

Welcome to [E-XFL.COM](https://www.e-xfl.com)

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

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

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

#### Details

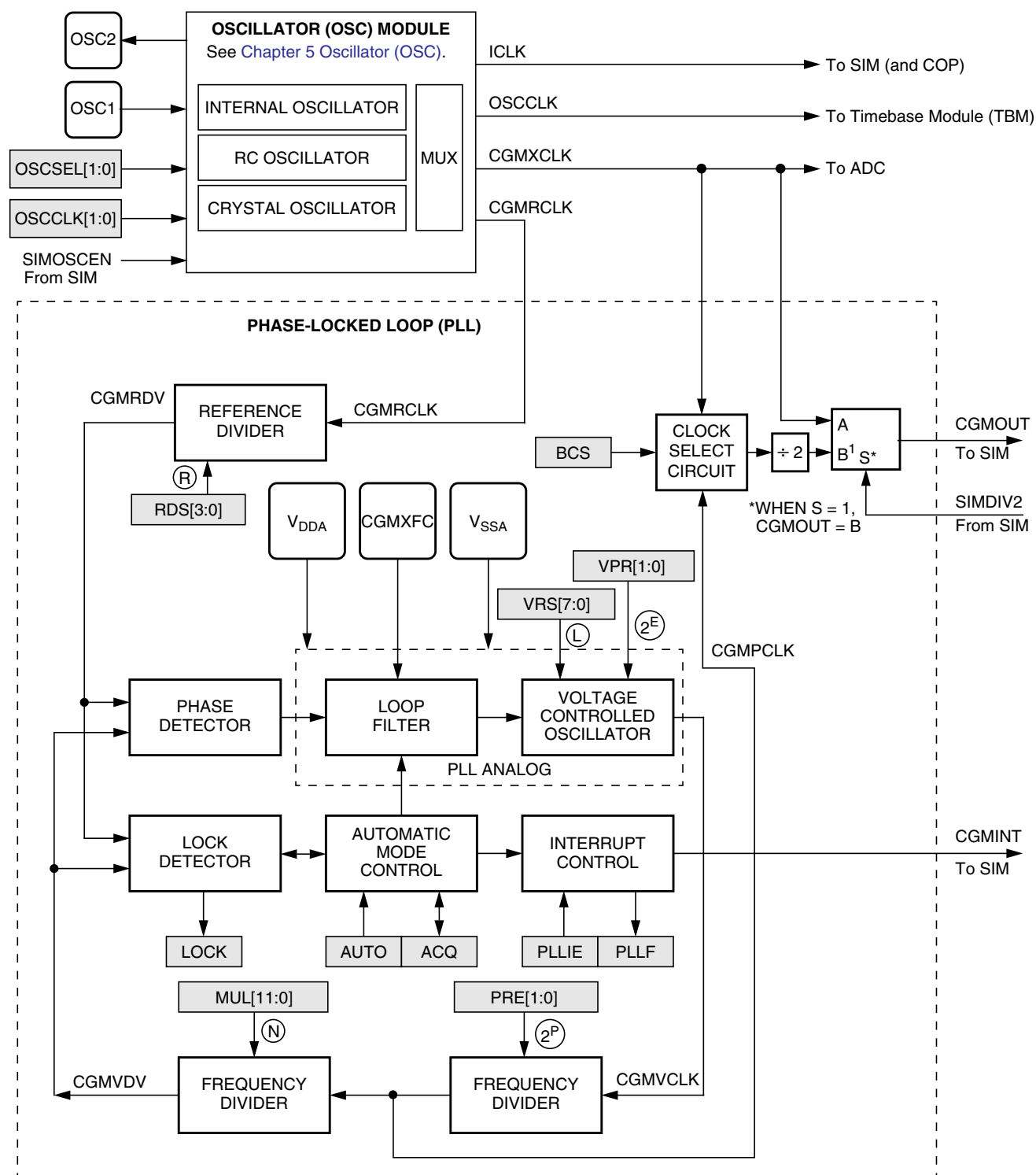
Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	I <sup>2</sup> C, IRSCI, SCI, SPI
Peripherals	LED, LVD, POR, PWM
Number of I/O	32
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	42-SDIP (0.600", 15.24mm)
Supplier Device Package	42-PDIP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908ap32cb">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908ap32cb</a>

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$0033	Timer 2 Channel 1 Status and Control Register (T2SC1)	Read: CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write: 0							
		Reset: 0	0	0	0	0	0	0	0
\$0034	Timer 2 Channel 1 Register High (T2CH1H)	Read: Bit 15	14	13	12	11	10	9	Bit 8
		Write:							
		Reset:	Indeterminate after reset						
\$0035	Timer 2 Channel 1 Register Low (T2CH1L)	Read: Bit 7	6	5	4	3	2	1	Bit 0
		Write:							
		Reset:	Indeterminate after reset						
\$0036	PLL Control Register (PCTL)	Read: PLLIE	PLL F	PLL ON	BCS	PRE1	PRE0	VPR1	VPR0
		Write:							
		Reset: 0	0	1	0	0	0	0	0
\$0037	PLL Bandwidth Control Register (PBWC)	Read: AUTO	LOCK	ACQ	0	0	0	0	R
		Write:							
		Reset: 0	0	0	0	0	0	0	0
\$0038	PLL Multiplier Select Register High (PMSH)	Read: 0	0	0	0	MUL11	MUL10	MUL9	MUL8
		Write:							
		Reset: 0	0	0	0	0	0	0	0
\$0039	PLL Multiplier Select Register Low (PMSL)	Read: MUL7	MUL6	MUL5	MUL4	MUL3	MUL2	MUL1	MUL0
		Write:							
		Reset: 0	1	0	0	0	0	0	0
\$003A	PLL VCO Range Select Register (PMRS)	Read: VRS7	VRS6	VRS5	VRS4	VRS3	VRS2	VRS1	VRS0
		Write:							
		Reset: 0	1	0	0	0	0	0	0
\$003B	PLL Reference Divider Select Register (PMDS)	Read: 0	0	0	0	RDS3	RDS2	RDS1	RDS0
		Write:							
		Reset: 0	0	0	0	0	0	0	1
\$003C	Unimplemented	Read:							
		Write:							
		Reset:							
\$003D	Unimplemented	Read:							
		Write:							
		Reset:							
\$003E	Unimplemented	Read:							
		Write:							
		Reset:							
\$003F	Unimplemented	Read:							
		Write:							
		Reset:							

U = Unaffected      X = Indeterminate      = Unimplemented      R = Reserved

**Figure 2-2. Control, Status, and Data Registers (Sheet 5 of 9)**

### Clock Generator Module (CGM)



### Figure 6-1. CGM Block Diagram

register disables the PLL and clears the BCS bit in the PLL control register (PCTL). (See [6.3.8 Base Clock Selector Circuit](#) and [6.3.7 Special Programming Exceptions](#).) Reset initializes the register to \$40 for a default range multiply value of 64.

#### NOTE


*The VCO range select bits have built-in protection such that they cannot be written when the PLL is on (PLLON = 1) and such that the VCO clock cannot be selected as the source of the base clock (BCS = 1) if the VCO range select bits are all clear.*

*The PLL VCO range select register must be programmed correctly. Incorrect programming can result in failure of the PLL to achieve lock.*

### 6.5.5 PLL Reference Divider Select Register

The PLL reference divider select register (PMDS) contains the programming information for the modulo reference divider.

Address:	\$003B							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	RDS3	RDS2	RDS1	RDS0
Write:								
Reset:	0	0	0	0	0	0	0	1

 = Unimplemented

**Figure 6-9. PLL Reference Divider Select Register (PMDS)**

#### RDS[3:0] — Reference Divider Select Bits

These read/write bits control the modulo reference divider that selects the reference division factor, R. (See [6.3.3 PLL Circuits](#) and [6.3.6 Programming the PLL](#).) RDS[3:0] cannot be written when the PLLON bit in the PCTL is set. A value of \$00 in the reference divider select register configures the reference divider the same as a value of \$01. (See [6.3.7 Special Programming Exceptions](#).) Reset initializes the register to \$01 for a default divide value of 1.

#### NOTE

*The reference divider select bits have built-in protection such that they cannot be written when the PLL is on (PLLON = 1).*

#### NOTE

*The default divide value of 1 is recommended for all applications.*

## 6.6 Interrupts

When the AUTO bit is set in the PLL bandwidth control register (PBWC), the PLL can generate a CPU interrupt request every time the LOCK bit changes state. The PLLIE bit in the PLL control register (PCTL) enables CPU interrupts from the PLL. PLLF, the interrupt flag in the PCTL, becomes set whether interrupts are enabled or not. When the AUTO bit is clear, CPU interrupts from the PLL are disabled and PLLF reads as logic 0.

Software should read the LOCK bit after a PLL interrupt request to see if the request was due to an entry into lock or an exit from lock. When the PLL enters lock, the divided VCO clock, CGMPCLK, divided by two can be selected as the CGMOUT source by setting BCS in the PCTL. When the PLL exits lock, the

\$FE04	Interrupt Status Register 1 (INT1)	Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE05	Interrupt Status Register 2 (INT2)	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE06	Interrupt Status Register 3 (INT3)	Read:	0	IF21	IF20	IF19	IF18	IF17	IF16	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figure 7-2. SIM I/O Register Summary

## 7.2 SIM Bus Clock Control and Generation

The bus clock generator provides system clock signals for the CPU and peripherals on the MCU. The system clocks are generated from an incoming clock, CGMOUT, as shown in Figure 7-3. This clock can come from either an external oscillator or from the on-chip PLL. (See Chapter 6 Clock Generator Module (CGM).)

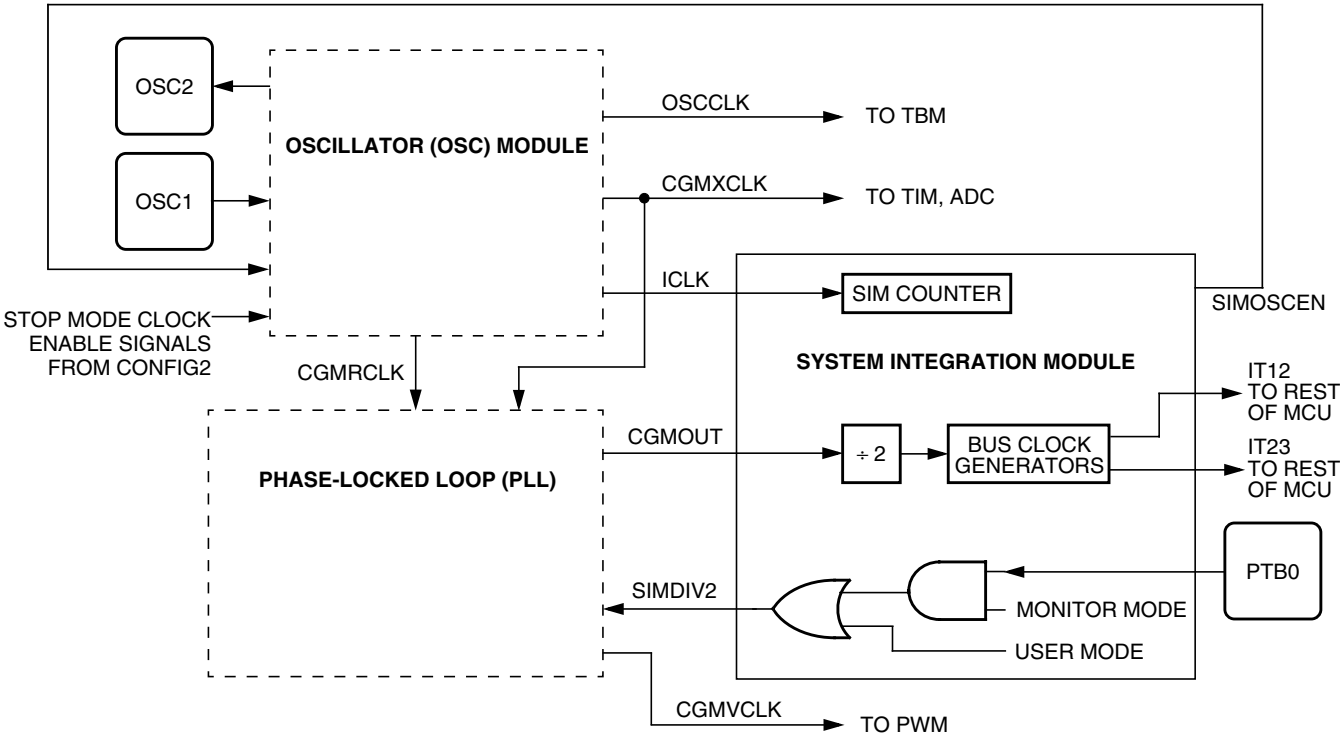


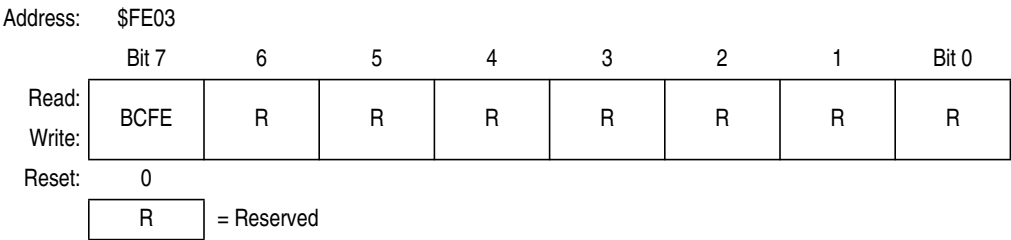
Figure 7-3. CGM Clock Signals

### 7.2.1 Bus Timing

In user mode, the internal bus frequency is either the oscillator output (CGMXCLK) divided by four or the divided PLL output (CGMPCLK) divided by four.

### 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

## Chapter 8

# Monitor ROM (MON)

### 8.1 Introduction

This section describes the monitor ROM (MON) and the monitor mode entry methods. The monitor ROM allows complete testing of the MCU through a single-wire interface with a host computer. Monitor mode entry can be achieved without use of the higher test voltage,  $V_{TST}$ , as long as vector addresses \$FFFE and \$FFFF are blank, thus reducing the hardware requirements for in-circuit programming. In addition, to simplify user coding, routines are also stored in the monitor ROM area for FLASH memory program /erase.

### 8.2 Features

Features of the monitor ROM include:

- Normal user-mode pin functionality
- One pin dedicated to serial communication between monitor ROM and host computer
- Standard mark/space non-return-to-zero (NRZ) communication with host computer
- Execution of code in RAM or FLASH
- FLASH memory security feature<sup>(1)</sup>
- 959 bytes monitor ROM code size (\$FC00–\$FDFF and \$FE10–\$FFCE)
- Monitor mode entry without high voltage,  $V_{TST}$ , if reset vector is blank (\$FFFE and \$FFFF contain \$FF)
- Standard monitor mode entry if high voltage,  $V_{TST}$ , is applied to  $\overline{IRQ1}$
- Resident routines for in-circuit programming

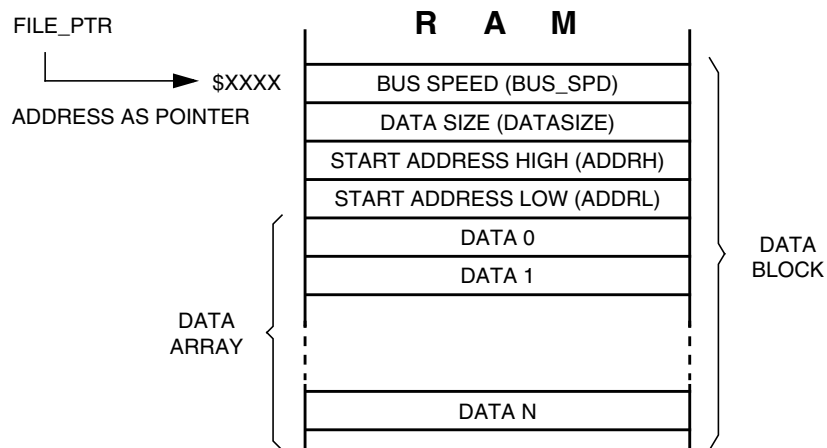
### 8.3 Functional Description

The monitor ROM receives and executes commands from a host computer. [Figure 8-1](#) shows an example circuit used to enter monitor mode and communicate with a host computer via a standard RS-232 interface.

Simple monitor commands can access any memory address. In monitor mode, the MCU can execute code downloaded into RAM by a host computer while most MCU pins retain normal operating mode functions. All communication between the host computer and the MCU is through the PTA0 pin. A level-shifting and multiplexing interface is required between PTA0 and the host computer. PTA0 is used in a wired-OR configuration and requires a pullup resistor.

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.

During the software execution, it does not consume any dedicated RAM location, the run-time heap will extend the system stack, all other RAM location will not be affected.



**Figure 8-9. Data Block Format for ROM-Resident Routines**

The control and data bytes are described below.

- **Bus speed** — This one byte indicates the operating bus speed of the MCU. The value of this byte should be equal to 4 times the bus speed. E.g., for a 4MHz bus, the value is 16 (\$10). This control byte is useful where the MCU clock source is switched between the PLL clock and the crystal clock.
- **Data size** — This one byte indicates the number of bytes in the data array that are to be manipulated. The maximum data array size is 255. Routines EE\_WRITE and EE\_READ are restricted to manipulate a data array between 7 to 15 bytes. Whereas routines ERARNGE and MON\_ERARNGE do not manipulate a data array, thus, this data size byte has no meaning.
- **Start address** — These two bytes, high byte followed by low byte, indicate the start address of the FLASH memory to be manipulated.
- **Data array** — This data array contains data that are to be manipulated. Data in this array are programmed to FLASH memory by the programming routines: PRGRNGE, MON\_PRGRNGE, EE\_WRITE. For the read routines: LDRNGE and EE\_READ, data is read from FLASH and stored in this array.

### 8.5.1 PRGRNGE

PRGRNGE is used to program a range of FLASH locations with data loaded into the data array.

**Table 8-11. PRGRNGE Routine**

<b>Routine Name</b>	PRGRNGE
<b>Routine Description</b>	Program a range of locations
<b>Calling Address</b>	\$FC34
<b>Stack Used</b>	15 bytes
<b>Data Block Format</b>	Bus speed (BUS_SPD) Data size (DATASIZE) Start address high (ADDRH) Start address (ADDRL) Data 1 (DATA1) : Data N (DATAN)



Address: T1CH0H, \$0026 and T2CH0H, \$0031

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	14	13	12	11	10	9	Bit 8
Write:								
Reset:	Indeterminate after reset							

**Figure 9-12. TIM Channel 0 Register High (TCH0H)**

Address: T1CH0L, \$0027 and T2CH0L, \$0032

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

**Figure 9-13. TIM Channel 0 Register Low (TCH0L)**

Address: T1CH1H, \$0029 and T2CH1H, \$0034

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	14	13	12	11	10	9	Bit 8
Write:								
Reset:	Indeterminate after reset							

**Figure 9-14. TIM Channel 1 Register High (TCH1H)**

Address: T1CH1L, \$002A and T2CH1L, \$0035

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

**Figure 9-15. TIM Channel 1 Register Low (TCH1L)**

# Timebase Module (TBM)

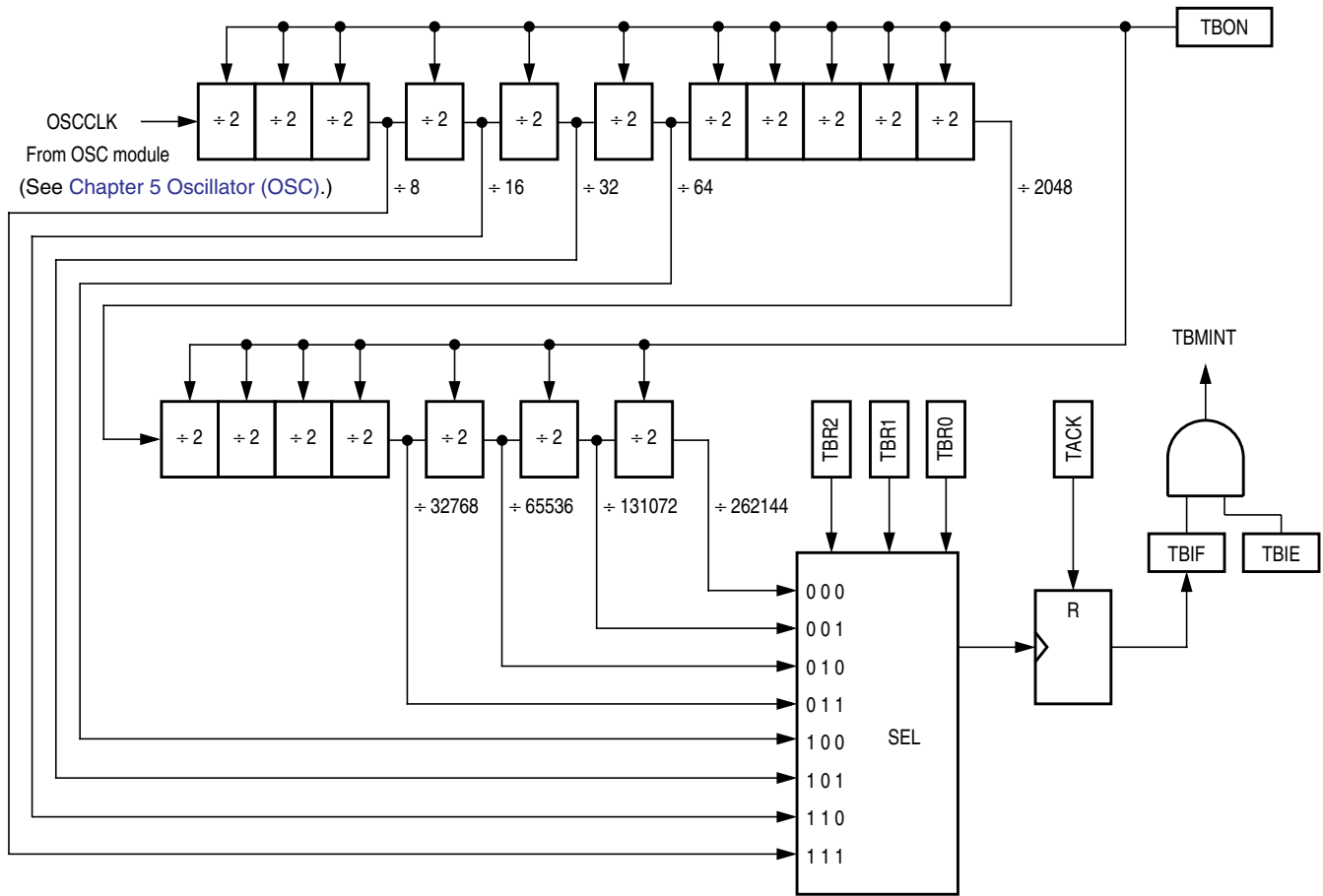


Figure 10-1. Timebase Block Diagram

## 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

## 10.5 Interrupts

The timebase module can interrupt the CPU on a regular basis with a rate defined by TBR[2:0]. When the timebase counter chain rolls over, the TBIF flag is set. If the TBIE bit is set, enabling the timebase interrupt, the counter chain overflow will generate a CPU interrupt request. The interrupt vector is defined in [Table 2-1 . Vector Addresses](#).

Interrupts must be acknowledged by writing a logic 1 to the TACK bit.

## 10.6 Low-Power Modes

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

### 10.6.1 Wait Mode

The timebase module remains active after execution of the WAIT instruction. In wait mode, the timebase register is not accessible by the CPU.

If the timebase functions are not required during wait mode, reduce the power consumption by stopping the timebase before enabling the WAIT instruction.

### 10.6.2 Stop Mode

The timebase module may remain active after execution of the STOP instruction if the oscillator has been enabled to operate during stop mode through the stop mode oscillator enable bit (STOP\_ICLKDIS, STOP\_RCLKEN, or STOP\_XCLKEN) for the selected oscillator in the CONFIG2 register. The timebase module can be used in this mode to generate a periodic walk-up from stop mode.

If the oscillator has not been enabled to operate in stop mode, the timebase module will not be active during stop mode. In stop mode the timebase register is not accessible by the CPU.

If the timebase functions are not required during stop mode, reduce the power consumption by stopping the timebase before enabling the STOP instruction.

[illegible]

### Figure 11-5. SCI Receiver Block Diagram

## WAKE — Wakeup Condition Bit

This read/write bit determines which condition wakes up the SCI: a logic 1 (address mark) in the most significant bit position of a received character or an idle condition on the RxD pin. Reset clears the WAKE bit.

1 = Address mark wakeup

0 = Idle line wakeup

## ILTY — Idle Line Type Bit

This read/write bit determines when the SCI starts counting logic 1s as idle character bits. The counting begins either after the start bit or after the stop bit. If the count begins after the start bit, then a string of logic 1s preceding the stop bit may cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. Reset clears the ILTY bit.

1 = Idle character bit count begins after stop bit

0 = Idle character bit count begins after start bit

## PEN — Parity Enable Bit

This read/write bit enables the SCI parity function. (See [Table 11-5.](#)) When enabled, the parity function inserts a parity bit in the most significant bit position. (See [Figure 11-3.](#)) Reset clears the PEN bit.

1 = Parity function enabled

0 = Parity function disabled

## PTY — Parity Bit

This read/write bit determines whether the SCI generates and checks for odd parity or even parity. (See [Table 11-5.](#)) Reset clears the PTY bit.

1 = Odd parity

0 = Even parity

### NOTE

*Changing the PTY bit in the middle of a transmission or reception can generate a parity error.*

**Table 11-5. Character Format Selection**

Control Bits		Character Format				
M	PEN:PTY	Start Bits	Data Bits	Parity	Stop Bits	Character Length
0	0X	1	8	None	1	10 bits
1	0X	1	9	None	1	11 bits
0	10	1	7	Even	1	10 bits
0	11	1	7	Odd	1	10 bits
1	10	1	8	Even	1	11 bits
1	11	1	8	Odd	1	11 bits

$$\left| \frac{170 - 176}{170} \right| \times 100 = 3.53\%$$

### 12.5.3.6 Receiver Wakeup

So that the MCU can ignore transmissions intended only for other receivers in multiple-receiver systems, the receiver can be put into a standby state. Setting the receiver wakeup bit, RWU, in IRSCC2 puts the receiver into a standby state during which receiver interrupts are disabled.

Depending on the state of the WAKE bit in IRSCC1, either of two conditions on the RxD pin can bring the receiver out of the standby state:

- Address mark — An address mark is a logic 1 in the most significant bit position of a received character. When the WAKE bit is set, an address mark wakes the receiver from the standby state by clearing the RWU bit. The address mark also sets the SCI receiver full bit, SCRF. Software can then compare the character containing the address mark to the user-defined address of the receiver. If they are the same, the receiver remains awake and processes the characters that follow. If they are not the same, software can set the RWU bit and put the receiver back into the standby state.
- Idle input line condition — When the WAKE bit is clear, an idle character on the RxD pin wakes the receiver from the standby state by clearing the RWU bit. The idle character that wakes the receiver does not set the receiver idle bit, IDLE, or the SCI receiver full bit, SCRF. The idle line type bit, ILTY, determines whether the receiver begins counting logic 1s as idle character bits after the start bit or after the stop bit.

#### NOTE

*Clearing the WAKE bit after the RxD pin has been idle may cause the receiver to wake up immediately.*

### 12.5.3.7 Receiver Interrupts

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

- SCI receiver full (SCRF) — The SCRF bit in IRSCS1 indicates that the receive shift register has transferred a character to the IRSCDR. SCRF can generate a receiver interrupt request. Setting the SCI receive interrupt enable bit, SCRIE, in IRSCC2 enables the SCRF bit to generate receiver CPU interrupts.
- Idle input (IDLE) — The IDLE bit in IRSCS1 indicates that 10 or 11 consecutive logic 1s shifted in from the RxD pin. The idle line interrupt enable bit, ILIE, in IRSCC2 enables the IDLE bit to generate CPU interrupt requests.

### 12.5.3.8 Error Interrupts

The following receiver error flags in IRSCS1 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 IRSCDR. The previous character remains in the IRSCDR, and the new character is lost. The overrun interrupt enable bit, ORIE, in IRSCC3 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 IRSCC3 enables NF to generate SCI error CPU interrupt requests.

### M — Mode (Character Length) Bit

This read/write bit determines whether SCI characters are eight or nine bits long. (See [Table 12-6](#).) 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

### WAKE — Wakeup Condition Bit

This read/write bit determines which condition wakes up the SCI: a logic 1 (address mark) in the most significant bit position of a received character or an idle condition on the RxD pin. Reset clears the WAKE bit.

- 1 = Address mark wakeup
- 0 = Idle line wakeup

### ILTY — Idle Line Type Bit

This read/write bit determines when the SCI starts counting logic 1s as idle character bits. The counting begins either after the start bit or after the stop bit. If the count begins after the start bit, then a string of logic 1s preceding the stop bit may cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. Reset clears the ILTY bit.

- 1 = Idle character bit count begins after stop bit
- 0 = Idle character bit count begins after start bit

### PEN — Parity Enable Bit

This read/write bit enables the SCI parity function. (See [Table 12-6](#).) When enabled, the parity function inserts a parity bit in the most significant bit position. (See [Figure 12-6](#).) Reset clears the PEN bit.

- 1 = Parity function enabled
- 0 = Parity function disabled

### PTY — Parity Bit

This read/write bit determines whether the SCI generates and checks for odd parity or even parity. (See [Table 12-6](#).) Reset clears the PTY bit.

- 1 = Odd parity
- 0 = Even parity

### NOTE

*Changing the PTY bit in the middle of a transmission or reception can generate a parity error.*

**Table 12-6. Character Format Selection**

Control Bits		Character Format				
M	PEN:PTY	Start Bits	Data Bits	Parity	Stop Bits	Character Length
0	0X	1	8	None	1	10 bits
1	0X	1	9	None	1	11 bits
0	10	1	7	Even	1	10 bits
0	11	1	7	Odd	1	10 bits
1	10	1	8	Even	1	11 bits
1	11	1	8	Odd	1	11 bits

### NF — Receiver Noise Flag Bit

This clearable, read-only bit is set when the SCI detects noise on the RxD pin. NF generates an SCI error CPU interrupt request if the NEIE bit in IRSCC3 is also set. Clear the NF bit by reading IRSCS1 and then reading the IRSCDR. Reset clears the NF bit.

- 1 = Noise detected
- 0 = No noise detected

### FE — Receiver Framing Error Bit

This clearable, read-only bit is set when a logic 0 is accepted as the stop bit. FE generates an SCI error CPU interrupt request if the FEIE bit in IRSCC3 also is set. Clear the FE bit by reading IRSCS1 with FE set and then reading the IRSCDR. Reset clears the FE bit.

- 1 = Framing error detected
- 0 = No framing error detected

### PE — Receiver Parity Error Bit

This clearable, read-only bit is set when the SCI detects a parity error in incoming data. PE generates an SCI error CPU interrupt request if the PEIE bit in IRSCC3 is also set. Clear the PE bit by reading IRSCS1 with PE set and then reading the IRSCDR. Reset clears the PE bit.

- 1 = Parity error detected
- 0 = No parity error detected

## 12.9.5 IRSCI Status Register 2

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

- Break character detected
- Incoming data

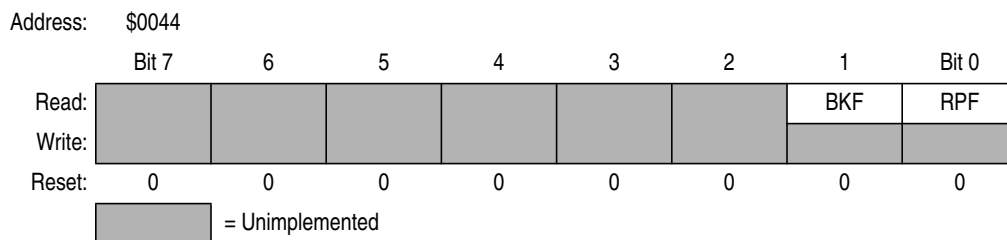


Figure 12-17. IRSCI Status Register 2 (IRSCS2)

### BKF — Break Flag Bit

This clearable, read-only bit is set when the SCI detects a break character on the RxD pin. In IRSCS1, the FE and SCRF bits are also set. In 9-bit character transmissions, the R8 bit in IRSCC3 is cleared. BKF does not generate a CPU interrupt request. Clear BKF by reading IRSCS2 with BKF set and then reading the IRSCDR. 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



## Serial Peripheral Interface Module (SPI)

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$0010	SPI Control Register (SPCR)	Read: SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
		Write:							
		Reset:	0	0	1	0	1	0	0
\$0011	SPI Status and Control Register (SPSCR)	Read: SPRF	ERRIE	OVRF	MODF	SPTIE	MODFEN	SPR1	SPR0
		Write:							
		Reset:	0	0	0	0	1	0	0
\$0012	SPI Data Register (SPDR)	Read: R7	R6	R5	R4	R3	R2	R1	R0
		Write: T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset						

= Unimplemented
  = Reserved

**Figure 13-1. SPI I/O Register Summary**

## 13.4 Functional Description

Figure 13-2 shows the structure of the SPI module.

The SPI module allows full-duplex, synchronous, serial communication between the MCU and peripheral devices, including other MCUs. Software can poll the SPI status flags or SPI operation can be interrupt-driven.

The following paragraphs describe the operation of the SPI module.

### 13.4.1 Master Mode

The SPI operates in master mode when the SPI master bit, SPMSTR, is set.

#### NOTE

*Configure the SPI modules as master or slave before enabling them. Enable the master SPI before enabling the slave SPI. Disable the slave SPI before disabling the master SPI. (See 13.13.1 SPI Control Register.)*

Only a master SPI module can initiate transmissions. Software begins the transmission from a master SPI module by writing to the transmit data register. If the shift register is empty, the byte immediately transfers to the shift register, setting the SPI transmitter empty bit, SPTIE. The byte begins shifting out on the MOSI pin under the control of the serial clock. (See Figure 13-3.)

The SPR1 and SPR0 bits control the baud rate generator and determine the speed of the shift register. (See 13.13.2 SPI Status and Control Register.) Through the SPSCCK pin, the baud rate generator of the master also controls the shift register of the slave peripheral.

As the byte shifts out on the MOSI pin of the master, another byte shifts in from the slave on the master's MISO pin. The transmission ends when the receiver full bit, SPRF, becomes set. At the same time that SPRF becomes set, the byte from the slave transfers to the receive data register. In normal operation, SPRF signals the end of a transmission. Software clears SPRF by reading the SPI status and control register with SPRF set and then reading the SPI data register. Writing to the SPI data register clears the SPTIE bit.

Since the SPTE bit cannot be cleared during a break with the BCFE bit cleared, a write to the transmit data register in break mode does not initiate a transmission nor is this data transferred into the shift register. Therefore, a write to the SPDR in break mode with the BCFE bit cleared has no effect.

### 13.12 I/O Signals

The SPI module has five I/O pins and shares four of them with a parallel I/O port. They are:

- MISO — Data received
- MOSI — Data transmitted
- SPSCCK — Serial clock
- $\overline{SS}$  — Slave select
- CGND — Clock ground (internally connected to  $V_{SS}$ )

The SPI has limited inter-integrated circuit ( $I^2C$ ) capability (requiring software support) as a master in a single-master environment. To communicate with  $I^2C$  peripherals, MOSI becomes an open-drain output when the SPWOM bit in the SPI control register is set. In  $I^2C$  communication, the MOSI and MISO pins are connected to a bidirectional pin from the  $I^2C$  peripheral and through a pullup resistor to  $V_{DD}$ .

#### 13.12.1 MISO (Master In/Slave Out)

MISO is one of the two SPI module pins that transmits serial data. In full duplex operation, the MISO pin of the master SPI module is connected to the MISO pin of the slave SPI module. The master SPI simultaneously receives data on its MISO pin and transmits data from its MOSI pin.

Slave output data on the MISO pin is enabled only when the SPI is configured as a slave. The SPI is configured as a slave when its SPMSTR bit is logic 0 and its  $\overline{SS}$  pin is at logic 0. To support a multiple-slave system, a logic 1 on the  $\overline{SS}$  pin puts the MISO pin in a high-impedance state.

When enabled, the SPI controls data direction of the MISO pin regardless of the state of the data direction register of the shared I/O port.

#### 13.12.2 MOSI (Master Out/Slave In)

MOSI is one of the two SPI module pins that transmits serial data. In full-duplex operation, the MOSI pin of the master SPI module is connected to the MOSI pin of the slave SPI module. The master SPI simultaneously transmits data from its MOSI pin and receives data on its MISO pin.

When enabled, the SPI controls data direction of the MOSI pin regardless of the state of the data direction register of the shared I/O port.

#### 13.12.3 SPSCCK (Serial Clock)


The serial clock synchronizes data transmission between master and slave devices. In a master MCU, the SPSCCK pin is the clock output. In a slave MCU, the SPSCCK pin is the clock input. In full-duplex operation, the master and slave MCUs exchange a byte of data in eight serial clock cycles.

When enabled, the SPI controls data direction of the SPSCCK pin regardless of the state of the data direction register of the shared I/O port.

### 14.6.7 MMIIC CRC Data Register (MMCRCDR)

Address: \$004E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MMCRCD7	MMCRCD6	MMCRCD5	MMCRCD4	MMCRCD3	MMCRCD2	MMCRCD1	MMCRCD0
Write:								
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

**Figure 14-10. MMIIC CRC Data Register (MMCRCDR)**

When the MMIIC module is enabled, MMEN = 1, and the CRC buffer full flag is set (MMCRCBF = 1), data in this read-only register contains the generated CRC byte for the last byte of received or transmitted data.


A CRC byte is generated for each received and transmitted data byte and loaded to the CRC data register. The MMCRCBF bit will be set to indicate the CRC byte is ready in the CRC data register.

Reading the CRC data register clears the MMCRCBF bit. If the CRC data register is not read, the MMCRCBF bit will be cleared by hardware before the next CRC byte is loaded.

### 14.6.8 MMIIC Frequency Divider Register (MMFDR)

Address: \$004F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	MMBR2	MMBR1	MMBR0
Write:								
Reset:	0	0	0	0	0	1	0	0

 = Unimplemented

**Figure 14-11. MMIIC Frequency Divider Register (MMFDR)**

The three bits in the frequency divider register (MMFDR) selects the divider to divide the bus clock to the desired baud rate for the MMIIC data transfer.

Table 14-2 shows the divider values for MMBR[2:0].

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

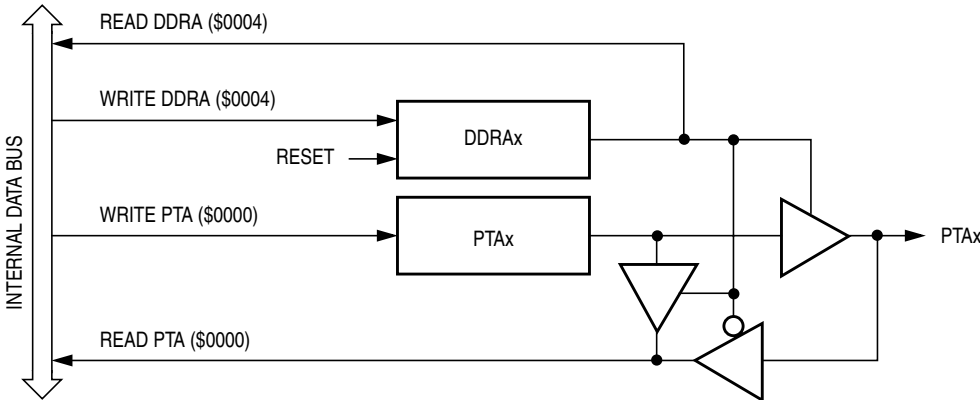
These read/write bits control port A data direction. Reset clears DDRA[7:0], configuring all port A pins as inputs.

- 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 <sup>(1)</sup>	Input, Hi-Z <sup>(2)</sup>	DDRA[7:0]	Pin	PTA[7:0] <sup>(3)</sup>
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.

Table 16-5 summarizes the operation of the port D pins.

**Table 16-5. Port D Pin Functions**

DDRD Bit	PTD Bit	I/O Pin Mode	Accesses to DDRD	Accesses to PTD	
			Read/Write	Read	Write
0	X <sup>(1)</sup>	Input, Hi-Z <sup>(2)</sup>	DDRD[7:0]	Pin	PTD[7:0] <sup>(3)</sup>
1	X	Output	DDRD[7:0]	PTD[7:0]	PTD[7:0]

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