

Welcome to [E-XFL.COM](#)

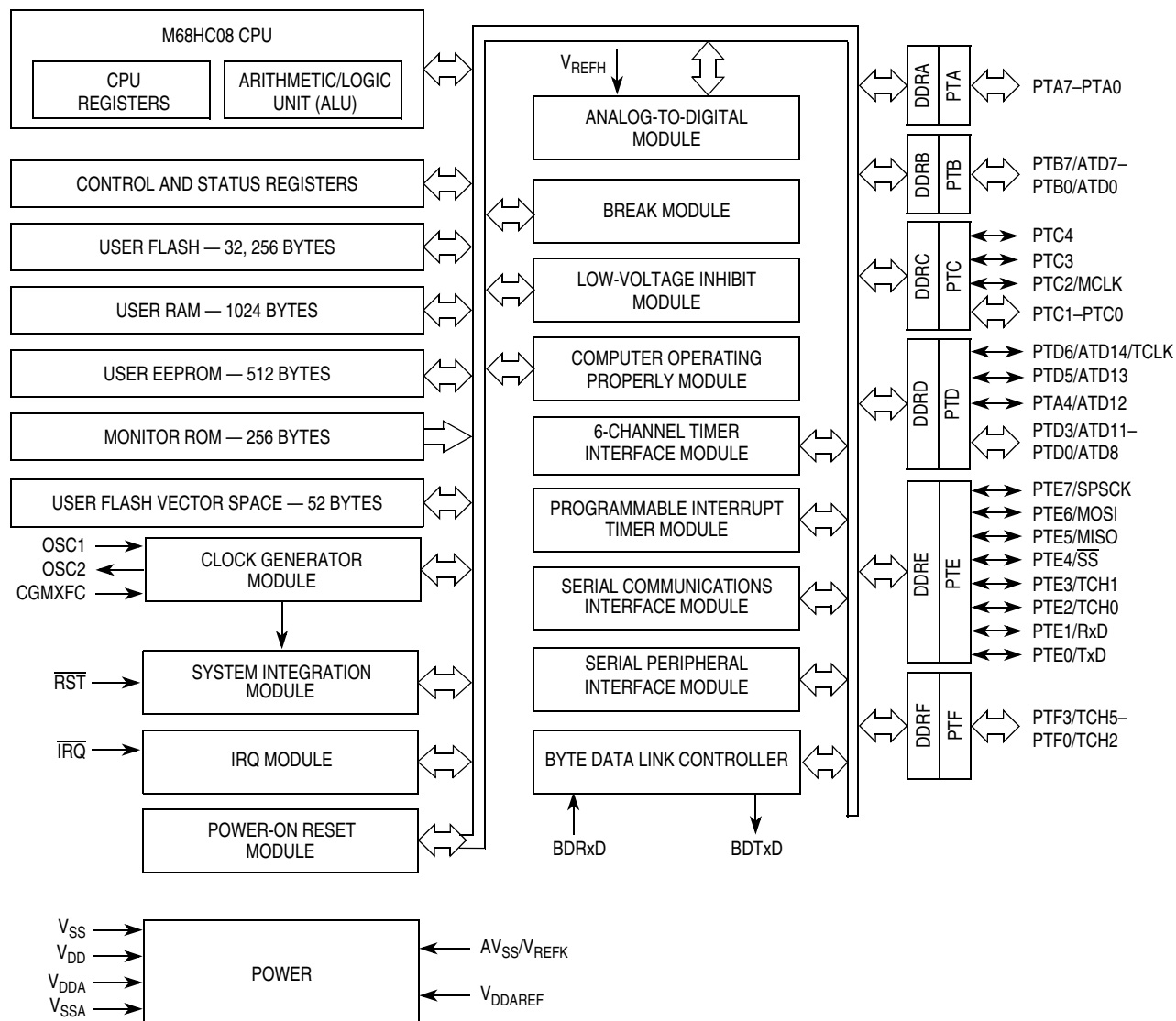
### What is "[Embedded - Microcontrollers](#)"?

"[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.

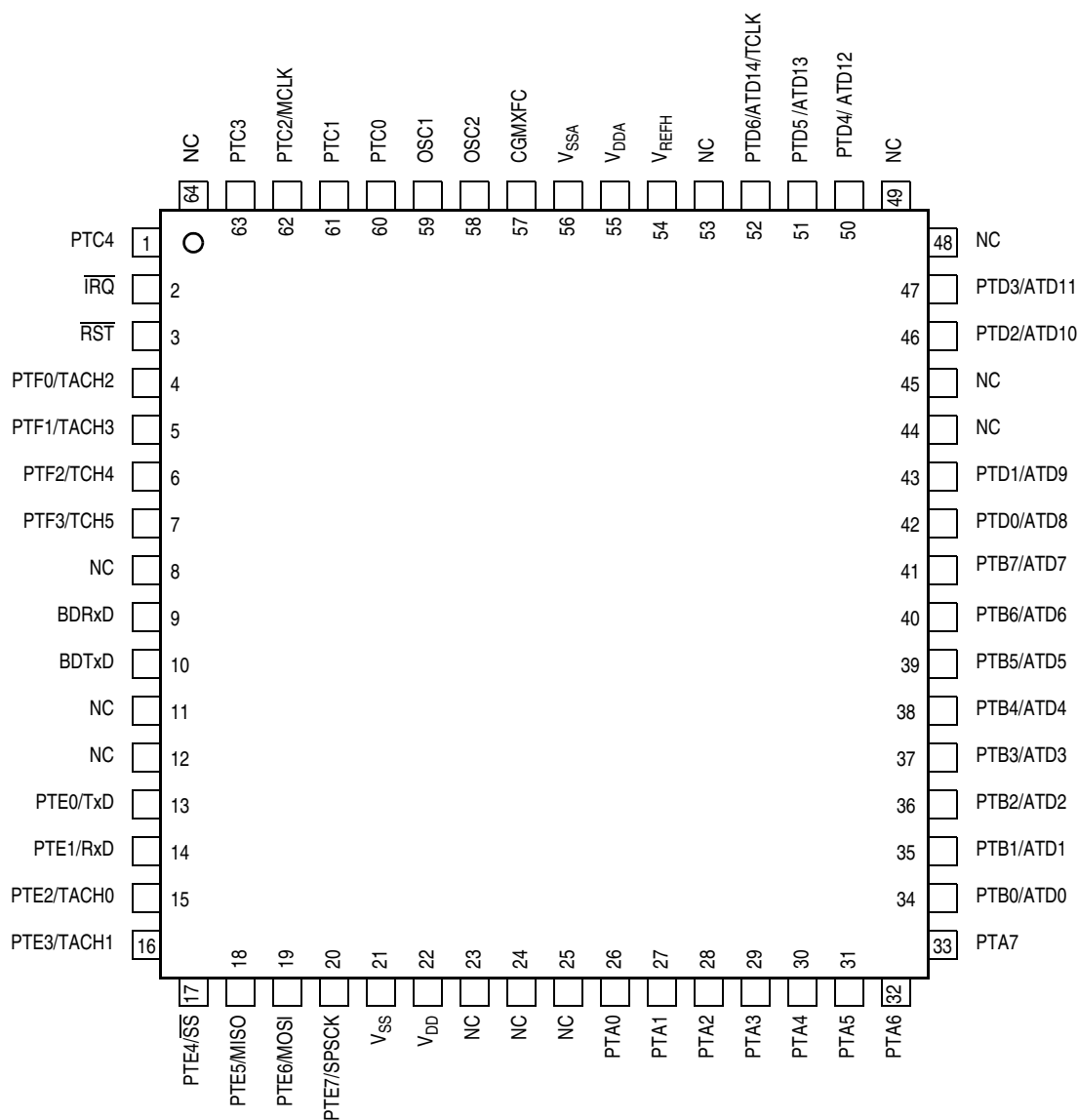
### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8.4MHz
Connectivity	SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	40
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 15x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LCC (J-Lead)
Supplier Device Package	52-PLCC (19.1x19.1)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908as32acfner">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908as32acfner</a>



**Figure 1-1. MCU Block Diagram for the MC68HC908AS32A**



**Figure 1-3. 64-Pin QFP Assignments (Top View)**

## NOTE

*The following pin descriptions are just a quick reference. For a more detailed representation, see Chapter 13 Input/Output Ports.*

### 1.4.1 Power Supply Pins ( $V_{DD}$ and $V_{SS}$ )

$V_{DD}$  and  $V_{SS}$  are the power supply and ground pins. The MCU operates from a single power supply.

Fast signal transitions on MCU pins place high, short-duration current demands on the power supply. To prevent noise problems, take special care to provide power supply bypassing at the MCU as shown in Figure 1-4. Place the C1 bypass capacitor as close to the MCU as possible. Use a high-frequency response ceramic capacitor for C1. C2 is an optional bulk current bypass capacitor for use in applications that require the port pins to source high current levels.

## Memory

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0005	Data Direction Register B (DDRB) See page 153.	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0006	Data Direction Register C (DDRC) See page 155.	Read:	MCLKEN	0	0	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0007	Data Direction Register D (DDRD) See page 158.	Read:	0	DDRD6	DDRD5	DDRD4	DDRD3	DDR2	DDRD1	DDRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0008	Port E Data Register (PTE) See page 159.	Read:	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0
		Write:								
		Reset:	Unaffected by reset							
\$0009	Port F Data Register (PTF) See page 161.	Read:	0	0	0	0	PTF3	PTF2	PTF1	PTF0
		Write:								
		Reset:	Unaffected by reset							
\$000A	Reserved		R	R	R	R	R	R	R	R
\$000B	Reserved		R	R	R	R	R	R	R	R
\$000C	Data Direction Register E (DDRE) See page 160.	Read:	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000D	Data Direction Register F (DDRF) See page 162.	Read:	0	0	0	0	DDRF3	DDRF2	DDRF1	DDRF0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000E	Reserved		R	R	R	R	R	R	R	R
\$000F	Reserved		R	R	R	R	R	R	R	R
\$0010	SPI Control Register (SPCR) See page 221.	Read:	SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
		Write:								
		Reset:	0	0	1	0	1	0	0	0
\$0011	SPI Status and Control Register (SPSCR) See page 223.	Read:	SPRF	ERRIE	OVRF	MODF	SPTE	MODFEN	SPR1	SPR0
		Write:								
		Reset:	0	0	0	0	1	0	0	0
\$0012	SPI Data Register (SPDR) See page 225.	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Indeterminate after reset							

= Unimplemented    
 R = Reserved    
 U = Unaffected

**Figure 2-2. I/O Data, Status and Control Registers (Sheet 2 of 6)**

### 4.3.1.6 Digital Loopback Mode

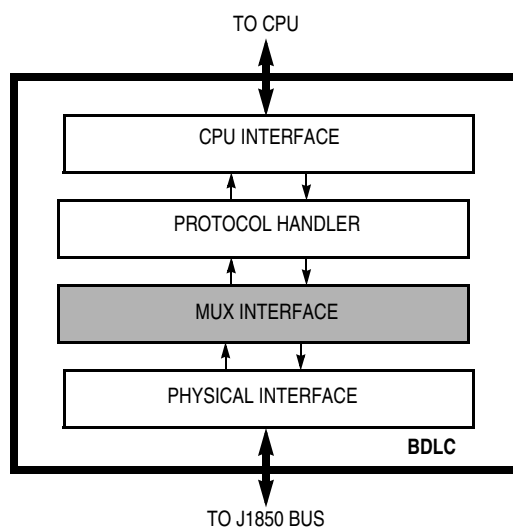
When a bus fault has been detected, the digital loopback mode is used to determine if the fault condition is caused by failure in the node's internal circuits or elsewhere in the network, including the node's analog physical interface. In this mode, the transmit digital output pin (BDTxD) and the receive digital input pin (BDRxD) of the digital interface are disconnected from the analog physical interface and tied together to allow the digital portion of the BDLC to transmit and receive its own messages without driving the J1850 bus.

### 4.3.1.7 Analog Loopback Mode

Analog loopback is used to determine if a bus fault has been caused by a failure in the node's off-chip analog transceiver or elsewhere in the network. The BCLD analog loopback mode does not modify the digital transmit or receive functions of the BDLC. It does, however, ensure that once analog loopback mode is exited, the BDLC will wait for an idle bus condition before participation in network communication resumes. If the off-chip analog transceiver has a loopback mode, it usually causes the input to the output drive stage to be looped back into the receiver, allowing the node to receive messages it has transmitted without driving the J1850 bus. In this mode, the output to the J1850 bus is typically high impedance. This allows the communication path through the analog transceiver to be tested without interfering with network activity. Using the BDLC analog loopback mode in conjunction with the analog transceiver's loopback mode ensures that, once the off-chip analog transceiver has exited loopback mode, the BCLD will not begin communicating before a known condition exists on the J1850 bus.

## 4.4 BDLC MUX Interface

The MUX interface is responsible for bit encoding/decoding and digital noise filtering between the protocol handler and the physical interface.



**Figure 4-4. BDLC Block Diagram**

## Chapter 5

# Clock Generator Module (CGM)

### 5.1 Introduction

The CGM generates the crystal clock signal, CGMXCLK, which operates at the frequency of the crystal. The CGM also generates the base clock signal, CGMOUT, from which the system clocks are derived. CGMOUT is based on either the crystal clock divided by two or the phase-locked loop (PLL) clock, CGMVCLK, divided by two. The PLL is a frequency generator designed for use with 1-MHz to 8-MHz crystals or ceramic resonators. The PLL can generate an 8-MHz bus frequency without using high frequency crystals.

### 5.2 Features

Features include:

- Phase-locked loop with output frequency in integer multiples of the crystal reference
- Programmable hardware voltage-controlled oscillator (VCO) for low-jitter operation
- Automatic bandwidth control mode for low-jitter operation
- Automatic frequency lock detector
- CPU interrupt on entry or exit from locked condition

### 5.3 Functional Description

The CGM consists of three major submodules:

- Crystal oscillator circuit — The crystal oscillator circuit generates the constant crystal frequency clock, CGMXCLK.
- Phase-locked loop (PLL) — The PLL generates the programmable VCO frequency clock CGMVCLK.
- Base clock selector circuit — This software-controlled circuit selects either CGMXCLK divided by two or the VCO clock, CGMVCLK, divided by two as the base clock, CGMOUT. The system clocks are derived from CGMOUT.

Figure 5-2 shows the structure of the CGM.

# Chapter 7

## Configuration Register (CONFIG2)

### 7.1 Introduction

This section describes the configuration register (CONFIG2). This register contains bits that configures the device to either the MC68HC08AZxx emulator or the MC68HC08ASxx emulator

### 7.2 Functional Description

The configuration register is a write-once register. Out of reset, the configuration register will read the default. Once the register is written, further writes will have no effect until a reset occurs.

Address: \$FE09

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	EEDIVCLK	R	R	R	AS32A	R	R	R
Write:								
Reset:	0	0	0	1	1	0	0	0

R = Reserved

**Figure 7-1. Configuration Register (CONFIG2)**

#### EEDIVCLK — EEPROM Timebase Divider Clock Select Bit

This bit selects the reference clock source for the EEPROM timebase divider module.

- 1 = EExDIV clock input is driven by internal bus clock
- 0 = EExDIV clock input is driven by CGMXCLK

#### AS32A— Device Indicator Bit

This bit is used to distinguish MC68HC908AS32A from older non-A suffix versions.

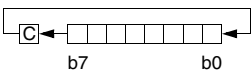
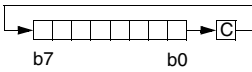
- 1 = A version
- 0 = Non-A version

Table 9-1. Instruction Set Summary (Sheet 3 of 6)

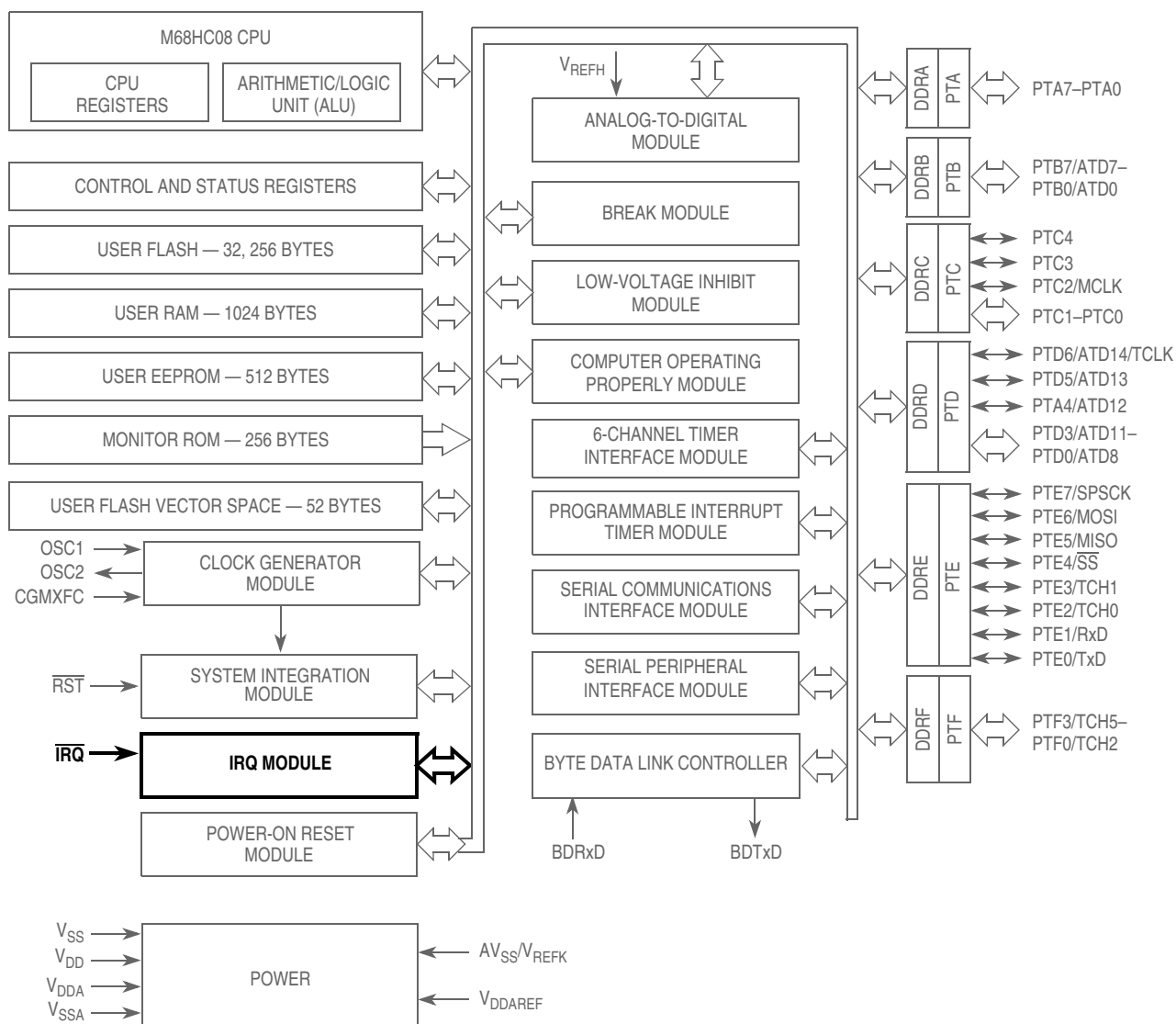
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
CLR <i>opr</i> CLRA CLR <sub>X</sub> CLR <sub>H</sub> CLR <i>opr</i> , <sub>X</sub> CLR <sub>X</sub> CLR <i>opr</i> ,SP	Clear	M ← \$00 A ← \$00 X ← \$00 H ← \$00 M ← \$00 M ← \$00 M ← \$00	0	–	–	0	1	–	DIR INH INH INH IX1 IX SP1	3F 4F 5F 8C 6F 7F 9E6F	dd  ff ff	3 1 1 1 3 2 4
CMP # <i>opr</i> CMP <i>opr</i> CMP <i>opr</i> CMP <i>opr</i> , <sub>X</sub> CMP <i>opr</i> , <sub>X</sub> CMP <sub>X</sub> CMP <i>opr</i> ,SP CMP <i>opr</i> ,SP	Compare A with M	(A) – (M)	†	–	–	†	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	A1 B1 C1 D1 E1 F1 9EE1 9ED1	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
COM <i>opr</i> COMA COM <sub>X</sub> COM <i>opr</i> , <sub>X</sub> COM <sub>X</sub> COM <i>opr</i> ,SP	Complement (One's Complement)	M ← (M̄) = \$FF – (M) A ← (Ā) = \$FF – (M) X ← (X̄) = \$FF – (M) M ← (M̄) = \$FF – (M) M ← (M̄) = \$FF – (M) M ← (M̄) = \$FF – (M)	0	–	–	†	†	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd  ff ff ff	4 1 1 4 3 5
CPHX # <i>opr</i> CPHX <i>opr</i>	Compare H:X with M	(H:X) – (M:M + 1)	†	–	–	†	†	†	IMM DIR	65 75	ii ii+1 dd	3 4
CPX # <i>opr</i> CPX <i>opr</i> CPX <i>opr</i> CPX <sub>X</sub> CPX <i>opr</i> , <sub>X</sub> CPX <i>opr</i> , <sub>X</sub> CPX <i>opr</i> ,SP CPX <i>opr</i> ,SP	Compare X with M	(X) – (M)	†	–	–	†	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	A3 B3 C3 D3 E3 F3 9EE3 9ED3	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
DAA	Decimal Adjust A	(A) <sub>10</sub>	U	–	–	†	†	†	INH	72		2
DBNZ <i>opr</i> , <i>rel</i> DBNZ <sub>A</sub> <i>rel</i> DBNZ <sub>X</sub> <i>rel</i> DBNZ <i>opr</i> , <sub>X</sub> , <i>rel</i> DBNZ <sub>X</sub> , <i>rel</i> DBNZ <i>opr</i> ,SP, <i>rel</i>	Decrement and Branch if Not Zero	A ← (A) – 1 or M ← (M) – 1 or X ← (X) – 1 PC ← (PC) + 3 + <i>rel</i> ? (result) ≠ 0 PC ← (PC) + 2 + <i>rel</i> ? (result) ≠ 0 PC ← (PC) + 2 + <i>rel</i> ? (result) ≠ 0 PC ← (PC) + 3 + <i>rel</i> ? (result) ≠ 0 PC ← (PC) + 2 + <i>rel</i> ? (result) ≠ 0 PC ← (PC) + 4 + <i>rel</i> ? (result) ≠ 0	–	–	–	–	–	–	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	5 3 3 5 4 6
DEC <i>opr</i> DECA DEC <sub>X</sub> DEC <i>opr</i> , <sub>X</sub> DEC <sub>X</sub> DEC <i>opr</i> ,SP	Decrement	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1 M ← (M) – 1	†	–	–	†	†	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd  ff ff	4 1 1 4 3 5
DIV	Divide	A ← (H:A)/(X) H ← Remainder	–	–	–	–	†	†	INH	52		7
EOR # <i>opr</i> EOR <i>opr</i> EOR <i>opr</i> EOR <i>opr</i> , <sub>X</sub> EOR <i>opr</i> , <sub>X</sub> EOR <sub>X</sub> EOR <i>opr</i> ,SP EOR <i>opr</i> ,SP	Exclusive OR M with A	A ← (A ⊕ M)	0	–	–	†	†	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A8 B8 C8 D8 E8 F8 9EE8 9ED8	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
INC <i>opr</i> INCA INC <sub>X</sub> INC <i>opr</i> , <sub>X</sub> INC <sub>X</sub> INC <i>opr</i> ,SP	Increment	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1 M ← (M) + 1	†	–	–	†	†	–	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd  ff ff	4 1 1 4 3 5



Table 9-1. Instruction Set Summary (Sheet 5 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z				
PULA	Pull A from Stack	$SP \leftarrow (SP + 1); \text{Pull (A)}$	–	–	–	–	–	INH	86		2
PULH	Pull H from Stack	$SP \leftarrow (SP + 1); \text{Pull (H)}$	–	–	–	–	–	INH	8A		2
PULX	Pull X from Stack	$SP \leftarrow (SP + 1); \text{Pull (X)}$	–	–	–	–	–	INH	88		2
ROL <i>opr</i> ROLA ROLX ROL <i>opr</i> ,X ROL ,X ROL <i>opr</i> ,SP	Rotate Left through Carry		↑	–	–	–	↑	DIR INH INH IX1 IX SP1	39 49 59 69 79 9E69	dd ff ff	4 1 1 4 3 5
ROR <i>opr</i> RORA RORX ROR <i>opr</i> ,X ROR ,X ROR <i>opr</i> ,SP	Rotate Right through Carry		↑	–	–	–	↑	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	4 1 1 4 3 5
RSP	Reset Stack Pointer	$SP \leftarrow \$FF$	–	–	–	–	–	INH	9C		1
RTI	Return from Interrupt	$SP \leftarrow (SP + 1); \text{Pull (CCR)}$ $SP \leftarrow (SP + 1); \text{Pull (A)}$ $SP \leftarrow (SP + 1); \text{Pull (X)}$ $SP \leftarrow (SP + 1); \text{Pull (PCH)}$ $SP \leftarrow (SP + 1); \text{Pull (PCL)}$	↑	↑	↑	↑	↑	INH	80		7
RTS	Return from Subroutine	$SP \leftarrow SP + 1; \text{Pull (PCH)}$ $SP \leftarrow SP + 1; \text{Pull (PCL)}$	–	–	–	–	–	INH	81		4
SBC # <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> ,X SBC <i>opr</i> ,X SBC ,X SBC <i>opr</i> ,SP SBC <i>opr</i> ,SP	Subtract with Carry	$A \leftarrow (A) - (M) - (C)$	↑	–	–	–	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A2 B2 C2 D2 E2 F2 9EE2 9ED2	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
SEC	Set Carry Bit	$C \leftarrow 1$	–	–	–	–	1	INH	99		1
SEI	Set Interrupt Mask	$I \leftarrow 1$	–	–	1	–	–	INH	9B		2
STA <i>opr</i> STA <i>opr</i> STA <i>opr</i> ,X STA <i>opr</i> ,X STA ,X STA <i>opr</i> ,SP STA <i>opr</i> ,SP	Store A in M	$M \leftarrow (A)$	0	–	–	↑	↑	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh ll ee ff ff ff ff ee ff	3 4 4 3 2 4 5
STHX <i>opr</i>	Store H:X in M	$(M:M + 1) \leftarrow (H:X)$	0	–	–	↑	↑	DIR	35	dd	4
STOP	Enable Interrupts, Stop Processing, Refer to MCU Documentation	$I \leftarrow 0$ ; Stop Processing	–	–	0	–	–	INH	8E		1
STX <i>opr</i> STX <i>opr</i> STX <i>opr</i> ,X STX <i>opr</i> ,X STX ,X STX <i>opr</i> ,SP STX <i>opr</i> ,SP	Store X in M	$M \leftarrow (X)$	0	–	–	↑	↑	DIR EXT IX2 IX1 IX SP1 SP2	BF CF DF EF FF 9EEF 9EDF	dd hh ll ee ff ff ff ff ee ff	3 4 4 3 2 4 5
SUB # <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> ,X SUB <i>opr</i> ,X SUB ,X SUB <i>opr</i> ,SP SUB <i>opr</i> ,SP	Subtract	$A \leftarrow (A) - (M)$	↑	–	–	–	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A0 B0 C0 D0 E0 F0 9EE0 9ED0	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5

## External Interrupt Module (IRQ)



**Figure 10-1. Block Diagram Highlighting IRQ Block and Pins**

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

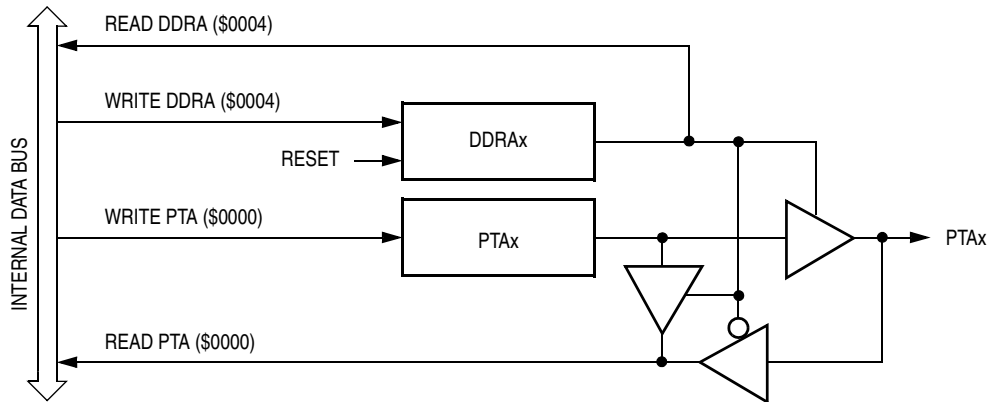
These read/write bits control port A data direction. Reset clears DDRA[7:0], configuring all port A pins as inputs.

- 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 13-3 shows the port A I/O logic.



**Figure 13-3. Port A I/O Circuit**

When bit DDRAx is a 1, reading address \$0000 reads the PTAx data latch. When bit DDRAx is a 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 13-1 summarizes the operation of the port A pins.

**Table 13-1. Port A Pin Functions**

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

X = don't care

Hi-Z = high impedance

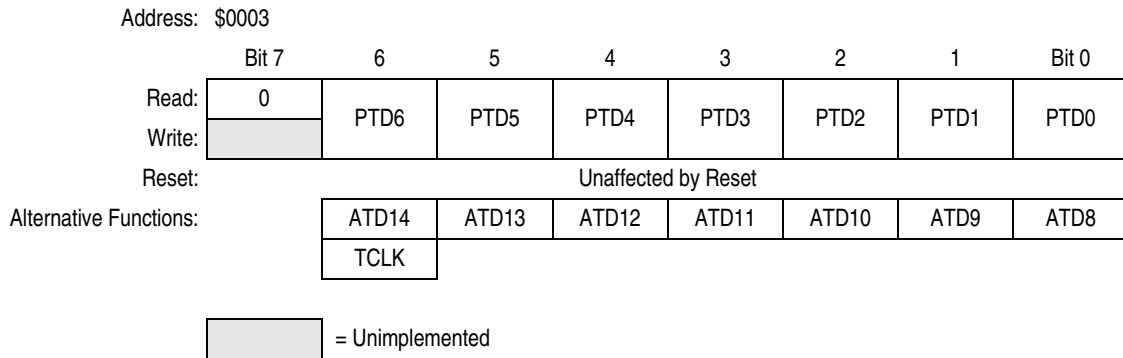
1. Writing affects data register, but does not affect input.

## 13.5 Port D

Port D is an 7-bit general-purpose I/O port.

### 13.5.1 Port D Data Register

Port D is a 7-bit special function port that shares seven of its pins with the analog to digital converter and two with the timer interface modules.



**Figure 13-10. Port D Data Register (PTD)**

#### PTD[6:0] — Port D Data Bits

PTD[6:0] are read/write, software programmable bits. Data direction of PTD[6:0] pins are under the control of the corresponding bit in data direction register D.

#### ATD[14:8] — ADC Channel Status Bits

PTD6/ATD14/TCLK–PTD0/ATD8 are seven of the 15 ADC channels. The ADC channel select bits, CH[4:0], determine whether the PTD6/ATD14/TCLK–PTD0/ATD8 pins are ADC channels or general-purpose I/O pins. If an ADC channel is selected and a read of this corresponding bit in the port B data register occurs, the data will be 0 if the data direction for this bit is programmed as an input. Otherwise, the data will reflect the value in the data latch. See Chapter 3 Analog-to-Digital Converter (ADC).

Data direction register D (DDRD) does not affect the data direction of port D pins that are being used by the TIM. However, the DDRD bits always determine whether reading port D returns the states of the latches or a 0.

#### TCLK — Timer Clock Input Bit

The PTD6/ATD14/TACLK pin is the external clock input for the timer interface module (TIM). The prescaler select bits, PS[2:0], select PTD6/ATD14/TACLK as the TIM clock input. See Chapter 17 Timer Interface Module (TIM).

When not selected as the TIM clock, PTD6/ATD14/TACLK is available for general-purpose I/O. While TCLK is selected corresponding DDRD bits have no effect.

### 13.5.2 Data Direction Register D

Data direction register D determines whether each port D pin is an input or an output. Writing a 1 to a DDRD bit enables the output buffer for the corresponding port D pin; a 0 disables the output buffer.

## 13.6 Port E

Port E is an 8-bit special function port that shares two of its pins with the TIM, two of its pins with the serial communications interface module (SCI), and four of its pins with the serial peripheral interface module (SPI).

### 13.6.1 Port E Data Register

The port E data register contains a data latch for each of the eight port E pins.

Address: \$0008								
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0
Write:								
Reset:	Unaffected by reset							
Alternative Function:	SPSCK	MOSI	MISO	SS	TCH1	TCH0	RxD	TxD

**Figure 13-13. Port E Data Register (PTE)**

#### PTE[7:0] — Port E Data Bits

PTE[7:0] are read/write, software programmable bits. Data direction of each port E pin is under the control of the corresponding bit in data direction register E.

#### SPSCK — SPI Serial Clock Bit

The PTE7/SPSCK pin is the serial clock input of an SPI slave module and serial clock output of an SPI master module. When the SPE bit is clear, the PTE7/SPSCK pin is available for general-purpose I/O. See Chapter 16 Serial Peripheral Interface (SPI).

#### MOSI — Master Out/Slave In Bit

The PTE6/MOSI pin is the master out/slave in terminal of the SPI module. When the SPE bit is clear, the PTE6/MOSI pin is available for general-purpose I/O.

#### MISO — Master In/Slave Out Bit

The PTE5/MISO pin is the master in/slave out terminal of the SPI module. When the SPI enable bit, SPE, is clear, the SPI module is disabled, and the PTE5/MISO pin is available for general-purpose I/O. See Chapter 16 Serial Peripheral Interface (SPI).

#### $\overline{SS}$ — Slave Select Bit

The PTE4/ $\overline{SS}$  pin is the slave select input of the SPI module. When the SPE bit is clear, or when the SPI master bit (SPMSTR) is set and MODFEN bit is low, the PTE4/ $\overline{SS}$  pin is available for general-purpose I/O. See 16.12.4  $\overline{SS}$  (Slave Select). When the SPI is enabled as a slave, the DDRF0 bit in data direction register E (DDRE) has no effect on the PTE4/ $\overline{SS}$  pin.

#### NOTE

*Data direction register E (DDRE) does not affect the data direction of port E pins that are being used by the SPI module. However, the DDRE bits always determine whether reading port E returns the states of the latches or the states of the pins. See Table 13-5.*

### 14.4.3.1 Character Length

The receiver can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCC1) determines character length. When receiving 9-bit data, bit R8 in SCI control register 2 (SCC2) is the ninth bit (bit 8). When receiving 8-bit data, bit R8 is a copy of the eighth bit (bit 7).

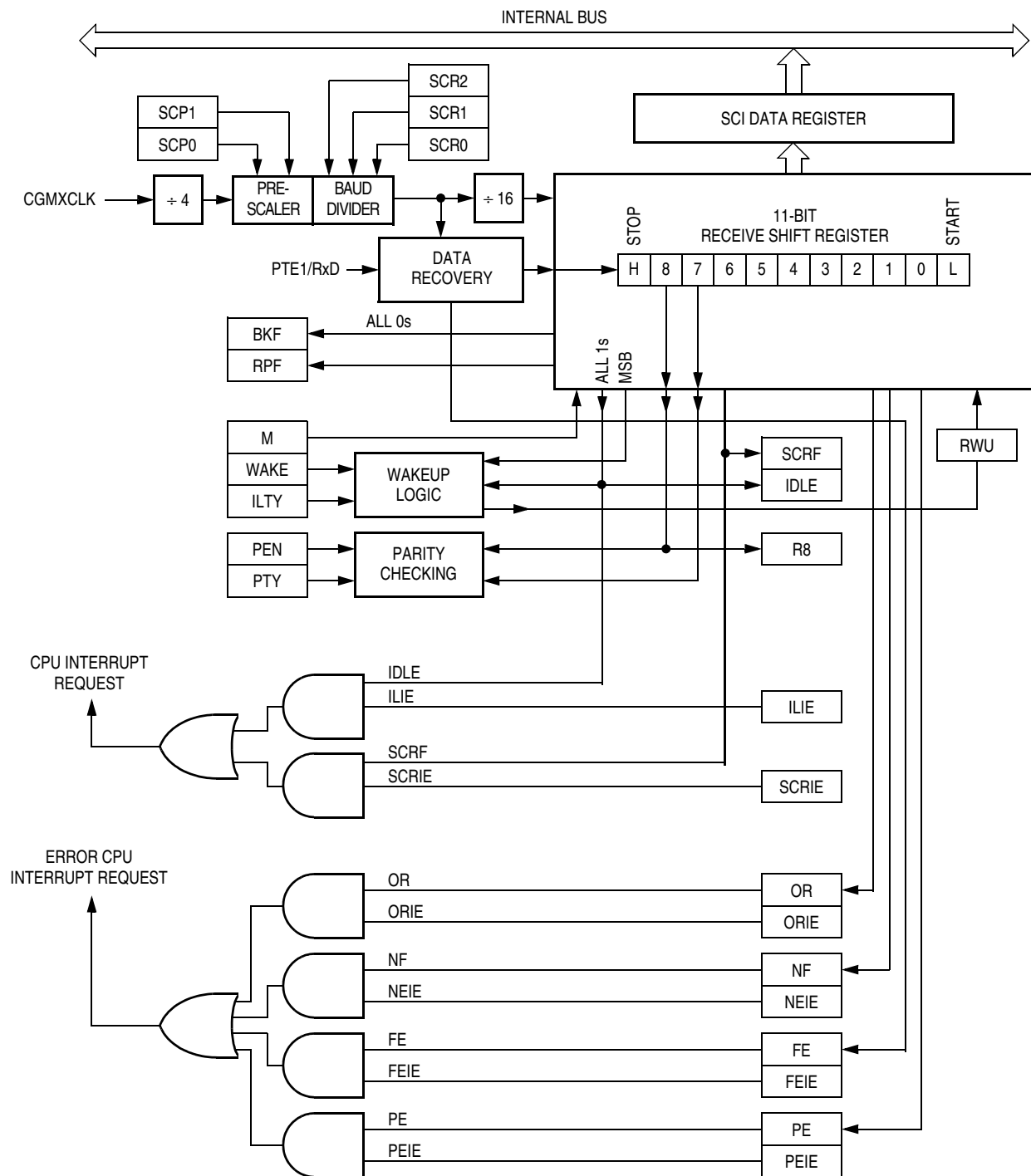


Figure 14-5. SCI Receiver Block Diagram

Start bit verification is not successful if any two of the three verification samples are 1s. If start bit verification is not successful, the RT clock is reset and a new search for a start bit begins.

To determine the value of a data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 14-3 summarizes the results of the data bit samples.

**Table 14-3. Data Bit Recovery**

RT8, RT9, and RT10 Samples	Data Bit Determination	Noise Flag
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

**NOTE**

*The RT8, RT9, and RT10 samples do not affect start bit verification. If any or all of the RT8, RT9, and RT10 start bit samples are 1s following a successful start bit verification, the noise flag (NF) is set and the receiver assumes that the bit is a start bit.*

To verify a stop bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 14-4 summarizes the results of the stop bit samples.

**Table 14-4. Stop Bit Recovery**

RT8, RT9, and RT10 Samples	Framing Error Flag	Noise Flag
000	1	0
001	1	1
010	1	1
011	0	1
100	1	1
101	0	1
110	0	1
111	0	0

#### 14.4.3.4 Framing Errors

If the data recovery logic does not detect a 1 where the stop bit should be in an incoming character, it sets the framing error bit, FE, in SCS1. A break character also sets the FE bit because a break character has no stop bit. The FE bit is set at the same time that the SCRF bit is set.

### 14.8.3 SCI Control Register 3

SCI control register 3:

- Stores the ninth SCI data bit received and the ninth SCI data bit to be transmitted.
- Enables the following interrupts:
  - Receiver overrun interrupts
  - Noise error interrupts
  - Framing error interrupts
  - Parity error interrupts

Address:	\$0015							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
Write:								
Reset:	U	U	0	0	0	0	0	0
	= Unimplemented			= Reserved		U = Unaffected		

**Figure 14-11. SCI Control Register 3 (SCC3)**

#### R8 — Received Bit 8

When the SCI is receiving 9-bit characters, R8 is the read-only ninth bit (bit 8) of the received character. R8 is received at the same time that the SCDR receives the other 8 bits.

When the SCI is receiving 8-bit characters, R8 is a copy of the eighth bit (bit 7). Reset has no effect on the R8 bit.

#### T8 — Transmitted Bit 8

When the SCI is transmitting 9-bit characters, T8 is the read/write ninth bit (bit 8) of the transmitted character. T8 is loaded into the transmit shift register at the same time that the SCDR is loaded into the transmit shift register. Reset has no effect on the T8 bit.

#### ORIE — Receiver Overrun Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the receiver overrun bit, OR.

- 1 = SCI error CPU interrupt requests from OR bit enabled
- 0 = SCI error CPU interrupt requests from OR bit disabled

#### NEIE — Receiver Noise Error Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the noise error bit, NE. Reset clears NEIE.

- 1 = SCI error CPU interrupt requests from NE bit enabled
- 0 = SCI error CPU interrupt requests from NE bit disabled

#### FEIE — Receiver Framing Error Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the framing error bit, FE. Reset clears FEIE.

- 1 = SCI error CPU interrupt requests from FE bit enabled
- 0 = SCI error CPU interrupt requests from FE bit disabled

#### PEIE — Receiver Parity Error Interrupt Enable Bit

This read/write bit enables SCI receiver CPU interrupt requests generated by the parity error bit, PE. Reset clears PEIE.

- 1 = SCI error CPU interrupt requests from PE bit enabled
- 0 = SCI error CPU interrupt requests from PE bit disabled




### 14.8.4 SCI Status Register 1

SCI status register 1 contains flags to signal the following conditions:

- Transfer of SCDR data to transmit shift register complete
- Transmission complete
- Transfer of receive shift register data to SCDR complete
- Receiver input idle
- Receiver overrun
- Noisy data
- Framing error
- Parity error

Address:	\$0016							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
Write:								
Reset:	1	1	0	0	0	0	0	0

 = Unimplemented

**Figure 14-12. SCI Status Register 1 (SCS1)**

#### SCTE — SCI Transmitter Empty Bit

This clearable, read-only bit is set when the SCDR transfers a character to the transmit shift register. SCTE can generate an SCI transmitter CPU interrupt request. When the SCTIE bit in SCC2 is set, SCTE generates an SCI transmitter CPU interrupt request. In normal operation, clear the SCTE bit by reading SCS1 with SCTE set and then writing to SCDR. Reset sets the SCTE bit.

- 1 = SCDR data transferred to transmit shift register
- 0 = SCDR data not transferred to transmit shift register

#### TC — Transmission Complete Bit

This read-only bit is set when the SCTE bit is set, and no data, preamble, or break character is being transmitted. TC generates an SCI transmitter CPU interrupt request if the TCIE bit in SCC2 is also set. TC is cleared automatically when data, preamble, or break is queued and ready to be sent. There may be up to 1.5 transmitter clocks of latency between queueing data, preamble, and break and the transmission actually starting. Reset sets the TC bit.

- 1 = No transmission in progress
- 0 = Transmission in progress

#### SCRF — SCI Receiver Full Bit

This clearable, read-only bit is set when the data in the receive shift register transfers to the SCI data register. SCRF can generate an SCI receiver CPU interrupt request. When the SCRIE bit in SCC2 is set the SCRF generates a CPU interrupt request. In normal operation, clear the SCRF bit by reading SCS1 with SCRF set and then reading the SCDR. Reset clears SCRF.

- 1 = Received data available in SCDR
- 0 = Data not available in SCDR

#### IDLE — Receiver Idle Bit

This clearable, read-only bit is set when 10 or 11 consecutive 1s appear on the receiver input. IDLE generates an SCI receiver CPU interrupt request if the ILIE bit in SCC2 is also set. Clear the IDLE bit by reading SCS1 with IDLE set and then reading the SCDR. After the receiver is enabled, it must

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

To protect status bits during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a two-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is at 0. After the break, doing the second step clears the status bit.

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

## 16.12 I/O Signals

The SPI module has four I/O pins and shares three of them with a parallel I/O port.

- MISO — Data received
- MOSI — Data transmitted
- SPCK — Serial clock
- $\overline{SS}$  — Slave select
- $V_{SS}$  — Clock ground

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  $V_{DD}$ .

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

MISO is one of the two SPI module pins that transmit 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 0 and its  $\overline{SS}$  pin is low. To support a multiple-slave system, a 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.

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

MOSI is one of the two SPI module pins that transmit 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.

Clearing the toggle-on-overflow bit, TOVx, inhibits output toggles on TIM overflows. Subsequent output compares try to force the output to a state it is already in and have no effect. The result is a 0% duty cycle output.

Setting the channel x maximum duty cycle bit (CHxMAX) and setting the TOVx bit generates a 100% duty cycle output (see 17.8.4 TIM Channel Status and Control Registers).

## 17.4 Interrupts

The following TIM sources can generate interrupt requests:

- TIM overflow flag (TOF) — The TOF bit is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. The TIM overflow interrupt enable bit, TOIE, enables TIM overflow CPU interrupt requests. TOF and TOIE are in the TIM status and control register.
- TIM channel flags (CH5F–CH0F) — The CHxF bit is set when an input capture or output compare occurs on channel x. Channel x TIM CPU interrupt requests are controlled by the channel x interrupt enable bit, CHxIE.

## 17.5 Low-Power Modes

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

### 17.5.1 Wait Mode

The TIM remains active after the execution of a WAIT instruction. In wait mode, the TIM registers are not accessible by the CPU. Any enabled CPU interrupt request from the TIM can bring the MCU out of wait mode.

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

### 17.5.2 Stop Mode

The TIM is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions or the state of the TIM counter. TIM operation resumes when the MCU exits stop mode.

## 17.6 TIM During Break Interrupts

A break interrupt stops the TIM counter and inhibits input captures.

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state (see Figure 15-19. SIM Break Flag Control Register (SBFCR)).

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

To protect status bits during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a 2-step read/write clearing procedure. If software does the first step on such a bit before the



## How to Reach Us:

### Home Page:

[www.freescale.com](http://www.freescale.com)

### E-mail:

[support@freescale.com](mailto:support@freescale.com)

### USA/Europe or Locations Not Listed:

Freescale Semiconductor  
Technical Information Center, CH370  
1300 N. Alma School Road  
Chandler, Arizona 85224  
+1-800-521-6274 or +1-480-768-2130  
[support@freescale.com](mailto:support@freescale.com)

### Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH  
Technical Information Center  
Schatzbogen 7  
81829 Muenchen, Germany  
+44 1296 380 456 (English)  
+46 8 52200080 (English)  
+49 89 92103 559 (German)  
+33 1 69 35 48 48 (French)  
[support@freescale.com](mailto:support@freescale.com)

### Japan:

Freescale Semiconductor Japan Ltd.  
Headquarters  
ARCO Tower 15F  
1-8-1, Shimo-Meguro, Meguro-ku,  
Tokyo 153-0064  
Japan  
0120 191014 or +81 3 5437 9125  
[support.japan@freescale.com](mailto:support.japan@freescale.com)

### Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.  
Technical Information Center  
2 Dai King Street  
Tai Po Industrial Estate  
Tai Po, N.T., Hong Kong  
+800 2666 8080  
[support.asia@freescale.com](mailto:support.asia@freescale.com)

### For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center  
P.O. Box 5405  
Denver, Colorado 80217  
1-800-441-2447 or 303-675-2140  
Fax: 303-675-2150  
[LDCForFreescaleSemiconductor@hibbertgroup.com](mailto:LDCForFreescaleSemiconductor@hibbertgroup.com)

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2005, 2006. All rights reserved.