E pins.

Table 18-6. Port E Pin Functions

DDRE Bit	PTE Bit	I/O Pin Mode	Accesses to DDRE	Accesses to PTE	
			Read/Write	Read	Write
0	X(1)	Input, Hi-Z(2)	DDRE[7:0]	Pin	PTE[7:0](3)
1	X	Output	DDRE[7:0]	PTE[7:0]	PTE[7:0]

Notes:
1. X = don't care; except.
2. Hi-Z = high impedance.
3. Writing affects data register, but does not affect input.

18.7.3 Port E LED Control Register (LEDE)

The port-E LED control register (LEDE) controls the direct LED drive capability on PTE7–PTE0 pins. Each bit is individually configurable and requires that the data direction register, DDRE, bit be configured as an output.

Address: $000E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LEDE7	LEDE6	LEDE5	LEDE4	LEDE3	LEDE2	LEDE1	LEDE0
Reset:	0	0	0	0	0	0	0	0

Figure 18-19. Port E LED Control Register (LEDE)

LEDE[7:0] — Port E 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 E pin is configured for direct LED drive, with 15 mA current sinking capability

0 = Corresponding port E pin is configured for standard drive

Section 20. Keyboard Interrupt Module (KBI)

20.1 Contents

20.2 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 (except PTD6 and PTD7), an internal 30kΩ pullup device is also enabled on the pin.

NOTE: *PTD6/KBI6/SCL–PTD7/KBI7/SDA pins do not have internal pullup devices. These two pins are open-drain when configured as outputs. User should connect pullup devices when using these two pins for KBI function.*