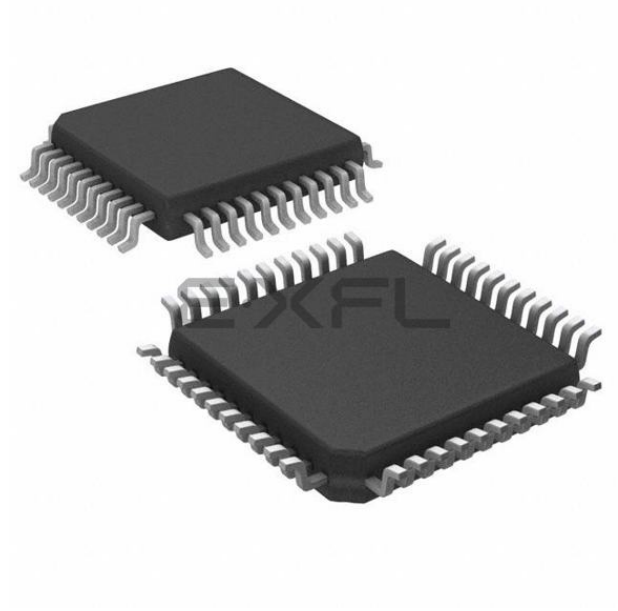


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Not For New Designs |
| Core Processor | HC08 |
| Core Size | 8-Bit |
| Speed | 8MHz |
| Connectivity | 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) | 4.5V ~ 5.5V |
| Data Converters | A/D 8x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-QFP |
| Supplier Device Package | 44-QFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908ap32acfber |

# Revision History

| Date | Revision Level | Description | Page Number(s) |
|---|---|---|---|
| October 2007 | 3 | Clarified or updated information in Chapter 7 System Integration Module (SIM).<br>In Chapter 8 Monitor Mode (MON), corrected and updated monitor mode entry details.<br>In Chapter 9 Timer Interface Module (TIM), updated functional details.<br>In 15.7 I/O Registers, corrected COCO bit description.<br>In 16.1 Introduction, added unused pin information.<br>In Chapter 21 Break Module (BRK), updated functional details.<br>In 22.5 5V DC Electrical Characteristics, updated stop $I_{DD}$ values. | — |
| January 2007 | 2 | 15.7.2 ADC Clock Control Register — Changed “The ADC clock should be set to between 500 kHz and 2 MHz” to “The ADC clock should be set to between 500 kHz and 1 MHz” | 250 |
| Mar 2005 | 1 | First general release. | — |

## Chapter 21
## Break Module (BRK)



## Chapter 22
## Electrical Specifications

## Chapter 23
## Mechanical Specifications



## Chapter 24
## Ordering Information

## 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 |

## 6.8 Acquisition/Lock Time Specifications

The acquisition and lock times of the PLL are, in many applications, the most critical PLL design parameters. Proper design and use of the PLL ensures the highest stability and lowest acquisition/lock times.

### 6.8.1 Acquisition/Lock Time Definitions

Typical control systems refer to the acquisition time or lock time as the reaction time, within specified tolerances, of the system to a step input. In a PLL, the step input occurs when the PLL is turned on or when it suffers a noise hit. The tolerance is usually specified as a percent of the step input or when the output settles to the desired value plus or minus a percent of the frequency change. Therefore, the reaction time is constant in this definition, regardless of the size of the step input. For example, consider a system with a 5 percent acquisition time tolerance. If a command instructs the system to change from 0 Hz to 1 MHz, the acquisition time is the time taken for the frequency to reach 1 MHz ±50 kHz. 50 kHz = 5% of the 1 MHz step input. If the system is operating at 1 MHz and suffers a –100 kHz noise hit, the acquisition time is the time taken to return from 900 kHz to 1 MHz ±5 kHz. 5 kHz = 5% of the 100 kHz step input.

Other systems refer to acquisition and lock times as the time the system takes to reduce the error between the actual output and the desired output to within specified tolerances. Therefore, the acquisition or lock time varies according to the original error in the output. Minor errors may not even be registered. Typical PLL applications prefer to use this definition because the system requires the output frequency to be within a certain tolerance of the desired frequency regardless of the size of the initial error.

### 6.8.2 Parametric Influences on Reaction Time

Acquisition and lock times are designed to be as short as possible while still providing the highest possible stability. These reaction times are not constant, however. Many factors directly and indirectly affect the acquisition time.

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 6.3.3 PLL Circuits, 6.3.6 Programming the PLL, and 6.5.5 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 6.8.3 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.

### 6.8.3 Choosing a Filter

As described in 6.8.2 Parametric Influences on Reaction Time, the external filter network is critical to the stability and reaction time of the PLL. The PLL is also dependent on reference frequency and supply voltage.

Either of the filter networks in Figure 6-10 is recommended when using a 4MHz reference clock (CGMRCLK). Figure 6-10 **(a)** is used for applications requiring better stability. Figure 6-10 **(b)** is used in low-cost applications where stability is not critical.

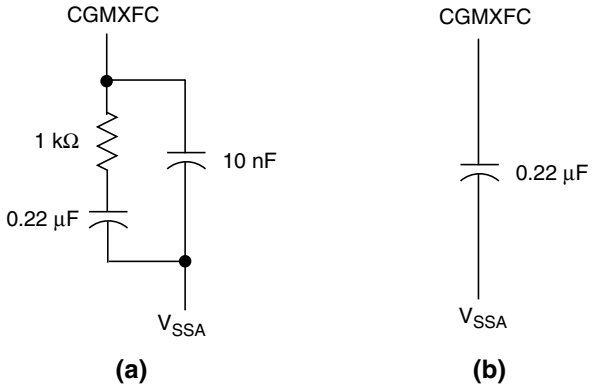



**Figure 6-10. PLL Filter**

At power-on, these events occur:

- A POR pulse is generated.
- The internal reset signal is asserted.
- The SIM enables CGMOUT.
- Internal clocks to the CPU and modules are held inactive for 4096 ICLK cycles to allow stabilization of the oscillator.
- The  pin is driven low during the oscillator stabilization time.
- The POR bit of the SIM reset status register (SRSR) is set and all other bits in the register are cleared.

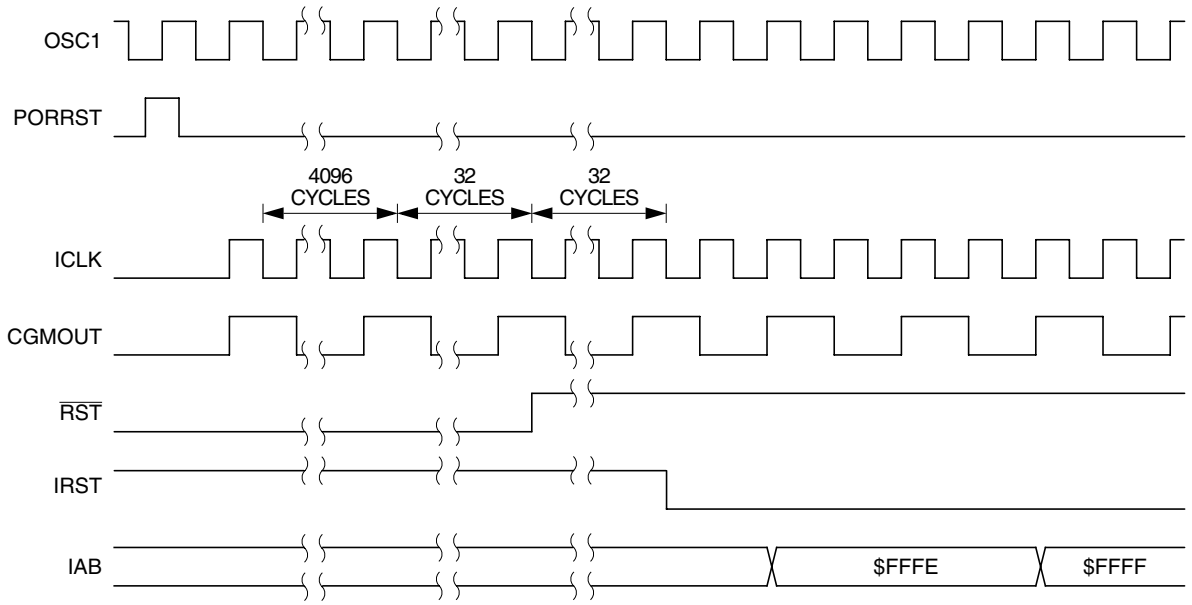



**Figure 7-7. POR Recovery**

### *7.3.2.2 Computer Operating Properly (COP) Reset*

An input to the SIM is reserved for the COP reset signal. The overflow of the COP counter causes an internal reset and sets the COP bit in the SIM reset status register (SRSR). The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

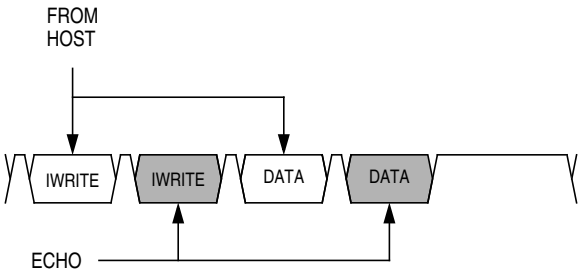
To prevent a COP module timeout, write any value to location $FFFF. Writing to location $FFFF clears the COP counter and bits 12 through 5 of the SIM counter. The SIM counter output, which occurs at least every $2^{13} - 2^4$ ICLK cycles, drives the COP counter. The COP should be serviced as soon as possible out of reset to guarantee the maximum amount of time before the first timeout.

The COP module is disabled if the $\overline{\text{RST}}$ pin or the $\overline{\text{IRQ1}}$ pin is held at $V_{TST}$ while the MCU is in monitor mode. The COP module can be disabled only through combinational logic conditioned with the high voltage signal on the $\overline{\text{RST}}$ or the $\overline{\text{IRQ1}}$ pin. This prevents the COP from becoming disabled as a result of external noise. During a break state, $V_{TST}$ on the $\overline{\text{RST}}$ pin disables the COP module.

### *7.3.2.3 Illegal Opcode Reset*

The SIM decodes signals from the CPU to detect illegal instructions. An illegal instruction sets the ILOP bit in the SIM reset status register (SRSR) and causes a reset.

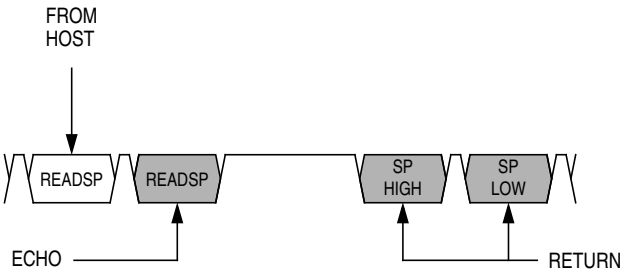
**Table 8-7. IWRITE (Indexed Write) Command**

| | |
|---|---|
| **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 8-8. 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** |  |

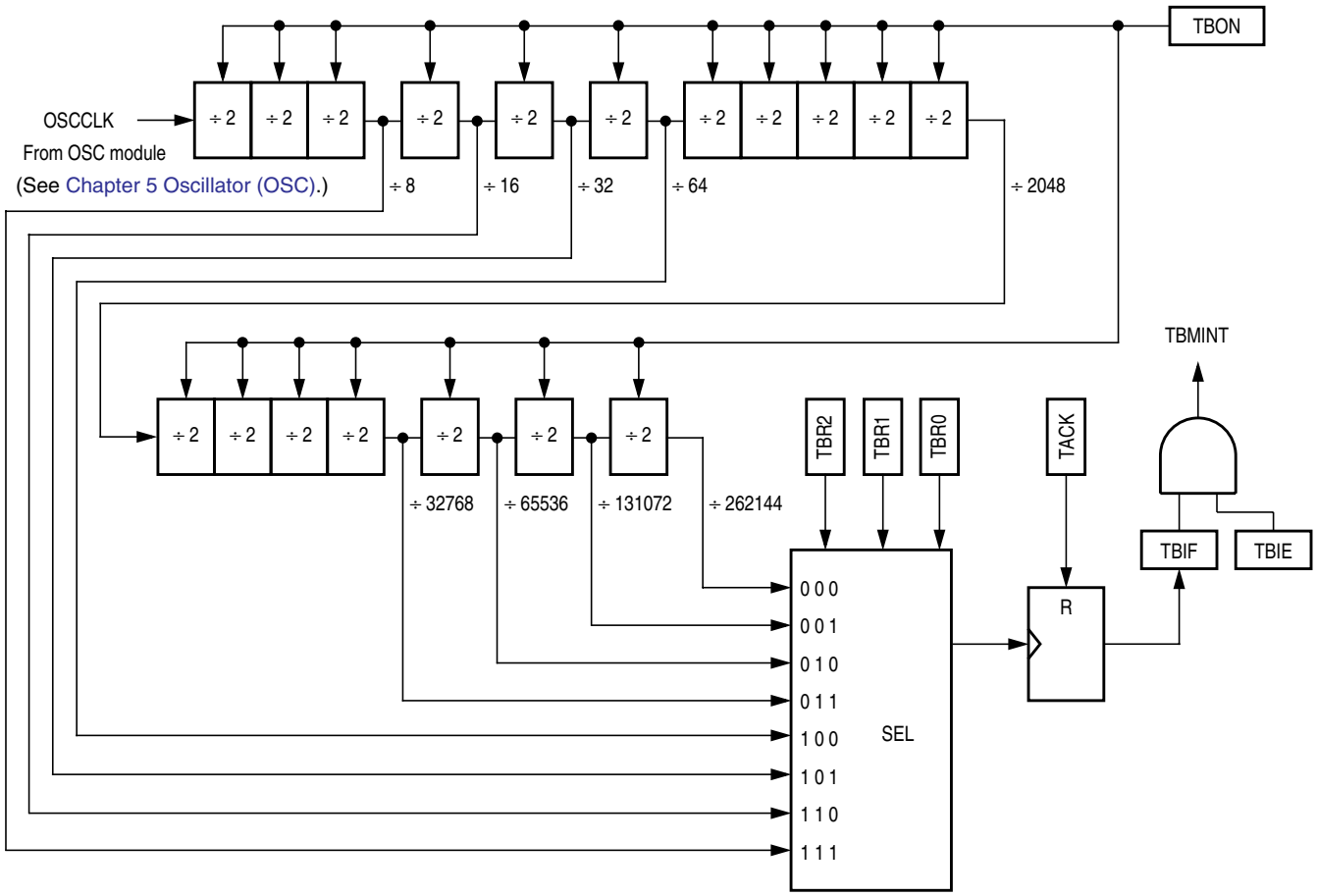



**Figure 10-1. Timebase Block Diagram**

## 10.4 Timebase Register Description

The timebase has one register, the TBCR, which is used to enable the timebase interrupts and set the rate.

Address: $0051

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read: | TBIF | TBR2 | TBR1 | TBR0 | 0 | TBIE | TBON | R |
| Write: | | | | | TACK | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented  R = Reserved

**Figure 10-2. Timebase Control Register (TBCR)**

**TBIF — Timebase Interrupt Flag**

This read-only flag bit is set when the timebase counter has rolled over.

1 = Timebase interrupt pending
0 = Timebase interrupt not pending

### 11.4.3.3 Data Sampling

The receiver samples the RxD pin at the RT clock rate. The RT clock is an internal signal with a frequency 16 times the baud rate. To adjust for baud rate mismatch, the RT clock is resynchronized at the following times (see Figure 11-6):

- After every start bit
- After the receiver detects a data bit change from logic 1 to logic 0 (after the majority of data bit samples at RT8, RT9, and RT10 returns a valid logic 1 and the majority of the next RT8, RT9, and RT10 samples returns a valid logic 0)

To locate the start bit, data recovery logic does an asynchronous search for a logic 0 preceded by three logic 1s. When the falling edge of a possible start bit occurs, the RT clock begins to count to 16.

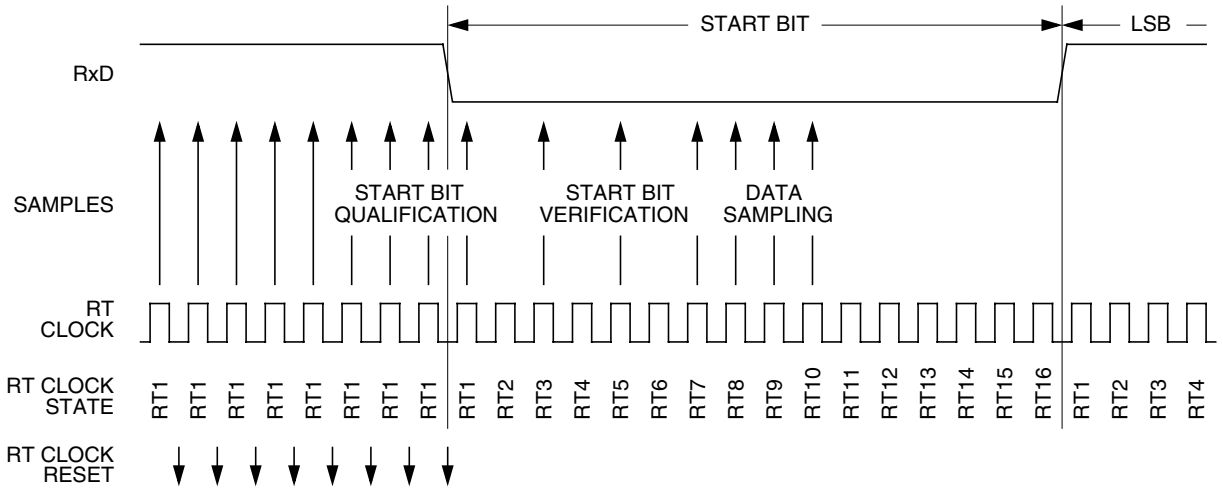



**Figure 11-6. Receiver Data Sampling**

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7. Table 11-2 summarizes the results of the start bit verification samples.

**Table 11-2. Start Bit Verification**

| RT3, RT5, and RT7 Samples | Start Bit Verification | Noise Flag |
|---|---|---|
| 000 | Yes | 0 |
| 001 | Yes | 1 |
| 010 | Yes | 1 |
| 011 | No | 0 |
| 100 | Yes | 1 |
| 101 | No | 0 |
| 110 | No | 0 |
| 111 | No | 0 |

### 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:<br>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

- 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.

## 14.3 I/O Pins

The MMIIC module uses two I/O pins, shared with standard port I/O pins. The full name of the MMIIC I/O pins are listed in Table 14-1. The generic pin name appear in the text that follows.

The SDA and SDL pins are open-drain. When configured as general purpose output pins (PTB0 and PTB1), pullup resistors must be connected to these pins.

**Table 14-1. Pin Name Conventions**

| MMIIC Generic Pin Names: | Full MCU Pin Names: | Pin Selected for MMIIC Function By: |
|---|---|---|
| SDA | PTB0/SDA | MMEN bit in MMCR1 ($0049) |
| SCL | PTB1/SCL | |

| Addr. | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $0048 | MMIIC Address Register (MMADR) | Read: | MMAD7 | MMAD6 | MMAD5 | MMAD4 | MMAD3 | MMAD2 | MMAD1 | MMEXTAD |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $0049 | MMIIC Control Register 1 (MMCR1) | Read: | MMEN | MMIEN | 0 | 0 | MMTXAK | REPSEN | MMCRCBYTE | 0 |
| | | Write: | | | MMCLRBB | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $004A | MMIIC Control Register 2 (MMCR2) | Read: | MMALIF | MMNAKIF | MMBB | MMAST | MMRW | 0 | 0 | MMCRCEF |
| | | Write: | 0 | 0 | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Unaffected |
| $004B | MMIIC Status Register (MMSR) | Read: | MMRXIF | MMTXIF | MMATCH | MMSRW | MMRXAK | MMCRCBF | MMTXBE | MMRXBF |
| | | Write: | 0 | 0 | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| $004C | MMIIC Data Transmit Register (MMDTR) | Read: | MMTD7 | MMTD6 | MMTD5 | MMTD4 | MMTD3 | MMTD2 | MMTD1 | MMTD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $004D | MMIIC Data Receive Register (MDDRR) | Read: | MMRD7 | MMRD6 | MMRD5 | MMRD4 | MMRD3 | MMRD2 | MMRD1 | MMRD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $004E | MMIIC CRC Data Register (MMCRDR) | Read: | MMCRCD7 | MMCRCD6 | MMCRCD5 | MMCRCD4 | MMCRCD3 | MMCRCD2 | MMCRCD1 | MMCRCD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $004F | MMIIC Frequency Divider Register (MMFDR) | Read: | 0 | 0 | 0 | 0 | 0 | MMBR2 | MMBR1 | MMBR0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

= Unimplemented

**Figure 14-1. MMIIC I/O Register Summary**

## 14.4 Multi-Master IIC System Configuration

The multi-master IIC system uses a serial data line SDA and a serial clock line SCL for data transfer. All devices connected to it must have open collector (drain) outputs and the logical-AND function is performed on both lines by two pull-up resistors.

### ASCAN — Auto-scan Mode Enable Bit

This bit enable/disable the auto-scan mode. Reset clears this bit.

1 = Auto-scan mode is enabled
0 = Auto-scan mode is disabled

Auto-scan mode should not be enabled when ADC continuous conversion is enabled; i.e. when ADCO=1.

### DDRA[7:0] — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA[7:0], configuring all port A pins as inputs.

1 = Corresponding port A pin configured as output
0 = Corresponding port A pin configured as input

***NOTE***

*Avoid glitches on port A pins by writing to the port A data register before changing data direction register A bits from 0 to 1.*

Figure 16-4 shows the port A I/O logic.

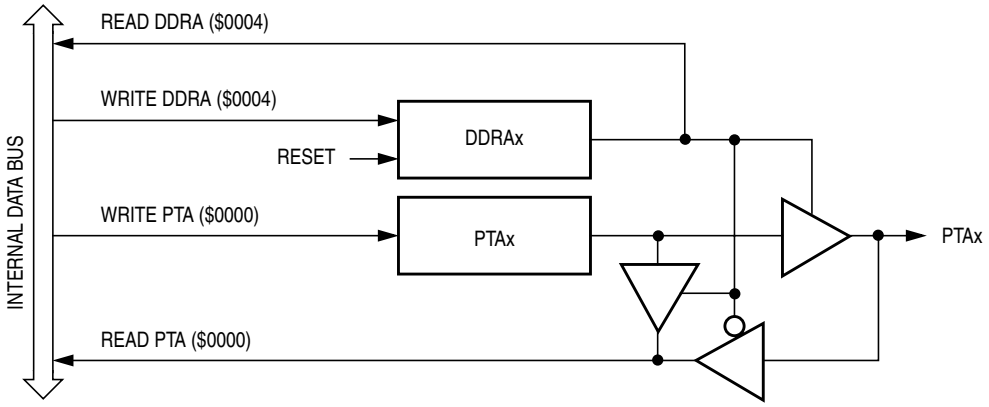



**Figure 16-4. Port A I/O Circuit**

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 16-2 summarizes the operation of the port A pins.

**Table 16-2. 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] |

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

## 18.4 Functional Description

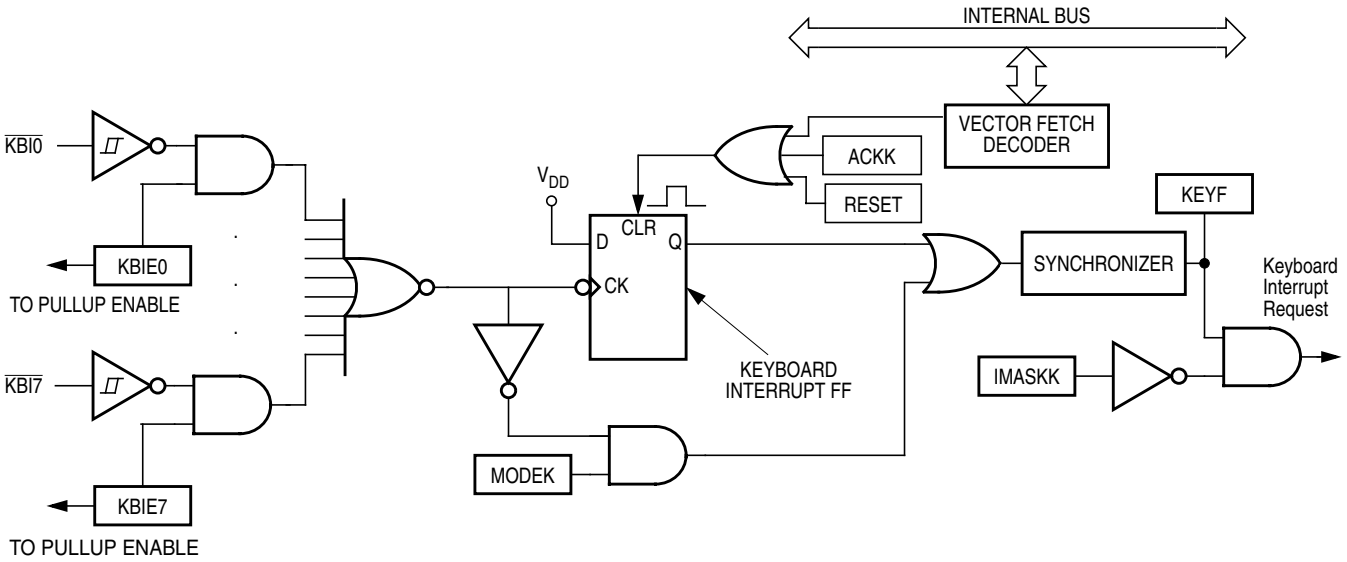



**Figure 18-2. Keyboard Interrupt Block Diagram**

Writing to the KBIE7–KBIE0 bits in the keyboard interrupt enable register independently enables or disables each port D pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin in port D also enables its internal pull-up device. A logic 0 applied to an enabled keyboard interrupt pin latches a keyboard interrupt request.

A keyboard interrupt is latched when one or more keyboard pins goes low after all were high. The MODEK bit in the keyboard status and control register controls the triggering mode of the keyboard interrupt.

- If the keyboard interrupt is edge-sensitive only, a falling edge on a keyboard pin does not latch an interrupt request if another keyboard pin is already low. To prevent losing an interrupt request on one pin because another pin is still low, software can disable the latter pin while it is low.
- If the keyboard interrupt is falling edge- and low level-sensitive, an interrupt request is present as long as any keyboard pin is low.

If the MODEK bit is set, the keyboard interrupt pins are both falling edge- and low level-sensitive, and both of the following actions must occur to clear a keyboard interrupt request:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the interrupt request. Software may generate the interrupt acknowledge signal by writing a logic 1 to the ACKK bit in the keyboard status and control register KBSCR. The ACKK bit is useful in applications that poll the keyboard interrupt pins and require software to clear the keyboard interrupt request. Writing to the ACKK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACKK does not affect subsequent transitions on the keyboard interrupt pins. A falling edge that occurs after writing to the ACKK bit latches another interrupt request. If the keyboard interrupt mask bit, IMASKK, is clear, the CPU loads the program counter with the vector address at locations $FFE0 and $FFE1.
- Return of all enabled keyboard interrupt pins to logic 1 — As long as any enabled keyboard interrupt pin is at logic 0, the keyboard interrupt remains set.

The vector fetch or software clear and the return of all enabled keyboard interrupt pins to logic 1 may occur in any order.

If the MODEK bit is clear, the keyboard interrupt pin is falling-edge-sensitive only. With MODEK clear, a vector fetch or software clear immediately clears the keyboard interrupt request.

Reset clears the keyboard interrupt request and the MODEK bit, clearing the interrupt request even if a keyboard interrupt pin stays at logic 0.

The keyboard flag bit (KEYF) in the keyboard status and control register can be used to see if a pending interrupt exists. The KEYF bit is not affected by the keyboard interrupt mask bit (IMASKK) which makes it useful in applications where polling is preferred.

To determine the logic level on a keyboard interrupt pin, use the data direction register to configure the pin as an input and read the data register.

***NOTE***

*Setting a keyboard interrupt enable bit (KBIEx) forces the corresponding keyboard interrupt pin to be an input, overriding the data direction register. However, the data direction register bit must be a logic 0 for software to read the pin.*

### 18.4.1 Keyboard Initialization

When a keyboard interrupt pin is enabled, it takes time for the internal pull-up to reach a logic 1. Therefore a false interrupt can occur as soon as the pin is enabled.

To prevent a false interrupt on keyboard initialization:

1. Mask keyboard interrupts by setting the IMASKK bit in the keyboard status and control register.
2. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.
3. Write to the ACKK bit in the keyboard status and control register to clear any false interrupts.
4. Clear the IMASKK bit.

An interrupt signal on an edge-triggered pin can be acknowledged immediately after enabling the pin. An interrupt signal on an edge- and level-triggered interrupt pin must be acknowledged after a delay that depends on the external load.

Another way to avoid a false interrupt:

1. Configure the keyboard pins as outputs by setting the appropriate DDR bits in data direction register.
2. Write logic 1s to the appropriate data register bits.
3. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.

## 18.5 Keyboard Interrupt Registers

Two registers control the operation of the keyboard interrupt module:

- Keyboard Status and Control Register — $001A
- Keyboard Interrupt Enable Register — $001B

### 18.5.1 Keyboard Status and Control Register

- Flags keyboard interrupt requests
- Acknowledges keyboard interrupt requests
- Masks keyboard interrupt requests
- Controls keyboard interrupt triggering sensitivity

## 22.10 CGM Electrical Specification

**Table 22-10. CGM Electrical Specifications**

| Characteristic | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Reference frequency | $f_{RDV}$ | 1 | — | 8 | MHz |
| Range nominal multiplies | $f_{NOM}$ | — | 125 | — | kHz |
| VCO center-of-range frequency | $f_{VRS}$ | 125k | — | 40M | Hz |
| VCO range linear range multiplier | L | 1 | — | 255 | |
| VCO power-of-two-range multiplier | $2^E$ | 1 | — | 4 | |
| VCO multiply factor | N | 1 | — | 4095 | |
| VCO prescale multiplier | $2^P$ | 1 | — | 8 | |
| Reference divider factor | R | 1 | 1 | 15 | |
| VCO operating frequency | $f_{VCLK}$ | 125k | — | 40M | Hz |
| Manual acquisition time | $t_{LOCK}$ | — | — | 50 | ms |
| Automatic lock time | $t_{LOCK}$ | — | — | 50 | ms |
| PLL jitter[1] | $f_J$ | 0 | — | $f_{RCLK} \times 0.025\% \times 2^P N/4$ | Hz |

1. Deviation of average bus frequency over 2ms. N = VCO multiplier.