MOTOROLA
intelligence everywhere™

digital dna™

MC68HC908JB16

Technical Data

M68HC08
Microcontrollers

MC68HC908JB16/D
Rev. 1, 5/2002

MOTOROLA.COM/SEMICONDUCTORS

MC68HC908JB16

Technical Data

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola 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 which may be provided in Motorola 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. Motorola does not convey any license under its patent rights nor the rights of others. Motorola 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 Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola 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 Motorola was negligent regarding the design or manufacture of the part. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

Revision History

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://motorola.com/semiconductors

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Revision History

Date	Revision Level	Description	Page Number(s)
May 2002	1	First general release.	—

List of Sections

List of Sections

Table of Contents

Section 1. General Description

Section 2. Memory Map

Section 3. Random-Access Memory (RAM)

Section 4. FLASH Memory

Section 5. Configuration Register (CONFIG)

Section 6. Central Processor Unit (CPU)

Section 7. Oscillator (OSC)

Section 8. System Integration Module (SIM)

Section 9. Monitor ROM (MON)

Section 10. Timer Interface Module (TIM)

Section 11. Universal Serial Bus Module (USB)

Section 12. Serial Communications Interface Module (SCI)

Section 13. Clock Generator Module (CGM)

Section 14. Input/Output (I/O) Ports

Section 15. External Interrupt (IRQ)

Section 16. Keyboard Interrupt Module (KBI)

Section 17. Computer Operating Properly (COP)

Section 18. Low-Voltage Inhibit (LVI)

Section 19. Break Module (BRK)

Section 20. Electrical Specifications

Section 21. Mechanical Specifications

Section 22. Ordering Information

List of Figures

List of Figures

Figure	Title	Page

List of Figures

Figure	Title	Page

List of Figures

List of Tables

List of Tables

Section 1. General Description

1.1 Contents

1.2 Introduction

The MC68HC908JB16 is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). The M68HC08 Family is based on the customer-specified integrated circuit (CSIC) design strategy.

1.3 Features

Features of the MC68HC908JB16 MCU include the following:

- High-performance M68HC08 architecture
- Fully upward-compatible object code with M6805, M146805, and M68HC05 families
- Low-power design; fully static with stop and wait modes
- 6-MHz internal bus frequency
- 16,384 bytes of on-chip FLASH memory with security[1] feature
- 384 bytes of on-chip random access memory (RAM)
- Up to 21 general-purpose input/output (I/O) pins, including:
  - 15 shared-function I/O pins
  - 8-bit keyboard interrupt port
  - 10 mA high current drive for PS/2 connection on 2 pins (with USB module disabled)
  - 1 dedicated I/O pin, with 25 mA direct drive for infrared LED (32-pin package)
  - 6 dedicated I/O pins, with 25 mA direct drive for infrared LED on 2 pins and 10 mA direct drive for normal LED on 4 pins (28-pin package)
- Two 16-bit, 2-channel timer interface modules (TIM1 and TIM2) with selectable input capture, output compare, PWM capability on each channel, and external clock input option (TCLK)
- Universal Serial Bus specification 2.0 low-speed functions:
  - 1.5 Mbps data rate
  - On-chip 3.3 V regulator
  - Endpoint 0 with 8-byte transmit buffer and 8-byte receive buffer
  - Endpoint 1 with 8-byte transmit buffer
  - Endpoint 2 with 8-byte transmit buffer and 8-byte receive buffer

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

- Serial communications interface module (SCI)
- Dual clock generator modules (CGM) (32-pin package)
- In-circuit programming capability using USB communication or standard serial link on PTA0 pin
- System protection features:
  - Optional computer operating properly (COP) reset
  - Optional Low-voltage detection with reset
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- Master reset pin with internal pull-up and power-on reset
- $\overline{\text{IRQ}}$ interrupt pin with internal pull-up and schmitt-trigger input
- 32-pin low-profile quad flat pack (LQFP) and 28-pin small outline integrated circuit package (SOIC)

Features of the CPU08 include the following:

- Enhanced HC05 programming model
- Extensive loop control functions
- 16 addressing modes (eight more than the HC05)
- 16-bit index register and stack pointer
- Memory-to-memory data transfers
- Fast 8 $\times$ 8 multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Third party C language support

1.4 MCU Block Diagram

Figure 1-1 shows the structure of the MC68HC908JB16.

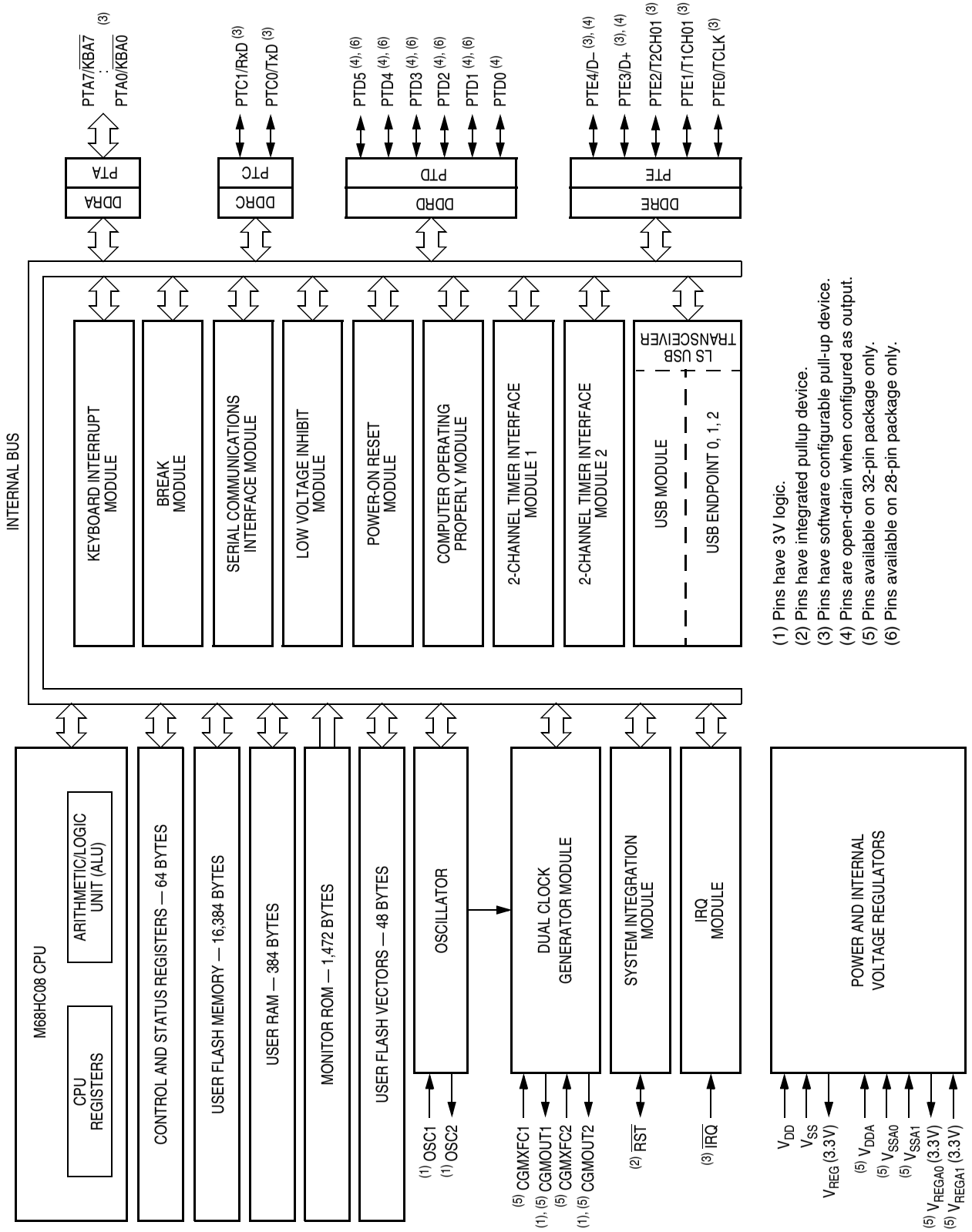
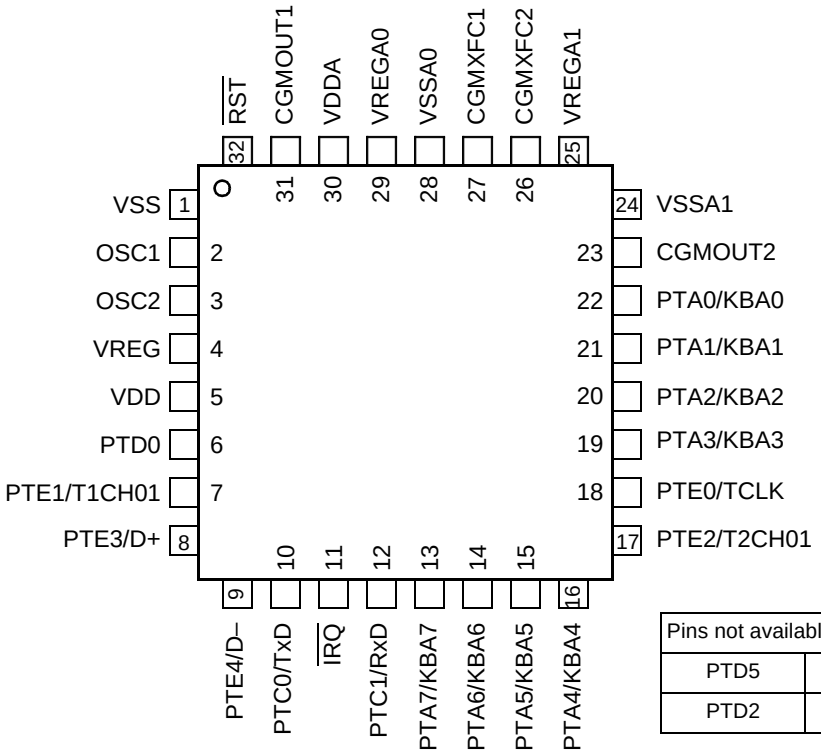


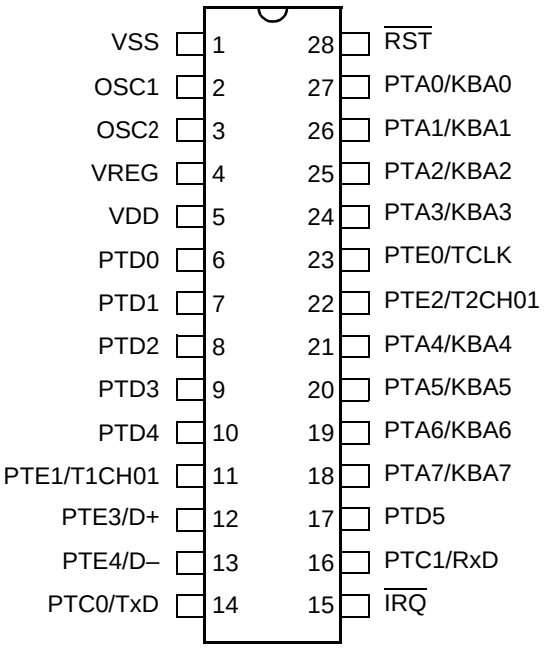
Figure 1-1. MC68HC908JB16 MCU Block Diagram

1.5 Pin Assignments



Pins not available on 32-pin package:		
PTD5	PTD4	PTD3
PTD2	PTD1	

Figure 1-2. 32-Pin LQFP Pin Assignment



Pins not available on 28-pin package:		
CGMXFC1	CGMXFC2	
CGMOUT1	CGMOUT2	
VREGA0	VREGA1	
VSSA0	VSSA1	VDDA

Figure 1-3. 28-Pin SOIC Pin Assignment

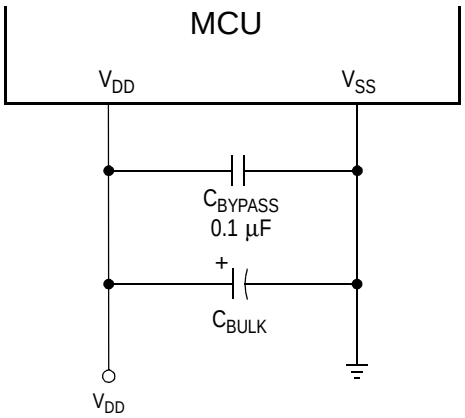
1.6 Pin Functions

Description of pin functions are provided here.

1.6.1 Power Supply Pins (V_{DD}, 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 **Figure 1-4** shows. Place the bypass capacitors as close to the MCU power pins as possible. Use high-frequency-response ceramic capacitors for C_{BYPASS}. C_{BULK} are optional bulk current bypass capacitors for use in applications that require the port pins to source high current levels.



NOTE: Values shown are typical values.

Figure 1-4. Power Supply Bypassing

1.6.2 Voltage Regulator Output Pin (V_{REG})

V_{REG} is the 3.3V output of the on-chip voltage regulator. V_{REG} is used internally for the MCU operation and the USB data driver. It is also used to supply the voltage for the external pullup resistor required on the USB's D– line. The V_{REG} pin requires an external bulk capacitor 4.7µF or larger and a 0.1µF ceramic bypass capacitor as **Figure 1-5** shows. Place the bypass capacitors as close to the V_{REG} pin as possible.

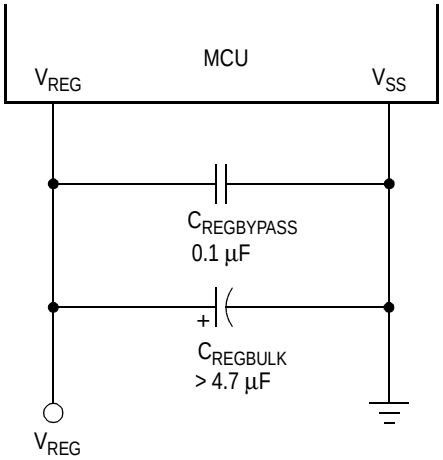


Figure 1-5. Regulator Supply Capacitor Configuration

1.6.3 Oscillator Pins (OSC1 and OSC2)

The OSC1 and OSC2 pins are the connections for the on-chip oscillator circuit.

1.6.4 External Reset Pin ($\overline{RST}$)

A logic zero on the $\overline{RST}$ pin forces the MCU to a known start-up state. $\overline{RST}$ is bidirectional, allowing a reset of the entire system. It is driven low when any internal reset source is asserted. The $\overline{RST}$ pin contains an internal pullup device to V_{DD}. (See **Section 8. System Integration Module (SIM)**.)

1.6.5 External Interrupt Pins ($\overline{IRQ}$, PTE4/D–)

$\overline{IRQ}$ is an asynchronous external interrupt pin. $\overline{IRQ}$ is also the pin to enter Monitor mode. The $\overline{IRQ}$ pin contains a software configurable pullup device to V_{DD}. PTE4/D– can be programmed to trigger the IRQ interrupt. (See **Section 15. External Interrupt (IRQ)**.)

1.6.6 CGM Power Supply Pins (V_{DDA}, V_{SSA0}, V_{SSA1})

V_{DDA} is the power supply pin, V_{SSA0} and V_{SSA1} are the ground pins for the analog portion of the clock generator modules (CGMs). Connect V_{DDA} to the same voltage potential as V_{DD}. Connect V_{SSA0} and V_{SSA1} pins to the same voltage potential as V_{SS}. Decoupling of these pins should be as per the digital supply.

1.6.7 CGM Voltage Regulator Out (V_{REGA0})

V_{REGA0} is the 3.3V output of the second on-chip voltage regulator. V_{REGA0} is used for CGM1 and CGM2 operation. Decoupling of this pin should be as per the digital V_{REG}.

1.6.8 CGM Voltage Regulator In (V_{REGA1})

V_{REGA1} is the 3.3V input pin for CGM2. Connect V_{REGA1} directly to V_{REGA0}. Decoupling of V_{REGA1} pin should be as per V_{REGA0}.

1.6.9 External Filter Capacitor Pins (CGMXFC1, CGMXFC2)

CGMXFC1 and CGMXFC2 are external capacitor connections for the respective CGMs.

1.6.10 CGM Clock Output Pins (CGMOUT1, CGMOUT2)

CGMOUT1 and CGMOUT2 are buffered VCO outputs of the respective CGMs.

1.6.11 Port A Input/Output (I/O) Pins (PTA7/$\overline{\text{KBA7}}$–PTA0/$\overline{\text{KBA0}}$)

PTA7/$\overline{\text{KBA7}}$–PTA0/$\overline{\text{KBA0}}$ are general-purpose bidirectional I/O port pins. (See **Section 14. Input/Output (I/O) Ports**.) Each pin contains a software configurable pullup device to V_{DD} when the pin is configured as an input. (See **14.7 Port Options**.) Each pin can also be programmed as an external keyboard interrupt pin. (See **Section 16. Keyboard Interrupt Module (KBI)**.)

1.6.12 Port C I/O Pins (PTC1/RxD, PTC0/TxD)

Port C is a 2-bit special function port that shares its pins with the SCI module. (See **Section 14. Input/Output (I/O) Ports**.) Each pin contains a software configurable pullup device to V_{DD} when the pin is configured as an input. (See **14.7 Port Options**.)

1.6.13 Port D I/O Pins (PTD5–PTD0)

PTD5–PTD0 are general-purpose bidirectional I/O port pins; open-drain when configured as output. (See **Section 14. Input/Output (I/O) Ports**.) PTD5–PTC2 are software configurable to be 10mA sink pins for direct LED connections. PTD1–PTD0 are software configurable to be 25mA sink pins for direct infrared LED connections. (See **14.7 Port Options**.)

1.6.14 Port E I/O Pins (PTE4/D–, PTE3/D+, PTE2/T2CH01, PTE1/T1CH01, PTE0/TCLK)

Port E is a 5-bit special function port that shares two of its pins with the USB module and three of its pins with the two timer interface modules.

Each PTE2–PTE0 pin contains a software configurable pullup device to V_{DD} when the pin is configured as an input or output.

When the USB module is disabled, the PTE4 and PTE3 pins are general-purpose bidirectional I/O port pins with 10mA sink capability. Each pin is open-drain when configured as an output; and each pin contains a software configurable 5kΩ pullup to V_{DD} when configured as an input**.** The PTE4 pin can also be enabled to trigger the IRQ interrupt.

When the USB module is enabled, the PTE4/D– and PTE3/D+ pins become the USB module D– and D+ pins. The USB D– pin contains a 1.5kΩ software configurable pullup device to V_{REG}. (See **Section 10. Timer Interface Module (TIM)**, **Section 11. Universal Serial Bus Module (USB)** and **Section 14. Input/Output (I/O) Ports**.)

NOTE: *Any unused inputs and I/O ports should be tied to an appropriate logic level (either V_{DD} or V_{SS}). Although the I/O ports of the MC68HC908JB16 do not require termination, termination is recommended to reduce the possibility of static damage.*

Summary of the pin functions are provided in **Table 1-1**.

Table 1-1. Summary of Pin Functions

PIN NAME	PIN DESCRIPTION	IN/OUT	VOLTAGE LEVEL
V_{DD}	Power supply.	IN	4.0 to 5.5V
V_{SS}	Power supply ground.	OUT	0V
V_{REG}	3.3V regulated output from MCU.	OUT	V_{REG} (3.3V)
$\overline{RST}$	Reset input, active low. With internal pull-up and schmitt trigger input.	IN/OUT	V_{DD}
$\overline{IRQ}$	External IRQ pin; with programmable internal pull-up and schmitt trigger input.	IN	V_{DD}
	Used for mode entry selection.	IN	V_{REG} to V_{TST}
OSC1	Crystal oscillator input.	IN	V_{REG}
OSC2	Crystal oscillator output; inverting of OSC1 signal.	OUT	V_{REG}
V_{DDA}(1)	Analog power supply.	IN	4.0 to 5.5V
V_{SSA0}(1) V_{SSA1}(1)	Analog power supply ground.	OUT	0V
V_{REGA0}(1)	3.3V regulated output from MCU.	OUT	V_{REGA0} (3.3V)
V_{REGA1}(1)	3.3V input for CGM2.	IN	V_{REGA0}
CGMXFC1(1)	CGM1 external filter capacitor connection.	OUT	V_{REGA0}
CGMXFC2(1)	CGM2 external filter capacitor connection.	OUT	V_{REGA0}
CGMOUT1(1)	CGM1 clock output.	OUT	V_{REGA0}
CGMOUT2(1)	CGM2 clock output.	OUT	V_{REGA0}
PTA0/$\overline{KBA0}$: PTA7/$\overline{KBA7}$	8-bit general purpose I/O port.	IN/OUT	V_{DD}
	Pins as keyboard interrupts, $\overline{KBA0}$–$\overline{KBA7}$.	IN	V_{DD}
	Each pin has programmable internal pullup when configured as input.	IN	V_{DD}

Table 1-1. Summary of Pin Functions

PIN NAME	PIN DESCRIPTION	IN/OUT	VOLTAGE LEVEL
PTC0/TxD PTC1/RxD	2-bit general purpose I/O port.	IN/OUT	V_{DD}
	Each pin has programmable internal pull-up device.	IN	V_{DD}
	PTC0 as TxD of SCI module.	OUT	V_{DD}
	PTC1 as RxD of SCI module.	IN	V_{DD}
PTD0–PTD5(2)	6-bit general purpose I/O port; open-drain when configured as output.	IN OUT	V_{DD} V_{REG} or V_{DD}
	PTD0–PTD1 have configurable 25mA sink for infrared LED.	OUT	V_{REG} or V_{DD}
	PTD2–PTD5 have configurable 10mA sink for LED.	OUT	V_{REG} or V_{DD}
PTE0/TCLK PTE1/T1CH01 PTE2/T2CH01	PTE0–PTE2 are general purpose I/O lines.	IN/OUT	V_{DD}
	PTE0–PTE2 have programmable internal pullup when configured as input or output.	IN/OUT	V_{DD}
	PTE0 as TCLK of TIM1 and TIM2.	IN	V_{DD}
	PTE1 as T1CH01 of TIM1.	IN/OUT	V_{DD}
	PTE2 as T2CH01 of TIM2.	IN/OUT	V_{DD}
PTE3/D+ PTE4/D–	PTE3–PTE4 general purpose I/O lines; open-drain when configured as output.	IN OUT	V_{DD} V_{REG} or V_{DD}
	PTE3–PTE4 have programmable internal pullup when configured as input.	IN	V_{DD}
	PTE3 as D+ of USB module.	IN/OUT	V_{REG}
	PTE4 as D– of USB module.	IN/OUT	V_{REG}
	PTE4 as additional IRQ interrupt.	IN	V_{DD}

Notes:
1. Pin available on 32-pin package only.
2. PTD[5:1] pins available on 28-pin package only.

General Description

Section 2. Memory Map

2.1 Contents

2.2 Introduction

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

- 16,384 bytes of FLASH memory
- 384 bytes of random-access memory (RAM)
- 48 bytes of user-defined vectors
- 1,024 + 448 bytes of monitor ROM

2.3 Unimplemented Memory Locations

Accessing an unimplemented location can cause an illegal address reset if illegal address resets are enabled. In the memory map (**Figure 2-1**) and in register figures in this document, unimplemented locations are shaded.

2.4 Reserved Memory Locations

Accessing a reserved location can have unpredictable effects on MCU operation. In the **Figure 2-1** and in register figures in this document, reserved locations are marked with the word Reserved or with the letter R.

2.5 Input/Output (I/O) Section

Most of the control, status, and data registers are in the zero page area of $0000–$007F. Additional I/O registers have these addresses:

- $FE00; SIM break status register, SBSR
- $FE01; SIM reset status register, SRSR
- $FE02; Reserved
- $FE03; SIM break flag control register, SBFCR
- $FE04; Interrupt status register 1, INT1
- $FE05; Interrupt status register 2, INT2
- $FE06; Reserved
- $FE07; Reserved
- $FE08; FLASH control register, FLCR
- $FE09; FLASH block protect register, FLBPR
- $FE0A; Reserved
- $FE0B; Reserved
- $FE0C; Break address register high, BRKH
- $FE0D; Break address register low, BRKL
- $FE0E; Break status and control register, BRKSCR
- $FE0F; Reserved
- $FFFF; COP control register, COPCTL

Data registers are shown in **Figure 2-2**. **Table 2-1** is a list of vector locations.

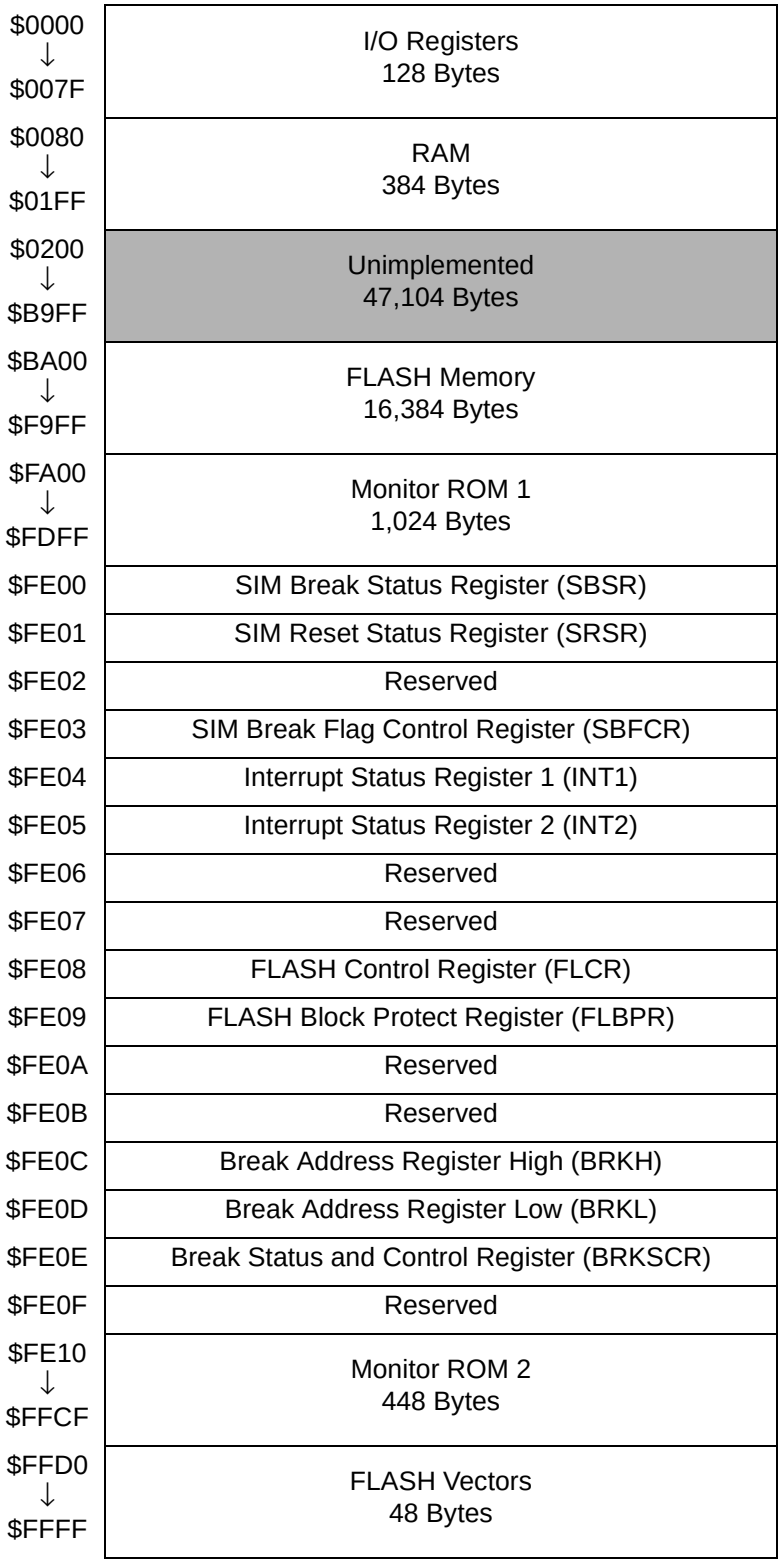


Figure 2-1. Memory Map

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA)	Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Reset:	Unaffected by reset							
$0001	Reserved	Read: Write:	R	R	R	R	R	R	R	R
		Reset:								
$0002	Port C Data Register (PTC)	Read:	0	0	0	0	0	0	PTC1	PTC0
		Write:								
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PTD)	Read:	0	0	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Write:								
		Reset:	Unaffected by reset							
$0004	Data Direction Register A (DDRA)	Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Reset:	0*	0	0	0	0	0	0	0

* DDRA7 bit is reset by POR or LVI reset only.

$0005	Reserved	Read: Write:	R	R	R	R	R	R	R	R
		Reset:								
$0006	Data Direction Register C (DDRC)	Read:	0	0	0	0	0	0	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007	Data Direction Register D (DDRD)	Read:	0	0	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0008	Port E Data Register (PTE)	Read:	0	0	0	PTE4	PTE3	PTE2	PTE1	PTE0
		Write:								
		Reset:	Unaffected by reset							
$0009	Data Direction Register E (DDRE)	Read:	0	0	0	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 1 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000A	Timer 1 Status and Control Register (T1SC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$000B	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$000C	Timer 1 Counter Register High (T1CNTH)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000D	Timer 1 Counter Register Low (T1CNTL)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000E	Timer 1 Counter Modulo Register High (T1MODH)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$000F	Timer 1 Counter Modulo Register Low (T1MODL)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0010	Timer 1 Channel 0 Status and Control Register (T1SC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0011	Timer 1 Channel 0 Register High (T1CH0H)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	Indeterminate after reset							
$0012	Timer 1 Channel 0 Register Low (T1CH0L)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	Indeterminate after reset							
$0013	Timer 1 Channel 1 Status and Control Register (T1SC1)	Read:	CH1F	CH1IE	CH01IE	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 2 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0014	Timer 1 Channel 1 Register High (T1CH1H)	Read: Write:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Reset:	Indeterminate after reset							
$0015	Timer 1 Channel 1 Register Low (T1CH1L)	Read: Write:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Reset:	Indeterminate after reset							
$0016	Keyboard Status and Control Register (KBSCR)	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$0017	Keyboard Interrupt Enable Register (KBIER)	Read: Write:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Reset:	0	0	0	0	0	0	0	0
$0018	USB Interrupt Register 2 (UIR2)	Read:	0	0	0	0	0	0	0	0
		Write:	EOPFR	RSTFR	TXD2FR	RXD2FR	TDX1FR	RESUMFR	TXD0FR	RXD0FR
		Reset:	0	0	0	0	0	0	0	0
$0019	USB Control Register 2 (UCR2)	Read: Write:	T2SEQ	STALL2	TX2E	RX2E	TP2SIZ3	TP2SIZ2	TP2SIZ1	TP2SIZ0
		Reset:	0	0	0	0	0	0	0	0
$001A	USB Control Register 3 (UCR3)	Read:	TX1ST	0	OSTALL0	ISTALL0	0	PULLEN	ENABLE2	ENABLE1
		Write:		TX1STR						
		Reset:	0	0	0	0	0	0*	0	0

* PULLEN bit is reset by POR or LVI reset only.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001B	USB Control Register 4 (UCR4)	Read:	0	0	0	0	0	FUSBO	FDP	FDM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001C	IRQ Option Control Register (IOCR)	Read:	0	0	0	0	0	PTE4IF	PTE4IE	IRQPD
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001D	Port Option Control Register (POCR)	Read:	PTE20P	PTDLDD	PTDILDD	PTE4P	PTE3P	PCP	R	PAP
		Write:								
		Reset:	0	0	0	0	0	0	0	0

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 3 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001E	IRQ Status and Control Register (INTSCR)	Read:	0	0	0	0	IRQF	0	IMASK	MODE
		Write:						ACK		
		Reset:	0	0	0	0	0	0	0	0
$001F	Configuration Register (CONFIG)†	Read:	LVIDR	LVI5OR3	URSTD	LVID	SSREC	COPRS	STOP	COPD
		Write:								
		Reset:	0*	0*	0*	0*	0	0	0	0

† One-time writable register after each reset.
* LVIDR, LVI5OR3, URSTD, and LVID, are reset by POR or LVI reset only.

$0020	USB Endpoint 0 Data Register 0 (UE0D0)	Read:	UE0R07	UE0R06	UE0R05	UE0R04	UE0R03	UE0R02	UE0R01	UE0R00
		Write:	UE0T07	UE0T06	UE0T05	UE0T04	UE0T03	UE0T02	UE0T01	UE0T00
		Reset:	Unaffected by reset							
$0021	USB Endpoint 0 Data Register 1 (UE0D1)	Read:	UE0R17	UE0R16	UE0R15	UE0R14	UE0R13	UE0R12	UE0R11	UE0R10
		Write:	UE0T17	UE0T16	UE0T15	UE0T14	UE0T13	UE0T12	UE0T11	UE0T10
		Reset:	Unaffected by reset							
$0022	USB Endpoint 0 Data Register 2 (UE0D2)	Read:	UE0R27	UE0R26	UE0R25	UE0R24	UE0R23	UE0R22	UE0R21	UE0R20
		Write:	UE0T27	UE0T26	UE0T25	UE0T24	UE0T23	UE0T22	UE0T21	UE0T20
		Reset:	Unaffected by reset							
$0023	USB Endpoint 0 Data Register 3 (UE0D3)	Read:	UE0R37	UE0R36	UE0R35	UE0R34	UE0R33	UE0R32	UE0R31	UE0R30
		Write:	UE0T37	UE0T36	UE0T35	UE0T34	UE0T33	UE0T32	UE0T31	UE0T30
		Reset:	Unaffected by reset							
$0024	USB Endpoint 0 Data Register 4 (UE0D4)	Read:	UE0R47	UE0R46	UE0R45	UE0R44	UE0R43	UE0R42	UE0R41	UE0R40
		Write:	UE0T47	UE0T46	UE0T45	UE0T44	UE0T43	UE0T42	UE0T41	UE0T40
		Reset:	Unaffected by reset							
$0025	USB Endpoint 0 Data Register 5 (UE0D5)	Read:	UE0R57	UE0R56	UE0R55	UE0R54	UE0R53	UE0R52	UE0R51	UE0R50
		Write:	UE0T57	UE0T56	UE0T55	UE0T54	UE0T53	UE0T52	UE0T51	UE0T50
		Reset:	Unaffected by reset							
$0026	USB Endpoint 0 Data Register 6 (UE0D6)	Read:	UE0R67	UE0R66	UE0R65	UE0R64	UE0R63	UE0R62	UE0R61	UE0R60
		Write:	UE0T67	UE0T66	UE0T65	UE0T64	UE0T63	UE0T62	UE0T61	UE0T60
		Reset:	Unaffected by reset							
$0027	USB Endpoint 0 Data Register 7 (UE0D7)	Read:	UE0R77	UE0R76	UE0R75	UE0R74	UE0R73	UE0R72	UE0R71	UE0R70
		Write:	UE0T77	UE0T76	UE0T75	UE0T74	UE0T73	UE0T72	UE0T71	UE0T70
		Reset:	Unaffected by reset							

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0028	USB Endpoint 1 Data Register 0 (UE1D0)	Read:								
		Write:	UE1T07	UE1T06	UE1T05	UE1T04	UE1T03	UE1T02	UE1T01	UE1T00
		Reset:	Unaffected by reset							
$0029	USB Endpoint 1 Data Register 1 (UE1D1)	Read:								
		Write:	UE1T17	UE1T16	UE1T15	UE1T14	UE1T13	UE1T12	UE1T11	UE1T10
		Reset:	Unaffected by reset							
$002A	USB Endpoint 1 Data Register 2 (UE1D2)	Read:								
		Write:	UE1T27	UE1T26	UE1T25	UE1T24	UE1T23	UE1T22	UE1T21	UE1T20
		Reset:	Unaffected by reset							
$002B	USB Endpoint 1 Data Register 3 (UE1D3)	Read:								
		Write:	UE1T37	UE1T36	UE1T35	UE1T34	UE1T33	UE1T32	UE1T31	UE1T30
		Reset:	Unaffected by reset							
$002C	USB Endpoint 1 Data Register 4 (UE1D4)	Read:								
		Write:	UE1T47	UE1T46	UE1T45	UE1T44	UE1T43	UE1T42	UE1T41	UE1T40
		Reset:	Unaffected by reset							
$002D	USB Endpoint 1 Data Register 5 (UE1D5)	Read:								
		Write:	UE1T57	UE1T56	UE1T55	UE1T54	UE1T53	UE1T52	UE1T51	UE1T50
		Reset:	Unaffected by reset							
$002E	USB Endpoint 1 Data Register 6 (UE1D6)	Read:								
		Write:	UE1T67	UE1T66	UE1T65	UE1T64	UE1T63	UE1T62	UE1T61	UE1T60
		Reset:	Unaffected by reset							
$002F	USB Endpoint 1 Data Register 7 (UE1D7)	Read:								
		Write:	UE1T77	UE1T76	UE1T75	UE1T74	UE1T73	UE1T72	UE1T71	UE1T70
		Reset:	Unaffected by reset							
$0030	USB Endpoint 2 Data Register 0 (UE2D0)	Read:	UE2R07	UE2R06	UE2R05	UE2R04	UE2R03	UE2R02	UE2R01	UE2R00
		Write:	UE2T07	UE2T06	UE2T05	UE2T04	UE2T03	UE2T02	UE2T01	UE2T00
		Reset:	Unaffected by reset							
$0031	USB Endpoint 2 Data Register 1 (UE2D1)	Read:	UE2R17	UE2R16	UE2R15	UE2R14	UE2R13	UE2R12	UE2R11	UE2R10
		Write:	UE2T17	UE2T16	UE2T15	UE2T14	UE2T13	UE2T12	UE2T11	UE2T10
		Reset:	Unaffected by reset							

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 5 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0032	USB Endpoint 2 Data Register 2 (UE2D2)	Read:	UE2R27	UE2R26	UE2R25	UE2R24	UE2R23	UE2R22	UE2R21	UE2R20
		Write:	UE2T27	UE2T26	UE2T25	UE2T24	UE2T23	UE2T22	UE2T21	UE2T20
		Reset:	Unaffected by reset							
$0033	USB Endpoint 2 Data Register 3 (UE2D3)	Read:	UE2R37	UE2R36	UE2R35	UE2R34	UE2R33	UE2R32	UE2R31	UE2R30
		Write:	UE2T37	UE2T36	UE2T35	UE2T34	UE2T33	UE2T32	UE2T31	UE2T30
		Reset:	Unaffected by reset							
$0034	USB Endpoint 2 Data Register 4 (UE2D4)	Read:	UE2R47	UE2R46	UE2R45	UE2R44	UE2R43	UE2R42	UE2R41	UE2R40
		Write:	UE2T47	UE2T46	UE2T45	UE2T44	UE2T43	UE2T42	UE2T41	UE2T40
		Reset:	Unaffected by reset							
$0035	USB Endpoint 2 Data Register 5 (UE2D5)	Read:	UE2R57	UE2R56	UE2R55	UE2R54	UE2R53	UE2R52	UE2R51	UE2R50
		Write:	UE2T57	UE2T56	UE2T55	UE2T54	UE2T53	UE2T52	UE2T51	UE2T50
		Reset:	Unaffected by reset							
$0036	USB Endpoint 2 Data Register 6 (UE2D6)	Read:	UE2R67	UE2R66	UE2R65	UE2R64	UE2R63	UE2R62	UE2R61	UE2R60
		Write:	UE2T67	UE2T66	UE2T65	UE2T64	UE2T63	UE2T62	UE2T61	UE2T60
		Reset:	Unaffected by reset							
$0037	USB Endpoint 2 Data Register 7 (UE2D7)	Read:	UE2R77	UE2R76	UE2R75	UE2R74	UE2R73	UE2R72	UE2R71	UE2R70
		Write:	UE2T77	UE2T76	UE2T75	UE2T74	UE2T73	UE2T72	UE2T71	UE2T70
		Reset:	Unaffected by reset							
$0038	USB Address Register (UADDR)	Read:	USBEN	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0
		Write:								
		Reset:	0*	0	0	0	0	0	0	0

* USBEN bit is reset by POR or LVI reset only.

$0039	USB Interrupt Register 0 (UIR0)	Read:	EOPIE	SUSPND	TXD2IE	RXD2IE	TXD1IE	0	TXD0IE	RXD0IE
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$003A	USB Interrupt Register 1 (UIR1)	Read:	EOPF	RSTF	TXD2F	RXD2F	TXD1F	RESUMF	TXD0F	RXD0F
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$003B	USB Control Register 0 (UCR0)	Read:	T0SEQ	0	TX0E	RX0E	TP0SIZ3	TP0SIZ2	TP0SIZ1	TP0SIZ0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 6 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$003C	USB Control Register 1 (UCR1)	Read: Write:	T1SEQ	STALL1	TX1E	FRESUM	TP1SIZ3	TP1SIZ2	TP1SIZ1	TP1SIZ0
		Reset:	0	0	0	0	0	0	0	0
$003D	USB Status Register 0 (USR0)	Read:	R0SEQ	SETUP	0	0	RP0SIZ3	RP0SIZ2	RP0SIZ1	RP0SIZ0
		Write:								
		Reset:	Unaffected by reset							
$003E	USB Status Register 1 (USR1)	Read:	R2SEQ	TXACK	TXNAK	TXSTL	RP2SIZ3	RP2SIZ2	RP2SIZ1	RP2SIZ0
		Write:								
		Reset:	U	0	0	0	U	U	U	U
$003F	Unimplemented	Read: Write:								
$0040	Timer 2 Status and Control Register (T2SC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0041	Reserved	Read: Write:	R	R	R	R	R	R	R	R
		Reset:								
$0042	Timer 2 Counter Register High (T2CNTH)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0043	Timer 2 Counter Register Low (T2CNTL)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0044	Timer 2 Counter Modulo Register High (T2MODH)	Read: Write:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Reset:	1	1	1	1	1	1	1	1
$0045	Timer 2 Counter Modulo Register Low (T2MODL)	Read: Write:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Reset:	1	1	1	1	1	1	1	1

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 7 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0046	Timer 2 Channel 0 Status and Control Register (T2SC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0047	Timer 2 Channel 0 Register High (T2CH0H)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	Indeterminate after reset							
$0048	Timer 2 Channel 0 Register Low (T2CH0L)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	Indeterminate after reset							
$0049	Timer 2 Channel 1 Status and Control Register (T2SC1)	Read:	CH1F	CH1IE	CH01IE	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$004A	Timer 2 Channel 1 Register High (T2CH1H)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	Indeterminate after reset							
$004B	Timer 2 Channel 1 Register Low (T2CH1L)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	Indeterminate after reset							
$004C	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$004D	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$004E	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$004F	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 8 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0050	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$0051	PLL Bandwidth Control Register (PBWC)	Read:	R	LOCK1	R	PLLON1	R	LOCK2	R	PLLON2
		Write:								
		Reset:		0		0		0		0
$0052	VCO Control Register (PVCR)	Read:	VCO_7	VCO_6	VCO_5	VCO_4	VCO_3	VCO_2	VCO_1	VCO_0
		Write:								
		Reset:	0	0	1	1	0	0	0	0
$0053	PLL1 N & R Divider Select Register High (PNRH1)	Read:	VDS1_11	VDS1_10	VDS1_9	VDS1_8	0	0	RDS1_9	RDS1_8
		Write:								
		Reset:	0	0	1	0	0	0	0	0
$0054	PLL1 N Divider Select Register Low (PNSL1)	Read:	VDS1_7	VDS1_6	VDS1_5	VDS1_4	VDS1_3	VDS1_2	VDS1_1	VDS1_0
		Write:								
		Reset:	0	1	1	1	1	1	0	1
$0055	PLL1 R Divider Select Register Low (PRSL1)	Read:	RDS1_7	RDS1_6	RDS1_5	RDS1_4	RDS1_3	RDS1_2	RDS1_1	RDS1_0
		Write:								
		Reset:	1	0	0	1	0	0	0	0
$0056	PLL2 N & R Divider Select Register High (PNRH2)	Read:	VDS2_11	VDS2_10	VDS2_9	VDS2_8	0	0	RDS2_9	RDS2_8
		Write:								
		Reset:	0	0	1	0	0	0	0	0
$0057	PLL2 N Divider Select Register Low (PNSL1)	Read:	VDS2_7	VDS2_6	VDS2_5	VDS2_4	VDS2_3	VDS2_2	VDS2_1	VDS2_0
		Write:								
		Reset:	0	1	1	1	1	1	0	1
$0058	PLL2 R Divider Select Register Low (PRSL2)	Read:	RDS2_7	RDS2_6	RDS2_5	RDS2_4	RDS2_3	RDS2_2	RDS2_1	RDS2_0
		Write:								
		Reset:	1	0	0	1	0	0	0	0
$0059	Phase Detector Control Register (PDCR)	Read:	PHD_7	PHD_6	PHD_5	PHD_4	PHD_3	PHD_2	PHD_1	PHD_0
		Write:								
		Reset:	1	0	0	1	0	0	0	0

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 9 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$005A	SCI Control Register 1 (SCC1)	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$005B	SCI Control Register 2 (SCC2)	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$005C	SCI Control Register 3 (SCC3)	Read:	R8	T8	DMARE	DMATE	ORIE	NEIE	FEIE	PEIE
		Write:								
		Reset:	U	U	0	0	0	0	0	0
$005D	SCI Status Register 1 (SCS1)	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:								
		Reset:	1	1	0	0	0	0	0	0
$005E	SCI Status Register 2 (SCS2)	Read:	0	0	0	0	0	0	BKF	RPF
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$005F	SCI Data Register (SCDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	U	U	U	U	U	U	U	U
$0060	SCI Baud Rate Register (SCBR)	Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:								
		Reset:	0	0	0	0		0	0	0
$0061	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$0062	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$0063	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 10 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0064 to $007F	Unimplemented	Read:								
		Write:								
		Reset:								
$FE00	SIM Break Status Register (SBSR)	Read:	R	R	R	R	R	R	SBSW	R
		Write:							Note	
		Reset:							0	
$FE01	SIM Reset Status Register (SRSR)	Read:	POR	PIN	COP	ILOP	ILAD	USB	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
$FE02	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$FE03	SIM Break Flag Control Register (SBFCR)	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$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	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$FE07	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								

Note: Writing a logic 0 clears SBSW.

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 11 of 12)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE08	FLASH Control Register (FLCR)	Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE09	FLASH Block Protect Register (FLBPR)	Read:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0A	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$FE0B	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$FE0C	Break Address High Register (BRKH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0D	Break Address Low Register (BRKL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0E	Break Status and Control Register (BRKSCR)	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FFFF	COP Control Register (COPCTL)	Read:	Low byte of reset vector							
		Write:	Clears COP counter (any value)							
		Reset:	Unaffected by reset							

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 12 of 12)

Table 2-1. Vector Addresses

Vector Priority	Vector	Address	Vector
Lowest	IF14	$FFE0	Keyboard Vector (High)
		$FFE1	Keyboard Vector (Low)
	IF13	$FFE2	SCI Transmit Vector (High)
		$FFE3	SCI Transmit Vector (Low)
	IF12	$FFE4	SCI Receive Vector (High)
		$FFE5	SCI Receive Vector (Low)
	IF11	$FFE6	SCI Error Vector (High)
		$FFE7	SCI Error Vector (Low)
	IF10	$FFE8	TIM2 Overflow Vector (High)
		$FFE9	TIM2 Overflow Vector (Low)
	IF9	$FFEA	TIM2 Channel 0 and 1 Vector (High)
		$FFEB	TIM2 Channel 0 and 1 Vector (Low)
	IF8	$FFEC	TIM2 Channel 1 Vector (High)
		$FFED	TIM2 Channel 1 Vector (Low)
	IF7	$FFEE	TIM2 Channel 0 Vector (High)
		$FFEF	TIM2 Channel 0 Vector (Low)
	IF6	$FFF0	TIM1 Overflow Vector (High)
		$FFF1	TIM1 Overflow Vector (Low)
	IF5	$FFF2	TIM1 Channel 0 and 1 Vector (High)
		$FFF3	TIM1 Channel 0 and 1 Vector (Low)
	IF4	$FFF4	TIM1 Channel 1 Vector (High)
		$FFF5	TIM1 Channel 1 Vector (Low)
	IF3	$FFF6	TIM1 Channel 0 Vector (High)
		$FFF7	TIM1 Channel 0 Vector (Low)
	IF2	$FFF8	IRQ Vector (High)
		$FFF9	IRQ Vector (Low)
	IF1	$FFFA	USB Vector (High)
		$FFFB	USB Vector (Low)
	—	$FFFC	SWI Vector (High)
		$FFFD	SWI Vector (Low)
	—	$FFFE	Reset Vector (High)
Highest		$FFFF	Reset Vector (Low)

Section 3. Random-Access Memory (RAM)

3.1 Contents

3.2 Introduction

This section describes the 384 bytes of RAM (random-access memory).

3.3 Functional Description

Addresses $0080 through $01FF are RAM locations. The location of the stack RAM is programmable. The 16-bit stack pointer allows the stack to be anywhere in the 64K-byte memory space.

NOTE: *For correct operation, the stack pointer must point only to RAM locations.*

Within page zero are 128 bytes of RAM. Because the location of the stack RAM is programmable, all page zero RAM locations can be used for I/O control and user data or code. When the stack pointer is moved from its reset location at $00FF out of page zero, direct addressing mode instructions can efficiently access all page zero RAM locations. Page zero RAM, therefore, provides ideal locations for frequently accessed global variables.

Before processing an interrupt, the CPU uses five bytes of the stack to save the contents of the CPU registers.

NOTE: *For M6805 compatibility, the H register is not stacked.*

During a subroutine call, the CPU uses two bytes of the stack to store the return address. The stack pointer decrements during pushes and increments during pulls.

NOTE: *Be careful when using nested subroutines. The CPU may overwrite data in the RAM during a subroutine or during the interrupt stacking operation.*

Section 4. FLASH Memory

4.1 Contents

4.2 Introduction

This section describes the operation of the embedded FLASH memory. This memory can be read, programmed, and erased from a single external supply. The program and erase operations are enabled through the use of an internal charge pump.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE08	FLASH Control Register (FLCR)	Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE09	FLASH Block Protect Register (FLBPR)	Read:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 4-1. FLASH I/O Register Summary

4.3 Functional Description

The FLASH memory consists of an array of 16,384 bytes for user memory plus a block of 48 bytes for user interrupt vectors. *An erased bit reads as logic 1 and a programmed bit reads as a logic 0.* The FLASH memory is block erasable. The minimum erase block size is 512 bytes. Program and erase operation operations are facilitated through control bits in FLASH Control Register (FLCR).The address ranges for the FLASH memory are shown as follows:

- $BA00–$F9FF (user memory, 16,384 bytes)
- $FFD0–$FFFF (user interrupt vectors, 48 bytes)

Programming tools are available from Motorola. Contact your local Motorola representative for more information.

NOTE: *A security feature prevents viewing of the FLASH contents.*[1]

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

4.4 FLASH Control Register

The FLASH control register (FLCR) controls FLASH program and erase operation.

Address: $FE08

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 4-2. FLASH Control Register (FLCR)

HVEN — High Voltage Enable Bit

This read/write bit enables high voltage from the charge pump to the memory for either program or erase operation. It can only be set if either PGM=1 or ERASE=1 and the sequence for erase or program/verify is followed.

1 = High voltage enabled to array and charge pump on
0 = High voltage disabled to array and charge pump off

MASS — Mass Erase Control Bit

This read/write bit configures the memory for mass erase operation or block erase operation when the ERASE bit is set.

1 = Mass Erase operation selected
0 = Block Erase operation selected

ERASE — Erase Control Bit

This read/write bit configures the memory for erase operation. This bit and the PGM bit should not be set to 1 at the same time.

1 = Erase operation selected
0 = Erase operation not selected

PGM — Program Control Bit

This read/write bit configures the memory for program operation. This bit and the ERASE bit should not be set to 1 at the same time.

1 = Program operation selected
0 = Program operation not selected

4.5 FLASH Block Erase Operation

Use the following procedure to erase a block of FLASH memory. A block consists of 512 consecutive bytes starting from addresses $X000, $X200, $X400, $X600, $X800, $XA00, $XC00 or $XE00. The 48-byte user interrupt vectors area also forms a block. Any block within the 16K bytes user memory area ($BA00–$F9FF) can be erased alone.

NOTE: *The 48-byte user interrupt vectors, $FFD0–$FFFF, cannot be erased by the block erase operation because of security reasons. Mass erase is required to erase this block.*

1. Set the ERASE bit and clear the MASS bit in the FLASH control register.
2. Write any data to any FLASH address within the address range of the block to be erased.
3. Wait for a time, t_{nvs} (5µs).
4. Set the HVEN bit.
5. Wait for a time t_{Erase} (10ms).
6. Clear the ERASE bit.
7. Wait for a time, t_{nvh} (5µs).
8. Clear the HVEN bit.
9. After time, t_{rcv} (1µs), the memory can be accessed in read mode again.

NOTE: *Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps.*

4.6 FLASH Mass Erase Operation

Use the following procedure to erase the entire FLASH memory:

1. Set both the ERASE bit and the MASS bit in the FLASH control register.
2. Write any data to any FLASH address within the address range $FFD0–$FFFF.
3. Wait for a time, t_{nvs} (5μs).
4. Set the HVEN bit.
5. Wait for a time t_{MErase} (200ms).
6. Clear the ERASE bit.
7. Wait for a time, t_{nvhl} (100μs).
8. Clear the HVEN bit.
9. After time, t_{rcv} (1μs), the memory can be accessed in read mode again.

NOTE: *Programming and erasing of FLASH locations cannot be performed by executing code from the FLASH memory; the code must be executed from RAM. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps.*

4.7 FLASH Program Operation

Programming of the FLASH memory is done on a row basis. A row consists of 64 consecutive bytes starting from addresses $XX00, $XX40, $XX80 or $XXC0. The procedure for programming a row of the FLASH memory is outlined below:

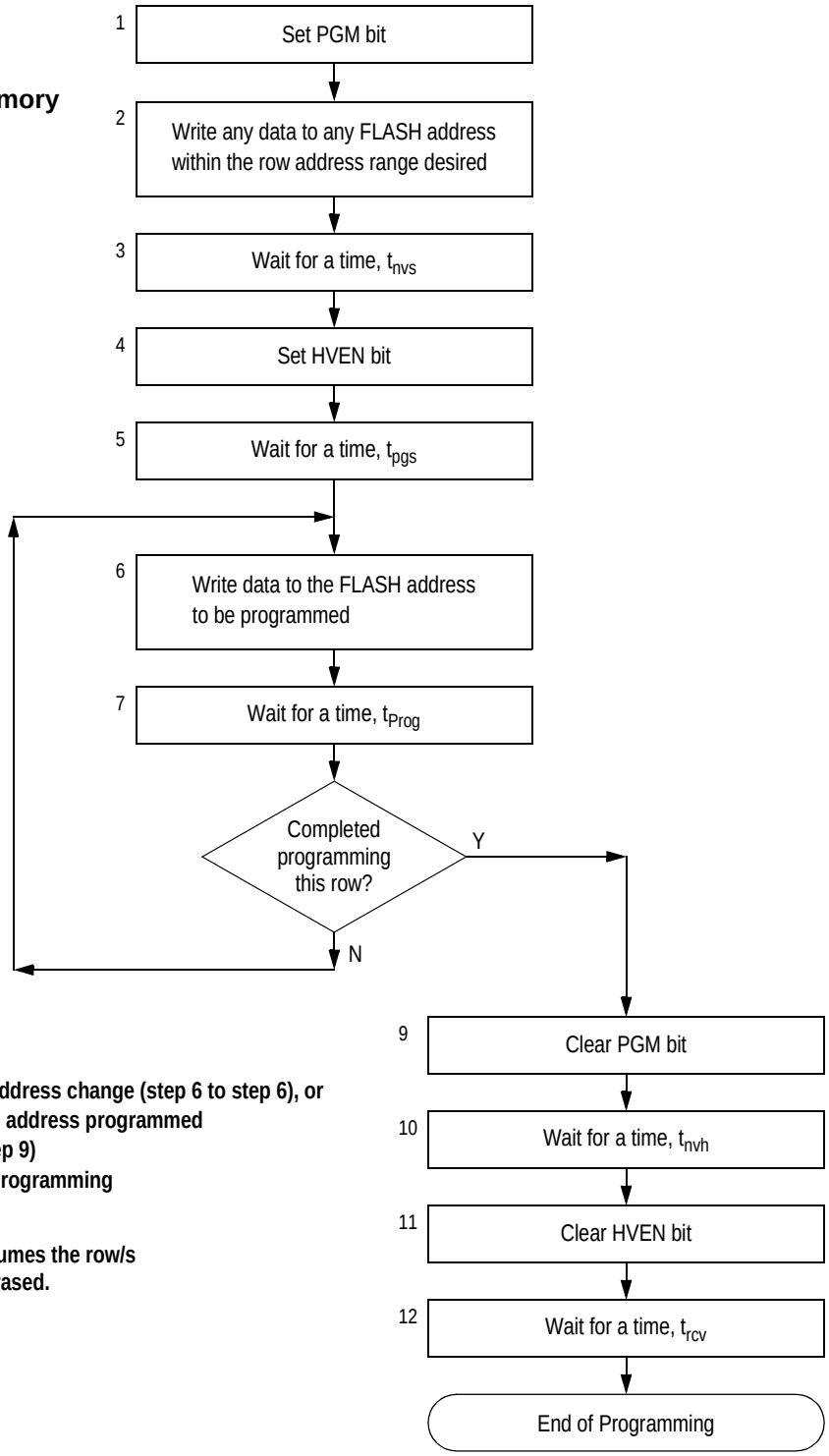
1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
2. Write any data to any FLASH address within the address range of the row to be programmed.
3. Wait for a time, t_{nvs} (5µs).
4. Set the HVEN bit.
5. Wait for a time, t_{pgs} (10µs).
6. Write data to the byte being programmed.
7. Wait for time, t_{Prog} (30µs).
8. Repeat steps 6 and 7 until all the bytes within the row are programmed.
9. Clear the PGM bit.
10. Wait for time, t_{nvh} (5µs).
11. Clear the HVEN bit.
12. After time, t_{rcv} (1µs), the memory can be accessed in read mode again.

This program sequence is repeated throughout the memory until all data is programmed.

NOTE: *Programming and erasing of FLASH locations cannot be performed by executing code from the FLASH memory; the code must be executed from RAM. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps. Do not exceed t_{Prog} maximum. See **20.14 FLASH Memory Characteristics**.*

Figure 4-3 shows a flowchart representation for programming the FLASH memory.

Algorithm for programming a row (64 bytes) of FLASH memory



NOTE:

The time between each FLASH address change (step 6 to step 6), or the time between the last FLASH address programmed to clearing PGM bit (step 6 to step 9) must not exceed the maximum programming time, t_{Prog} max.

This row program algorithm assumes the row/s to be programmed are initially erased.

Figure 4-3. FLASH Programming Flowchart

4.8 FLASH Protection

Due to the ability of the on-board charge pump to erase and program the FLASH memory in the target application, provision is made to protect blocks of memory from unintentional erase or program operations due to system malfunction. This protection is done by use of a FLASH Block Protect Register (FLBPR). The FLBPR determines the range of the FLASH memory which is to be protected. The range of the protected area starts from a location defined by FLBPR and ends to the bottom of the FLASH memory ($FFFF). When the memory is protected, the HVEN bit cannot be set in either ERASE or PROGRAM operations.

NOTE: *When the FLBPR is cleared (all 0's), the entire FLASH memory is protected from being programmed and erased. When all the bits are set, the entire FLASH memory is accessible for program and erase.*

4.8.1 FLASH Block Protect Register

The FLASH block protect register is implemented as an 8-bit I/O register. The 7 bits of the 8-bit content of this register determine the starting location of the protected range within the FLASH memory.

Address: $FE09

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
Reset:	0	0	0	0	0	0	0	0

Figure 4-4. FLASH Block Protect Register (FLBPR)

BPR[7:0] — FLASH Block Protect Register Bit 7 to Bit 0

BPR[7:1] represent bits [15:9] of a 16-bit memory address; bits [8:0] are logic 0's.

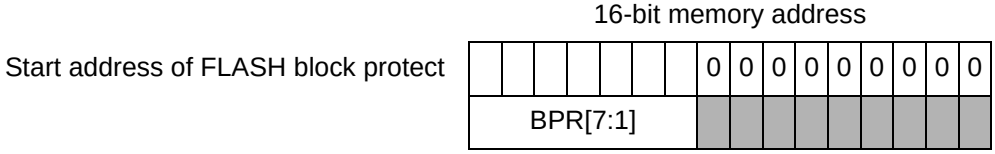


Figure 4-5. FLASH Block Protect Start Address

BPR0 is used only for BPR[7:0] = $FF, for no block protection.

The resultant 16-bit address is used for specifying the start address of the FLASH memory for block protection. The FLASH is protected from this start address to the end of FLASH memory, at $FFFF. With this mechanism, the protect start address can be X000, X200, X400, X600, X800, XA00, XC00, or XE00 within the FLASH memory.

Examples of protect start address:

BPR[7:0]	Start of Address of Protect Range
$00 to $BA	The entire FLASH memory is protected.
$BC (**1011 1100**)	$BC00 (**1011 1100** 0000 0000)
$BE (**1011 1110**)	$BE00 (**1011 1110** 0000 0000)
$C0 (**1100 0000**)	$C000 (**1100 0000** 0000 0000)
$C2 (**1100 0010**)	$C200 (**1100 0010** 0000 0000)
and so on...	
$FE	$FFD0–$FFFF (User vectors)
$FF	The entire FLASH memory is not protected.

Note:
The end address of the protected range is always $FFFF.

4.9 ROM-Resident Routines

ROM-resident routines can be called by a program running in user mode or in monitor mode (see **Section 9. Monitor ROM (MON)**) for FLASH programming, erasing, and verifying. The range of the FLASH memory must be unprotected (see **4.8 FLASH Protection**) before calling the erase or programming routine.

Table 4-1. ROM-Resident Routines

Routine Name	Call Address	Description
VERIFY	$FC03	FLASH verify routine
ERASE	$FC06	FLASH mass or block erase routine
PROGRAM	$FC09	FLASH program routine

4.9.1 Variables

The ROM-resident routines use three variables: CTRLBYT, CPUSPD and LADDR; and one data buffer. The minimum size of the data buffer is one byte and the maximum size is 64 bytes.

CPUSPD must be set before calling the erase or programming routines, and should be set to four times the value of the CPU internal bus speed in MHz. For example: for CPU speed of 6MHz, CPUSPD should be set to 24.

Table 4-2. Summary of FLASH Routine Variables

Variable	Address	Description
CTRLBYT	$0088	Control byte for setting mass or block erase.
CPUSPD	$0089	Timing adjustment for different CPU speeds.
LADDR	$008A–$008B	Last FLASH address to be programmed.
DATABUF	$0100–$013F	Data buffer for programming and verifying.

4.9.2 ERASE Routine

The ERASE routine erases the entire or a block of FLASH memory. The routine does not check for a blank range before or after erase.

NOTE: *A block erase cannot be performed on the last block of FLASH memory (user vector at $FFD0)–$FFFF).*

Table 4-3. ERASE Routine

Routine	ERASE
Calling Address	$FC06
Stack Use	5 Bytes
Input	CPUSPD — CPU speed HX — Contains any address in the range to be erased CTRLBYT — Mass or block erase Mass erase if bit 6 = 1 Block erase if bit 6 = 0

4.9.3 PROGRAM Routine

The PROGRAM routine programs a range of addresses in FLASH memory, which does not have to be on page boundaries, either at the begin or end address.

Table 4-4. PROGRAM Routine

Routine	PROGRAM
Calling Address	$FC09
Stack Use	7 Bytes
Input	CPUSPD — CPU speed HX — FLASH start address to be programmed LADDR — FLASH end address to be programmed DATABUF — Contains the data to be programmed

4.9.4 VERIFY Routine

The VERIFY routine reads and verifies a range of FLASH memory.

Table 4-5. VERIFY Routine

Routine	VERIFY
Calling Address	$FC03
Stack Use	6 Bytes
Input	HX — FLASH start address to be verified LADDR — FLASH end address to be verified DATABUF — Contains the data to be verified
Output	C Bit — C bit is set if verify passes DATABUF — Contains the data in the range of the FLASH memory

Section 5. Configuration Register (CONFIG)

5.1 Contents

5.2 Introduction

This section describes the configuration register, CONFIG. The configuration register enables or disables these options:

- Low voltage inhibit (LVI) module control and voltage trip point selection
- USB reset
- Stop mode recovery time (2048 or 4096 OSCDCLK cycles)
- COP timeout period ($2^{18} – 2^4$ or $2^{13} – 2^4$ OSCDCLK cycles)
- STOP instruction
- Computer operating properly module (COP)

5.3 Functional Description

The configuration register is used in the initialization of various options. The configuration register can be written once after each reset. All of the configuration register bits are cleared during reset. Since the various options affect the operation of the MCU, it is recommended that this register be written immediately after reset. The configuration register is located at $001F. The configuration register may be read at anytime.

5.4 Configuration Register

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LVIDR	LVI5OR3	URSTD	LVID	SSREC	COPRS	STOP	COPD
Reset:	0*	0*	0*	0*	0	0	0	0

* LVIDR, LVI5OR3, URSTD, and LVID bits are reset by POR (power-on reset) or LVI reset only.

Figure 5-1. Configuration Register (CONFIG)

LVIDR — LVI Disable Bit for V_{REG}

LVIDR disables the LVI circuit for V_{REG}. (See **Section 18. Low-Voltage Inhibit (LVI)**.)

1 = LVI circuit for V_{REG} disabled

0 = LVI circuit for V_{REG} enabled

NOTE: *There is no LVI circuit for V_{REGA}.*

LVI5OR3 — LVI Trip Point Voltage Select Bit for V_{DD}

LVI5OR3 selects the trip point voltage of the LVI circuit for V_{DD}. (See **Section 18. Low-Voltage Inhibit (LVI)**.)

1 = LVI trips at 3.3V

0 = LVI trips at 2.4V

URSTD — USB Reset Disable Bit

URSTD disables the USB reset signal generating an internal reset to the CPU and internal registers. Instead, it will generate an interrupt request to the CPU. (See **Section 11. Universal Serial Bus Module (USB)**.)

1 = USB reset generates a USB interrupt request to CPU

0 = USB reset generates a chip reset

LVID — LVI Disable Bit for V_{DD}

LVID disables the LVI circuit for V_{DD}. (See **Section 18. Low-Voltage Inhibit (LVI)**.)

1 = LVI circuit for V_{DD} disabled

0 = LVI circuit for V_{DD} enabled

SSREC — Short Stop Recovery Bit

SSREC enables the CPU to exit stop mode with a delay of 2048 OSCDCLK cycles instead of a 4096 OSCDCLK cycle delay.

1 = Stop mode recovery after 2048 OSCDCLK cycles
0 = Stop mode recovery after 4096 OSCDCLK cycles

NOTE: *Exiting stop mode by pulling reset will result in the long stop recovery.*

If using an external crystal oscillator, do not set the SSREC bit.

COPRS — COP Rate Select Bit

COPRS selects the COP timeout period. Reset clears COPRS. (See **Section 17. Computer Operating Properly (COP)**.)

1 = COP timeout period is $2^{13} – 2^4$ OSCDCLK cycles
0 = COP timeout period is $2^{18} – 2^4$ OSCDCLK cycles

STOP — STOP Instruction Enable Bit

STOP enables the STOP instruction.

1 = STOP instruction enabled
0 = STOP instruction treated as illegal opcode

COPD — COP Disable Bit

COPD disables the COP module. (See **Section 17. Computer Operating Properly (COP)**.)

1 = COP module disabled
0 = COP module enabled

Configuration Register (CONFIG)

Section 6. Central Processor Unit (CPU)

6.1 Contents

6.2 Introduction

The M68HC08 CPU (central processor unit) is an enhanced and fully object-code-compatible version of the M68HC05 CPU. The *CPU08 Reference Manual* (Motorola document order number CPU08RM/AD) contains a description of the CPU instruction set, addressing modes, and architecture.

6.3 Features

Feature of the CPU include:

- Object code fully upward-compatible with M68HC05 Family
- 16-bit stack pointer with stack manipulation instructions
- 16-Bit index register with X-register manipulation instructions
- 6-MHz CPU internal bus frequency
- 64-Kbyte program/data memory space
- 16 addressing modes
- Memory-to-memory data moves without using accumulator
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- Enhanced binary-coded decimal (BCD) data handling
- Modular architecture with expandable internal bus definition for extension of addressing range beyond 64-Kbytes
- Low-power stop and wait modes

6.4 CPU Registers

Figure 6-1 shows the five CPU registers. CPU registers are not part of the memory map.

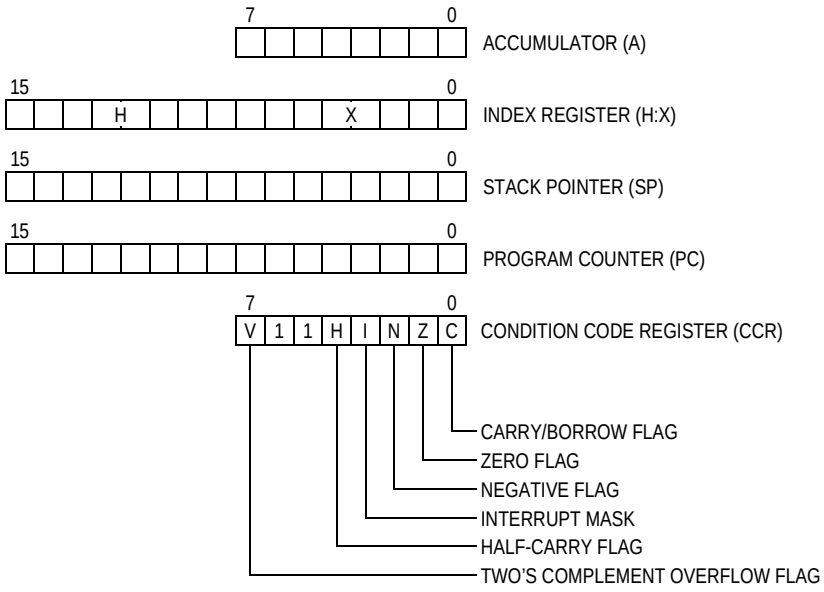


Figure 6-1. CPU Registers

6.4.1 Accumulator

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and the results of arithmetic/logic operations.

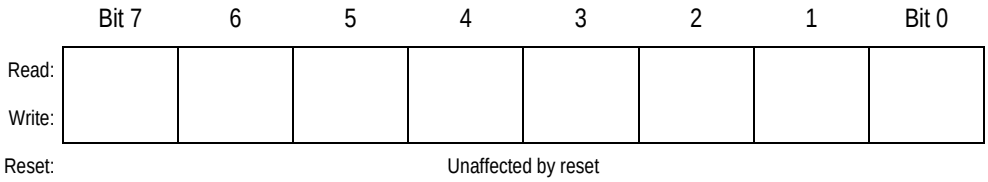


Figure 6-2. Accumulator (A)

6.4.2 Index Register

The 16-bit index register allows indexed addressing of a 64K-byte memory space. H is the upper byte of the index register, and X is the lower byte. H:X is the concatenated 16-bit index register.

In the indexed addressing modes, the CPU uses the contents of the index register to determine the conditional address of the operand.

The index register can serve also as a temporary data storage location.

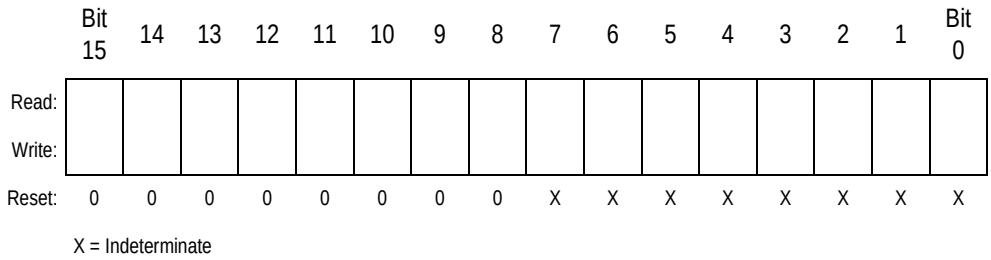


Figure 6-3. Index Register (H:X)

6.4.3 Stack Pointer

The stack pointer is a 16-bit register that contains the address of the next location on the stack. During a reset, the stack pointer is preset to $00FF. The reset stack pointer (RSP) instruction sets the least significant byte to $FF and does not affect the most significant byte. The stack pointer decrements as data is pushed onto the stack and increments as data is pulled from the stack.

In the stack pointer 8-bit offset and 16-bit offset addressing modes, the stack pointer can function as an index register to access data on the stack. The CPU uses the contents of the stack pointer to determine the conditional address of the operand.

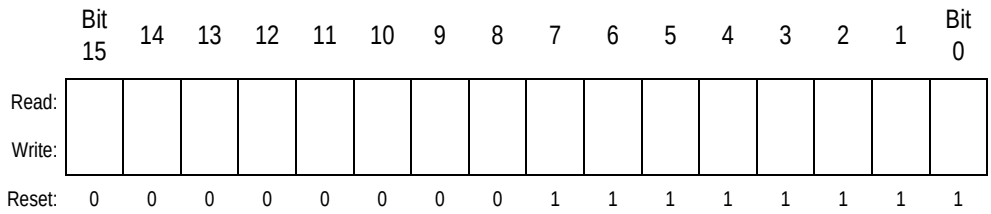


Figure 6-4. Stack Pointer (SP)

NOTE: *The location of the stack is arbitrary and may be relocated anywhere in RAM. Moving the SP out of page 0 ($0000 to $00FF) frees direct address (page 0) space. For correct operation, the stack pointer must point only to RAM locations.*

6.4.4 Program Counter

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched.

Normally, the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

During reset, the program counter is loaded with the reset vector address located at $FFFE and $FFFF. The vector address is the address of the first instruction to be executed after exiting the reset state.

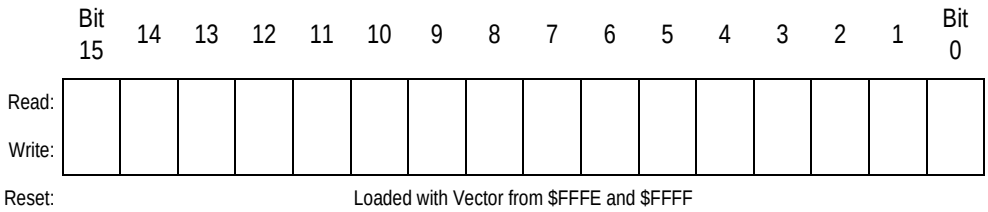


Figure 6-5. Program Counter (PC)

6.4.5 Condition Code Register

The 8-bit condition code register contains the interrupt mask and five flags that indicate the results of the instruction just executed. Bits 6 and 5 are set permanently to logic 1. The following paragraphs describe the functions of the condition code register.

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	V	1	1	H	I	N	Z	C
Reset:	X	1	1	X	1	X	X	X

X = Indeterminate

Figure 6-6. Condition Code Register (CCR)

V — Overflow Flag

The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag.

1 = Overflow
0 = No overflow

H — Half-Carry Flag

The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C flags to determine the appropriate correction factor.

1 = Carry between bits 3 and 4
0 = No carry between bits 3 and 4

I — Interrupt Mask

When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the interrupt vector is fetched.

1 = Interrupts disabled
0 = Interrupts enabled

NOTE: *To maintain M6805 Family compatibility, the upper byte of the index register (H) is not stacked automatically. If the interrupt service routine modifies H, then the user must stack and unstack H using the PSHH and PULH instructions.*

After the I bit is cleared, the highest-priority interrupt request is serviced first.

A return-from-interrupt (RTI) instruction pulls the CPU registers from the stack and restores the interrupt mask from the stack. After any reset, the interrupt mask is set and can be cleared only by the clear interrupt mask software instruction (CLI).

N — Negative Flag

The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result.

1 = Negative result
0 = Non-negative result

Z — Zero Flag

The CPU sets the zero flag when an arithmetic operation, logic operation, or data manipulation produces a result of $00.

1 = Zero result
0 = Non-zero result

C — Carry/Borrow Flag

The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some instructions — such as bit test and branch, shift, and rotate — also clear or set the carry/borrow flag.

1 = Carry out of bit 7
0 = No carry out of bit 7

6.5 Arithmetic/Logic Unit (ALU)

The ALU performs the arithmetic and logic operations defined by the instruction set.

Refer to the *CPU08 Reference Manual* (Motorola document order number CPU08RM/AD) for a description of the instructions and addressing modes and more detail about the architecture of the CPU.

6.6 Low-Power Modes

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

6.6.1 Wait Mode

The WAIT instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling interrupts. After exit from wait mode by interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock.

6.6.2 Stop Mode

The STOP instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling external interrupts. After exit from stop mode by external interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock.

After exiting stop mode, the CPU clock begins running after the oscillator stabilization delay.

6.7 CPU During Break Interrupts

If the break module is enabled, a break interrupt causes the CPU to execute the software interrupt instruction (SWI) at the completion of the current CPU instruction. (See **Section 19. Break Module (BRK)**.) The program counter vectors to $FFFC–$FFFD ($FEFC–$FEFD in monitor mode).

A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation if the break interrupt has been deasserted.

6.8 Instruction Set Summary

Table 6-1 provides a summary of the M68HC08 instruction set.

6.9 Opcode Map

The opcode map is provided in **Table 6-2**.

Table 6-1. Instruction Set Summary (Sheet 1 of 8)

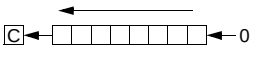
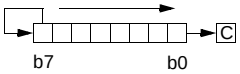
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X ADC *opr*,SP ADC *opr*,SP	Add with Carry	A ← (A) + (M) + (C)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A9 B9 C9 D9 E9 F9 9EE9 9ED9	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X ADD *opr*,SP ADD *opr*,SP	Add without Carry	A ← (A) + (M)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	AB BB CB DB EB FB 9EEB 9EDB	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
AIS #*opr*	Add Immediate Value (Signed) to SP	SP ← (SP) + (16 « M)	–	–	–	–	–	–	IMM	A7	ii	2
AIX #*opr*	Add Immediate Value (Signed) to H:X	H:X ← (H:X) + (16 « M)	–	–	–	–	–	–	IMM	AF	ii	2
AND #*opr* AND *opr* AND *opr* AND *opr*,X AND *opr*,X AND ,X AND *opr*,SP AND *opr*,SP	Logical AND	A ← (A) & (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A4 B4 C4 D4 E4 F4 9EE4 9ED4	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X ASL *opr*,SP	Arithmetic Shift Left (Same as LSL)		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR *opr*,X ASR *opr*,SP	Arithmetic Shift Right		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	37 47 57 67 77 9E67	dd ff ff	4 1 1 4 3 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + rel ? (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BCLR *n*, *opr*	Clear Bit n in M	Mn ← 0	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	4 4 4 4 4 4 4 4

Table 6-1. Instruction Set Summary (Sheet 2 of 8)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? (C) = 1	–	–	–	–	–	–	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? (Z) = 1	–	–	–	–	–	–	REL	27	rr	3
BGE *opr*	Branch if Greater Than or Equal To (Signed Operands)	PC ← (PC) + 2 + *rel* ? (N ⊕ V) = 0	–	–	–	–	–	–	REL	90	rr	3
BGT *opr*	Branch if Greater Than (Signed Operands)	PC ← (PC) + 2 + *rel* ? (Z) | (N ⊕ V) = 0	–	–	–	–	–	–	REL	92	rr	3
BHCC *rel*	Branch if Half Carry Bit Clear	PC ← (PC) + 2 + *rel* ? (H) = 0	–	–	–	–	–	–	REL	28	rr	3
BHCS *rel*	Branch if Half Carry Bit Set	PC ← (PC) + 2 + *rel* ? (H) = 1	–	–	–	–	–	–	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? (C) | (Z) = 0	–	–	–	–	–	–	REL	22	rr	3
BHS *rel*	Branch if Higher or Same (Same as BCC)	PC ← (PC) + 2 + *rel* ? (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BIH *rel*	Branch if $\overline{IRQ}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{IRQ}$ = 1	–	–	–	–	–	–	REL	2F	rr	3
BIL *rel*	Branch if $\overline{IRQ}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{IRQ}$ = 0	–	–	–	–	–	–	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X BIT *opr*,SP BIT *opr*,SP	Bit Test	(A) & (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A5 B5 C5 D5 E5 F5 9EE5 9ED5	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
BLE *opr*	Branch if Less Than or Equal To (Signed Operands)	PC ← (PC) + 2 + *rel* ? (Z) | (N ⊕ V) = 1	–	–	–	–	–	–	REL	93	rr	3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? (C) = 1	–	–	–	–	–	–	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? (C) | (Z) = 1	–	–	–	–	–	–	REL	23	rr	3
BLT *opr*	Branch if Less Than (Signed Operands)	PC ← (PC) + 2 + *rel* ? (N ⊕ V) =1	–	–	–	–	–	–	REL	91	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? (I) = 0	–	–	–	–	–	–	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? (N) = 1	–	–	–	–	–	–	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? (I) = 1	–	–	–	–	–	–	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? (Z) = 0	–	–	–	–	–	–	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? (N) = 0	–	–	–	–	–	–	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel*	–	–	–	–	–	–	REL	20	rr	3

Table 6-1. Instruction Set Summary (Sheet 3 of 8)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
BRCLR *n*,*opr*,*rel*	Branch if Bit *n* in M Clear	PC ← (PC) + 3 + *rel* ? (Mn) = 0	–	–	–	–	–	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2	–	–	–	–	–	–	REL	21	rr	3
BRSET *n*,*opr*,*rel*	Branch if Bit *n* in M Set	PC ← (PC) + 3 + *rel* ? (Mn) = 1	–	–	–	–	–	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BSET *n*,*opr*	Set Bit *n* in M	Mn ← 1	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	4 4 4 4 4 4 4 4
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	–	–	–	–	–	–	REL	AD	rr	4
CBEQ *opr*,*rel* CBEQA #*opr*,*rel* CBEQX #*opr*,*rel* CBEQ *opr*,X+,*rel* CBEQ X+,*rel* CBEQ *opr*,SP,*rel*	Compare and Branch if Equal	PC ← (PC) + 3 + rel ? (A) – (M) = $00 PC ← (PC) + 3 + rel ? (A) – (M) = $00 PC ← (PC) + 3 + rel ? (X) – (M) = $00 PC ← (PC) + 3 + rel ? (A) – (M) = $00 PC ← (PC) + 2 + rel ? (A) – (M) = $00 PC ← (PC) + 4 + rel ? (A) – (M) = $00	–	–	–	–	–	–	DIR IMM IMM IX1+ IX+ SP1	31 41 51 61 71 9E61	dd rr ii rr ii rr ff rr rr ff rr	5 4 4 5 4 6
CLC	Clear Carry Bit	C ← 0	–	–	–	–	–	0	INH	98		1
CLI	Clear Interrupt Mask	I ← 0	–	–	0	–	–	–	INH	9A		2
CLR *opr* CLRA CLRX CLRH CLR *opr*,X CLR ,X CLR *opr*,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

Table 6-1. Instruction Set Summary (Sheet 4 of 8)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X CMP *opr*,SP CMP *opr*,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 ee ff	2 3 4 4 3 2 4 5
COM *opr* COMA COMX COM *opr*,X COM ,X COM *opr*,SP	Complement (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (M) X ← ($\overline{X}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	0	–	–	↕	↕	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd ff ff	4 1 1 4 3 5
CPHX #*opr* CPHX *opr*	Compare H:X with M	(H:X) – (M:M + 1)	↕	–	–	↕	↕	↕	IMM DIR	65 75	ii ii+1 dd	3 4
CPX #*opr* CPX *opr* CPX *opr* CPX ,X CPX *opr*,X CPX *opr*,X CPX *opr*,SP CPX *opr*,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 ee ff	2 3 4 4 3 2 4 5
DAA	Decimal Adjust A	$(A)_{10}$	U	–	–	↕	↕	↕	INH	72		2
DBNZ *opr,rel* DBNZA *rel* DBNZX *rel* DBNZ *opr*,X,*rel* DBNZ X,*rel* DBNZ *opr*,SP,*rel*	Decrement and Branch if Not Zero	A ← (A) – 1 or M ← (M) – 1 or X ← (X) – 1 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 4 + *rel* ? (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 *opr* DECA DECX DEC *opr*,X DEC ,X DEC *opr*,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 #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X EOR *opr*,SP EOR *opr*,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 ee ff	2 3 4 4 3 2 4 5

Table 6-1. Instruction Set Summary (Sheet 5 of 8)

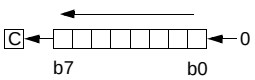
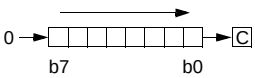
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
INC *opr* INCA INCX INC *opr*,X INC ,X INC *opr*,SP	Increment	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1 M ← (M) + 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff	4 1 1 4 3 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff 	2 3 4 3 2
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + *n* (*n* = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Unconditional Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff 	4 5 6 5 4
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X LDA *opr*,SP LDA *opr*,SP	Load A from M	A ← (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A6 B6 C6 D6 E6 F6 9EE6 9ED6	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LDHX #*opr* LDHX *opr*	Load H:X from M	H:X ← (M:M + 1)	0	–	–	↕	↕	–	IMM DIR	45 55	ii jj dd	3 4
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X LDX *opr*,SP LDX *opr*,SP	Load X from M	X ← (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	AE BE CE DE EE FE 9EEE 9EDE	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LSL *opr* LSLA LSLX LSL *opr*,X (Same as ASL) LSL ,X LSL *opr*,SP	Logical Shift Left		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
LSR *opr* LSRA LSR*X* LSR *opr*,X LSR ,X LSR *opr*,SP	Logical Shift Right		↕	–	–	0	↕	↕	DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff	4 1 1 4 3 5

Table 6-1. Instruction Set Summary (Sheet 6 of 8)

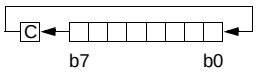
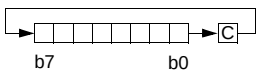
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
MOV *opr,opr* MOV *opr,*X+ MOV *#opr,opr* MOV X+*,opr*	Move	$(M)_{Destination} \leftarrow (M)_{Source}$ H:X ← (H:X) + 1 (IX+D, DIX+)	0	–	–	↕	↕	–	DD DIX+ IMD IX+D	4E 5E 6E 7E	dd dd dd ii dd dd	5 4 4 4
MUL	Unsigned multiply	X:A ← (X) × (A)	–	0	–	–	–	0	INH	42		5
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X NEG *opr*,SP	Negate (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	30 40 50 60 70 9E60	dd ff ff	4 1 1 4 3 5
NOP	No Operation	None	–	–	–	–	–	–	INH	9D		1
NSA	Nibble Swap A	A ← (A[3:0]:A[7:4])	–	–	–	–	–	–	INH	62		3
ORA *#opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X ORA *opr*,SP ORA *opr*,SP	Inclusive OR A and M	A ← (A) | (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	AA BA CA DA EA FA 9EEA 9EDA	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
PSHA	Push A onto Stack	Push (A); SP ← (SP) – 1	–	–	–	–	–	–	INH	87		2
PSHH	Push H onto Stack	Push (H); SP ← (SP) – 1	–	–	–	–	–	–	INH	8B		2
PSHX	Push X onto Stack	Push (X); SP ← (SP) – 1	–	–	–	–	–	–	INH	89		2
PULA	Pull A from Stack	SP ← (SP + 1); Pull (A)	–	–	–	–	–	–	INH	86		2
PULH	Pull H from Stack	SP ← (SP + 1); Pull (H)	–	–	–	–	–	–	INH	8A		2
PULX	Pull X from Stack	SP ← (SP + 1); Pull (X)	–	–	–	–	–	–	INH	88		2
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X ROL *opr*,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 *opr* RORA RORX ROR *opr*,X ROR ,X ROR *opr*,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 ← $FF	–	–	–	–	–	–	INH	9C		1

Table 6-1. Instruction Set Summary (Sheet 7 of 8)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	↕	INH	80		7
RTS	Return from Subroutine	SP ← SP + 1; Pull (PCH) SP ← SP + 1; Pull (PCL)	–	–	–	–	–	–	INH	81		4
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X SBC *opr*,SP SBC *opr*,SP	Subtract with Carry	A ← (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 ee ff	2 3 4 4 3 2 4 5
SEC	Set Carry Bit	C ← 1	–	–	–	–	–	1	INH	99		1
SEI	Set Interrupt Mask	I ← 1	–	–	1	–	–	–	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X STA *opr*,SP STA *opr*,SP	Store A in M	M ← (A)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh ll ee ff ff ff ee ff	3 4 4 3 2 4 5
STHX *opr*	Store H:X in M	(M:M + 1) ← (H:X)	0	–	–	↕	↕	–	DIR	35	dd	4
STOP	Enable $\overline{\text{IRQ}}$ Pin; Stop Oscillator	I ← 0; Stop Oscillator	–	–	0	–	–	–	INH	8E		1
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X STX *opr*,SP STX *opr*,SP	Store X in M	M ← (X)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP1 SP2	BF CF DF EF FF 9EEF 9EDF	dd hh ll ee ff ff ff ee ff	3 4 4 3 2 4 5
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X SUB *opr*,SP SUB *opr*,SP	Subtract	A ← (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 ee ff	2 3 4 4 3 2 4 5

Table 6-1. Instruction Set Summary (Sheet 8 of 8)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	–	–	1	–	–	–	INH	83		9
TAP	Transfer A to CCR	CCR ← (A)	↕	↕	↕	↕	↕	↕	INH	84		2
TAX	Transfer A to X	X ← (A)	–	–	–	–	–	–	INH	97		1
TPA	Transfer CCR to A	A ← (CCR)	–	–	–	–	–	–	INH	85		1
TST *opr* TSTA TSTX TST *opr*,X TST ,X TST *opr*,SP	Test for Negative or Zero	(A) – $00 or (X) – $00 or (M) – $00	0	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3D 4D 5D 6D 7D 9E6D	dd ff ff	3 1 1 3 2 4
TSX	Transfer SP to H:X	H:X ← (SP) + 1	–	–	–	–	–	–	INH	95		2
TXA	Transfer X to A	A ← (X)	–	–	–	–	–	–	INH	9F		1
TXS	Transfer H:X to SP	(SP) ← (H:X) – 1	–	–	–	–	–	–	INH	94		2

A	Accumulator
C	Carry/borrow bit
CCR	Condition code register
dd	Direct address of operand
dd rr	Direct address of operand and relative offset of branch instruction
DD	Direct to direct addressing mode
DIR	Direct addressing mode
DIX+	Direct to indexed with post increment addressing mode
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing
EXT	Extended addressing mode
ff	Offset byte in indexed, 8-bit offset addressing
H	Half-carry bit
H	Index register high byte
hh ll	High and low bytes of operand address in extended addressing
I	Interrupt mask
ii	Immediate operand byte
IMD	Immediate source to direct destination addressing mode
IMM	Immediate addressing mode
INH	Inherent addressing mode
IX	Indexed, no offset addressing mode
IX+	Indexed, no offset, post increment addressing mode
IX+D	Indexed with post increment to direct addressing mode
IX1	Indexed, 8-bit offset addressing mode
IX1+	Indexed, 8-bit offset, post increment addressing mode
IX2	Indexed, 16-bit offset addressing mode
M	Memory location
N	Negative bit
n	Any bit
opr	Operand (one or two bytes)
PC	Program counter
PCH	Program counter high byte
PCL	Program counter low byte
REL	Relative addressing mode
rel	Relative program counter offset byte
rr	Relative program counter offset byte
SP1	Stack pointer, 8-bit offset addressing mode
SP2	Stack pointer 16-bit offset addressing mode
SP	Stack pointer
U	Undefined
V	Overflow bit
X	Index register low byte
Z	Zero bit
&	Logical AND
|	Logical OR
⊕	Logical EXCLUSIVE OR
( )	Contents of
–( )	Negation (two's complement)
#	Immediate value
«	Sign extend
←	Loaded with
?	If
:	Concatenated with
↕	Set or cleared
—	Not affected

Table 6-2. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write						Control		Register/Memory							
	DIR	DIR	REL	DIR	INH	INH	IX1	SP1	IX	INH	INH	IMM	DIR	EXT	IX2	SP2	IX1	SP1	IX
MSB / LSB	0	1	2	3	4	5	6	9E6	7	8	9	A	B	C	D	9ED	E	9EE	F
0	5 BRSET0 3 DIR	4 BSET0 2 DIR	3 BRA 2 REL	4 NEG 2 DIR	1 NEGA 1 INH	1 NEGX 1 INH	4 NEG 2 IX1	5 NEG 3 SP1	3 NEG 1 IX	7 RTI 1 INH	3 BGE 2 REL	2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	4 SUB 3 IX2	5 SUB 4 SP2	3 SUB 2 IX1	4 SUB 3 SP1	2 SUB 1 IX
1	5 BRCLR0 3 DIR	4 BCLR0 2 DIR	3 BRN 2 REL	5 CBEQ 3 DIR	4 CBEQA 3 IMM	4 CBEQX 3 IMM	5 CBEQ 3 IX1+	6 CBEQ 4 SP1	4 CBEQ 2 IX+	4 RTS 1 INH	3 BLT 2 REL	2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	4 CMP 3 IX2	5 CMP 4 SP2	3 CMP 2 IX1	4 CMP 3 SP1	2 CMP 1 IX
2	5 BRSET1 3 DIR	4 BSET1 2 DIR	3 BHI 2 REL		5 MUL 1 INH	7 DIV 1 INH	3 NSA 1 INH		2 DAA 1 INH		3 BGT 2 REL	2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	4 SBC 3 IX2	5 SBC 4 SP2	3 SBC 2 IX1	4 SBC 3 SP1	2 SBC 1 IX
3	5 BRCLR1 3 DIR	4 BCLR1 2 DIR	3 BLS 2 REL	4 COM 2 DIR	1 COMA 1 INH	1 COMX 1 INH	4 COM 2 IX1	5 COM 3 SP1	3 COM 1 IX	9 SWI 1 INH	3 BLE 2 REL	2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	4 CPX 3 IX2	5 CPX 4 SP2	3 CPX 2 IX1	4 CPX 3 SP1	2 CPX 1 IX
4	5 BRSET2 3 DIR	4 BSET2 2 DIR	3 BCC 2 REL	4 LSR 2 DIR	1 LSRA 1 INH	1 LSRX 1 INH	4 LSR 2 IX1	5 LSR 3 SP1	3 LSR 1 IX	2 TAP 1 INH	2 TXS 1 INH	2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	4 AND 3 IX2	5 AND 4 SP2	3 AND 2 IX1	4 AND 3 SP1	2 AND 1 IX
5	5 BRCLR2 3 DIR	4 BCLR2 2 DIR	3 BCS 2 REL	4 STHX 2 DIR	3 LDHX 3 IMM	4 LDHX 2 DIR	3 CPHX 3 IMM		4 CPHX 2 DIR	1 TPA 1 INH	2 TSX 1 INH	2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	4 BIT 3 IX2	5 BIT 4 SP2	3 BIT 2 IX1	4 BIT 3 SP1	2 BIT 1 IX
6	5 BRSET3 3 DIR	4 BSET3 2 DIR	3 BNE 2 REL	4 ROR 2 DIR	1 RORA 1 INH	1 RORX 1 INH	4 ROR 2 IX1	5 ROR 3 SP1	3 ROR 1 IX	2 PULA 1 INH		2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	4 LDA 3 IX2	5 LDA 4 SP2	3 LDA 2 IX1	4 LDA 3 SP1	2 LDA 1 IX
7	5 BRCLR3 3 DIR	4 BCLR3 2 DIR	3 BEQ 2 REL	4 ASR 2 DIR	1 ASRA 1 INH	1 ASRX 1 INH	4 ASR 2 IX1	5 ASR 3 SP1	3 ASR 1 IX	2 PSHA 1 INH	1 TAX 1 INH	2 AIS 2 IMM	3 STA 2 DIR	4 STA 3 EXT	4 STA 3 IX2	5 STA 4 SP2	3 STA 2 IX1	4 STA 3 SP1	2 STA 1 IX
8	5 BRSET4 3 DIR	4 BSET4 2 DIR	3 BHCC 2 REL	4 LSL 2 DIR	1 LSLA 1 INH	1 LSLX 1 INH	4 LSL 2 IX1	5 LSL 3 SP1	3 LSL 1 IX	2 PULX 1 INH	1 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	4 EOR 3 IX2	5 EOR 4 SP2	3 EOR 2 IX1	4 EOR 3 SP1	2 EOR 1 IX
9	5 BRCLR4 3 DIR	4 BCLR4 2 DIR	3 BHCS 2 REL	4 ROL 2 DIR	1 ROLA 1 INH	1 ROLX 1 INH	4 ROL 2 IX1	5 ROL 3 SP1	3 ROL 1 IX	2 PSHX 1 INH	1 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	4 ADC 3 IX2	5 ADC 4 SP2	3 ADC 2 IX1	4 ADC 3 SP1	2 ADC 1 IX
A	5 BRSET5 3 DIR	4 BSET5 2 DIR	3 BPL 2 REL	4 DEC 2 DIR	1 DECA 1 INH	1 DECX 1 INH	4 DEC 2 IX1	5 DEC 3 SP1	3 DEC 1 IX	2 PULH 1 INH	2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	4 ORA 3 IX2	5 ORA 4 SP2	3 ORA 2 IX1	4 ORA 3 SP1	2 ORA 1 IX
B	5 BRCLR5 3 DIR	4 BCLR5 2 DIR	3 BMI 2 REL	5 DBNZ 3 DIR	3 DBNZA 2 INH	3 DBNZX 2 INH	5 DBNZ 3 IX1	6 DBNZ 4 SP1	4 DBNZ 2 IX	2 PSHH 1 INH	2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	4 ADD 3 IX2	5 ADD 4 SP2	3 ADD 2 IX1	4 ADD 3 SP1	2 ADD 1 IX
C	5 BRSET6 3 DIR	4 BSET6 2 DIR	3 BMC 2 REL	4 INC 2 DIR	1 INCA 1 INH	1 INCX 1 INH	4 INC 2 IX1	5 INC 3 SP1	3 INC 1 IX	1 CLRH 1 INH	1 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2		3 JMP 2 IX1		2 JMP 1 IX
D	5 BRCLR6 3 DIR	4 BCLR6 2 DIR	3 BMS 2 REL	3 TST 2 DIR	1 TSTA 1 INH	1 TSTX 1 INH	3 TST 2 IX1	4 TST 3 SP1	2 TST 1 IX		1 NOP 1 INH	4 BSR 2 REL	4 JSR 2 DIR	5 JSR 3 EXT	6 JSR 3 IX2		5 JSR 2 IX1		4 JSR 1 IX
E	5 BRSET7 3 DIR	4 BSET7 2 DIR	3 BIL 2 REL		5 MOV 3 DD	4 MOV 2 DIX+	4 MOV 3 IMD		4 MOV 2 IX+D	1 STOP 1 INH	*	2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	4 LDX 3 IX2	5 LDX 4 SP2	3 LDX 2 IX1	4 LDX 3 SP1	2 LDX 1 IX
F	5 BRCLR7 3 DIR	4 BCLR7 2 DIR	3 BIH 2 REL	3 CLR 2 DIR	1 CLRA 1 INH	1 CLRX 1 INH	3 CLR 2 IX1	4 CLR 3 SP1	2 CLR 1 IX	1 WAIT 1 INH	1 TXA 1 INH	2 AIX 2 IMM	3 STX 2 DIR	4 STX 3 EXT	4 STX 3 IX2	5 STX 4 SP2	3 STX 2 IX1	4 STX 3 SP1	2 STX 1 IX

INH Inherent
IMM Immediate
DIR Direct
EXT Extended
DD Direct-Direct
IX+D Indexed-Direct
REL Relative
IX Indexed, No Offset
IX1 Indexed, 8-Bit Offset
IX2 Indexed, 16-Bit Offset
IMD Immediate-Direct
DIX+ Direct-Indexed
SP1 Stack Pointer, 8-Bit Offset
SP2 Stack Pointer, 16-Bit Offset
IX+ Indexed, No Offset with Post Increment
IX1+ Indexed, 1-Byte Offset with Post Increment

*Pre-byte for stack pointer indexed instructions

MSB / LSB	
0	High Byte of Opcode in Hexadecimal
Low Byte of Opcode in Hexadecimal	Cycles
0	Opcode Mnemonic
5 BRSET0 3 DIR	Number of Bytes / Addressing Mode

Section 7. Oscillator (OSC)

7.1 Contents

7.2 Introduction

The oscillator circuit is designed for use with crystals or ceramic resonators. The oscillator circuit generates the crystal clock signal, OSCXCLK, and passes through a clock doubler to produce OSCDCLK. This clock doubler clock is further divided by two before being passed on to the system integration module (SIM) for bus clock generation. **Figure 7-1** shows the structure of the oscillator. The oscillator requires various external components.

The MC68HC908JB16 operates from a nominal 12MHz crystal, providing a 6MHz internal bus clock. The 12MHz clock is required for various modules, such as the CGMs and USB. The clock doubler clock, OSCDCLK, is used as the base clock for the COP module.

7.3 Oscillator External Connections

In its typical configuration, the oscillator requires five external components. The crystal oscillator is normally connected in a Pierce oscillator configuration, as shown in **Figure 7-1**. This figure shows only the logical representation of the internal components and may not represent actual circuitry. The oscillator configuration uses five components:

- Crystal, X_1 (nominally 12 MHz)
- Fixed capacitor, C_1
- Tuning capacitor, C_2 (can also be a fixed capacitor)
- Feedback resistor, R_B
- Series resistor, R_S (not required for 12 MHz)

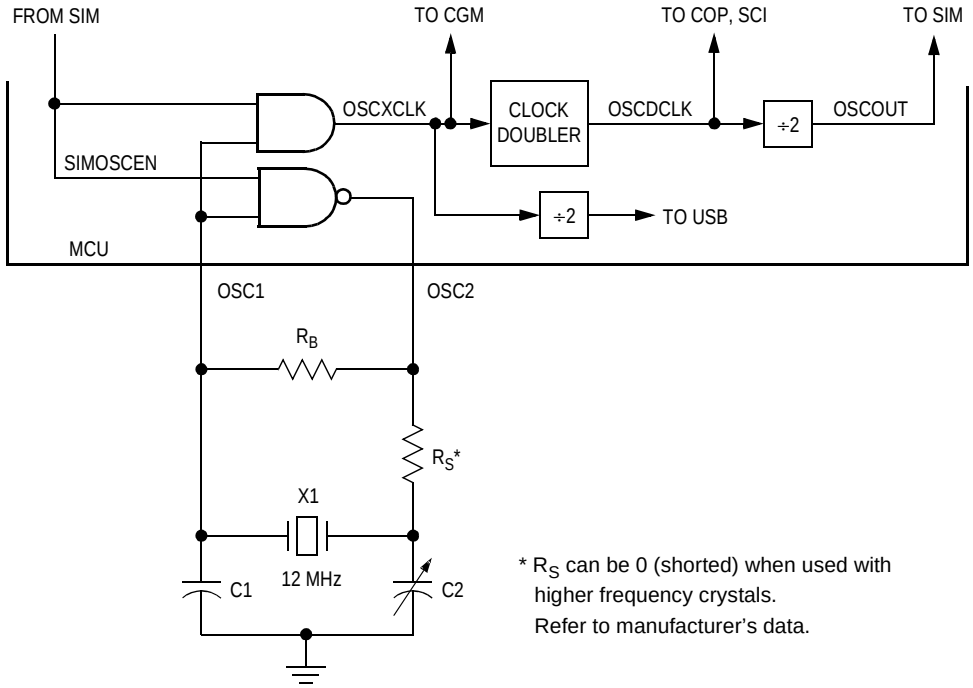


Figure 7-1. Oscillator External Connections

The series resistor (R_S) is included in the diagram to follow strict Pierce oscillator guidelines and may not be required for all ranges of operation, especially with high-frequency crystals. Refer to the crystal manufacturer's data for more information.

7.4 I/O Signals

The following paragraphs describe the oscillator input/output (I/O) signals.

7.4.1 Crystal Amplifier Input Pin (OSC1)

The OSC1 pin is an input to the crystal oscillator amplifier.

7.4.2 Crystal Amplifier Output Pin (OSC1)

The OSC2 pin is the output of the crystal oscillator inverting amplifier.

7.4.3 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal comes from the system integration module (SIM) and enables the oscillator.

7.4.4 Crystal Output Frequency Signal (OSCXCLK)

OSCXCLK is the crystal oscillator output signal. It runs at the full speed of the crystal (f_{XCLK}) and comes directly from the crystal oscillator circuit. **Figure 7-1** shows only the logical relation of OSCXCLK to OSC1 and OSC2 and may not represent the actual circuitry. The duty cycle of OSCXCLK is unknown and may depend on the crystal and other external factors. Also, the frequency and amplitude of OSCXCLK can be unstable at startup.

7.4.5 Clock Doubler Out (OSCDCLK)

OSCDCLK is the clock doubler output signal. It runs at twice the speed of the crystal (f_{XCLK}) and comes from the clock doubler circuit.

7.4.6 Oscillator Out (OSCOUT)

OSCOUT is the divide-by-two signal after the clock doubler circuit. It runs at the same speed as OSCXCLK, at crystal frequency (f_{XCLK}). This signal goes to the SIM, which generates the MCU clocks. OSCOUT will be divided-by-two again in the SIM and results in the internal bus frequency being one half of the crystal frequency.

7.5 Low-Power Modes

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

7.5.1 Wait Mode

The WAIT instruction has no effect on the oscillator logic. OSCXCLK continues to drive to the MCU.

7.5.2 Stop Mode

The STOP instruction disables the OSCXCLK output.

7.6 Oscillator During Break Mode

The oscillator continues to drive OSCXCLK when the chip enters the break state.

Section 8. System Integration Module (SIM)

8.1 Contents

8.2 Introduction

This section describes the system integration module (SIM). Together with the CPU, the SIM controls all MCU activities. The SIM is a system state controller that coordinates CPU and exception timing. A block diagram of the SIM is shown in **Figure 8-1**. **Figure 8-2** is a summary of the SIM I/O registers. The SIM is responsible for:

- Bus clock generation and control for CPU and peripherals
  - Stop/wait/reset/break entry and recovery
  - Internal clock control
- Master reset control, including power-on reset (POR) and COP timeout
- Interrupt control:
  - Acknowledge timing
  - Arbitration control timing
  - Vector address generation
- CPU enable/disable timing
- Modular architecture expandable to 128 interrupt sources

Table 8-1 shows the internal signal names used in this section.

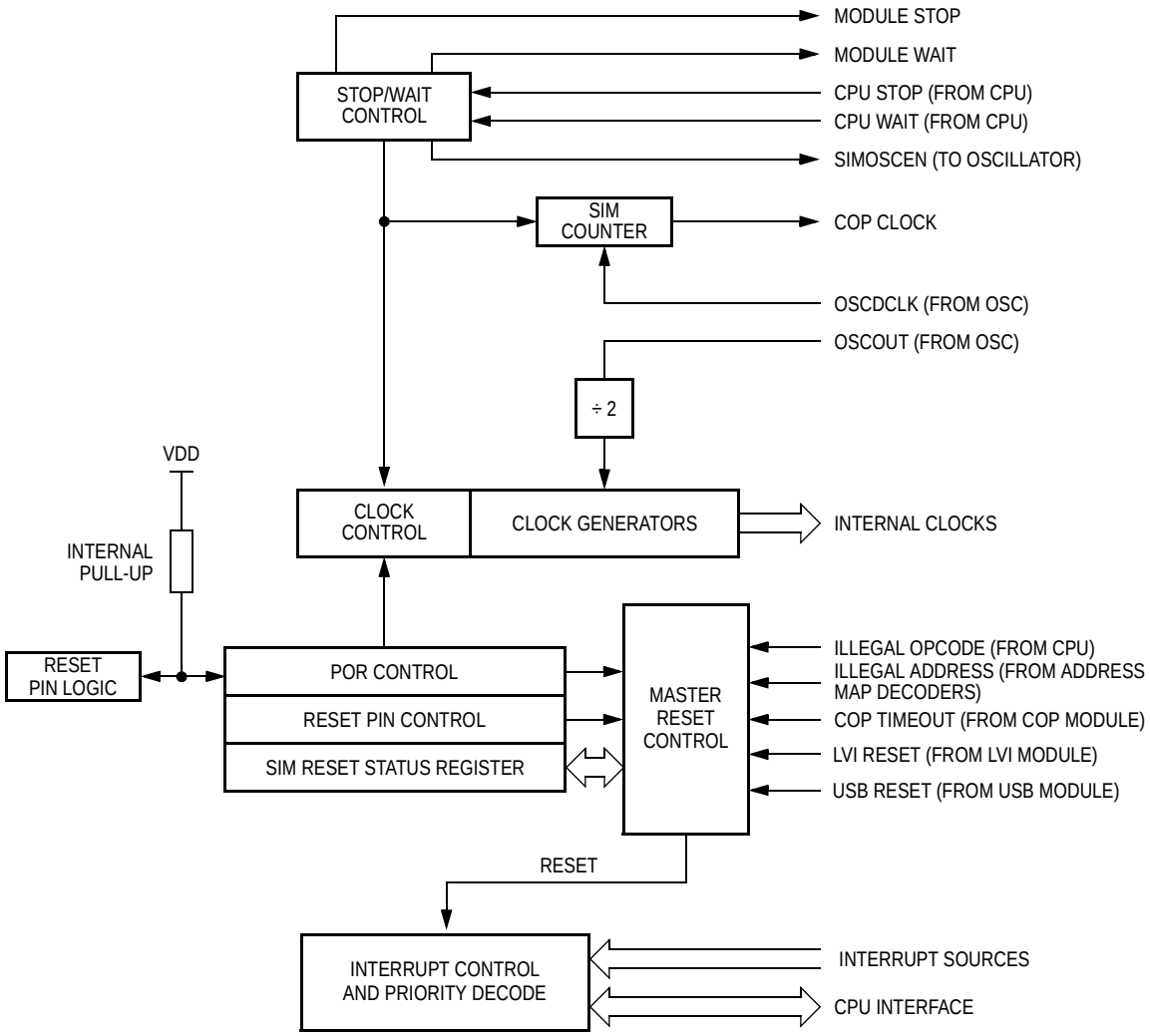


Figure 8-1. SIM Block Diagram

Table 8-1. SIM Module Signal Name Conventions

Signal Name	Description
OSCDCLK	Clock doubler output which has twice the frequency of OSC1 from the oscillator
OSCOUT	The OSCDCLK frequency divided by two. This signal is again divided by two in the SIM to generate the internal bus clocks. (Bus clock = OSCDCLK ÷ 4 = OSCXCLK ÷ 2)
IAB	Internal address bus
IDB	Internal data bus
PORRST	Signal from the power-on reset module to the SIM
IRST	Internal reset signal
R/$\overline{W}$	Read/write signal

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE00	SIM Break Status Register (SBSR)	Read:	R	R	R	R	R	R	SBSW	R
		Write:							See note	
		Reset:							0	
Note: Writing a logic 0 clears SBSW.										
$FE01	SIM Reset Status Register (SRSR)	Read:	POR	PIN	COP	ILOP	ILAD	USB	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
$FE03	SIM Break Flag Control Register (SBFCR)	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$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

Figure 8-2. SIM I/O Register Summary

8.3 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, OSCOUT, as shown in **Figure 8-3**.

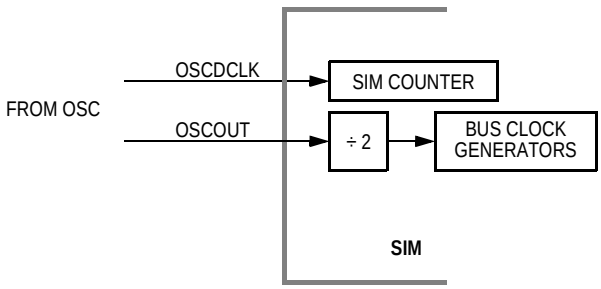


Figure 8-3. SIM Clock Signals

8.3.1 Bus Timing

In user mode, the internal bus frequency is the oscillator frequency divided by two.

8.3.2 Clock Startup from POR or LVI Reset

When the power-on reset (POR) module or the low-voltage inhibit module generates a reset, the clocks to the CPU and peripherals are inactive and held in an inactive phase until after the 4096 OSCDCLK cycle POR timeout has completed. The $\overline{\text{RST}}$ pin is driven low by the SIM during this entire period. The IBUS clocks start upon completion of the timeout.

8.3.3 Clocks in Stop Mode and Wait Mode

Upon exit from stop mode by an interrupt, break, or reset, the SIM allows OSCDCLK to clock the SIM counter. The CPU and peripheral clocks do not become active until after the stop delay timeout. This timeout is selectable as 4096 or 2048 OSCDCLK cycles. (See **8.7.2 Stop Mode**.)

In wait mode, the CPU clocks are inactive. The SIM also produces two sets of clocks for other modules. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

8.4 Reset and System Initialization

The MCU has the following reset sources:

- Power-on reset module (POR)
- External reset pin ($\overline{\text{RST}}$)
- Computer operating properly module (COP)
- Illegal opcode
- Illegal address
- Universal serial bus module (USB)
- Low-voltage inhibit module (LVI)

All of these resets produce the vector $FFFE–FFFF ($FEFE–FEFF in monitor mode) and assert the internal reset signal (IRST). IRST causes all registers to be returned to their default values and all modules to be returned to their reset states.

An internal reset clears the SIM counter (see **8.5 SIM Counter**), but an external reset does not. Each of the resets sets a corresponding bit in the SIM reset status register (SRSR). (See **8.8 SIM Registers**.)

8.4.1 External Pin Reset

The $\overline{\text{RST}}$ pin circuit includes an internal pullup device. Pulling the asynchronous $\overline{\text{RST}}$ pin low halts all processing. The PIN bit of the SIM reset status register (SRSR) is set as long as $\overline{\text{RST}}$ is held low for a minimum of 67 OSCDCLK cycles, assuming that neither the POR nor the LVI was the source of the reset. See **Table 8-2** for details. **Figure 8-4** shows the relative timing.

Table 8-2. PIN Bit Set Timing

Reset Type	Number of Cycles Required to Set PIN
POR/LVI	4163 (4096 + 64 + 3)
All others	67 (64 + 3)

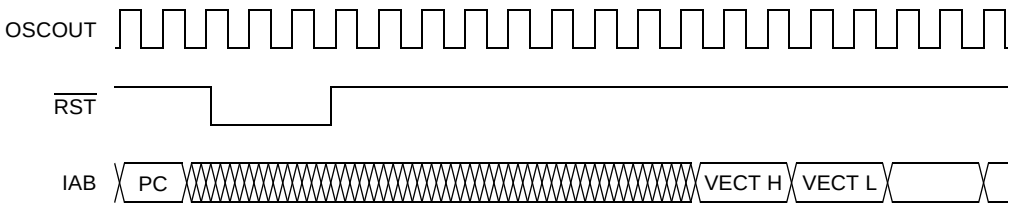


Figure 8-4. External Reset Timing

8.4.2 Active Resets from Internal Sources

All internal reset sources actively pull the $\overline{\text{RST}}$ pin low for 32 OSCDCLK cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles. (See **Figure 8-5**.) An internal reset can be caused by an illegal address, illegal opcode, COP timeout, LVI, the USB module or POR. (See **Figure 8-6 . Sources of Internal Reset**.)

NOTE: *For LVI or POR resets, the SIM cycles through 4096 OSCDCLK cycles during which the SIM forces the $\overline{\text{RST}}$ pin low. The internal reset signal then follows the sequence from the falling edge of $\overline{\text{RST}}$ shown in **Figure 8-5**.*

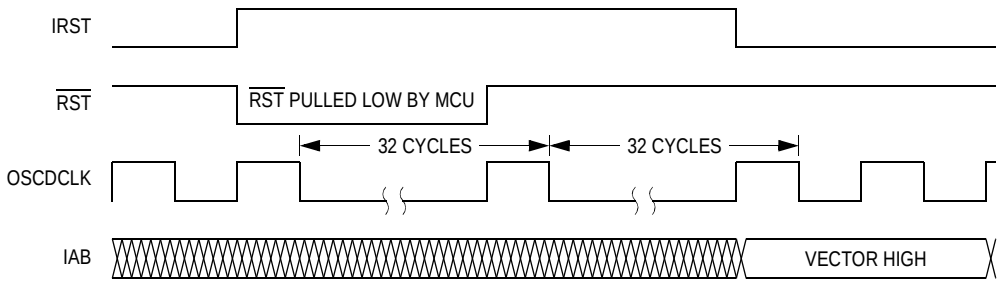


Figure 8-5. Internal Reset Timing

The COP reset is asynchronous to the bus clock.

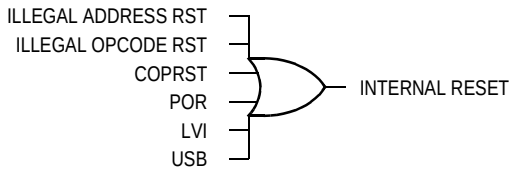


Figure 8-6. Sources of Internal Reset

The active reset feature allows the part to issue a reset to peripherals and other chips within a system built around the MCU.

8.4.2.1 Power-On Reset

When power is first applied to the MCU, the power-on reset module (POR) generates a pulse to indicate that power-on has occurred. The external reset pin ($\overline{RST}$) is held low while the SIM counter counts out 4096 OSCDCLK cycles. Sixty-four OSCDCLK cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur.

At power-on, the following events occur:

- A POR pulse is generated.
- The internal reset signal is asserted.
- The SIM enables the oscillator to drive OSCDCLK.
- Internal clocks to the CPU and modules are held inactive for 4096 OSCDCLK cycles to allow stabilization of the oscillator.
- The $\overline{RST}$ 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.

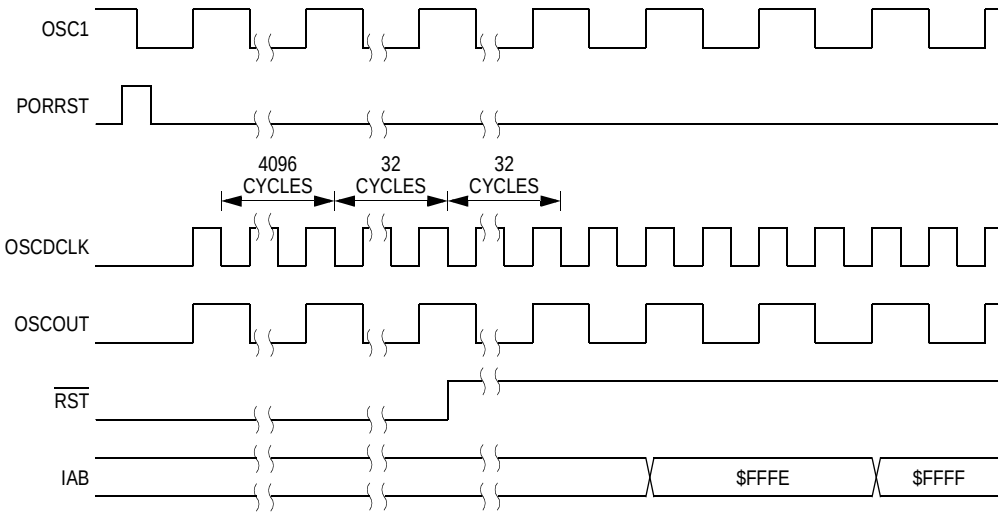


Figure 8-7. POR Recovery

8.4.2.2 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 stages 12 through 5 of the SIM counter. The SIM counter output, which occurs at least every $2^{12} – 2^4$ OSCDCLK 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 $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ 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.

8.4.2.3 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.

If the stop enable bit, STOP, in the mask option register is logic 0, the SIM treats the STOP instruction as an illegal opcode and causes an illegal opcode reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

8.4.2.4 Illegal Address Reset

An opcode fetch from an unmapped address generates an illegal address reset. The SIM verifies that the CPU is fetching an opcode prior to asserting the ILAD bit in the SIM reset status register (SRSR) and resetting the MCU. A data fetch from an unmapped address does not generate a reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

8.4.2.5 Low-Voltage Inhibit (LVI) Reset

The low-voltage inhibit module (LVI) asserts its output to the SIM when the V_{DD} or V_{REG} voltage falls to the LVI reset voltage, V_{TRIP}. The LVI bit in the SIM reset status register (SRSR) is set, and the external reset pin ($\overline{RST}$) is held low while the SIM counter counts out 4096 OSCDCLK cycles. Sixty-four OSCDCLK cycles later, the CPU is released from reset to allow the reset vector sequence to occur. The SIM actively pulls down the $\overline{RST}$ pin for all internal reset sources.

8.4.2.6 Universal Serial Bus (USB) Reset

The USB module will detect a reset signaled on the bus by the presence of an extended SE0 at the USB data pins of a device. The MCU seeing a single-ended 0 on its USB data inputs for more than 2.5µs treats that signal as a reset. After the reset is removed, the device will be in the attached, but not yet addressed or configured, state (refer to Section 9.1 USB Devices of the *Universal Serial Bus Specification Rev. 2.0*). The device must be able to accept the device address via a SET_ADDRESS command (refer to Section 9.4 of the *Universal Serial Bus Specification Rev. 2.0*) no later than 10ms after the reset is removed.

USB reset can be disabled to generate an internal reset. It can be configured to generate IRQ interrupt. (See **Section 5. Configuration Register (CONFIG)**.)

NOTE: *USB reset is disabled when the USB module is disabled by clearing the USBEN bit of the USB address register (UADDR).*

8.4.2.7 Registers Values After Different Resets

Some registers are reset by POR or LVI reset only. **Table 8-3** shows the registers or register bits which are unaffected by normal resets.

Table 8-3. Registers not Affected by Normal Reset

Bits	Registers	After Reset (except POR or LVI)	After POR or LVI
LVIDR, LVI5OR3, URSTD, LVID	CONFIG	Unaffected	0
USBEN	UADDR	Unaffected	0
PULLEN	UCR3	Unaffected	0
All	USR0, USR1	Unaffected	Indeterminate
All	UE0D0–UE0D7	Unaffected	Indeterminate
All	UE1D0–UE1D7	Unaffected	Indeterminate
All	UE2D0–UE2D7	Unaffected	Indeterminate
All	PTA, PTC, PTD, and PTE	Unaffected	Indeterminate
DDRA7	DDRA	Unaffected	0

8.5 SIM Counter

The SIM counter is used by the power-on reset module (POR) and in stop mode recovery to allow the oscillator time to stabilize before enabling the internal bus (IBUS) clocks. The SIM counter also serves as a prescaler for the computer operating properly module (COP). The SIM counter uses 12 stages for counting, followed by a 13th stage that triggers a reset of SIM counters and supplies the clock for the COP module. The SIM counter is clocked by the falling edge of OSCDCLK.

8.5.1 SIM Counter During Power-On Reset

The power-on reset module (POR) detects power applied to the MCU. At power-on, the POR circuit asserts the signal PORRST. Once the SIM is initialized, it enables the oscillator to drive the bus clock state machine.

8.5.2 SIM Counter During Stop Mode Recovery

The SIM counter also is used for stop mode recovery. The STOP instruction clears the SIM counter. After an interrupt, break, or reset, the SIM senses the state of the short stop recovery bit, SSREC, in the configuration register (CONFIG). If the SSREC bit is a logic 1, then the stop recovery is reduced from the normal delay of 4096 OSCDCLK cycles down to 2048 OSCDCLK cycles. This is ideal for applications using canned oscillators that do not require long startup times from stop mode. External crystal applications should use the full stop recovery time, that is, with SSREC cleared in the configuration register (CONFIG).

8.5.3 SIM Counter and Reset States

External reset has no effect on the SIM counter. (See **8.7.2 Stop Mode** for details.) The SIM counter is free-running after all reset states. (See **8.4.2 Active Resets from Internal Sources** for counter control and internal reset recovery sequences.)

8.6 Exception Control

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

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

8.6.1 Interrupts

An interrupt temporarily changes the sequence of program execution to respond to a particular event. **Figure 8-8** flow charts the handling of system interrupts.

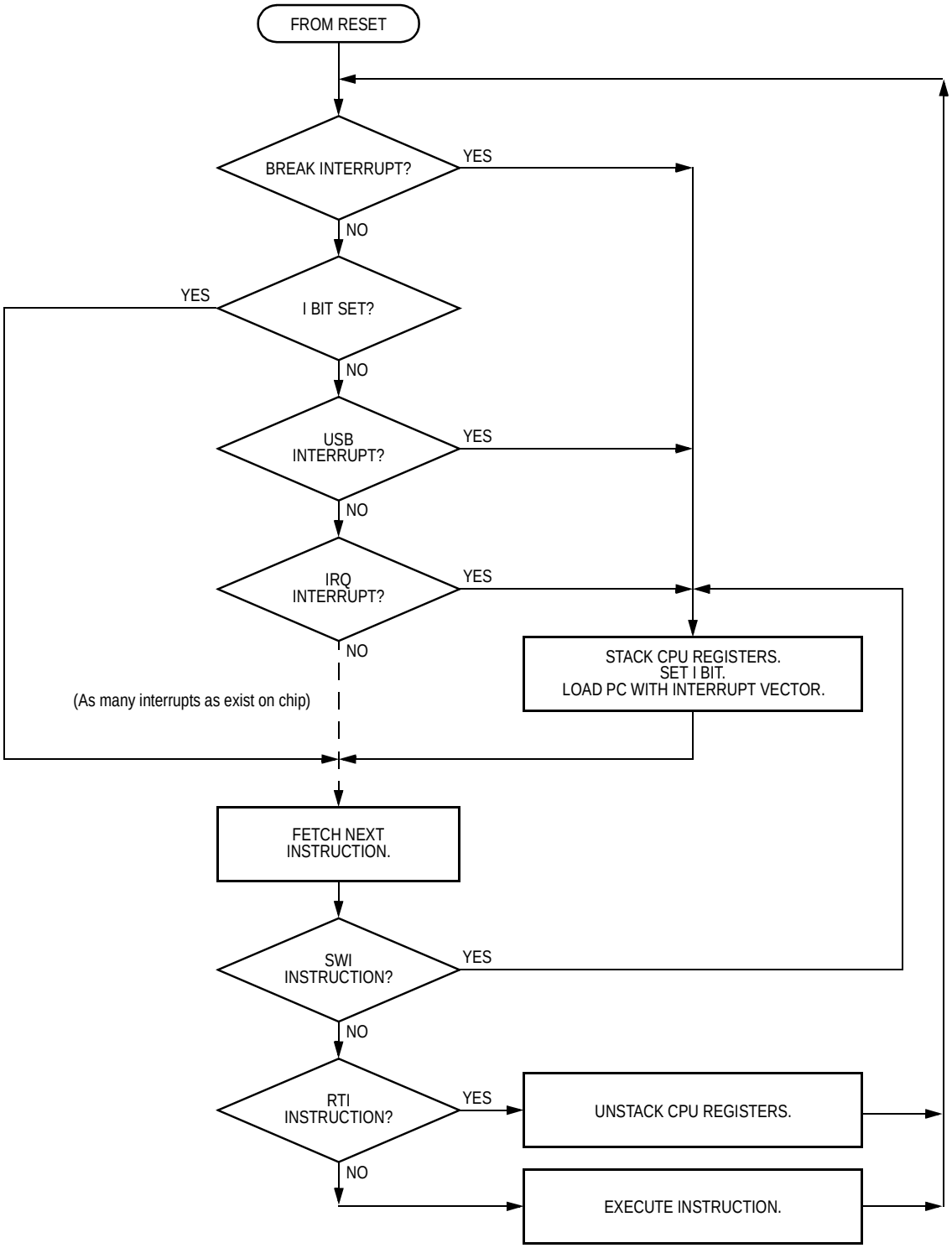


Figure 8-8. Interrupt Processing

Interrupts are latched and arbitration is performed in the SIM at the start of interrupt processing. The arbitration result is a constant that the CPU uses to determine which vector to fetch. Once an interrupt is latched by the SIM, no other interrupt can take precedence, regardless of priority, until the latched interrupt is serviced or the I bit is cleared.

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. **Figure 8-9** shows interrupt entry timing. **Figure 8-10** shows interrupt recovery timing.

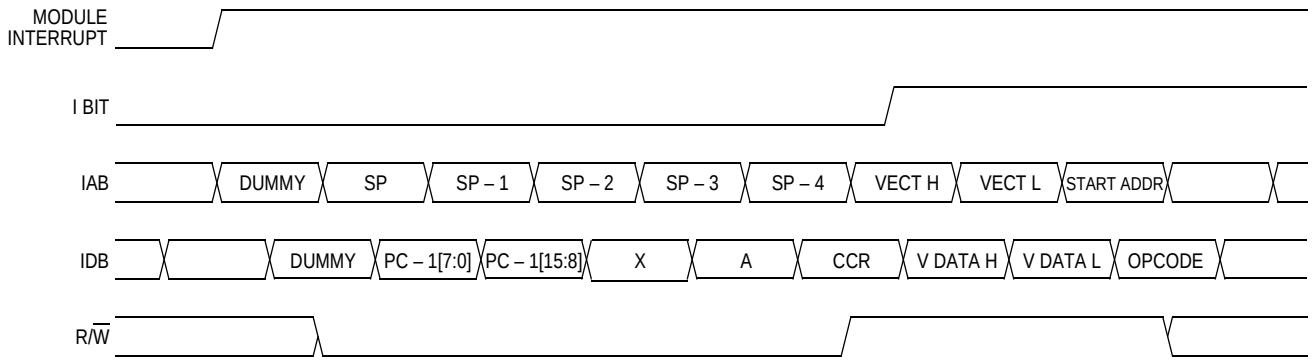


Figure 8-9. Interrupt Entry

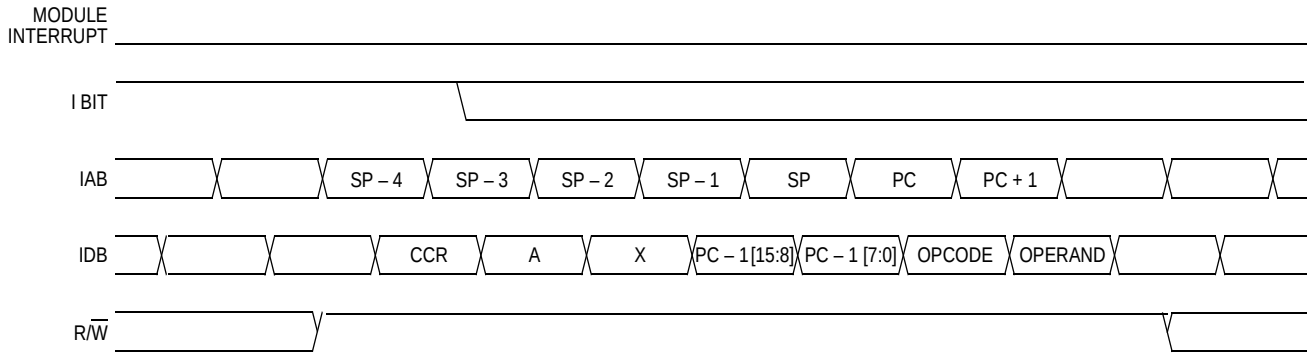


Figure 8-10. Interrupt Recovery

8.6.1.1 Hardware Interrupts

A hardware interrupt does not stop the current instruction. Processing of a hardware interrupt begins after completion of the current instruction. When the current instruction is complete, the SIM checks all pending hardware interrupts. If interrupts are not masked (I bit clear in the condition code register) and if the corresponding interrupt enable bit is set, the SIM proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If more than one interrupt is pending at the end of an instruction execution, the highest priority interrupt is serviced first. **Figure 8-11** demonstrates what happens when two interrupts are pending. If an interrupt is pending upon exit from the original interrupt service routine, the pending interrupt is serviced before the LDA instruction is executed.

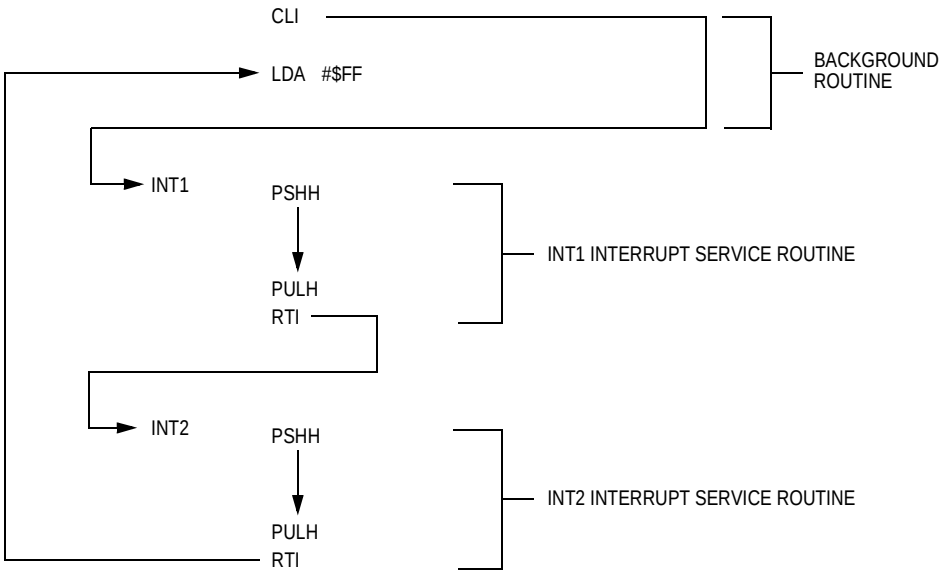


Figure 8-11. Interrupt Recognition Example

The LDA opcode is prefetched by both the INT1 and INT2 RTI instructions. However, in the case of the INT1 RTI prefetch, this is a redundant operation.

NOTE: *To maintain compatibility with the M6805 Family, the H register is not pushed on the stack during interrupt entry. If the interrupt service routine modifies the H register or uses the indexed addressing mode, software should save the H register and then restore it prior to exiting the routine.*

8.6.1.2 SWI Instruction

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

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

8.6.2 Interrupt Status Registers

The flags in the interrupt status registers identify maskable interrupt sources. **Table 8-4** summarizes the interrupt sources and the interrupt status register flags that they set. The interrupt status registers can be useful for debugging.

8.6.2.1 Interrupt Status Register 1

Address: $FE04

	Bit 7	6	5	4	3	2	1	Bit 0
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

R = Reserved

Figure 8-12. Interrupt Status Register 1 (INT1)

IF6–IF1 — Interrupt Flags 6–1

These flags indicate the presence of interrupt requests from the sources shown in **Table 8-4**.

1 = Interrupt request present
0 = No interrupt request present

Bit 1 and Bit 0 — Always read 0

Table 8-4. Interrupt Sources

Source	Flags	Mask[1]	INT Register Flag	Priority[2]	Vector Address
Reset	None	None	None	0	$FFFE–$FFFF
SWI instruction	None	None	None	0	$FFFC–$FFFD
USB reset interrupt	RSTF	URSTD	IF1	1	$FFFA–$FFFB
USB endpoint 0 transmit	TXD0F	TXD0IE			
USB endpoint 0 receive	RXD0F	RXD0IE			
USB endpoint 1 transmit	TXD1F	TXD1IE			
USB endpoint 2 transmit	TXD2F	TXD2IE			
USB endpoint 2 receive	RXD2F	RXD2IE			
USB end of packet	EOPF	EOPIE			
USB resume interrupt	RESUMF	—			
IRQ interrupt ($\overline{\text{IRQ}}$, PTE4)	IRQF, PTE4IF	IMASK	IF2	2	$FFF8–$FFF9
TIM 1 channel 0	CH0F	CH0IE	IF3	3	$FFF6–$FFF7
TIM 1 channel 1	CH1F	CH1IE	IF4	4	$FFF4–$FFF5
TIM 1 channel 0 & 1	CH0F & CH1F	CH01IE	IF5	5	$FFF2–$FFF3
TIM 1 overflow	TOF	TOIE	IF6	6	$FFF0–$FFF1
TIM 2 channel 0	CH0F	CH0IE	IF7	7	$FFEE–$FFEF
TIM 2 channel 1	CH1F	CH1IE	IF8	8	$FFEC–$FFED
TIM 2 channel 0 & 1	CH0F & CH1F	CH01IE	IF9	9	$FFEA–$FFEB
TIM 2 overflow	TOF	TOIE	IF10	10	$FFE8–$FFE9
SCI receiver overrun	OR	ORIE	IF11	11	$FFE6–$FFE7
SCI noise fag	NF	NEIE			
SCI framing error	FE	FEIE			
SCI parity error	PE	PEIE			
SCI receiver full	SCRF	SCRIE	IF12	12	$FFE4–$FFE5
SCI input idle	IDLE	ILIE			
SCI transmitter empty	SCTE	SCTIE	IF13	13	$FFE2–$FFE3
SCI transmission complete	TC	TCIE			
Keyboard interrupt	KEYF	IMASKK	IF14	14	$FFE0–$FFE1

Notes:
1. The I bit in the condition code register is a global mask for all interrupt sources except the SWI instruction.
2. Highest priority = 0.

8.6.2.2 Interrupt Status Register 2

Address: $FE05

	Bit 7	6	5	4	3	2	1	Bit 0
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

R = Reserved

Figure 8-13. Interrupt Status Register 2 (INT2)

IF14–IF7 — Interrupt Flags 14–7

These flags indicate the presence of interrupt requests from the sources shown in **Table 8-4**.

1 = Interrupt request present

0 = No interrupt request present

8.6.3 Reset

All reset sources always have equal and highest priority and cannot be arbitrated.

8.6.4 Break Interrupts

The break module can stop normal program flow at a software-programmable break point by asserting its break interrupt output. (See **Section 19. Break Module (BRK)**.) The SIM puts the CPU into the break state by forcing it to the SWI vector location. Refer to the break interrupt subsection of each module to see how each module is affected by the break state.

8.6.5 Status Flag Protection in Break Mode

The SIM controls whether status flags contained in other modules can be cleared during break mode. The user can select whether flags are protected from being cleared by properly initializing the break clear flag enable bit (BCFE) in the break flag control register (BFCR).

Protecting flags in break mode ensures that set flags will not be cleared while in break mode. This protection allows registers to be freely read and written during break mode without losing status flag information.

Setting the BCFE bit enables the clearing mechanisms. Once cleared in break mode, a flag remains cleared even when break mode is exited. Status flags with a 2-step clearing mechanism — for example, a read of one register followed by the read or write of another — are protected, even when the first step is accomplished prior to entering break mode. Upon leaving break mode, execution of the second step will clear the flag as normal.

8.7 Low-Power Modes

Executing the WAIT or STOP instruction puts the MCU in a low-power-consumption mode for standby situations. The SIM holds the CPU in a non-clocked state. The operation of each of these modes is described here. Both STOP and WAIT clear the interrupt mask (I) in the condition code register, allowing interrupts to occur.

8.7.1 Wait Mode

In wait mode, the CPU clocks are inactive while the peripheral clocks continue to run. **Figure 8-14** shows the timing for wait mode entry.

A module that is active during wait mode can wake up the CPU with an interrupt if the interrupt is enabled. Stacking for the interrupt begins one cycle after the WAIT instruction during which the interrupt occurred. In wait mode, the CPU clocks are inactive. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

Wait mode can also be exited by a reset or break. A break interrupt during wait mode sets the SIM break stop/wait bit, SBSW, in the SIM break status register (SBSR). If the COP disable bit, COPD, in the configuration register (CONFIG) is logic 0, then the computer operating properly module (COP) is enabled and remains active in wait mode.

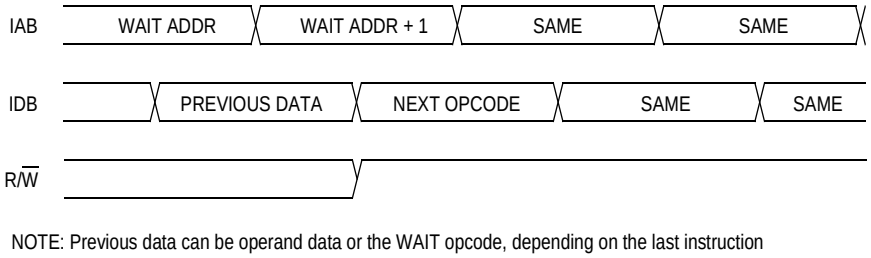


Figure 8-14. Wait Mode Entry Timing

Figure 8-15 and **Figure 8-16** show the timing for WAIT recovery.

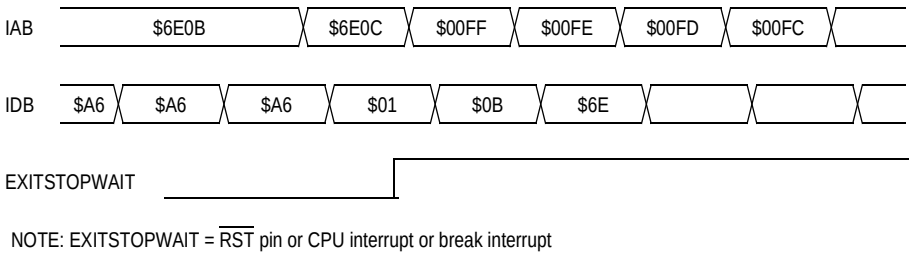


Figure 8-15. Wait Recovery from Interrupt or Break

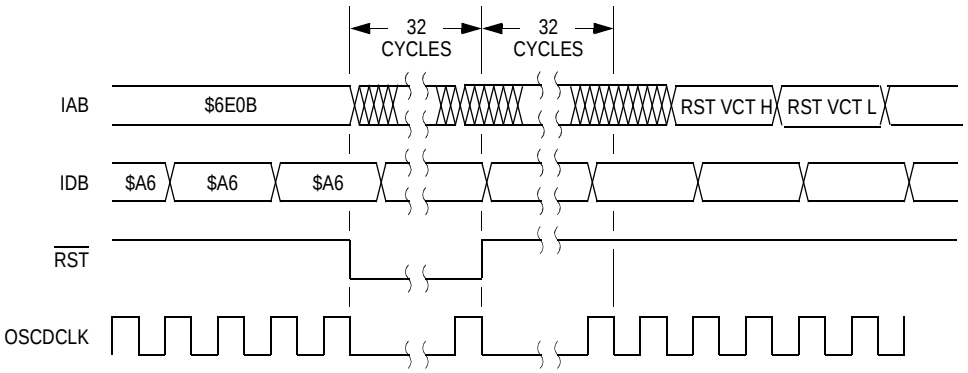


Figure 8-16. Wait Recovery from Internal Reset

8.7.2 Stop Mode

In stop mode, the SIM counter is reset and the system clocks are disabled. An interrupt request from a module can cause an exit from stop mode. Stacking for interrupts begins after the selected stop recovery time has elapsed. Reset or break also causes an exit from stop mode.

The SIM disables the oscillator signals (OSCOUT and OSCDCLK) in stop mode, stopping the CPU and peripherals. Stop recovery time is selectable using the SSREC bit in the configuration register (CONFIG). If SSREC is set, stop recovery is reduced from the normal delay of 4096 OSCDCLK cycles down to 2048. This is ideal for applications using canned oscillators that do not require long startup times from stop mode.

NOTE: *External crystal applications should use the full stop recovery time by clearing the SSREC bit.*

A break interrupt during stop mode sets the SIM break stop/wait bit (SBSW) in the SIM break status register (SBSR).

The SIM counter is held in reset from the execution of the STOP instruction until the beginning of stop recovery. It is then used to time the recovery period. **Figure 8-17** shows stop mode entry timing.

NOTE: *To minimize stop current, all pins configured as inputs should be driven to a logic 1 or logic 0.*

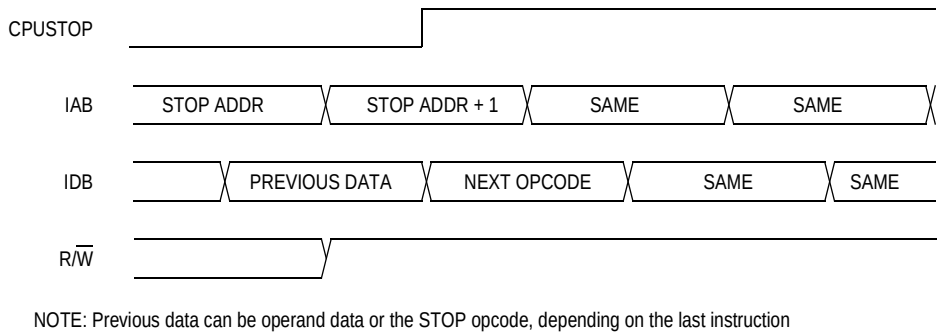


Figure 8-17. Stop Mode Entry Timing

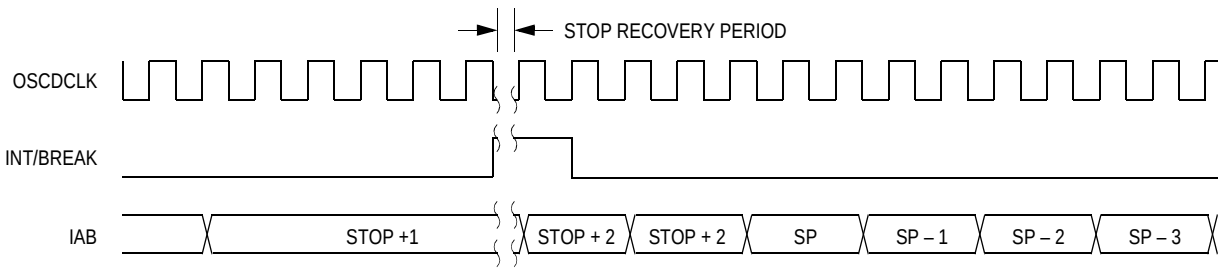


Figure 8-18. Stop Mode Recovery from Interrupt or Break

8.8 SIM Registers

The SIM has three memory mapped registers.

- SIM break status register (SBSR)
- SIM reset status register (SRSR)
- SIM break flag control register (SBFCR)

8.8.1 SIM Break Status Register (SBSR)

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

Address: $FE00

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	R	R	R	R	R	SBSW	R
Write:							Note 1	
Reset:							0	

Note 1. Writing a logic 0 clears SBSW.

R = Reserved

Figure 8-19. SIM Break Status Register (SBSR)

SBSW — SIM Break Stop/Wait

This status bit is useful in applications requiring a return to wait or stop mode after exiting from a break interrupt. Clear SBSW by writing a logic 0 to it. Reset clears SBSW.

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

SBSW can be read within the break state SWI routine. The user can modify the return address on the stack by subtracting one from it.

The following code is an example of this. Writing 0 to the SBSW bit clears it.

```
;This code works if the H register has been pushed onto the stack in the break
;service routine software. This code should be executed at the end of the break
;service routine software.

 HIBYTE   EQU     5

 LOBYTE   EQU     6

;         If not SBSW, do RTI

          BRCLR   SBSW,SBSR, RETURN    ;See if wait mode or stop mode was exited by
                                       ;break.

          TST     LOBYTE,SP            ;If RETURNLO is not zero,

          BNE     DOLO                 ;then just decrement low byte.

          DEC     HIBYTE,SP            ;Else deal with high byte, too.

 DOLO     DEC     LOBYTE,SP            ;Point to WAIT/STOP opcode.

 RETURN   PULH                         ;Restore H register.
          RTI
```

8.8.2 SIM Reset Status Register (SRSR)

This register contains seven flags that show the source of the last reset. All flag bits are cleared automatically following a read of the register. The register is initialized on power-up as shown with the POR bit set and all other bits cleared. However, during a POR or any other internal reset, the $\overline{\text{RST}}$ pin is pulled low. After the pin is released, it will be sampled 32 OSCDCLK cycles later. If the pin is not above a V_{IH} at that time, then the PIN bit in the SRSR may be set in addition to whatever other bits are set.

Address: $FE01

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	POR	PIN	COP	ILOP	ILAD	USB	LVI	0
Write:								
POR:	1	0	0	0	0	0	0	0

= Unimplemented

Figure 8-20. SIM Reset Status Register (SRSR)

POR — Power-On Reset Bit

1 = Last reset caused by POR circuit

0 = Read of SRSR

PIN — External Reset Bit
1 = Last reset caused by external reset pin ($\overline{RST}$)
0 = POR or read of SRSR

COP — Computer Operating Properly Reset Bit
1 = Last reset caused by COP counter
0 = POR or read of SRSR

ILOP — Illegal Opcode Reset Bit
1 = Last reset caused by an illegal opcode
0 = POR or read of SRSR

ILAD — Illegal Address Reset Bit (opcode fetches only)
1 = Last reset caused by an opcode fetch from an illegal address
0 = POR or read of SRSR

USB — Universal Serial Bus Reset Bit
1 = Last reset caused by the USB module
0 = POR or read of SRSR

LVI — Low Voltage Inhibit Reset Bit
1 = Last reset caused by the LVI circuit
0 = POR or read of SRSR

8.8.3 SIM Break Flag Control Register (SBFCR)

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

Address: $FE03

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	BCFE	R	R	R	R	R	R	R
Reset:	0							

R = Reserved

Figure 8-21. 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

Section 9. Monitor ROM (MON)

9.1 Contents

9.2 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 host computer. This mode is also used for programming and erasing of FLASH memory in the MCU. Monitor mode entry can be achieved without use of the higher voltage, V_{TST}, as long as vector addresses $FFFE and $FFFF are blank, thus reducing the hardware requirements for in-circuit programming.

9.3 Features

Features of the monitor ROM include the following:

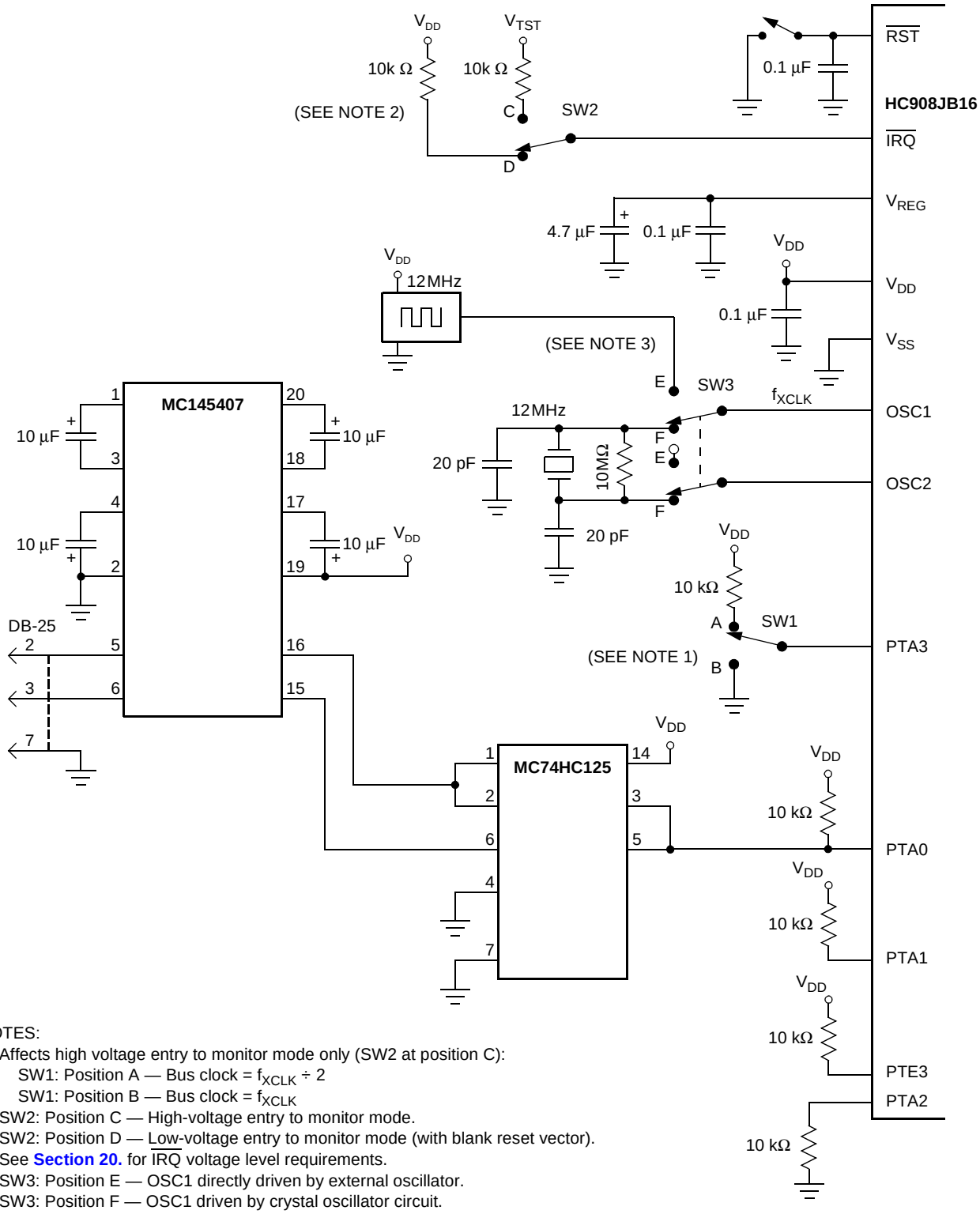
- Normal user-mode pin functionality
- One pin dedicated to serial communication between monitor ROM and host computer
- Standard mark/space non-return-to-zero (NRZ) communication with host computer
- Execution of code in RAM or FLASH
- FLASH memory security feature[1]
- FLASH memory programming interface
- 1,472 bytes monitor ROM code size
- 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 $\overline{IRQ}$

9.4 Functional Description

The monitor ROM receives and executes commands from a host computer. **Figure 9-1** shows a 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 host-computer code in RAM 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 pull-up resistor.

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



NOTES:
1. Affects high voltage entry to monitor mode only (SW2 at position C):
   SW1: Position A — Bus clock = $f_{XCLK} \div 2$
   SW1: Position B — Bus clock = f_{XCLK}
2. SW2: Position C — High-voltage entry to monitor mode.
   SW2: Position D — Low-voltage entry to monitor mode (with blank reset vector).
   See **Section 20.** for $\overline{IRQ}$ voltage level requirements.
3. SW3: Position E — OSC1 directly driven by external oscillator.
   SW3: Position F — OSC1 driven by crystal oscillator circuit.

Figure 9-1. Monitor Mode Circuit

9.4.1 Entering Monitor Mode

Table 9-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a POR and will allow communication at 19200 baud provided one of the following sets of conditions is met:

1. If $\overline{IRQ}$ = V_{TST}:
   - External clock on OSC1 is 12 MHz
   - PTA3 = high
   - PTE3 = high
2. If $FFFE & $FFFF is blank (contains $FF):
   - External clock on OSC1 is 12 MHz
   - $\overline{IRQ}$ = V_{DD}
   - PTE3 = high

Table 9-1. Mode Entry Requirements and Options

$\overline{IRQ}$	$FFFE and $FFFF	PTE3	PTA3[1]	PTA2	PTA1	PTA0	External Clock, f_{XCLK}	Bus Frequency, f_{BUS}	Comments
V_{TST}[2]	X	1	0	0	1	1	12 MHz	12 MHz (f_{XCLK})	High-voltage entry to monitor mode. 38400 baud communication on PTA0. COP disabled.
V_{TST}[2]	X	1	1	0	1	1	12 MHz	6 MHz (f_{XCLK} ÷ 2)	High-voltage entry to monitor mode. 19200 baud communication on PTA0. COP disabled.
V_{DD}	BLANK (contain $FF)	1	X	X	X	1	12 MHz	6 MHz (f_{XCLK} ÷ 2)	Low-voltage entry to monitor mode. 19200 baud communication on PTA0. COP disabled.
V_{DD}	NOT BLANK	1	X	X	X	X	12 MHz	6 MHz (f_{XCLK} ÷ 2)	Enters user mode. If $FFFE and $FFFF is blank, MCU will encounter an illegal address reset.

Factory use only

Notes:
1. PTA3 = 0: Bypasses the divide-by-two prescaler to SIM when using V_{TST} for monitor mode entry.
2. See **Section 20. Electrical Specifications** for V_{TST} voltage level requirements.

If V_{TST} is applied to $\overline{IRQ}$ and PTA3 is low upon monitor mode entry (**Table 9-1** condition set 1), the bus frequency is a equal to the external clock, f_{XCLK}. If PTA3 is high with V_{TST} applied to $\overline{IRQ}$ upon monitor mode entry (**Table 9-1** condition set 2), the bus frequency is a divide-by-two of the external clock. Holding the PTA3 pin low when entering monitor mode causes a bypass of a divide-by-two stage at the oscillator *only if V_{TST} is applied to $\overline{IRQ}$.* In this event, the OSCOUT frequency is equal to the OSCDCLK frequency.

Entering monitor mode with V_{TST} on $\overline{IRQ}$, the COP is disabled as long as V_{TST} is applied to either the $\overline{IRQ}$ or the $\overline{RST}$. (See **Section 8. System Integration Module (SIM)** for more information on modes of operation.)

If entering monitor mode without high voltage on $\overline{IRQ}$ and reset vector being blank ($FFFE and $FFFF) (**Table 9-1** condition set 3, where $\overline{IRQ}$ applied voltage is V_{DD}), then all port A pin requirements and conditions, including the PTA3 frequency divisor selection, are not in effect. This is to reduce circuit requirements when performing in-circuit programming.

Entering monitor mode with the reset vector being blank, the COP is always disabled regardless of the state of $\overline{IRQ}$ or the $\overline{RST}$.

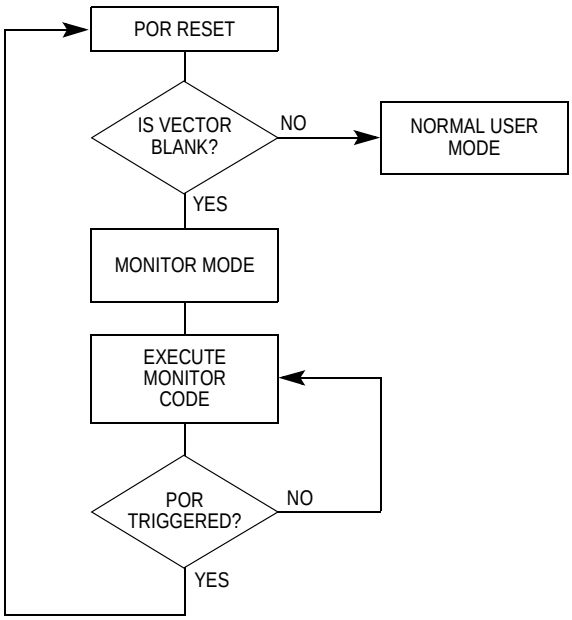


Figure 9-2. Low-Voltage Monitor Mode Entry Flowchart

Figure 9-2. shows a simplified diagram of the monitor mode entry when the reset vector is blank and $\overline{IRQ}$ = V_{DD}. An external clock of 12 MHz is required for a baud rate of 19200.

Enter monitor mode with the pin configuration shown in Figure 9-1 by pulling $\overline{RST}$ low and then high. The rising edge of $\overline{RST}$ latches monitor mode. Once monitor mode is latched, the values on the specified pins can change.

Once out of reset, the MCU waits for the host to send eight security bytes. (See 9.5 Security.) After the security bytes, the MCU sends a break signal (10 consecutive logic zeros) to the host, indicating that it is ready to receive a command. The break signal also provides a timing reference to allow the host to determine the necessary baud rate.

In monitor mode, the MCU uses different vectors for reset, SWI (software interrupt), and break interrupt than those for user mode. The alternate vectors are in the $FE page instead of the $FF page and allow code execution from the internal monitor firmware instead of user code.

Table 9-2 is a summary of the vector differences between user mode and monitor mode.

Table 9-2. Monitor Mode Vector Differences

Modes	Functions						
	COP	Reset Vector High	Reset Vector Low	Break Vector High	Break Vector Low	SWI Vector High	SWI Vector Low
User	Enabled	$FFFE	$FFFF	$FFFC	$FFFD	$FFFC	$FFFD
Monitor	Disabled(1)	$FEFE	$FEFF	$FEFC	$FEFD	$FEFC	$FEFD

Notes:
1. If the high voltage (V_{TST}) is removed from the $\overline{IRQ}$ pin or the $\overline{RST}$ pin, the SIM asserts its COP enable output. The COP is a mask option enabled or disabled by the COPD bit in the configuration register.

9.4.2 Data Format

Communication with the monitor ROM is in standard non-return-to-zero (NRZ) mark/space data format. Transmit and receive baud rates must be identical.

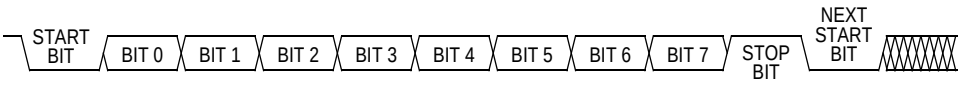


Figure 9-3. Monitor Data Format

9.4.3 Break Signal

A start bit (logic 0) followed by nine logic 0 bits is a break signal. When the monitor receives a break signal, it drives the PTA0 pin high for the duration of two bits and then echoes back the break signal.

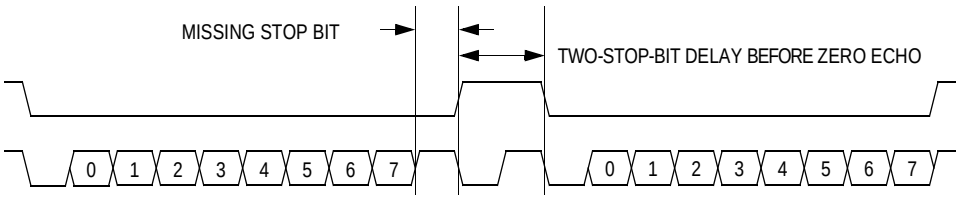


Figure 9-4. Break Transaction

9.4.4 Baud Rate

The communication baud rate is dependant on oscillator frequency, f_{XCLK}. The state of PTA3 also affects baud rate if entry to monitor mode is by $\overline{IRQ} = V_{TST}$. When PTA3 is high, the divide by ratio is 625. If the PTA3 pin is at logic zero upon entry into monitor mode, the divide by ratio is 312.

Table 9-3. Monitor Baud Rate Selection

Monitor Mode Entry By:	Oscillator Clock Frequency, f_{CLK}	PTA3	Baud Rate
$\overline{IRQ} = V_{TST}$	12 MHz	0	38400 bps
	12 MHz	1	19200 bps
Blank reset vector, $\overline{IRQ} = V_{DD}$	12 MHz	X	19200 bps

9.4.5 Commands

The monitor ROM uses the following commands:

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

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

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

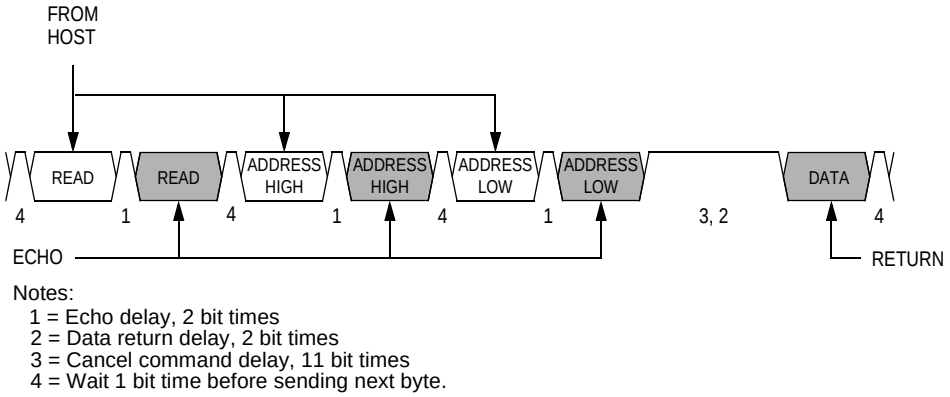
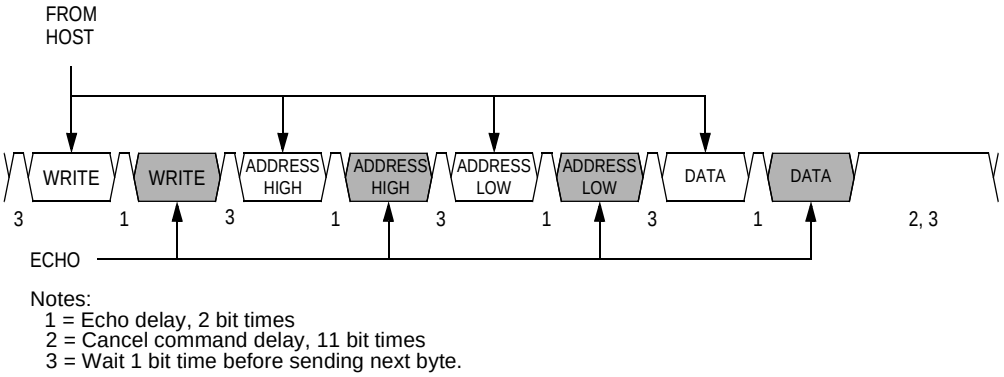


Figure 9-5. Read Transaction



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

Figure 9-6. Write Transaction

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

Table 9-4. READ (Read Memory) Command

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

Command Sequence

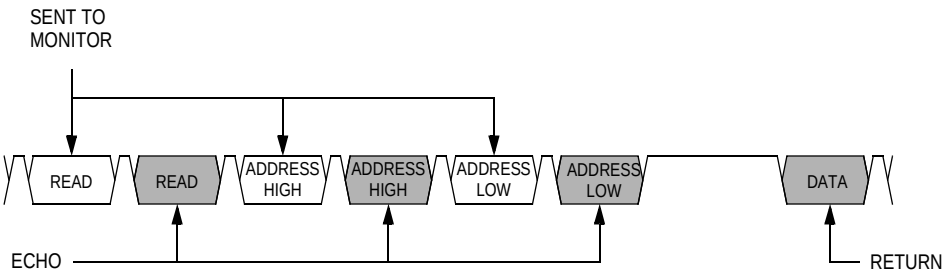


Table 9-5. WRITE (Write Memory) Command

Description	Write byte to memory
Operand	Specifics 2-byte address in high byte:low byte order; low byte followed by data byte
Data Returned	None
Opcode	$49

Command Sequence

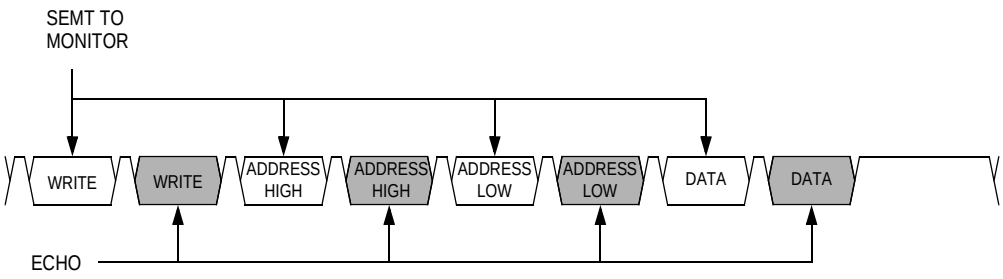


Table 9-6. IREAD (Indexed Read) Command

Description	Read next 2 bytes in memory from last address accessed
Operand	Specifies 2-byte address in high byte:low byte order
Data Returned	Returns contents of next two addresses
Opcode	$1A

Command Sequence

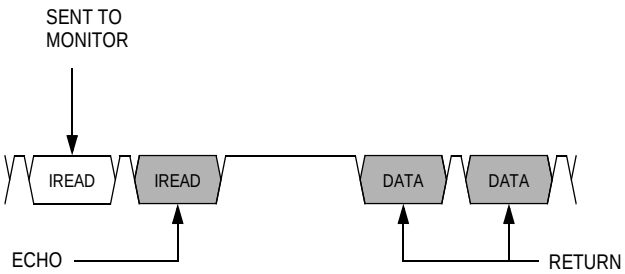
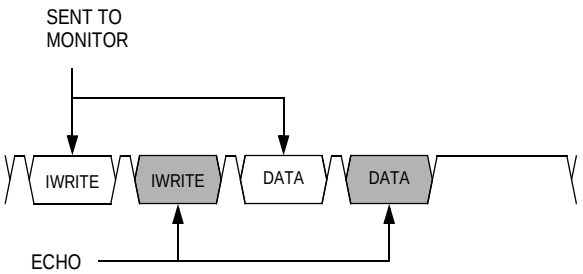


Table 9-7. IWRITE (Indexed Write) Command

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

NOTE: *A sequence of IREAD or IWRITE commands can sequentially access a block of memory over the full 64k-byte memory map.*

Table 9-8. READSP (Read Stack Pointer) Command

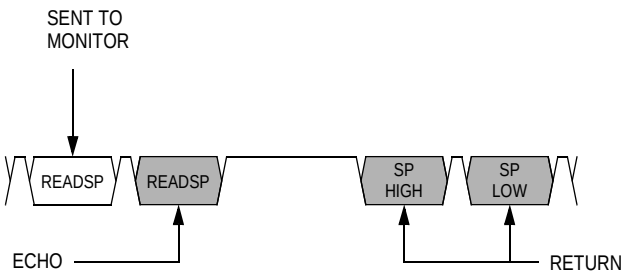
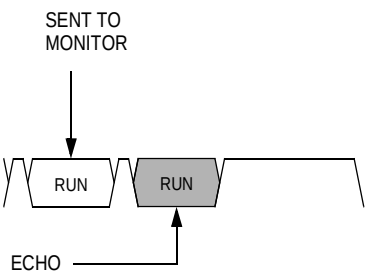
Description	Reads stack pointer
Operand	None
Data Returned	Returns stack pointer in high byte:low byte order
Opcode	$0C
Command Sequence	

Table 9-9. RUN (Run User Program) Command

Description	Executes RTI instruction
Operand	None
Data Returned	None
Opcode	$28
Command Sequence	

The MCU executes the SWI and PSHH instructions when it enters monitor mode. The RUN command tells the MCU to execute the PULH and RTI instructions. Before sending the RUN command, the host can modify the stacked CPU registers to prepare to run the host program. The READSP command returns the incremented stack pointer value, SP + 1. The high and low bytes of the program counter are at addresses SP + 5 and SP + 6.

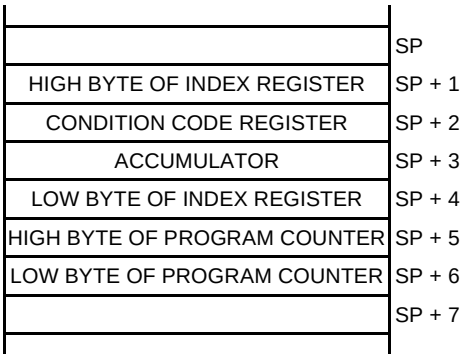


Figure 9-7. Stack Pointer at Monitor Mode Entry

9.5 Security

A security feature discourages unauthorized reading of FLASH locations while in monitor mode. The host can bypass the security feature at monitor mode entry by sending eight security bytes that match the bytes at locations $FFF6–$FFFD. Locations $FFF6–$FFFD contain user-defined data.

NOTE: *Do not leave locations $FFF6–$FFFD blank. For security reasons, program locations $FFF6–$FFFD.*

During monitor mode entry, the MCU waits after the power-on reset for the host to send the eight security bytes on pin PTA0. If the received bytes match those at locations $FFF6–$FFFD, the host bypasses the security feature and can read all FLASH locations and execute code from FLASH. Security remains bypassed until a power-on or an LVI reset occurs. If the reset was not a power-on or an LVI reset, security remains bypassed and security code entry is not required.
(See **Figure 9-8**.)

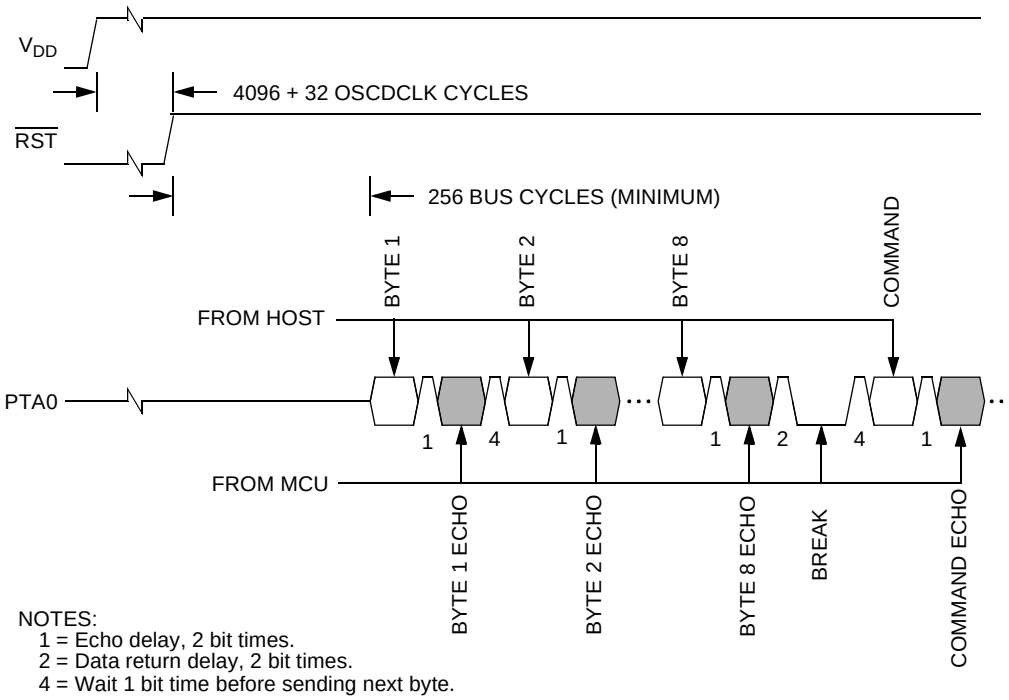


Figure 9-8. Monitor Mode Entry Timing

Upon power-on reset, if the received bytes of the security code do not match the data at locations $FFF6–$FFFD, the host fails to bypass the security feature. The MCU remains in monitor mode, but reading a FLASH location returns an invalid value and trying to execute code from FLASH causes an illegal address reset. After receiving the eight security bytes from the host, the MCU transmits a break character, signifying that it is ready to receive a command.

NOTE: *The MCU does not transmit a break character until after the host sends the eight security bytes.*

To determine whether the security code entered is correct, check to see if bit 6 of RAM address $80 is set. If it is, then the correct security code has been entered and FLASH can be accessed.

If the security sequence fails, the device should be reset by a power-on reset and brought up in monitor mode to attempt another entry. After failing the security sequence, the FLASH module can also be mass erased by executing an erase routine that was downloaded into internal RAM. The mass erase operation clears the security code locations so that all eight security bytes become $FF (blank).

9.5.1 Extended Security

To further disable monitor mode functions, the monitor commands can be disabled by writing $7B to the FLASH location $FFD1 and $87 to the FLASH location $FFD0. **Table 9-10** shows the security settings that affect monitor mode operations.

Table 9-10. Monitor Mode Security

Extended Security	Monitor Mode Entry Security	Monitor Functions Available
NOT SET	BYPASSED	Read/write of RAM and FLASH.
	FAILED	Read/write of RAM. Read of FLASH disabled. FLASH can only be mass erased.
SET	BYPASSED	Read/write of RAM and FLASH disabled.
	FAILED	Read/write of RAM. Read of FLASH disabled. FLASH can only be mass erased.

Section 10. Timer Interface Module (TIM)

10.1 Contents

10.2 Introduction

This section describes the timer interface (TIM) module. The TIM is a two-channel timer that provides a timing reference with input capture, output compare, and pulse-width-modulation functions. **Figure 10-1** is a block diagram of the TIM.

This particular MCU has two timer interface modules which are denoted as TIM1 and TIM2.

NOTE: *TIM1 and TIM2 each have channel 0 and channel 1 I/Os connected together, forming a common I/O. Because of this common I/O, both channels should not be simultaneously configured for output compare functions, otherwise, port pin contention will occur.*

10.3 Features

Features of the TIM include:

- Two input capture/output compare channels on one common I/O:
  - Rising-edge, falling-edge, or any-edge input capture trigger
  - Set, clear, or toggle output compare action
- Buffered and unbuffered pulse-width-modulation (PWM) signal generation
- Programmable TIM clock input
  - 7-frequency internal bus clock prescaler selection
  - External TIM clock input (bus frequency ÷2 maximum)
- Free-running or modulo up-count operation
- Toggle any channel pin on overflow
- TIM counter stop and reset bits

10.4 Pin Name Conventions

The text that follows describes both timers, TIM1 and TIM2. The TIM input/output (I/O) pin names are T[1,2]CH01 (timer channel 01), where “1” is used to indicate TIM1 and “2” is used to indicate TIM2. The two TIMs share two I/O pins with two I/O port pins. The full names of the TIM I/O pins are listed in **Table 10-1**. The generic pin names appear in the text that follows.

Table 10-1. Pin Name Conventions

TIM Generic Pin Names:		T[1,2]CH01	TCLK
Full TIM Pin Names:	TIM1	PTE1/T1CH01	PTE0/TCLK
	TIM2	PTE2/T2CH01	

NOTE: *References to either timer 1 or timer 2 may be made in the following text by omitting the timer number. For example, TCH01 may refer generically to T1CH01 and T2CH01.*

10.5 Functional Description

Figure 10-1 shows the structure of the TIM. The central component of the TIM is the 16-bit TIM counter that can operate as a free-running counter or a modulo up-counter. The TIM counter provides the timing reference for the input capture and output compare functions. The TIM counter modulo registers, TMODH:TMODL, control the modulo value of the TIM counter. Software can read the TIM counter value at any time without affecting the counting sequence.

Channel 0 and channel 1 I/Os are connected together, forming a common I/O. Although the two TIM channels are programmable independently as input capture channels, the input capture signal will be the same for both channels. Output compare functions should only be enabled for one channel to avoid I/O contention.

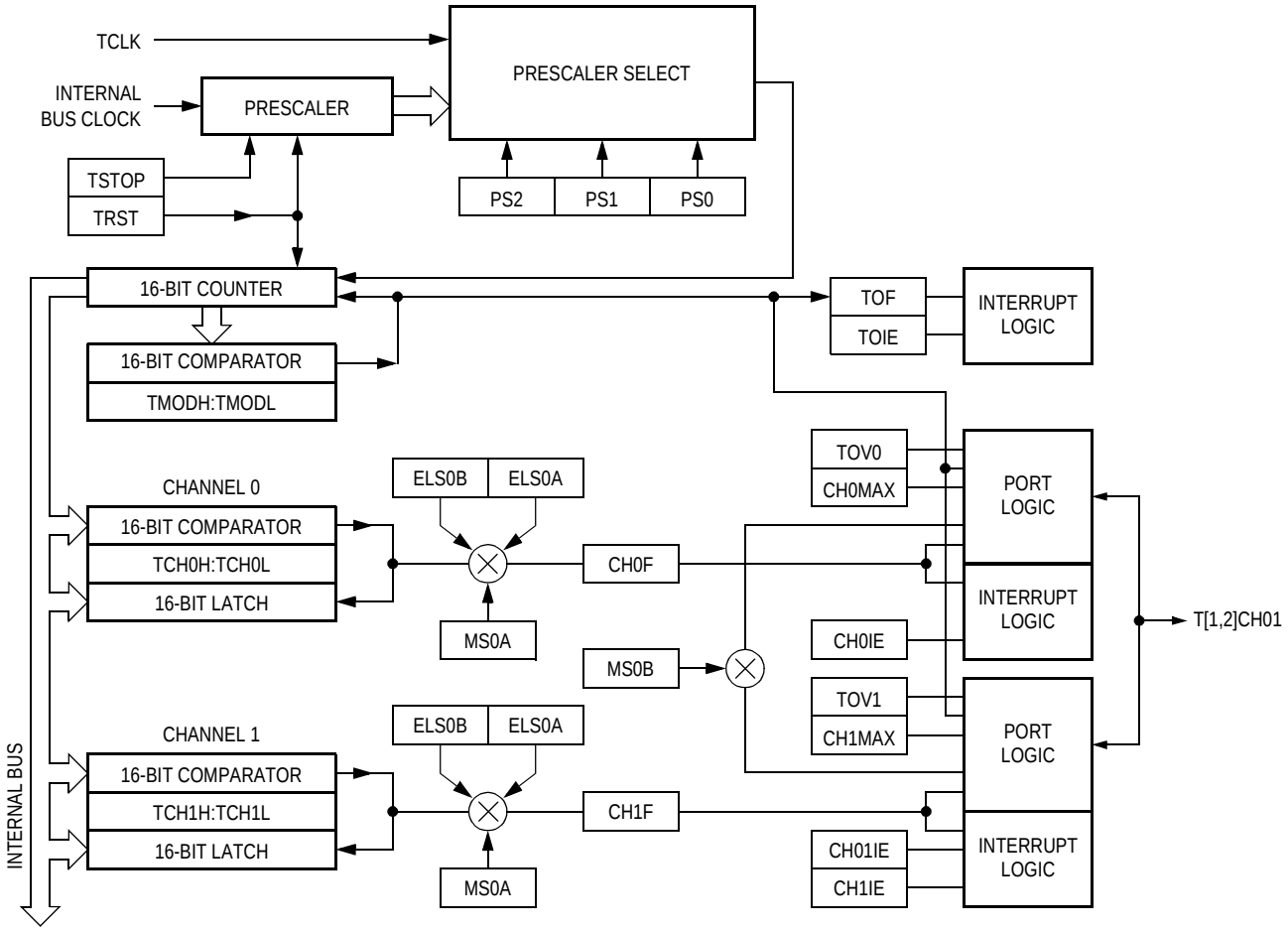


Figure 10-1. TIM Block Diagram

Figure 10-2 summarizes the timer registers.

NOTE: *References to either timer 1 or timer 2 may be made in the following text by omitting the timer number. For example, TSC may generically refer to both T1SC and T2SC.*

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000A	Timer 1 Status and Control Register (T1SC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$000C	Timer 1 Counter Register High (T1CNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000D	Timer 1 Counter Register Low (T1CNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000E	Timer 1 Counter Modulo Register High (T1MODH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$000F	Timer 1 Counter Modulo Register Low (T1MODL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0010	Timer 1 Channel 0 Status and Control Register (T1SC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0011	Timer 1 Channel 0 Register High (T1CH0H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0012	Timer 1 Channel 0 Register Low (T1CH0L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0013	Timer 1 Channel 1 Status and Control Register (T1SC1)	Read:	CH1F	CH1IE	CH01IE	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-2. TIM I/O Register Summary (Sheet 1 of 3)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0014	Timer 1 Channel 1 Register High (T1CH1H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							
$0015	Timer 1 Channel 1 Register Low (T1CH1L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							
$0040	Timer 2 Status and Control Register (T2SC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0042	Timer 2 Counter Register High (T2CNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0043	Timer 2 Counter Register Low (T2CNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0044	Timer 2 Counter Modulo Register High (T2MODH)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
$0045	Timer 2 Counter Modulo Register Low (T2MODL)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$0046	Timer 2 Channel 0 Status and Control Register (T2SC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0047	Timer 2 Channel 0 Register High (T2CH0H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							

= Unimplemented

Figure 10-2. TIM I/O Register Summary (Sheet 2 of 3)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0048	Timer 2 Channel 0 Register Low (T2CH0L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							
$0049	Timer 2 Channel 1 Status and Control Register (T2SC1)	Read:	CH1F	CH1IE	CH01IE	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$004A	Timer 2 Channel 1 Register High (T2CH1H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							
$004B	Timer 2 Channel 1 Register Low (T2CH1L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							

= Unimplemented

Figure 10-2. TIM I/O Register Summary (Sheet 3 of 3)

10.5.1 TIM Counter Prescaler

The TIM clock source can be one of the seven prescaler outputs or the TIM clock pin, PTE0/TCLK. The prescaler generates seven clock rates from the internal bus clock. The prescaler select bits, PS[2:0], in the TIM status and control register (TSC) select the TIM clock source.

10.5.2 Input Capture

With the input capture function, the TIM can capture the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the TIM latches the contents of the TIM counter into the TIM channel registers, TCHxH:TCHxL. The polarity of the active edge is programmable. Input captures can generate TIM CPU interrupt requests.

10.5.3 Output Compare

With the output compare function, the TIM can generate a periodic pulse with a programmable polarity, duration, and frequency. When the counter reaches the value in the registers of an output compare channel, the TIM can set, clear, or toggle the channel pin. Output compares can generate TIM CPU interrupt requests.

10.5.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in **10.5.3 Output Compare**. The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change an output compare value could cause incorrect operation for up to two counter overflow periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that counter overflow period. Also, using a TIM overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIM may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the output compare value on channel x:

- When changing to a smaller value, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current output compare pulse. The interrupt routine has until the end of the counter overflow period to write the new value.
- When changing to a larger output compare value, enable TIM overflow interrupts and write the new value in the TIM overflow interrupt routine. The TIM overflow interrupt occurs at the end of the current counter overflow period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same counter overflow period.

10.5.3.2 Buffered Output Compare

Channels 0 and 1 can be linked to form a buffered output compare channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the output.

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The output compare value in the TIM channel 0 registers initially controls the output on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the output after the TIM overflows. At each subsequent overflow, the TIM channel registers (0 or 1) that control the output are the ones written to last. TSC0 controls and monitors the buffered output compare function, and TIM channel 1 status and control register (TSC1) is unused. While the MS0B bit is set, the channel 1 pin, TCH1, is available as a general-purpose I/O pin.

NOTE: *In buffered output compare operation, do not write new output compare values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered output compares.*

10.5.4 Pulse Width Modulation (PWM)

By using the toggle-on-overflow feature with an output compare channel, the TIM can generate a PWM signal. The value in the TIM counter modulo registers determines the period of the PWM signal. The channel pin toggles when the counter reaches the value in the TIM counter modulo registers. The time between overflows is the period of the PWM signal.

As **Figure 10-3** shows, the output compare value in the TIM channel registers determines the pulse width of the PWM signal. The time between overflow and output compare is the pulse width. Program the TIM to clear the channel pin on output compare if the state of the PWM pulse is logic 1. Program the TIM to set the pin if the state of the PWM pulse is logic 0.

The value in the TIM counter modulo registers and the selected prescaler output determines the frequency of the PWM output. The frequency of an 8-bit PWM signal is variable in 256 increments. Writing $00FF (255) to the TIM counter modulo registers produces a PWM period of 256 times the internal bus clock period if the prescaler select value is $000. See **10.10.1 TIM Status and Control Register**.

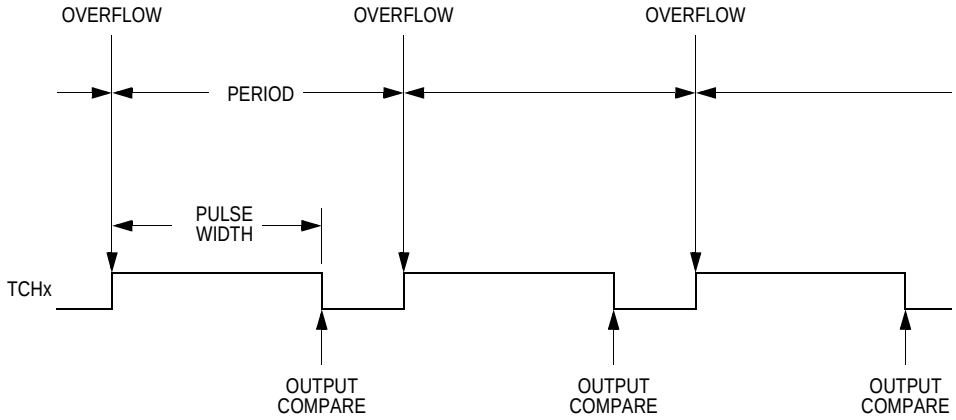


Figure 10-3. PWM Period and Pulse Width

The value in the TIM channel registers determines the pulse width of the PWM output. The pulse width of an 8-bit PWM signal is variable in 256 increments. Writing $0080 (128) to the TIM channel registers produces a duty cycle of 128/256 or 50%.

10.5.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in **10.5.4 Pulse Width Modulation (PWM)**. The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change a pulse width value could cause incorrect operation for up to two PWM periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that PWM period. Also, using a TIM overflow interrupt routine to write a new, smaller pulse width value may cause the compare to be missed. The TIM may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the PWM pulse width on channel x:

- When changing to a shorter pulse width, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current pulse. The interrupt routine has until the end of the PWM period to write the new value.
- When changing to a longer pulse width, enable TIM overflow interrupts and write the new value in the TIM overflow interrupt routine. The TIM overflow interrupt occurs at the end of the current PWM period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same PWM period.

NOTE: *In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare also can cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.*

10.5.4.2 Buffered PWM Signal Generation

Channels 0 and 1 can be linked to form a buffered PWM channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The TIM channel 0 registers initially control the pulse width on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIM channel registers (0 or 1) that control the pulse width are the ones written to last. TSC0 controls and monitors the buffered PWM function, and TIM channel 1 status and control register (TSC1) is unused. While the MS0B bit is set, the channel 1 pin, TCH1, is available as a general-purpose I/O pin.

NOTE: *In buffered PWM signal generation, do not write new pulse width values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered PWM signals.*

10.5.4.3 PWM Initialization

To ensure correct operation when generating unbuffered or buffered PWM signals, use the following initialization procedure:

1. In the TIM status and control register (TSC):
   a. Stop the TIM counter by setting the TIM stop bit, TSTOP.
   b. Reset the TIM counter and prescaler by setting the TIM reset bit, TRST.
2. In the TIM counter modulo registers (TMODH:TMODL), write the value for the required PWM period.
3. In the TIM channel x registers (TCHxH:TCHxL), write the value for the required pulse width.
4. In TIM channel x status and control register (TSCx):
   a. Write 0:1 (for unbuffered output compare or PWM signals) or 1:0 (for buffered output compare or PWM signals) to the mode select bits, MSxB:MSxA. (See **Table 10-3**.)
   b. Write 1 to the toggle-on-overflow bit, TOVx.
   c. Write 1:0 (to clear output on compare) or 1:1 (to set output on compare) to the edge/level select bits, ELSxB:ELSxA. The output action on compare must force the output to the complement of the pulse width level. (See **Table 10-3**.)

NOTE: *In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare can also cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.*

5. In the TIM status control register (TSC), clear the TIM stop bit, TSTOP.

Setting MS0B links channels 0 and 1 and configures them for buffered PWM operation. The TIM channel 0 registers (TCH0H:TCH0L) initially control the buffered PWM output. TIM status control register 0 (TSCR0) controls and monitors the PWM signal from the linked channels.

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 **10.10.4 TIM Channel Status and Control Registers**.)

10.6 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 (CH1F: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. Channel x TIM CPU interrupt requests are enabled when CHxIE = 1. CHxF and CHxIE are in the TIM channel x status and control register.

10.7 Low-Power Modes

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

10.7.1 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.

10.7.2 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 after an external interrupt.

10.8 TIM During Break Interrupts

A break interrupt stops the TIM counter.

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 **8.8.3 SIM Break Flag Control Register (SBFCR)**.)

To allow software to clear status bits during a break interrupt, write a logic 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 logic 0 to the BCFE bit. With BCFE at logic 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 break, the bit cannot change during the break state as long as BCFE is at logic 0. After the break, doing the second step clears the status bit.

10.9 I/O Signals

Port E shares three of its pins with the TIM. PTE0/TCLK is an external clock input to the TIM prescaler. The two TIM channel I/O pins are PTE1/T1CH01 and PTE2/T2CH01.

10.9.1 TIM Clock Pin (PTE0/TCLK)

PTE0/TCLK is an external clock input that can be the clock source for the TIM counter instead of the prescaled internal bus clock. Select the PTE0/TCLK input by writing logic 1s to the three prescaler select bits, PS[2:0]. (See **10.10.1 TIM Status and Control Register**.) The minimum TCLK pulse width, $TCLK_{LMIN}$ or $TCLK_{HMIN}$, is:

$$\frac{1}{\text{bus frequency}} + t_{SU}$$

The maximum TCLK frequency is:

$$\text{bus frequency} \div 2$$

PTE0/TCLK is available as a general-purpose I/O pin when not used as the TIM clock input. When the PTE0/TCLK pin is the TIM clock input, it is an input regardless of the state of the DDRE0 bit in data direction register E.

10.9.2 TIM Channel I/O Pins (PTE1/T1CH01:PTE2/T2CH01)

Each TIM I/O pin is programmable independently as an input capture pin or an output compare pin, or configured as buffered output compare or buffered PWM pins.

10.10 I/O Registers

NOTE: *References to either timer 1 or timer 2 may be made in the following text by omitting the timer number. For example, TSC may generically refer to both T1SC and T2SC.*

These I/O registers control and monitor operation of the TIM:

- TIM status and control register (TSC)
- TIM counter registers (TCNTH:TCNTL)
- TIM counter modulo registers (TMODH:TMODL)
- TIM channel status and control registers (TSC0, TSC1)
- TIM channel registers (TCH0H:TCH0L, TCH1H:TCH1L)

10.10.1 TIM Status and Control Register

The TIM status and control register (TSC):

- Enables TIM overflow interrupts
- Flags TIM overflows
- Stops the TIM counter
- Resets the TIM counter
- Prescales the TIM counter clock

Address: T1SC, $000A and T2SC, $0040

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
Write:	0			TRST				
Reset:	0	0	1	0	0	0	0	0

= Unimplemented

Figure 10-4. TIM Status and Control Register (TSC)

TOF — TIM Overflow Flag Bit

This read/write flag is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. Clear TOF by reading the TIM status and control register when TOF is set and then writing a logic 0 to TOF. If another TIM overflow occurs before the clearing sequence is complete, then writing logic 0 to TOF has no effect. Therefore, a TOF interrupt request cannot be lost due to inadvertent clearing of TOF. Reset clears the TOF bit. Writing a logic 1 to TOF has no effect.

1 = TIM counter has reached modulo value
0 = TIM counter has not reached modulo value

TOIE — TIM Overflow Interrupt Enable Bit

This read/write bit enables TIM overflow interrupts when the TOF bit becomes set. Reset clears the TOIE bit.

1 = TIM overflow interrupts enabled
0 = TIM overflow interrupts disabled

TSTOP — TIM Stop Bit

This read/write bit stops the TIM counter. Counting resumes when TSTOP is cleared. Reset sets the TSTOP bit, stopping the TIM counter until software clears the TSTOP bit.

1 = TIM counter stopped
0 = TIM counter active

NOTE: *Do not set the TSTOP bit before entering wait mode if the TIM is required to exit wait mode.*

TRST — TIM Reset Bit

Setting this write-only bit resets the TIM counter and the TIM prescaler. Setting TRST has no effect on any other registers. Counting resumes from $0000. TRST is cleared automatically after the TIM counter is reset and always reads as logic 0. Reset clears the TRST bit.

1 = Prescaler and TIM counter cleared
0 = No effect

NOTE: *Setting the TSTOP and TRST bits simultaneously stops the TIM counter at a value of $0000.*

PS[2:0] — Prescaler Select Bits

These read/write bits select one of the seven prescaler outputs as the input to the TIM counter as **Table 10-2** shows. Reset clears the PS[2:0] bits.

Table 10-2. Prescaler Selection

PS2	PS1	PS0	TIM Clock Source
0	0	0	Internal bus clock ÷ 1
0	0	1	Internal bus clock ÷ 2
0	1	0	Internal bus clock ÷ 4
0	1	1	Internal bus clock ÷ 8
1	0	0	Internal bus clock ÷ 16
1	0	1	Internal bus clock ÷ 32
1	1	0	Internal bus clock ÷ 64
1	1	1	TCLK

10.10.2 TIM Counter Registers

The two read-only TIM counter registers contain the high and low bytes of the value in the TIM counter. Reading the high byte (TCNTH) latches the contents of the low byte (TCNTL) into a buffer. Subsequent reads of TCNTH do not affect the latched TCNTL value until TCNTL is read. Reset clears the TIM counter registers. Setting the TIM reset bit (TRST) also clears the TIM counter registers.

NOTE: *If you read TCNTH during a break interrupt, be sure to unlatch TCNTL by reading TCNTL before exiting the break interrupt. Otherwise, TCNTL retains the value latched during the break.*

Address: T1CNTH, $000C and T2CNTH, $0042

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

= Unimplemented

Figure 10-5. TIM Counter Registers High (TCNTH)

Address: T1CNTL, $000D and T2CNTL, $0043

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

= Unimplemented

Figure 10-6. TIM Counter Registers Low (TCNTL)

10.10.3 TIM Counter Modulo Registers

The read/write TIM modulo registers contain the modulo value for the TIM counter. When the TIM counter reaches the modulo value, the overflow flag (TOF) becomes set, and the TIM counter resumes counting from $0000 at the next timer clock. Writing to the high byte (TMODH) inhibits the TOF bit and overflow interrupts until the low byte (TMODL) is written. Reset sets the TIM counter modulo registers.

Address: T1MODH, $000E and T2MODH, $0044

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

Figure 10-7. TIM Counter Modulo Register High (TMODH)

Address: T1MODL, $000F and T2MODL, $0045

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

Figure 10-8. TIM Counter Modulo Register Low (TMODL)

NOTE: *Reset the TIM counter before writing to the TIM counter modulo registers.*

10.10.4 TIM Channel Status and Control Registers

Each of the TIM channel status and control registers:

- Flags input captures and output compares
- Enables input capture and output compare interrupts
- Selects input capture, output compare, or PWM operation
- Selects high, low, or toggling output on output compare
- Selects rising edge, falling edge, or any edge as the active input capture trigger
- Selects output toggling on TIM overflow
- Selects 0% and 100% PWM duty cycle
- Selects buffered or unbuffered output compare/PWM operation

Address: T1SC0, $0010 and T2SC0, $0046

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Figure 10-9. TIM Channel 0 Status and Control Register (TSC0)

Address: T1SC1, $0013 and T2SC1, $0049

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH1F	CH1IE	CH01IE	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Figure 10-10. TIM Channel 1 Status and Control Register (TSC1)

CHxF — Channel x Flag Bit

When channel x is an input capture channel, this read/write bit is set when an active edge occurs on the channel x pin. When channel x is an output compare channel, CHxF is set when the value in the TIM counter registers matches the value in the TIM channel x registers.

When TIM CPU interrupt requests are enabled (CHxIE = 1), clear CHxF by reading TIM channel x status and control register with CHxF set and then writing a logic 0 to CHxF. If another interrupt request occurs before the clearing sequence is complete, then writing logic 0 to CHxF has no effect. Therefore, an interrupt request cannot be lost due to inadvertent clearing of CHxF.

Reset clears the CHxF bit. Writing a logic 1 to CHxF has no effect.

1 = Input capture or output compare on channel x
0 = No input capture or output compare on channel x

CHxIE — Channel x Interrupt Enable Bit

This read/write bit enables TIM CPU interrupt service requests on channel x.

Reset clears the CHxIE bit.

1 = Channel x CPU interrupt requests enabled
0 = Channel x CPU interrupt requests disabled

CH01IE — CH0F and CH1F Interrupt Enable Bit

This read/write bit enables TIM CPU interrupt service requests when CH0F and CH1F are set.

Reset clears the CH01IE bit.

1 = CPU interrupt requests when CH0F and CH1F are set
0 = No CPU interrupt requests when CH0F and CH1F are set

MS0B — Mode Select Bit B

This read/write bit selects buffered output compare/PWM operation. MS0B exists only in the TIM1 channel 0 and TIM2 channel 0 status and control registers.

Setting MS0B disables the channel 1 status and control register.

Reset clears the MS0B bit.

1 = Buffered output compare/PWM operation enabled
0 = Buffered output compare/PWM operation disabled

MSxA — Mode Select Bit A

When ELSxB:ELSxA ≠ 0:0, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation. See **Table 10-3**.

1 = Unbuffered output compare/PWM operation
0 = Input capture operation

When ELSxB:ELSxA = 0:0, this read/write bit selects the initial output level of the TCHx pin. See **Table 10-3**. Reset clears the MSxA bit.

1 = Initial output level low
0 = Initial output level high

NOTE: *Before changing a channel function by writing to the MSxB or MSxA bit, set the TSTOP and TRST bits in the TIM status and control register.*

ELSxB and ELSxA — Edge/Level Select Bits

When channel x is an input capture channel, these read/write bits control the active edge-sensing logic on channel x.

When channel x is an output compare channel, ELSxB and ELSxA control the channel x output behavior when an output compare occurs.

When ELSxB and ELSxA are both clear, channel x is not connected to an I/O port, and pin TCHx is available as a general-purpose I/O pin. **Table 10-3** shows how ELSxB and ELSxA work. Reset clears the ELSxB and ELSxA bits.

Table 10-3. Mode, Edge, and Level Selection

MS0B:MSxA	ELSxB:ELSxA	Mode	Configuration
X0	00	Output preset	Pin under port control; initial output level high
X1	00		Pin under port control; initial output level low
00	01	Input capture	Capture on rising edge only
00	10		Capture on falling edge only
00	11		Capture on rising or falling edge
01	01	Output compare or PWM(1)	Toggle output on compare
01	10		Clear output on compare
01	11		Set output on compare
1X	01	Buffered output compare or buffered PWM	Toggle output on compare
1X	10		Clear output on compare
1X	11		Set output on compare

Notes:
1. Enable only one channel for unbuffered output compare or PWM functions. Avoid the following configuration: MS0B = 0, MS0A = 1, MS1A = 1, and ELSxB:A ≠ 00

NOTE: *Before enabling a TIM channel register for input capture operation, make sure that the TCHx pin is stable for at least two bus clocks.*

TOVx — Toggle On Overflow Bit

When channel x is an output compare channel, this read/write bit controls the behavior of the channel x output when the TIM counter overflows. When channel x is an input capture channel, TOVx has no effect.

Reset clears the TOVx bit.

1 = Channel x pin toggles on TIM counter overflow

0 = Channel x pin does not toggle on TIM counter overflow

NOTE: *When TOVx is set, a TIM counter overflow takes precedence over a channel x output compare if both occur at the same time.*

CHxMAX — Channel x Maximum Duty Cycle Bit

When the TOVx bit is at logic 1, setting the CHxMAX bit forces the duty cycle of buffered and unbuffered PWM signals to 100%. As **Figure 10-11** shows, the CHxMAX bit takes effect in the cycle after it is set or cleared. The output stays at the 100% duty cycle level until the cycle after CHxMAX is cleared.

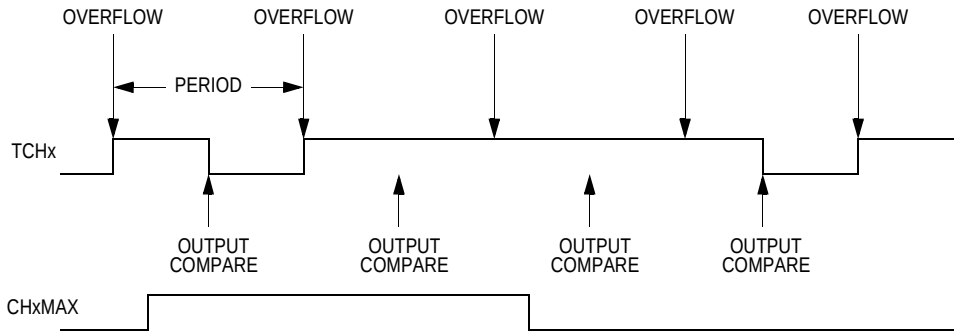


Figure 10-11. CHxMAX Latency

10.10.5 TIM Channel Registers

These read/write registers contain the captured TIM counter value of the input capture function or the output compare value of the output compare function. The state of the TIM channel registers after reset is unknown.

In input capture mode (MSxB:MSxA = 0:0), reading the high byte of the TIM channel x registers (TCHxH) inhibits input captures until the low byte (TCHxL) is read.

In output compare mode (MSxB:MSxA ≠ 0:0), writing to the high byte of the TIM channel x registers (TCHxH) inhibits output compares until the low byte (TCHxL) is written.

Address: T1CH0H, $0011 and T2CH0H, $0047

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

Figure 10-12. TIM Channel 0 Register High (TCH0H)

Address: T1CH0L, $0012 and T2CH0L $0048

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

Figure 10-13. TIM Channel 0 Register Low (TCH0L)

Address: T1CH1H, $0014 and T2CH1H, $004A

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

Figure 10-14. TIM Channel 1 Register High (TCH1H)

Address: T1CH1L, $0015 and T2CH1L, $004B

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

Figure 10-15. TIM Channel 1 Register Low (TCH1L)

Section 11. Universal Serial Bus Module (USB)

11.1 Contents

11.2 Introduction

This section describes the universal serial bus (USB) module. The USB module is designed to serve as a low-speed (LS) USB device per the *Universal Serial Bus Specification Rev. 2.0.* Control and interrupt data transfers are supported. Endpoint 0 functions as a transmit/receive control endpoint; endpoint 1 functions as interrupt transmit endpoint; endpoint 2 functions as interrupt transmit or receive endpoint.

11.3 Features

Features of the USB module include:

- Universal Serial Bus Specification 2.0 low-speed functions
- 1.5 Mbps data rate
- On-chip 3.3V regulator
- Endpoint 0 with 8-byte transmit buffer and 8-byte receive buffer
- Endpoint 1 with 8-byte transmit buffer
- Endpoint 2 with 8-byte transmit buffer and 8-byte receive buffer
- USB data control logic:
  - Control endpoint 0 and interrupt endpoints 1 and 2
  - Packet decoding/generation
  - CRC generation and checking
  - NRZI (Non-Return-to Zero Inserted) encoding/decoding
  - Bit-stuffing
- USB reset options:
  - Internal MCU reset generation
  - CPU interrupt request generation
- Suspend and resume operations, with remote wakeup support
- USB-generated interrupts:
  - Transaction interrupt driven
  - Resume interrupt
  - End-of-packet interrupt
  - USB reset
- STALL, NAK, and ACK handshake generation

11.4 Pin Name Conventions

The USB share two I/O pins with two port E I/O pins. The full name of the USB I/O pin is listed in **Table 11-1**. The generic pin name appear in the text that follows.

Table 11-1. USB Module Pin Name Conventions

USB Generic Pin Names:	D+	D–
Full USB Pin Names:	PTE3/D+	PTE4/D–

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0018	USB Interrupt Register 2 (UIR2)	Read:	0	0	0	0	0	0	0	0
		Write:	EOPFR	RSTFR	TXD2FR	RXD2FR	TXD1FR	RESUMFR	TXD0FR	RXD0FR
		Reset:	0	0	0	0	0	0	0	0
$0019	USB Control Register 2 (UCR2)	Read:	T2SEQ	STALL2	TX2E	RX2E	TP2SIZ3	TP2SIZ2	TP2SIZ1	TP2SIZ0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001A	USB Control Register 3 (UCR3)	Read:	TX1ST	0	OSTALL0	ISTALL0	0	PULLEN	ENABLE2	ENABLE1
		Write:		TX1STR						
		Reset:	0	0	0	0	0	0*	0	0
$001B	USB Control Register 4 (UCR4)	Read:	0	0	0	0	0	FUSBO	FDP	FDM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0020	USB Endpoint 0 Data Register 0 (UE0D0)	Read:	UE0R07	UE0R06	UE0R05	UE0R04	UE0R03	UE0R02	UE0R01	UE0R00
		Write:	UE0T07	UE0T06	UE0T05	UE0T04	UE0T03	UE0T02	UE0T01	UE0T00
		Reset:	Unaffected by reset							
$0021	USB Endpoint 0 Data Register 1 (UE0D1)	Read:	UE0R17	UE0R16	UE0R15	UE0R14	UE0R13	UE0R12	UE0R11	UE0R10
		Write:	UE0T17	UE0T16	UE0T15	UE0T14	UE0T13	UE0T12	UE0T11	UE0T10
		Reset:	Unaffected by reset							

* PULLEN bit is reset by POR or LVI reset only.

= Unimplemented U = Unaffected by reset

Figure 11-1. USB I/O Register Summary (Sheet 1 of 4)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0022	USB Endpoint 0 Data Register 2 (UE0D2)	Read:	UE0R27	UE0R26	UE0R25	UE0R24	UE0R23	UE0R22	UE0R21	UE0R20
		Write:	UE0T27	UE0T26	UE0T25	UE0T24	UE0T23	UE0T22	UE0T21	UE0T20
		Reset:	Unaffected by reset							
$0023	USB Endpoint 0 Data Register 3 (UE0D3)	Read:	UE0R37	UE0R36	UE0R35	UE0R34	UE0R33	UE0R32	UE0R31	UE0R30
		Write:	UE0T37	UE0T36	UE0T35	UE0T34	UE0T33	UE0T32	UE0T31	UE0T30
		Reset:	Unaffected by reset							
$0024	USB Endpoint 0 Data Register 4 (UE0D4)	Read:	UE0R47	UE0R46	UE0R45	UE0R44	UE0R43	UE0R42	UE0R41	UE0R40
		Write:	UE0T47	UE0T46	UE0T45	UE0T44	UE0T43	UE0T42	UE0T41	UE0T40
		Reset:	Unaffected by reset							
$0025	USB Endpoint 0 Data Register 5 (UE0D5)	Read:	UE0R57	UE0R56	UE0R55	UE0R54	UE0R53	UE0R52	UE0R51	UE0R50
		Write:	UE0T57	UE0T56	UE0T55	UE0T54	UE0T53	UE0T52	UE0T51	UE0T50
		Reset:	Unaffected by reset							
$0026	USB Endpoint 0 Data Register 6 (UE0D6)	Read:	UE0R67	UE0R66	UE0R65	UE0R64	UE0R63	UE0R62	UE0R61	UE0R60
		Write:	UE0T67	UE0T66	UE0T65	UE0T64	UE0T63	UE0T62	UE0T61	UE0T60
		Reset:	Unaffected by reset							
$0027	USB Endpoint 0 Data Register 7 (UE0D7)	Read:	UE0R77	UE0R76	UE0R75	UE0R74	UE0R73	UE0R72	UE0R71	UE0R70
		Write:	UE0T77	UE0T76	UE0T75	UE0T74	UE0T73	UE0T72	UE0T71	UE0T70
		Reset:	Unaffected by reset							
$0028	USB Endpoint 1 Data Register 0 (UE1D0)	Read:								
		Write:	UE1T07	UE1T06	UE1T05	UE1T04	UE1T03	UE1T02	UE1T01	UE1T00
		Reset:	Unaffected by reset							
$0029	USB Endpoint 1 Data Register 1 (UE1D1)	Read:								
		Write:	UE1T17	UE1T16	UE1T15	UE1T14	UE1T13	UE1T12	UE1T11	UE1T10
		Reset:	Unaffected by reset							
$002A	USB Endpoint 1 Data Register 2 (UE1D2)	Read:								
		Write:	UE1T27	UE1T26	UE1T25	UE1T24	UE1T23	UE1T22	UE1T21	UE1T20
		Reset:	Unaffected by reset							
$002B	USB Endpoint 1 Data Register 3 (UE1D3)	Read:								
		Write:	UE1T37	UE1T36	UE1T35	UE1T34	UE1T33	UE1T32	UE1T31	UE1T30
		Reset:	Unaffected by reset							

= Unimplemented U = Unaffected by reset

Figure 11-1. USB I/O Register Summary (Sheet 2 of 4)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$002C	USB Endpoint 1 Data Register 4 (UE1D4)	Read:								
		Write:	UE1T47	UE1T46	UE1T45	UE1T44	UE1T43	UE1T42	UE1T41	UE1T40
		Reset:	Unaffected by reset							
$002D	USB Endpoint 1 Data Register5 (UE1D5)	Read:								
		Write:	UE1T57	UE1T56	UE1T55	UE1T54	UE1T53	UE1T52	UE1T51	UE1T50
		Reset:	Unaffected by reset							
$002E	USB Endpoint 1 Data Register 6 (UE1D6)	Read:								
		Write:	UE1T67	UE1T66	UE1T65	UE1T64	UE1T63	UE1T62	UE1T61	UE1T60
		Reset:	Unaffected by reset							
$002F	USB Endpoint 1 Data Register 7 (UE1D7)	Read:								
		Write:	UE1T77	UE1T76	UE1T75	UE1T74	UE1T73	UE1T72	UE1T71	UE1T70
		Reset:	Unaffected by reset							
$0030	USB Endpoint 2 Data Register 0 (UE2D0)	Read:	UE2R07	UE2R06	UE2R05	UE2R04	UE2R03	UE2R02	UE2R01	UE2R00
		Write:	UE2T07	UE2T06	UE2T05	UE2T04	UE2T03	UE2T02	UE2T01	UE2T00
		Reset:	Unaffected by reset							
$0031	USB Endpoint 2 Data Register 1 (UE2D1)	Read:	UE2R17	UE2R16	UE2R15	UE2R14	UE2R13	UE2R12	UE2R11	UE2R10
		Write:	UE2T17	UE2T16	UE2T15	UE2T14	UE2T13	UE2T12	UE2T11	UE2T10
		Reset:	Unaffected by reset							
$0032	USB Endpoint 2 Data Register 2 (UE2D2)	Read:	UE2R27	UE2R26	UE2R25	UE2R24	UE2R23	UE2R22	UE2R21	UE2R20
		Write:	UE2T27	UE2T26	UE2T25	UE2T24	UE2T23	UE2T22	UE2T21	UE2T20
		Reset:	Unaffected by reset							
$0033	USB Endpoint 2 Data Register 3 (UE2D3)	Read:	UE2R37	UE2R36	UE2R35	UE2R34	UE2R33	UE2R32	UE2R31	UE2R30
		Write:	UE2T37	UE2T36	UE2T35	UE2T34	UE2T33	UE2T32	UE2T31	UE2T30
		Reset:	Unaffected by reset							
$0034	USB Endpoint 2 Data Register 4 (UE2D4)	Read:	UE2R47	UE2R46	UE2R45	UE2R44	UE2R43	UE2R42	UE2R41	UE2R40
		Write:	UE2T47	UE2T46	UE2T45	UE2T44	UE2T43	UE2T42	UE2T41	UE2T40
		Reset:	Unaffected by reset							
$0035	USB Endpoint 2 Data Register 5 (UE2D5)	Read:	UE2R57	UE2R56	UE2R55	UE2R54	UE2R53	UE2R52	UE2R51	UE2R50
		Write:	UE2T57	UE2T56	UE2T55	UE2T54	UE2T53	UE2T52	UE2T51	UE2T50
		Reset:	Unaffected by reset							

= Unimplemented U = Unaffected by reset

Figure 11-1. USB I/O Register Summary (Sheet 3 of 4)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0036	USB Endpoint 2 Data Register 6 (UE2D6)	Read:	UE2R67	UE2R66	UE2R65	UE2R64	UE2R63	UE2R62	UE2R61	UE2R60
		Write:	UE2T67	UE2T66	UE2T65	UE2T64	UE2T63	UE2T62	UE2T61	UE2T60
		Reset:	Unaffected by reset							
$0037	USB Endpoint 2 Data Register 7 (UE2D7)	Read:	UE2R77	UE2R76	UE2R75	UE2R74	UE2R73	UE2R72	UE2R71	UE2R70
		Write:	UE2T77	UE2T76	UE2T75	UE2T74	UE2T73	UE2T72	UE2T71	UE2T70
		Reset:	Unaffected by reset							
$0038	USB Address Register (UADDR)	Read:	USBEN	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0
		Write:								
		Reset:	0*	0	0	0	0	0	0	0
$0039	USB Interrupt Register 0 (UIR0)	Read:	EOPIE	SUSPND	TXD2IE	RXD2IE	TXD1IE	0	TXD0IE	RXD0IE
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$003A	USB Interrupt Register 1 (UIR1)	Read:	EOPF	RSTF	TXD2F	RXD2F	TXD1F	RESUMF	TXD0F	RXD0F
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$003B	USB Control Register 0 (UCR0)	Read:	T0SEQ	0	TX0E	RX0E	TP0SIZ3	TP0SIZ2	TP0SIZ1	TP0SIZ0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$003C	USB Control Register 1 (UCR1)	Read:	T1SEQ	STALL1	TX1E	FRESUM	TP1SIZ3	TP1SIZ2	TP1SIZ1	TP1SIZ0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$003D	USB Status Register 0 (USR0)	Read:	R0SEQ	SETUP	0	0	RP0SIZ3	RP0SIZ2	RP0SIZ1	RP0SIZ0
		Write:								
		Reset:	Unaffected by reset							
$003E	USB Status Register 1 (USR1)	Read:	R2SEQ	TXACK	TXNAK	TXSTL	RP2SIZ3	RP2SIZ2	RP2SIZ1	RP2SIZ0
		Write:								
		Reset:	U	0	0	0	U	U	U	U

* USBEN bit is reset by POR or LVI reset only.

= Unimplemented U = Unaffected by reset

Figure 11-1. USB I/O Register Summary (Sheet 4 of 4)

11.5 Functional Description

Figure 11-2 shows the block diagram of the USB module. The USB module manages communications between the host and the USB function. The module is partitioned into three functional blocks. These blocks consist of a dual-function transceiver, the USB control logic, and the endpoint registers. The blocks are further detailed later in this section (see 11.7 Hardware Description).

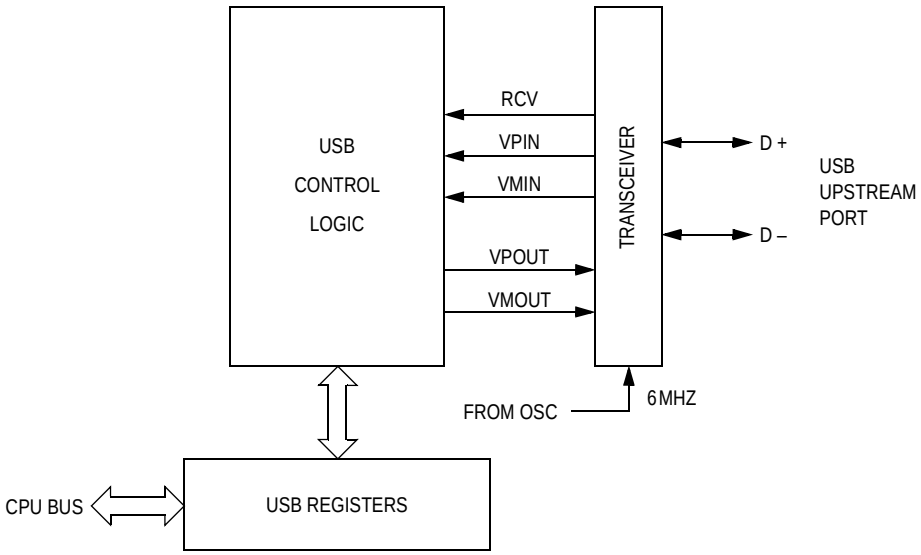


Figure 11-2. USB Block Diagram

11.5.1 USB Protocol

Figure 11-3 shows the various transaction types supported by the USB module. The transactions are portrayed as error free. The effect of errors in the data flow are discussed later.

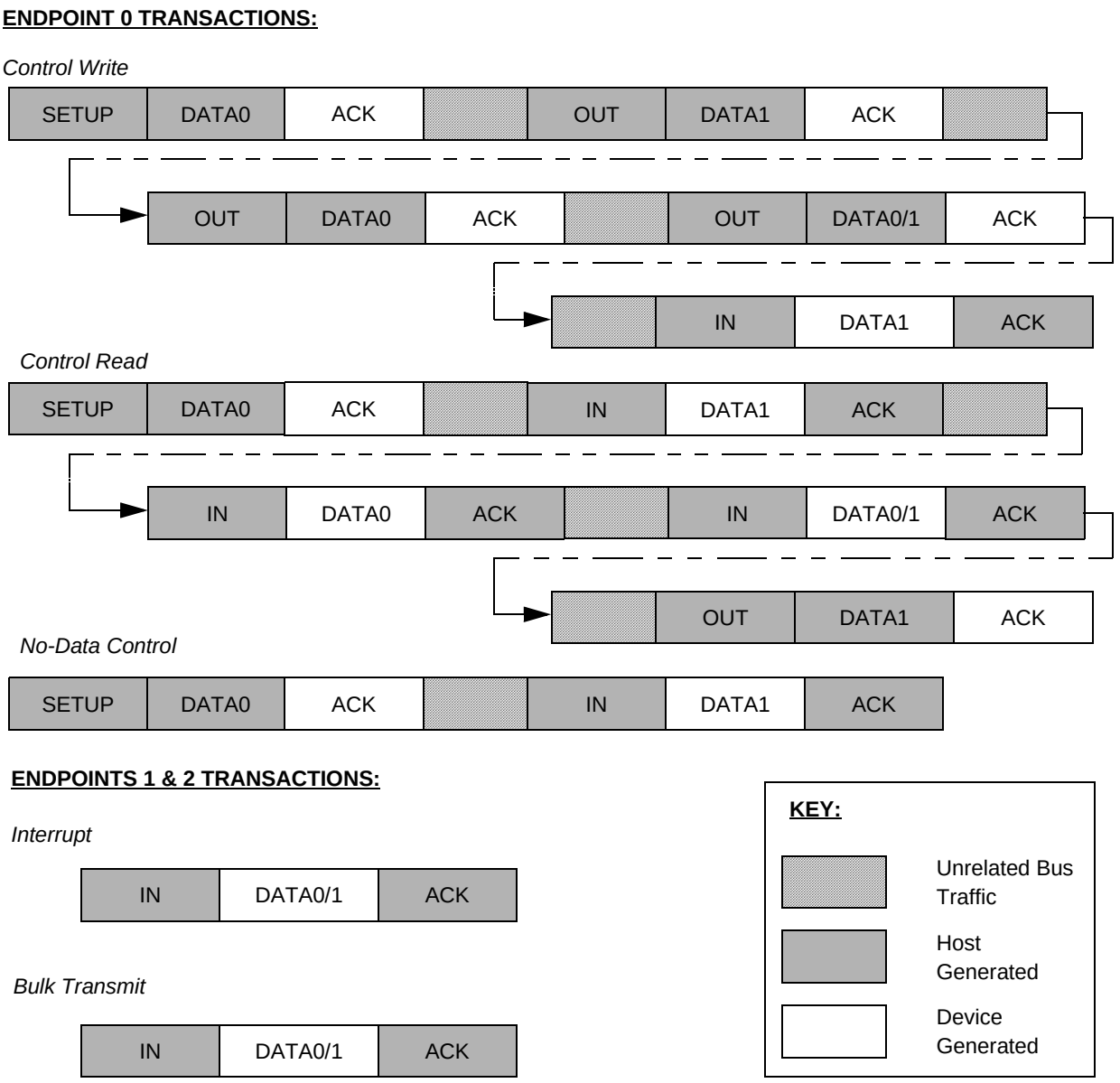


Figure 11-3. Supported Transaction Types Per Endpoint

Each USB transaction is comprised of a series of packets. The USB module supports the packet types shown in **Figure 11-4**. Token packets are generated by the USB host and decoded by the USB device. Data and handshake packets are both decoded and generated by the USB device, depending on the type of transaction.

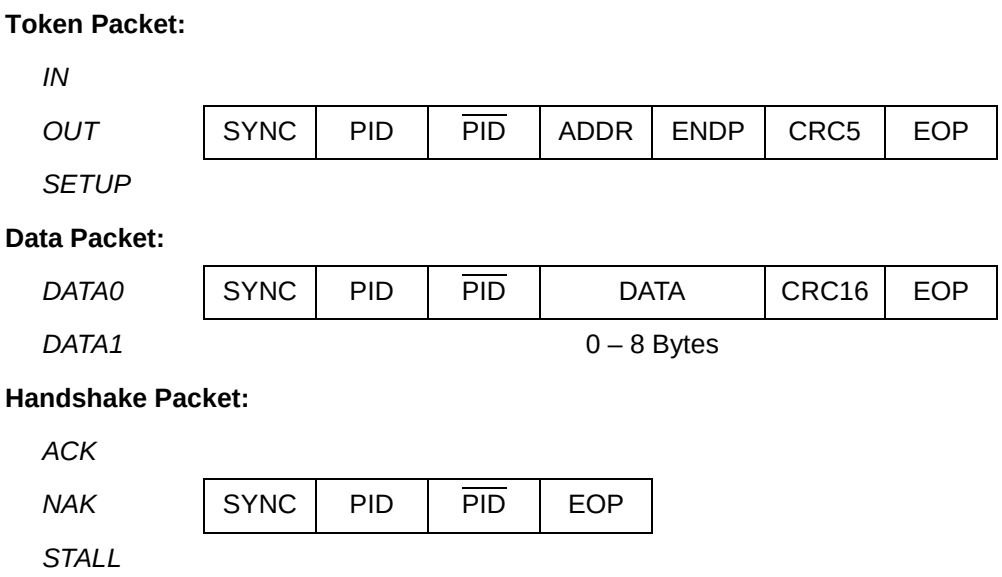


Figure 11-4. Supported USB Packet Types

The following sections detail each segment used to form a complete USB transaction.

11.5.1.1 Sync Pattern

The NRZI bit pattern shown in **Figure 11-5** is used as a synchronization pattern and is prefixed to each packet. This pattern is equivalent to a data pattern of seven 0s followed by a 1 ($80).

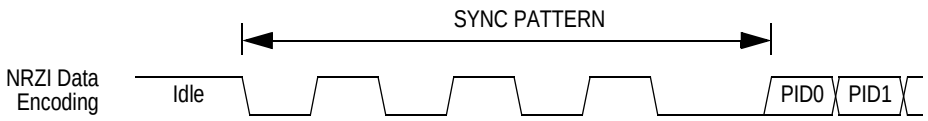


Figure 11-5. Sync Pattern

The start of a packet (SOP) is signaled by the originating port by driving the D+ and D– lines from the idle state (also referred to as the J state) to the opposite logic level (also referred to as the K state). This switch in levels represents the first bit of the sync field. **Figure 11-6** shows the data signaling and voltage levels for the start of packet and the sync pattern.

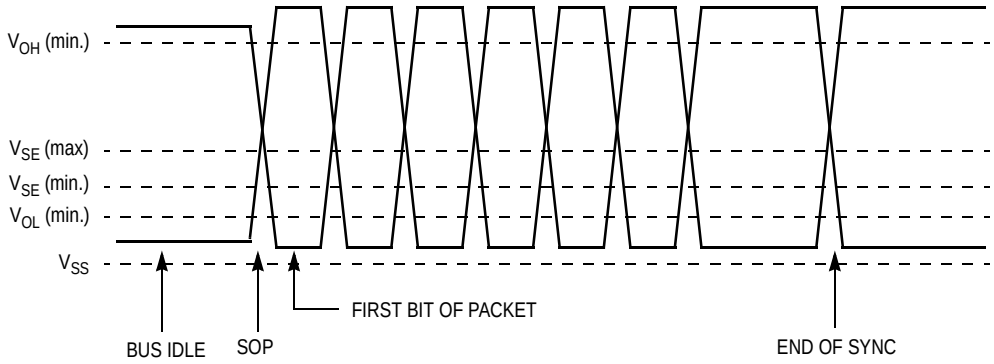


Figure 11-6. SOP, Sync Signaling, and Voltage Levels

11.5.1.2 Packet Identifier Field

The packet identifier field is an 8-bit number comprised of the 4-bit packet identification and its complement. The field follows the sync pattern and determines the direction and type of transaction on the bus. **Table 11-2** shows the packet identifier values for the supported packet types.

Table 11-2. Supported Packet Identifiers

Packet Identifier Value	Packet Identifier Type
%1001	IN Token
%0001	OUT Token
%1101	SETUP Token
%0011	DATA0 Packet
%1011	DATA1 Packet
%0010	ACK Handshake
%1010	NAK Handshake
%1110	STALL Handshake

11.5.1.3 Address Field (ADDR)

The address field is a 7-bit number that is used to select a particular USB device. This field is compared to the lower seven bits of the UADDR register to determine if a given transaction is targeting the MCU USB device.

11.5.1.4 Endpoint Field (ENDP)

The endpoint field is a 4-bit number that is used to select a particular endpoint within a USB device. For the MCU, this will be a binary number between 0 and 2 inclusive. Any other value will cause the transaction to be ignored.

11.5.1.5 Cyclic Redundancy Check (CRC)

Cyclic redundancy checks are used to verify the address and data stream of a USB transaction. This field is five bits wide for token packets and 16 bits wide for data packets. CRCs are generated in the transmitter and sent on the USB data lines after both the endpoint field and the data field.

11.5.1.6 End-of-Packet (EOP)

The single-ended 0 (SE0) state is used to signal an end-of-packet (EOP). The single-ended 0 state is indicated by both D+ and D– being below 0.8V. EOP will be signaled by driving D+ and D– to the single-ended 0 state for two bit times followed by driving the lines to the idle state for one bit time. The transition from the single-ended 0 to the idle state defines the end of the packet. The idle state is asserted for one bit time and then both the D+ and D– output drivers are placed in their high-impedance state. The bus termination resistors hold the bus in the idle state. **Figure 11-7** shows the data signaling and voltage levels for an end-of-packet transaction.

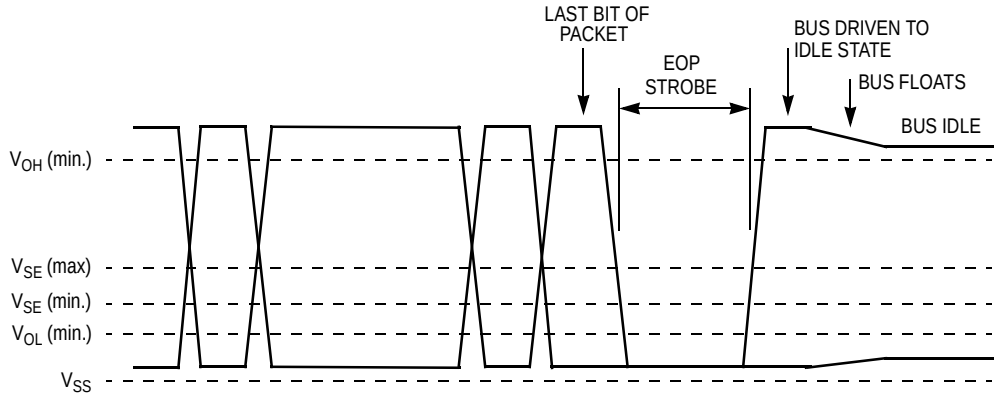


Figure 11-7. EOP Transaction Voltage Levels

The width of the SE0 in the EOP is about two bit times. The EOP width is measured with the same capacitive load used for maximum rise and fall times and is measured at the same level as the differential signal crossover points of the data lines.

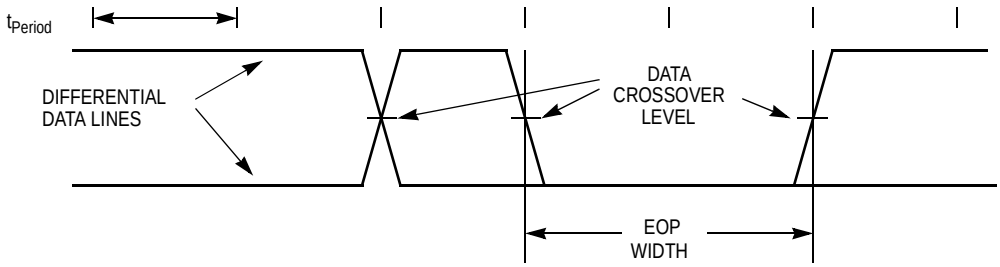


Figure 11-8. EOP Width Timing

11.5.2 Reset Signaling

The USB module will detect a reset signaled on the bus by the presence of an extended SE0 at the USB data pins of a device. The MCU seeing a single-ended 0 on its USB data inputs for more than 8µs treats that signal as a reset.

A USB sourced reset will hold the MCU in reset for the duration of the reset on the USB bus. The USB bit in the reset status register (SRSR) will be set after the internal reset is removed. Refer to **8.8.2 SIM Reset Status Register (SRSR)** for more detail. The MCU's reset recovery sequence is detailed in **Section 8. System Integration Module (SIM)**.

The reset flag bit (RSTF) in the USB interrupt register 1 (UIR1) also will be set after the internal reset is removed. Refer to **11.8.3 USB Interrupt Register 1** for more detail.

After a reset is removed, the device will be in the default, but not yet addressed or configured state (refer to Section 9.1 USB Device States of the *Universal Serial Bus Specification Rev. 2.0*). The device must be able to accept a device address via a SET_ADDRESS command (refer to Section 9.4 Standard Device Request in the *Universal Serial Bus Specification Rev. 2.0*) no later than 10ms after the reset is removed.

Reset can wake a device from the suspended mode.

NOTE: *USB reset can be configured not to generate a reset signal to the CPU by setting the URSTD bit of the configuration register (see **Section 5. Configuration Register (CONFIG)**). When a USB reset is detected, the CPU generates an USB interrupt.*

11.5.3 Suspend

The MCU supports suspend mode for low power. Suspend mode should be entered when the USB data lines are in the idle state for more than 3ms. Entry into suspend mode is controlled by the SUSPND bit in the USB interrupt register. Any low-speed bus activity should keep the device out of the suspend state. Low-speed devices are kept awake by periodic low-speed EOP signals from the host. This is referred to as low speed keep alive (refer to Section 11.8.4.1 Low-speed Keep-alive in the *Universal Serial Bus Specification Rev. 2.0*).

Firmware should monitor the EOPF flag and enter suspend mode by setting the SUSPND bit if an EOP is not detected for 3ms.

Per the USB specification, the bus powered USB system is required to draw less than 500µA from the V_{DD} supply when in the suspend state. This includes the current supplied by the voltage regulator to the 1.5kΩ to ground termination resistors placed at the host end of the USB bus. This low-current requirement means that firmware is responsible for entering stop mode once the USB module has been placed in the suspend state.

11.5.4 Resume After Suspend

The MCU can be activated from the suspend state by normal bus activity, a USB reset signal, or by a forced resume driven from the MCU.

11.5.4.1 Host Initiated Resume

The host signals resume by initiating resume signalling (K state) for at least 20ms followed by a standard low-speed EOP signal. This 20ms ensures that all devices in the USB network are awakened.

After resuming the bus, the host must begin sending bus traffic within 3ms to prevent the device from re-entering suspend mode.

11.5.4.2 USB Reset Signalling

Reset can wake a device from the suspended mode.

11.5.4.3 Remote Wakeup

The MCU also supports the remote wakeup feature. The firmware has the ability to exit suspend mode by signaling a resume state to the upstream host or hub. A non-idle state (K state) on the USB data lines is accomplished by asserting the FRESUM bit in the UCR1 register.

When using the remote wakeup capability, the firmware must wait for at least 5ms after the bus is in the idle state before sending the remote wakeup resume signaling. This allows the upstream devices to get into their suspend state and prepare for propagating resume signaling. The FRESUM bit should be asserted to cause the resume state on the USB data lines for at least 10ms, but not more than 15ms. Note that the resume signalling is controlled by the FRESUM bit and meeting the timing specifications is dependent on the firmware. When FRESUM is cleared by firmware, the data lines will return to their high-impedance state.

Refer to register definitions (see **11.8.6 USB Control Register 1**) for more information about how the force resume (FRESUM) bit can be used to initiate the remote wakeup feature.

11.5.5 Low-Speed Device

Low-speed devices are configured by the position of a pull-up resistor on the USB D– pin of the MCU. Low-speed devices are terminated as shown in **Figure 11-9** with the pull-up on the D– line.

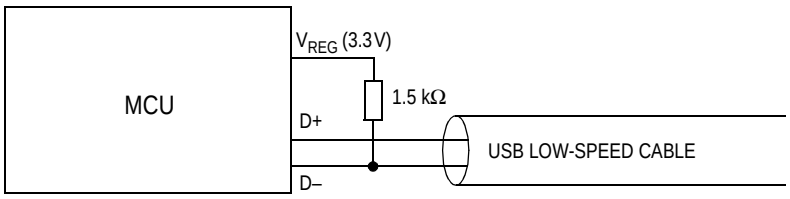


Figure 11-9. External Low-Speed Device Configuration

For low-speed transmissions, the transmitter's EOP width must be between 1.25µs and 1.50µs. These ranges include timing variations due to differential buffer delay and rise/fall time mismatches and to noise and other random effects. A low-speed receiver must accept a 670ns SE0 followed by a J transition as a valid EOP. An SE0 shorter than 330ns or an SE0 not followed by a J transition are rejected as an EOP. Any SE0 that is 8µs or longer is automatically a reset.

11.6 Clock Requirements

The low-speed data rate is nominally 1.5 Mbps. The OSCXCLK÷2 (6MHz) signal driven by the oscillator circuits is the clock source for the USB module and requires that a 12MHz oscillator circuit be connected to the OSC1 and OSC2 pins. The permitted frequency tolerance for low-speed functions is approximately ±1.5% (15,000 ppm). This tolerance includes inaccuracies from all sources: initial frequency accuracy, crystal capacitive loading, supply voltage on the oscillator, temperature, and aging. The jitter in the low-speed data rate must be less than 10ns.

11.7 Hardware Description

The USB module as previously shown in **Figure 11-2** contains three functional blocks: the low-speed USB transceiver, the USB control logic, and the USB registers. The following details the function of the regulator, transceiver, and control logic. See **11.8 I/O Registers** for details of register settings.

11.7.1 Voltage Regulator

The USB data lines are required by the USB specification to have an output voltage between 2.8V and 3.6V. The data lines also are required to have an external 1.5kΩ pull-up resistor connected between a data line and a voltage source between 3.0V and 3.6V. **Figure 11-10** shows the worst case electrical connection for the voltage regulator.

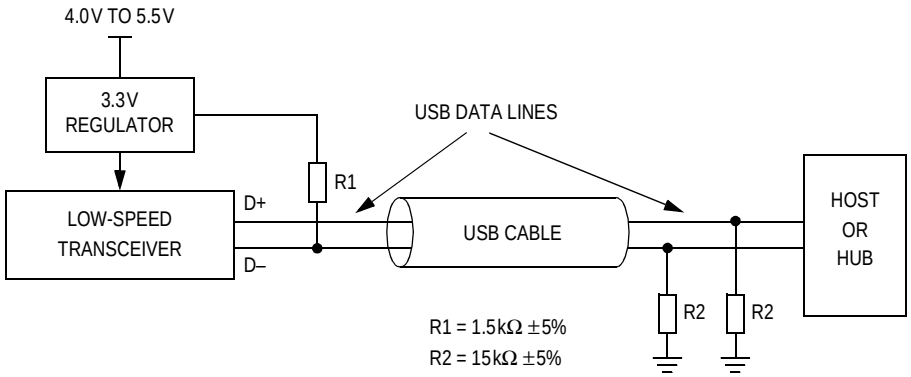


Figure 11-10. Regulator Electrical Connections

11.7.2 USB Transceiver

The USB transceiver provides the physical interface to the USB D+ and D– data lines. The transceiver is composed of two parts: an output drive circuit and a receiver.

11.7.2.1 Output Driver Characteristics

The USB transceiver uses a differential output driver to drive the USB data signal onto the USB cable. The static output swing of the driver in its low state is below the V_{OL} of 0.3V with a 1.5kΩ load to 3.6V and in its high state is above the V_{OH} of 2.8V with a 15kΩ load to ground. The output swings between the differential high and low state are well balanced to minimize signal skew. Slew rate control on the driver is used to minimize the radiated noise and cross talk. The driver's outputs support 3-state operation to achieve bidirectional half duplex operation. The driver can tolerate a voltage on the signal pins of –1.0V to 5.5V with respect to local ground reference without damage.

11.7.2.2 Low Speed (1.5 Mbps) Driver Characteristics

The rise and fall time of the signals on this cable are greater than 75ns and less than 300ns. The edges are matched to within ±20% to minimize RFI emissions and signal skew.

USB data transmission is done with differential signals. A differential input receiver is used to accept the USB data signal. A differential 1 on the bus is represented by D+ being at least 200mV more positive than D– as seen at the receiver, and a differential 0 is represented by D– being at least 200mV more positive than D+ as seen at the receiver. The signal cross over point must be between 1.3V and 2.0V.

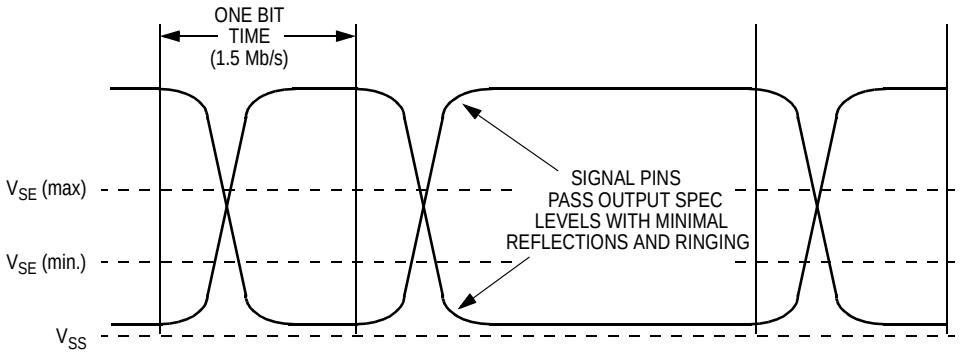


Figure 11-11. Receiver Characteristics

The receiver features an input sensitivity of 200mV when both differential data inputs are in the differential common mode range of 0.8V to 2.5V as shown in **Figure 11-12**. In addition to the differential receiver, there is a single-ended receiver (schmitt trigger) for each of the two data lines.

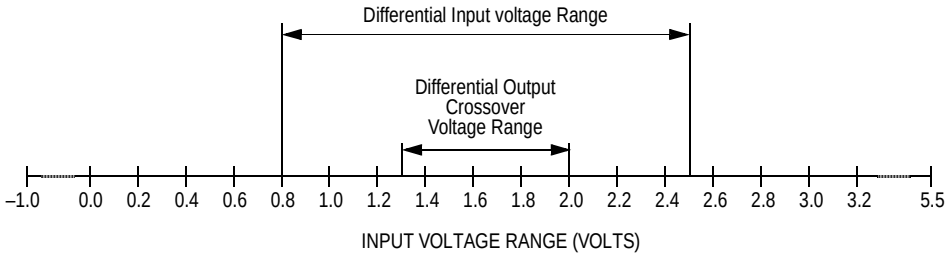


Figure 11-12. Differential Input Sensitivity Range

11.7.2.3 Receiver Data Jitter

The data receivers for all types of devices must be able to properly decode the differential data in the presence of jitter. The more of the bit time that any data edge can occupy and still be decoded, the more reliable the data transfer will be. Data receivers are required to decode differential data transitions that occur in a window plus and minus a nominal quarter bit time from the nominal (centered) data edge position.

Jitter will be caused by the delay mismatches and by mismatches in the source and destination data rates (frequencies). The receive data jitter budget for low speed is given in **Section 20. Electrical Specifications**. The specification includes the consecutive (next) and paired transition values for each source of jitter.

11.7.2.4 Data Source Jitter

The source of data can have some variation (jitter) in the timing of edges of the data transmitted. The time between any set of data transitions is $N \times T_{Period} \pm$ jitter time, where N is the number of bits between the transitions and T_{Period} is defined as the actual period of the data rate. The data jitter is measured with the same capacitive load used for maximum rise and fall times and is measured at the crossover points of the data lines as shown in **Figure 11-13**.

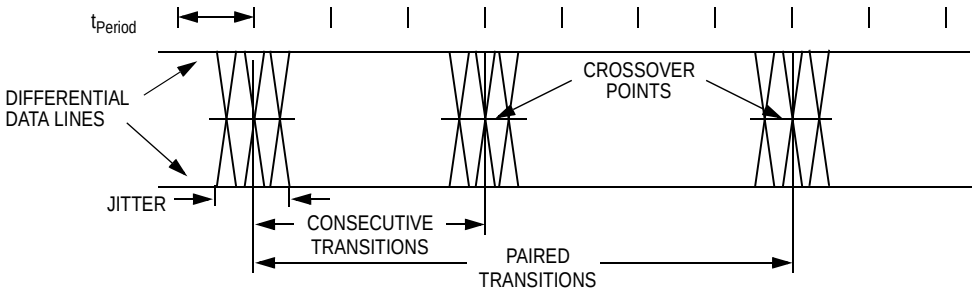


Figure 11-13. Data Jitter

For low-speed transmissions, the jitter time for any consecutive differential data transitions must be within ±25ns and within ±10ns for any set of paired differential data transitions. These jitter numbers include timing variations due to differential buffer delay, rise/fall time mismatches, internal clock source jitter, noise and other random effects.

11.7.2.5 Data Signal Rise and Fall Time

The output rise time and fall time are measured between 10% and 90% of the signal. Edge transition time for the rising and falling edges of low-speed signals is 75ns (minimum) into a capacitive load (C_L) of 200pF and 300ns (maximum) into a capacitive load of 600pF. The rising and falling edges should be transitioning (monotonic) smoothly when driving the cable to avoid excessive EMI.

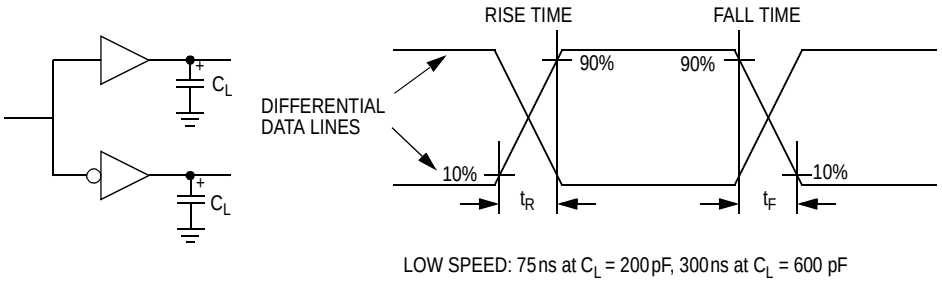


Figure 11-14. Data Signal Rise and Fall Time

11.7.3 USB Control Logic

The USB control logic manages data movement between the CPU and the transceiver. The control logic handles both transmit and receive operations on the USB. It contains the logic used to manipulate the transceiver and the endpoint registers.

The byte count buffer is loaded with the active transmit endpoints byte count value during transmit operations. This same buffer is used for receive transactions to count the number of bytes received and, upon the end of the transaction, transfer that number to the receive endpoints byte count register.

When transmitting, the control logic handles parallel-to-serial conversion, CRC generation, NRZI encoding, and bit stuffing.

When receiving, the control logic handles sync detection, packet identification, end-of-packet detection, bit (un)stuffing, NRZI decoding, CRC validation, and serial-to-parallel conversion. Errors detected by the control logic include bad CRC, timeout while waiting for EOP, and bit stuffing violations.

11.8 I/O Registers

These I/O registers control and monitor USB operation:

- USB address register (UADDR)
- USB control registers 0–4 (UCR0–UCR4)
- USB status registers 0–1 (USR0–USR1)
- USB interrupt registers 0–2 (UIR0–UIR2)
- USB endpoint 0 data registers 0–7 (UE0D0–UE0D7)
- USB endpoint 1 data registers 0–7 (UE1D0–UE1D7)
- USB endpoint 2 data registers 0–7 (UE2D0–UE2D7)

11.8.1 USB Address Register

Address: $0038

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	USBEN	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0
Reset:	0*	0	0	0	0	0	0	0

* USBEN bit is reset by POR or LVI reset only.

Figure 11-15. USB Address Register (UADDR)

USBEN — USB Module Enable

This read/write bit enables and disables the USB module and the USB pins. When USBEN is set, the USB module is enabled and the PTE4 interrupt is disabled. When USBEN is clear, the USB module will not respond to any tokens, USB reset and USB related interrupts are disabled, and pins PTE4/D– and PTE3/D+ function as high current open-drain I/O port pins PTE4 and PTE3.

1 = USB function enabled and PTE4 interrupt is disabled
0 = USB function disabled including USB interrupt, reset and reset interrupt

UADD[6:0] — USB Function Address

These bits specify the USB address of the device. Reset clears these bits.

11.8.2 USB Interrupt Register 0

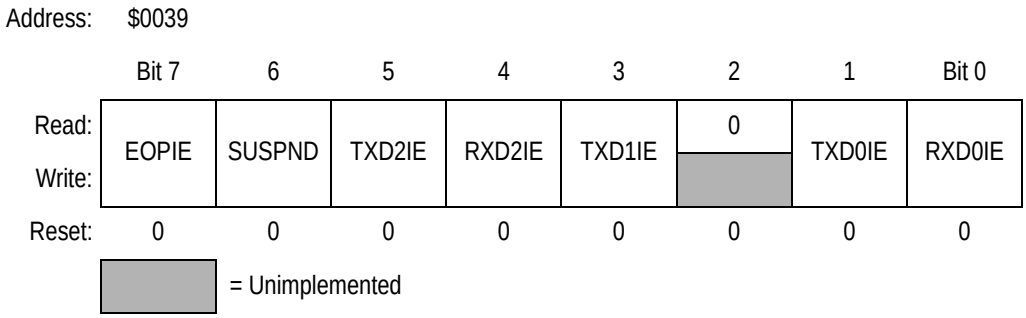


Figure 11-16. USB Interrupt Register 0 (UIR0)

EOPIE — End-of-Packet Detect Interrupt Enable

This read/write bit enables the USB to generate CPU interrupt requests when the EOPF bit becomes set. Reset clears the EOPIE bit.

1 = End-of-packet sequence detection can generate a CPU interrupt request
0 = End-of-packet sequence detection cannot generate a CPU interrupt request

SUSPND — USB Suspend Bit

To save power, this read/write bit should be set by the software if a 3 ms constant idle state is detected on the USB bus. Setting this bit puts the transceiver into a power-saving mode. The RESUMF flag must be cleared before setting SUSPND. Software must clear this bit after the resume flag (RESUMF) is set while this resume interrupt flag is serviced.

TXD2IE — Endpoint 2 Transmit Interrupt Enable

This read/write bit enables the transmit endpoint 2 to generate CPU interrupt requests when the TXD2F bit becomes set. Reset clears the TXD2IE bit.

1 = Transmit endpoint 2 can generate a CPU interrupt request
0 = Transmit endpoint 2 cannot generate a CPU interrupt request

RXD2IE — Endpoint 2 Receive Interrupt Enable

This read/write bit enables the receive endpoint 2 to generate CPU interrupt requests when the RXD2F bit becomes set. Reset clears the RXD2IE bit.

1 = Receive endpoint 2 can generate a CPU interrupt request
0 = Receive endpoint 2 cannot generate a CPU interrupt request

TXD1IE — Endpoint 1 Transmit Interrupt Enable

This read/write bit enables the transmit endpoint 1 to generate CPU interrupt requests when the TXD1F bit becomes set. Reset clears the TXD1IE bit.

1 = Transmit endpoints 1 can generate a CPU interrupt request
0 = Transmit endpoints 1 cannot generate a CPU interrupt request

TXD0IE — Endpoint 0 Transmit Interrupt Enable

This read/write bit enables the transmit endpoint 0 to generate CPU interrupt requests when the TXD0F bit becomes set. Reset clears the TXD0IE bit.

1 = Transmit endpoint 0 can generate a CPU interrupt request
0 = Transmit endpoint 0 cannot generate a CPU interrupt request

RXD0IE — Endpoint 0 Receive Interrupt Enable

This read/write bit enables the receive endpoint 0 to generate CPU interrupt requests when the RXD0F bit becomes set. Reset clears the RXD0IE bit.

1 = Receive endpoint 0 can generate a CPU interrupt request
0 = Receive endpoint 0 cannot generate a CPU interrupt request

11.8.3 USB Interrupt Register 1

Address: $003A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	EOPF	RSTF	TXD2F	RXD2F	TXD1F	RESUMF	TXD0F	RXD0F
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-17. USB Interrupt Register 1 (UIR1)

EOPF — End-of-Packet Detect Flag

This read-only bit is set when a valid end-of-packet sequence is detected on the D+ and D– lines. Software must clear this flag by writing a logic 1 to the EOPFR bit.

Reset clears this bit. Writing to EOPF has no effect.

1 = End-of-packet sequence has been detected
0 = End-of-packet sequence has not been detected

RSTF — USB Reset Flag

This read-only bit is set when a valid reset signal state is detected on the D+ and D– lines. If the URSTD bit of the configuration register (CONFIG) is clear, this reset detection will generate an internal reset signal to reset the CPU and other peripherals including the USB module. If the URSTD bit is set, this reset detection will generate an USB interrupt. This bit is cleared by writing a logic 1 to the RSTFR bit. This bit also is cleared by a POR reset.

NOTE: *The USB bit in the SRSR (see **8.8.2 SIM Reset Status Register (SRSR)**) is also a USB reset indicator.*

TXD2F — Endpoint 2 Data Transmit Flag

This read-only bit is set after the data stored in endpoint 2 transmit buffers has been sent and an ACK handshake packet from the host is received. Once the next set of data is ready in the transmit buffers, software must clear this flag by writing a logic 1 to the TXD2FR bit.

To enable the next data packet transmission, TX2E also must be set. If the TXD2F bit is not cleared, a NAK handshake will be returned in the next IN transaction.

Reset clears this bit. Writing to TXD2F has no effect.

1 = Transmit on endpoint 2 has occurred
0 = Transmit on endpoint 2 has not occurred

RXD2F — Endpoint 2 Data Receive Flag

This read-only bit is set after the USB module has received a data packet and responded with an ACK handshake packet. Software must clear this flag by writing a logic 1 to the RXD2FR bit after all of the received data has been read. Software also must set the RX2E bit to 1 to enable the next data packet reception. If the RXD2F bit is not cleared, a NAK handshake will be returned in the next OUT transaction.

Reset clears this bit. Writing to RXD2F has no effect.

1 = Receive on endpoint 2 has occurred
0 = Receive on endpoint 2 has not occurred

TXD1F — Endpoint 1 Data Transmit Flag

This read-only bit is set after the data stored in the endpoint 1 transmit buffer has been sent and an ACK handshake packet from the host is received. Once the next set of data is ready in the transmit buffers, software must clear this flag by writing a logic 1 to the TXD1FR bit. To enable the next data packet transmission, TX1E also must be set. If the TXD1F bit is not cleared, a NAK handshake will be returned in the next IN transaction.

Reset clears this bit. Writing to TXD1F has no effect.

1 = Transmit on endpoint 1has occurred
0 = Transmit on endpoint 1has not occurred

RESUMF — Resume Flag

This read-only bit is set when USB bus activity is detected while the SUSPND bit is set. Software must clear this flag by writing a logic 1 to the RESUMFR bit. Reset clears this bit. Writing a logic 0 to RESUMF has no effect.

1 = USB bus activity has been detected
0 = No USB bus activity has been detected

TXD0F — Endpoint 0 Data Transmit Flag

This read-only bit is set after the data stored in endpoint 0 transmit buffers has been sent and an ACK handshake packet from the host is received. Once the next set of data is ready in the transmit buffers, software must clear this flag by writing a logic 1 to the TXD0FR bit. To enable the next data packet transmission, TX0E also must be set. If the TXD0F bit is not cleared, a NAK handshake will be returned in the next IN transaction.

Reset clears this bit. Writing to TXD0F has no effect.

1 = Transmit on endpoint 0 has occurred
0 = Transmit on endpoint 0 has not occurred

RXD0F — Endpoint 0 Data Receive Flag

This read-only bit is set after the USB module has received a data packet and responded with an ACK handshake packet. Software must clear this flag by writing a logic 1 to the RXD0FR bit after all of the received data has been read. Software also must set the RX0E bit to 1 to enable the next data packet reception. If the RXD0F bit is not cleared, the USB will respond with a NAK handshake to any endpoint 0 OUT tokens; but does not respond to a SETUP token.

Reset clears this bit. Writing to RXD0F has no effect.

1 = Receive on endpoint 0 has occurred
0 = Receive on endpoint 0 has not occurred

11.8.4 USB Interrupt Register 2

Address: $0018

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	0
Write:	EOPFR	RSTFR	TXD2FR	RXD2FR	TXD1FR	RESUMFR	TXD0FR	RXD0FR
Reset:	0	0	0	0	0	0	0	0

Figure 11-18. USB Interrupt Register 2 (UIR2)

EOPFR — End-of-Packet Flag Reset

Writing a logic 1 to this write-only bit will clear the EOPF bit if it is set. Writing a logic 0 to the EOPFR has no effect. Reset clears this bit.

RSTFR — Clear Reset Indicator Bit

Writing a logic 1 to this write-only bit will clear the RSTF bit if it is set. Writing a logic 0 to the RSTFR has no effect. Reset clears this bit.

TXD2FR — Endpoint 2 Transmit Flag Reset

Writing a logic 1 to this write-only bit will clear the TXD2F bit if it is set. Writing a logic 0 to TXD2FR has no effect. Reset clears this bit.

RXD2FR — Endpoint 2 Receive Flag Reset

Writing a logic 1 to this write-only bit will clear the RXD2F bit if it is set. Writing a logic 0 to RXD2FR has no effect. Reset clears this bit.

TXD1FR — Endpoint 1 Transmit Flag Reset

Writing a logic 1 to this write-only bit will clear the TXD1F bit if it is set. Writing a logic 0 to TXD1FR has no effect. Reset clears this bit.

RESUMFR — Resume Flag Reset

Writing a logic 1 to this write-only bit will clear the RESUMF bit if it is set. Writing to RESUMFR has no effect. Reset clears this bit.

TXD0FR — Endpoint 0 Transmit Flag Reset

Writing a logic 1 to this write-only bit will clear the TXD0F bit if it is set. Writing a logic 0 to TXD0FR has no effect. Reset clears this bit.

RXD0FR — Endpoint 0 Receive Flag Reset

Writing a logic 1 to this write-only bit will clear the RXD0F bit if it is set. Writing a logic 0 to RXD0FR has no effect. Reset clears this bit.

11.8.5 USB Control Register 0

Address: $003B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	T0SEQ	0	TX0E	RX0E	TP0SIZ3	TP0SIZ2	TP0SIZ1	TP0SIZ0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 11-19. USB Control Register 0 (UCR0)

T0SEQ — Endpoint 0 Transmit Sequence Bit

This read/write bit determines which type of data packet (DATA0 or DATA1) will be sent during the next IN transaction directed at endpoint 0. Toggling of this bit must be controlled by software. Reset clears this bit.

1 = DATA1 token active for next endpoint 0 transmit
0 = DATA0 token active for next endpoint 0 transmit

TX0E — Endpoint 0 Transmit Enable

This read/write bit enables a transmit to occur when the USB host controller sends an IN token to endpoint 0. Software should set this bit when data is ready to be transmitted. It must be cleared by software when no more endpoint 0 data needs to be transmitted.

If this bit is 0 or the TXD0F is set, the USB will respond with a NAK handshake to any endpoint 0 IN tokens. Reset clears this bit.

1 = Data is ready to be sent
0 = Data is not ready. Respond with NAK

RX0E — Endpoint 0 Receive Enable

This read/write bit enables a receive to occur when the USB host controller sends an OUT token to endpoint 0. Software should set this bit when data is ready to be received. It must be cleared by software when data cannot be received.

If this bit is 0 or the RXD0F is set, the USB will respond with a NAK handshake to any endpoint 0 OUT tokens; but does not respond to a SETUP token. Reset clears this bit.

1 = Data is ready to be received
0 = Not ready for data. Respond with NAK

TP0SIZ3–TP0SIZ0 — Endpoint 0 Transmit Data Packet Size

These read/write bits store the number of transmit data bytes for the next IN token request for endpoint 0. These bits are cleared by reset.

11.8.6 USB Control Register 1

Address: $003C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	T1SEQ	STALL1	TX1E	FRESUM	TP1SIZ3	TP1SIZ2	TP1SIZ1	TP1SIZ0
Reset:	0	0	0	0	0	0	0	0

Figure 11-20. USB Control Register 1 (UCR1)

T1SEQ — Endpoint 1 Transmit Sequence Bit

This read/write bit determines which type of data packet (DATA0 or DATA1) will be sent during the next IN transaction directed to endpoint 1. Toggling of this bit must be controlled by software. Reset clears this bit.

1 = DATA1 token active for next endpoint 1 transmit
0 = DATA0 token active for next endpoint 1 transmit

STALL1 — Endpoint 1 Force Stall Bit

This read/write bit causes endpoint 1 to return a STALL handshake when polled by either an IN or OUT token by the USB host controller. Reset clears this bit.

1 = Send STALL handshake
0 = Default

TX1E — Endpoint 1 Transmit Enable

This read/write bit enables a transmit to occur when the USB host controller sends an IN token to endpoint 1. The appropriate endpoint enable bit, ENABLE1 bit in the UCR3 register, also should be set. Software should set the TX1E bit when data is ready to be transmitted. It must be cleared by software when no more data needs to be transmitted.

If this bit is 0 or the TXD1F is set, the USB will respond with a NAK handshake to any endpoint 1 directed IN tokens. Reset clears this bit.

1 = Data is ready to be sent
0 = Data is not ready. Respond with NAK

FRESUM — Force Resume

This read/write bit forces a resume state (K or non-idle state) onto the USB data lines to initiate a remote wakeup. Software should control the timing of the forced resume to be between 10 and 15 ms. Setting this bit will not cause the RESUMF bit to be set.

1 = Force data lines to K state
0 = Default

TP1SIZ3–TP1SIZ0 — Endpoint 1 Transmit Data Packet Size

These read/write bits store the number of transmit data bytes for the next IN token request for endpoint 1. These bits are cleared by reset.

11.8.7 USB Control Register 2

Address: $0019

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	T2SEQ	STALL2	TX2E	RX2E	TP2SIZ3	TP2SIZ2	TP2SIZ1	TP2SIZ0
Reset:	0	0	0	0	0	0	0	0

Figure 11-21. USB Control Register 2 (UCR2)

T2SEQ — Endpoint 2 Transmit Sequence Bit

This read/write bit determines which type of data packet (DATA0 or DATA1) will be sent during the next IN transaction directed to endpoint 2. Toggling of this bit must be controlled by software. Reset clears this bit.

1 = DATA1 token active for next endpoint 2 transmit
0 = DATA0 token active for next endpoint 2 transmit

STALL2 — Endpoint 2 Force Stall Bit

This read/write bit causes endpoint 2 to return a STALL handshake when polled by either an IN or OUT token by the USB host controller. Reset clears this bit.

1 = Send STALL handshake
0 = Default

TX2E — Endpoint 2 Transmit Enable

This read/write bit enables a transmit to occur when the USB host controller sends an IN token to endpoint 2. The appropriate endpoint enable bit, ENABLE2 bit in the UCR3 register, also should be set. Software should set the TX2E bit when data is ready to be transmitted. It must be cleared by software when no more data needs to be transmitted.

If this bit is 0 or the TXD2F is set, the USB will respond with a NAK handshake to any endpoint 2 directed IN tokens. Reset clears this bit.

1 = Data is ready to be sent
0 = Data is not ready. Respond with NAK

RX2E — Endpoint 2 Receive Enable

This read/write bit enables a receive to occur when the USB host controller sends an OUT token to endpoint 2. Software should set this bit when data is ready to be received. It must be cleared by software when data cannot be received.

If this bit is 0 or the RXD2F is set, the USB will respond with a NAK handshake to any endpoint 2 OUT tokens. Reset clears this bit.

1 = Data is ready to be received
0 = Not ready for data. Respond with NAK

TP2SIZ3–TP2SIZ0 — Endpoint 2 Transmit Data Packet Size

These read/write bits store the number of transmit data bytes for the next IN token request for endpoint 2. These bits are cleared by reset.

11.8.8 USB Control Register 3

Address: $001A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TX1ST	0	OSTALL0	ISTALL0	0	PULLEN	ENABLE2	ENABLE1
Write:		TX1STR						
Reset:	0	0	0	0	0	0*	0	0

= Unimplemented

* PULLEN bit is reset by POR or LVI reset only.

Figure 11-22. USB Control Register 3 (UCR3)

TX1ST — Endpoint 0 Transmit First Flag

This read-only bit is set if the endpoint 0 data transmit flag (TXD0F) is set when the USB control logic is setting the endpoint 0 data receive flag (RXD0F). In other words, if an unserviced endpoint 0 transmit flag is still set at the end of an endpoint 0 reception, then this bit will be set. This bit lets the firmware know that the endpoint 0 transmission happened before the endpoint 0 reception.

Reset clears this bit.

1 = IN transaction occurred before SETUP/OUT
0 = IN transaction occurred after SETUP/OUT

TX1STR — Clear Endpoint 0 Transmit First Flag

Writing a logic 1 to this write-only bit will clear the TX1ST bit if it is set. Writing a logic 0 to the TX1STR has no effect. Reset clears this bit.

OSTALL0 — Endpoint 0 Force STALL Bit for OUT token

This read/write bit causes endpoint 0 to return a STALL handshake when polled by an OUT token by the USB host controller. The USB hardware clears this bit when a SETUP token is received. Reset clears this bit.

1 = Send STALL handshake
0 = Default

ISTALL0 — Endpoint 0 Force STALL Bit for IN token

This read/write bit causes endpoint 0 to return a STALL handshake when polled by an IN token by the USB host controller. The USB hardware clears this bit when a SETUP token is received. Reset clears this bit.

1 = Send STALL handshake
0 = Default

PULLEN — Pull-up Enable

This read/write bit controls the pull-up option for the USB D– pin if the USB module is enabled.

1 = Configure D– pin to have internal pull-up
0 = Disconnect D– pin internal pull-up

ENABLE2 — Endpoint 2 Enable

This read/write bit enables endpoint 2 and allows the USB to respond to IN or OUT packets addressed to endpoint 2. Reset clears this bit.

1 = Endpoint 2 is enabled and can respond to an IN or OUT token
0 = Endpoint 2 is disabled

ENABLE1 — Endpoint 1 Enable

This read/write bit enables endpoint 1 and allows the USB to respond to IN packets addressed to endpoint 1. Reset clears this bit.

1 = Endpoint 1 is enabled and can respond to an IN token
0 = Endpoint 1 is disabled

11.8.9 USB Control Register 4

USB control register 4 directly controls the USB data pins D+ and D–. If the FUSBO bit, and the USBEN bit of the USB address register (UADDR) are set, the output buffers of the USB modules are enabled and the corresponding levels of the USB data pins D+ and D– are equal to the values set by the FDP and the FDM bits.

Address: $001B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	FUSBO	FDP	FDM
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-23. USB Control Register 4 (UCR4)

FUSBO — Force USB Output

This read/write bit enables the USB output buffers.

1 = Enables USB output buffers
0 = USB module in normal operation

FDP — Force D+

This read/write bit determinates the output level of D+.

1 = D+ at output high level
0 = D+ at output low level

FDM — Force D–

This read/write bit determinates the output level of D–.

1 = D– at output high level
0 = D– at output low level

NOTE: *Customers must be very careful when setting the UCR4 register. When the FUSBO and the USBEN bits are set, the USB module is in output mode and it will not recognize any USB signals including the USB reset signal. The UCR4 register is used for some special applications. Customers are not normally expected to use this register.*

11.8.10 USB Status Register 0

Address: $003D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R0SEQ	SETUP	0	0	RP0SIZ3	RP0SIZ2	RP0SIZ1	RP0SIZ0
Write:								
Reset:	Unaffected by reset							

= Unimplemented

Figure 11-24. USB Status Register 0 (USR0)

R0SEQ — Endpoint 0 Receive Sequence Bit

This read-only bit indicates the type of data packet last received for endpoint 0 (DATA0 or DATA1).

1 = DATA1 token received in last endpoint 0 receive
0 = DATA0 token received in last endpoint 0 receive

SETUP — SETUP Token Detect Bit

This read-only bit indicates that a valid SETUP token has been received.

1 = Last token received for endpoint 0 was a SETUP token
0 = Last token received for endpoint 0 was not a SETUP token

RP0SIZ3–RP0SIZ0 — Endpoint 0 Receive Data Packet Size

These read-only bits store the number of data bytes received for the last OUT or SETUP transaction for endpoint 0.

11.8.11 USB Status Register 1

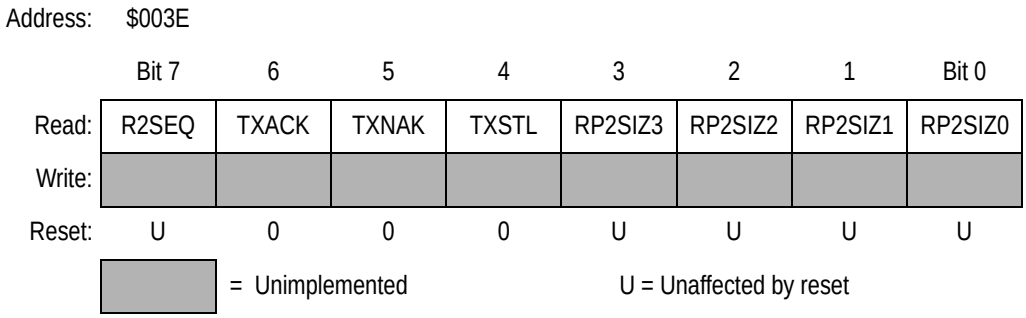


Figure 11-25. USB Status Register 2 (USR1)

R2SEQ — Endpoint 2 Receive Sequence Bit

This read-only bit indicates the type of data packet last received for endpoint 2 (DATA0 or DATA1).

1 = DATA1 token received in last endpoint 2 receive
0 = DATA0 token received in last endpoint 2 receive

TXACK — ACK Token Transmit Bit

This read-only bit indicates that an ACK token has been transmitted. This bit is updated at the end of the data transmission.

1 = Last token transmitted for endpoint 0 was an ACK token
0 = Last token transmitted for endpoint 0 was not an ACK token

TXNAK — NAK Token Transmit Bit

This read-only bit indicates that a TXNAK token has been transmitted. This bit is updated at the end of the data transmission.

1 = Last token transmitted for endpoint 0 was a NAK token
0 = Last token transmitted for endpoint 0 was not a NAK token

TXSTL — STALL Token Transmit Bit

This read-only bit indicates that a STALL token has been transmitted. This bit is updated at the end of the data transmission.

1 = Last token transmitted for endpoint 0 was a STALL token
0 = Last token transmitted for endpoint 0 was not a STALL token

RP2SIZ3–RP2SIZ0 — Endpoint 2 Receive Data Packet Size

These read-only bits store the number of data bytes received for the last OUT transaction for endpoint 2.

11.8.12 USB Endpoint 0 Data Registers

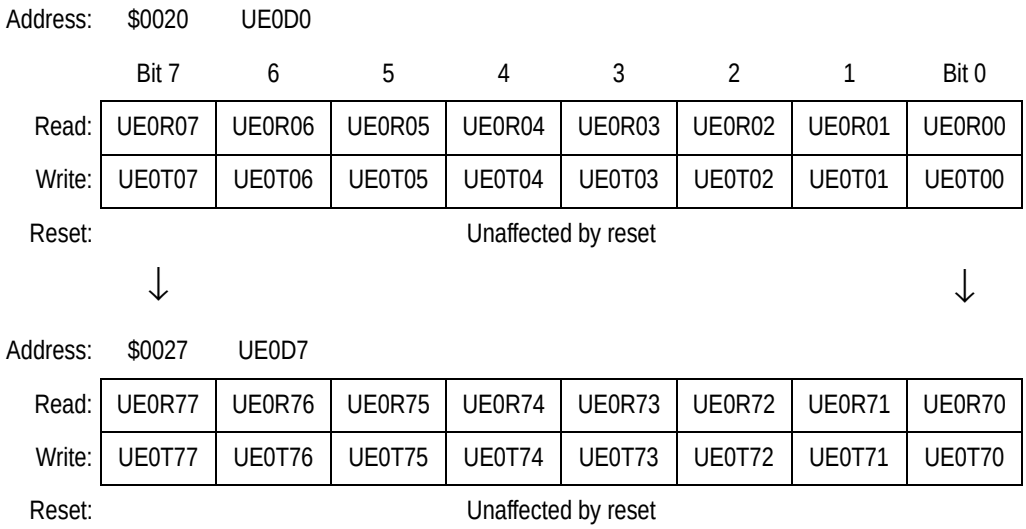


Figure 11-26. USB Endpoint 0 Data Registers (UE0D0–UE0D7)

UE0Rx7–UE0Rx0 — Endpoint 0 Receive Data Buffer

These read-only bits are serially loaded with OUT token or SETUP token data directed at endpoint 0. The data is received over the USB's D+ and D– pins.

UE0Tx7–UE0Tx0 — Endpoint 0 Transmit Data Buffer

These write-only buffers are loaded by software with data to be sent on the USB bus on the next IN token directed at endpoint 0.

11.8.13 USB Endpoint 1 Data Registers

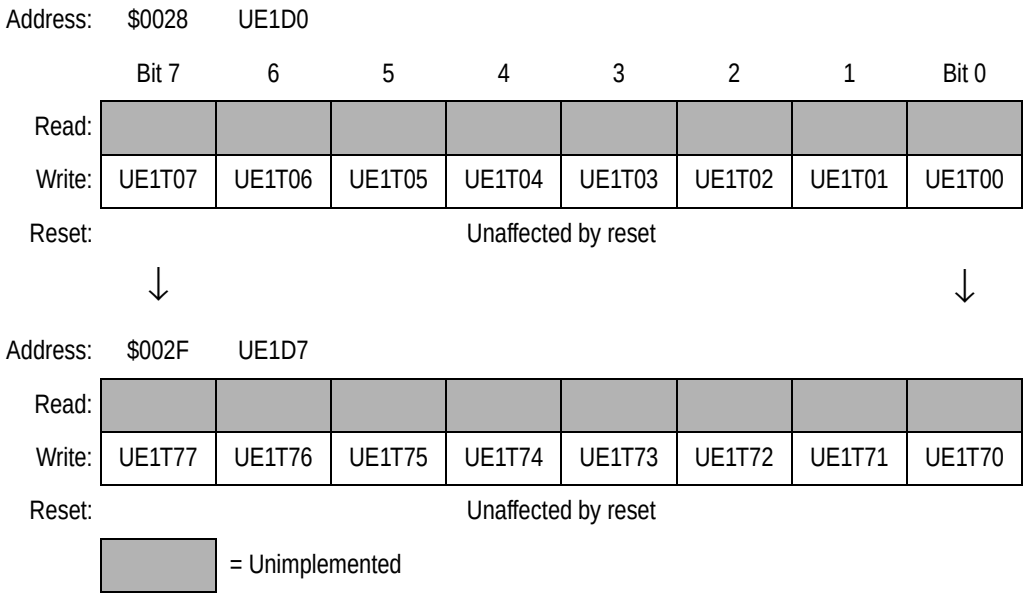


Figure 11-27. USB Endpoint 1 Data Registers (UE1D0–UE1D7)

UE1Tx7–UE1Tx0 — Endpoint 1 Transmit or Receive Data Buffer

These write-only buffers are loaded by software with data to be sent on the USB bus on the next IN token directed at endpoint 1.

11.8.14 USB Endpoint 2 Data Registers

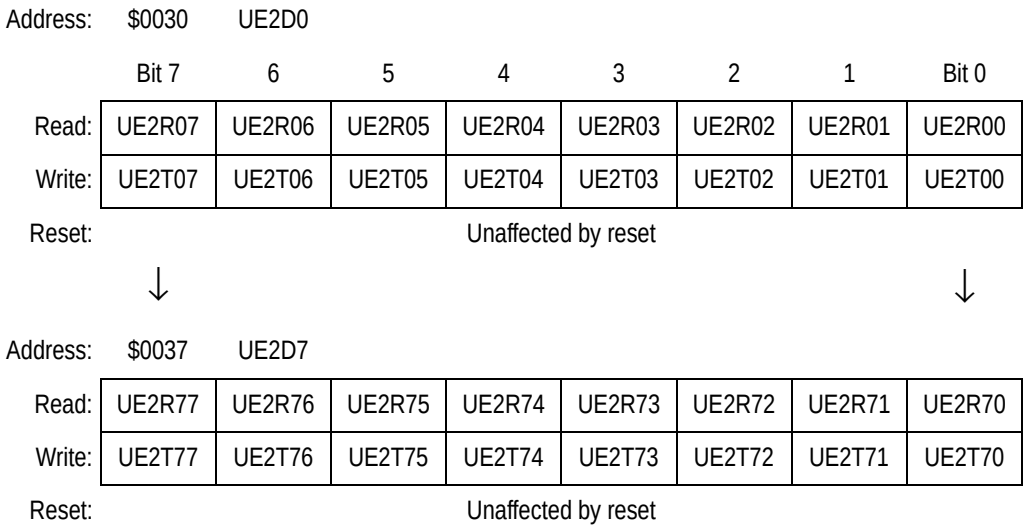


Figure 11-28. USB Endpoint 2 Data Registers (UE2D0–UE2D7)

UE2Rx7–UE2Rx0 — Endpoint 2 Receive Data Buffer

These read-only bits are serially loaded with OUT token data directed at endpoint 2. The data is received over the USB's D+ and D– pins.

UE2Tx7–UE2Tx0 — Endpoint 2 Transmit Data Buffer

These write-only buffers are loaded by software with data to be sent on the USB bus on the next IN token directed at endpoint 2.

11.9 USB Interrupts

The USB module is capable of generating interrupts and causing the CPU to execute the USB interrupt service routine. There are three types of USB interrupts:

- End-of-transaction interrupts signify either a completed transaction receive or transmit transaction.
- Resume interrupts signify that the USB bus is reactivated after having been suspended.
- End-of-packet interrupts signify that a low-speed end-of-packet signal was detected.

All USB interrupts share the same interrupt vector. Firmware is responsible for determining which interrupt is active.

11.9.1 USB End-of-Transaction Interrupt

There are five possible end-of-transaction interrupts:

- Endpoint 0 or 2 receive
- Endpoint 0, 1 or 2 transmit

End-of-transaction interrupts occur as detailed in the following sections.

11.9.1.1 Receive Control Endpoint 0

For a control OUT transaction directed at endpoint 0, the USB module will generate an interrupt by setting the RXD0F flag in the UIR0 register. The conditions necessary for the interrupt to occur are shown in the flowchart in **Figure 11-29**.

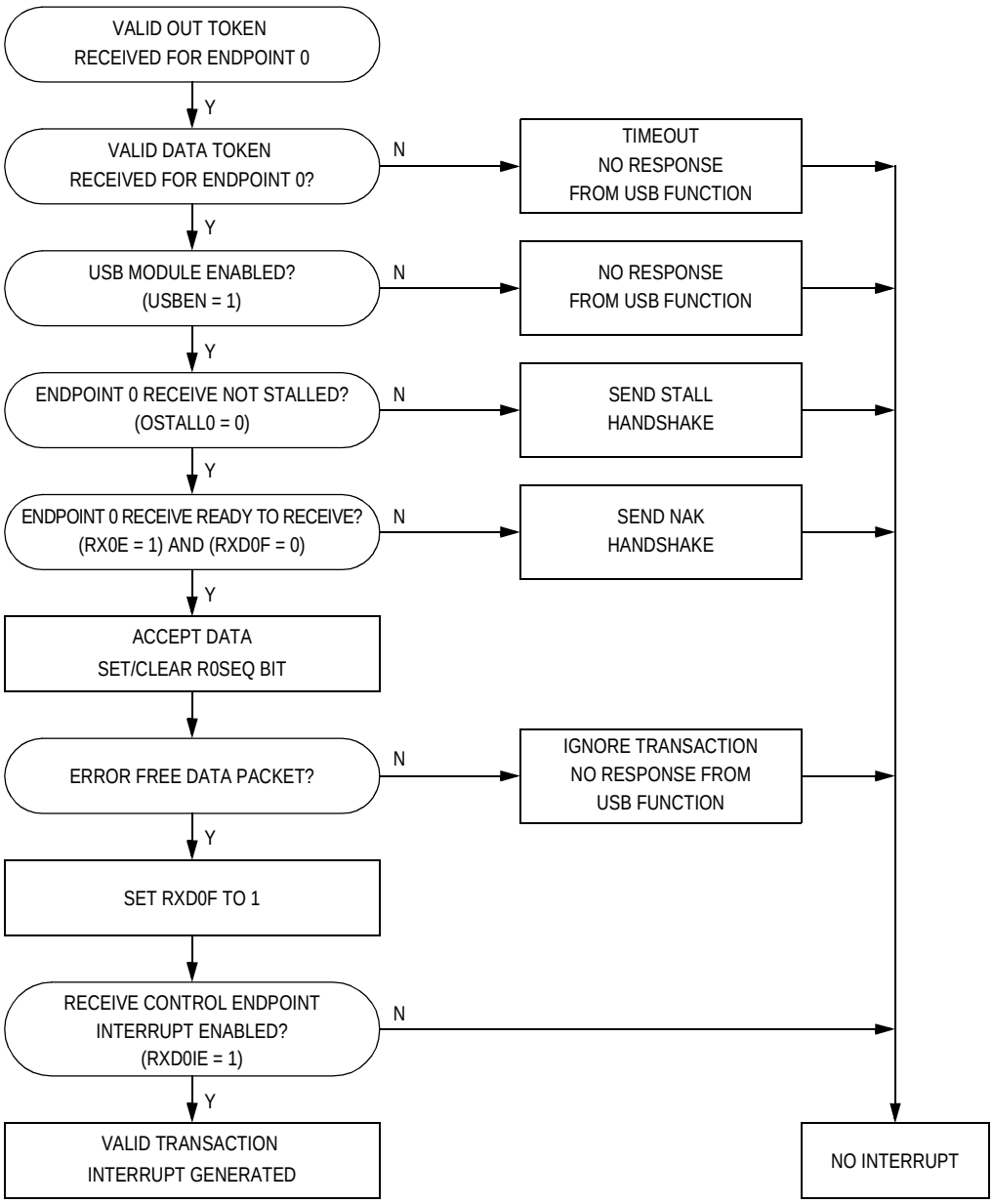


Figure 11-29. OUT Token Data Flow for Receive Endpoint 0

SETUP transactions cannot be stalled by the USB function. A SETUP received by a control endpoint will clear the ISTALL0 and OSTALL0 bits. The conditions for receiving a SETUP interrupt are shown in **Figure 11-30**.

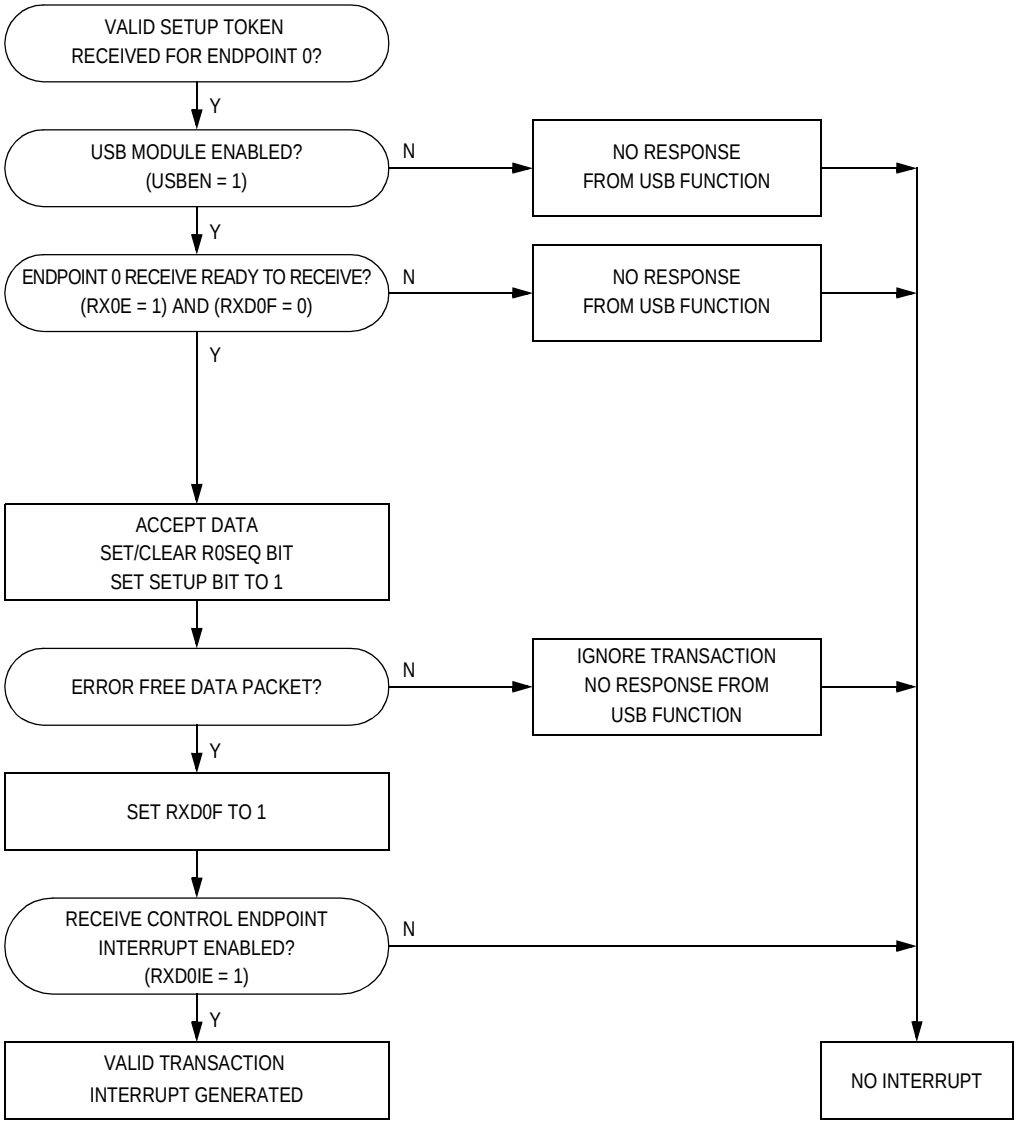


Figure 11-30. SETUP Token Data Flow for Receive Endpoint 0

11.9.1.2 Transmit Control Endpoint 0

For a control IN transaction directed at endpoint 0, the USB module will generate an interrupt by setting the TXD0F flag in the UIR1 register. The conditions necessary for the interrupt to occur are shown in the flowchart in **Figure 11-31**.

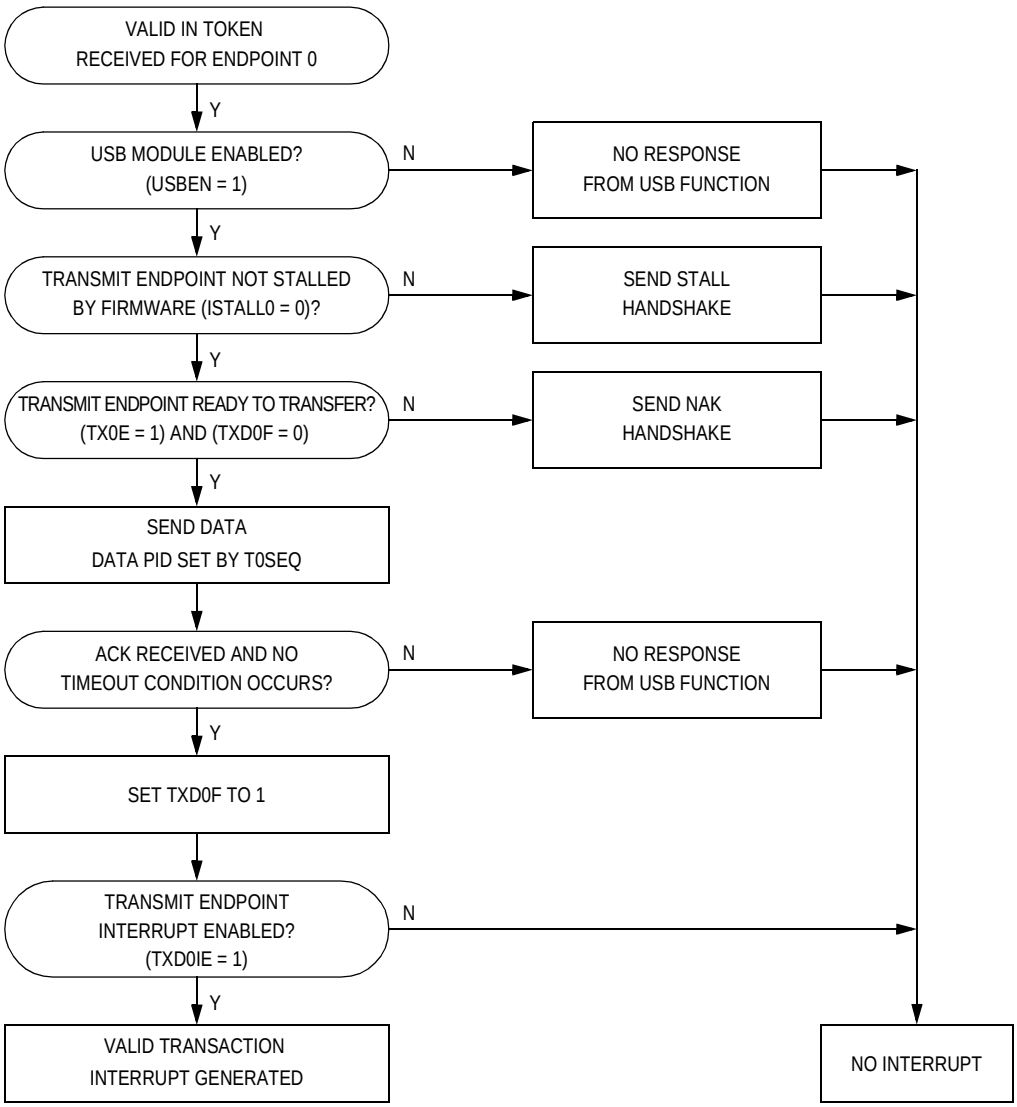


Figure 11-31. IN Token Data Flow for Transmit Endpoint 0

11.9.1.3 Transmit Endpoint 1

For an IN transaction directed at endpoint 1, the USB module will generate an interrupt by setting the TXD1F in the UIR1 register. The conditions necessary for the interrupt to occur are shown in **Figure 11-32**.

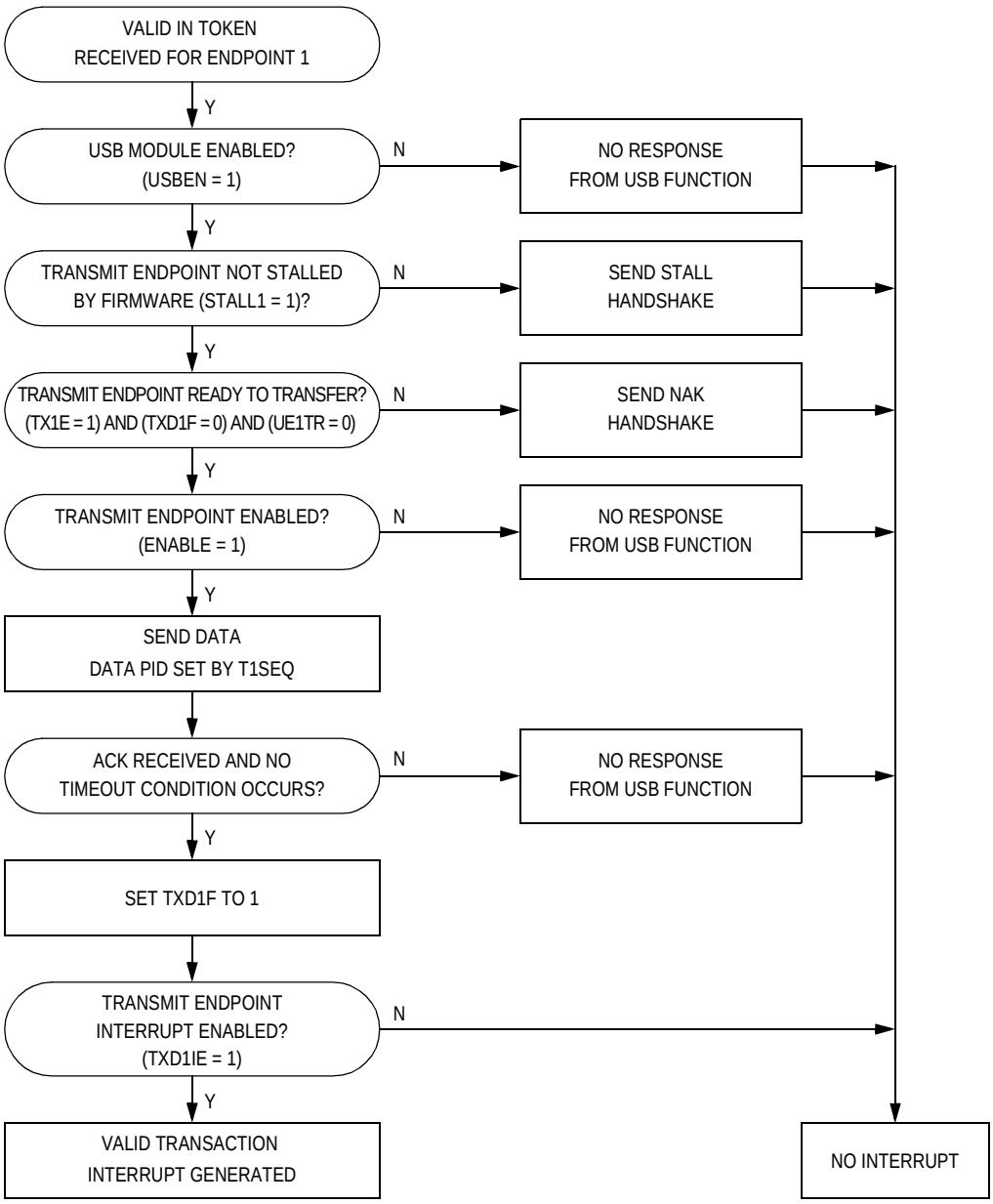


Figure 11-32. IN Token Data Flow for Transmit Endpoint 1

11.9.1.4 Transmit Endpoint 2

For an IN transaction directed at endpoint 2, the USB module will generate an interrupt by setting the TXD2F in the UIR1 register.

11.9.1.5 Receive Endpoint 2

For an OUT transaction directed at endpoint 2, the USB module will generate an interrupt by setting the RXD2F in the UIR1 register.

11.9.2 Resume Interrupt

The USB module will generate a CPU interrupt if low-speed bus activity is detected after entering the suspend state. A transition of the USB data lines to the non-idle state (K state) while in the suspend mode will set the RESUMF flag in the UIR1 register. There is no interrupt enable bit for this interrupt source and an interrupt will be executed if the I-bit in the CCR is cleared. A resume interrupt can only occur while the MCU is in the suspend mode.

11.9.3 End-of-Packet Interrupt

The USB module can generate a USB interrupt upon detection of an end-of-packet signal for low-speed devices. Upon detection of an end-of-packet signal, the USB module sets the EOPF bit and will generate a CPU interrupt if the EOPIE bit in the UIR0 register is set.

Section 12. Serial Communications Interface Module (SCI)

12.1 Contents

12.2 Introduction

This section describes the serial communications interface (SCI) module, which allows high-speed asynchronous communications with peripheral devices and other MCUs.

NOTE: *References to DMA (direct-memory access) and associated functions are only valid if the MCU has a DMA module. This MCU does not have the DMA function. Any DMA-related register bits should be left in their reset state for normal MCU operation.*

12.3 Features

Features of the SCI module include the following:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- 32 programmable baud rates
- Programmable 8-bit or 9-bit character length
- Separately enabled transmitter and receiver
- Separate receiver and transmitter CPU interrupt requests
- Programmable transmitter output polarity
- Baud rate clock source is OSCDCLK (2 × OSCXCLK)

- Two receiver wakeup methods:
  - Idle line wakeup
  - Address mark wakeup
- Interrupt-driven operation with eight interrupt flags:
  - Transmitter empty
  - Transmission complete
  - Receiver full
  - Idle receiver input
  - Receiver overrun
  - Noise error
  - Framing error
  - Parity error
- Receiver framing error detection
- Hardware parity checking
- 1/16 bit-time noise detection

12.4 Pin Name Conventions

The generic names of the SCI I/O pins are:

- RxD (receive data)
- TxD (transmit data)

SCI I/O (input/output) lines are implemented by sharing parallel I/O port pins. The full name of an SCI input or output reflects the name of the shared port pin. **Table 12-1** shows the full names and the generic names of the SCI I/O pins.

The generic pin names appear in the text of this section.

Table 12-1. Pin Name Conventions

Generic Pin Names:	RxD	TxD
Full Pin Names:	PTC1/RxD	PTC0/TxD

12.5 Functional Description

Figure 12-1 shows the structure of the SCI module. The SCI allows full-duplex, asynchronous, NRZ serial communication among the MCU and remote devices, including other MCUs. The transmitter and receiver of the SCI operate independently, although they use the same baud rate generator. During normal operation, the CPU monitors the status of the SCI, writes the data to be transmitted, and processes received data.

The baud rate clock source for the SCI is the OSCDCLK from the oscillator circuit, which is two times the crystal clock, OSCXCLK.

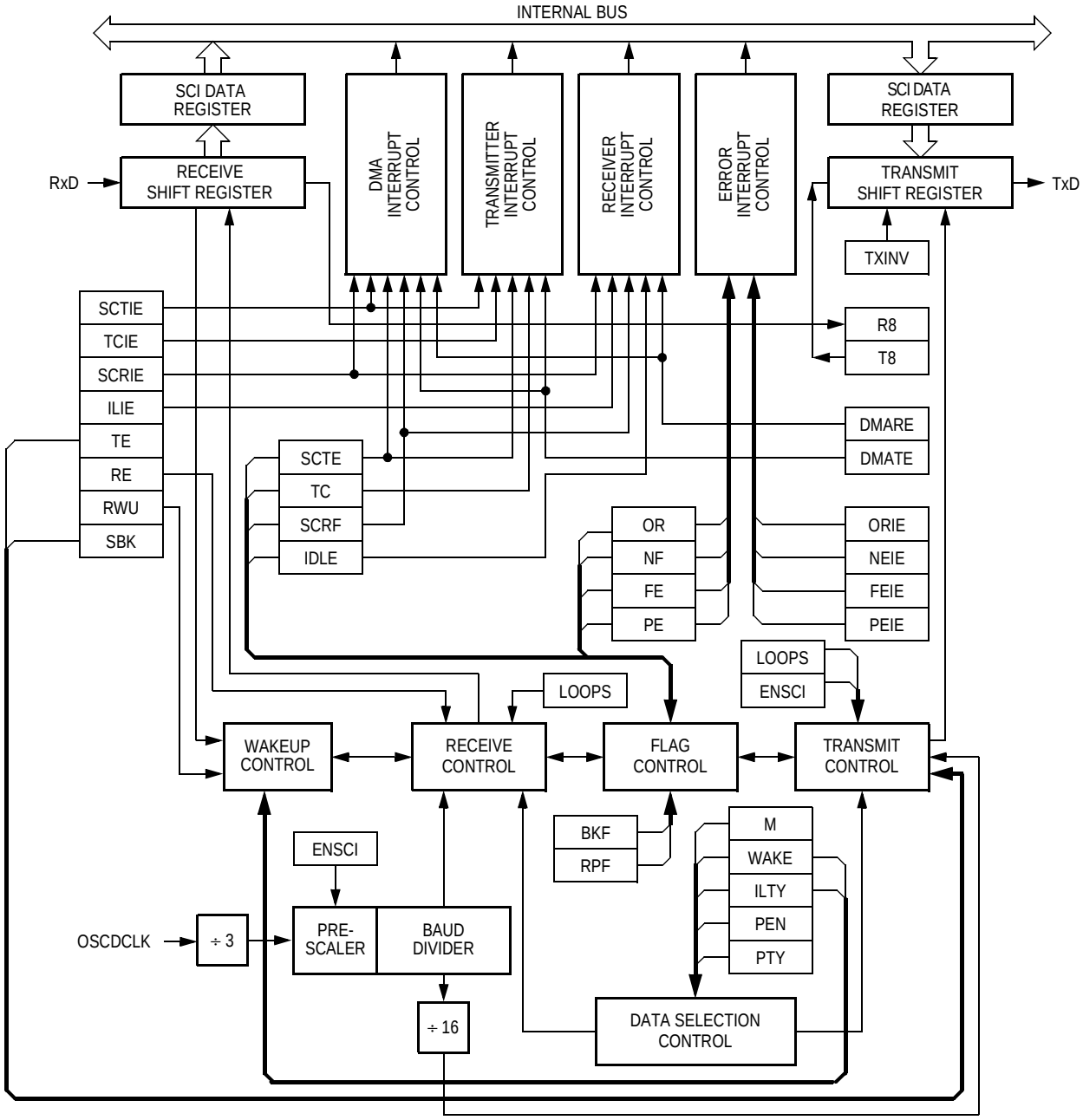


Figure 12-1. SCI Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$005A	SCI Control Register 1 (SCC1)	Read: Write:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Reset:	0	0	0	0	0	0	0	0
$005B	SCI Control Register 2 (SCC2)	Read: Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Reset:	0	0	0	0	0	0	0	0
$005C	SCI Control Register 3 (SCC3)	Read:	R8	T8	DMARE	DMATE	ORIE	NEIE	FEIE	PEIE
		Write:								
		Reset:	U	U	0	0	0	0	0	0
$005D	SCI Status Register 1 (SCS1)	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:								
		Reset:	1	1	0	0	0	0	0	0
$005E	SCI Status Register 2 (SCS2)	Read:							BKF	RPF
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$005F	SCI Data Register (SCDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							
$0060	SCI Baud Rate Register (SCBR)	Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:								
		Reset:	0	0	0	0		0	0	0

= Unimplemented R = Reserved U = Unaffected

Figure 12-2. SCI I/O Register Summary

12.5.1 Data Format

The SCI uses the standard non-return-to-zero mark/space data format illustrated in **Figure 12-3**.

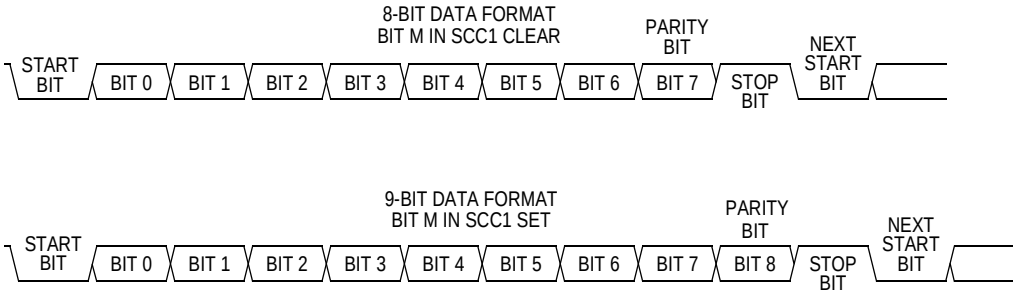


Figure 12-3. SCI Data Formats

12.5.2 Transmitter

Figure 12-4 shows the structure of the SCI transmitter.

The baud rate clock source for the SCI is the OSCDCLK.

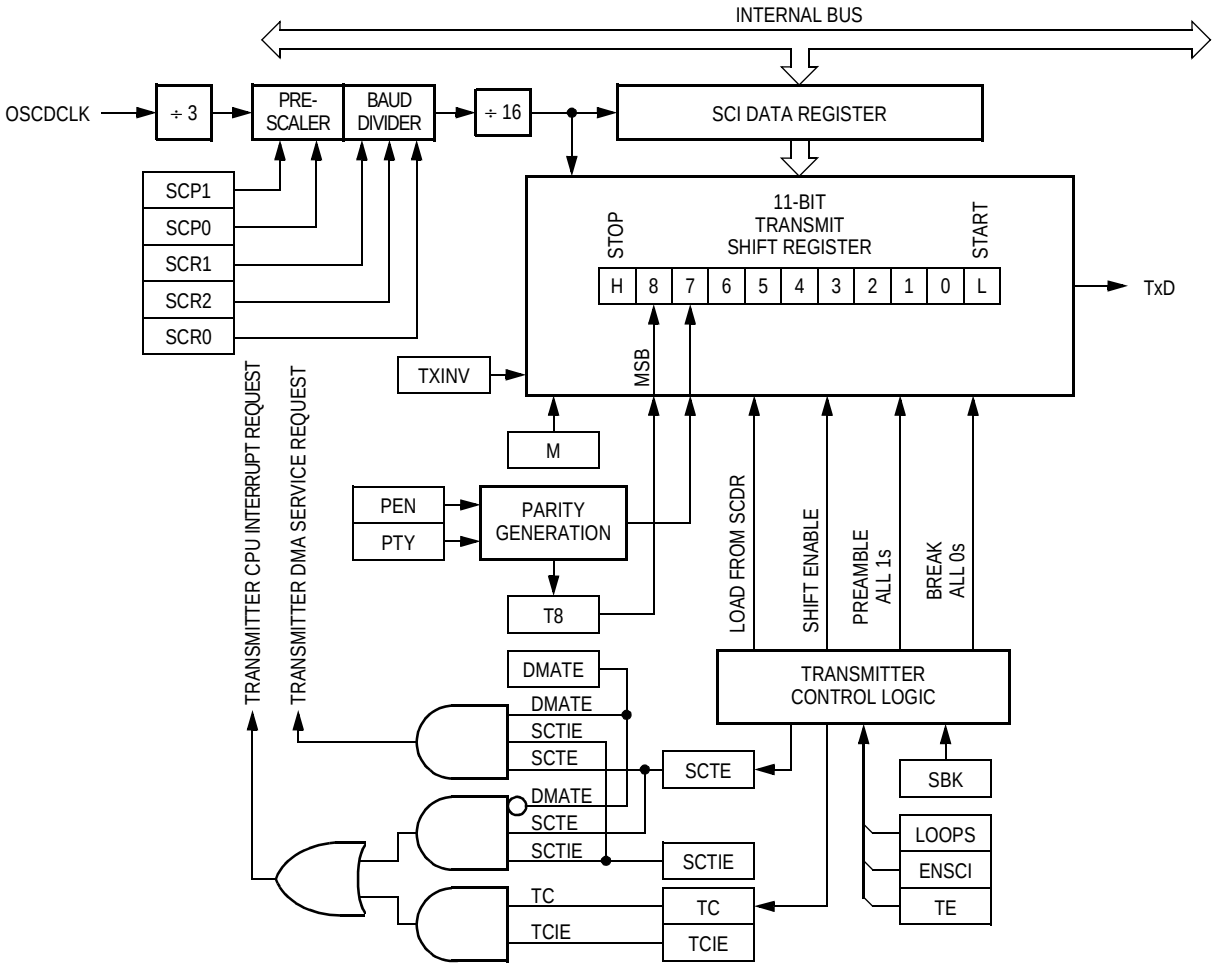


Figure 12-4. SCI Transmitter

12.5.2.1 Character Length

The transmitter 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 transmitting 9-bit data, bit T8 in SCI control register 3 (SCC3) is the ninth bit (bit 8).

12.5.2.2 Character Transmission

During an SCI transmission, the transmit shift register shifts a character out to the TxD pin. The SCI data register (SCDR) is the write-only buffer between the internal data bus and the transmit shift register. To initiate an SCI transmission:

1. Enable the SCI by writing a logic 1 to the enable SCI bit (ENSCI) in SCI control register 1 (SCC1).
2. Enable the transmitter by writing a logic 1 to the transmitter enable bit (TE) in SCI control register 2 (SCC2).
3. Clear the SCI transmitter empty bit by first reading SCI status register 1 (SCS1) and then writing to the SCDR.
4. Repeat step 3 for each subsequent transmission.

At the start of a transmission, transmitter control logic automatically loads the transmit shift register with a preamble of logic 1s. After the preamble shifts out, control logic transfers the SCDR data into the transmit shift register. A logic 0 start bit automatically goes into the least significant bit position of the transmit shift register. A logic 1 stop bit goes into the most significant bit position.

The SCI transmitter empty bit, SCTE, in SCS1 becomes set when the SCDR transfers a byte to the transmit shift register. The SCTE bit indicates that the SCDR can accept new data from the internal data bus. If the SCI transmit interrupt enable bit, SCTIE, in SCC2 is also set, the SCTE bit generates a transmitter CPU interrupt request.

When the transmit shift register is not transmitting a character, the TxD pin goes to the idle condition, logic 1. If at any time software clears the ENSCI bit in SCI control register 1 (SCC1), the transmitter and receiver relinquish control of the port pin.

12.5.2.3 Break Characters

Writing a logic 1 to the send break bit, SBK, in SCC2 loads the transmit shift register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on the M bit in SCC1. As long as SBK is at logic 1, transmitter logic continuously loads break characters into the transmit shift register. After software clears the SBK bit, the shift register finishes transmitting the last break character and then transmits at least one logic 1. The automatic logic 1 at the end of a break character guarantees the recognition of the start bit of the next character.

The SCI recognizes a break character when a start bit is followed by eight or nine logic 0 data bits and a logic 0 where the stop bit should be.

Receiving a break character has these effects on SCI registers:

- Sets the framing error bit (FE) in SCS1
- Sets the SCI receiver full bit (SCRF) in SCS1
- Clears the SCI data register (SCDR)
- Clears the R8 bit in SCC3
- Sets the break flag bit (BKF) in SCS2
- May set the overrun (OR), noise flag (NF), parity error (PE), or reception in progress flag (RPF) bits

12.5.2.4 Idle Characters

An idle character contains all logic 1s and has no start, stop, or parity bit. Idle character length depends on the M bit in SCC1. The preamble is a synchronizing idle character that begins every transmission.

If the TE bit is cleared during a transmission, the TxD pin becomes idle after completion of the transmission in progress. Clearing and then setting the TE bit during a transmission queues an idle character to be sent after the character currently being transmitted.

NOTE: *When queueing an idle character, return the TE bit to logic 1 before the stop bit of the current character shifts out to the TxD pin. Setting TE after the stop bit appears on TxD causes data previously written to the SCDR to be lost.*

Toggle the TE bit for a queued idle character when the SCTE bit becomes set and just before writing the next byte to the SCDR.

12.5.2.5 Inversion of Transmitted Output

The transmit inversion bit (TXINV) in SCI control register 1 (SCC1) reverses the polarity of transmitted data. All transmitted values, including idle, break, start, and stop bits, are inverted when TXINV is at logic 1. (See **12.9.1 SCI Control Register 1**.)

12.5.2.6 Transmitter Interrupts

These conditions can generate CPU interrupt requests from the SCI transmitter:

- SCI transmitter empty (SCTE) — The SCTE bit in SCS1 indicates that the SCDR 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 SCC2 enables the SCTE bit to generate transmitter CPU interrupt requests.
- Transmission complete (TC) — The TC bit in SCS1 indicates that the transmit shift register and the SCDR are empty and that no break or idle character has been generated. The transmission complete interrupt enable bit, TCIE, in SCC2 enables the TC bit to generate transmitter CPU interrupt requests.

12.5.3 Receiver

Figure 12-5 shows the structure of the SCI receiver.

12.5.3.1 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).

12.5.3.2 Character Reception

During an SCI reception, the receive shift register shifts characters in from the RxD pin. The SCI data register (SCDR) is the read-only buffer between the internal data bus and the receive shift register.

After a complete character shifts into the receive shift register, the data portion of the character transfers to the SCDR. The SCI receiver full bit, SCRF, in SCI status register 1 (SCS1) becomes set, indicating that the received byte can be read. If the SCI receive interrupt enable bit, SCRIE, in SCC2 is also set, the SCRF bit generates a receiver CPU interrupt request.

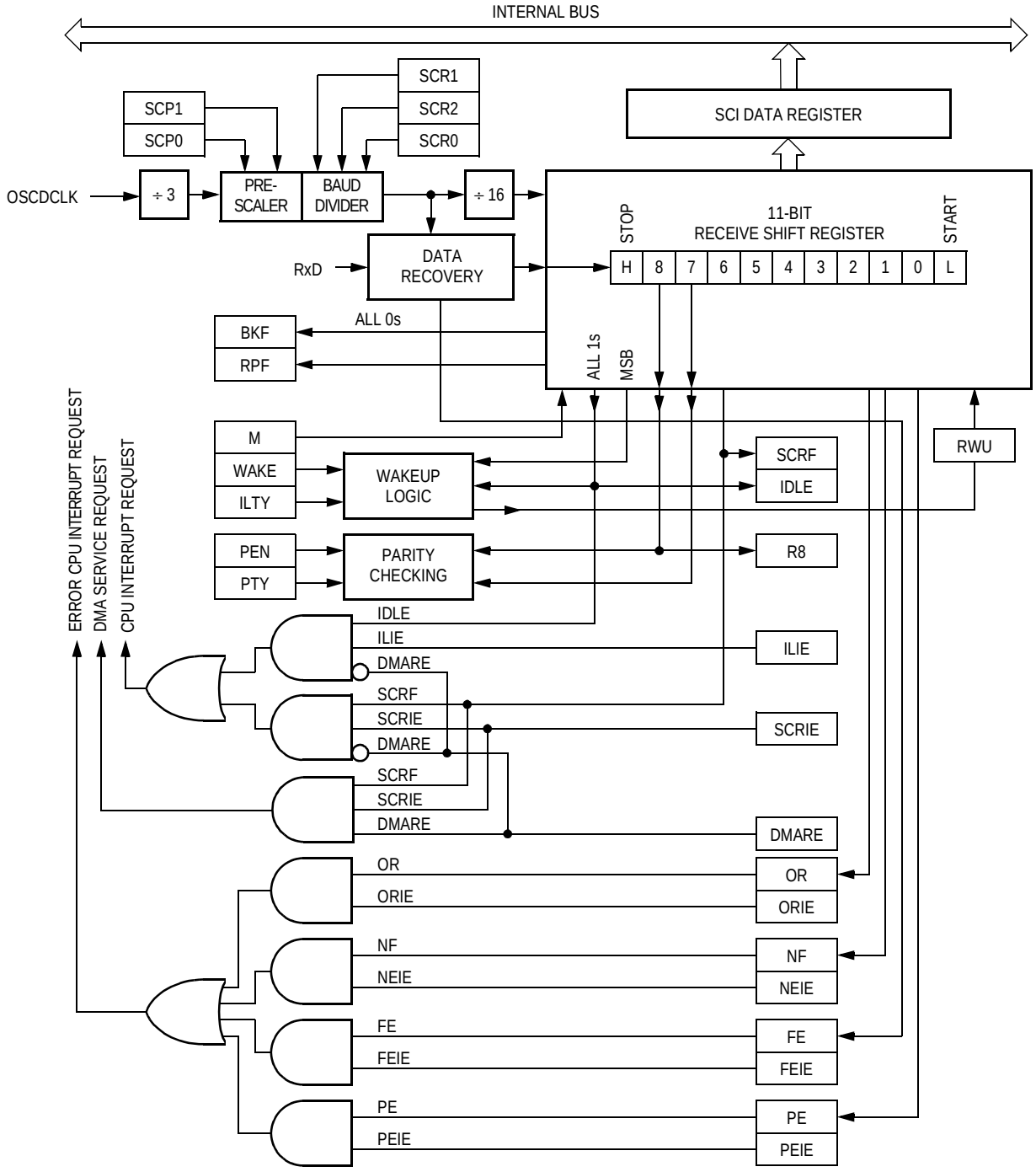


Figure 12-5. SCI Receiver Block Diagram

12.5.3.3 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 12-6**):

- 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.

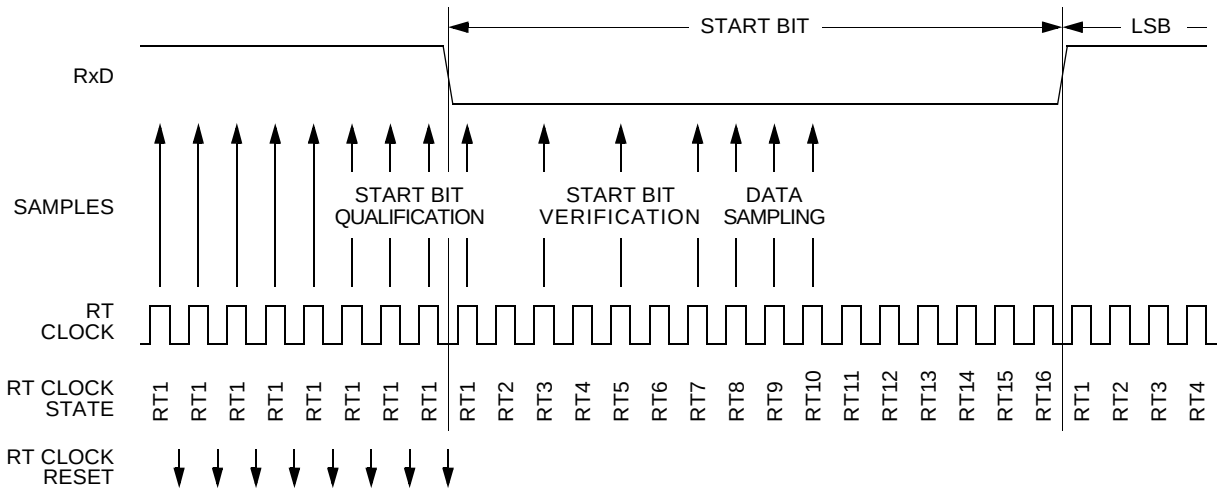


Figure 12-6. Receiver Data Sampling

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7. **Table 12-2** summarizes the results of the start bit verification samples.

Table 12-2. 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

Start bit verification is not successful if any two of the three verification samples are logic 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 12-3** summarizes the results of the data bit samples.

Table 12-3. 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 logic 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 12-4** summarizes the results of the stop bit samples.

Table 12-4. 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

12.5.3.4 Framing Errors

If the data recovery logic does not detect a logic 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.

12.5.3.5 Baud Rate Tolerance

A transmitting device may be operating at a baud rate below or above the receiver baud rate. Accumulated bit time misalignment can cause one of the three stop bit data samples to fall outside the actual stop bit. Then a noise error occurs. If more than one of the samples is outside the stop bit, a framing error occurs. In most applications, the baud rate

tolerance is much more than the degree of misalignment that is likely to occur.

As the receiver samples an incoming character, it resynchronizes the RT clock on any valid falling edge within the character. Resynchronization within characters corrects misalignments between transmitter bit times and receiver bit times.

Slow Data Tolerance

Figure 12-7 shows how much a slow received character can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.

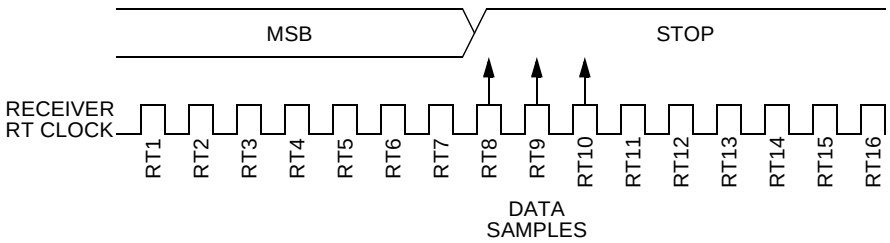


Figure 12-7. Slow Data

For an 8-bit character, data sampling of the stop bit takes the receiver 9 bit times × 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in **Figure 12-7**, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 9 bit times × 16 RT cycles + 3 RT cycles = 147 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit character with no errors is

$$\left|\frac{154 - 147}{154}\right| \times 100 = 4.54\%$$

For a 9-bit character, data sampling of the stop bit takes the receiver 10 bit times × 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in **Figure 12-7**, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 10 bit times × 16 RT cycles + 3 RT cycles = 163 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is

$$\left|\frac{170 - 163}{170}\right| \times 100 = 4.12\%$$

Fast Data Tolerance

Figure 12-8 shows how much a fast received character can be misaligned without causing a noise error or a framing error. The fast stop bit ends at RT10 instead of RT16 but is still there for the stop bit data samples at RT8, RT9, and RT10.

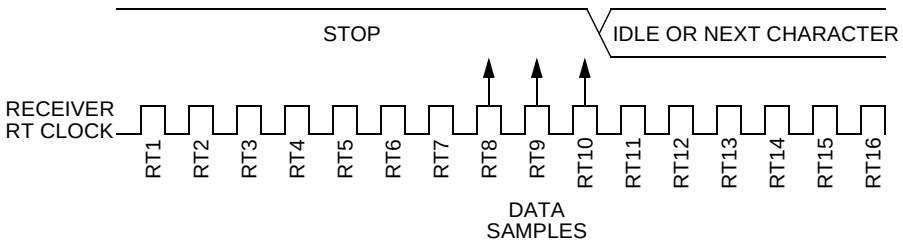


Figure 12-8. Fast Data

For an 8-bit character, data sampling of the stop bit takes the receiver 9 bit times × 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in **Figure 12-8**, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 10 bit times × 16 RT cycles = 160 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is

$$\left|\frac{154 - 160}{154}\right| \times 100 = 3.90\dot{}\%$$

For a 9-bit character, data sampling of the stop bit takes the receiver 10 bit times × 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in **Figure 12-8**, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 11 bit times × 16 RT cycles = 176 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is

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

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 SCC2 puts the receiver into a standby state during which receiver interrupts are disabled.

Depending on the state of the WAKE bit in SCC1, 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: *With the WAKE bit clear, setting the RWU 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 SCS1 indicates that the receive shift register has transferred a character to the SCDR. SCRF can generate a receiver CPU interrupt request. Setting the SCI receive interrupt enable bit, SCRIE, in SCC2 enables the SCRF bit to generate receiver CPU interrupts.
- Idle input (IDLE) — The IDLE bit in SCS1 indicates that 10 or 11 consecutive logic 1s shifted in from the RxD pin. The idle line interrupt enable bit, ILIE, in SCC2 enables the IDLE bit to generate CPU interrupt requests.

12.5.3.8 Error Interrupts

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

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

- Framing error (FE) — The FE bit in SCS1 is set when a logic 0 occurs where the receiver expects a stop bit. The framing error interrupt enable bit, FEIE, in SCC3 enables FE to generate SCI error CPU interrupt requests.
- Parity error (PE) — The PE bit in SCS1 is set when the SCI detects a parity error in incoming data. The parity error interrupt enable bit, PEIE, in SCC3 enables PE to generate SCI error CPU interrupt requests.

12.6 Low-Power Modes

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

12.6.1 Wait Mode

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

If SCI module functions are not required during wait mode, reduce power consumption by disabling the module before executing the WAIT instruction.

Refer to **8.7 Low-Power Modes** for information on exiting wait mode.

12.6.2 Stop Mode

The SCI module is inactive after the execution of a STOP instruction. The STOP instruction does not affect SCI register states. SCI module operation resumes after an external interrupt.

Because the internal clock is inactive during stop mode, entering stop mode during an SCI transmission or reception results in invalid data.

Refer to **8.7 Low-Power Modes** for information on exiting stop mode.

12.7 SCI During Break Module Interrupts

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.

To allow software to clear status bits during a break interrupt, write a logic 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 logic 0 to the BCFE bit. With BCFE at logic 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 break, the bit cannot change during the break state as long as BCFE is at logic 0. After the break, doing the second step clears the status bit.

12.8 I/O Signals

Port C shares two of its pins with the SCI module. The two SCI I/O pins are:

- PTC0/TxD — Transmit data
- PTC1/RxD — Receive data

12.8.1 TxD (Transmit Data)

The PTC0/TxD pin is the serial data output from the SCI transmitter. The SCI shares the PTC0/TxD pin with port C. When the SCI is enabled, the PTC0/TxD pin is an output regardless of the state of the DDRC0 bit in data direction register C (DDRC).

12.8.2 RxD (Receive Data)

The PTC1/RxD pin is the serial data input to the SCI receiver. The SCI shares the PTC1/RxD pin with port C. When the SCI is enabled, the PTC1/RxD pin is an input regardless of the state of the DDRC1 bit in data direction register C (DDRC).

12.9 I/O Registers

These I/O registers control and monitor SCI operation:

- SCI control register 1 (SCC1)
- SCI control register 2 (SCC2)
- SCI control register 3 (SCC3)
- SCI status register 1 (SCS1)
- SCI status register 2 (SCS2)
- SCI data register (SCDR)
- SCI baud rate register (SCBR)

12.9.1 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: $005A

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
Reset:	0	0	0	0	0	0	0	0

Figure 12-9. 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 12-5**.) 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

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 12-5**.) When enabled, the parity function inserts a parity bit in the most significant bit position. (See **Figure 12-3**.) 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 12-5**.) 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 12-5. Character Format Selection

Control Bits		Character Format				
M	**PEN and 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

12.9.2 SCI Control Register 2

SCI control register 2:

- Enables the following CPU interrupt requests:
  - Enables the SCTE bit to generate transmitter CPU interrupt requests
  - Enables the TC bit to generate transmitter CPU interrupt requests
  - Enables the SCRF bit to generate receiver CPU interrupt requests
  - Enables the IDLE bit to generate receiver CPU interrupt requests

- Enables the transmitter
- Enables the receiver
- Enables SCI wakeup
- Transmits SCI break characters

Address: $005B

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0

Figure 12-10. SCI Control Register 2 (SCC2)

SCTIE — SCI Transmit Interrupt Enable Bit

This read/write bit enables the SCTE bit to generate SCI transmitter CPU interrupt requests. Reset clears the SCTIE bit.

1 = SCTE enabled to generate CPU interrupt
0 = SCTE not enabled to generate CPU interrupt

TCIE — Transmission Complete Interrupt Enable Bit

This read/write bit enables the TC bit to generate SCI transmitter CPU interrupt requests. Reset clears the TCIE bit.

1 = TC enabled to generate CPU interrupt requests
0 = TC not enabled to generate CPU interrupt requests

SCRIE — SCI Receive Interrupt Enable Bit

This read/write bit enables the SCRF bit to generate SCI receiver CPU interrupt requests. Reset clears the SCRIE bit.

1 = SCRF enabled to generate CPU interrupt
0 = SCRF not enabled to generate CPU interrupt

ILIE — Idle Line Interrupt Enable Bit

This read/write bit enables the IDLE bit to generate SCI receiver CPU interrupt requests. Reset clears the ILIE bit.

1 = IDLE enabled to generate CPU interrupt requests
0 = IDLE not enabled to generate CPU interrupt requests

TE — Transmitter Enable Bit

Setting this read/write bit begins the transmission by sending a preamble of 10 or 11 logic 1s from the transmit shift register to the TxD pin. If software clears the TE bit, the transmitter completes any transmission in progress before the TxD returns to the idle condition (logic 1). Clearing and then setting TE during a transmission queues an idle character to be sent after the character currently being transmitted. Reset clears the TE bit.

1 = Transmitter enabled
0 = Transmitter disabled

NOTE: *Writing to the TE bit is not allowed when the enable SCI bit (ENSCI) is clear. ENSCI is in SCI control register 1.*

RE — Receiver Enable Bit

Setting this read/write bit enables the receiver. Clearing the RE bit disables the receiver but does not affect receiver interrupt flag bits. Reset clears the RE bit.

1 = Receiver enabled
0 = Receiver disabled

NOTE: *Writing to the RE bit is not allowed when the enable SCI bit (ENSCI) is clear. ENSCI is in SCI control register 1.*

RWU — Receiver Wakeup Bit

This read/write bit puts the receiver in a standby state during which receiver interrupts are disabled. The WAKE bit in SCC1 determines whether an idle input or an address mark brings the receiver out of the standby state and clears the RWU bit. Reset clears the RWU bit.

1 = Standby state
0 = Normal operation

SBK — Send Break Bit

Setting and then clearing this read/write bit transmits a break character followed by a logic 1. The logic 1 after the break character guarantees recognition of a valid start bit. If SBK remains set, the transmitter continuously transmits break characters with no logic 1s between them. Reset clears the SBK bit.

1 = Transmit break characters
0 = No break characters being transmitted

NOTE: *Do not toggle the SBK bit immediately after setting the SCTE bit. Toggling SBK before the preamble begins causes the SCI to send a break character instead of a preamble.*

12.9.3 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 these interrupts:
  - Receiver overrun interrupts
  - Noise error interrupts
  - Framing error interrupts
- Parity error interrupts

Address: $005C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R8	T8	DMARE	DMATE	ORIE	NEIE	FEIE	PEIE
Write:								
Reset:	U	U	0	0	0	0	0	0

= Unimplemented U = Unaffected

Figure 12-11. 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.

DMARE — DMA Receive Enable Bit

CAUTION: *The DMA module is not included on this MCU. Writing a logic 1 to DMARE or DMATE may adversely affect MCU performance.*

1 = DMA not enabled to service SCI receiver DMA service requests generated by the SCRF bit (SCI receiver CPU interrupt requests enabled)
0 = DMA not enabled to service SCI receiver DMA service requests generated by the SCRF bit (SCI receiver CPU interrupt requests enabled)

DMATE — DMA Transfer Enable Bit

CAUTION: *The DMA module is not included on this MCU. Writing a logic 1 to DMARE or DMATE may adversely affect MCU performance.*

1 = SCTE DMA service requests enabled; SCTE CPU interrupt requests disabled
0 = SCTE DMA service requests disabled; SCTE CPU interrupt requests enabled

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. (See **12.9.4 SCI Status Register 1**.) Reset clears PEIE.

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

12.9.4 SCI Status Register 1

SCI status register 1 (SCS1) contains flags to signal these 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: $005D

	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 12-12. 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 automatically cleared 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, 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 logic 1s appear on the receiver input. IDLE generates an SCI error 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 receive a valid character that sets the SCRF bit before an idle condition can set the IDLE bit. Also, after the IDLE bit has been cleared, a valid character must again set the SCRF bit before an idle condition can set the IDLE bit. Reset clears the IDLE bit.

1 = Receiver input idle
0 = Receiver input active (or idle since the IDLE bit was cleared)

OR — Receiver Overrun Bit

This clearable, read-only bit is set when software fails to read the SCDR before the receive shift register receives the next character. The OR bit generates an SCI error CPU interrupt request if the ORIE

bit in SCC3 is also set. The data in the shift register is lost, but the data already in the SCDR is not affected. Clear the OR bit by reading SCS1 with OR set and then reading the SCDR. Reset clears the OR bit.

1 = Receive shift register full and SCRF = 1
0 = No receiver overrun

Software latency may allow an overrun to occur between reads of SCS1 and SCDR in the flag-clearing sequence. **Figure 12-13** shows the normal flag-clearing sequence and an example of an overrun caused by a delayed flag-clearing sequence. The delayed read of SCDR does not clear the OR bit because OR was not set when SCS1 was read. Byte 2 caused the overrun and is lost. The next flag-clearing sequence reads byte 3 in the SCDR instead of byte 2.

In applications that are subject to software latency or in which it is important to know which byte is lost due to an overrun, the flag-clearing routine can check the OR bit in a second read of SCS1 after reading the data register.

NF — Receiver Noise Flag Bit

This clearable, read-only bit is set when the SCI detects noise on the RxD pin. NF generates an NF CPU interrupt request if the NEIE bit in SCC3 is also set. Clear the NF bit by reading SCS1 and then reading the SCDR. 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 SCC3 also is set. Clear the FE bit by reading SCS1 with FE set and then reading the SCDR. Reset clears the FE bit.

1 = Framing error detected
0 = No framing error detected

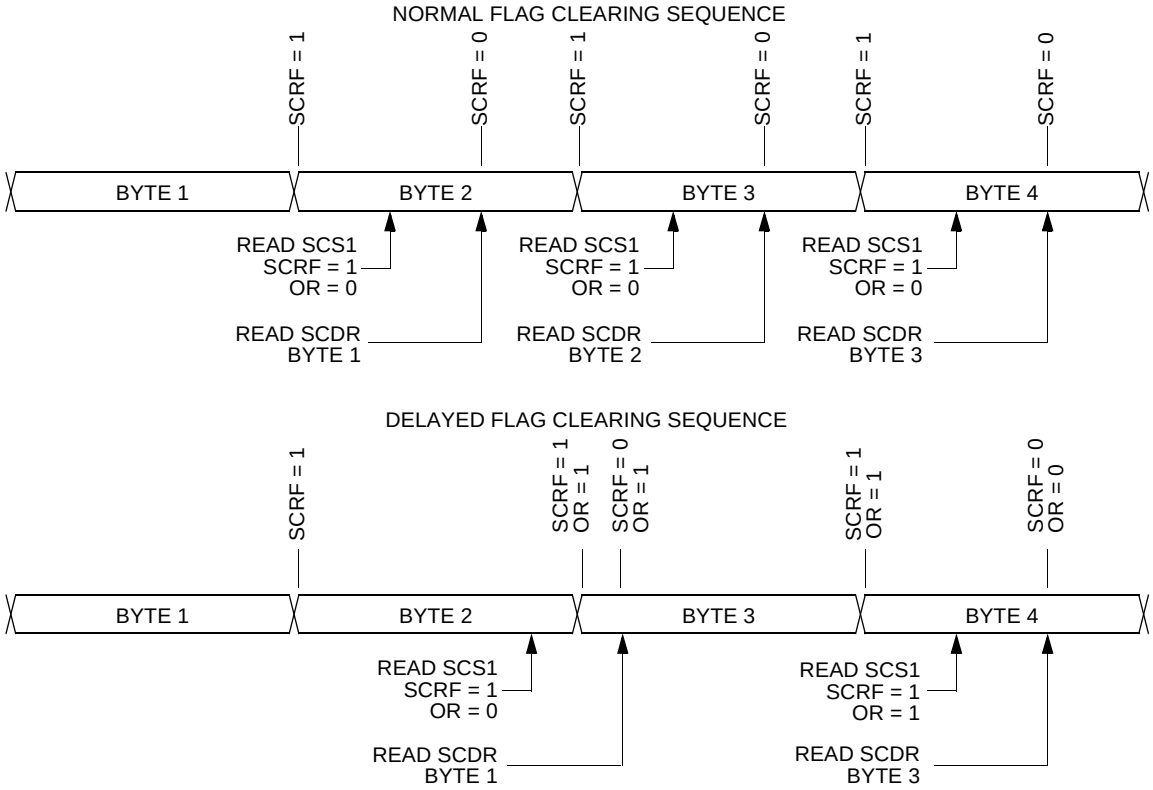


Figure 12-13. Flag Clearing Sequence

PE — Receiver Parity Error Bit

This clearable, read-only bit is set when the SCI detects a parity error in incoming data. PE generates a PE CPU interrupt request if the PEIE bit in SCC3 is also set. Clear the PE bit by reading SCS1 with PE set and then reading the SCDR. Reset clears the PE bit.

1 = Parity error detected

0 = No parity error detected

12.9.5 SCI Status Register 2

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

- Break character detected
- Incoming data

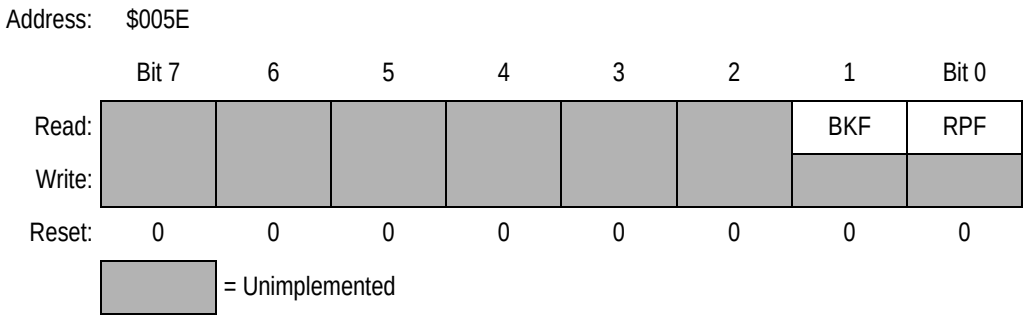


Figure 12-14. SCI Status Register 2 (SCS2)

BKF — Break Flag Bit

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

1 = Break character detected
0 = No break character detected

RPF — Reception in Progress Flag Bit

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

1 = Reception in progress
0 = No reception in progress

12.9.6 SCI Data Register

The SCI data register (SCDR) is the buffer between the internal data bus and the receive and transmit shift registers.

Address: $005F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R7	R6	R5	R4	R3	R2	R1	R0
Write:	T7	T6	T5	T4	T3	T2	T1	T0
Reset:	Unaffected by reset							

Figure 12-15. SCI Data Register (SCDR)

R7/T7–R0/T0 — Receive/Transmit Data Bits

Reading the SCI data register accesses the read-only received data bits, R7:R0. Writing to the SCI data register writes the data to be transmitted, T7:T0. Reset has no effect on the SCI data register.

NOTE: *Do not use read/modify/write instructions on the SCI data register.*

12.9.7 SCI Baud Rate Register

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

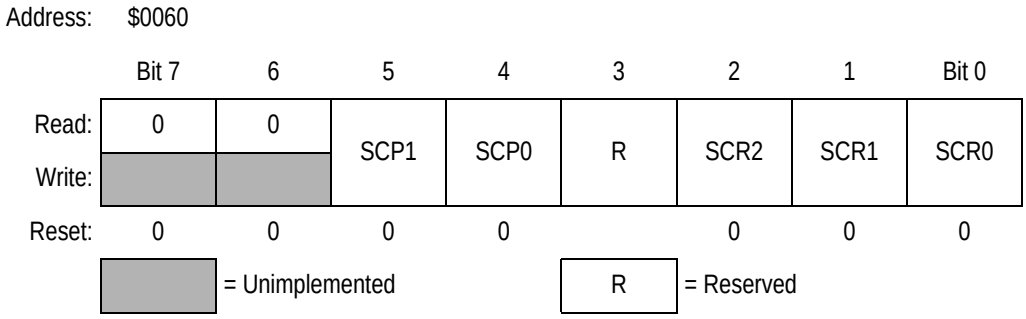


Figure 12-16. SCI Baud Rate Register (SCBR)

SCP1 and SCP0 — SCI Baud Rate Prescaler Bits

These read/write bits select the baud rate prescaler divisor as shown in **Table 12-6**. Reset clears SCP1 and SCP0.

Table 12-6. SCI Baud Rate Prescaling

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

SCR2–SCR0 — SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate divisor as shown in **Table 12-7**. Reset clears SCR2–SCR0.

Table 12-7. SCI Baud Rate Selection

SCR2, SCR1, and SCR0	Baud Rate Divisor (BD)
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

Use this formula to calculate the SCI baud rate:

$$\text{baud rate} = \frac{\text{SCI clock source}}{48 \times PD \times BD}$$

where:

SCI clock source = OSCDCLK
PD = prescaler divisor
BD = baud rate divisor

Table 12-8 shows the SCI baud rates that can be generated with a 24 MHz OSCDCLK (OSCXCLK = 12 MHz) as SCI clock source.

Table 12-8. SCI Baud Rate Selection Examples

SCP1 and SCP0	Prescaler Divisor (PD)	SCR2, SCR1, and SCR0	Baud Rate Divisor (BD)	Baud Rate (OSCDCLK=24MHz)
00	1	000	1	Baud rate settings not recommended
00	1	001	2	
00	1	010	4	
00	1	011	8	
00	1	100	16	
00	1	101	32	
00	1	110	64	
00	1	111	128	
01	3	000	1	
01	3	001	2	
01	3	010	4	
01	3	011	8	
01	3	100	16	
01	3	101	32	
01	3	110	64	
01	3	111	128	
10	4	000	1	
10	4	001	2	
10	4	010	4	
10	4	011	8	
10	4	100	16	
10	4	101	32	
10	4	110	64	
10	4	111	128	
11	13	000	1	38461.54
11	13	001	2	19230.77
11	13	010	4	9615.38
11	13	011	8	4807.69
11	13	100	16	2403.85
11	13	101	32	1201.92
11	13	110	64	600.96
11	13	111	128	300.48

Section 13. Clock Generator Module (CGM)

13.1 Contents

13.2 Introduction

This section describes the clock generation module (CGM). The CGM operates at the frequency of the crystal, OSCXCLK, and generates frequencies in the 27 MHz range. This frequency range is targeted for RF applications, such as in a local oscillator in a down conversion mixer receiver.

This particular MCU has two clock generation modules which are denoted as CGM1 and CGM2. Each CGM contains all the functional blocks for PLL control of a VCO.

NOTE: *References to either CGM1 or CGM2 may be made in the following text by omitting the CGM number. For example, CGMOUT may refer generically to CGMOUT1 and CGMOUT2, and LOCK bit may refer to LOCK1 and LOCK2 bits.*

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0051	PLL Bandwidth Control Register (PBWC)	Read:	R	LOCK1	R	PLLON1	R	LOCK2	R	PLLON2
		Write:								
		Reset:		0		0		0		0
$0052	VCO Control Register (PVCR)	Read:	VCO_7	VCO_6	VCO_5	VCO_4	VCO_3	VCO_2	VCO_1	VCO_0
		Write:								
		Reset:	0	0	1	1	0	0	0	0
$0053	PLL1 N & R Divider Select Register High (PNRH1)	Read:	VDS1_11	VDS1_10	VDS1_9	VDS1_8	0	0	RDS1_9	RDS1_8
		Write:								
		Reset:	0	0	1	0	0	0	0	0

Figure 13-1. CGM I/O Register Summary

$0054	PLL1 N Divider Select Register Low (PNSL1)	Read:	VDS1_7	VDS1_6	VDS1_5	VDS1_4	VDS1_3	VDS1_2	VDS1_1	VDS1_0
		Write:								
		Reset:	0	1	1	1	1	1	0	1
$0055	PLL1 R Divider Select Register Low (PRSL1)	Read:	RDS1_7	RDS1_6	RDS1_5	RDS1_4	RDS1_3	RDS1_2	RDS1_1	RDS1_0
		Write:								
		Reset:	1	0	0	1	0	0	0	0
$0056	PLL2 N & R Divider Select Register High (PNRH2)	Read:	VDS2_11	VDS2_10	VDS2_9	VDS2_8	0	0	RDS2_9	RDS2_8
		Write:								
		Reset:	0	0	1	0	0	0	0	0
$0057	PLL2 N Divider Select Register Low (PNSL1)	Read:	VDS2_7	VDS2_6	VDS2_5	VDS2_4	VDS2_3	VDS2_2	VDS2_1	VDS2_0
		Write:								
		Reset:	0	1	1	1	1	1	0	1
$0058	PLL2 R Divider Select Register Low (PRSL2)	Read:	RDS2_7	RDS2_6	RDS2_5	RDS2_4	RDS2_3	RDS2_2	RDS2_1	RDS2_0
		Write:								
		Reset:	1	0	0	1	0	0	0	0
$0059	Phase Detector Control Register (PDCR)	Read:	PHD_7	PHD_6	PHD_5	PHD_4	PHD_3	PHD_2	PHD_1	PHD_0
		Write:								
		Reset:	1	0	0	1	0	0	0	0

Figure 13-1. CGM I/O Register Summary

13.3 Functional Description

Figure 13-2 shows the structure of one CGM. There are two CGMs in this MCU.

The two CGMs are independently programmable, with their respective outputs at the CGMOUT1 and CGMOUT2 pins.

The following paragraphs describes the CGM circuit blocks and internal signals.

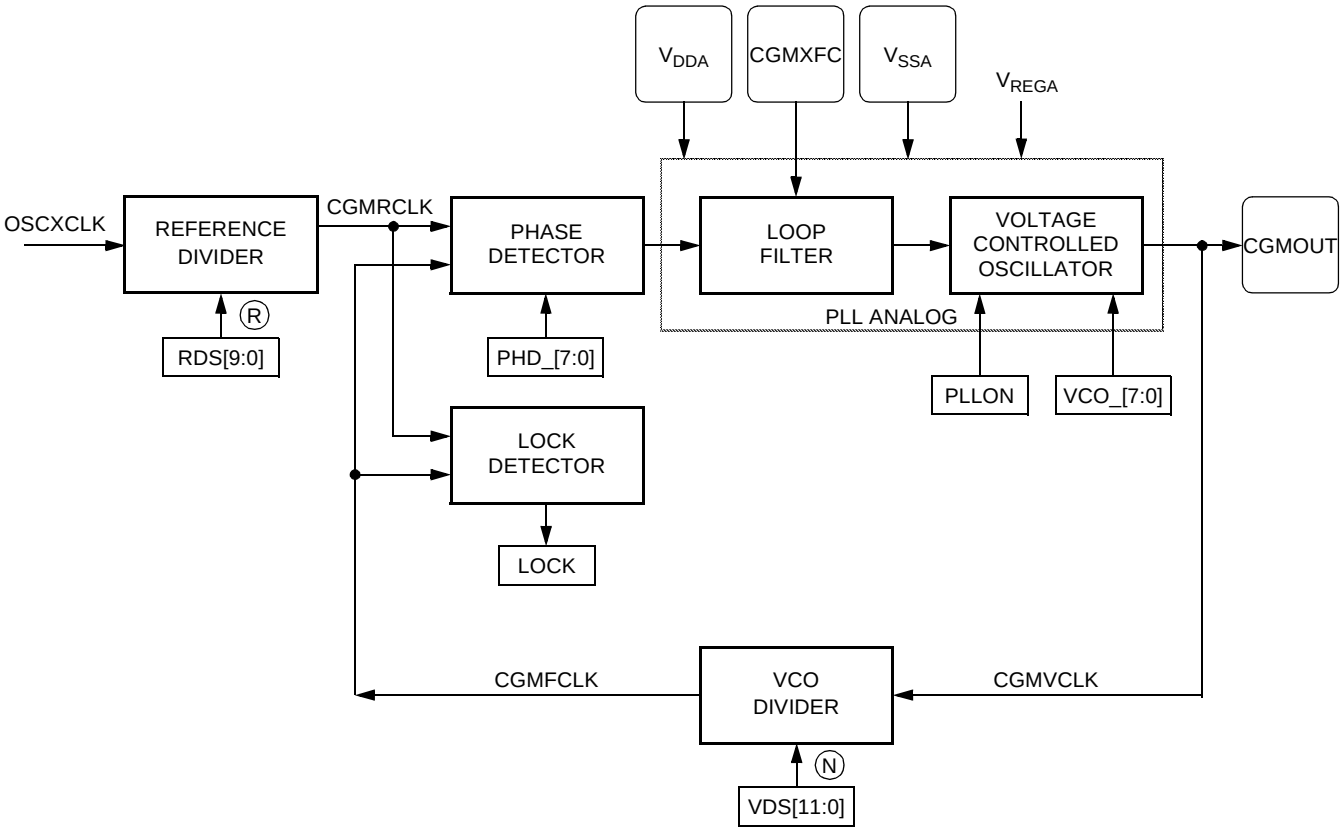


Figure 13-2. CGM Block Diagram

13.3.1 Reference Frequency Source (OSCXCLK)

The OSCXCLK signal is a buffered output of the crystal oscillator circuit and runs at a rate equal to the crystal frequency. OSCXCLK is used as the reference frequency source for both CGM modules.

13.3.2 Voltage Controlled Oscillator

The VCO clock frequency (CGMVCLK) is generated internally and it is proportional to the controlled voltage setting by the phase detector output. The VCO operating range is programmable for a wide range of frequencies and for maximum immunity to external noise. The CGMVCLK signal is also the CGM output signal, CGMOUT.

13.3.3 Reference Divider

The crystal oscillator frequency (OSCXCLK) is fed to the phase detector through a 10-bit programmable divider module R. The divider output (CGMRCLK) is equal to CGMXCLK divided by R and is used as the final reference signal for the phase detector.

13.3.4 VCO Frequency Divider

The VCO output clock (CGMVCLK) is fed to the phase detector through another 12-bit programmable divider module N. The divider output (CGMFCLK) is equal to CGMVCLK divided by N and it is the feedback signal for the phase detector.

13.3.5 Phase Detector

The phase detector compares the VCO feedback clock with the final reference clock. A correction pulse is generated based on the phase difference between the two signals. The loop filter then slightly alters the DC voltage on the external capacitor connected to pin CGMXFC base on the width and direction of the correction pulse.

13.3.6 Phase Detector Filter

The loop filter controls the dynamic characteristics of the PLL. The loop filter can make fast or low corrections depending on whether the phase detector is unlocked or stable.

13.3.7 Lock Detector

The lock detector compares the frequencies of the VCO feedback clock, CGMFCLK, and the final reference clock, CGMRCLK. Therefore, the speed of the lock detector is directly proportional to the final reference clock, CGMRCLK.

13.4 I/O Signals

The following paragraphs describe the CGM I/O signals.

13.4.1 CGM Power Supply Pins (V_{DDA}, V_{SSA0}, V_{SSA1})

V_{DDA} is the power supply pin, V_{SSA0} and V_{SSA1} are the ground pins for the analog portions of both CGMs.

13.4.2 CGM1 Voltage Regulator Out (V_{REGA0})

3.3V output of the on-chip voltage regulator for analog portions. Connect V_{REGA0} directly to V_{REGA1} as shown in **Figure 13-3**.

13.4.3 CGM2 Voltage Regulator In (V_{REGA1})

3.3V input for CGM2 analog portions. Connect V_{REGA0} directly to V_{REGA1} as shown in **Figure 13-3**.

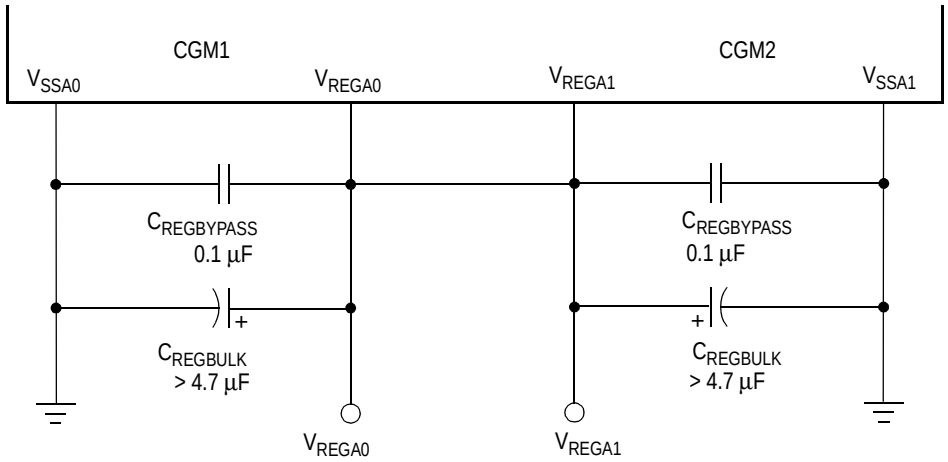


Figure 13-3. CGM Power Supply Connection

13.4.4 External Filter Capacitor Pins (CGMXFC1, CGMXFC2)

The CGMXFC1 and CGMXFC2 pins are required by the loop filter to filter out phase corrections for each PLL. An external filter network is connected to each pin. (See **13.5 CGMXFC External Connections**.)

13.4.5 CGM Clock Output Pins (CGMOUT1, CGMOUT2)

CGMOUT1 and CGMOUT2 are VCO output signals. The output signals are buffered through logic stages to output pins without degrading the loop performance.

13.5 CGMXFC External Connections

The external filter network is critical to the stability and reaction time of the PLL. The configurations shown in **Figure 13-4** (a) and (b) are recommended for connection to CGMXFC1 and CGMXFC2.

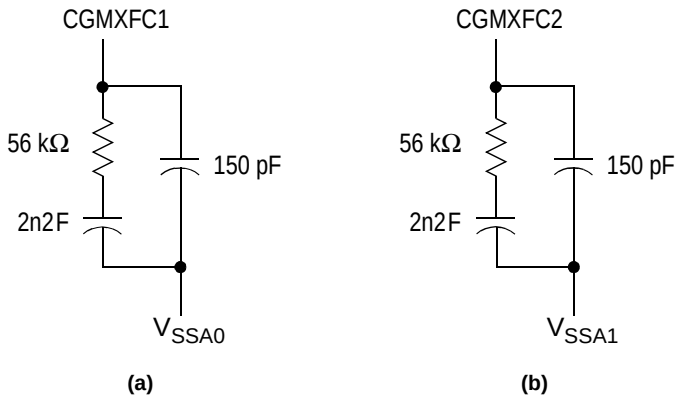


Figure 13-4. CGMXFC External Connections

13.6 CGMOUT External Connections

The output of CGM clock is a standard CMOS output with push-pull configuration. The output logic high and low levels are specified with corresponding DC loading current (see **20.13 CGM Electrical Characteristics**). The transient current is mainly determined by the maximum loading capacitor value.

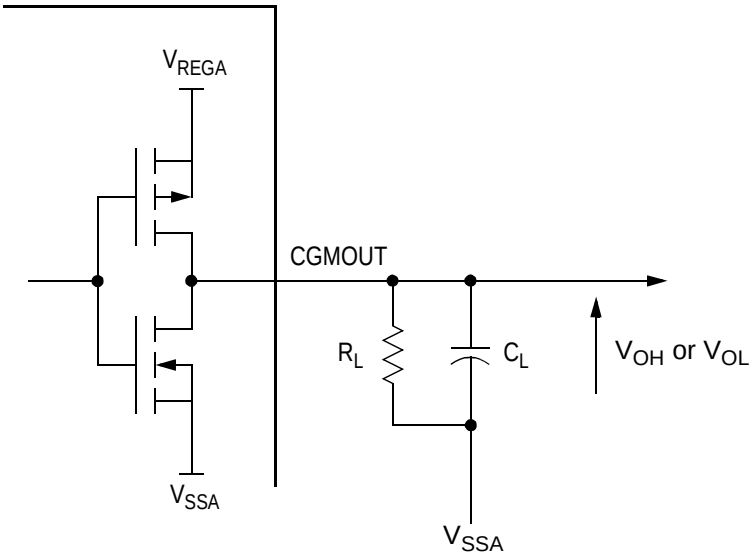


Figure 13-5. CGMOUT External Connections

13.7 Calculation of VCO Frequency

The relationship between the VCO frequency, f_{VCLK}, and the crystal reference frequency, f_{XCLK}, is:

$$f_{VCLK} = \frac{N}{R} \times f_{XCLK}$$

13.8 Programming the PLL

With the PLLs off (PLLON = 0), use the following procedure to initialize both PLLs:

1. Write $80 to the VCO control register (PVCR).
2. Write $70 to the phase detector register (PDCR).

Then for each PLL, use the following procedure to program the VCO and reference dividers:

3. Write data to the VCO and reference divider select register high (PNRH).
4. Write data to the VCO divider select register low (PNSL).
5. Write data to the reference divider select register low (PRSL).
6. Set PLLON = 1 in the bandwidth control register to enable the PLL.

To reprogram the PLL frequency, clear the PLLON bit (PLLON =0) and repeat steps 3 to 6.

NOTE: *Do not program both PLLs to the same frequency. A difference of 50kHz or more is recommended between the two PLL outputs.*

13.9 CGM I/O Registers

These registers control and monitor operation of the CGMs:

- Bandwidth control register (PBWC)
- VCO control register (PVCR)
- PLL1 VCO and reference divider select register high (PNRH1)
- PLL1 VCO divider select register low (PNSL1)
- PLL1 reference divider select register low (PRSL1)
- PLL2 VCO and reference divider select register high (PNRH2)
- PLL2 VCO divider select register low (PNSL2)
- PLL2 reference divider select register low (PRSL2)
- Phase detector control register (PDCR)

13.9.1 Bandwidth Control Register

The bandwidth control register (PBWC) contains control/status bits for both PLLs.

Address: $0051

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	LOCK1	R	PLLON1	R	LOCK2	R	PLLON2
Write:								
Reset:		0		0		0		0

= Unimplemented R = Reserved

Figure 13-6. PLL Bandwidth Control Register (PBCR)

LOCKx — Lock Indicator Bit

This read-only bit becomes set when the VCO clock is locked (running at the programmed frequency).

1 = VCO frequency correct or locked
0 = VCO frequency incorrect or unlocked

PLLONx — PLL On Bit

This read/write bit activates each PLL and enables the VCO clock.

1 = PLL on
0 = PLL off

13.9.2 VCO Control Register (PVCR)

The VCO control register configures the VCO for both PLLs.

Address: $0052

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	VCO_7	VCO_6	VCO_5	VCO_4	VCO_3	VCO_2	VCO_1	VCO_0
Write:								
Reset:	0	0	1	1	0	0	0	0

Figure 13-7. VCO Control Register (PVCR)

VCO_[7:0] — VCO Control Bits for both PLLs

Set VCO_[7:0] = $80 for maximum performance.

13.9.3 VCO and Reference Divider Select Registers High

The VCO and reference divider select registers high (PNRH1 and PNRH2) contain the programming information for the high byte of VCO feedback divider, N, and reference divider, R.

Address: $0053

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	VDS1_11	VDS1_10	VDS1_9	VDS1_8	0	0	RDS1_9	RDS1_8
Write:								
Reset:	0	0	1	0	0	0	0	0

= Unimplemented

Figure 13-8. PLL1 N & R Divider Select Register High (PNRH1)

Address: $0056

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	VDS2_11	VDS2_10	VDS2_9	VDS2_8	0	0	RDS2_9	RDS2_8
Write:								
Reset:	0	0	1	0	0	0	0	0

= Unimplemented

Figure 13-9. PLL2 N & R Divider Select Register High (PNRH2)

VDSx_[11:8] — VCO Divider Select Bits

These read/write bits control the high byte of the VCO feedback divider, N.

RDSx_[9:8] — Reference Divider Select Bits

These read/write bits control the high byte of the reference divider, R.

NOTE: *The VDSx_[11:8] and RDSx_[9:8] bits are not latched until the respective low bytes are written.*

13.9.4 VCO Divider Select Register Low

The VCO divider select registers low (PNSL1 and PNSL2) contain the programming information for the low byte of VCO feedback divider, N.

Address: $0054

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	VDS1_7	VDS1_6	VDS1_5	VDS1_4	VDS1_3	VDS1_2	VDS1_1	VDS1_0
Reset:	0	1	1	1	1	1	0	1

Figure 13-10. PLL1 N Divider Select Register Low (PNSL1)

Address: $0057

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	VDS2_7	VDS2_6	VDS2_5	VDS2_4	VDS2_3	VDS2_2	VDS2_1	VDS2_0
Reset:	0	1	1	1	1	1	0	1

Figure 13-11. PLL2 N Divider Select Register Low (PNSL2)

VDSx_[7:0] — VCO Divider Select Bits

These read/write bits control the low byte of the VCO feedback divider, N.

NOTE: *Writing to PNSL also latches the respective high bits, VDSx_[11:8].*

13.9.5 Reference Divider Select Register Low

The divider select registers low (PRSL1 and PRLS2) contain the programming information for the low byte of reference divider, R.

Address: $0055

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	RDS1_7	RDS1_6	RDS1_5	RDS1_4	RDS1_3	RDS1_2	RDS1_1	RDS1_0
Reset:	1	0	0	1	0	0	0	0

Figure 13-12. PLL1 R Divider Select Register Low (PRSL1)

Address: $0058

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	RDS2_7	RDS2_6	RDS2_5	RDS2_4	RDS2_3	RDS2_2	RDS2_1	RDS2_0
Reset:	1	0	0	1	0	0	0	0

Figure 13-13. PLL2 R Divider Select Register Low (PRSL2)

RDSx_[7:0] — Reference Divider Select Bits

These read/write bits control the high byte of the reference divider, R.

NOTE: *Writing to PRSL also latches the respective high bits, RDSx_[9:8].*

13.9.6 Phase Detector Control Register (PDCR)

The phase detector control register configures the phase detector for both PLLs.

Address: $0059

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PHD_7	PHD_6	PHD_5	PHD_4	PHD_3	PHD_2	PHD_1	PHD_0
Reset:	1	0	0	1	0	0	0	0

Figure 13-14. Phase Detector Control Register (PDCR)

PHD_[7:0] — Phase detector Control Bits for both PLLs

Set PHD_[7:0] = $70 for maximum performance.

13.10 Pre-Defined VCO Output Frequency Settings

The exact frequency values for the following required channels cannot be synthesized by using a reference frequency higher than 10kHz. An absolute offset frequency from +1.66kHz to +1.89kHz will be introduced for different channels and the maximum relative offset is only ±115Hz with 1.775kHz as the center point (see **Table 13-1 . Predefined Programming Setting for PLL**). The absolute offset frequency can be further minimized by reducing the crystal frequency by 60 ppm (360Hz) in actual application.

Table 13-1. Predefined Programming Setting for PLL

Channel Frequency (MHz)	Crystal Frequency (MHz)	Divider R	Reference Frequency (kHz)	Divider N	VCO Frequency (MHz)	Absolute Offset (kHz)
26.54	12	288	41.67	637	26.54166	+1.66
26.59	12	338	35.50	749	26.59171	+1.71
26.64	12	268	44.78	595	26.64179	+1.79
26.69	12	370	32.43	823	26.69189	+1.89
26.74	12	302	39.74	673	26.74172	+1.72

13.11 Low-Power Modes

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

13.11.1 Wait Mode

The CGMs remain active and all PLL registers are not affected in wait mode. If CGM functions are not required in wait mode, it can be disabled by the PLLONx bit in the bandwidth control register (PBWC).

13.11.2 Stop Mode

The CGM is inactive and all PLL registers are not affected in stop mode. CGM operation resumes when the MCU exits stop mode.

Clock Generator Module (CGM)

Section 14. Input/Output (I/O) Ports

14.1 Contents

14.2 Introduction

Twenty-one (21) bidirectional input-output (I/O) pins form four parallel ports. All I/O pins are programmable as inputs or outputs.

NOTE: *Connect any unused I/O pins to an appropriate logic level, either V_{DD} or V_{SS}. Although the I/O ports do not require termination for proper operation, termination reduces excess current consumption and the possibility of electrostatic damage.*

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA)	Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Reset:	Unaffected by reset							
$0001	Reserved	Read: Write:	R	R	R	R	R	R	R	R
		Reset:								
$0002	Port C Data Register (PTC)	Read:	0	0	0	0	0	0	PTC1	PTC0
		Write:								
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PTD)	Read:	0	0	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Write:								
		Reset:	Unaffected by reset							
$0004	Data Direction Register A (DDRA)	Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Reset:	0*	0	0	0	0	0	0	0
$0005	Reserved	Read: Write:	R	R	R	R	R	R	R	R
		Reset:								
$0006	Data Direction Register C (DDRC)	Read:	0	0	0	0	0	0	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007	Data Direction Register D (DDRD)	Read:	0	0	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0008	Port E Data Register (PTE)	Read:	0	0	0	PTE4	PTE3	PTE2	PTE1	PTE0
		Write:								
		Reset:	Unaffected by reset							

* DDRA7 bit is reset by POR or LVI reset only.

= Unimplemented R = Reserved

Figure 14-1. I/O Port Register Summary

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0009	Data Direction Register E (DDRE)	Read:	0	0	0	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001D	Port Option Control Register (POCR)	Read:	PTE20P	PTDLDD	PTDILDD	PTE4P	PTE3P	PCP	R	PAP
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved

Figure 14-1. I/O Port Register Summary

Table 14-1. Port Control Register Bits Summary

Port	Bit	DDR	Module Control			Pin
			Module	Register	Control Bit	
A	0	DDRA0	KBI	KBIER $0017	KBIE0	PTA0/$\overline{\text{KBA0}}$
	1	DDRA1			KBIE1	PTA1/$\overline{\text{KBA1}}$
	2	DDRA2			KBIE2	PTA2/$\overline{\text{KBA2}}$
	3	DDRA3			KBIE3	PTA3/$\overline{\text{KBA3}}$
	4	DDRA4			KBIE4	PTA4/$\overline{\text{KBA4}}$
	5	DDRA5			KBIE5	PTA5/$\overline{\text{KBA5}}$
	6	DDRA6			KBIE6	PTA6/$\overline{\text{KBA6}}$
	7	DDRA7			KBIE7	PTA7/$\overline{\text{KBA7}}$
C	0	DDRC0	SCI	SCC1 $005A	ENSCI	PTC0/TxD
	1	DDRC1				PTC1/RxD
D	0–5	DDRD[0:5]	—	—	—	PTD0–PTD5
E	0	DDRE0	TIM1 or TIM2	T1SC $000A or T2SC $0040	PS[2:0]	PTE0/TCLK
	1	DDRE1	TIM1	T1SC0 $0010 or T1SC1 $0013	ELS0B:ELS0A or ELS1B:ELS1A	PTE1/T1CH01
	2	DDRE2	TIM2	T2SC0 $0046 or T2SC1 $0049	ELS0B:ELS0A or ELS1B:ELS1A	PTE2/T2CH01
	3	DDRE3	USB	UADDR $0038	USBEN	PTE3/D+
	4	DDRE4				PTE4/D–

14.3 Port A

Port A is an 8-bit general-purpose bidirectional I/O port with software configurable pullups, and shares its pins with the keyboard interrupt module (KBI).

14.3.1 Port A Data Register

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

Address: $0000

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Reset:	Unaffected by reset							
Alternative Function:	$\overline{\text{KBA7}}$	$\overline{\text{KBA6}}$	$\overline{\text{KBA5}}$	$\overline{\text{KBA4}}$	$\overline{\text{KBA3}}$	$\overline{\text{KBA2}}$	$\overline{\text{KBA1}}$	$\overline{\text{KBA0}}$
Additional Function:	Optional pullup	Optional pullup	Optional pullup	Optional pullup	Optional pullup	Optional pullup	Optional pullup	Optional pullup

Figure 14-2. Port A Data Register (PTA)

PTA[7:0] — Port A Data Bits

These read/write bits are software programmable. Data direction of each port A pin is under the control of the corresponding bit in data direction register A. Reset has no effect on port A data.

The port A pullup control bit, PAP, in the port option control register (POCR) enables pullups on port A pins if the respective pin is configured as an input. (See **14.7 Port Options**.)

$\overline{\text{KBA7}}$–$\overline{\text{KBA0}}$ — Keyboard Interrupts

The keyboard interrupt enable bits, KBIE7–KBIE0, in the keyboard interrupt enable register (KBIER), enable the port A pins as external interrupt pins. (See **Section 16. Keyboard Interrupt Module (KBI)**.)

14.3.2 Data Direction Register A

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

Address: $0004

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset:	0*	0	0	0	0	0	0	0

* DDRA7 bit is reset by POR or LVI reset only.

Figure 14-3. Data Direction Register A (DDRA)

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 14-4 shows the port A I/O logic.

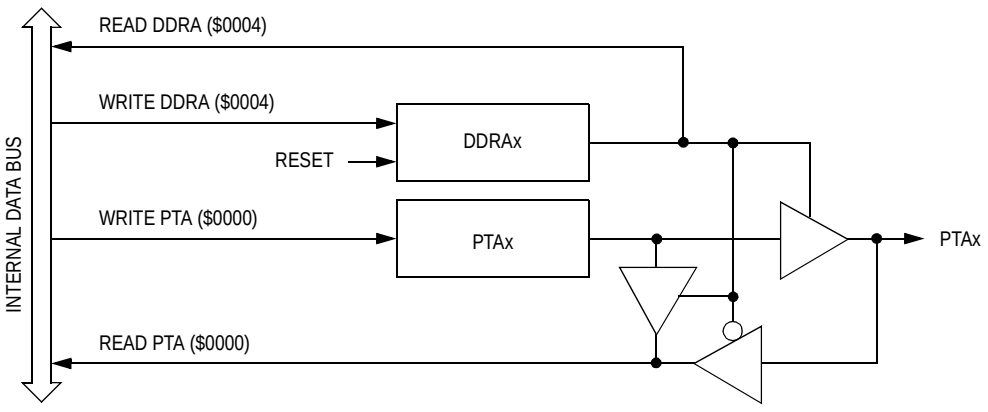


Figure 14-4. Port A I/O Circuit

When bit DDRAx is a logic 1, reading address $0000 reads the PTAx data latch. When bit 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 14-2** summarizes the operation of the port A pins.

Table 14-2. Port A Pin Functions

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

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

14.4 Port C

Port C is a 2-bit special function port that shares its pins with the serial communications interface (SCI) module. These pins have software configurable pullups.

14.4.1 Port C Data Register

The port C data register contains a data latch for each of the two port C pins.

Address: $0002

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	PTC1	PTC0
Write:								
Reset:	Unaffected by reset							
Alternative Function:							RxD	TxD
Additional Function:							Optional pullup	Optional pullup

Figure 14-5. Port C Data Register (PTC)

PTC[1:0] — Port C Data Bits

These read/write bits are software-programmable. Data direction of each port C pin is under the control of the corresponding bit in data direction register C. Reset has no effect on port C data.

The port C pullup enable bit, PCP, in the port option control register (POCR) enables pullups on PTC[1:0] if the respective pin is configured as an input. (See **14.7 Port Options**.)

TxD, RxD — SCI Data I/O Pins

The TxD and RxD pins are the transmit data output and receive data input for the SCI module. The SCI enable bit, ENSCI, in the SCI control register 1 enables the PTC0/TxD and PTC1/RxD pins as SCI TxD and RxD pins and overrides any control from the port I/O. See **Section 12. Serial Communications Interface Module (SCI)**.

14.4.2 Data Direction Register C

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

Address: $0006

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	DDRC1	DDRC0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 14-6. Data Direction Register C (DDRC)

DDRC[1:0] — Data Direction Register C Bits

These read/write bits control port C data direction. Reset clears DDRC[1:0], configuring all port C pins as inputs.

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

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

Figure 14-7 shows the port C I/O logic.

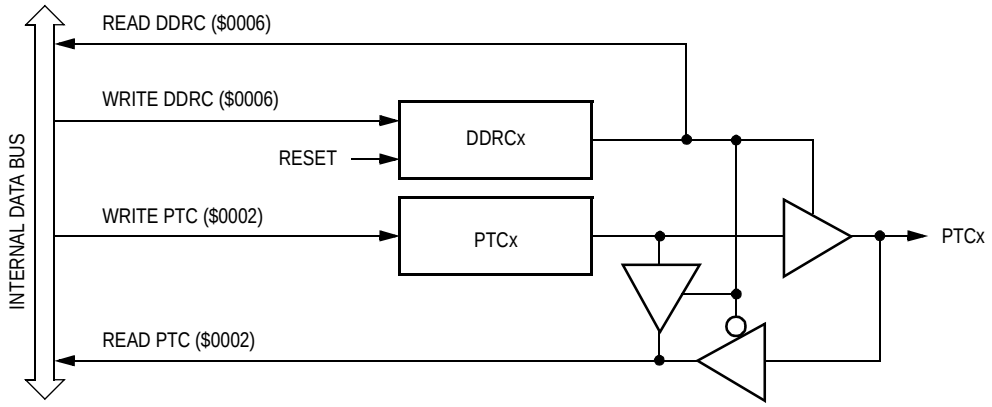


Figure 14-7. Port C I/O Circuit

When bit DDRCx is a logic 1, reading address $0002 reads the PTCx data latch. When bit DDRCx is a logic 0, reading address $0002 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. **Table 14-3** summarizes the operation of the port C pins.

Table 14-3. Port C Pin Functions

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

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

14.5 Port D

Port D is an 6-bit general-purpose bidirectional I/O port. These pins are open-drain when configured as output.

14.5.1 Port D Data Register

The port D data register contains a data latch for each of the six port D pins.

NOTE: *Bits 5–1 of PTD are not available in the 32-pin low-profile quad flat pack.*

Address: $0003

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
Write:								
Reset:	Unaffected by reset							
Additional Function:			10mA sink	10mA sink	10mA sink	10mA sink	25mA sink	25mA sink

Figure 14-8. Port D Data Register (PTD)

PTD[5:0] — Port D Data Bits

These read/write bits are software programmable. Data direction of each port D pin is under control of the corresponding bit in data direction register D. Reset has no effect on port D data.

The LED direct drive bit, PTDLDD, in the port option control register (POCR) controls the drive options for PTD5–PTD2 pins. The infrared LED drive bit, PTDILDD, in the POCR controls the drive options for PTD1–PTD0 pins. (See **14.7 Port Options**.)

14.5.2 Data Direction Register D

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

Address: $0007

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 14-9. Data Direction Register D (DDRD)

DDRD[5:0] — Data Direction Register D Bits

These read/write bits control port D data direction. Reset clears DDRD[5:0], configuring all port D pins as inputs.

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

Port D pins are open-drain when configured as output.

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

NOTE: *For those devices packaged in a 32-pin low-profile quad flat pack, PTD5–1 are not connected. DDRD5–1 should be set to a 1 to configure PTD5–1 as outputs.*

Figure 14-10 shows the port D I/O circuit logic.

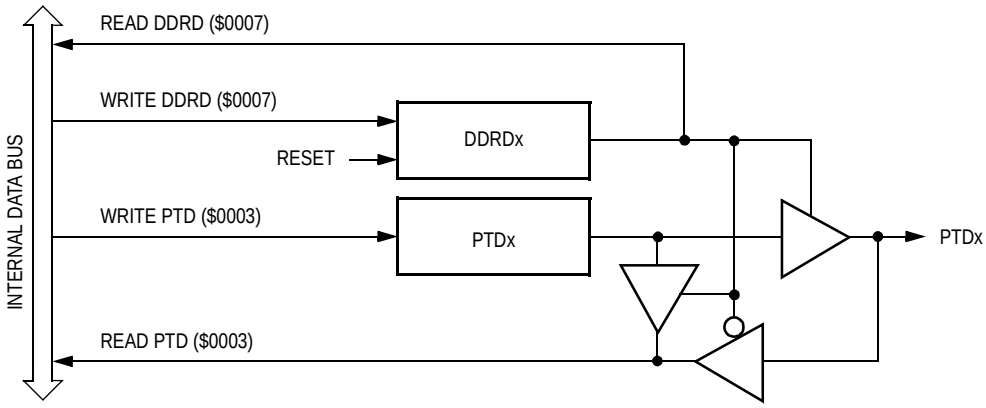


Figure 14-10. Port D I/O Circuit

When bit DDRDx is a logic 1, reading address $0003 reads the PTDx data latch. When bit DDRDx is a logic 0, reading address $0003 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. **Table 14-4** summarizes the operation of the port D pins.

Table 14-4. Port D Pin Functions

DDRD Bit	PTD Bit	I/O Pin Mode	Accesses to DDRD	Accesses to PTD	
			Read/Write	Read	Write
0	X(1)	Input, Hi-Z(2)	DDRD[5:0]	Pin	PTD[5:0](3)
1	X	Output	DDRD[5:0]	PTD[5:0]	PTD[5:0]

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

14.6 Port E

Port E is a 5-bit special function port that shares three of its pins with the timer interface modules (TIMs) and two of its pins with the USB data pins D+ and D–. PTE4 and PTE3 are open-drain when configured as output.

14.6.1 Port E Data Register

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

Address: $0008

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	PTE4	PTE3	PTE2	PTE1	PTE0
Write:								
Reset:	Unaffected by reset							
Alternative Function:				D–	D+	T2CH01	T1CH01	TCLK
Additional Function:				Optional pullup	Optional pullup	Optional pullup	Optional pullup	Optional pullup
Additional Function:				External interrupt				
				Open-drain	Open-drain			

= Unimplemented

Figure 14-11. Port E Data Register (PTE)

PTE[4:0] — Port E Data Bits

PTE[4:0] 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.

The PTE4 and PTE3 pullup enable bits, PTE4P and PTE3P, in the port option control register (POCR) enable 5kΩ pullups on PTE4 and PTE3 if the respective pin is configured as an input and the USB module is disabled. (See **14.7 Port Options**.)

The PTE[2:0] pullup enable bit, PTE20P, in the port option control register (POCR) enables pullups on PTE2–PTE0, regardless of the pin is configured as an input or an output. (See **14.7 Port Options**.)

PTE4 pin functions as an external interrupt when PTE4IE=1 in the IRQ option control register (IOCR) and USBEN=0 in the USB address register (USB disabled). (See **15.9 IRQ Option Control Register**.)

D– and D+ — USB Data Pins

D– and D+ are the differential data lines used by the USB module. (See **Section 11. Universal Serial Bus Module (USB)**.)

The USB module enable bit, USBEN, in the USB address register (UADDR) controls the pin options for PTE4/D– and PTE3/D+. When the USB module is enabled, PTE4/D– and PTE3/D+ function as USB data pins D– and D+. When the USB module is disabled, PTE4/D– and PTE3/D+ function as 10mA open-drain high current pins for PS/2 clock and data use.

The pullup enable bit, PULLEN, in the USB control register 3 (UCR3) enables a 1.5kΩ pullup on D– pin when the USB module is enabled. (See **11.8.8 USB Control Register 3**.)

NOTE: *PTE4/D– pin has two programmable pullup resistors. One is used for PTE4 when the USB module is disabled and another is used for D– when the USB module is enabled.*

T2CH01 and T1CH01 — Timer Channel I/O Bits

The PTE2/T2CH01 and PTE1/T1CH01 pins are the respective TIM2 and TIM1 input capture/output compare pins. The edge/level select bits, ELSxB and ELSxA, determine whether the PTE2/T2CH01 and PTE1/T1CH01 pins are timer channel I/O pins or general-purpose I/O pins. (See **Section 10. Timer Interface Module (TIM)**.)

TCLK — Timer Clock Input

The PTE0/TCLK pin is the external clock input for TIM1 and TIM2. The prescaler select bits, PS[2:0], select PTE0/TCLK as the TIM clock input. When not selected as the TIM clock, PTE0/TCLK is available for general purpose I/O. (See **Section 10. Timer Interface Module (TIM)**.)

NOTE: *Data direction register E (DDRE) does not affect the data direction of port E pins that are being used by the TIM. However, the DDRE bits always determine whether reading port E returns the states of the latches or the states of the pins.*

14.6.2 Data Direction Register E

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

Address: $0009

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-12. Data Direction Register E (DDRE)

DDRE[4:0] — Data Direction Register E Bits

These read/write bits control port E data direction. Reset clears DDRE[4:0], configuring all port E pins as inputs.

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

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

Figure 14-13 shows the port E I/O circuit logic.

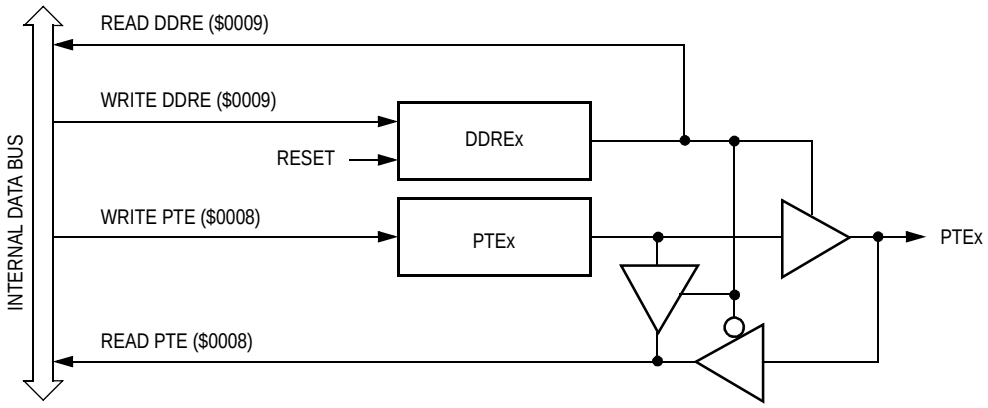


Figure 14-13. Port E I/O Circuit

When bit DDREx is a logic 1, reading address $0008 reads the PTEx data latch. When bit DDREx is a logic 0, reading address $0008 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. **Table 14-3** summarizes the operation of the port E pins.

Table 14-5. Port E Pin Functions

DDRE Bit	PTE Bit	I/O Pin Mode	Accesses to DDRE	Accesses to PTE	
			Read/Write	Read	Write
0	X(1)	Input, Hi-Z(2)	DDRE[4:0]	Pin	PTE[4:0](3)
1	X	Output	DDRE[4:0]	PTE[4:0]	PTE[4:0]

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

14.7 Port Options

All pins of port A, port C, and port E have programmable pullup resistors. Port D has programmable LED drive capability; PTD5–PTD2 each have 10 mA high current sink, and PTD1–PTD0 each have 25 mA high current sink.

14.7.1 Port Option Control Register

The port option control register controls the pullup options for port A, port C, and port E pins. It also controls the drive configuration on port D.

Address: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTE20P	PTDLDD	PTDILDD	PTE4P	PTE3P	PCP	R	PAP
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 14-14. Port Option Control Register (POCR)

PTE20P — Pins PTE[2:0] Pullup Enable

This read/write bit controls the pullup option for the PTE2–PTE0 pins, regardless whether the pins are input or output.

1 = Configure PTE2–PTE0 to have internal pullups
0 = Disconnect PTE2–PTE0 internal pullups

PTDLDD — LED Direct Drive Control

This read/write bit controls the output current capability of PTD5–PTD2 pins. When set, each port pin has 10 mA current sink limit. An LED can be connected directly between the port pin and V_{DD} without the need of a series resistor.

1 = PTD5–PTD2 configured for direct LED drive capability; when a pin is set as an output, the pin is an open-drain pin with 10 mA current sink limit
0 = PTD5–PTD2 configured as standard open-drain I/O port pins

PTDILDD — Infrared LED Drive Control

This read/write bit controls the output current capability of PTD1 and PTD0 pins. When set, each port pin has 25 mA current sink capability. An infrared LED can be connected directly between the port pin and V_{DD}.

1 = PTD1 and PTD0 configured for infrared LED drive capability; when a pin is set as an output, the pin is an open-drain pin with 25mA current sink capability
0 = PTD1 and PTD0 configured as standard open-drain I/O port pins

PTE4P — Pin PTE4 Pullup Enable

This read/write bit controls the pullup option for the PTE4 pin when the pin is configured as an input and the USB module is disabled.

1 = Configure PTE4 to have internal pullup
0 = Disconnect PTE4 internal pullup

NOTE: *When the USB module is enabled, the pullup controlled by PTE4P is disconnected; PTE4/D– pin functions as D– which has a 1.5kΩ programmable pull-up resistor. (See* ***11.8.8 USB Control Register 3****.)*

PTE3P — Pin PTE3 Pullup Enable

This read/write bit controls the pullup option for the PTE3 pin when the pin is configured as an input and the USB module is disabled.

1 = Configure PTE3 to have internal pullup
0 = Disconnect PTE3 internal pullup

PCP — Port C Pullup Enable

This read/write bit controls the pullup option for the PTC1 and PTC0 pins. When set, a pullup device is connected when a pin is configured as an input.

1 = Configure port C to have internal pullups
0 = Disconnect port C internal pullups

PAP — Port A Pullup Enable

This read/write bit controls the pullup option for the PTA7–PTA0 pins. When set, a pullup device is connected when a pin is configured as an input.

1 = Configure port A to have internal pullups
0 = Disconnect port A internal pullups

Section 15. External Interrupt (IRQ)

15.1 Contents

15.2 Introduction

The IRQ module provides two external interrupt inputs: one dedicated $\overline{\text{IRQ}}$ pin and one shared port pin, PTE4/D–.

15.3 Features

Features of the IRQ module include:

- Two external interrupt pins, $\overline{\text{IRQ}}$ and PTE4/D–
- $\overline{\text{IRQ}}$ interrupt control bits
- Hysteresis buffer
- Programmable edge-only or edge and level interrupt sensitivity
- Automatic interrupt acknowledge
- Low leakage $\overline{\text{IRQ}}$ pin for external RC wake up input
- Selectable internal pullup resistor

15.4 Functional Description

A logic 0 applied to the external interrupt pin can latch a CPU interrupt request. **Figure 15-1** shows the structure of the IRQ module.

Interrupt signals on the $\overline{\text{IRQ}}$ pin are latched into the IRQ latch. An interrupt latch remains set until one of the following actions occurs:

- Vector fetch — A vector fetch automatically generates an interrupt acknowledge signal that clears the IRQ latch.
- Software clear — Software can clear the interrupt latch by writing to the acknowledge bit in the interrupt status and control register (ISCR). Writing a logic 1 to the ACK bit clears the IRQ latch.
- Reset — A reset automatically clears the interrupt latch.

The external interrupt pin is falling-edge-triggered and is software-configurable to be either falling-edge or low-level-triggered. The MODE bit in the ISCR controls the triggering sensitivity of the $\overline{\text{IRQ}}$ pin.

When the interrupt pin is edge-triggered only, the CPU interrupt request remains set until a vector fetch, software clear, or reset occurs.

When the interrupt pin is both falling-edge and low-level-triggered, the CPU interrupt request remains set until both of the following occur:

- Vector fetch or software clear
- Return of the interrupt pin to logic one

The vector fetch or software clear may occur before or after the interrupt pin returns to logic 1. As long as the pin is low, the interrupt request remains pending. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

When set, the IMASK bit in the ISCR mask all external interrupt requests. A latched interrupt request is not presented to the interrupt priority logic unless the IMASK bit is clear.

NOTE: *The interrupt mask (I) in the condition code register (CCR) masks all interrupt requests, including external interrupt requests. (See **8.6 Exception Control**.)*

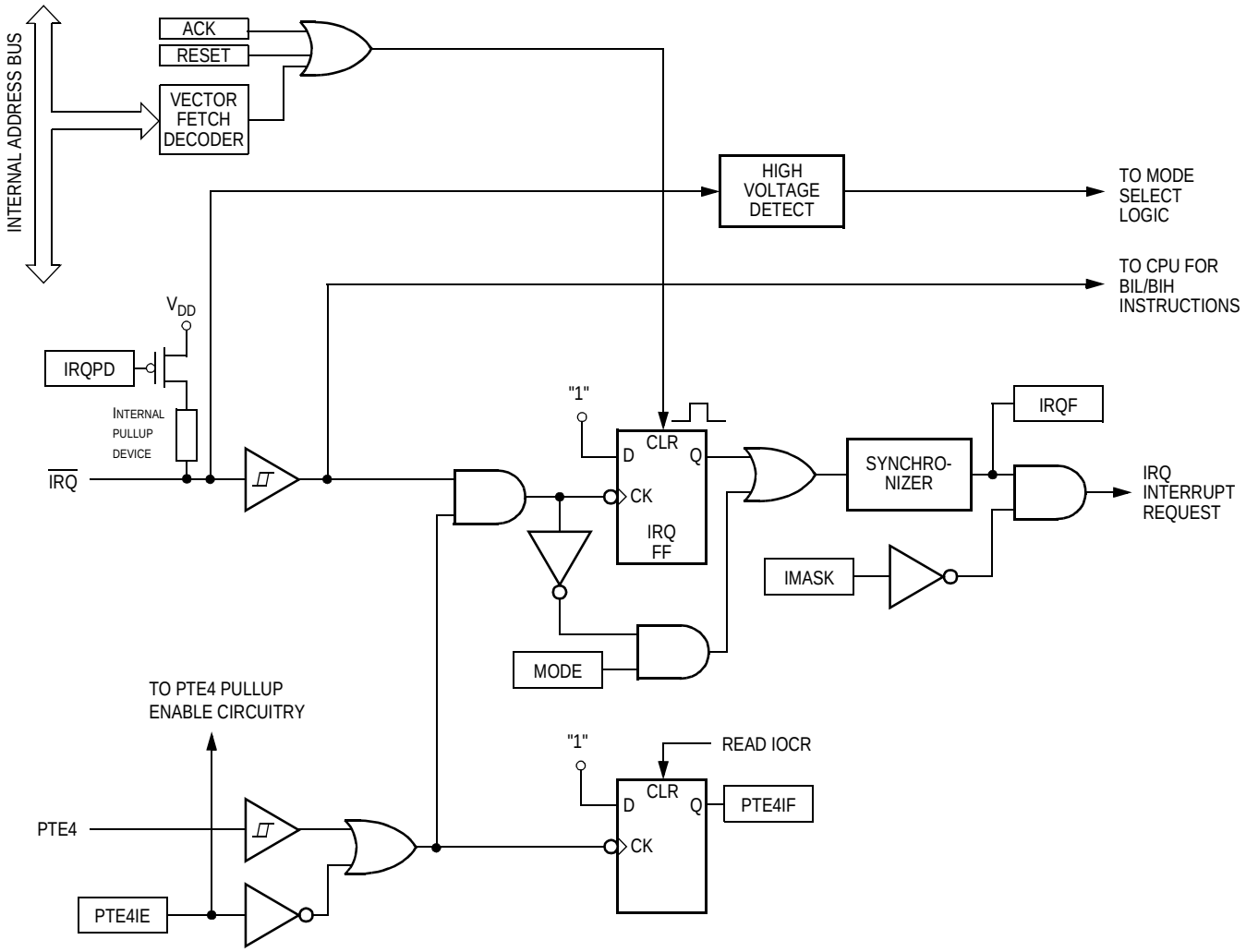


Figure 15-1. IRQ Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001C	IRQ Option Control Register (IOCR)	Read:	0	0	0	0	0	PTE4IF	PTE4IE	IRQPD
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001E	IRQ Status and Control Register (ISCR)	Read:	0	0	0	0	IRQF	0	IMASK	MODE
		Write:						ACK		
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 15-2. IRQ I/O Register Summary

15.5 $\overline{\text{IRQ}}$ Pin

The $\overline{\text{IRQ}}$ pin has a low leakage for input voltages ranging from 0V to V_{DD}; suitable for applications using RC discharge circuitry to wake up the MCU.

A logic 0 on the $\overline{\text{IRQ}}$ pin can latch an interrupt request into the IRQ latch. A vector fetch, software clear, or reset clears the IRQ latch.

If the MODE bit is set, the $\overline{\text{IRQ}}$ pin is both falling-edge-sensitive and low-level-sensitive. With MODE set, both of the following actions must occur to clear IRQ:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the latch. Software may generate the interrupt acknowledge signal by writing a logic 1 to the ACK bit in the interrupt status and control register (ISCR). The ACK bit is useful in applications that poll the $\overline{\text{IRQ}}$ pin and require software to clear the IRQ latch. Writing to the ACK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACK does not affect subsequent transitions on the $\overline{\text{IRQ}}$ pin. A falling edge that occurs after writing to the ACK bit latches another interrupt request. If the IRQ mask bit, IMASK, is clear, the CPU loads the program counter with the vector address at locations $FFF8 and $FFF9.
- Return of the $\overline{\text{IRQ}}$ pin to logic one — As long as the $\overline{\text{IRQ}}$ pin is at logic zero, IRQ remains active.

The vector fetch or software clear and the return of the $\overline{\text{IRQ}}$ pin to logic one may occur in any order. The interrupt request remains pending as long as the $\overline{\text{IRQ}}$ pin is at logic zero. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

If the MODE bit is clear, the $\overline{\text{IRQ}}$ pin is falling-edge-sensitive only. With MODE clear, a vector fetch or software clear immediately clears the IRQ latch.

The IRQF bit in the ISCR register can be used to check for pending interrupts. The IRQF bit is not affected by the IMASK bit, which makes it useful in applications where polling is preferred.

Use the BIH or BIL instruction to read the logic level on the $\overline{\text{IRQ}}$ pin.

NOTE: *When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine.*
An internal pullup resistor to V_{DD} is connected to $\overline{IRQ}$ pin; this can be disabled by setting the IRQPD bit in the IRQ option control register ($001C).

15.6 PTE4/D– Pin

The PTE4 pin is configured as an interrupt input to trigger the IRQ interrupt when the following conditions are satisfied:

- The USB module is disabled (USBEN = 0)
- PTE4 pin configured for external interrupt input (PTE4IE = 1)

Setting PTE4IE configures the PTE4 pin to an input pin with an internal pullup device. The PTE4 interrupt is "ORed" with the $\overline{\text{IRQ}}$ input to trigger the IRQ interrupt (see **Figure 15-1 . IRQ Module Block Diagram**). Therefore, the IRQ status and control register affects both the $\overline{\text{IRQ}}$ pin and the PTE4 pin. An interrupt on PTE4 also sets the PTE4 interrupt flag, PTE4IF, in the IRQ option control register (IOCR).

15.7 IRQ Module During Break Interrupts

The system integration module (SIM) controls whether the IRQ latch can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear the latches during the break state. (See **Section 8. System Integration Module (SIM)**.)

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

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

15.8 IRQ Status and Control Register

The IRQ status and control register (ISCR) controls and monitors operation of the IRQ module. The ISCR has the following functions:

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

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	IRQF	0	IMASK	MODE
Write:						ACK		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 15-3. IRQ Status and Control Register (ISCR)

IRQF — IRQ Flag

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

1 = IRQ interrupt pending
0 = IRQ interrupt not pending

ACK — IRQ Interrupt Request Acknowledge Bit

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

IMASK — IRQ Interrupt Mask Bit

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

1 = IRQ interrupt requests disabled
0 = IRQ interrupt requests enabled

MODE — IRQ Edge/Level Select Bit

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

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

15.9 IRQ Option Control Register

The IRQ option control register controls and monitors the external interrupt function available on the PTE4 pin. It also disables/enables the pullup resistor on the $\overline{IRQ}$ pin.

- Controls pullup option on $\overline{IRQ}$ pin
- Enables PTE4 pin for external interrupts to IRQ
- Shows the state of the PTE4 interrupt flag

Address: $001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	PTE4IF	PTE4IE	IRQPD
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 15-4. IRQ Option Control Register (IOCR)

PTE4IF — PTE4 Interrupt Flag

This read-only status bit is high when a falling edge on PTE4 pin is detected. PTE4IF bit clears when the IOCR is read.

1 = Falling edge on PTE4 is detected and PTE4IE is set
0 = Falling edge on PTE4 is not detected or PTE4IE is clear

PTE4IE — PTE4 Interrupt Enable

This read/write bit enables or disables the interrupt function on the PTE4 pin to trigger the IRQ interrupt. Setting the PTE4IE bit and clearing the USBEN bit in the USB address register configure the PTE4 pin for interrupt function to the IRQ interrupt. Setting PTE4IE also enables the internal pullup on PTE4 pin.

1 = PTE4 interrupt enabled; triggers IRQ interrupt
0 = PTE4 interrupt disabled

IRQPD — $\overline{IRQ}$ Pullup Disable

This read/write bit controls the pullup option for the $\overline{IRQ}$ pin.

1 = Internal pullup is disconnected
0 = Internal pull-up is connected between $\overline{IRQ}$ pin and V_{DD}

External Interrupt (IRQ)

Section 16. Keyboard Interrupt Module (KBI)

16.1 Contents

16.2 Introduction

The keyboard interrupt module (KBI) provides eight independently maskable external interrupts which are accessible via PTA0–PTA7 pins.

16.3 Features

Features of the keyboard interrupt module include:

- Eight keyboard interrupt pins with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Hysteresis buffers
- Programmable edge-only or edge- and level-interrupt sensitivity
- Exit from low-power modes

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0016	Keyboard Status and Control Register (KBSCR)	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$0017	Keyboard Interrupt Enable Register (KBIER)	Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 16-1. I/O Register Summary

16.4 Pin Name Conventions

The KBI share eight I/O pins with eight port A I/O pins. The full name of the I/O pins are listed in **Table 16-1**. The generic pin name appear in the text that follows.

Table 16-1. Pin Name Conventions

Full MCU Pin Name	KBI Generic Pin Name	Pin Selected for KBI Function by KBIEx Bit in KBIER
PTA7/KBA7–PTA0/KBA0	KBA7–KBA0	KBIE7–KBIE0

16.5 Functional Description

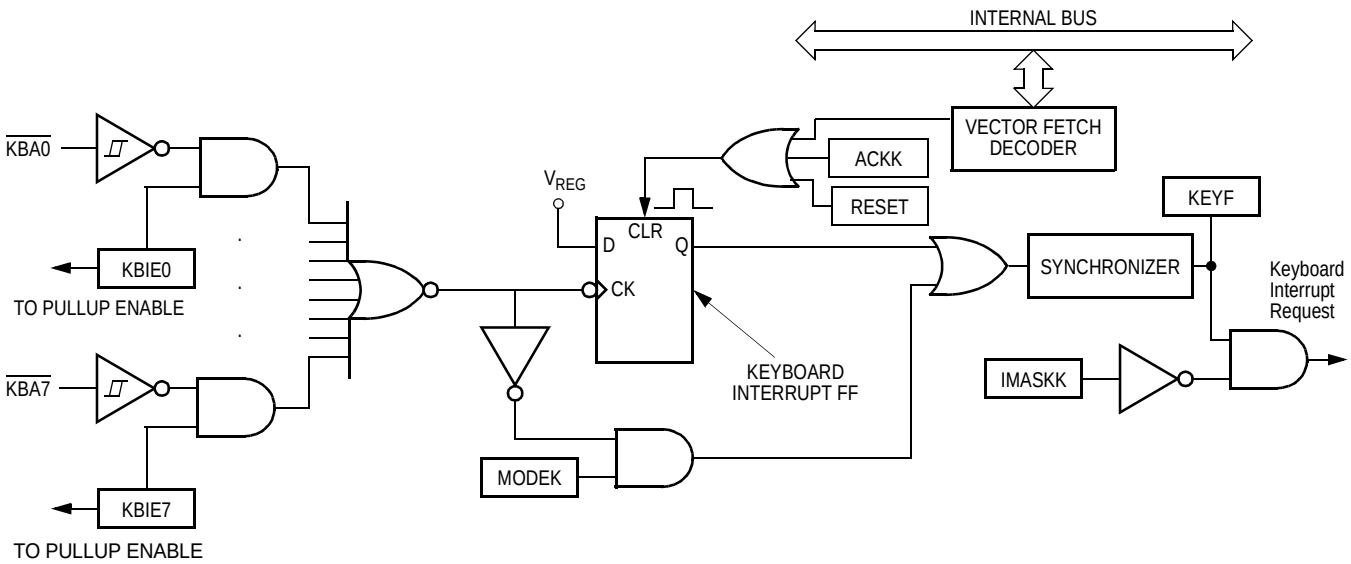


Figure 16-2. Keyboard Module Block Diagram

Writing to the KBIE7–KBIE0 bits in the keyboard interrupt enable register independently enables or disables each port A pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin also enables its internal pullup 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 also can 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.*

16.6 Keyboard Initialization

When a keyboard interrupt pin is enabled, it takes time for the pullup device 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 DDRA bits in data direction register A.
2. Write logic 1s to the appropriate port A data register bits.
3. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.

16.7 I/O Registers

These registers control and monitor operation of the keyboard module:

- Keyboard status and control register (KBSCR)
- Keyboard interrupt enable register (KBIER)

16.7.1 Keyboard Status and Control Register

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

Address: $0016

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
Write:						ACKK		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 16-3. Keyboard Status and Control Register (KBSCR)

KEYF — Keyboard Flag Bit

This read-only bit is set when a keyboard interrupt is pending. Reset clears the KEYF bit.

1 = Keyboard interrupt pending
0 = No keyboard interrupt pending

ACKK — Keyboard Acknowledge Bit

Writing a logic 1 to this write-only bit clears the keyboard interrupt request. ACKK always reads as logic 0. Reset clears ACKK.

IMASKK — Keyboard Interrupt Mask Bit

Writing a logic 1 to this read/write bit prevents the output of the keyboard interrupt mask from generating interrupt requests. Reset clears the IMASKK bit.

1 = Keyboard interrupt requests masked
0 = Keyboard interrupt requests not masked

MODEK — Keyboard Triggering Sensitivity Bit

This read/write bit controls the triggering sensitivity of the keyboard interrupt pins. Reset clears MODEK.

1 = Keyboard interrupt requests on falling edges and low levels
0 = Keyboard interrupt requests on falling edges only

16.7.2 Keyboard Interrupt Enable Register

The keyboard interrupt enable register enables or disables each port A pin to operate as a keyboard interrupt pin.

Address: $0017

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Reset:	0	0	0	0	0	0	0	0

Figure 16-4. Keyboard Interrupt Enable Register (KBIER)

KBIE7–KBIE0 — Keyboard Interrupt Enable Bits

Each of these read/write bits enables the corresponding keyboard interrupt pin to latch interrupt requests. Reset clears the keyboard interrupt enable register.

1 = PTAx/KBAx pin enabled as keyboard interrupt pin
0 = PTAx/KBAx not enabled as keyboard interrupt pin

16.8 Low-Power Modes

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

16.8.1 Wait Mode

The keyboard module remains active in wait mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of wait mode.

16.8.2 Stop Mode

The keyboard module remains active in stop mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of stop mode.

16.9 Keyboard Module During Break Interrupts

The system integration module (SIM) controls whether the keyboard interrupt latch can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state.

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

To protect the latch during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), writing to the keyboard acknowledge bit (ACKK) in the keyboard status and control register during the break state has no effect. (See **16.7.1 Keyboard Status and Control Register**.)

Section 17. Computer Operating Properly (COP)

17.1 Contents

17.2 Introduction

The computer operating properly (COP) module contains a free-running counter that generates a reset if allowed to overflow. The COP module helps software recover from runaway code. Prevent a COP reset by clearing the COP counter periodically. The COP module can be disabled through the COPD bit in the CONFIG register.

17.3 Functional Description

Figure 17-1 shows the structure of the COP module.

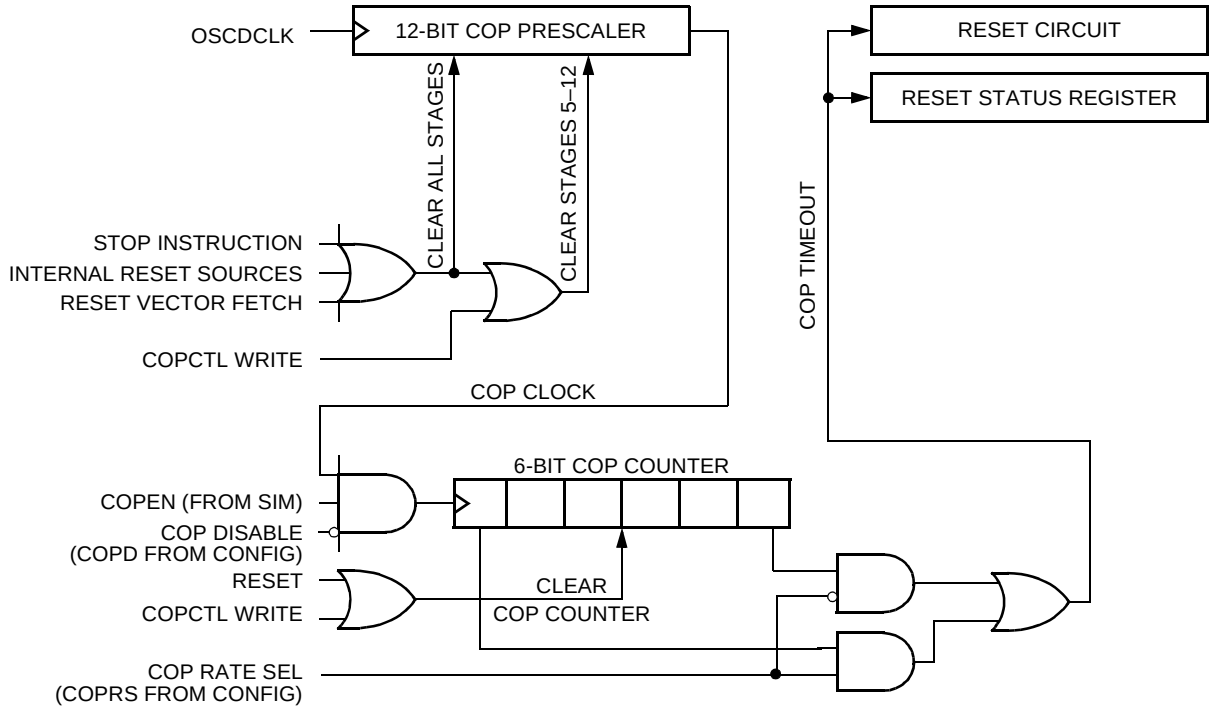


Figure 17-1. COP Block Diagram

The COP counter is a free-running 6-bit counter preceded by a 12-bit prescaler counter. If not cleared by software, the COP counter overflows and generates an asynchronous reset after $2^{18} - 2^4$ or $2^{13} - 2^4$ OSCDCLK cycles, depending on the state of the COP rate select bit, COPRS, in configuration register 1. With a $2^{18} - 2^4$ OSCDCLK cycle overflow option, a 24MHz OSCDCLK (12MHz crystal) gives a COP timeout period of 10.92ms. Writing any value to location $FFFF before an overflow occurs prevents a COP reset by clearing the COP counter and stages 12 through 5 of the prescaler.

NOTE: *Service the COP immediately after reset and before entering or after exiting stop mode to guarantee the maximum time before the first COP counter overflow.*

A COP reset pulls the $\overline{\text{RST}}$ pin low for 32 OSCDCLK cycles and sets the COP bit in the SIM reset status register (SRSR).

In monitor mode, the COP is disabled if the $\overline{\text{RST}}$ pin or the $\overline{\text{IRQ}}$ is held at V_{TST}. During the break state, V_{TST} on the $\overline{\text{RST}}$ pin disables the COP.

NOTE: *Place COP clearing instructions in the main program and not in an interrupt subroutine. Such an interrupt subroutine could keep the COP from generating a reset even while the main program is not working properly.*

17.4 I/O Signals

The following paragraphs describe the signals shown in **Figure 17-1**.

17.4.1 OSCDCLK

OSCDCLK is the crystal oscillator clock doubler output signal. Its frequency is two times the crystal frequency.

17.4.2 STOP Instruction

The STOP instruction clears the COP prescaler.

17.4.3 COPCTL Write

Writing any value to the COP control register (COPCTL) (see **17.5 COP Control Register**) clears the COP counter and clears bits 12 through 5 of the prescaler. Reading the COP control register returns the low byte of the reset vector.

17.4.4 Power-On Reset

The power-on reset (POR) circuit clears the COP prescaler 4096 OSCDCLK cycles after power-up.

17.4.5 Internal Reset

An internal reset clears the COP prescaler and the COP counter.

17.4.6 Reset Vector Fetch

A reset vector fetch occurs when the vector address appears on the data bus. A reset vector fetch clears the COP prescaler.

17.4.7 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the CONFIG register. (See **Figure 17-2**.)

17.4.8 COPRS (COP Rate Select)

The COPRS signal reflects the state of the COP rate select bit (COPRS) in the CONFIG register. (See **Figure 17-2**.)

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LVIDR	LVI5OR3	URSTD	LVID	SSREC	COPRS	STOP	COPD
Reset:	0*	0*	0*	0*	0	0	0	0

* LVIDR, LVI5OR3, URSTD, and LVID, are reset by POR or LVI reset only.

Figure 17-2. Configuration Register (CONFIG)

COPRS — COP Rate Select Bit

COPRS selects the COP timeout period. Reset clears COPRS.

1 = COP timeout period is $2^{13} – 2^4$ OSCDCLK cycles
0 = COP timeout period is $2^{18} – 2^4$ OSCDCLK cycles

COPD — COP Disable Bit

COPD disables the COP module.

1 = COP module disabled
0 = COP module enabled

17.5 COP Control Register

The COP control register is located at address $FFFF and overlaps the reset vector. Writing any value to $FFFF clears the COP counter and starts a new timeout period. Reading location $FFFF returns the low byte of the reset vector.

Address: $FFFF

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Low byte of reset vector							
Write:	Clears COP counter (any value)							
Reset:	Unaffected by reset							

Figure 17-3. COP Control Register (COPCTL)

17.6 Interrupts

The COP does not generate CPU interrupt requests.

17.7 Monitor Mode

When monitor mode is entered with V_{TST} on the $\overline{IRQ}$ pin, the COP is disabled as long as V_{TST} remains on the $\overline{IRQ}$ pin or the $\overline{RST}$ pin. When monitor mode is entered by having blank reset vectors and not having V_{TST} on the $\overline{IRQ}$ pin, the COP is automatically disabled until a POR occurs.

17.8 Low-Power Modes

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

17.8.1 Wait Mode

The COP remains active during wait mode. To prevent a COP reset during wait mode, periodically clear the COP counter in a CPU interrupt routine.

17.8.2 Stop Mode

Stop mode turns off the OSCDCLK input to the COP and clears the COP prescaler. Service the COP immediately before entering or after exiting stop mode to ensure a full COP timeout period after entering or exiting stop mode.

To prevent inadvertently turning off the COP with a STOP instruction, a configuration option is available that disables the STOP instruction. When the STOP bit in the configuration register has the STOP instruction is disabled, execution of a STOP instruction results in an illegal opcode reset.

17.9 COP Module During Break Mode

The COP is disabled during a break interrupt when V_{TST} is present on the $\overline{RST}$ pin.

Section 18. Low-Voltage Inhibit (LVI)

18.1 Contents

18.2 Introduction

This section describes the low-voltage inhibit (LVI) module, which monitors the voltage on the V_{DD} pin and V_{REG} pin. and can force a reset when the V_{DD} or V_{REG} voltage falls below the LVI trip falling voltage.

18.3 Features

Features of the LVI module include:

- Independent voltage monitoring circuits for V_{DD} and V_{REG}
- Independent LVI circuit disable for V_{DD} and V_{REG}
- Selectable LVI trip voltage for V_{DD}

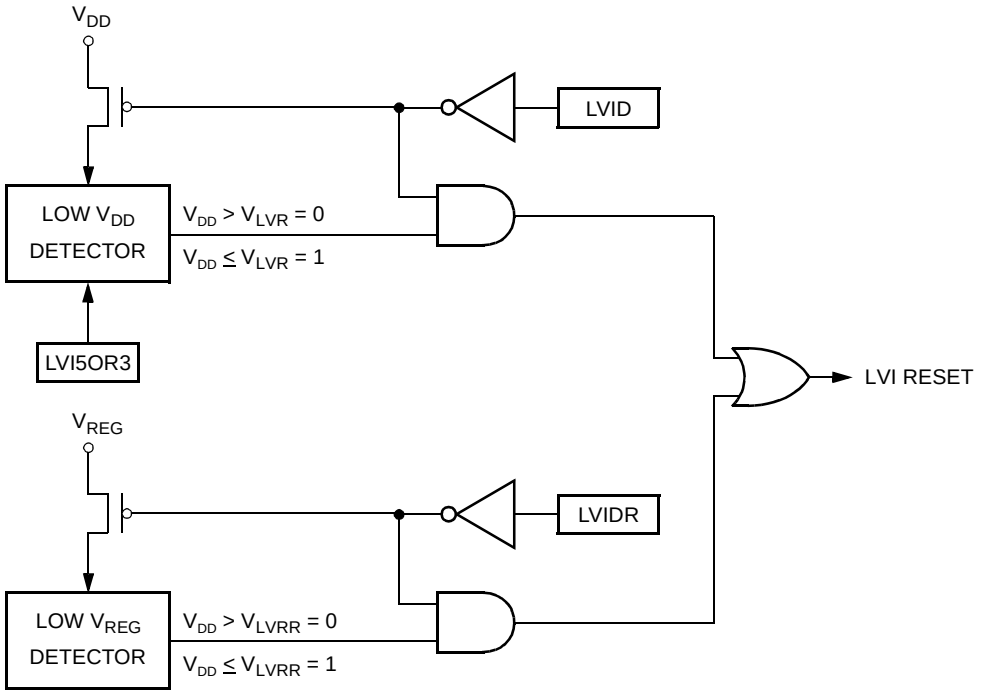


Figure 18-1. LVI Module Block Diagram

18.4 Functional Description

Figure 18-1 shows the structure of the LVI module. The LVI is enabled out of reset. The LVI module contains independent bandgap reference circuit and comparator for monitoring the V_{DD} voltage and the V_{REG} voltage. An LVI reset performs a MCU internal reset and drives the $\overline{RST}$ pin low to provide low-voltage protection to external peripheral devices.

18.4.1 Low V_{DD} Detector

The low V_{DD} detector circuit monitors the V_{DD} voltage and forces a LVI reset when the V_{DD} voltage falls below the trip voltage. The LVI5OR3 bit in the configuration register (CONFIG) selects the trip point voltage. The V_{DD} LVI circuit can be disabled by the setting the LVID bit in CONFIG. See **8.4.2.5 Low-Voltage Inhibit (LVI) Reset** for details of the interaction between the SIM and the LVI.

18.4.2 Low V_{REG} Detector

The low V_{REG} detector circuit monitors the V_{REG} voltage and forces a LVI reset when the V_{REG} voltage falls below the trip voltage. The V_{REG} LVI circuit can be disabled by the setting the LVIDR bit in CONFIG.

NOTE: *There is no LVI circuit for V_{REGA}.*

18.5 LVI Control and Configuration

Three bits in the configuration register (CONFIG) control the operation of the LVI module.

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LVIDR	LVI5OR3	URSTD	LVID	SSREC	COPRS	STOP	COPD
Reset:	0*	0*	0*	0*	0	0	0	0

= Unimplemented

* LVIDR, LVI5OR3, URSTD, and LVID bits are reset by POR (power-on reset) or LVI reset only.

Figure 18-2. Configuration Register (CONFIG)

LVIDR — LVI Disable Bit for V_{REG}

LVIDR disables the LVI circuit for V_{REG}.

1 = LVI circuit for V_{REG} disabled
0 = LVI circuit for V_{REG} enabled

LVI5OR3 — LVI Trip Point Voltage Select Bit for V_{DD}

LVI5OR3 selects the trip point voltage of the LVI circuit for V_{DD}. See **Section 20. Electrical Specifications** for the trip voltage tolerances.

1 = LVI trips at 3.3V
0 = LVI trips at 2.4V

LVID — LVI Disable Bit for V_{DD}

LVID disables the LVI circuit for V_{DD}.

1 = LVI circuit for V_{DD} disabled
0 = LVI circuit for V_{DD} enabled

18.6 Low-Power Modes

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

18.6.1 Wait Mode

If enabled, the LVI module remains active in wait mode.

18.6.2 Stop Mode

If enabled, the LVI module remains active in stop mode.

Section 19. Break Module (BRK)

19.1 Contents

19.2 Introduction

This section describes the break module. The break module can generate a break interrupt that stops normal program flow at a defined address to enter a background program.

19.3 Features

Features of the break module include:

- Accessible input/output (I/O) registers during the break interrupt
- CPU-generated break interrupts
- Software-generated break interrupts
- COP disabling during break interrupts

19.4 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal to the CPU. The CPU then loads the instruction register with a software interrupt instruction (SWI) after completion of the current CPU instruction. The program counter vectors to $FFFC and $FFFD ($FEFC and $FEFD in monitor mode).

The following events can cause a break interrupt to occur:

- A CPU-generated address (the address in the program counter) matches the contents of the break address registers.
- Software writes a logic 1 to the BRKA bit in the break status and control register.

When a CPU-generated address matches the contents of the break address registers, the break interrupt begins after the CPU completes its current instruction. A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation. **Figure 19-1** shows the structure of the break module.

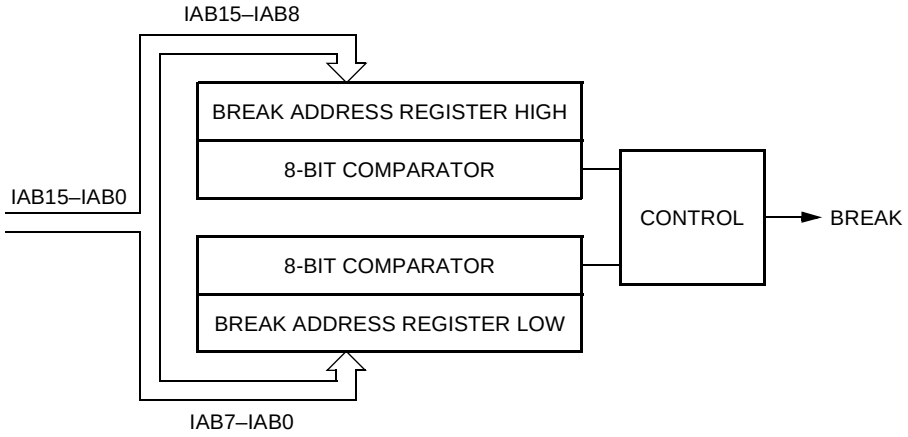


Figure 19-1. Break Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE00	SIM Break Status Register (SBSR)	Read:	R	R	R	R	R	R	SBSW	R
		Write:							Note	
		Reset:							0	
$FE03	SIM Break Flag Control Register (SBFCR)	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE0C	Break Address Register High (BRKH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0D	Break Address Register Low (BRKL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0E	Break Status and Control Register (BRKSCR)	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

Note: Writing a logic 0 clears SBSW.

= Unimplemented R = Reserved

Figure 19-2. Break Module I/O Register Summary

19.4.1 Flag Protection During Break Interrupts

The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state.

19.4.2 CPU During Break Interrupts

The CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with $FFFC and $FFFD ($FEFC and $FEFD in monitor mode)

The break interrupt begins after completion of the CPU instruction in progress. If the break address register match occurs on the last cycle of a CPU instruction, the break interrupt begins immediately.

19.4.3 TIM During Break Interrupts

A break interrupt stops the timer counters.

19.4.4 COP During Break Interrupts

The COP is disabled during a break interrupt when V_{TST} is present on the $\overline{RST}$ pin.

19.5 Low-Power Modes

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

19.5.1 Wait Mode

If enabled, the break module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if SBSW is set (see **Section 8. System Integration Module (SIM)**). Clear the SBSW bit by writing logic 0 to it.

19.5.2 Stop Mode

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

19.6 Break Module Registers

These registers control and monitor operation of the break module:

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

19.6.1 Break Status and Control Register

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

Address: $FE0E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BRKE	BRKA	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 19-3. Break Status and Control Register (BRKSCR)

BRKE — Break Enable Bit

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

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

BRKA — Break Active Bit

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

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

19.6.2 Break Address Registers

The break address registers (BRKH and BRKL) contain the high and low bytes of the desired breakpoint address. Reset clears the break address registers.

Address: $FE0C

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

Figure 19-4. Break Address Register High (BRKH)

Address: $FE0D

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

Figure 19-5. Break Address Register Low (BRKL)

19.6.3 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from wait mode. The flag is useful in applications requiring a return to wait mode after exiting from a break interrupt.

Address: $FE00

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	R	R	R	R	R	SBSW	R
Write:							Note	
Reset:							0	

Note: Writing a logic 0 clears SBSW. R = Reserved

Figure 19-6. SIM Break Status Register (SBSR)

SBSW — SIM Break Stop/Wait Bit

This status bit is useful in applications requiring a return to wait or stop mode after exiting from a break interrupt. Clear SBSW by writing a logic 0 to it. Reset clears SBSW.

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

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

```
;This code works if the H register has been pushed onto the stack in the break
;service routine software. This code should be executed at the end of the break
;service routine software.

 HIBYTE   EQU     5

 LOBYTE   EQU     6

;         If not SBSW, do RTI

          BRCLR   SBSW,SBSR, RETURN    ;See if wait mode or stop mode was exited by
                                       ;break.

          TST     LOBYTE,SP            ;If RETURNLO is not zero,

          BNE     DOLO                 ;then just decrement low byte.

          DEC     HIBYTE,SP            ;Else deal with high byte, too.

 DOLO     DEC     LOBYTE,SP            ;Point to WAIT/STOP opcode.

 RETURN   PULH                         ;Restore H register.
          RTI
```

19.6.4 SIM Break Flag Control Register

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

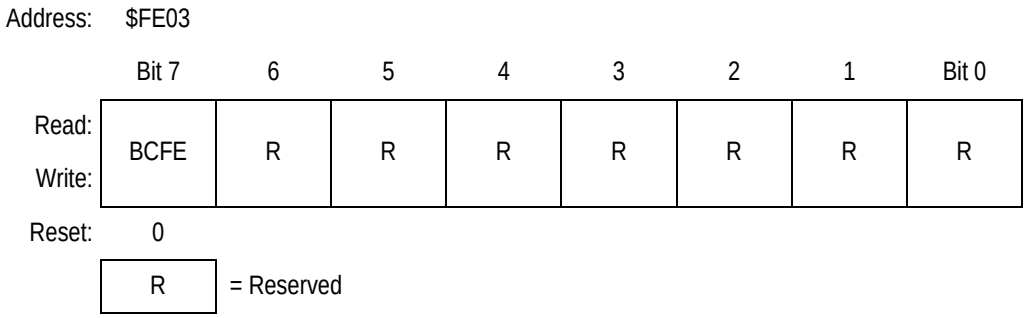


Figure 19-7. SIM Break Flag Control Register (SBFCR)

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

Section 20. Electrical Specifications

20.1 Contents

20.2 Introduction

This section contains electrical and timing specifications.

20.3 Absolute Maximum Ratings

Maximum ratings are the extreme limits to which the MCU can be exposed without permanently damaging it.

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to* ***20.6 DC Electrical Characteristics*** *for guaranteed operating conditions.*

Characteristic(1)	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to +6.0	V
Input voltage PTE4/D–, PTE3/D+ Others	V_{IN}	$V_{SS}-1.0$ to $V_{DD}+0.3$ $V_{SS}-0.3$ to $V_{DD}+0.3$	V
Mode entry voltage, $\overline{IRQ}$ pin	V_{TST}	V_{SS}–0.3 to +8	V
Maximum current per pin excluding V_{DD} and V_{SS}	I	± 25	mA
Storage temperature	T_{STG}	–55 to +150	°C
Maximum current out of V_{SS}/V_{SS}	I_{MVSS}	100	mA
Maximum current into V_{DD}/V_{DDA}	I_{MVDD}	100	mA

Notes:
1. Voltages referenced to V_{SS}

NOTE: *This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{IN} and V_{OUT} be constrained to the range $V_{SS} \le (V_{IN}$ or $V_{OUT}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD}).*

20.4 Functional Operating Range

Characteristic	Symbol	Value	Unit
Operating temperature range	T_A	0 to 70	°C
Operating voltage range	V_{DD}	4.0 to 5.5	V

20.5 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance LQFP (32 pins) SOIC (28 pins)	θ_{JA}	 95 70	°C/W
I/O pin power dissipation	$P_{I/O}$	User-Determined	W
Power dissipation(1)	P_D	$P_D = (I_{DD} \times V_{DD}) + (I_{DDA} \times V_{DDA}) + P_{I/O} = K/(T_J + 273\ °C)$	W
Constant(2)	K	$P_D \times (T_A + 273\ °C) + P_D^2 \times \theta_{JA}$	W/°C
Average junction temperature	T_J	$T_A + (P_D \times \theta_{JA})$	°C
Maximum junction temperature	T_{JM}	100	°C

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

20.6 DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Regulator output voltage	V_{REG} V_{REGA0}	3.0 2.9	3.3 3.3	3.6 3.7	V
Output high voltage (I_{Load} = –2.0 mA) PTA0–PTA7, PTC0–PTC1, PTE0–PTE2	V_{OH}	V_{DD}–0.8	—	—	V
Output low voltage (I_{Load} = 1.6 mA) All I/O pins (I_{Load} = 25 mA) PTD0–PTD1 in ILDD mode (I_{Load} = 10 mA) PTE3–PTE4 with USB is disabled	V_{OL}	— — —	— — —	0.4 0.5 0.4	V
Input high voltage OSC1 All ports, $\overline{IRQ}$, $\overline{RST}$	V_{IH}	$0.7 \times V_{REG}$ $0.7 \times V_{DD}$	— —	V_{REG} V_{DD}	V
Input low voltage OSC1 All ports, $\overline{IRQ}$, $\overline{RST}$	V_{IL}	V_{SS} V_{SS}	— —	$0.3 \times V_{REG}$ $0.3 \times V_{DD}$	V
Output low current (V_{OL} = 2.0 V) PTD2–PTD5 in LDD mode	I_{OL}	10	13	20	mA
V_{DD} supply current, V_{DD} = 5.25V, f_{OP} = 6MHz Run, with low speed USB[3] Run, with USB suspended[3] Wait, with low speed USB[4] Wait, with USB suspended[4] Stop (0 °C to 70°C)[5]	I_{DD}	 — — — — —	 7.0 6.5 3.0 2.5 60	 8.5 8.0 5.0 4.0 100	 mA mA mA mA µA
I/O ports Hi-Z leakage current	I_{IL}	—	—	± 10	µA
Input current	I_{IN}	—	—	± 1	µA
Capacitance Ports (as input or output)	C_{Out} C_{In}	— —	— —	12 8	pF
POR re-arm voltage[6]	V_{POR}	0	—	100	mV
POR rise-time ramp rate[7]	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{TST}	V_{DD} + 2.5		8	V
Pullup resistors Port A, port C, PTE0–PTE2, $\overline{RST}$, $\overline{IRQ}$ (to V_{DD}) PTE3–PTE4 with USB module disabled (to V_{DD}) D– with USB module enabled (to V_{REG})	R_{PU}	 20 4 1.1	 35 5 1.5	 50 6 2.0	 kΩ
V_{DD} LVI trip point voltage (LVI5OR3 = 0)	V_{LVR}	2.0	2.4	2.8	V
V_{DD} LVI trip point voltage (LVI5OR3 = 1)		2.8	3.3	3.8	
V_{REG} LVI trip point voltage		2.0	2.2	2.6	

Notes:
1. V_{DD} = 4.0 to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
3. Run (operating) I_{DD} measured using external square wave clock source (f_{XCLK} = 12 MHz). All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD}. Measured with all modules enabled.
4. Wait I_{DD} measured using external square wave clock source (f_{XCLK} = 12 MHz); all inputs 0.2 V from rail; no dc loads; less than 100 pF on all outputs. C_L = 20 pF on OSC2; 15 kΩ ± 5% termination resistors on D+ and D– pins; all ports configured as inputs; OSC2 capacitance linearly affects wait I_{DD}
5. STOP I_{DD} measured with USB in suspend mode; OSC1 grounded; no port pins sourcing current.
6. Maximum is highest voltage that POR is guaranteed.
7. If minimum V_{DD} is not reached before the internal POR reset is released, $\overline{RST}$ must be driven low externally until minimum V_{DD} is reached.

20.7 Control Timing

Characteristic(1)	Symbol	Min	Max	Unit
Internal operating frequency(2)	f_{OP}	—	6	MHz
$\overline{RST}$ input pulse width low(3)	t_{IRL}	125	—	ns

Notes:
1. V_{DD} = 4.0 to 5.5 Vdc; V_{SS} = 0 Vdc; timing shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Some modules may require a minimum frequency greater than dc for proper operation; see appropriate table for this information.
3. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.

20.8 Oscillator Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Crystal frequency(1)	f_{XCLK}	1	12	12	MHz
External clock Reference frequency(1), (2)	f_{XCLK}	dc	12	12	MHz
Crystal load capacitance(3)	C_L	—	—	—	
Crystal fixed capacitance(3)	C_1	—	$2 \times C_L$	—	
Crystal tuning capacitance(3)	C_2	—	$2 \times C_L$	—	
Feedback bias resistor	R_B	—	10 MΩ	—	
Series resistor(3), (4)	R_S	—	—	—	

Notes:
1. The USB module is designed to operate with f_{XCLK} = 12 MHz.
2. No more than 10% duty cycle deviation from 50%.
3. Consult crystal vendor data sheet.
4. Not required for high-frequency crystals.

20.9 TImer Interface Module Characteristics

Characteristic	Symbol	Min	Max	Unit
Input capture pulse width	t_{TIH}, t_{TIL}	$1/f_{OP}$	—	ns
Input clock pulse width	t_{TCH}, t_{TCL}	$(1/f_{OP}) + 5$	—	ns

20.10 USB DC Electrical Characteristics

Characteristic[1]	Symbol	Conditions	Min	Typ	Max	Unit
Hi-Z state data line leakage	I_{LO}	$0\ V<V_{IN}<3.3\ V$	–10		+10	µA
Voltage input high (driven)	V_{IH}		2.0			V
Voltage input high (floating)	V_{IHZ}		2.7		3.6	V
Voltage input low	V_{IL}				0.8	V
Differential input sensitivity	V_{DI}	|(D+) – (D–)|	0.2			V
Differential common mode range	V_{CM}	Includes V_{DI} Range	0.8		2.5	V
Static output low	V_{OL}	R_L of 1.425 K to 3.6 V			0.3	V
Static output high	V_{OH}	R_L of 14.25 K to GND	2.8		3.6	V
Output signal crossover voltage	V_{CRS}		1.3	—	2.0	V
Regulator bypass capacitor	$C_{REGBYPASS}$			0.1		µF
Regulator bulk capacitor	$C_{REGBULK}$		4.7			µF

Notes:
1. V_{DD} = 4.0 to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.

20.11 USB Low-Speed Source Electrical Characteristics

Characteristic[1]	Symbol	Conditions	Min	Typ	Max	Unit
Internal operating frequency	f_{OP}	—	—	6	—	MHz
Transition time[2] Rise time Fall time	 t_R t_F	 C_L = 200 pF C_L = 600 pF C_L = 200 pF C_L = 600 pF	 75 75	 — —	 300 300	 ns
Rise/Fall time matching	t_{RFM}	t_R/t_F	80	—	120	%
Low speed data rate	t_{DRATE}	1.5 Mbs ± 1.5%	1.4775 676.8	1.500 666.0	1.5225 656.8	Mbs ns
Source differential driver jitter To next transition For paired transitions	 t_{DDJ1} t_{DDJ2}	C_L = 600 pF Measured at crossover point	 –25 –10	 — —	 25 10	ns
Receiver data jitter tolerance To next transition For paired transitions	 t_{DJR1} t_{DJR2}	C_L = 600 pF Measured at crossover point	 –75 –45	 — —	 75 45	ns
Source SEO interval of EOP	t_{LEOPT}	Measured at crossover point	1.25	—	1.50	µs
Source jitter for differential transition to SE0 transition[3]		Measured at crossover point		667		ns
Receiver SEO interval of EOP Must reject as EOP Must accept	 t_{LEOPR1} t_{LEOPR2}	Measured at crossover point	 210 670	 — —	 — —	ns
Width of SEO interval during differential transition	t_{LST}	Measured at crossover point	—	—	210	ns

Notes:
1. All voltages are measured from local ground, unless otherwise specified. All timings use a capacitive load of 50 pF, unless otherwise specified. Low-speed timings have a 1.5kΩ pullup to 2.8 V on the D– data line.
2. Transition times are measured from 10% to 90% of the data signal. The rising and falling edges should be smoothly transitioning (monotonic). Capacitive loading includes 50 pF of tester capacitance.
3. The two transitions are a (nominal) bit time apart.

20.12 USB Signaling Levels

Bus State	Signaling Levels	
	Transmit	Receive
Differential 1	D+ > V_{OH} (min) and D– < V_{OL} (max)	(D+) – (D–) > 200 mV
Differential 0	D– > V_{OH} (min) and D– < V_{OL} (max)	(D–) – (D+) > 200 mV
Single-ended 0 (SE0)	D+ and D– < V_{OL} (max)	D+ and D– < V_{IL} (max)
Data J state (low speed)	Differential 0	Differential 0
Data K state (low speed)	Differential 1	Differential 1
Idle state (low speed)	NA	D– > V_{IHZ} (min) and D+ < V_{IL} (max)
Resume state	Differential 1	Differential 1
Start of packet (SOP)	Data lines switch from Idle to K State	
End of packet (EOP)	SE0 for approximately 2 bit times(1) followed by a J state for 1 bit time	SE0 for ≥ 1 bit time(2) followed by a J state for 1 bit time
Reset	NA	D+ and D– < V_{IL} (max) for ≥ 8µs

Notes:
1. The width of EOP is defined in bit times relative to the speed of transmission.
2. The width of EOP is defined in bit times relative to the device type receiving the EOP. The bit time is approximate.

20.13 CGM Electrical Characteristics

Characteristic(1)	Condition	Symbol	Min	Typ(2)	Max	Unit
CGM power supply		V_{DDA}	4.0	5	5.5	V
CGM supply current	Both PLLs enabled One PLL enabled	I_{DDA}	— —	6 3	10 5	mA mA
CGM regulator output voltage		V_{REGA0}	2.9	3.3	3.7	V
CGM output high voltage	(I_{Load} = –2.0 mA)	V_{OH}	V_{REGA0}–0.8	—	—	V
CGM output low voltage	(I_{Load} = 1.6 mA)	V_{OL}	—	—	0.4	V
CGM output equivalent capacitance		C_{OUT}	—	—	12	pF
CGM output rise and fall time	R_L = 2kΩ C_L = 10pF 10% to 90%	t_R/t_F	—	—	8	ns
Output resistive load		R_L	2	—	—	kΩ

Characteristic[1]	Condition	Symbol	Min	Typ[2]	Max	Unit
Output capacitive load		C_L	—	—	10	pF
PLL frequency list Rx LO 1 Rx LO 2 Rx LO 3 Rx LO 4 Rx LO 5			—	 26.54 26.59 26.64 26.69 26.74	—	MHz
PLL output signal frequency accuracy	Exclude crystal OSC tolerance		—	±100 ±4	—	Hz ppm
PLL output signal phase noise level	At ±1kHz offset from carrier		—	–40	—	dBc/Hz
VCO frequency range			26	—	28	MHz
PLL lock duration	channel to channel[3]		—	10	—	ms
	Wait/stop mode to active[4]		—	20	—	ms
Duration for Lock bit detection	Within ±10% final frequency[5]		—	10	—	ms
PLL stop duration	PLL module from active to disable mode.		—	—	1	ms
PLL output sideband noise level[6]	At offset >4kHz At offset >42.5kHz		—	–40 –50	—	dBc
PLL output channel intermodulation products[7]	At offset >42.5kHz		—	–50	—	dBc

Notes:

1. V_{DDA} = 4.0 to 5.5 Vdc, V_{SSA} = 0 Vdc, $T_A = T_L$ to T_H, with the pre-defined programming setting for the PLL (see **13.10 Pre-Defined VCO Output Frequency Settings**) and under steady state condition, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
3. Defined as the total time for PLL module switching from channel-to-channel and the frequency is stable with ±60ppm. The reference frequency should be greater than 32kHz.
4. Defined as the total time for PLL module active from wait/stop mode to the frequency is stable with ±60ppm error. The reference frequency should be greater than 32kHz.
5. Defined as the total time for PLL Lock bit setup from un-lock to lock state with the frequency is stable with ±10% error. The reference frequency should be greater than 32kHz.
6. Side-band component generate from reference frequency modulation on carrier.
7. Noise component generate from adjacent channel carrier when both PLLs are enable.

20.14 FLASH Memory Characteristics

Characteristic	Symbol	Min	Max	Unit
RAM data retention voltage	V_{RDR}	1.3	—	V
FLASH block size	—	512		Bytes
FLASH programming size	—	64		Bytes
FLASH read bus clock frequency	f_{Read} (1)	32 k	8.4 M	Hz
FLASH block erase time	t_{Erase} (2)	10	—	ms
FLASH mass erase time	t_{MErase} (3)	200	—	ms
FLASH PGM/ERASE to HVEN set up time	t_{nvs}	5	—	µs
FLASH high-voltage hold time	t_{nvh}	5	—	µs
FLASH high-voltage hold time (mass erase)	t_{nvhl}	100	—	µs
FLASH program hold time	t_{pgs}	10	—	µs
FLASH program time	t_{Prog}	20	40	µs
FLASH return to read time	t_{rcv} (4)	1	—	µs
FLASH cumulative program hv period	t_{HV} (5)	—	8	ms
FLASH row erase endurance (6)	—	10k	—	Cycles
FLASH row program endurance (7)	—	10k	—	Cycles
FLASH data retention time (8)	—	10	—	Years

Notes:

1. f_{Read} is defined as the frequency range for which the FLASH memory can be read.
2. If the page erase time is longer than t_{Erase} (Min), there is no erase-disturb, but it reduced the endurance of the FLASH memory.
3. If the mass erase time is longer than t_{MErase} (Min), there is no erase-disturb, but it reduces the endurance of the FLASH memory.
4. t_{rcv} is defined as the time it needs before the FLASH can be read after turning off the high voltage charge pump, by clearing HVEN to logic 0.
5. t_{HV} is defined as the cumulative high voltage programming time to the same row before next erase.
6. The minimum row endurance value specifies each row of the FLASH memory is guaranteed to work for at least this many erase / program cycles.
7. The minimum row endurance value specifies each row of the FLASH memory is guaranteed to work for at least this many erase / program cycles.
8. The FLASH is guaranteed to retain data over the entire operating temperature range for at least the minimum time specified.

Section 21. Mechanical Specifications

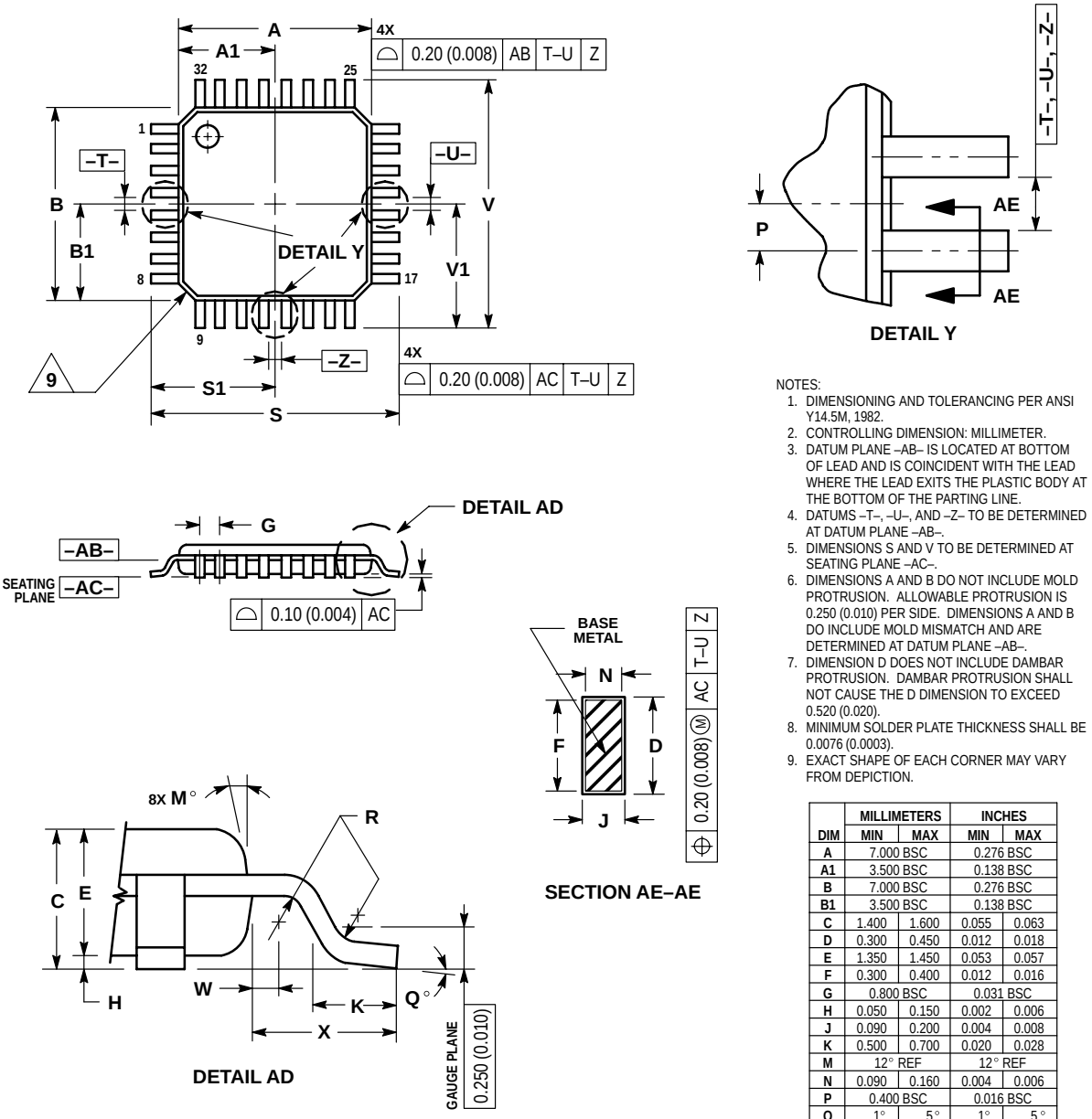
21.1 Contents

21.2 Introduction

This section gives the dimensions for:

- 32-pin low-profile quad flat pack (case #873A)
- 28-pin small outline integrated circuit package (case #751F)

21.3 32-Pin Low-Profile Quad Flat Pack (LQFP)

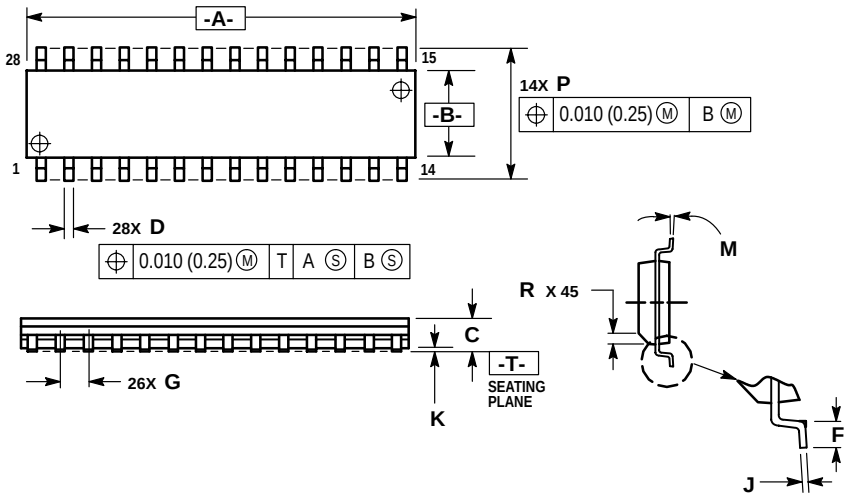


NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE –AB– IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS –T–, –U–, AND –Z– TO BE DETERMINED AT DATUM PLANE –AB–.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE –AC–.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.250 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE –AB–.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE D DIMENSION TO EXCEED 0.520 (0.020).
8. MINIMUM SOLDER PLATE THICKNESS SHALL BE 0.0076 (0.0003).
9. EXACT SHAPE OF EACH CORNER MAY VARY FROM DEPICTION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	7.000 BSC		0.276 BSC	
A1	3.500 BSC		0.138 BSC	
B	7.000 BSC		0.276 BSC	
B1	3.500 BSC		0.138 BSC	
C	1.400	1.600	0.055	0.063
D	0.300	0.450	0.012	0.018
E	1.350	1.450	0.053	0.057
F	0.300	0.400	0.012	0.016
G	0.800 BSC		0.031 BSC	
H	0.050	0.150	0.002	0.006
J	0.090	0.200	0.004	0.008
K	0.500	0.700	0.020	0.028
M	12° REF		12° REF	
N	0.090	0.160	0.004	0.006
P	0.400 BSC		0.016 BSC	
Q	1°	5°	1°	5°
R	0.150	0.250	0.006	0.010
S	9.000 BSC		0.354 BSC	
S1	4.500 BSC		0.177 BSC	
V	9.000 BSC		0.354 BSC	
V1	4.500 BSC		0.177 BSC	
W	0.200 REF		0.008 REF	
X	1.000 REF		0.039 REF	

Figure 21-1. 32-Pin LQFP (Case #873A)

21.4 28-Pin Small Outline Integrated Circuit (SOIC)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 (0.005) TOTAL IN EXCESS OF D DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	17.80	18.05	0.701	0.711
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.41	0.90	0.016	0.035
G	1.27 BSC		0.050 BSC	
J	0.23	0.32	0.009	0.013
K	0.13	0.29	0.005	0.011
M	0°	8°	0°	8°
P	10.01	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

Figure 21-2. 28-Pin SOIC (Case #751F)

Mechanical Specifications

Section 22. Ordering Information

22.1 Contents

22.2 Introduction

This section contains ordering numbers for the MC68HC908JB16.

22.3 MC Order Numbers

Table 22-1. MC Order Numbers

MC Order Number	Package	Operating Temperature Range
MC68HC908JB16DW	28-pin SOIC	0 °C to +70 °C
MC68HC908JB16FA	32-pin LQFP	0 °C to +70 °C

Ordering Information

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.;
Silicon Harbour Centre, 2 Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

http://motorola.com/semiconductors

MC68HC908JB16/D
Rev. 1.0
5/2002