MC68HC08LT8

Data Sheet

M68HC08
Microcontrollers

MC68HC08LT8
Rev. 1
3/2006

freescale.com

freescale™ semiconductor

MC68HC08LT8

Data Sheet

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://freescale.com/

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)
March, 2006	1	First general release.	N/A

List of Chapters

Table of Contents

Chapter 1
General Description

Chapter 2
Memory

Chapter 3
Configuration Register (CONFIG)

Chapter 4
System Integration Module (SIM)

Chapter 5
Oscillator (OSC)

Chapter 6
Timer Interface Module (TIM)

Chapter 7
Programmable Periodic Interrupt (PPI)

Chapter 8
Liquid Crystal Display (LCD) Driver

Chapter 9
Input/Output (I/O) Ports

Chapter 10
External Interrupt (IRQ)

Chapter 11
Keyboard Interrupt Module (KBI)

Chapter 12
Computer Operating Properly (COP)

Chapter 13
Low-Voltage Inhibit (LVI)

Chapter 14
Central Processor Unit (CPU)

Chapter 15
Development Support

Chapter 16
Electrical Specifications

Chapter 17
Ordering Information and Mechanical Specifications

Chapter 1
General Description

1.1 Introduction

The MC68HC08LT8 is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

1.2 Features

Features include:

- 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
- Maximum internal bus frequency:
  - 4-MHz at 5-V operating voltage
  - 2-MHz at 3-V operating voltage
- Dual oscillator module
  - 32.768 kHz crystal oscillator
  - 1 to 16 MHz crystal oscillator
- 8,192 bytes user read-only memory (ROM) with security[(1)]
- 128 bytes of on-chip random-access memory (RAM)
- Two 16-bit, 2-channel timer interface modules (TIM1 and TIM2) with selectable input capture, output compare, and pulse-width modulation (PWM) capability on each channel
- Programmable periodic interrupt (PPI)
- 4/3 backplanes and static with maximum 24/25 frontplanes liquid crystal display (LCD) driver
- Up to 38 general-purpose input/output (I/O) ports:
  - 4 keyboard interrupt with internal pull up
  - 2 × 15 mA high current sink pins
- System protection features:
  - Optional computer operating properly (COP) reset
  - Optional low-voltage detection with reset and selectable trip points for 3-V and 5-V operation
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- Master reset pin with internal pull-up and power-on reset
- $\overline{\text{IRQ}}$ with schmitt-trigger input and programmable pull up
- 44-pin low-profile quad flat pack (LQFP)

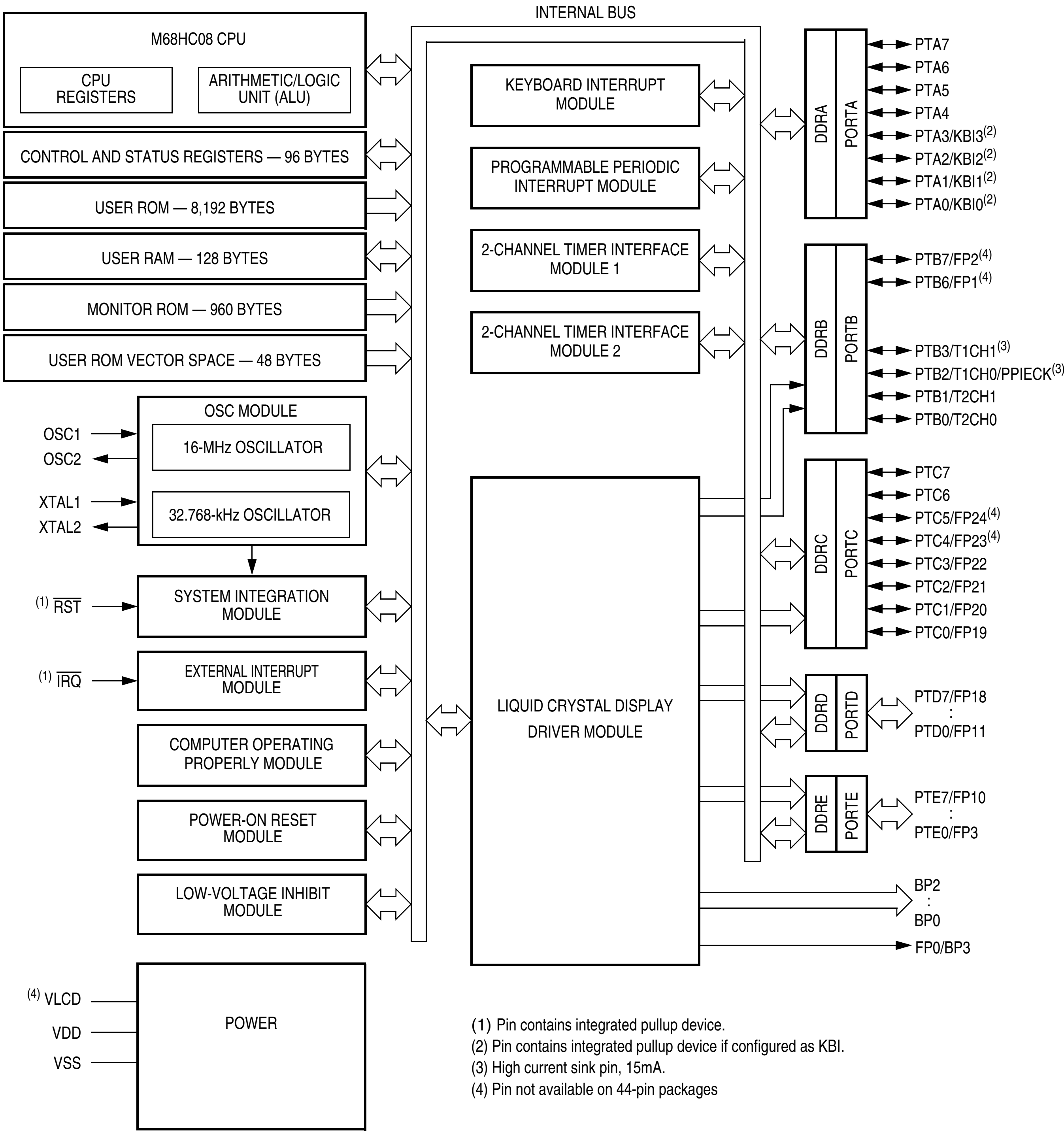
1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the ROM difficult for unauthorized users.

- Specific features in 44-pin package are:
  - 34 general-purpose I/Os
  - 4/3 backplanes and static with maximum 20/21 frontplanes liquid crystal display (LCD) driver

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 × 8 multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Efficient C language support

1.3 MCU Block Diagram

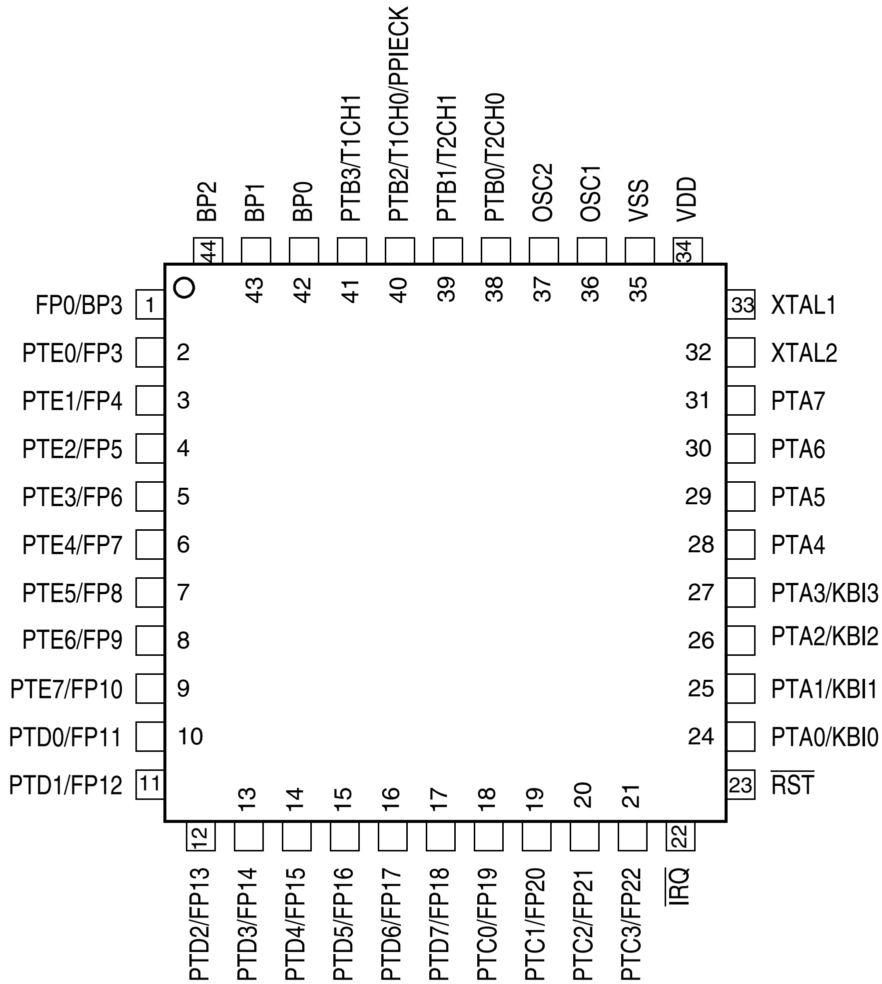
Figure 1-1 shows the structure of the MC68HC08LT8.



(1) Pin contains integrated pullup device.
(2) Pin contains integrated pullup device if configured as KBI.
(3) High current sink pin, 15mA.
(4) Pin not available on 44-pin packages

Figure 1-1. MC68HC08LT8 Block Diagram

1.4 Pin Assignments



Pins not available on 44-pin package		
PTC5/FP24	Internal pads are unconnected	Set these unused port I/Os to output low
PTC4/FP23		
PTC7/FP2		
PTC6/FP1		
VLCD	Internal pad connect to VDD	—

Figure 1-2. 44-Pin LQFP Pin Assignment

1.5 Pin Functions

Description of the pin functions are provided in Table 1-1.

Table 1-1. Pin Functions

Pin Name	Pin Description	Input/Output	Voltage Level
V_{DD}	Power supply	Input	5 V or 3 V
V_{SS}	Power supply ground	Output	0V
V_{LCD}	LCD bias voltage	Input	V_{DD}
$\overline{RST}$	Reset input, active low; with internal pull up and Schmitt trigger input	Input/output	V_{DD}
$\overline{IRQ}$	External $\overline{IRQ}$ pin; with programmable internal pull up and Schmitt trigger input	Input	V_{DD}
	Used for monitor mode entry	Input	V_{DD} to V_{TST}
OSC1	Crystal input for 16-MHz system clock	Input	V_{DD}
OSC2	Crystal oscillator output; inverted OSC1 signal	Output	V_{DD}
XTAL1	Crystal input for 32.768-kHz for subsystem clock	Input	V_{DD}
XTAL2	Crystal oscillator output; inverted XTAL1 signal	Output	V_{DD}
BP0–BP2	LCD backplane drivers	Output	V_{DD}
BP3/FP0	LCD backplane driver BP3 or frontplane driver FP0	Output	V_{DD}
PTA0/KBI0 PTA1/KBI1 PTA2/KBI2 PTA3/KBI3 PTA4 PTA5 PTA6 PTA7	8-bit general-purpose I/O port	Input/output	V_{DD}
	PTA0–PTA3 as keyboard interrupts with pull-up device, KBI0–KBI3	Input	V_{DD}
PTB0/T2CH0 PTB1/T2CH1 PTB2/T1CH0/PPIECK PTB3/T1CH1 PTB6/FP1 PTB7/FP2	8-bit general-purpose I/O port, with high current sinks on PTB2–PTB3	Input/output	V_{DD}
	PTB0 as T2CH0 of TIM2	Input/output	V_{DD}
	PTB1 as T2CH1 of TIM2	Input/output	V_{DD}
	PTB2 as PPIECK; external clock source input for PPI	Input	V_{DD}
	PTB2 as T1CH0 of TIM1	Input/output	V_{DD}
	PTB3 as T1CH1 of TIM1	Input/output	V_{DD}
	PTB6–PTB7 as LCD frontplane drivers, FP1–FP2	Output	V_{DD}

Continued on next page

Table 1-1. Pin Functions (Continued)

Pin Name	Pin Description	Input/Output	Voltage Level
PTC0/FP19 PTC1/FP20 PTC2/FP21 PTC3/FP22 PTC4/FP23 PTC5/FP24 PTC6 PTC7	8-bit general-purpose I/O port	Input/output	V_{DD}
	PTC0–PTC5 as LCD frontplane drivers, FP19–FP24	Output	V_{DD}
PTD0/FP11 PTD1/FP12 PTD2/FP13 PTD3/FP14 PTD4/FP15 PTD5/FP16 PTD6/FP17 PTD7/FP18	8-bit general-purpose I/O port	Input/output	V_{DD}
	PTD0–PTD7 as LCD frontplane drivers, FP11–FP18	Output	V_{DD}
PTE0/FP3 PTE1/FP4 PTE2/FP5 PTE3/FP6 PTE4/FP7 PTE5/FP8 PTE6/FP9 PTE7/FP10	8-bit general-purpose I/O port	Input/output	V_{DD}
	PTE0–PTE7 as LCD frontplane drivers, FP3–FP10	Output	V_{DD}

Chapter 2
Memory

2.1 Introduction

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

- 8,192 bytes of user read-only memory (ROM)
- 128 bytes of random-access memory (RAM)
- 48 bytes of user-defined vectors
- 960 bytes of monitor ROM

2.2 I/O Section

Addresses $0000–$007F, shown in Figure 2-2, contain most of the control, status, and data registers.

Additional I/O registers have the following addresses:

- $FE0F; Low-voltage inhibit status register, LVISR
- $FFFF; COP control register, COPCTL

2.3 Monitor ROM

The 350 bytes at addresses $FE20–$FF7D are reserved ROM addresses that contain the instructions for the monitor functions.

$0000 ↓ $007F	I/O REGISTERS 128 BYTES
$0080 ↓ $00FF	RAM 128 BYTES
$0100 ↓ $0B96	UNIMPLEMENTED 2,711 BYTES
$0B97 ↓ $0DEF	RESERVED FOR MONITOR ROM 601 BYTES
$0DF0 ↓ $DDFF	UNIMPLEMENTED 53,264 BYTES
$DE00 ↓ $FDFF	User ROM 8,192 BYTES
$FE00 ↓ $FE0E	SYSTEM REGISTERS 15 BYTES
$FE0F	LVI STATUS REGISTER (LVISR)
$FE10 ↓ $FE1F	UNIMPLEMENTED 16 BYTES
$FE20 ↓ $FF7D	MONITOR ROM 350 BYTES
$FF7E	UNIMPLEMENTED
$FF7F ↓ $FF96	MONITOR JUMP TABLE 24 BYTES
$FF97 ↓ $FFCF	UNIMPLEMENTED 57 BYTES
$FFD0 ↓ $FFFF	USER ROM VECTORS 48 BYTES

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:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PTB)	Read:	PTB7	PTB6	0	0	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PTC)	Read:	PTC7	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
		Write:								
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PTD)	Read:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Write:								
		Reset:	Unaffected by reset							
$0004	Data Direction Register A (DDRA)	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB)	Read:	DDRB7	DDRB6	0	0	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0006	Data Direction Register C (DDRC)	Read:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007	Data Direction Register D (DDRD)	Read:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0008	Data Direction Register E (DDRE)	Read:	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0009	Port E Data Register (PTE)	Read:	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0
		Write:								
		Reset:	Unaffected by reset							
$000A	Unimplemented									

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 1 of 7)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000B	Unimplemented									
$000C	Port-B High Current Drive Control Register (HDB)	Read:	0	0	0	0	HDB3	HDB2	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000D ↓ $0018	Unimplemented									
$0019	PPI1 Status and Control Register (PPI1SCR)	Read:	PPI1L	PPI1MSK	PPICLKS1	PPI1CLKS0	0	PPI1IE2	PPI1IE1	PPI1IE0
		Write:								
		Reset:	0	0	1	0	0	0	0	0
$001A	Unimplemented									
$001B	Keyboard Status and Control Register (KBSCR)	Read:	R	R	R	R	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$001C	Keyboard Interrupt Enable Register (KBIER)	Read:	0	0	0	0	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001D	Configuration Register 2 (CONFIG2)(1)	Read:	STOP_ XCLKEN	STOP_ XTALEN	PEE	PDE	PCEH	PCEL	LVISEL1	LVISEL0
		Write:								
		Reset:	0	0	0	0	0	0	0(2)	1(2)
$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 1 (CONFIG1)(1)	Read:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	R	SSREC	STOP	COPD
		Write:								
		Reset:	0	0	0	0(2)	0	0	0	0

1. One-time writable register after each reset.
2. One time writable after each POR and reset by POR only.

U = Unaffected X = Indeterminate [shaded] = Unimplemented R = Reserved

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

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0020	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
$0021	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
$0022	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
$0023	Timer 1 Counter Modulo Register High (T1MODH)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
$0024	Timer 1 Counter Modulo Register Low (T1MODL)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$0025	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
$0026	Timer 1 Channel 0 Register High (T1CH0H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							
$0027	Timer 1 Channel 0 Register Low (T1CH0L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							
$0028	Timer 1 Channel 1 Status and Control Register (T1SC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0029	Timer 1 Channel 1 Register High (T1CH1H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							
$002A	Timer 1 Channel 1 Register Low (T1CH1L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							

U = Unaffected   X = Indeterminate   [shaded] = Unimplemented   R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 3 of 7)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$002B	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
$002C	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
$002D	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
$002E	Timer 2 Counter Modulo Register High (T2MODH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$002F	Timer 2 Counter Modulo Register Low (T2MODL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0030	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
$0031	Timer 2 Channel 0 Register High (T2CH0H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0032	Timer 2 Channel 0 Register Low (T2CH0L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0033	Timer 2 Channel 1 Status and Control Register (T2SC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0034	Timer 2 Channel 1 Register High (T2CH1H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0035	Timer 2 Channel 1 Register Low (T2CH1L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							

U = Unaffected X = Indeterminate [shaded] = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 7)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0036 ↓ $004E	Unimplemented									
$004F	LCD Clock Register (LCDCLK)	Read:	0	FCCTL1	FCCTL0	DUTY1	DUTY0	LCLK2	LCLK1	LCLK0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0050	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:								
$0051	LCD Control Register (LCDCR)	Read:	LCDE	0	FC	LC	LCCON3	LCCON2	LCCON1	LCCON0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0052	LCD Data Register (LDAT1)	Read:	F1B3	F1B2	F1B1	F1B0	F0B3	F0B2	F0B1	F0B0
		Write:								
		Reset:	U	U	U	U	U	U	U	U
$0053	LCD Data Register (LDAT2)	Read:	F3B3	F3B2	F3B1	F3B0	F2B3	F2B2	F2B1	F2B0
		Write:								
		Reset:	U	U	U	U	U	U	U	U
$0054	LCD Data Register (LDAT3)	Read:	F5B3	F5B2	F5B1	F5B0	F4B3	F4B2	F4B1	F4B0
		Write:								
		Reset:	U	U	U	U	U	U	U	U
$0055	LCD Data Register (LDAT4)	Read:	F7B3	F7B2	F7B1	F7B0	F6B3	F6B2	F6B1	F6B0
		Write:								
		Reset:	U	U	U	U	U	U	U	U
$0056	LCD Data Register (LDAT5)	Read:	F9B3	F9B2	F9B1	F9B0	F8B3	F8B2	F8B1	F8B0
		Write:								
		Reset:	U	U	U	U	U	U	U	U
$0057	LCD Data Register (LDAT6)	Read:	F11B3	F11B2	F11B1	F11B0	F10B3	F10B2	F10B1	F10B0
		Write:								
		Reset:	U	U	U	U	U	U	U	U
$0058	LCD Data Register (LDAT7)	Read:	F13B3	F13B2	F13B1	F13B0	F12B3	F12B2	F12B1	F12B0
		Write:								
		Reset:	U	U	U	U	U	U	U	U

U = Unaffected X = Indeterminate [shaded] = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 5 of 7)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0059	LCD Data Register (LDAT8)	Read: Write:	F15B3	F15B2	F15B1	F15B0	F14B3	F14B2	F14B1	F14B0
		Reset:	U	U	U	U	U	U	U	U
$005A	LCD Data Register (LDAT9)	Read: Write:	F17B3	F17B2	F17B1	F17B0	F16B3	F16B2	F16B1	F16B0
		Reset:	U	U	U	U	U	U	U	U
$005B	LCD Data Register (LDAT10)	Read: Write:	F19B3	F19B2	F19B1	F19B0	F18B3	F18B2	F18B1	F18B0
		Reset:	U	U	U	U	U	U	U	U
$005C	LCD Data Register (LDAT11)	Read: Write:	F21B3	F21B2	F21B1	F21B0	F20B3	F20B2	F20B1	F20B0
		Reset:	U	U	U	U	U	U	U	U
$005D	LCD Data Register (LDAT12)	Read: Write:	F23B3	F23B2	F23B1	F23B0	F22B3	F22B2	F22B1	F22B0
		Reset:	U	U	U	U	U	U	U	U
$005E	LCD Data Register (LDAT13)	Read:	0	0	0	0	F24B3	F24B2	F24B1	F24B0
		Write:								
		Reset:	0	0	0	0	U	U	U	U
$005F ↓ $007F	Unimplemented									
$FE00	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.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE01	Reset Status Register (SRSR)	Read:	POR	PIN	COP	ILOP	ILAD	0	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
$FE02	Reserved		R	R	R	R	R	R	R	R

U = Unaffected   X = Indeterminate   (shaded) = Unimplemented   R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 6 of 7)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE03	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:	0	0	0	0	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE06	Interrupt Status Register 3 (INT3)	Read:	0	0	0	0	0	0	IF16	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE07 ↓ $FE0B	Reserved		R	R	R	R	R	R	R	R
$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
$FE0F	Low-Voltage Inhibit Status Register (LVISR)	Read:	LVIOUT	LVIIE	LVIIF	0	0	0	0	0
		Write:				LVIIAK				
		Reset:	0	0	0	0	0	0	0	0
$FFFF	COP Control Register (COPCTL)	Read:	Low byte of reset vector							
		Write:	Writing clears COP counter (any value)							
		Reset:	Unaffected by reset							

U = Unaffected X = Indeterminate [shaded] = Unimplemented R = Reserved

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

Table 2-1. Vector Addresses

Vector Priority	INT Flag	Address	Vector
Lowest ↑	IF16	$FFDC	KBI
		$FFDD	
	—	$FFDE ↓ $FFE9	Not used
	IF10	$FFE8	PPI1
		$FFE9	
	IF9	$FFEA	TIM2 overflow
		$FFEB	
	IF8	$FFEC	TIM2 channel 1
		$FFED	
	IF7	$FFEE	TIM2 channel 0
		$FFEF	
	IF6	$FFF0	TIM1 overflow
		$FFF1	
	IF5	$FFF2	TIM1 channel 1
		$FFF3	
	IF4	$FFF4	TIM1 channel 0
		$FFF5	
	IF3	$FFF6	Not used
		$FFF7	
	IF2	$FFF8	LVI
		$FFF9	
	IF1	$FFFA	$\overline{\text{IRQ}}$
		$FFFB	
	—	$FFFC	SWI
		$FFFD	
↓	—	$FFFE	Reset
Highest		$FFFF	

2.4 Random-Access Memory (RAM)

The 512 bytes RAM are located from $0080 through $027F. The location of the stack RAM is programmable. The 16-bit stack pointer allows the stack to be anywhere in the 64-Kbyte 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, direct addressing mode instructions can access efficiently 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.

2.5 Read-Only Memory (ROM)

The 8,192 bytes user ROM are located from $DE00 through $FDFF, plus a block of 48 bytes for user interrupt vectors from $FFD0 through $FFFF.

Chapter 3
Configuration Register (CONFIG)

3.1 Introduction

This section describes the configuration registers, CONFIG1 and CONFIG2.

The configuration registers enable or disable these options:

- Computer operating properly module (COP)
- COP timeout period ($2^{13}–2^4$ or $2^{18}–2^4$ CGMXCLK cycles)
- Crystal oscillators during stop mode
- Low voltage inhibit (LVI) module power
- LVI module reset
- LVI module in stop mode
- LVI module voltage trip point selection
- STOP instruction
- Stop mode recovery time (32 or 4096 CGMXCLK cycles)
- LCD frontplanes FP3–FP10 on port E
- LCD frontplanes FP11–FP18 on port D
- LCD frontplanes FP19–FP24 on port C

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001D	Configuration Register 2 (CONFIG2)(1)	Read: / Write:	STOP_ XCLKEN	STOP_ XTALEN	PEE	PDE	PCEH	PCEL	LVISEL1	LVISEL0
		Reset:	0	0	0	0	0	0	0(2)	1(2)
$001F	Configuration Register 1 (CONFIG1)(1)	Read: / Write:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	R	SSREC	STOP	COPD
		Reset:	0	0	0	0(2)	0	0	0	0

1. One-time writable register after each reset.
2. LVIT1, LVIT0, and LVIPWRD reset to 0 by a power-on reset (POR) only.

R = Reserved

Figure 3-1. CONFIG Registers Summary

3.2 Functional Description

The configuration registers are used in the initialization of various options. The configuration registers 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 these registers be written immediately after reset. The configuration registers are located at $001D and $001F. The configuration registers may be read at anytime.

NOTE

The options except LVIT[1:0] and LVIPWRD are one-time writable by the user after each reset. The LVIT[1:0] and LVIPWRD bits are one-time writable by the user only after each POR (power-on reset). The CONFIG registers are not in the FLASH memory but are special registers containing one-time writable latches after each reset. Upon a reset, the CONFIG registers default to predetermined settings as shown in Figure 3-2 and Figure 3-3.

The mask option register (MOR) is used to select the oscillator option for the MCU: crystal oscillator or RC oscillator. The MOR is implemented as a byte in FLASH memory. Hence, writing to the MOR requires programming the byte.

3.3 Configuration Register 1 (CONFIG1)

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	R	SSREC	STOP	COPD
Reset:	0	0	0	U	0	0	0	0
POR:	0	0	0	0	0	0	0	0

R = Reserved   U = Unaffected

Figure 3-2. Configuration Register 1 (CONFIG1)

COPRS — COP Rate Select

COPRS selects the COP time-out period. Reset clears COPRS.

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

LVISTOP — Low Voltage Inhibit Enable in Stop Mode

When the LVIPWRD bit is clear, setting the LVISTOP bit enables the LVI to operate during stop mode. Reset clears LVISTOP.

1 = LVI enabled during stop mode
0 = LVI disabled during stop mode

LVIRSTD — Low Voltage Inhibit Reset Disable

LVIRSTD disables the reset signal from the LVI module. Reset clears LVIRSTOP.

1 = LVI module reset disabled
0 = LVI module reset enabled

LVIPWRD — Low Voltage Inhibit Power Disable

LVIPWRD disables the LVI module. This bit is reset to 0 by a POR only.

1 = LVI module disabled
0 = LVI module enabled

NOTE

Exiting stop mode by pulling reset will result in the long stop recovery. If using an external crystal, do not set the SSREC bit.

SSREC — Short Stop Recovery Bit

SSREC enables the CPU to exit stop mode with a delay of 32 CGMXCLK cycles instead of a 4096 CGMXCLK cycle delay.

1 = Stop mode recovery after 32 CGMXCLK cycles
0 = Stop mode recovery after 4096 CGMXCLK 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. Reset clears COPD.

1 = COP module disabled
0 = COP module enabled

3.4 Configuration Register 2 (CONFIG2)

Address: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	STOP_ XCLKEN	STOP_ XTALEN	PEE	PDE	PCEH	PCEL	LVISEL1	LVISEL0
Reset:	0	0	0	0	0	0	U	U
POR:	0	0	0	0	0	0	0	1

R = Reserved   U = Unaffected

Figure 3-3. Configuration Register 2 (CONFIG2)

STOP_XCLKEN — Crystal Oscillator Stop Mode Enable (OSC)

Setting STOP_XCLKEN enables the crystal oscillator on OSC1 and OSC2 to continue operating during stop mode. Reset clears this bit.

1 = Crystal oscillator enabled on OSC pins during stop mode
0 = Crystal oscillator disabled on OSC pins during stop mode

STOP_XTALEN — Crystal Oscillator Stop Mode Enable (XTAL)

Setting STOP_XTALEN enables the crystal oscillator on XTAL1 and XTAL2 to continue operating during stop mode. Reset clears this bit.

1 = Crystal oscillator enabled on XTAL pins during stop mode
0 = Crystal oscillator disabled on XTAL pins during stop mode

PEE — Port E Enable for LCD Drive

Setting PEE configures the PTE0/FP3–PTE7/FP10 pins for LCD frontplane driver use. Reset clears this bit.

1 = PTE0/FP3–PTE7/FP10 pins configured as LCD frontplane driver pins: FP3–FP10
0 = PTE0/FP3–PTE7/FP10 pins configured as standard I/O pins: PTE0–PTE7

PDE — Port D Enable for LCD Drive

Setting PDE configures the PTD0/FP11–PTD7/FP18 pins for LCD frontplane driver use. Reset clears this bit.

1 = PTD0/FP11–PTD7/FP18 pins configured as LCD frontplane driver pins: FP11–FP18
0 = PTD0/FP11–PTD7/FP18 pins configured as standard I/O pins: PTD0–PTD7

PCEH — Port C High Nibble Enable for LCD Drive

Setting PCEH configures the PTC4/FP23–PTC5/FP24 pins for LCD frontplane driver use. Reset clears this bit.

1 = PTC4/FP23–PTC5/FP24 pins configured as LCD frontplane driver pins: FP23–FP24
0 = PTC4/FP23–PTC5/FP24 pins configured as standard I/O pins: PTC4–PTC5

PCEL — Port C Low Nibble Enable for LCD Drive

Setting PCEL configures the PTC0/FP19–PTC3/FP22 pins for LCD frontplane driver use. Reset clears this bit.

1 = PTC0/FP19–PTC3/FP22 pins configured as LCD frontplane driver pins: FP19–FP22
0 = PTC0/FP19–PTC3/FP22 pins configured as standard I/O pins: PTC0–PTC3

LVISEL1, LVISEL0 — LVI Trip Voltage Selection

These two bits determine at which level of V_{DD} the LVI module will come into action. LVISEL1 and LVISEL0 are set to the default configuration by a power-on reset only.

Table 3-1. Trip Voltage Selection

LVISEL1	LVISEL0	Comments[(1)]
0	0	Reserved
0	1	For V_{DD} = 3 V operation (default after POR)
1	0	For V_{DD} = 5 V operation
1	1	Reserved

1. See Chapter 16 Electrical Specifications for full parameters.

Chapter 4
System Integration Module (SIM)

4.1 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. 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 4-1. Signal Name Conventions

Signal Name	Description
CGMXCLK	Oscillator clock from oscillator module
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

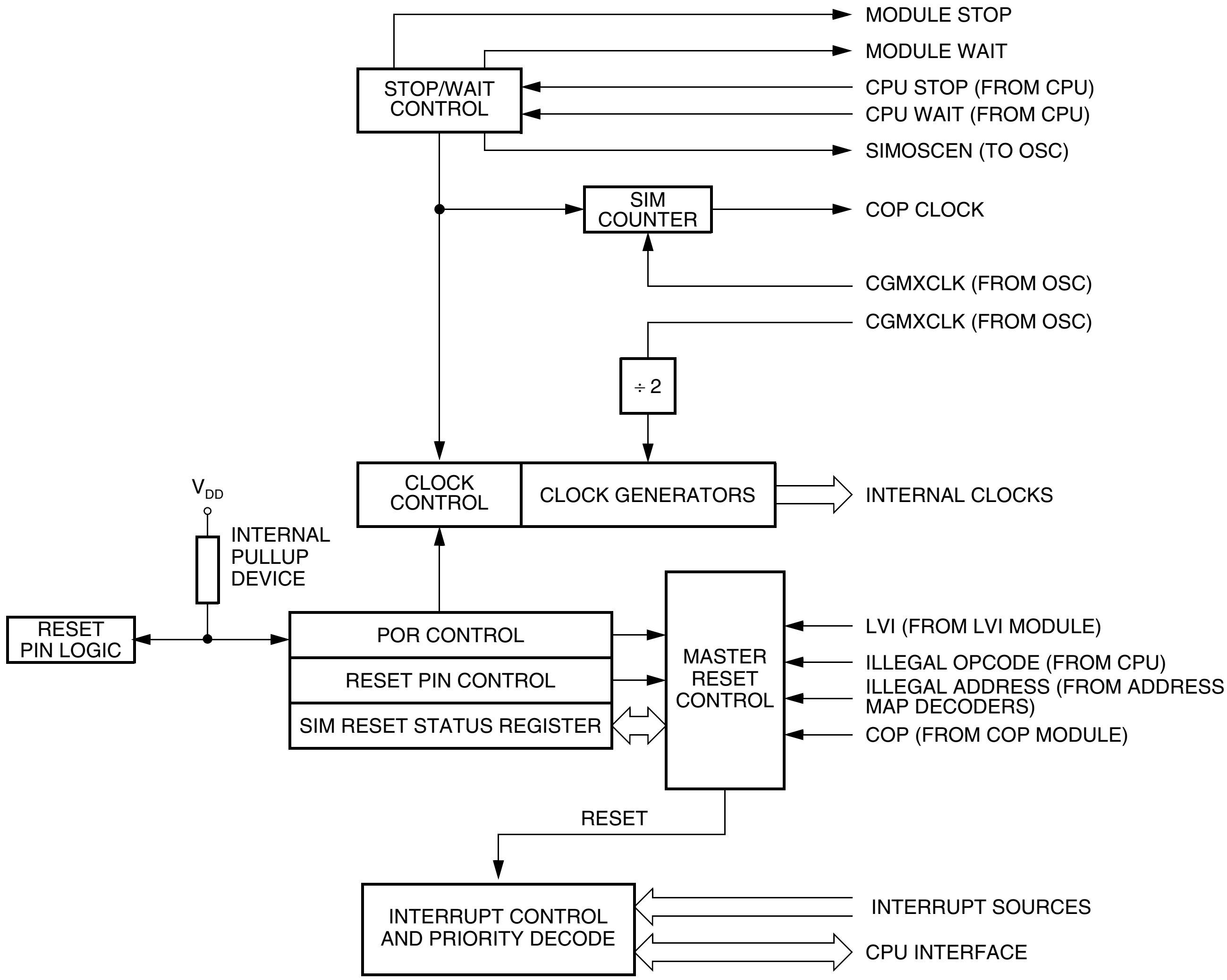


Figure 4-1. SIM Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE00	Break Status Register (BSR)	Read:	R	R	R	R	R	R	SBSW	R
		Write:							NOTE	
		Reset:	0	0	0	0	0	0	0	0
Note: Writing a 0 clears SBSW.										
$FE01	Reset Status Register (RSR)	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
$FE02	Reserved		R	R	R	R	R	R	R	R
$FE03	Break Flag Control Register (BFCR)	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							

Figure 4-2. SIM I/O Register Summary

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 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:	0	0	0	0	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE06	Interrupt Status Register 3 (INT3)	Read:	0	0	0	0	0	0	IF16	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented   R = Reserved

Figure 4-2. SIM I/O Register Summary (Continued)

4.2 SIM Bus Clock Control and Generation

The bus clock generator provides system clock signals for the CPU and peripherals on the MCU. The system clocks are generated from an incoming clock, CGMXCLK as shown in Figure 4-3. This clock can come from either the oscillator module or from the on-chip PLL. (See Chapter 5 Oscillator (OSC).)

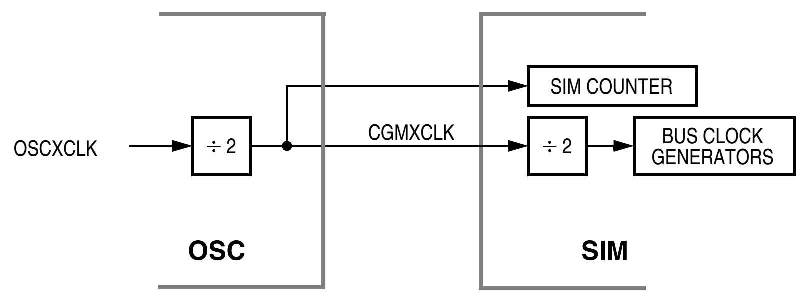


Figure 4-3. SIM Clock Signals

4.2.1 Bus Timing

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

4.2.2 Clock Start-up from POR or LVI Reset

When the power-on reset 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 CGMXCLK cycle POR timeout has completed. The $\overline{RST}$ pin is driven low by the SIM during this entire period. The IBUS clocks start upon completion of the timeout.

4.2.3 Clocks in Stop Mode and Wait Mode

Upon exit from stop mode by an interrupt, break, or reset, the SIM allows CGMXCLK 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 32 CGMXCLK cycles. (See 4.6.2 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.

4.3 Reset and System Initialization

The MCU has these reset sources:

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

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 4.4 SIM Counter), but an external reset does not. Each of the resets sets a corresponding bit in the SIM reset status register (SRSR). (See 4.7 SIM Registers.)

4.3.1 External Pin Reset

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

Table 4-2. PIN Bit Set Timing

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

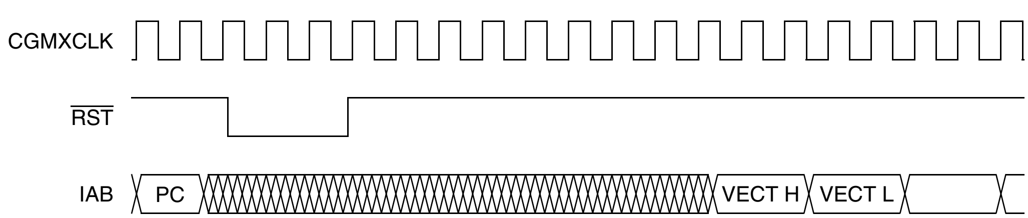


Figure 4-4. External Reset Timing

4.3.2 Active Resets from Internal Sources

All internal reset sources actively pull the $\overline{RST}$ pin low for 32 CGMXCLK cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles (see Figure 4-5). An internal reset can be caused by an illegal address, illegal opcode, COP timeout, LVI, or POR (see Figure 4-6).

NOTE

For LVI or POR resets, the SIM cycles through 4096 + 32 CGMXCLK 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 4-5.

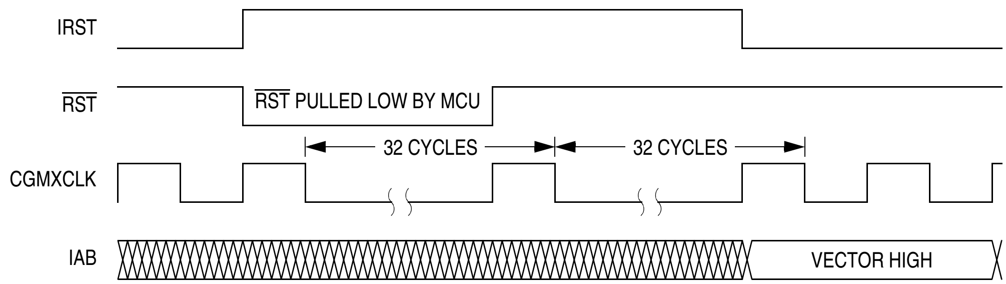


Figure 4-5. Internal Reset Timing

The COP reset is asynchronous to the bus clock.

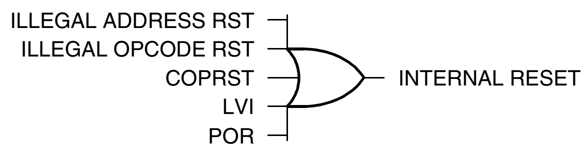


Figure 4-6. Sources of Internal Reset

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

4.3.2.1 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 ($\overline{\text{RST}}$) is held low while the SIM counter counts out 4096 + 32 CGMXCLK cycles. Thirty-two CGMXCLK cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur.

At power-on, these events occur:

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

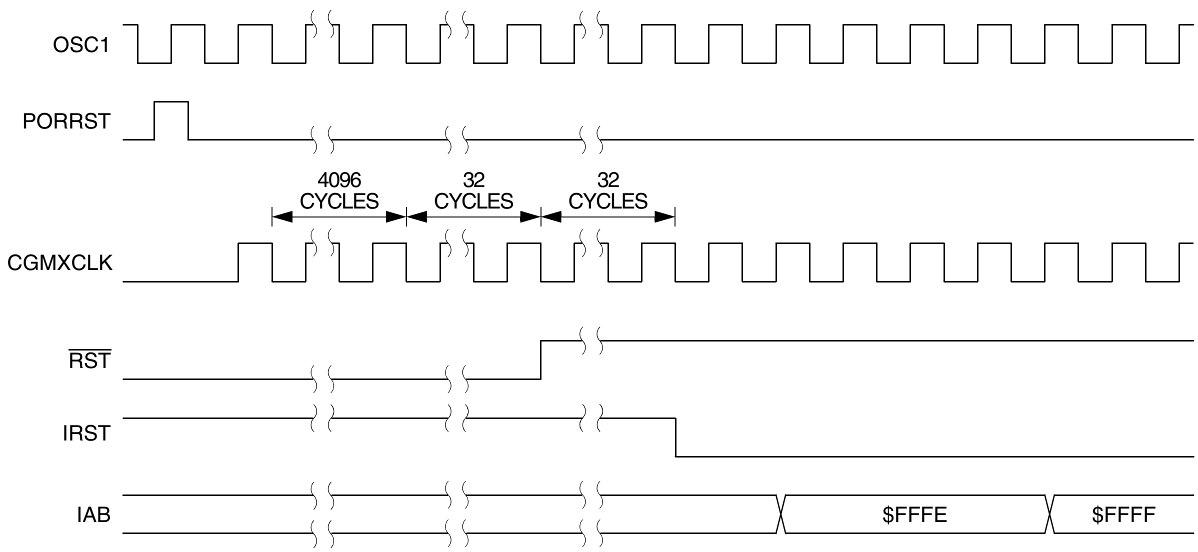


Figure 4-7. POR Recovery

4.3.2.2 Computer Operating Properly (COP) Reset

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

To prevent a COP module timeout, write any value to location $FFFF. Writing to location $FFFF clears the COP counter and bits 12 through 5 of the SIM counter. The SIM counter output, which occurs at least every $2^{13} - 2^4$ CGMXCLK 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 $\overline{RST}$ pin or the $\overline{IRQ}$ 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 $\overline{RST}$ or the $\overline{IRQ}$ pin. This prevents the COP from becoming disabled as a result of external noise. During a break state, V_{TST} on the $\overline{RST}$ pin disables the COP module.

4.3.2.3 Illegal Opcode Reset

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

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 $\overline{RST}$ pin for all internal reset sources.

4.3.2.4 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 $\overline{RST}$ pin for all internal reset sources.

4.3.2.5 Low-Voltage Inhibit (LVI) Reset

The low-voltage inhibit module (LVI) asserts its output to the SIM when the V_{DD} voltage falls to the LVI_{TRIPF} voltage. 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 + 32 CGMXCLK cycles. Thirty-two CGMXCLK 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.

4.3.2.6 Monitor Mode Entry Module Reset (MODRST)

The monitor mode entry module reset (MODRST) asserts its output to the SIM when monitor mode is entered in the condition where the reset vectors are blank ($FF). (See Chapter 15 Development Support.) When MODRST gets asserted, an internal reset occurs. The SIM actively pulls down the $\overline{RST}$ pin for all internal reset sources.

4.4 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 overflow supplies the clock for the COP module. The SIM counter is 12 bits long and is clocked by the falling edge of CGMXCLK.

4.4.1 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 module (OSC) to drive the bus clock state machine.

4.4.2 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 1 (CONFIG1). If the SSREC bit is a logic 1, then the stop recovery is reduced from the normal delay of 4096 CGMXCLK cycles down to 32 CGMXCLK cycles. This is ideal for applications using canned oscillators that do not require long start-up times from stop mode. External crystal applications should use the full stop recovery time, that is, with SSREC cleared.

4.4.3 SIM Counter and Reset States

External reset has no effect on the SIM counter. (See 4.6.2 Stop Mode for details.) The SIM counter is free-running after all reset states. (See 4.3.2 Active Resets from Internal Sources for counter control and internal reset recovery sequences.)

4.5 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

4.5.1 Interrupts

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. Figure 4-8 shows interrupt entry timing, and Figure 4-9 shows interrupt recovery timing.

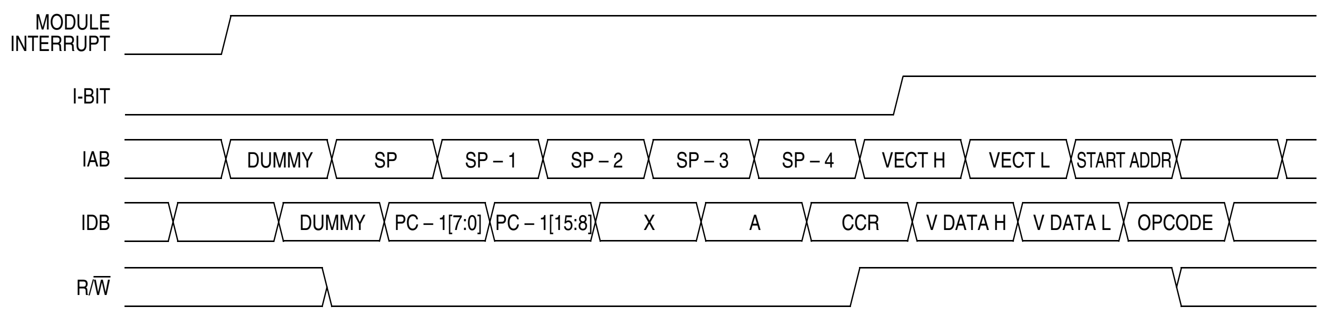


Figure 4-8. Interrupt Entry Timing

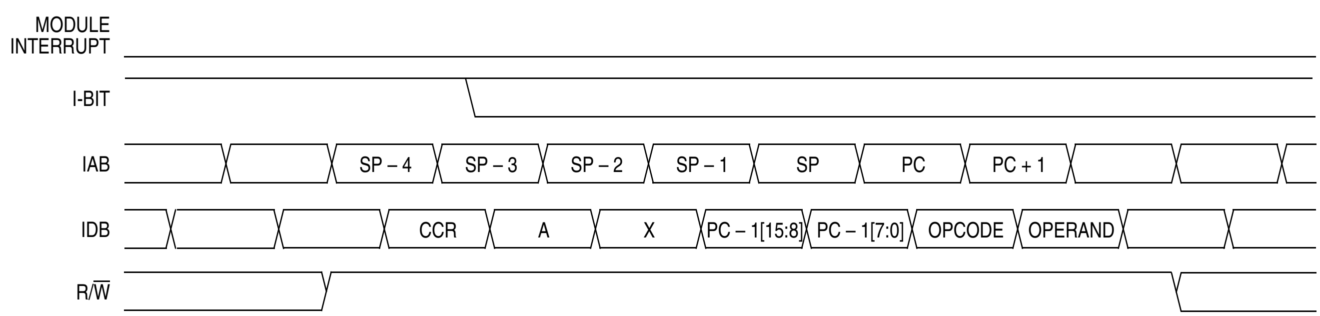


Figure 4-9. Interrupt Recovery Timing

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).
(See Figure 4-10.)

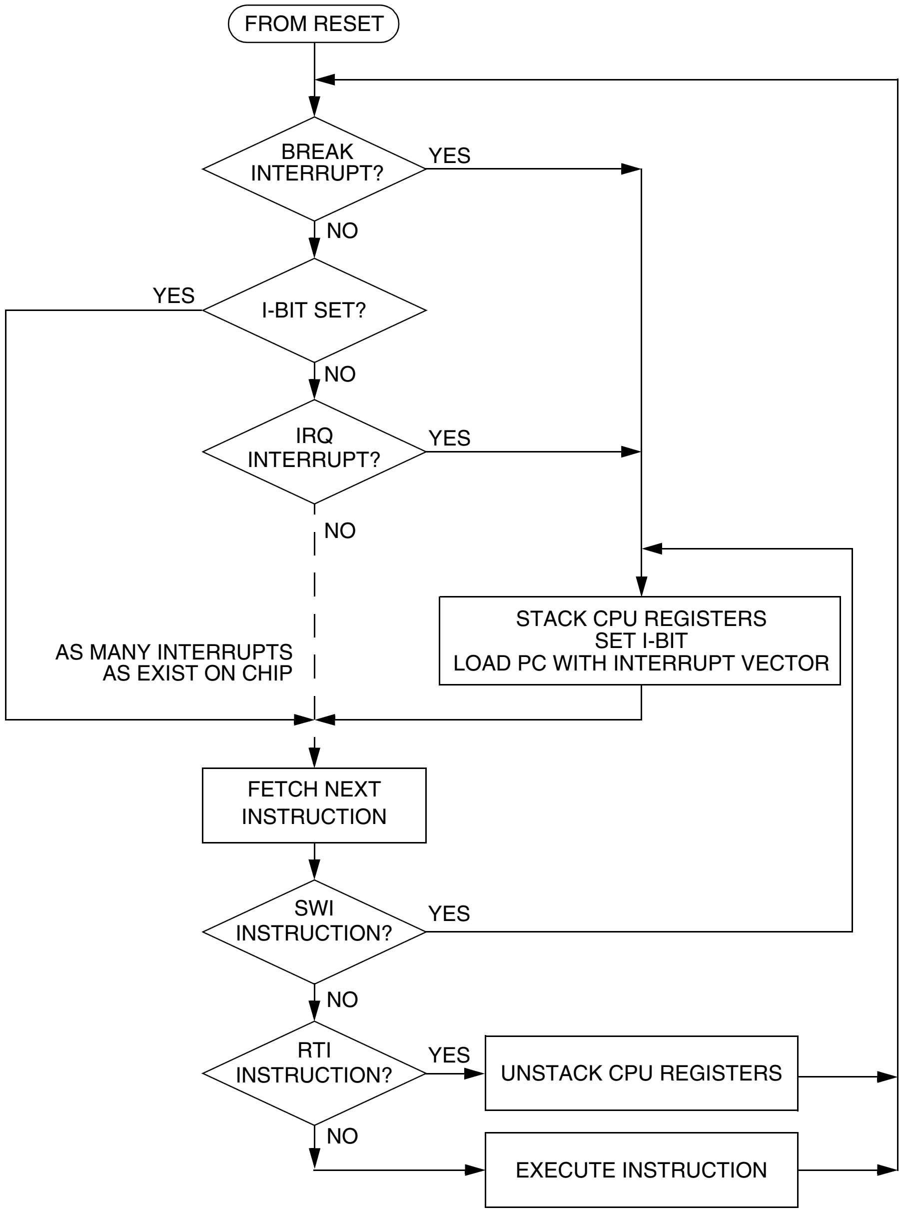


Figure 4-10. Interrupt Processing

4.5.1.1 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 4-11 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.

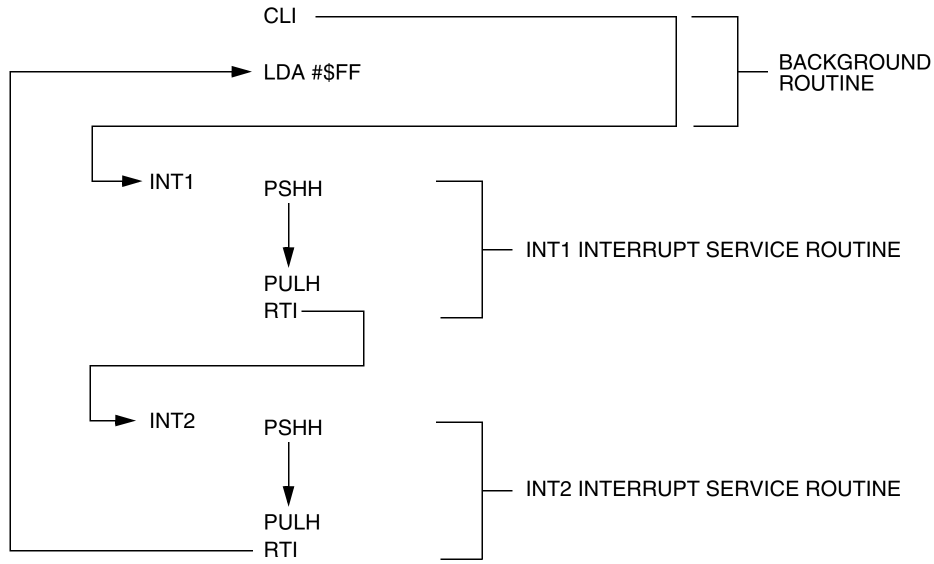


Figure 4-11. Interrupt Recognition Example

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.

4.5.1.2 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.

4.5.2 Interrupt Status Registers

The flags in the interrupt status registers identify maskable interrupt sources.

Table 2-1 summarizes the interrupt sources and the interrupt status register flags that they set.

4.5.2.1 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 4-12. Interrupt Status Register 1 (INT1)

IF6–IF1 — Interrupt Flags 1–6

These flags indicate the presence of interrupt requests from the sources shown in Table 2-1.

1 = Interrupt request present
0 = No interrupt request present

Bit 0 and Bit 1 — Always read 0

4.5.2.2 Interrupt Status Register 2

Address: $FE05

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	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 4-13. Interrupt Status Register 2 (INT2)

IF10–IF7 — Interrupt Flags 10–7

These flags indicate the presence of interrupt requests from the sources shown in Table 2-1.

1 = Interrupt request present
0 = No interrupt request present

4.5.2.3 Interrupt Status Register 3

Address: $FE06

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	IF16	0
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 4-14. Interrupt Status Register 3 (INT3)

IF16 — Interrupt Flag 16

This flag indicates the presence of an interrupt request from the source shown in Table 2-1.

1 = Interrupt request present
0 = No interrupt request present

4.5.3 Reset

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

4.5.4 Break Interrupts

The break module can stop normal program flow at a software-programmable break point by asserting its break interrupt output. (See Chapter 15 Development Support.) 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.

4.5.5 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 initialize the break clear flag enable bit (BCFE) in the SIM break flag control register (SBFCR).

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.

4.6 Low-Power Modes

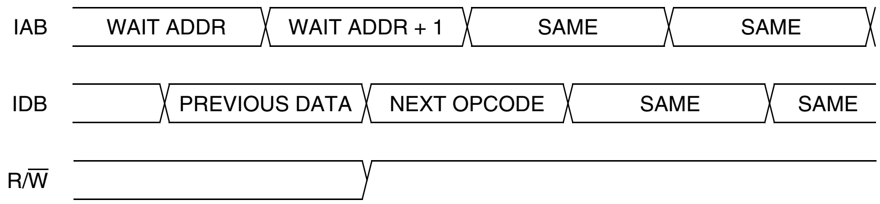
Executing the WAIT or STOP instruction puts the MCU in a low power-consumption mode for standby situations. The SIM holds the CPU in a non-clocked state. The operation of each of these modes is described in the following subsections. Both STOP and WAIT clear the interrupt mask (I) in the condition code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

4.6.1 Wait Mode

In wait mode, the CPU clocks are inactive while the peripheral clocks continue to run. Figure 4-15 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.

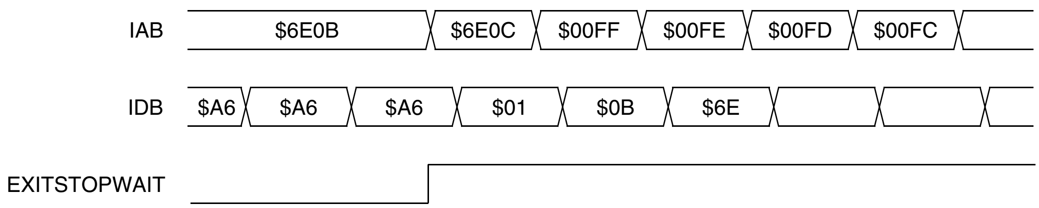
Wait mode also can be exited by a reset or break. A break interrupt during wait mode sets the SIM break stop/wait bit, SBSW, in the SIM break status register (SBSR). If the COP disable bit, COPD, in the mask option register is logic 0, then the computer operating properly module (COP) is enabled and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



NOTE: Previous data can be operand data or the WAIT opcode, depending on the last instruction.

Figure 4-15. Wait Mode Entry Timing

Figure 4-16 and Figure 4-17 show the timing for WAIT recovery.



NOTE: EXITSTOPWAIT = RST pin OR CPU interrupt OR break interrupt

Figure 4-16. Wait Recovery from Interrupt or Break

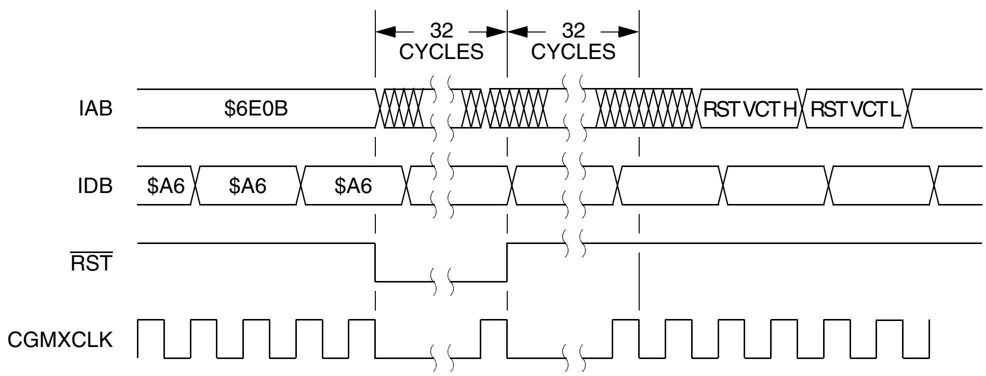


Figure 4-17. Wait Recovery from Internal Reset

4.6.2 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 output (CGMXCLK) in stop mode, stopping the CPU and peripherals. Stop recovery time is selectable using the SSREC bit in the configuration register 1 (CONFIG1). If SSREC is set, stop recovery is reduced from the normal delay of 4096 CGMXCLK cycles down to 32. This is ideal for applications using canned oscillators that do not require long start-up 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 4-18 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.

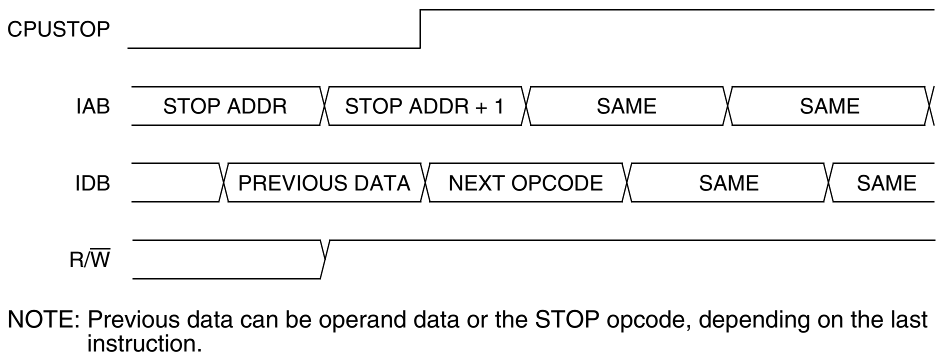


Figure 4-18. Stop Mode Entry Timing

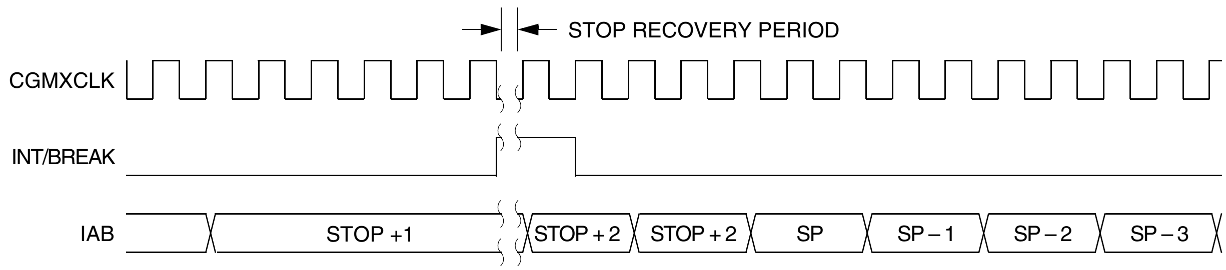


Figure 4-19. Stop Mode Recovery from Interrupt or Break

4.7 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)

4.7.1 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from stop mode 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	

R = Reserved

1. Writing a clears SBSW.

Figure 4-20. Break Status Register (BSR)

SBSW — Break Wait Bit

This status bit is set when a break interrupt causes an exit from wait mode or stop mode. 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 1 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
LOBYTE    EQU
          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
```

4.7.2 SIM Reset Status Register

This register contains six flags that show the source of the last reset provided all previous reset status bits have been cleared. Clear the SIM reset status register by reading it. A power-on reset sets the POR bit and clears all other bits in the register

Address: $FE01

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

= Unimplemented

Figure 4-21. Reset Status Register (RSR)

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 ($\overline{\text{RST}}$)
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

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

4.7.3 SIM Break Flag Control Register

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 4-22. Break Flag Control Register (BFCR)

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

BCFE — Break Clear Flag Enable Bit
This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.
1 = Status bits clearable during break
0 = Status bits not clearable during break

Chapter 5
Oscillator (OSC)

5.1 Introduction

The oscillator module provides the reference clocks for the MCU system and bus. Two oscillators are running on the device:

- 1–16 MHz crystal oscillator (OSC) — built-in oscillator that requires an external crystal or ceramic-resonator. This oscillator drives the main bus and other MCU subsystems except the LCD and PPI modules.
- 32.768kHz crystal oscillator (XTAL) — built-in oscillator that requires an external crystal or ceramic-resonator. This oscillator drives the LCD and PPI modules.

5.2 Functional Overview

The X-tal oscillator circuit is designed for use with an external crystal or ceramic resonator to provide accurate clock source.

In its typical configuration, the X-tal oscillator is connected in a Pierce oscillator configuration, as shown in Figure 5-1. 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 & X_2
- Fixed capacitor, C_1 & C_3
- Tuning capacitor, C_2 & C_4 (can also be a fixed capacitor)
- Feedback resistor, R_{B1} & R_{B2}
- Series resistor, R_S (for XTAL1 and XTAL2 only)

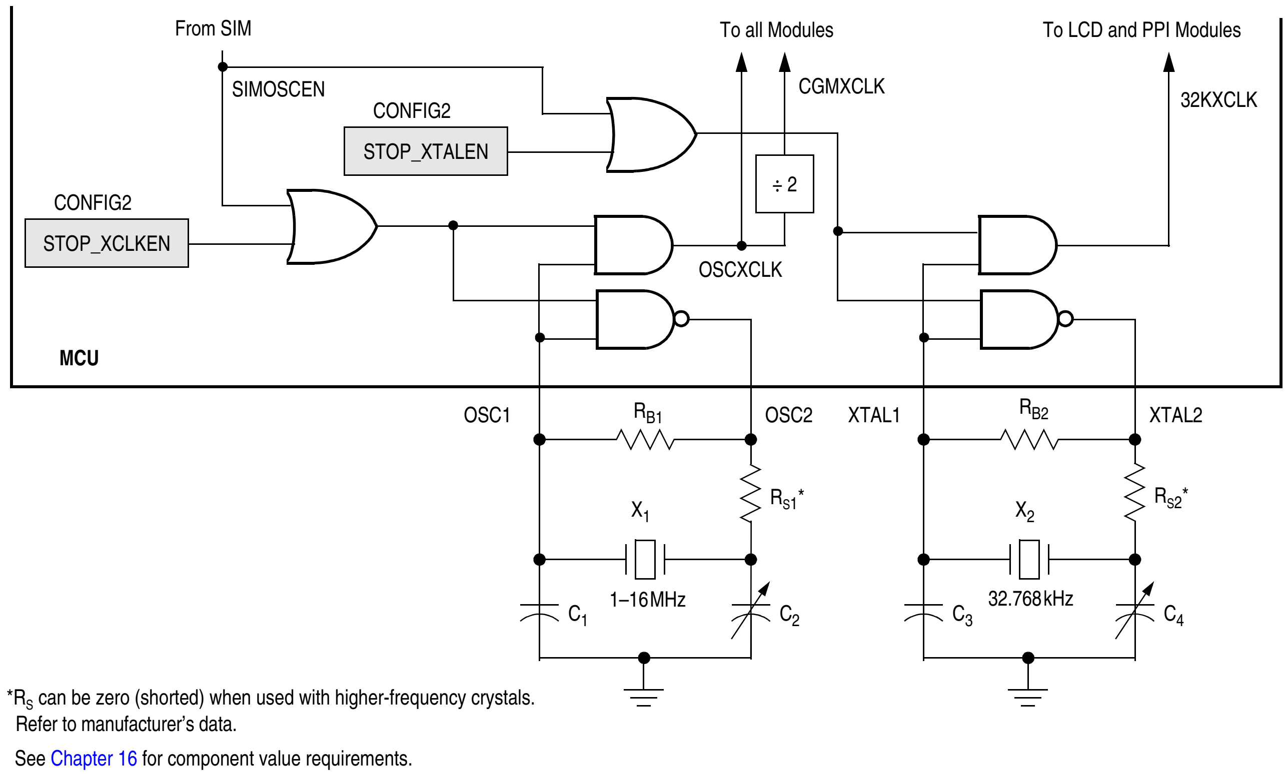


Figure 5-1. Oscillator External Connections

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.

5.3 I/O Signals

The following paragraphs describe the oscillator I/O signals.

5.3.1 Crystal Amplifier Input Pin (OSC1, XTAL1)

OSC1 and XTAL1 pin are input to the crystal oscillator amplifier. Schmitt trigger and glitch filter are implemented on this pin in order to improve the EMC performance. See Chapter 16 Electrical Specifications for detail specification of the glitch filter.

5.3.2 Crystal Amplifier Output Pin (OSC2 & XTAL2)

OSC2 or XTAL2 pin is the output of the crystal oscillator inverting amplifier.

5.3.3 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal from the system integration module (SIM) enables/disables the x-tal oscillator circuit.

5.4 Low Power Modes

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

5.4.1 Wait Mode

The WAIT instruction has no effect on the oscillator module.

5.4.2 Stop Mode

If STOP_XCLKEN = 1 CGMXCLK will keep on running during STOP mode else CGMXCLK will be stopped during STOP mode.

If STOP_XTALEN = 1 32KXCLK will keep on running during STOP mode else 32KXCLK will be stopped during STOP mode.

5.5 Oscillator During Break Mode

The oscillator will continue to drive CGMXCLK when the device enters the break state.

Chapter 6
Timer Interface Module (TIM)

6.1 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 6-1 is a block diagram of the TIM.

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

6.2 Features

Features of the TIM include:

- Two input capture/output compare channels:
  - 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 with 7-frequency internal bus clock prescaler selection
- Free-running or modulo up-count operation
- Toggle any channel pin on overflow
- TIM counter stop and reset bits

6.3 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]CH0 (timer channel 0) and T[1,2]CH1 (timer channel 1), where “1” is used to indicate TIM1 and “2” is used to indicate TIM2. The two TIMs share four I/O pins with four I/O port pins. The full names of the TIM I/O pins are listed in Table 6-1. The generic pin names appear in the text that follows.

Table 6-1. Pin Name Conventions

TIM Generic Pin Names:		**T[1,2]CH0**	**T[1,2]CH1**
Full TIM Pin Names:	TIM1	PTB2/T1CH0/PPIECK	PTB3/T1CH1
	TIM2	PTB0/T2CH0	PTB1/T2CH1

NOTE

References to either timer 1 or timer 2 may be made in the following text by omitting the timer number. For example, TCH0 may refer generically to T1CH0 and T2CH0, and TCH1 may refer to T1CH1 and T2CH1.

6.4 Functional Description

Figure 6-1 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.

The two TIM channels (per timer) are programmable independently as input capture or output compare channels.

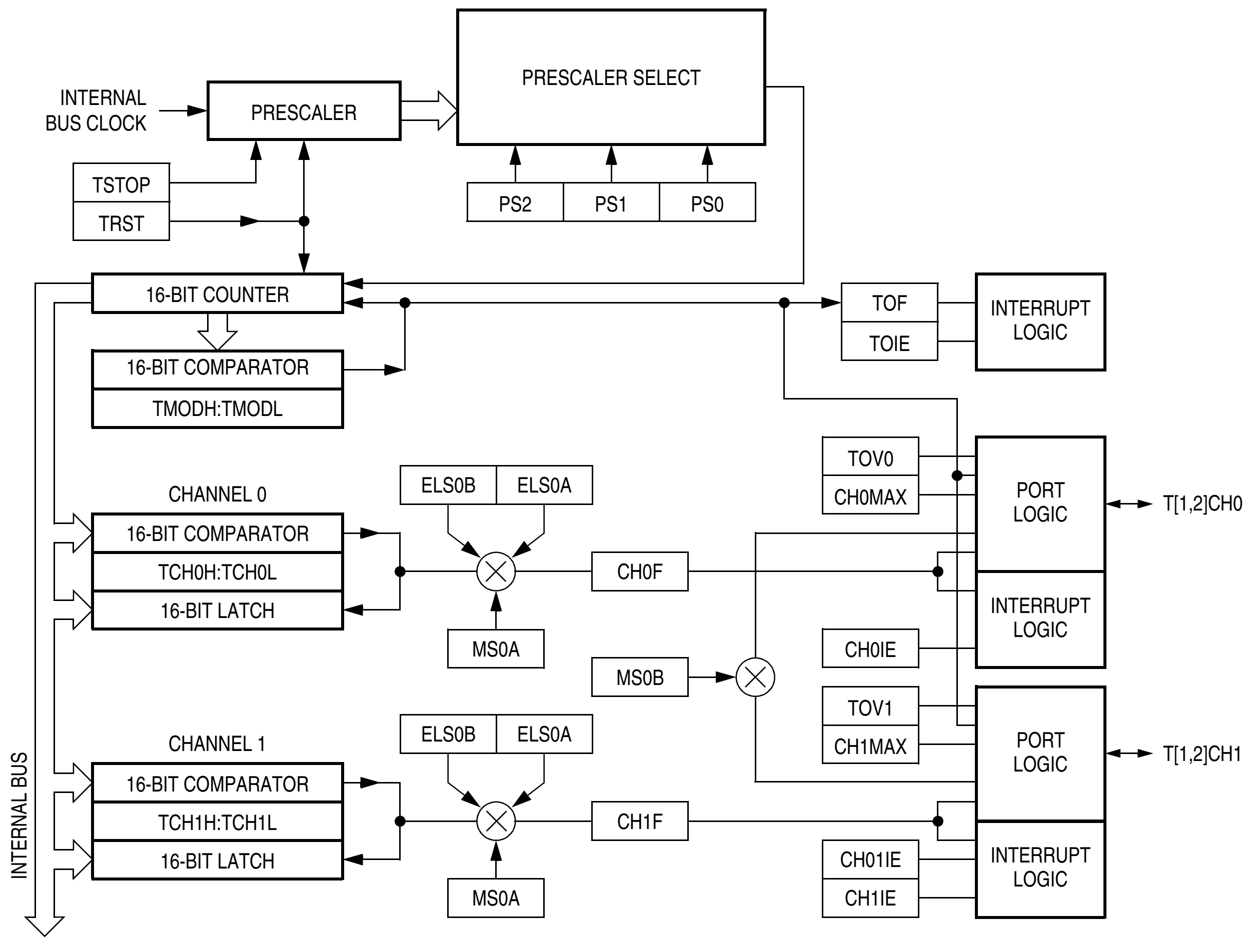


Figure 6-1. TIM Block Diagram

Figure 6-2 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
$0020	TIM1 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
$0021	TIM1 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
$0022	TIM1 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
$0023	TIM Counter Modulo Register High (TMODH)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
$0024	TIM1 Counter Modulo Register Low (T1MODL)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$0025	TIM1 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
$0026	TIM1 Channel 0 Register High (T1CH0H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							
$0027	TIM1 Channel 0 Register Low (T1CH0L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							
$0028	TIM1 Channel 1 Status and Control Register (T1SC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0029	TIM1 Channel 1 Register High (T1CH1H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	Indeterminate after reset							
$002A	TIM1 Channel 1 Register Low (T1CH1L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	Indeterminate after reset							
$002B	TIM2 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
$002C	TIM2 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
$002D	TIM2 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
$002E	TIM2 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

(shaded) = Unimplemented

Figure 6-2. TIM I/O Register Summary (Sheet 1 of 2)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$002F	TIM2 Counter Modulo Register Low (T2MODL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0030	TIM2 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
$0031	TIM2 Channel 0 Register High (T2CH0H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0032	TIM2 Channel 0 Register Low (T2CH0L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0033	TIM2 Channel 1 Status and Control Register (T2SC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0034	TIM2 Channel 1 Register High (T2CH1H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0035	TIM2 Channel 1 Register Low (T2CH1L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							

= Unimplemented

Figure 6-2. TIM I/O Register Summary (Sheet 2 of 2)

6.4.1 TIM Counter Prescaler

The TIM clock source can be one of the seven prescaler outputs. 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 select the TIM clock source.

6.4.2 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.

6.4.3 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.

6.4.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in 6.4.3 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.

6.4.3.2 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.

6.4.4 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 6-3 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 6.9.1 TIM Status and Control Register.

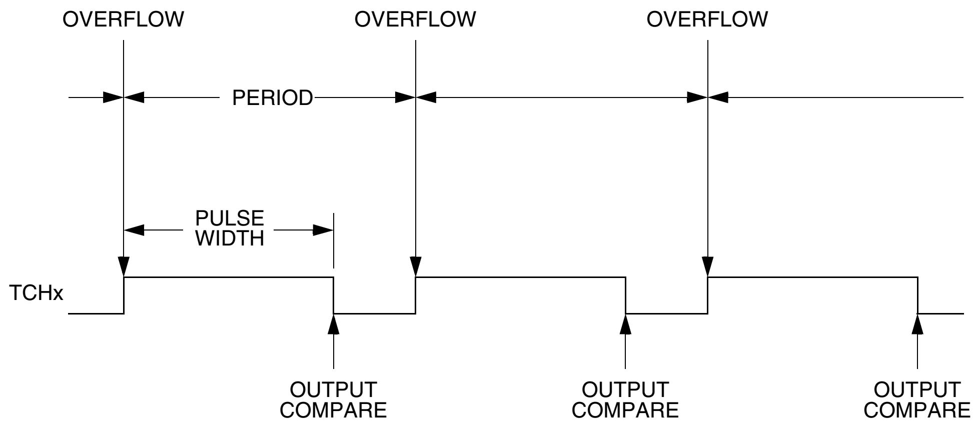


Figure 6-3. PWM Period and Pulse Width

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%.

6.4.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in 6.4.4 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.

6.4.4.2 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.

6.4.4.3 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 6-3.)
   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 6-3.)

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 6.9.4 TIM Channel Status and Control Registers.)

6.5 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.

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 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.

6.6.2 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.

6.7 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 break flag control register (BFCR) enables software to clear status bits during the break state. (See 21.5.4 SIM Break Flag Control Register.)

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.

6.8 I/O Signals

Port B shares four of its pins with the TIM. The four TIM channel I/O pins are T1CH0, T1CH1, T2CH0, and T2CH1 as described in 6.3 Pin Name Conventions.

Each channel I/O pin is programmable independently as an input capture pin or an output compare pin. T1CH0 and T2CH0 can be configured as buffered output compare or buffered PWM pins.

6.9 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)

6.9.1 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, $0020 and T2SC, $002B

<table>
<tr><th></th><th>Bit 7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>Bit 0</th></tr>
<tr><td>Read:</td><td>TOF</td><td rowspan="2">TOIE</td><td rowspan="2">TSTOP</td><td>0</td><td>0</td><td rowspan="2">PS2</td><td rowspan="2">PS1</td><td rowspan="2">PS0</td></tr>
<tr><td>Write:</td><td>0</td><td>TRST</td><td>(unimplemented)</td></tr>
<tr><td>Reset:</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

(shaded) = Unimplemented

Figure 6-4. 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 6-2 shows. Reset clears the PS[2:0] bits.

Table 6-2. 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	Not available

6.9.2 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, $0021 and T2CNTH, $002C

	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

Figure 6-5. TIM Counter Registers High (TCNTH)

Address: T1CNTL, $0022 and T2CNTL, $002D

	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

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

6.9.3 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, $0023 and T2MODH, $002E

	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 6-7. TIM Counter Modulo Register High (TMODH)

Address: T1MODL, $0024 and T2MODL, $002F

	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 6-8. TIM Counter Modulo Register Low (TMODL)

NOTE

Reset the TIM counter before writing to the TIM counter modulo registers.

6.9.4 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, $0025 and T2SC0, $0030

	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 6-9. TIM Channel 0 Status and Control Register (TSC0)

Address: T1SC1, $0028 and T2SC1, $0033

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

Figure 6-10. 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

MSxB — Mode Select Bit B

This read/write bit selects buffered output compare/PWM operation. MSxB 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 and reverts TCH1 to general-purpose I/O.
Reset clears the MSxB 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 6-3.
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 6-3. 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 (TSC).

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 6-3 shows how ELSxB and ELSxA work. Reset clears the ELSxB and ELSxA bits.

Table 6-3. Mode, Edge, and Level Selection

MSxB: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	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

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 6-11 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.

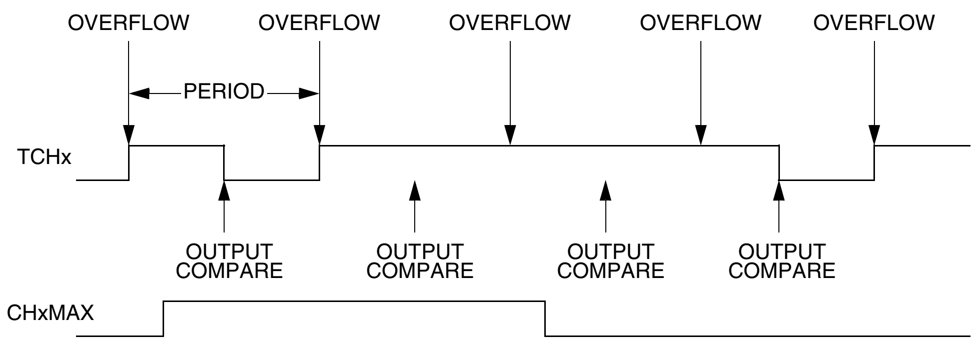


Figure 6-11. CHxMAX Latency

6.9.5 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, $0026 and T2CH0H, $0031

	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 6-12. TIM Channel 0 Register High (TCH0H)

Address: T1CH0L, $0027 and T2CH0L $0032

	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 6-13. TIM Channel 0 Register Low (TCH0L)

Address: T1CH1H, $0029 and T2CH1H, $0034

	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 6-14. TIM Channel 1 Register High (TCH1H)

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

	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 6-15. TIM Channel 1 Register Low (TCH1L)

Chapter 7
Programmable Periodic Interrupt (PPI)

7.1 Introduction

This section describes the programmable periodic interrupt (PPI) module. The PPI will generate periodic interrupts at user selectable rates using a counter clocked by the selected clock.

7.2 Features

Features of the PPI include:

- Seven user selectable periodic interrupts
- User selectable clock source:
  - 32kHz (32KXCLK) clock from crystal oscillator
  - External clock from PPIECK pin

7.3 Functional Description

The PPI module generates periodic interrupt requests to the CPU. When PPI counter reaches the defined count, it generates an interrupt request. The latched status of interrupt generation of the PPI can be read directly from the PPI1L bit.

The PPI counter can count and generate interrupts even when the MCU is in stop mode if the corresponding clock source is enabled.

Figure 7-1 is a block diagram of the PPI.

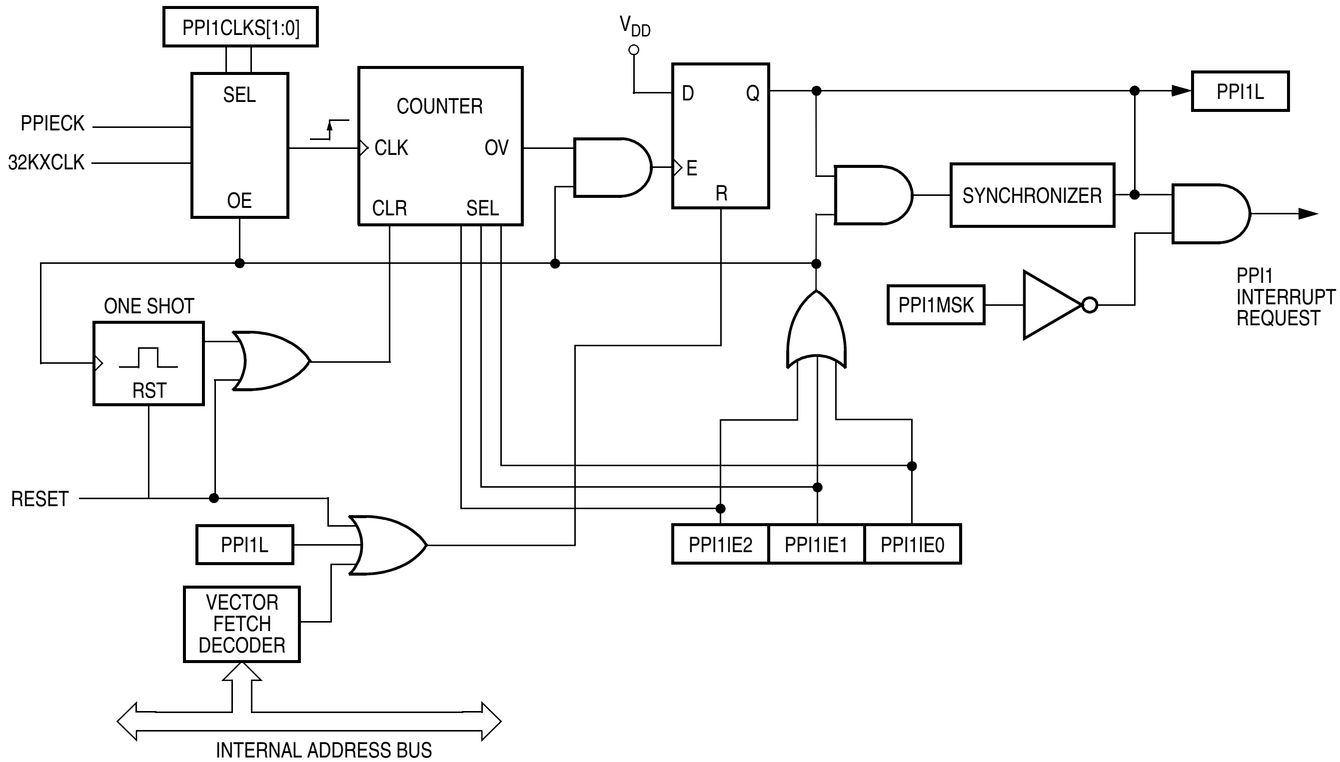


Figure 7-1. Programmable Periodic Interrupt Block Diagram

7.4 I/O Pins

The external clock input option of the PPI is from the PPIECK pin and is selected by the clock select bits, PPI1CLKS[1:0]. The maximum PPIECK frequency is four times the bus frequency.

7.5 Low-Power Modes

The PPI module remains active (crystal clock source is not affected if crystal clock is enabled in stop mode; counter can count and can generate interrupts) in wait and stop mode if proper clocking source is supplied.

7.6 PPI1 Status and Control Register (PPI1SCR)

The PPI1 status and control register (PPI1SCR) controls and monitors the operation of the PPI module.

Address: $0019

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PPI1L	PPI1MSK	PPICLKS1	PPI1CLKS0	0	PPI1IE2	PPI1IE1	PPI1IE0
Write:								
Reset:	0	0	1	0	0	0	0	0

Figure 7-2. PPI1 Status and Control Register (PPI1SCR)

PPI1L — PPI1 Interrupt Flag

This read/write bit indicates a interrupt request is generated by PPI1 and is pending for acknowledgement. This bit generates an interrupt to the CPU if PPI1MSK=0.
The PPI1L bit is cleared by writing logic 1 to it.
1 = Read: PPI1 interrupt request is pending / Write: PPI1 interrupt acknowledge
0 = No PPI1 interrupt request is pending

PPI1MSK — PPI1 Interrupt Mask

Writing a logic one to this read/write bit disables PPI1 interrupt requests. Reset clears PPI1MSK.
1 = PPI1 interrupt requests disabled
0 = PPI1 interrupt requests enabled

PPI1CLKS[1:0] — PPI1 Clock Source Select Bits

These two bits select the clock source for the PPI.

Table 7-1. PPI1 Clock Source Selection

PPI1CLKS[1:0]	Clock Source for PPI1
00	Reserved
01	External clock from PPIECK pin
10	32KXCLK from OSC module
11	Reserved

PPI1IE[2:0] —PPI1 Interrupt Period Select Bits

These three bits select the PPI interrupt period. The PPI is disabled when PPI1IE[2:0] are zero and no interrupts are generated.

Table 7-2. PPI1 Interrupt Period Selection

PPI1IE[2:0]	Interrupt Period
000	PPI and its associated interrupts are disabled
001	512 PPI counts
010	1,024 PPI counts
011	2,048 PPI counts
100	4,096 PPI counts
101	8,192 PPI counts
110	16,384 PPI counts
111	32,768 PPI counts

Chapter 8
Liquid Crystal Display (LCD) Driver

8.1 Introduction

This section describes the liquid crystal display (LCD) driver module. The LCD driver module can drive a maximum of 25 frontplanes and 4 backplanes, depending on the LCD duty selected.

8.2 Features

Features of the LCD driver module include the following:

- Software programmable driver segment configurations:
  - 24 frontplanes × 4 backplanes (96 segments)
  - 25 frontplanes × 3 backplanes (75 segments)
  - 25 frontplanes × 1 backplane (25 segments)
- LCD bias voltages generated by internal resistor ladder
- Software programmable contrast control

8.3 Pin Name Conventions and I/O Register Addresses

Three dedicated I/O pins are for the backplanes, BP0–BP2; twenty four frontplanes, FP1–FP24, are shared with port B, C, D, and E pins. FP0 and BP3 shares the same pin and configured by the DUTY[1:0] bits in the LCD clock register.

The full names of the LCD output pins are shown in Table 8-1. The generic pin names appear in the text that follows.

Table 8-1. Pin Name Conventions

LCD Generic Pin Name	Full MCU Pin Name	Pin Selected for LCD Function by:
FP0/BP3	FP0/BP3	—
BP0–BP2	BP0–BP2	—
FP1–FP2	PTB6/FP1–PTB7/FP2	LCDE in LCDCR
FP3–FP10	PTE0/FP3–PTE7/FP10	PEE in CONFIG2 LCDE in LCDCR
FP11–FP18	PTD0/FP11–PTD7/FP18	PDE in CONFIG2 LCDE in LCDCR
FP19–FP24	PTC0/FP19–PTC5/FP24	PCEL:PCEH in CONFIG2 LCDE in LCDCR

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$004F	LCD Clock Register (LCDCLK)	Read:	0	FCCTL1	FCCTL0	DUTY1	DUTY0	LCLK2	LCLK1	LCLK0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0051	LCD Control Register (LCDCR)	Read:	LCDE	0	FC	LC	LCCON3	LCCON2	LCCON1	LCCON0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0052	LCD Data Register 1 (LDAT1)	Read: Write:	F1B3	F1B2	F1B1	F1B0	F0B3	F0B2	F0B1	F0B0
		Reset:	U	U	U	U	U	U	U	U
$0053	LCD Data Register 2 (LDAT2)	Read: Write:	F3B3	F3B2	F3B1	F3B0	F2B3	F2B2	F2B1	F2B0
		Reset:	U	U	U	U	U	U	U	U
$0054	LCD Data Register 3 (LDAT3)	Read: Write:	F5B3	F5B2	F5B1	F5B0	F4B3	F4B2	F4B1	F4B0
		Reset:	U	U	U	U	U	U	U	U
$0055	LCD Data Register 4 (LDAT4)	Read: Write:	F7B3	F7B2	F7B1	F7B0	F6B3	F6B2	F6B1	F6B0
		Reset:	U	U	U	U	U	U	U	U
$0056	LCD Data Register 5 (LDAT5)	Read: Write:	F9B3	F9B2	F9B1	F9B0	F8B3	F8B2	F8B1	F8B0
		Reset:	U	U	U	U	U	U	U	U
$0057	LCD Data Register 6 (LDAT6)	Read: Write:	F11B3	F11B2	F11B1	F11B0	F10B3	F10B2	F10B1	F10B0
		Reset:	U	U	U	U	U	U	U	U
$0058	LCD Data Register 7 (LDAT7)	Read: Write:	F13B3	F13B2	F13B1	F13B0	F12B3	F12B2	F12B1	F12B0
		Reset:	U	U	U	U	U	U	U	U
$0059	LCD Data Register 8 (LDAT8)	Read: Write:	F15B3	F15B2	F15B1	F15B0	F14B3	F14B2	F14B1	F14B0
		Reset:	U	U	U	U	U	U	U	U
$005A	LCD Data Register 9 (LDAT9)	Read: Write:	F17B3	F17B2	F17B1	F17B0	F16B3	F16B2	F16B1	F16B0
		Reset:	U	U	U	U	U	U	U	U
$005B	LCD Data Register 10 (LDAT10)	Read: Write:	F19B3	F19B2	F19B1	F19B0	F18B3	F18B2	F18B1	F18B0
		Reset:	U	U	U	U	U	U	U	U
$005C	LCD Data Register 11 (LDAT11)	Read: Write:	F21B3	F21B2	F21B1	F21B0	F20B3	F20B2	F20B1	F20B0
		Reset:	U	U	U	U	U	U	U	U
$005D	LCD Data Register 12 (LDAT12)	Read: Write:	F23B3	F23B2	F23B1	F23B0	F22B3	F22B2	F22B1	F22B0
		Reset:	U	U	U	U	U	U	U	U
$005E	LCD Data Register 13 (LDAT13)	Read:	0	0	0	0	F24B3	F24B2	F24B1	F24B0
		Write:								
		Reset:	0	0	0	0	U	U	U	U

U = Unaffected   [shaded] = Unimplemented

Figure 8-1. LCD I/O Register Summary

8.4 Functional Description

Figure 8-2 shows a block diagram of the LCD driver module, and Figure 8-3 shows a simplified schematic of the LCD system.

The LCD driver module uses a 1/3 biasing method. The LCD power is supplied by the V_{LCD} pin. Voltages V_{LCD1}, V_{LCD2}, and V_{LCD3} are generated by an internal resistor ladder.

The LCD data registers, LDAT1–LDAT13, control the LCD segments' ON/OFF, with each data register controlling two frontplanes. When a logic 1 is written to a FxBx bit in the data register, the corresponding frontplane-backplane segment will turn ON. When a logic 0 is written, the segment will turn OFF.

When the LCD driver module is disabled (LCDE = 0), the LCD display will be OFF, all backplane and frontplane drivers have the same potential as V_{DD}. The resistor ladder is disconnected from V_{DD} to reduce power consumption.

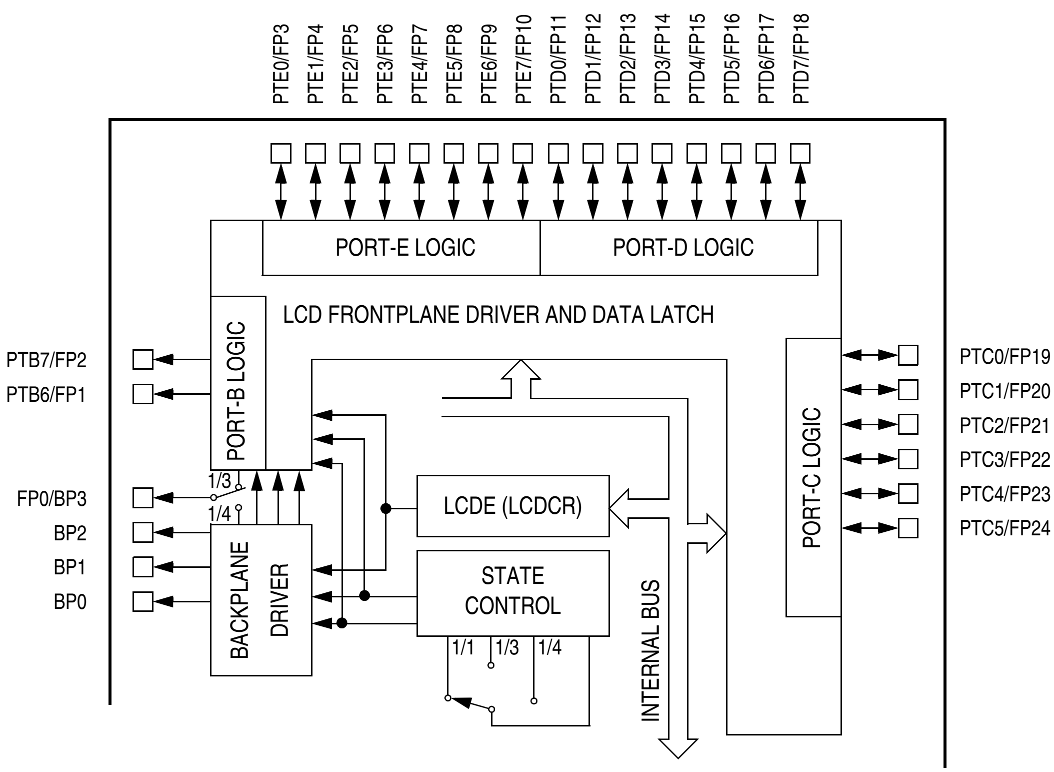


Figure 8-2. LCD Block Diagram

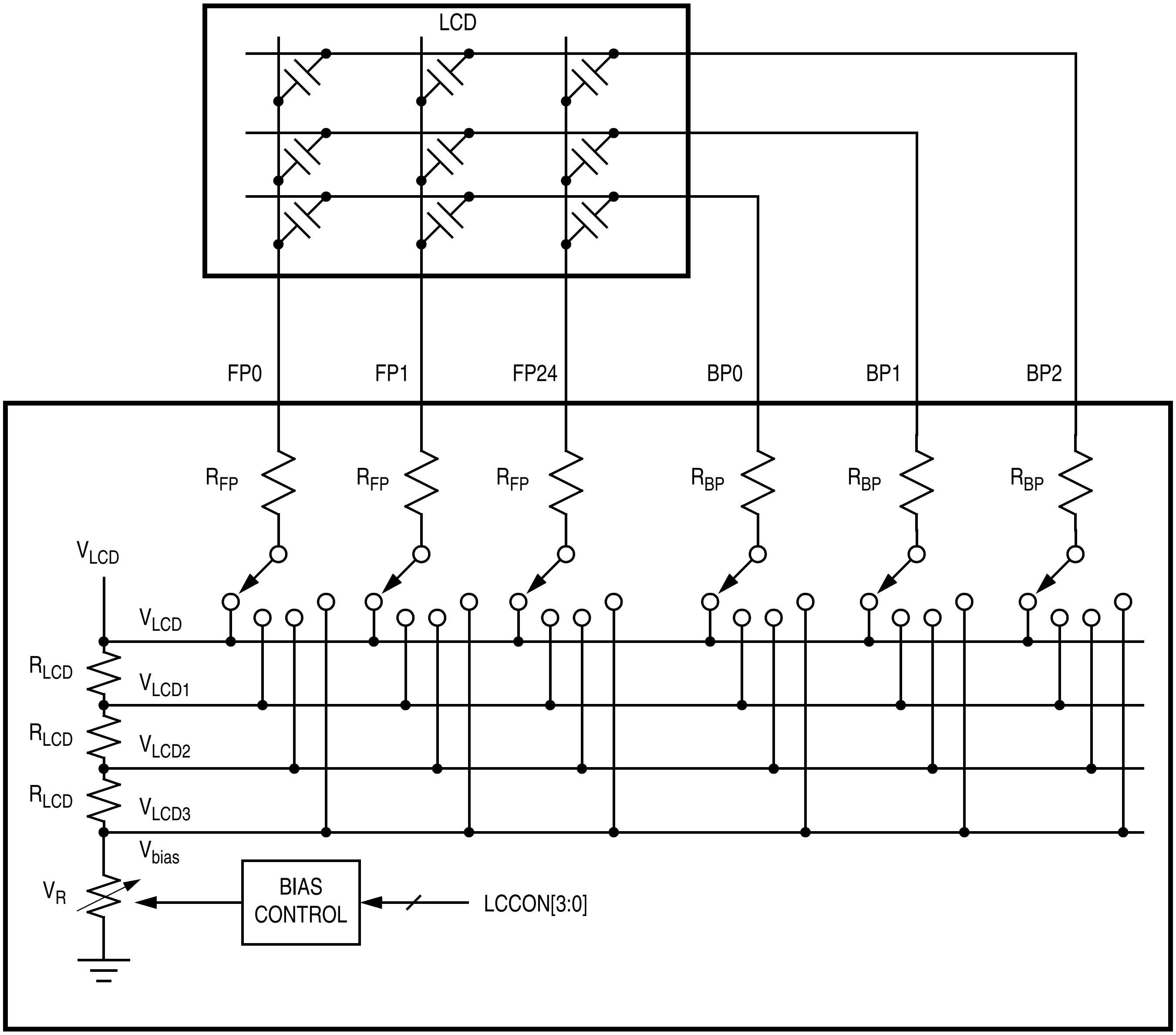


Figure 8-3. Simplified LCD Schematic (1/3 Duty, 1/3 Bias)

8.4.1 LCD Duty

The setting of the LCD output waveform duty is dependent on the number of backplane drivers required. Three LCD duties are available:

- Static duty — BP0 is used only
- 1/3 duty — BP0, BP1, and BP3 are used
- 1/4 duty — BP0, BP1, BP2, and BP3 are used

When the LCD driver module is enabled the backplane waveforms for the selected duty are driven out of the backplane pins. The backplane waveforms are periodic and are shown are shown in Figure 8-5, Figure 8-6, and Figure 8-7.

8.4.2 LCD Voltages (V_{LCD}, V_{LCD1}, V_{LCD2}, V_{LCD3})

The voltage V_{LCD} is from the V_{LCD} pin and must not exceed V_{DD}. V_{LCD1}, V_{LCD2}, and V_{LCD3} are internal bias voltages for the LCD driver waveforms. They are derived from V_{LCD} using a resistor ladder (see Figure 8-3).

The relative potential of the LCD voltages are:

- $V_{LCD} = V_{DD}$
- $V_{LCD1} = 2/3 \times (V_{LCD} - V_{bias})$
- $V_{LCD2} = 1/3 \times (V_{LCD} - V_{bias})$
- $V_{LCD3} = V_{bias}$

The V_{LCD3} bias voltage, V_{bias}, is controlled by the LCD contrast control bits, LCCON[2:0].

8.4.3 LCD Cycle Frame

The LCD driver module uses the 32KXCLK (see Chapter 5 Oscillator (OSC)) as the input reference clock. This clock is divided to produce the LCD waveform base clock, LCDCLK, by configuring the LCLK[2:0] bits in the LCD clock register. The LCDCLK clocks the backplane and the frontplane output waveforms.

The LCD cycle frame is determined by the equation:

$$\text{LCD CYCLE FRAME} = \frac{1}{\text{LCD WAVEFORM BASE CLOCK} \times \text{DUTY}}$$

For example, for 1/3 duty and 256 Hz waveform base clock:

$$\text{LCD CYCLE FRAME} = \frac{1}{256 \times (1/3)}$$

$$= 11.72\ \text{ms}$$

8.4.4 Fast Charge and Low Current

The default value for each of the bias resistors (see Figure 8-3), R_{LCD}, in the resistor ladder is approximately 37 kΩ at V_{LCD} = 3 V. The relatively high current drain through the 37 kΩ resistor ladder may not be suitable for some LCD panel connections. Lowering this current is possible by setting the LC bit in the LCD control register, switching the R_{LCD} value to 146 kΩ.

Although the lower current drain is desirable, but in some LCD panel connections, the higher current is required to drive the capacitive load of the LCD panel. In most cases, the higher current is only required when the LCD waveforms change state (the rising and falling edges in the LCD output waveforms). The fast charge option is designed to have the high current for the switching and the low current for the steady state. Setting the FC bit in the LCD control register selects the fast charge option. The R_{LCD} value is set to 37 kΩ (for high current) for a fraction of time for each LCD waveform switching edge, and then back to 146 kΩ for the steady state period. The duration of the fast charge time is set by configuring the FCCTL[1:0] bits in the LCD clock register, and can be LCDCLK/32, LCDCLK/64, or LCDCLK/128. Figure 8-4 shows the fast charge clock relative to the BP0 waveform.

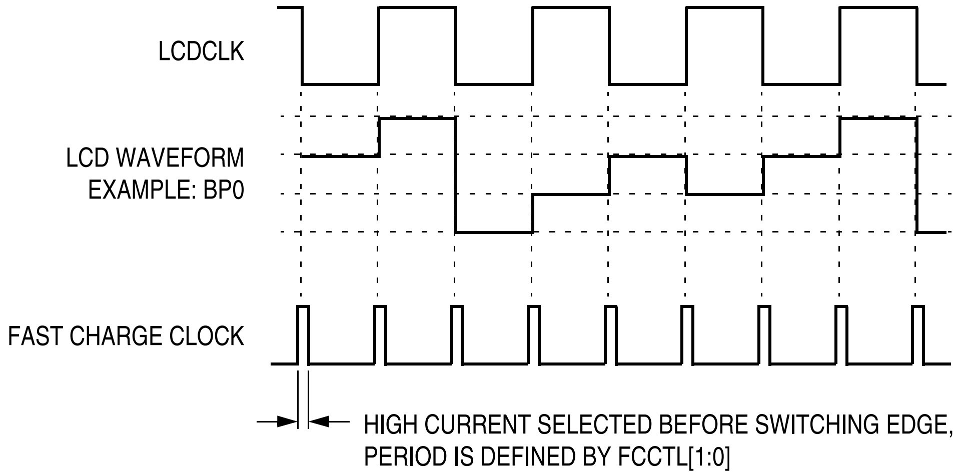


Figure 8-4. Fast Charge Timing

8.4.5 Contrast Control

The contrast of the connected LCD panel can be adjusted by configuring the LCCON[3:0] bits in the LCD control register. The LCCON[3:0] bits provide a 16-step contrast control, which adjusts the bias voltage in the resistor ladder for LCD voltage, V_{LCD3}. The relative voltages, V_{LCD1} and V_{LCD2}, are altered accordingly. For example, setting LCCON[3:0] = $F, the relative panel potential voltage (V_{LCD} – V_{LCD3}) is reduced from maximum 3.3V to approximate 2.45V.

8.5 Low-Power Modes

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

8.5.1 Wait Mode

The LCD driver module continues normal operation in wait mode. If the LCD is not required in wait mode, power down the LCD module by clearing the LCDE bit before executing the WAIT instruction.

8.5.2 Stop Mode

For continuous LCD module operation in stop mode, the oscillator stop mode enable bit (STOP_XTALEN in CONFIG2 register) must be set before executing the STOP instruction. When STOP_XTALEN is set, 32KXCLK continues to drive the LCD module.

If STOP_XTALEN bit is cleared, the LCD module is inactive after the execution of a STOP instruction. The STOP instruction does not affect LCD register states. LCD module operation resumes after an external interrupt. To further reduce power consumption, the LCD module should be powered-down by clearing the LCDE bit before executing the STOP instruction.

8.6 I/O Signals

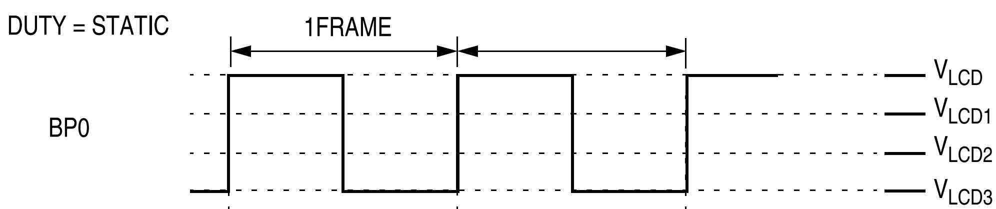
The LCD driver module has twenty-eight (28) output pins.

- FP0/BP3 (multiplexed; selected as FP0 or BP3 by DUTY[1:0])
- BP0–BP2
- FP1–FP2 (shared with port B)
- FP3–FP10 (shared with port E)
- FP11–FP18 (shared with port D)
- FP19–FP24 (shared with port C)

8.6.1 BP0–BP3 (Backplane Drivers)

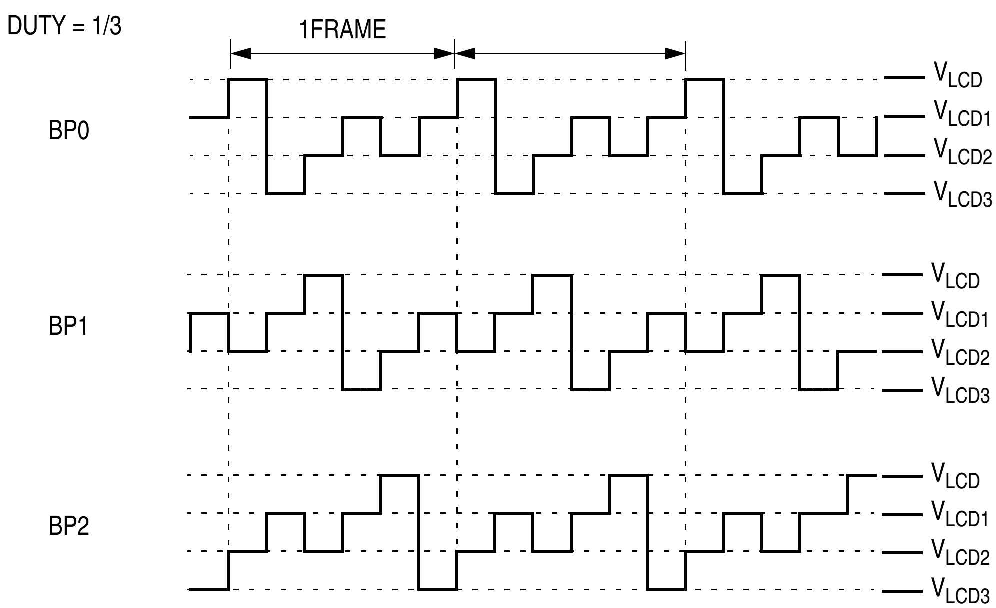
BP0–BP3 are the backplane driver output pins. These are connected to the backplane of the LCD panel. Depending on the LCD duty selected, the voltage waveforms in Figure 8-5, Figure 8-6, and Figure 8-7 appear on the backplane pins.

BP3 pin is only used when 1/4 duty is selected. The pin becomes FP0 for static and 1/3 duty operations.



NOTES:
1. BP1, BP2, and BP3 are not used.
2. At static duty, 1FRAME is equal to the cycle of LCD waveform base clock.

Figure 8-5. Static LCD Backplane Driver Waveform



NOTES:
1. BP3 is not used.
2. At 1/3 duty, 1FRAME has three times the cycle of LCD waveform base clock.

Figure 8-6. 1/3 Duty LCD Backplane Driver Waveforms

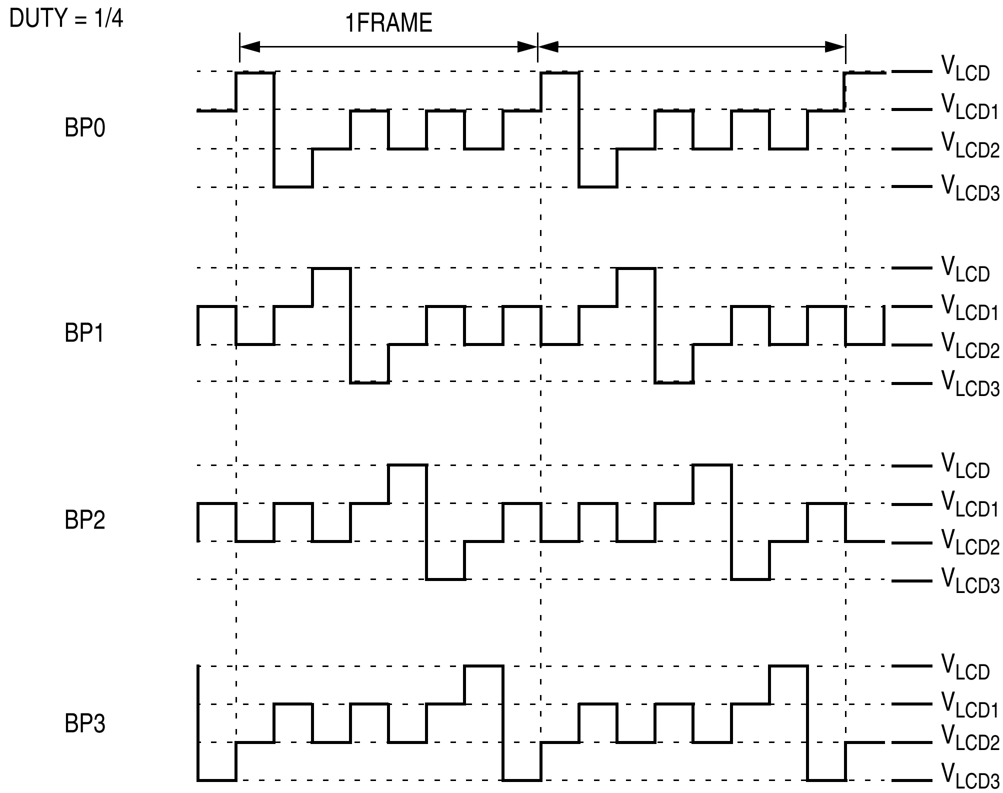


Figure 8-7. 1/4 Duty LCD Backplane Driver Waveforms

8.6.2 FP0–FP24 (Frontplane Drivers)

FP0–FP24 are the frontplane driver output pins. These are connected to the frontplane of the LCD panel. Depending on LCD duty selected and the contents in the LCD data registers, the voltage waveforms in Figure 8-8, Figure 8-9, Figure 8-10 and Figure 8-11 appear on the frontplane pins.

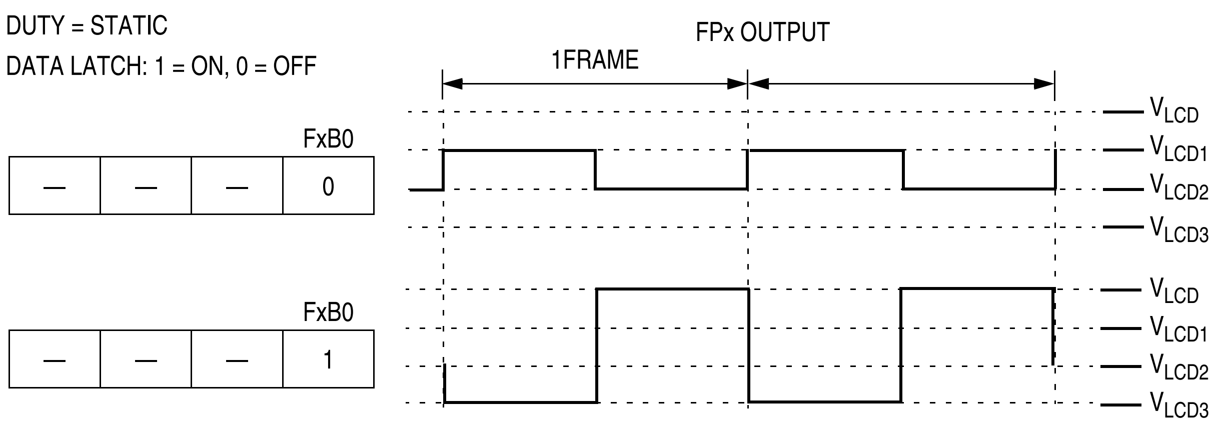


Figure 8-8. Static LCD Frontplane Driver Waveforms

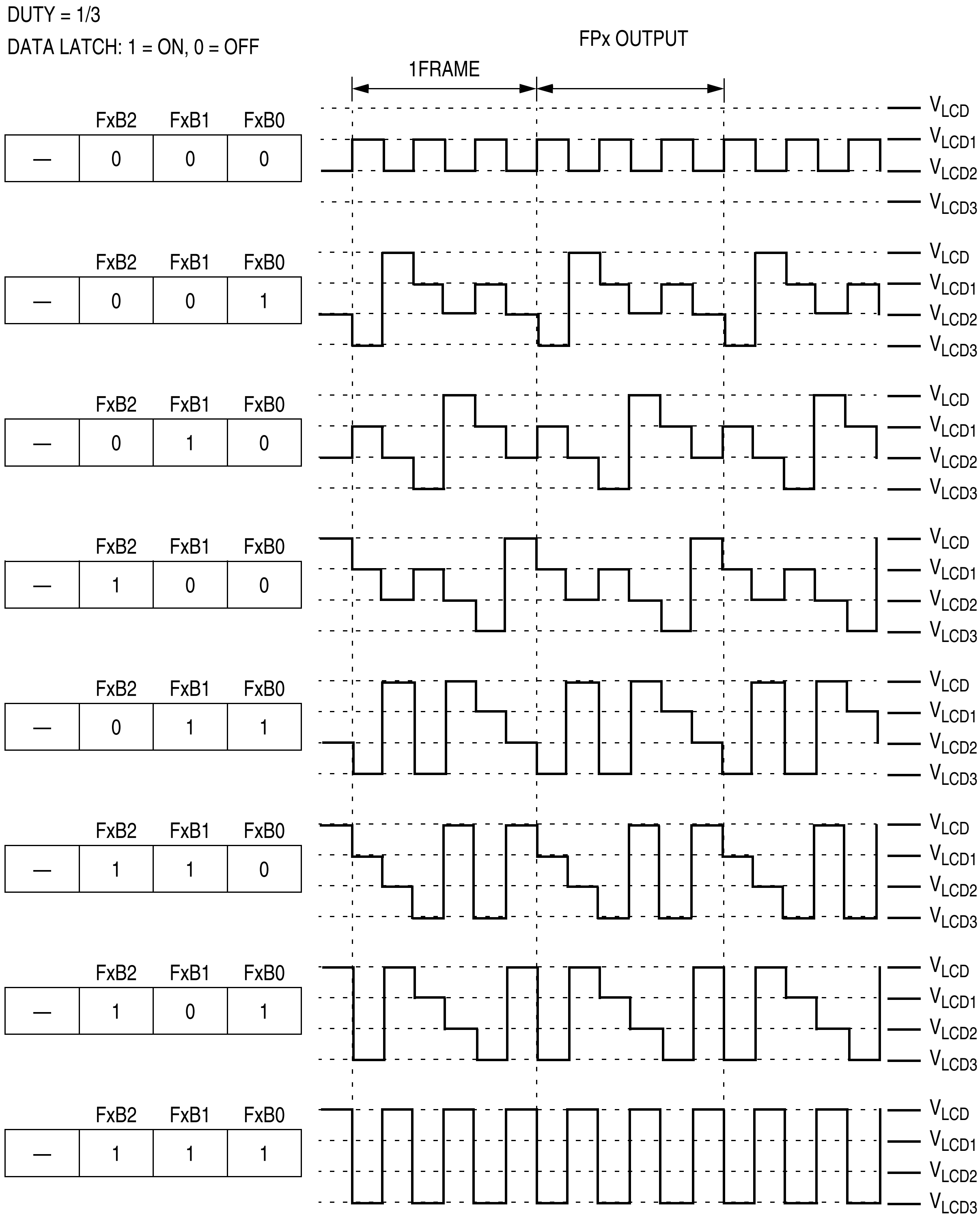


Figure 8-9. 1/3 Duty LCD Frontplane Driver Waveforms

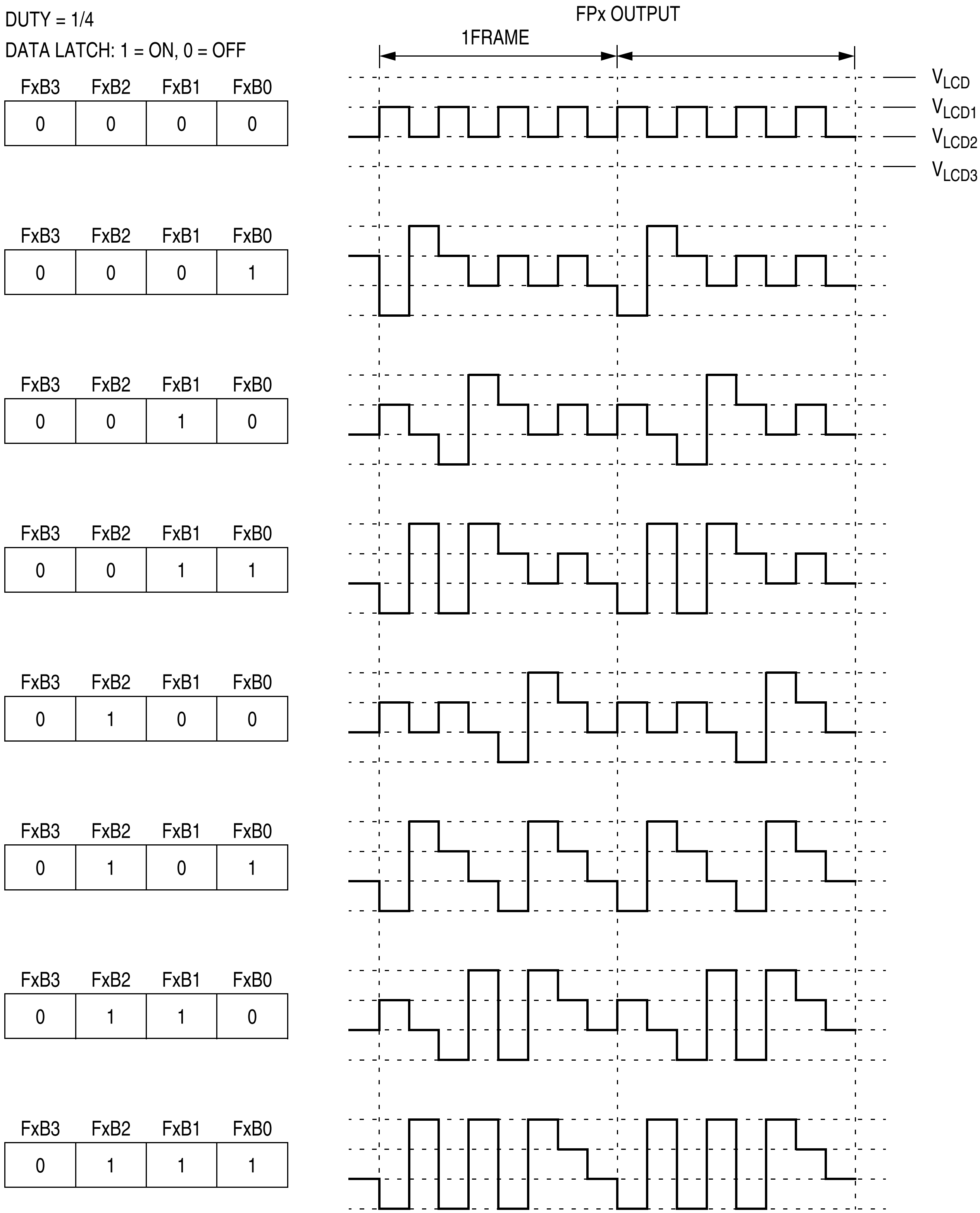


Figure 8-10. 1/4 Duty LCD Frontplane Driver Waveforms

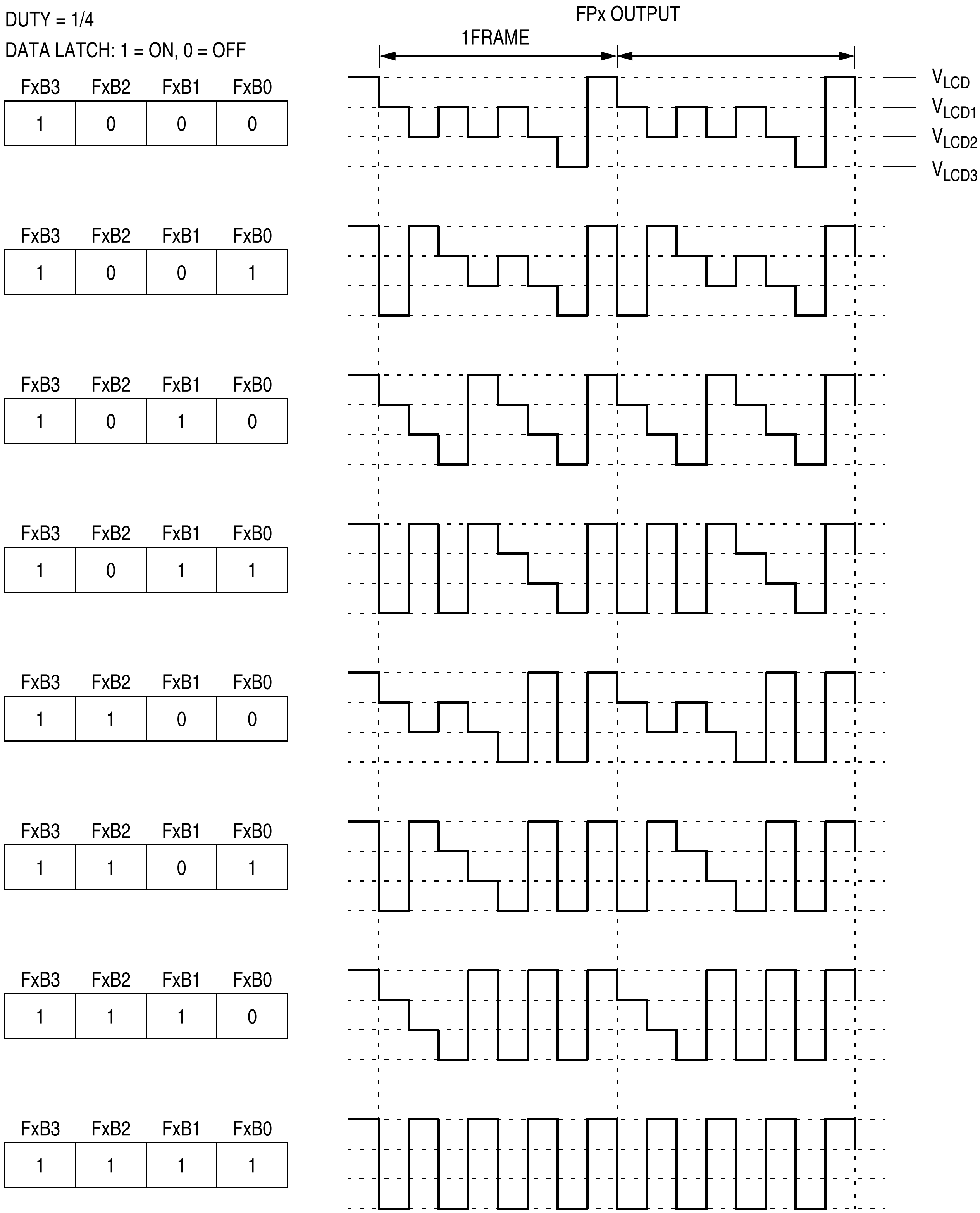
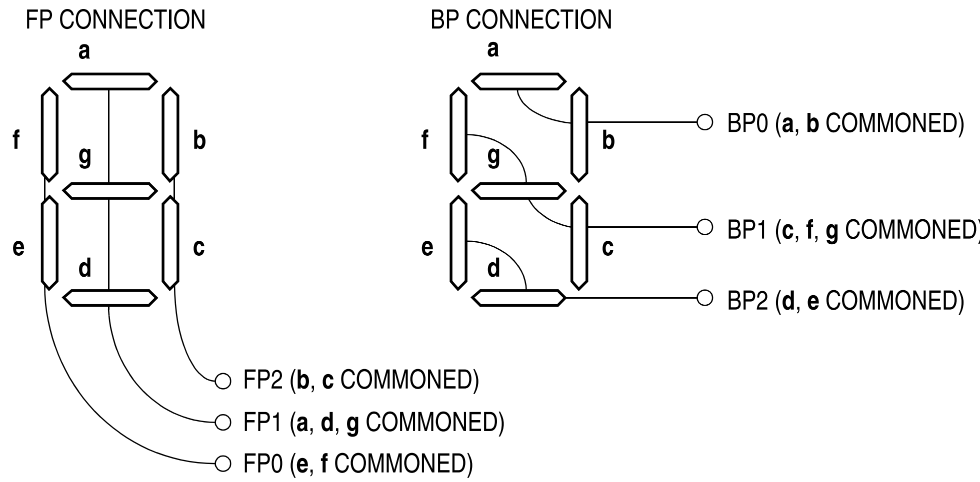


Figure 8-11. 1/4 Duty LCD Frontplane Driver Waveforms (continued)

8.7 Seven Segment Display Connection

The following shows an example for connecting a 7-segment LCD display to the LCD driver.

The example uses 1/3 duty cycle, with pins BP0, BP1, BP2, FP0, FP1, and FP2 connected as shown in Figure 8-12. The output waveforms are shown in Figure 8-13.



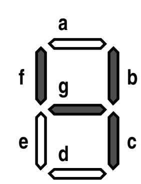
The segment assignments for each bit in the data registers are:

	F1B3	F1B2	F1B1	F1B0	F0B3	F0B2	F0B1	F0B0
LDAT1 $0052	—	d	g	a	—	e	f	—
	←	FP1		→	←	FP0		→

	F3B3	F3B2	F3B1	F3B0	F2B3	F2B2	F2B1	F2B0
LDAT2 $0053	—	—	—	—	—	—	c	b
					←	FP2		→

To display the character "4": LDAT1 = X010X01X, LDAT2 = XXXXXX11

LDAT1 $0052	X	0	1	0	X	0	1	X
LDAT2 $0053	X	X	X	X	X	X	1	1



X = don’t care

Figure 8-12. 7-Segment Display Example

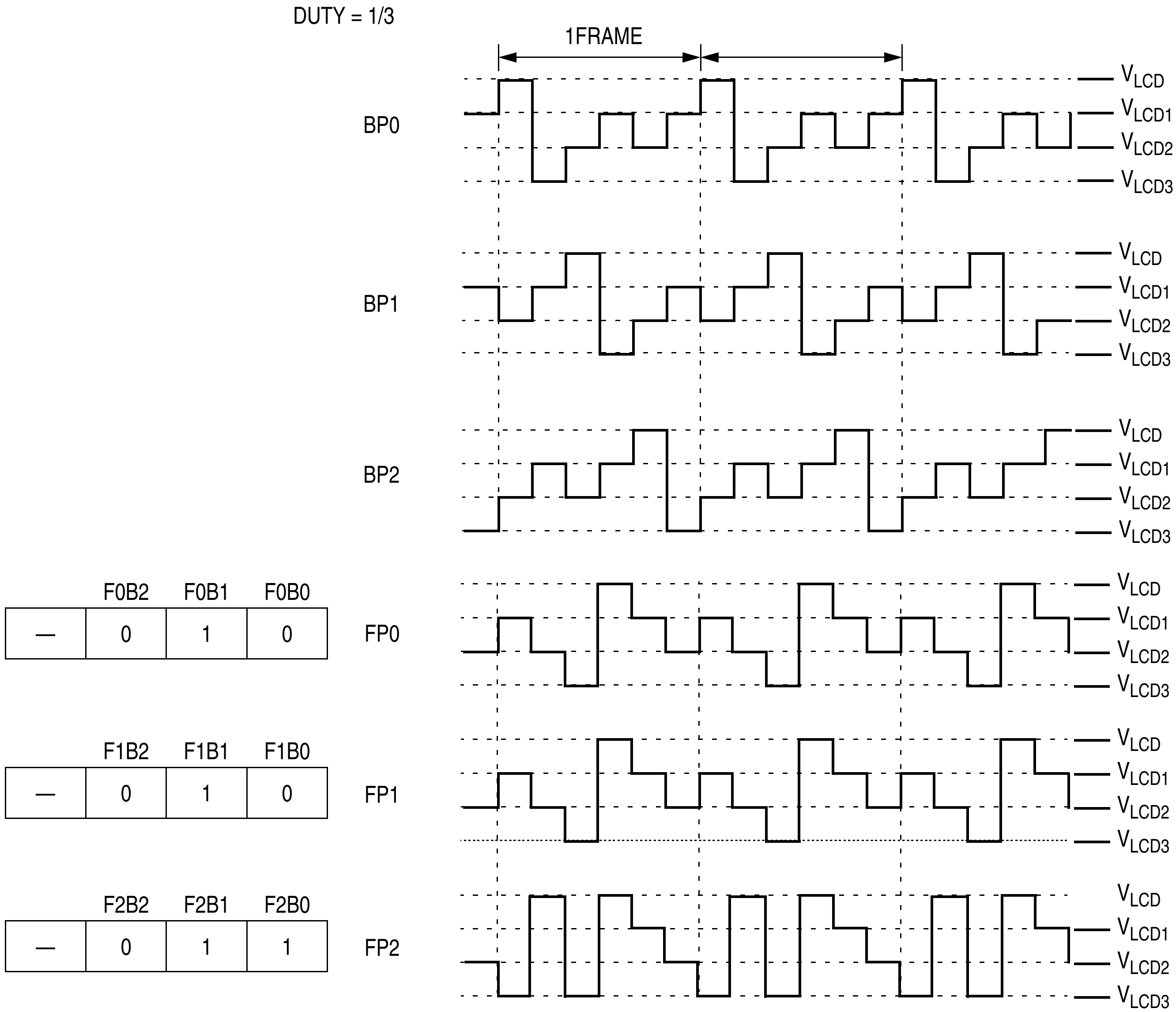


Figure 8-13. BP0–BP2 and FP0–FP2 Output Waveforms for 7-Segment Display Example

The voltage waveform across the "f" segment of the LCD (between BP1 and FP0) is illustrated in Figure 8-14. As shown in the waveform, the voltage peaks reach the LCD-ON voltage, V_{LCD}, therefore, the segment will be ON.

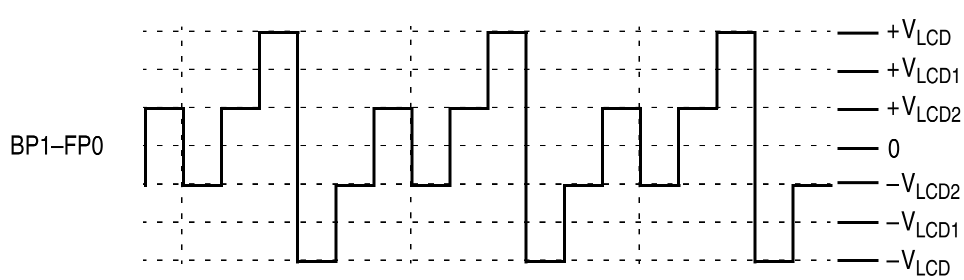


Figure 8-14. "f" Segment Voltage Waveform

The voltage waveform across the "e" segment of the LCD (between BP2 and FP0) is illustrated in Figure 8-15. As shown in the waveform, the voltage peaks do not reach the LCD-ON voltage, V_{LCD}, therefore, the segment will be OFF.

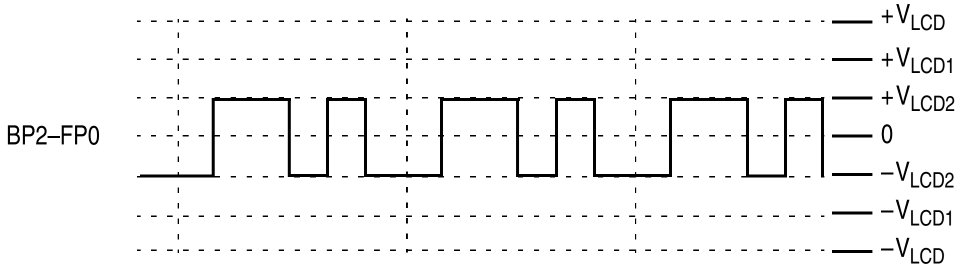


Figure 8-15. "e" Segment Voltage Waveform

8.8 I/O Registers

Fifteen (15) registers control LCD driver module operation:

- LCD control register (LCDCR)
- LCD clock register (LCDCLK)
- LCD data registers (LDAT1–LDAT13)

8.8.1 LCD Control Register (LCDCR)

The LCD control register (LCDCR):

- Enables the LCD driver module
- Selects bias resistor value and fast-charge control
- Selects LCD contrast

Address: $0051

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	LCDE	0	FC	LC	LCCON3	LCCON2	LCCON1	LCCON0
Write:		(unimplemented)						
Reset:	0	0	0	0	0	0	0	0

(shaded) = Unimplemented

Figure 8-16. LCD Control Register (LCDCR)

LCDE — LCD Enable

This read/write bit enables the LCD driver module; the backplane and frontplane drive LCD waveforms out of BPx and FPx pins. Reset clears the LCDE bit.

1 = LCD driver module enabled
0 = LCD driver module disabled

FC — Fast Charge
LC — Low Current

These read/write bits are used to select the value of the resistors in resistor ladder for LCD voltages. Reset clears the FC and LC bits.

Table 8-2. Resistor Ladder Selection

FC	LC	Action
X	0	Each resistor is approximately 37 kΩ (default)
0	1	Each resistor is approximately 146 kΩ
1	1	Fast charge mode

LCCON[3:0] — LCD Contrast Control

These read/write bits select the bias voltage, V_{bias}. This voltage controls the contrast of the LCD. Maximum contrast is set when LCCON[3:0] =%0000;
minimum contrast is set when LCCON[3:0] =%1111.

Table 8-3. LCD Bias Voltage Control

LCCON3	LCCON2	LCCON1	LCCON0	Bias Voltage (approximate % of V_{DD})
0	0	0	0	0.6
0	0	0	1	2.9
0	0	1	0	5.2
0	0	1	1	7.4
0	1	0	0	9.6
0	1	0	1	11.6
0	1	1	0	13.5
0	1	1	1	15.3
1	0	0	0	17.2
1	0	0	1	18.8
1	0	1	0	20.5
1	0	1	1	22.0
1	1	0	0	23.6
1	1	0	1	25.0
1	1	1	0	26.4
1	1	1	1	27.7

8.8.2 LCD Clock Register (LCDCLK)

The LCD clock register (LCDCLK):

- Selects the fast charge duty cycle
- Selects LCD driver duty cycle
- Selects LCD waveform base clock

Address: $004F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	FCCTL1	FCCTL0	DUTY1	DUTY0	LCLK2	LCLK1	LCLK0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 8-17. LCD Clock Register (LCDCLK)

FCCTL[1:0] — Fast Charge Duty Cycle Select

These read/write bits select the duty cycle of the fast charge duration. Reset clears these bits. (See 8.4.4 Fast Charge and Low Current)

Table 8-4. Fast Charge Duty Cycle Selection

FCCTL1:FCCTL0	Fast Charge Duty Cycle
00	In each LCDCLK/2 period, each bias resistor is reduced to 37 kΩ for a duration of LCDCLK/32.
01	In each LCDCLK/2 period, each bias resistor is reduced to 37 kΩ for a duration of LCDCLK/64.
10	In each LCDCLK/2 period, each bias resistor is reduced to 37 kΩ for a duration of LCDCLK/128.
11	Not used

DUTY[1:0] — Duty Cycle Select

These read/write bits select the duty cycle of the LCD driver output waveforms. The multiplexed FP0/BP3 pin is controlled by the duty cycle selected. Reset clears these bits.

Table 8-5. LCD Duty Cycle Selection

DUTY1:DUTY0	Description
00	Static selected; FP0/BP3 pin function as FP0.
01	1/3 duty cycle selected; FP0/BP3 pin functions as FP0.
10	1/4 duty cycle selected; FP0/BP3 pin functions as BP3.
11	Not used

LCLK[2:0] — LCD Waveform Base Clock Select

These read/write bits selects the LCD waveform base clock. Reset clears these bits.

Table 8-6. LCD Waveform Base Clock Selection

LCLK2	LCLK1	LCLK0	Divide Ratio	LCD Waveform Base Clock Frequency LCDCLK (Hz)	LCD Frame Rate f_{XTAL} = 32.768 kHz	
				f_{XTAL} = 32.768 kHz	1/3 duty	1/4 duty
0	0	0	128	256	85.3	64
0	0	1	256	128	42.7	32
0	1	0	512	64	21.3	16
0	1	1	1024	32	10.7	8
1	0	0	Reserved			
1	0	1	Reserved			
1	1	0	Reserved			
1	1	1	Reserved			

8.8.3 LCD Data Registers (LDAT1–LDAT17)

The thirteen (13) LCD data registers enable and disable the drive to the corresponding LCD segments.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0052	LCD Data Register 1 (LDAT1)	Read: Write:	F1B3	F1B2	F1B1	F1B0	F0B3	F0B2	F0B1	F0B0
		Reset:	U	U	U	U	U	U	U	U
$0053	LCD Data Register 2 (LDAT2)	Read: Write:	F3B3	F3B2	F3B1	F3B0	F2B3	F2B2	F2B1	F2B0
		Reset:	U	U	U	U	U	U	U	U
$0054	LCD Data Register 3 (LDAT3)	Read: Write:	F5B3	F5B2	F5B1	F5B0	F4B3	F4B2	F4B1	F4B0
		Reset:	U	U	U	U	U	U	U	U
$0055	LCD Data Register 4 (LDAT4)	Read: Write:	F7B3	F7B2	F7B1	F7B0	F6B3	F6B2	F6B1	F6B0
		Reset:	U	U	U	U	U	U	U	U
$0056	LCD Data Register 5 (LDAT5)	Read: Write:	F9B3	F9B2	F9B1	F9B0	F8B3	F8B2	F8B1	F8B0
		Reset:	U	U	U	U	U	U	U	U
$0057	LCD Data Register 6 (LDAT6)	Read: Write:	F11B3	F11B2	F11B1	F11B0	F10B3	F10B2	F10B1	F10B0
		Reset:	U	U	U	U	U	U	U	U
$0058	LCD Data Register 7 (LDAT7)	Read: Write:	F13B3	F13B2	F13B1	F13B0	F12B3	F12B2	F12B1	F12B0
		Reset:	U	U	U	U	U	U	U	U

U = Unaffected   [shaded] = Unimplemented

Figure 8-18. LCD Data Registers 1–13 (LDAT1–LDAT13)

Address	Register		Bit 7	6	5	4	3	2	1	Bit 0
$0059	LCD Data Register 8 (LDAT8)	Read: / Write:	F15B3	F15B2	F15B1	F15B0	F14B3	F14B2	F14B1	F14B0
		Reset:	U	U	U	U	U	U	U	U
$005A	LCD Data Register 9 (LDAT9)	Read: / Write:	F17B3	F17B2	F17B1	F17B0	F16B3	F16B2	F16B1	F16B0
		Reset:	U	U	U	U	U	U	U	U
$005B	LCD Data Register 10 (LDAT10)	Read: / Write:	F19B3	F19B2	F19B1	F19B0	F18B3	F18B2	F18B1	F18B0
		Reset:	U	U	U	U	U	U	U	U
$005C	LCD Data Register 11 (LDAT11)	Read: / Write:	F21B3	F21B2	F21B1	F21B0	F20B3	F20B2	F20B1	F20B0
		Reset:	U	U	U	U	U	U	U	U
$005D	LCD Data Register 12 (LDAT12)	Read: / Write:	F23B3	F23B2	F23B1	F23B0	F22B3	F22B2	F22B1	F22B0
		Reset:	U	U	U	U	U	U	U	U
$005E	LCD Data Register 13 (LDAT13)	Read:	0	0	0	0	F24B3	F24B2	F24B1	F24B0
		Write:								
		Reset:	0	0	0	0	U	U	U	U

U = Unaffected ▒ = Unimplemented

Figure 8-18. LCD Data Registers 1–13 (LDAT1–LDAT13)

Chapter 9
Input/Output (I/O) Ports

9.1 Introduction

Thirty-eight (38) bidirectional input-output (I/O) pins form six 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:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PTB)	Read:	PTB7	PTB6	0	0	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PTC)	Read:	PTC7	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
		Write:								
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PTD)	Read:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Write:								
		Reset:	Unaffected by reset							
$0004	Data Direction Register A (DDRA)	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB)	Read:	DDRB7	DDRB6	0	0	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0006	Data Direction Register C (DDRC)	Read:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007	Data Direction Register D (DDRD)	Read:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0008	Data Direction Register E (DDRE)	Read:	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0009	Port E Data Register (PTE)	Read:	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0
		Write:								
		Reset:	Unaffected by reset							
$000C	Port-B High Current Drive Control Register (HDB)	Read:	R	PPI1L	HDB5	HDB4	HDB3	HDB2	PPI1CLKS1	PPI1CLKS0
		Write:								
		Reset:			0	0	0	0	0	0

U = Unaffected X = Indeterminate (shaded) = Unimplemented R = Reserved

Figure 9-1. I/O Port Register Summary

Table 9-1. Port Control Register Bits Summary (Sheet 1 of 2)

Port	Bit	DDR	Module Control			Pin
			Module	Register	Control Bit	
A	0	DDRA0	KBI	KBIER ($001C)	KBIE0	PTA0/KBI0
	1	DDRA1			KBIE1	PTA1/KBI1
	2	DDRA2			KBIE2	PTA2/KBI2
	3	DDRA3			KBIE3	PTA3/KBI3
	4	DDRA4	—	—	—	PTA4
	5	DDRA5				PTA5
	6	DDRA6				PTA6
	7	DDRA7				PTA7
B	0	DDRB0	TIM2	T2SC0 ($0030)	ELS0B:ELS0A	PTB0/T2CH0
	1	DDRB1		T2SC1 ($0033)	ELS1B:ELS1A	PTB1/T2CH1
	2	DDRB2	TIM1	T1SC0 ($0025) HDB ($000C)	ELS0B:ELS0A PPI1CLKS[1:0]	PTB2/T1CH0/PPIECK
	3	DDRB3		T1SC1 ($0028)	ELS1B:ELS1A	PTB3/T1CH1
	6	DDRB6	LCD	LCDCR ($0051)	LCDE	PTB6/FP1(1)
	7	DDRB7				PTB7/FP2(1)
C	0	DDRC0	LCD	CONFIG2 ($001D) LCDCR ($0051)	PCEL LCDE	PTC0/FP19
	1	DDRC1				PTC1/FP20
	2	DDRC2				PTC2/FP21
	3	DDRC3				PTC3/FP22
	4	DDRC4			PCEH LCDE	PTC4/FP23(1)
	5	DDRC5				PTC5/FP24(1)
	6	DDRC6	—	—	—	PTC6
	7	DDRC7	—	—	—	PTC7
D	0	DDRD0	LCD	CONFIG2 ($001D) LCDCR ($0051)	PDE LCDE	PTD0/FP11
	1	DDRD1				PTD1/FP12
	2	DDRD2				PTD2/FP13
	3	DDRD3				PTD3/FP14
	4	DDRD4				PTD4/FP15
	5	DDRD5				PTD5/FP16
	6	DDRD6				PTD6/FP17
	7	DDRD7				PTD7/FP18

Table 9-1. Port Control Register Bits Summary (Sheet 2 of 2)

Port	Bit	DDR	Module Control			Pin
			Module	Register	Control Bit	
E	0	DDRE0	LCD	CONFIG2 ($001D) LCDCR ($0051)	PEE LCDE	PTE0/FP3
	1	DDRE1				PTE1/FP4
	2	DDRE2				PTE2/FP5
	3	DDRE3				PTE3/FP6
	4	DDRE4				PTE4/FP7
	5	DDRE5				PTE5/FP8
	6	DDRE6				PTE6/FP9
	7	DDRE7				PTE7/FP10

1. Pins not available on 44-pin package.

9.2 Port A

Port A is an 8-bit special function port that shares four of its port pins with the keyboard interrupt module (KBI).

9.2.1 Port A Data Register (PTA)

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:					KBI3	KBI2	KBI1	KBI0

Figure 9-2. 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.

KBI[3:0] — Keyboard Interrupt Channels 3 to 0

KBI[3:0] are pins used for the keyboard interrupt input. The corresponding input, KBI[3:0], can be enabled in the keyboard interrupt enable register, KBIER. Port pins used as KBI input will override any control from the port I/O logic. See Section 20. Keyboard Interrupt Module (KBI).

9.2.2 Data Direction Register A (DDRA)

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

Figure 9-3. 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 9-4 shows the port A I/O logic.

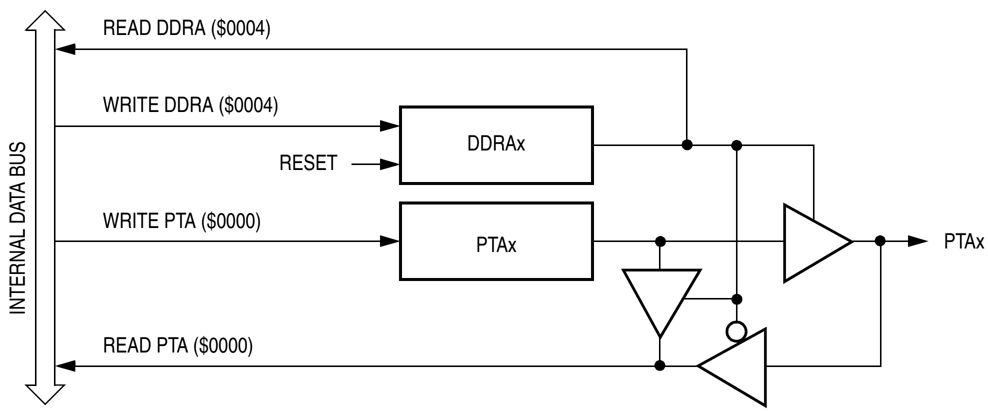


Figure 9-4. Port A I/O Circuit

When DDRAx is a logic 1, reading address $0000 reads the PTAx data latch. When DDRAx is a logic 0, reading address $0000 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 9-2 summarizes the operation of the port A pins.

Table 9-2. Port A Pin Functions

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

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

9.3 Port B

Port B is an 6-bit special function port that shares four of its port pins with the two timers (TIM1 and TIM2) and two of its ports pins with the liquid crystal display (LCD) driver module. Port pin, PTB2, is also shared with the external clock input of the programmable periodic interrupt (PPI) module.

NOTE
PTB6–PTB7 pins are not available on 44-pin package.

9.3.1 Port B Data Register (PTB)

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

Address: $0001

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTB7	PTB6	0	0	PTB3	PTB2	PTB1	PTB0
Write:								
Reset:	Unaffected by reset							
Alternative Functions:	FP2	FP1			T1CH1	T1CH0	T2CH1	T2CH0
						PPIECK		
Additional Functions:					High current sink			

Figure 9-5. Port B Data Register (PTB)

PTB[7:6, 3:0] — Port B Data Bits
These read/write bits are software programmable. Data direction of each port B pin is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.

T1CH[1:0] — Timer 1 Channel I/O Bits
The T1CH1 and T1CH0 pins are the TIM1 input capture/output compare pins. The edge/level select bits, ELSxB:ELSxA, determine whether the PTB2/T1CH0 and PTB3/T1CH1 pins are timer channel I/O pins or general-purpose I/O pins. See Chapter 6 Timer Interface Module (TIM).

T2CH[1:0] — Timer 2 Channel I/O Bits
The T2CH1 and T2CH0 pins are the TIM2 input capture/output compare pins. The edge/level select bits, ELSxB:ELSxA, determine whether the PTB0/T2CH0 and PTB1/T2CH1 pins are timer channel I/O pins or general-purpose I/O pins. See Chapter 6 Timer Interface Module (TIM).

PPIECK — External Clock Source Input for PPI
The PPIECK pin is the external clock input to the PPI module. It is selected by setting the bits PPI1CLKS[1:0] = 01 in the port B high current drive control register. See 7.6 PPI1 Status and Control Register (PPI1SCR).

FP[2:1] — LCD Driver Frontplanes 2–1
FP[2:1] are pins used for the frontplane output of the LCD driver module. The enable bit, LCDE, in the LCDCR register determine whether the PTB7/FP2–PTB6/FP1 pins are LCD frontplane driver pins or general-purpose I/O pins. See Chapter 8 Liquid Crystal Display (LCD) Driver.

9.3.2 Data Direction Register B (DDRB)

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

NOTE

For devices packaged in a 44-pin package, PTB6–PTB7 are not connected. DDRB6:7 should be set to a 1 to configure PTB6–PTB7 as outputs.

Address: $0005

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	DDRB7	DDRB6	0	0	DDRB3	DDRB2	DDRB1	DDRB0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 9-6. Data Direction Register B (DDRB)

DDRB[7:6, 3:0] — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB[7:6, 3:0], configuring all port B pins as inputs.

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

NOTE

Avoid glitches on port B pins by writing to the port B data register before changing data direction register B bits from 0 to 1. Figure 9-7 shows the port B I/O logic.

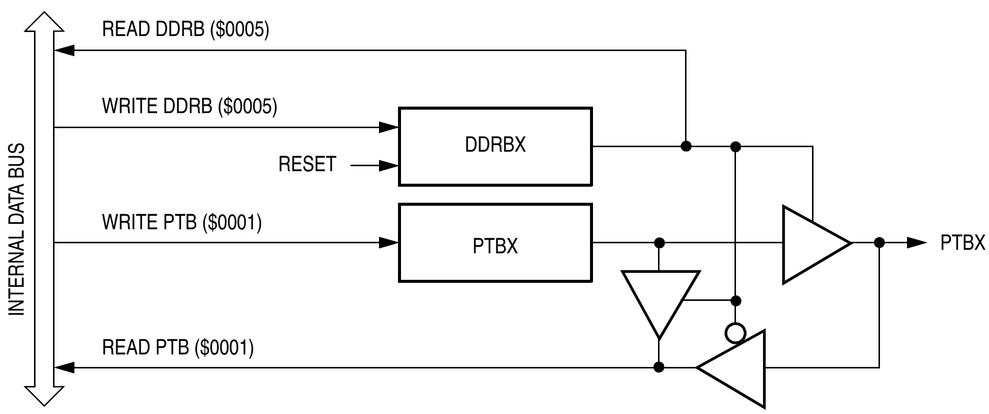


Figure 9-7. Port B I/O Circuit

When DDRBx is a logic 1, reading address $0001 reads the PTBx data latch. When DDRBx is a logic 0, reading address $0001 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 9-3 summarizes the operation of the port B pins.

Table 9-3. Port B Pin Functions

DDRB Bit	PTB Bit	I/O Pin Mode	Accesses to DDRB	Accesses to PTB	
			Read/Write	Read	Write
0	X(1)	Input, Hi-Z(2)	DDRB[7:6, 3:0]	Pin	PTB[7:6, 3:0](3)
1	X	Output	DDRB[7:6, 3:0]	PTB[7:6, 3:0]	PTB[7:6, 3:0]

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

9.3.3 Port B High Current Drive Control Register (HDB)

The port-B high current drive control register (HDB) controls the high current drive capability on PTB[3:2]. Each bit is individually configurable and requires that the data direction register, DDRB, bit be configured as an output.

Address: $000C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	HDB3	HDB2	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 9-8. Port B High Current Drive Control Register (HDB)

HDB[3:2] — Port B High Current Drive Enable Bits

These read/write bits are software programmable to enable the direct LED drive on an output port pin.

1 = Corresponding port B pin is configured to high current sink direct LED drive.
0 = Corresponding port B pin is configured to standard drive

9.4 Port C

Port C is an 8-bit special function port that shares five of its port pins with the liquid crystal display (LCD) driver module.

NOTE

PTC4–PTC5 pins are not available on 44-pin package.

9.4.1 Port C Data Register (PTC)

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

Address: $0002

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTC7	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
Reset:	Unaffected by reset							
Alternative Function:			FP24	FP23	FP22	FP21	FP20	FP19

Figure 9-9. Port C Data Register (PTC)

PTC[7:0] — 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.

FP[24:19] — LCD Driver Frontplanes 24–19

FP[24:19] are pins used for the frontplane output of the LCD driver module. The enable bits, PCEH and PCEL, in the CONFIG2 register, and LCDE bit in the LCDCR register determine whether the PTC5/FP24–PTC4/FP23 and PTC3/FP22–PTC0/FP19 pins are LCD frontplane driver pins or general-purpose I/O pins. See Chapter 8 Liquid Crystal Display (LCD) Driver.

9.4.2 Data Direction Register C (DDRC)

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.

NOTE

For devices packaged in a 44-pin package, PTC4–PTC5 are not connected. DDRC4:5 should be set to a 1 to configure PTC4–PTC5 as outputs.

Address: $0006

	Bit 7	6	5	4	3	2	1	Bit 0
Read:/Write:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
Reset:	0	0	0	0	0	0	0	0

Figure 9-10. Data Direction Register C (DDRC)

DDRC[7:0] — Data Direction Register C Bits

These read/write bits control port C data direction. Reset clears DDRC[7:0], 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 9-11 shows the port C I/O logic.

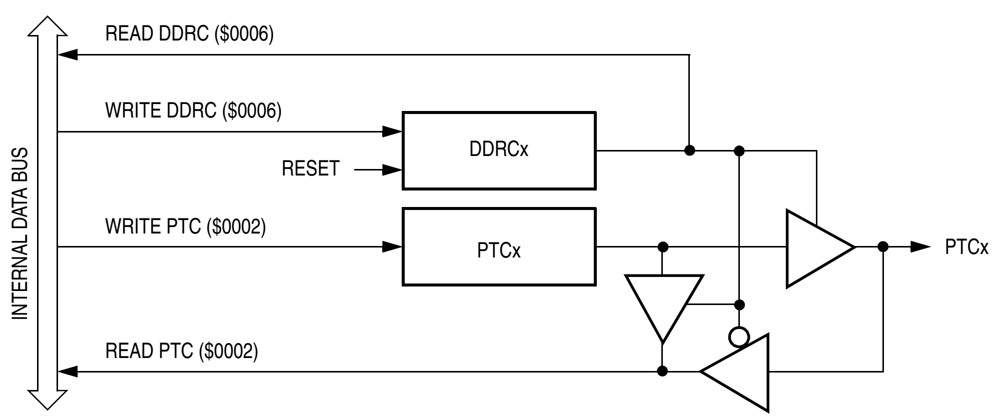


Figure 9-11. Port C I/O Circuit

When DDRCx is a logic 1, reading address $0002 reads the PTCx data latch. When 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 9-4 summarizes the operation of the port C pins.

Table 9-4. 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[7:0]	Pin	PTC[7:0][(3)]
1	X	Output	DDRC[7:0]	PTC[7:0]	PTC[7:0]

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

9.5 Port D

Port D is an 8-bit special function port that shares all of its port pins with the liquid crystal display (LCD) driver module.

9.5.1 Port D Data Register (PTD)

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

Address: $0003

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
Reset:	Unaffected by reset							
Alternative Function:	FP18	FP17	FP16	FP15	FP14	FP13	FP12	FP11

Figure 9-12. Port D Data Register (PTD)

PTD[7:0] — Port D Data Bits

These read/write bits are software programmable. Data direction of each port D pin is under the control of the corresponding bit in data direction register D. Reset has no effect on port D data.

FP[18:11] — LCD Driver Frontplanes 18–11

FP[18:11] are pins used for the frontplane output of the LCD driver module. The enable bit, PDE, in the CONFIG2 register and LCDE bit in the LCDCR register, determines whether the PTD7/FP18–PTD0/FP11 pins are LCD frontplane driver pins or general-purpose I/O pins. See Chapter 8 Liquid Crystal Display (LCD) Driver.

9.5.2 Data Direction Register D (DDRD)

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: Write:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Reset:	0	0	0	0	0	0	0	0

Figure 9-13. Data Direction Register D (DDRD)

DDRD[7:0] — Data Direction Register D Bits

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

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

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. Figure 9-14 shows the port D I/O logic.

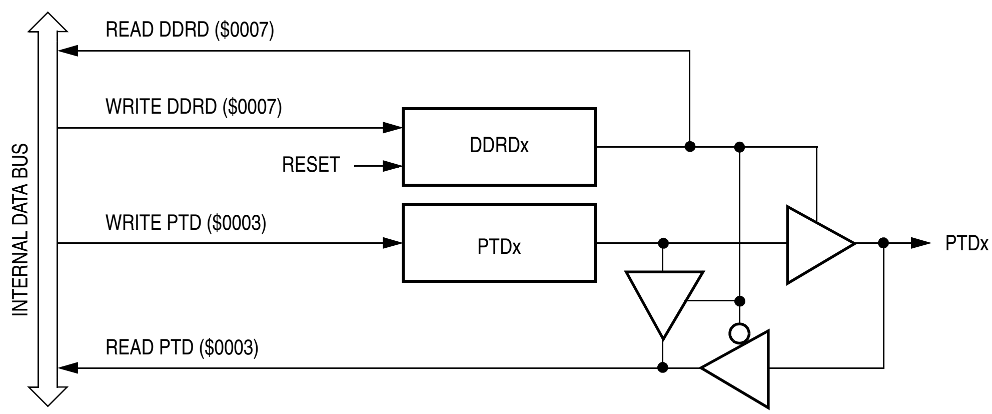


Figure 9-14. Port D I/O Circuit

When DDRDx is a logic 1, reading address $0003 reads the PTDx data latch. When 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 9-5 summarizes the operation of the port D pins.

Table 9-5. Port D Pin Functions

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

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

9.6 Port E

Port E is an 8-bit special function port that shares all of its port pins with the liquid crystal display (LCD) driver module.

9.6.1 Port E Data Register (PTE)

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

Address: $0009

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0
Reset:	Unaffected by reset							
Alternative Function:	FP10	FP9	FP8	FP7	FP6	FP5	FP4	FP3

Figure 9-15. Port E Data Register (PTE)

PTE[7:0] — Port E Data Bits

These read/write bits are software programmable. Data direction of each port E pin is under the control of the corresponding bit in data direction register E. Reset has no effect on port E data.

FP[10:3] — LCD Driver Frontplanes 10–3

FP[10:3] are pins used for the frontplane output of the LCD driver module. The enable bit, PEE, in the CONFIG2 register and LCDE bit in the LCDCR register, determines whether the PTE7/FP10–PTE0/FP3 pins are LCD frontplane driver pins or general-purpose I/O pins. See Chapter 8 Liquid Crystal Display (LCD) Driver.

9.6.2 Data Direction Register E (DDRE)

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: $0008

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
Reset:	0	0	0	0	0	0	0	0

Figure 9-16. Data Direction Register E (DDRE)

DDRE[7:0] — Data Direction Register E Bits

These read/write bits control port E data direction. Reset clears DDRE[7:0], 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 9-14 shows the port E I/O logic.

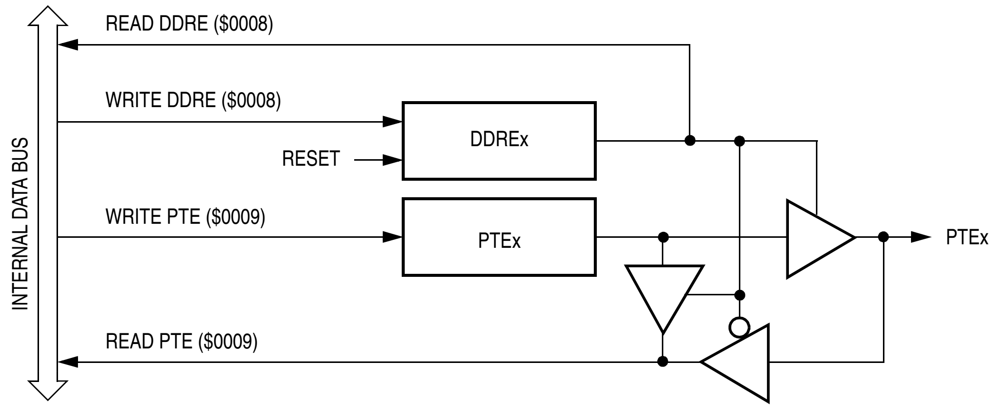


Figure 9-17. Port E I/O Circuit

When DDREx is a logic 1, reading address $0009 reads the PTEx data latch. When DDREx is a logic 0, reading address $0009 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit.

Table 9-5 summarizes the operation of the port E pins.

Table 9-6. Port E Pin Functions

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

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

Chapter 10
External Interrupt (IRQ)

10.1 Introduction

The external interrupt (IRQ) module provides a maskable interrupt input.

10.2 Features

Features of the IRQ module include the following:

- A dedicated external interrupt pin ($\overline{IRQ}$)
- IRQ interrupt control bits
- Hysteresis buffer
- Spike filter
- Programmable edge-only or edge and level interrupt sensitivity
- Automatic interrupt acknowledge
- Selectable internal pullup resistor

10.3 Functional Description

A logic zero applied to the external interrupt pin can latch a CPU interrupt request. Figure 10-1 shows the structure of the IRQ module.

Interrupt signals on the $\overline{IRQ}$ 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 (INTSCR). Writing a logic one 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 falling-edge and low-level-triggered. The MODE bit in the INTSCR controls the triggering sensitivity of the $\overline{IRQ}$ 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 one. 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 INTSCR 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 4.5 Exception Control.)

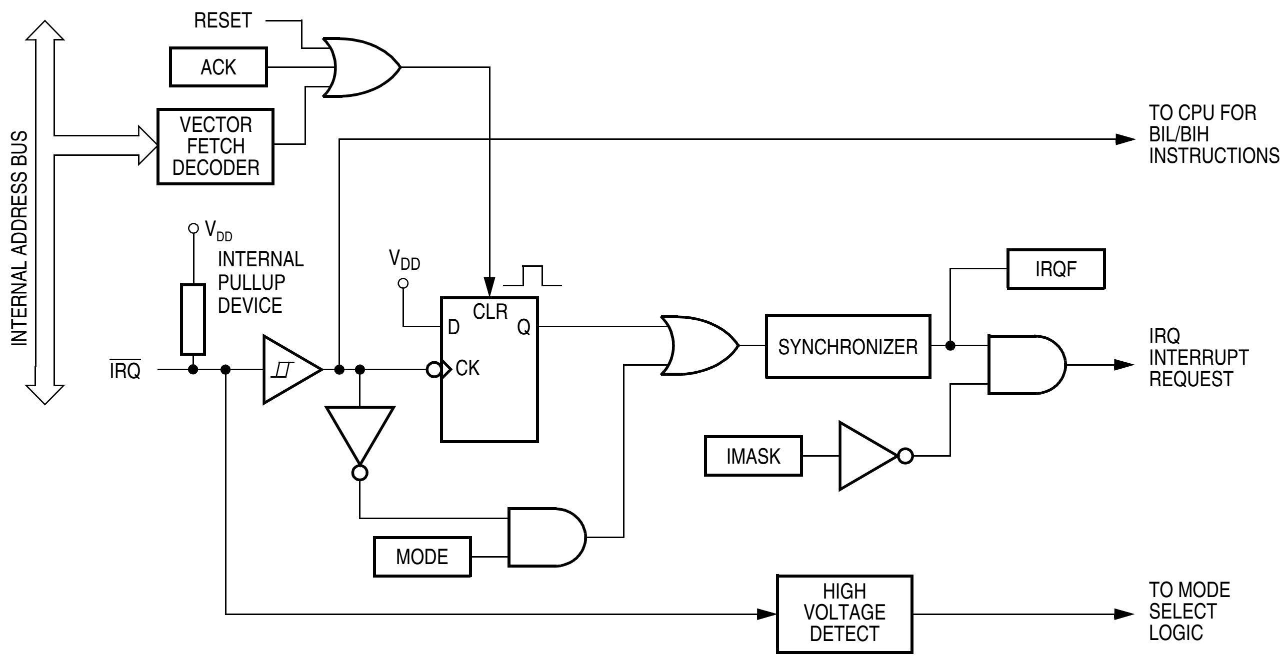


Figure 10-1. IRQ Module Block Diagram

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

(shaded) = Unimplemented

Figure 10-2. IRQ I/O Register Summary

10.3.1 $\overline{\text{IRQ}}$ Pin

A logic zero 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 one to the ACK bit in the interrupt status and control register (INTSCR). 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 $FFFA and $FFFB.
- 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.d 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.

NOTE

An internal pull-up resistor to V_{DD} *is connected to the* $\overline{\text{IRQ}}$ *pin; this can be disabled by setting the IRQPUD bit in the CONFIG2 register ($001E).*

10.4 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 Chapter 4 System Integration Module (SIM).)

To allow software to clear the IRQ latch during a break interrupt, write a logic one 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 zero to the BCFE bit. With BCFE at logic zero (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.

10.5 IRQ Status and Control Register (INTSCR)

The IRQ status and control register (INTSCR) controls and monitors operation of the IRQ module. The INTSCR has the following functions:

- Shows the state of the IRQ flag
- Clears the IRQ latch
- Masks IRQ and interrupt request
- Controls triggering sensitivity of the $\overline{\text{IRQ}}$ interrupt 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

(shaded) = Unimplemented

Figure 10-3. IRQ Status and Control Register (INTSCR)

IRQF — IRQ Flag Bit

This read-only status bit is high when the IRQ interrupt is pending.
1 = $\overline{\text{IRQ}}$ interrupt pending
0 = $\overline{\text{IRQ}}$ interrupt not pending

ACK — IRQ Interrupt Request Acknowledge Bit

Writing a logic one to this write-only bit clears the IRQ latch. ACK always reads as logic zero. Reset clears ACK.

IMASK — IRQ Interrupt Mask Bit

Writing a logic one 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 = $\overline{\text{IRQ}}$ interrupt requests on falling edges and low levels
0 = $\overline{\text{IRQ}}$ interrupt requests on falling edges only

Chapter 11
Keyboard Interrupt Module (KBI)

11.1 Introduction

The keyboard interrupt module (KBI) provides four independently maskable external interrupts which are accessible via PTA0–PTA3. When a port pin is enabled for keyboard interrupt function, an internal pull-up device is also enabled on the pin.

11.2 Features

Features of the keyboard interrupt module include the following:

- Four keyboard interrupt pins with pull-up devices
- Separate keyboard interrupt enable bits and one keyboard interrupt mask
- 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
$001B	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
$001C	Keyboard Interrupt Enable Register (KBIER)	Read:	0	0	0	0	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

(Shaded) = Unimplemented

Figure 11-1. KBI I/O Register Summary

11.3 I/O Pins

The eight keyboard interrupt pins are shared with standard port I/O pins. The full name of the KBI pins are listed in Table 11-1. The generic pin name appear in the text that follows.

Table 11-1. Pin Name Conventions

KBI Generic Pin Name	Full MCU Pin Name	Pin Selected for KBI Function by KBIEx Bit in KBIER
KBI0–KBI3	PTA0/KBI0–PTA3/KBI3	KBIE0–KBIE3

11.4 Functional Description

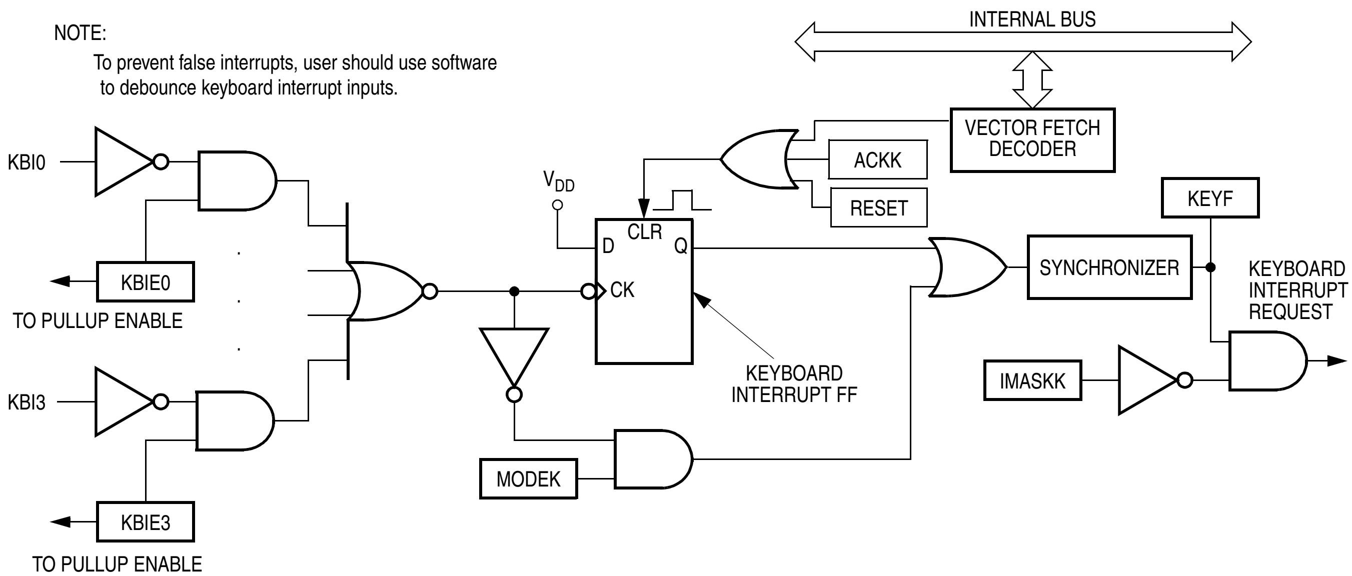


Figure 11-2. Keyboard Interrupt Block Diagram

Writing to the KBIE3–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 in port A also enables its internal pull-up device. A logic 0 applied to an enabled keyboard interrupt pin latches a keyboard interrupt request.

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

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

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

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

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

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

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

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

To determine the logic level on a keyboard interrupt pin, disable the pull-up device, use the data direction register to configure the pin as an input and then 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.

11.4.1 Keyboard Initialization

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

To prevent a false interrupt on keyboard initialization:

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

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

Another way to avoid a false interrupt:

1. Configure the keyboard pins as outputs by setting the appropriate DDRA bits in the data direction register A.
2. Write logic 1's 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.

11.5 Keyboard Interrupt Registers

Two registers control the operation of the keyboard interrupt module:

- Keyboard status and control register
- Keyboard interrupt enable register

11.5.1 Keyboard Status and Control Register

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

Address: $001B

	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

(shaded) = Unimplemented

Figure 11-3. Keyboard Status and Control Register (KBSCR)

KEYF — Keyboard Flag Bit

This read-only bit is set when a keyboard interrupt is pending on port A. 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 on port A. 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 on port A. 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 on port A. Reset clears MODEK.

1 = Keyboard interrupt requests on falling edges and low levels
0 = Keyboard interrupt requests on falling edges only

11.5.2 Keyboard Interrupt Enable Register

The port-A keyboard interrupt enable register enables or disables each port-A pin to operate as a keyboard interrupt pin.

Address: $001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	KBIE3	KBIE2	KBIE1	KBIE0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 11-4. Keyboard Interrupt Enable Register (KBIER)

KBIE3–KBIE0 — Port-A Keyboard Interrupt Enable Bits

Each of these read/write bits enables the corresponding keyboard interrupt pin on port-A to latch interrupt requests. Reset clears the keyboard interrupt enable register.

1 = KBIx pin enabled as keyboard interrupt pin
0 = KBIx pin not enabled as keyboard interrupt pin

11.6 Low-Power Modes

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

11.6.1 Wait Mode

The keyboard modules remain 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.

11.6.2 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.

11.7 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.

Chapter 12
Computer Operating Properly (COP)

12.1 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 CONFIG1 register.

12.2 Functional Description

Figure 12-1 shows the structure of the COP module.

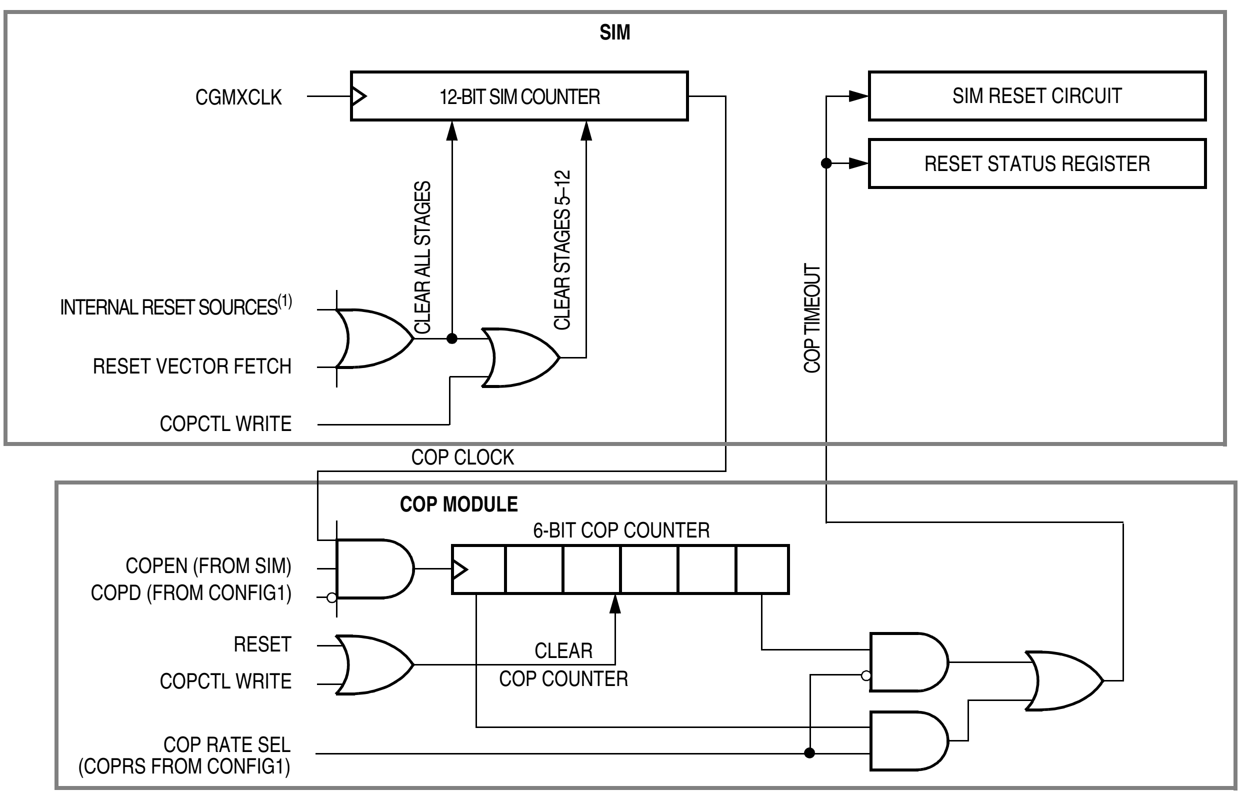


Figure 12-1. COP Block Diagram

The COP counter is a free-running 6-bit counter preceded by the 12-bit system integration module (SIM) counter. If not cleared by software, the COP counter overflows and generates an asynchronous reset after $2^{18} - 2^4$ or $2^{13} - 2^4$ CGMXCLK cycles; depending on the state of the COP rate select bit, COPRS, in CONFIG1 register. 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 SIM counter.

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 $\overline{RST}$ pin low for 32 × CGMXCLK cycles and sets the COP bit in the reset status register (RSR). (See 4.7.2 Reset Status Register (RSR).).

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.

12.3 I/O Signals

The following paragraphs describe the signals shown in Figure 12-1.

12.3.1 CMGXCLK

CGMXCLK is the crystal oscillator output signal. CGMXCLK frequency is equal to the crystal frequency.

12.3.2 COPCTL Write

Writing any value to the COP control register (COPCTL) (see 12.4 COP Control Register) clears the COP counter and clears bits 12 through 5 of the SIM counter. Reading the COP control register returns the low byte of the reset vector.

12.3.3 Power-On Reset

The power-on reset (POR) circuit in the SIM clears the SIM counter 4096 × CGMXCLK cycles after power-up.

12.3.4 Internal Reset

An internal reset clears the SIM counter and the COP counter.

12.3.5 Reset Vector Fetch

A reset vector fetch occurs when the vector address appears on the data bus. A reset vector fetch clears the SIM counter.

12.3.6 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the CONFIG1 register. (See 3.3 Configuration Register 1 (CONFIG1).)

12.3.7 COPRS (COP Rate Select)

The COPRS signal reflects the state of the COP rate select bit (COPRS) in the configuration register 1. (See 3.3 Configuration Register 1 (CONFIG1).)

12.4 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:	Clear COP counter							
Reset:	Unaffected by reset							

Figure 12-2. COP Control Register (COPCTL)

12.5 Interrupts

The COP does not generate CPU interrupt requests.

12.6 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.

12.7 Low-Power Modes

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

12.7.1 Wait Mode

The COP continues to operate during wait mode. To prevent a COP reset during wait mode, periodically clear the COP counter in a CPU interrupt routine.

12.7.2 Stop Mode

Stop mode turns off the CGMXCLK 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.

12.8 COP Module During Break Mode

The COP is disabled during a break interrupt when V_{TST} is present on the $\overline{RST}$ pin.

Chapter 13
Low-Voltage Inhibit (LVI)

13.1 Introduction

This section describes the low-voltage inhibit (LVI) module, which monitors the voltage on the V_{DD} pin and can force a reset when the V_{DD} voltage falls below the LVI trip falling voltage, V_{TRIPF}.

13.2 Features

Features of the LVI module include:

- Programmable LVI interrupt and reset
- Selectable LVI trip voltage
- Programmable stop mode operation

13.3 Functional Description

Figure 13-1 shows the structure of the LVI module.

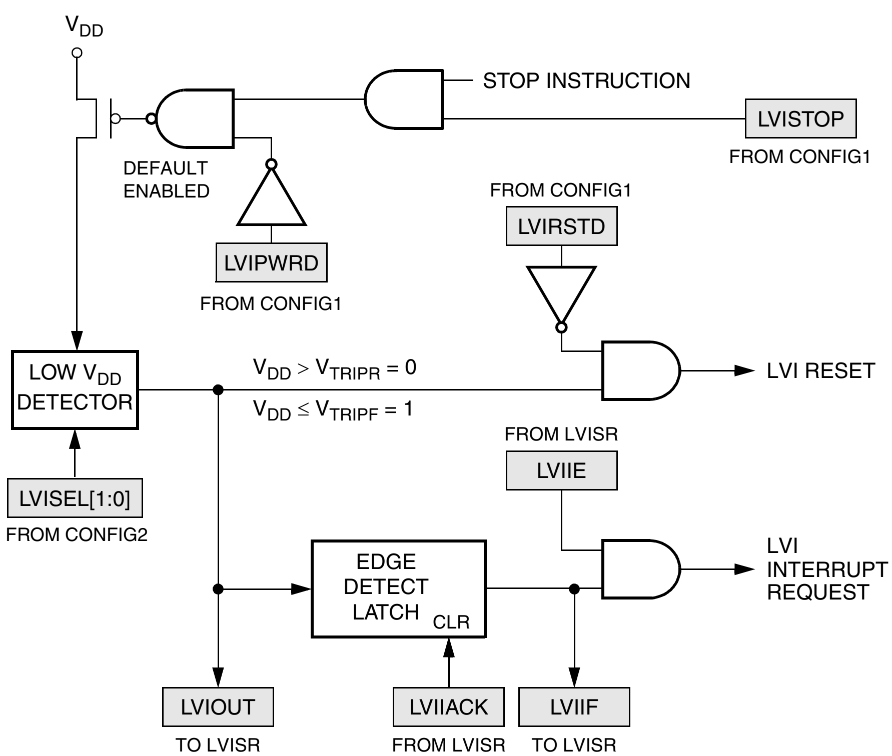


Figure 13-1. LVI Module Block Diagram

The LVI is enabled out of reset. The LVI module contains a bandgap reference circuit and comparator. Clearing the LVI power disable bit, LVIPWRD, enables the LVI to monitor V_{DD} voltage. Clearing the LVI reset disable bit, LVIRSTD, enables the LVI module to generate a reset when V_{DD} falls below a voltage, V_{TRIPF}. Setting the LVI enable in stop mode bit, LVISTOP, enables the LVI to operate in stop mode.

The LVI trip point selection bits, LVISEL[1:0], select the trip point voltage, V_{TRIPF}, to be configured for 5V or 3V operation. The actual trip points are shown in Chapter 16 Electrical Specifications.

Setting LVI interrupt enable bit, LVIIE, enables LVI interrupts whenever the LVIOUT bit toggles (from logic 0 to logic 1, or from logic 1 to logic 0).

NOTE

After a power-on reset (POR) the LVI's default mode of operation is 3V. If a 5V system is used, the user must modified the LVISEL[1:0] bits to raise the trip point to 5V operation. Note that this must be done after every power-on reset since the default will revert back to 3V mode after each power-on reset. If the V_{DD} supply is below the 3V mode trip voltage when POR is released, the MCU will immediately go into reset. The LVI in this case will hold the MCU in reset until either V_{DD} goes above the rising 3V trip point, V_{TRIPR}, which will release reset or V_{DD} decreases to approximately 0V which will re-trigger the power-on reset.

LVISTOP, LVIPWRD, LVIRSTD, and LVISEL[1:0] are in the configuration registers. See Section 5. Configuration Registers (CONFIG) for details of the LVI's configuration bits. Once an LVI reset occurs, the MCU remains in reset until V_{DD} rises above a voltage, V_{TRIPR}, which causes the MCU to exit reset. See 4.3.2.5 Low-Voltage Inhibit (LVI) Reset for details of the interaction between the SIM and the LVI. The output of the comparator controls the state of the LVIOUT flag in the LVI status register (LVISR). The LVIIE, LVIIF, and LVIIACK bits in the LVISR control LVI interrupt functions.

An LVI reset also drives the $\overline{RST}$ pin low to provide low-voltage protection to external peripheral devices.

13.3.1 Polled LVI Operation

In applications that can operate at V_{DD} levels below the V_{TRIPF} level, software can monitor V_{DD} by polling the LVIOUT bit, or by setting the LVI interrupt enable bit, LVIIE, to enable interrupt requests. In the configuration register 1 (CONFIG1), the LVIPWRD bit must be at logic 0 to enable the LVI module, and the LVIRSTD bit must be at logic 1 to disable LVI resets.

The LVI interrupt flag, LVIIF, is set whenever the LVIOUT bit changes state (toggles). When LVIF is set, a CPU interrupt request is generated if the LVIIE is also set. In the LVI interrupt service subroutine, LVIIF bit can be cleared by writing a logic 1 to the LVI interrupt acknowledge bit, LVIIACK.

13.3.2 Forced Reset Operation

In applications that require V_{DD} to remain above the V_{TRIPF} level, enabling LVI resets allows the LVI module to reset the MCU when V_{DD} falls below the V_{TRIPF} level. In the configuration register 1 (CONFIG1), the LVIPWRD and LVIRSTD bits must be at logic 0 to enable the LVI module and to enable LVI resets.

If LVIIE is set to enable LVI interrupts when LVIRSTD is cleared, LVI reset has a higher priority over LVI interrupt. In this case, when V_{DD} falls below the V_{TRIPF} level, an LVI reset will occur, and the LVIIE bit will be cleared.

13.3.3 Voltage Hysteresis Protection

Once the LVI has triggered (by having V_{DD} fall below V_{TRIPF}), the LVI will maintain a reset condition until V_{DD} rises above the rising trip point voltage, V_{TRIPR}. This prevents a condition in which the MCU is continually entering and exiting reset if V_{DD} is approximately equal to V_{TRIPF}. V_{TRIPR} is greater than V_{TRIPF} by the hysteresis voltage, V_{HYS}.

13.3.4 LVI Trip Selection

The trip point selection bits, LVISEL[1:0], in the CONFIG2 register select whether the LVI is configured for 5 V or 3 V operation. (See Chapter 3 Configuration Register (CONFIG).)

NOTE

The MCU is guaranteed to operate at a minimum supply voltage. The trip point (V_{TRIPF} [5 V] or V_{TRIPF} [3 V]) may be lower than this. (See Chapter 16 Electrical Specifications for the actual trip point voltages.)

13.4 LVI Status Register

The LVI status register (LVISR) controls LVI interrupt functions and indicates if the V_{DD} voltage was detected below the V_{TRIPF} level.

Address: $FE0F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	LVIOUT	LVIIE	LVIIF	0	0	0	0	0
Write:				LVIIACK				
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 13-2. LVI Status Register (LVISR)

LVIOUT — LVI Output Bit

This read-only flag becomes set when the V_{DD} voltage falls below the V_{TRIPF} trip voltage (see Table 13-1). Reset clears the LVIOUT bit.

Table 13-1. LVIOUT Bit Indication

V_{DD}	LVIOUT
$V_{DD} > V_{TRIPR}$	0
$V_{DD} < V_{TRIPF}$	1
$V_{TRIPF} < V_{DD} < V_{TRIPR}$	Previous value

LVIIE — LVI Interrupt Enable Bit

This read/write bit enables the LVIIF bit to generate CPU interrupt requests. Reset clears the LVIIE bit.

1 = LVIIF can generate CPU interrupt requests
0 = LVIIF cannot generate CPU interrupt requests

LVIIF — LVI Interrupt Flag

This clearable, read-only flag is set whenever the LVIOUT bit toggles. Reset clears the LVIIF bit.

1 = LVIOUT has toggled
0 = LVIOUT has not toggled

LVIIACK — LVI Interrupt Acknowledge Bit

Writing a logic 1 to this write-only bit clears the LVI interrupt flag, LVIIF. LVIIACK always reads as logic 0.

1 = Clears LVIIF bit
0 = No effect

13.5 Low-Power Modes

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

13.5.1 Wait Mode

If enabled, the LVI module remains active in wait mode. If enabled to generate resets or interrupts, the LVI module can generate a reset or an interrupt and bring the MCU out of wait mode.

13.5.2 Stop Mode

If enabled in stop mode (LVISTOP = 1), the LVI module remains active in stop mode. If enabled to generate resets or interrupts, the LVI module can generate a reset or an interrupt and bring the MCU out of stop mode.

NOTE

If enabled to generate both resets and interrupts, there will be no LVI interrupts, as resets have a higher priority.

Chapter 14
Central Processor Unit (CPU)

14.1 Introduction

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

14.2 Features

Features 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
- 4-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

14.3 CPU Registers

Figure 14-1 shows the five CPU registers. CPU registers are not part of the memory map.

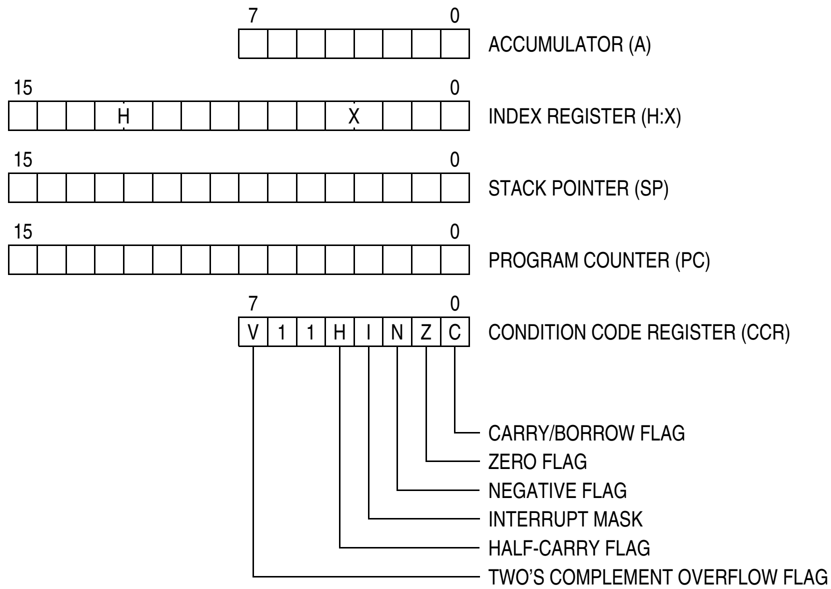


Figure 14-1. CPU Registers

14.3.1 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.

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

Figure 14-2. Accumulator (A)

14.3.2 Index Register

The 16-bit index register allows indexed addressing of a 64-Kbyte 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.

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

X = Indeterminate

Figure 14-3. Index Register (H:X)

14.3.3 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.

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

Figure 14-4. Stack Pointer (SP)

NOTE

The location of the stack is arbitrary and may be relocated anywhere in random-access memory (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.

14.3.4 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.

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
Read: Write:																
Reset:	Loaded with vector from $FFFE and $FFFF															

Figure 14-5. Program Counter (PC)

14.3.5 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 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 14-6. 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

14.4 Arithmetic/Logic Unit (ALU)

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

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

14.5 Low-Power Modes

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

14.5.1 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

14.5.2 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.

14.6 CPU During Break Interrupts

If a break module is present on the MCU, the CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with $FFFC:$FFFD or with $FEFC:$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.

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.

14.7 Instruction Set Summary

Table 14-1 provides a summary of the M68HC08 instruction set.

Table 14-1. Instruction Set Summary (Sheet 1 of 6)

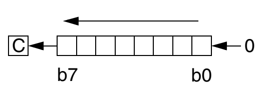
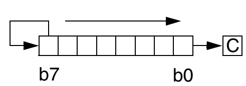
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X ADC *opr*,SP ADC *opr*,SP	Add with Carry	A ← (A) + (M) + (C)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A9 B9 C9 D9 E9 F9 9EE9 9ED9	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
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

Table 14-1. Instruction Set Summary (Sheet 2 of 6)

Source Form	Operation	Description	Effect on CCR V	H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
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
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

Table 14-1. Instruction Set Summary (Sheet 3 of 6)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
CLR *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
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
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

Table 14-1. Instruction Set Summary (Sheet 4 of 6)

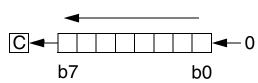
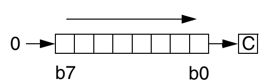
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff	2 3 4 3 2
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,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 LSL ,X LSL *opr*,SP	Logical Shift Left (Same as ASL)		↕	–	–	↕	↕	↕	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
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

Table 14-1. Instruction Set Summary (Sheet 5 of 6)

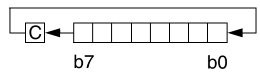
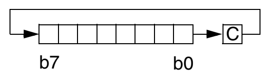
Source Form	Operation	Description	Effect on CCR V	H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
PULA	Pull A from Stack	SP ← (SP + 1); Pull (A)	–	–	–	–	–	–	INH	86		2
PULH	Pull H from Stack	SP ← (SP + 1); Pull (H)	–	–	–	–	–	–	INH	8A		2
PULX	Pull X from Stack	SP ← (SP + 1); Pull (X)	–	–	–	–	–	–	INH	88		2
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
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 Interrupts, Stop Processing, Refer to MCU Documentation	I ← 0; Stop Processing	–	–	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 14-1. Instruction Set Summary (Sheet 6 of 6)

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
WAIT	Enable Interrupts; Wait for Interrupt	I bit ← 0; Inhibit CPU clocking until interrupted	–	–	0	–	–	–	INH	8F		1

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

14.8 Opcode Map

See Table 14-2.

Table 14-2. 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
LSB \ MSB	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

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

Chapter 15
Development Support

15.1 Introduction

This section describes the break module, the monitor module (MON), and the monitor mode entry methods.

15.2 Break Module (BRK)

The break module can generate a break interrupt that stops normal program flow at a defined address to enter a background program.

Features include:

- Accessible input/output (I/O) registers during the break Interrupt
- Central processor unit (CPU) generated break interrupts
- Software-generated break interrupts
- Computer operating properly (COP) disabling during break interrupts

15.2.1 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal ($\overline{\text{BKPT}}$) to the SIM. The SIM then causes the CPU to load 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 one 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 15-1 shows the structure of the break module.

When the internal address bus matches the value written in the break address registers or when software writes a 1 to the BRKA bit in the break status and control register, 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)

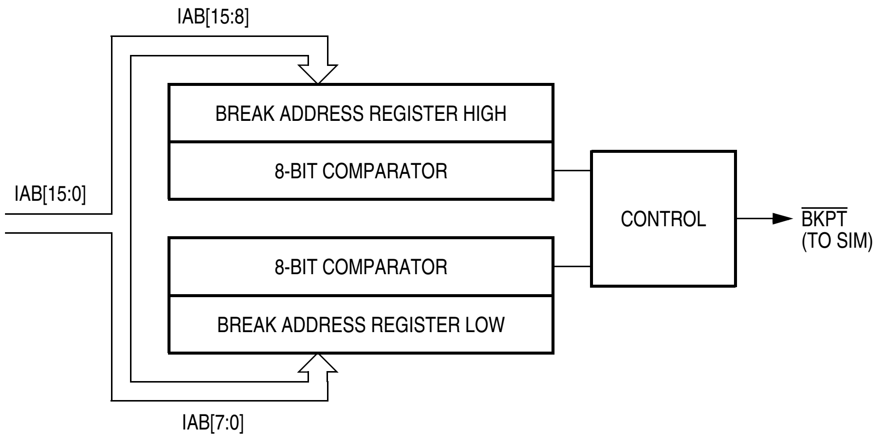


Figure 15-1. Break Module Block Diagram

The break interrupt timing is:

- When a break address is placed at the address of the instruction opcode, the instruction is not executed until after completion of the break interrupt routine.
- When a break address is placed at an address of an instruction operand, the instruction is executed before the break interrupt.
- When software writes a 1 to the BRKA bit, the break interrupt occurs just before the next instruction is executed.

By updating a break address and clearing the BRKA bit in a break interrupt routine, a break interrupt can be generated continuously.

CAUTION

A break address should be placed at the address of the instruction opcode. When software does not change the break address and clears the BRKA bit in the first break interrupt routine, the next break interrupt will not be generated after exiting the interrupt routine even when the internal address bus matches the value written in the break address registers.

15.2.1.1 Flag Protection During Break Interrupts

The system integration module (SIM) controls whether or not module status bits 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. (See 4.7.3 SIM Break Flag Control Register and the “Break Interrupts” subsection for each module.)

15.2.1.2 TIM During Break Interrupts

A break interrupt stops the timer counter.

15.2.1.3 COP During Break Interrupts

The COP is disabled during a break interrupt when V_{TST} is present on the $\overline{RST}$ pin.

15.2.2 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)
- Break status register (BSR)
- Break flag control register (BFCR)

15.2.2.1 Break Status and Control Register (BRKSCR)

The break status and control register 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

(Shaded) = Unimplemented

Figure 15-2. 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 zero to bit 7. Reset clears the BRKE bit.

1 = Breaks enabled on 16-bit address match
0 = Breaks disabled

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a logic one to BRKA generates a break interrupt. Clear BRKA by writing a logic zero to it before exiting the break routine. Reset clears the BRKA bit.

1 = Break address match
0 = No break address match

15.2.2.2 Break Address Registers

The break address registers 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 15-3. 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 15-4. Break Address Register Low (BRKL)

15.2.2.3 Break Status Register

The 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	

R = Reserved

1. Writing a logic zero clears SBSW.

Figure 15-5. Break Status Register (BSR)

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 zero 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.

15.2.2.4 Break Flag Control Register (BFCR)

The break 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 15-6. Break Flag Control Register (BFCR)

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

15.2.3 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes. If enabled, the break module will remain enabled in wait and stop modes. However, since the internal address bus does not increment in these modes, a break interrupt will never be triggered.

15.3 Monitor Module (MON)

The monitor module allows complete testing of the microcontroller unit (MCU) through a single-wire interface with a host computer. Monitor mode entry can be achieved without use of the higher test voltage, V_{TST}, as long as vector addresses $FFFE and $FFFF are blank, thus reducing the hardware requirements for in-circuit programming.

Features of the monitor module include:

- Normal user-mode pin functionality
- One pin dedicated to serial communication between MCU and host computer
- Standard non-return-to-zero (NRZ) communication with host computer
- Standard communication baud rate (9600 @ 2.4576-MHz internal operating frequency)
- Execution of code in random-access memory (RAM) or ROM
- ROM security feature[1]
- Use of external 4.9152MHz or 9.8304MHz oscillator to generate internal operating frequency of 2.4576 MHz
- Normal monitor mode entry if V_{TST} is applied to $\overline{IRQ}$

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

Chapter 16
Electrical Specifications

16.1 Introduction

This section contains electrical and timing specifications.

16.2 Absolute Maximum Ratings

Maximum ratings are the extreme limits to which the microcontroller unit (MCU) can be exposed without permanently damaging it.

NOTE

This device is not guaranteed to operate properly at the maximum ratings. Refer to 16.5 5-V DC Electrical Characteristics and 16.6 3-V DC Electrical Characteristics for guaranteed operating conditions.

Table 16-1. Absolute Maximum Ratings

Characteristic(1)	Symbol	Value	Unit
Supply voltage	V_{DD}	−0.3 to +6.0	V
LCD voltage	V_{LCD}	V_{SS} to +6.0	
Input voltage	V_{IN}	V_{SS}−0.3 to V_{DD} +0.3	V
Mode entry voltage, $\overline{IRQ}$ pin	V_{TST}	V_{SS}−0.3 to +8.5	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}	I_{MVSS}	100	mA
Maximum current into V_{DD}	I_{MVDD}	100	mA

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 $V_{SS} \leq (V_{IN}$ or $V_{OUT}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD}.)

16.3 Functional Operating Range

Table 16-2. Operating Range

Characteristic	Symbol	Value	Unit
Operating temperature range	T_A (T_L to T_H)	– 40 to +85	°C
Operating voltage range	V_{DD}	1.8 to 5.5(1)	V
LCD voltage	V_{LCD}	V_{SS} to V_{DD}	V

1. For normal operation, user should ensure that the device supply voltage is above the LVI trip voltage.

16.4 Thermal Characteristics

Table 16-3. Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance 44-pin LQFP	θ_{JA}	85	°C/W
I/O pin power dissipation	$P_{I/O}$	User determined	W
Power dissipation(1)	P_D	$P_D = (I_{DD} \times V_{DD}) + P_{I/O} = K/(T_J + 273\ °C)$	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

1. Power dissipation is a function of temperature.
2. K constant unique to the device. K can be determined for a known T_A and measured P_D. With this value of K, P_D and T_J can be determined for any value of T_A.

16.5 5-V DC Electrical Characteristics

Table 16-4. DC Electrical Characteristics (5V)

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage (I_{LOAD} = –1.6mA) All ports	V_{OH}	V_{DD}–0.7	—	—	V
Output low voltage (I_{LOAD} = 2mA) All ports except PTB2–PTB3 (I_{LOAD} = 15mA) PTB2–PTB3	V_{OL}	—	—	0.7	V
Input high voltage All ports, $\overline{RST}$, $\overline{IRQ}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage All ports, $\overline{RST}$, $\overline{IRQ}$, OSC1	V_{IL}	V_{SS}	—	0.3 × V_{DD}	V
V_{DD} supply current, f_{OP} = 4MHz (R_{LCD} = 146kΩ) Run[3] with all modules on Wait[4] with all modules off (except PPI, LCD, and LVI) Stop[5] –40 to 85°C (All modules off) 25°C (XTAL, LCD, PPI enabled) 25°C (XTAL, LCD, PPI, LVI enabled) 25°C (XTAL, LCD, PPI, LVI, OSC enabled)	I_{DD}	— — — — — —	5 3 0.3 25 175 1.5	12 6 1 50 190 3	mA mA μA μA μA mA
Digital 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 rearm voltage[6]	V_{POR}	750	—	—	mV
POR rise time ramp rate[7]	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{TST}	V_{DD} + 2.5	—	9.1	V
Pullup resistors[8] PTA0–PTA3 as KBI0–KBI3, $\overline{RST}$, $\overline{IRQ}$	R_{PU}	50	100	200	kΩ
Low-voltage inhibit, trip falling voltage	V_{TRIPF}	3.90	4.20	4.50	V
Low-voltage inhibit, trip rising voltage	V_{TRIPR}	3.95	4.30	4.60	V
Low-voltage inhibit reset/recovery hysteresis	V_{HYS}	—	100	—	mV

1. V_{DD} = 4.5 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_{OP} = 4MHz). All inputs 0.2V 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_{OP} = 4MHz). All inputs 0.2V 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 wait I_{DD}.
5. Stop I_{DD} measured with 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.
8. R_{PU} is measured at V_{DD} = 5.0V.

16.6 3-V DC Electrical Characteristics

Table 16-5. DC Electrical Characteristics (3 V)

Characteristic(1)	Symbol	Min	Typ(2)	Max	Unit
Output high voltage (I_{LOAD} = –0.4 mA) All ports	V_{OH}	V_{DD} – 0.7	—	—	V
Output low voltage (I_{LOAD} = 0.5 mA) All ports except PTB2–PTB3 (I_{LOAD} = 10 mA) PTB2–PTB3	V_{OL}	—	—	0.7	V
Input high voltage All ports, $\overline{RST}$, $\overline{IRQ}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage All ports, $\overline{RST}$, $\overline{IRQ}$, OSC1	V_{IL}	V_{SS}	—	0.3 × V_{DD}	V
V_{DD} supply current, f_{OP} = 2 MHz (R_{LCD} = 146 kΩ) Run(3) with all modules on Wait(4) with all modules off, except PPI, LCD, and LVI Stop(5) –40 to 85°C (All modules off) 25°C (XTAL, LCD, PPI enabled) 25°C (XTAL, LCD, PPI, LVI enabled) 25°C (XTAL, LCD, PPI, LVI, OSC enabled)	I_{DD}	 — — — — — —	 3 2 0.2 7 150 1	 6 4 1 25 165 2	 mA mA μA μA μA mA
Digital 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 rearm voltage(6)	V_{POR}	750	—	—	mV
POR rise time ramp rate(7)	R_{POR}	0.02	—	—	V/ms
Monitor mode entry voltage	V_{TST}	V_{DD} + 2.5	—	9.1	V
Pullup resistors(8) PTA0–PTA3 as KBI0–KBI3, $\overline{RST}$, $\overline{IRQ}$	R_{PU}	50	100	200	kΩ
Low-voltage inhibit, trip falling voltage	V_{TRIPF}	1.80	1.90	2.00	V
Low-voltage inhibit, trip rising voltage	V_{TRIPR}	1.85	1.97	2.10	V
Low-voltage inhibit reset/recovery hysteresis	V_{HYS}	—	70	—	mV

1. V_{DD} = 2.7 to 3.3 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_{OP} = 2 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_{OP} = 2 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 wait I_{DD}.
5. Stop I_{DD} measured with 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.
8. R_{PU} is measured at V_{DD} = 5.0 V.

16.7 2-V DC Electrical Characteristics

Table 16-6. DC Electrical Characteristics (2V)

Characteristic(1)	Symbol	Min	Typ(2)	Max	Unit
Output high voltage (I_{LOAD} = –0.4 mA) All ports	V_{OH}	V_{DD} – 0.5	—	—	V
Output low voltage (I_{LOAD} = 0.5 mA) All ports except PTB2–PTB3 (I_{LOAD} = 8 mA) PTB2–PTB3	V_{OL}	—	—	0.5	V
Input high voltage All ports, $\overline{RST}$, $\overline{IRQ}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage All ports, $\overline{RST}$, $\overline{IRQ}$, OSC1	V_{IL}	V_{SS}	—	0.3 × V_{DD}	V
V_{DD} supply current, f_{OP} = 1 MHz (R_{LCD} = 146 kΩ) Run(3) with all modules on Wait(4) with all modules off, except PPI, LCD, and LVI Stop(5) –40 to 85°C (All modules off) 25°C (XTAL, LCD, PPI enabled) 25°C (XTAL, LCD, PPI, LVI enabled) 25°C (XTAL, LCD, PPI, LVI, OSC enabled)	I_{DD}	— — — — — —	0.8 0.6 0.2 5 147 0.5	2 1.8 1 15 155 1	mA mA μA μA μA mA
Digital 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 rearm voltage(6)	V_{POR}	750	—	—	mV
POR rise time ramp rate(7)	R_{POR}	0.02	—	—	V/ms
Monitor mode entry voltage	V_{TST}	V_{DD} + 2.5	—	9.1	V
Pullup resistors(8) PTA0–PTA3 as KBI0–KBI3, $\overline{RST}$, $\overline{IRQ}$	R_{PU}	50	100	200	kΩ

1. V_{DD} = 1.8 to 2.2 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_{OP} = 1 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_{OP} = 1 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 wait I_{DD}.
5. Stop I_{DD} measured with 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.
8. R_{PU} is measured at V_{DD} = 5.0 V.

16.8 5-V Control Timing

Table 16-7. Control Timing (5V)

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency	f_{OP}	—	4	MHz
$\overline{RST}$ input pulse width low[2]	t_{IRL}	100	—	ns
$\overline{IRQ}$ interrupt pulse width low (edge-triggered)[3]	t_{ILIH}	100	—	ns
$\overline{IRQ}$ interrupt pulse period[3]	t_{ILIL}	Note[4]	—	t_{CYC}

1. V_{DD} = 4.5 to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H; timing shown with respect to 20% V_{DD} and 70% V_{SS}, unless otherwise noted.
2. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.
3. Values are based on characterization results, not tested in production.
4. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1 t_{CYC}.

16.9 3-V Control Timing

Table 16-8. Control Timing (3V)

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency	f_{OP}	—	2	MHz
$\overline{RST}$ input pulse width low[2]	t_{IRL}	250	—	ns
$\overline{IRQ}$ interrupt pulse width low (edge-triggered)[3]	t_{ILIH}	250	—	ns
$\overline{IRQ}$ interrupt pulse period[3]	t_{ILIL}	Note[4]	—	t_{CYC}

1. V_{DD} = 2.7 to 3.3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H; timing shown with respect to 20% V_{DD} and 70% V_{SS}, unless otherwise noted.
2. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.
3. Values are based on characterization results, not tested in production.
4. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1 t_{CYC}.

16.10 2-V Control Timing

Table 16-9. Control Timing (2V)

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency	f_{OP}	—	1	MHz
$\overline{RST}$ input pulse width low[2]	t_{IRL}	500	—	ns
$\overline{IRQ}$ interrupt pulse width low (edge-triggered)[3]	t_{ILIH}	500	—	ns
$\overline{IRQ}$ interrupt pulse period[3]	t_{ILIL}	Note[4]	—	t_{CYC}

1. V_{DD} = 1.8 to 2.2 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H; timing shown with respect to 20% V_{DD} and 70% V_{SS}, unless otherwise noted.
2. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.
3. Values are based on characterization results, not tested in production.
4. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1 t_{CYC}.

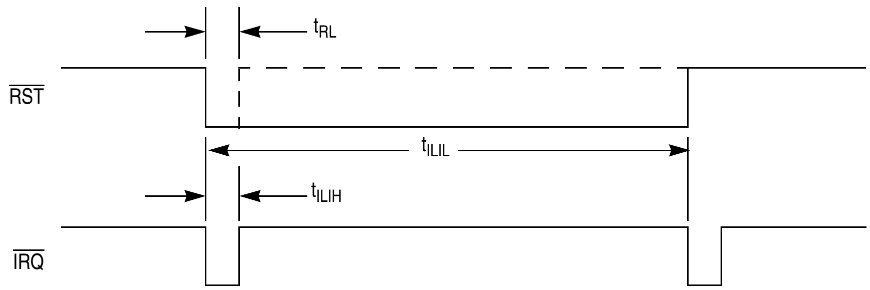


Figure 16-1. $\overline{RST}$ and $\overline{IRQ}$ Timing

16.11 Oscillator Characteristics

Table 16-10. Oscillator Specifications

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator 1 on OSC1 and OSC2 (for system bus)					
External reference clock to OSC1 (1)	f_{OSC}	dc	—	16M	Hz
Crystal reference frequency (2)	$f_{XTALCLK1}$	1M	—	16M	Hz
Crystal load capacitance (3)	C_L	—	—	—	
Crystal fixed capacitance (3)	C_1	—	18	—	pF
Crystal tuning capacitance (3)	C_2	—	18	—	pF
Feedback bias resistor	R_{B1}	—	4.7	—	MΩ
Series resistor	R_{S1}	—	0	—	kΩ
Oscillator 2 on XTAL1 and XTAL2 (for LCD and PPI)					
Crystal reference frequency (2)	$f_{XTALCLK2}$	—	32.768k	—	Hz
Crystal load capacitance (3)	C_L	—	—	—	
Crystal fixed capacitance (3)	C_3	—	10	—	pF
Crystal tuning capacitance (3)	C_4	—	10	—	pF
Feedback bias resistor	R_{B2}	—	10	—	MΩ
Series resistor	R_{S2}	—	10	—	kΩ

1. No more than 10% duty cycle deviation from 50%.
2. Use fundamental mode only, do not use overtone crystals or overtone ceramic resonators.
3. Consult crystal vendor data sheet.

16.12 Timer Interface Module Characteristics

Table 16-11. Timer Interface Module Characteristics

Characteristic	Symbol	Min	Max	Unit
Input capture pulse width	t_{TIH}, t_{TIL}	$1/f_{OP}$	—	

Chapter 17
Ordering Information and Mechanical Specifications

17.1 Introduction

This section contains order numbers for the MC68HC08LT8. Dimensions are given for:

- 44-pin low-profile quad flat pack (LQFP)

17.2 MC Order Numbers

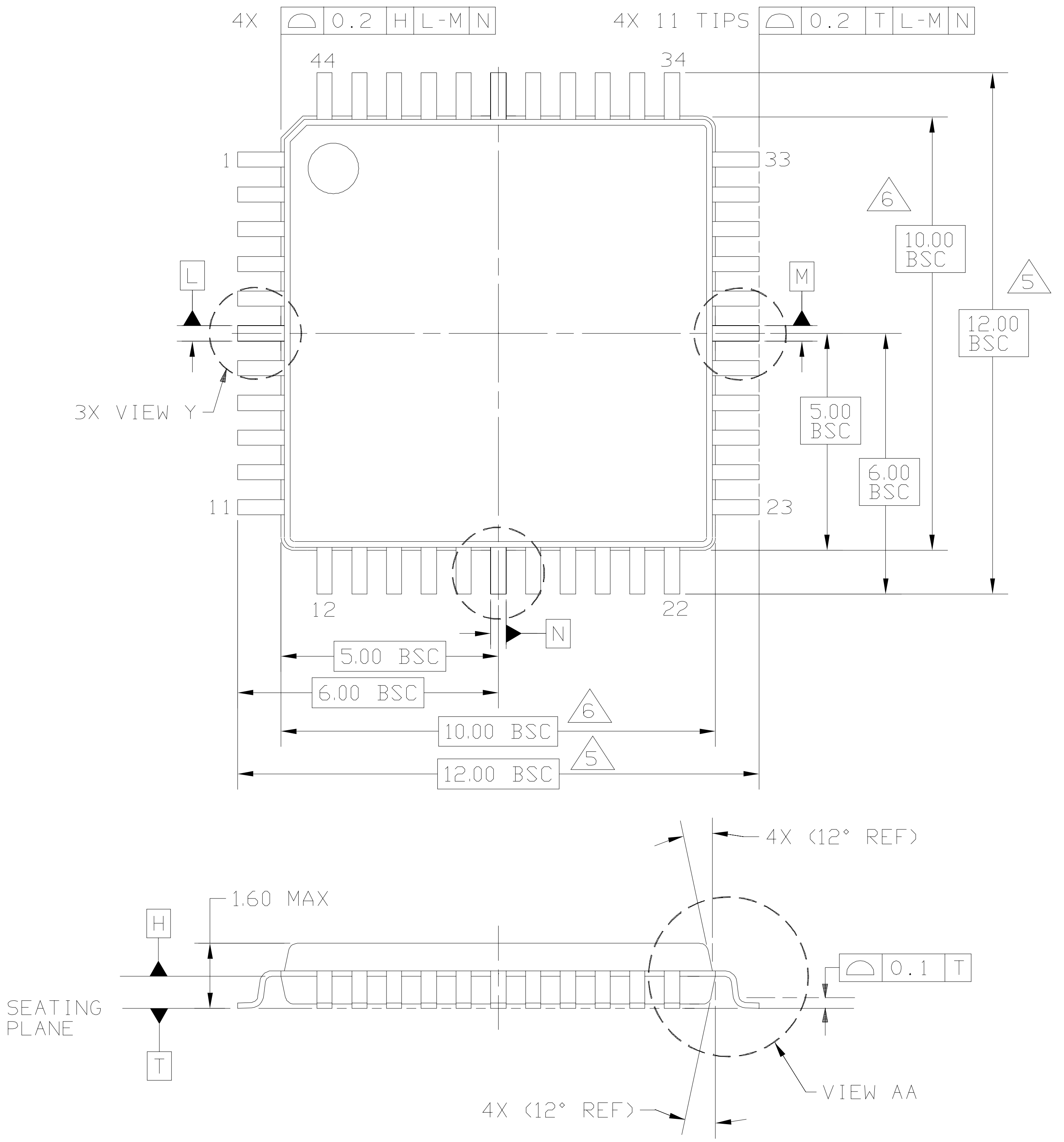
These part numbers are generic numbers only. To place an order, ROM code must be submitted to the ROM Processing Center (RPC).

Table 17-1. MC Order Numbers

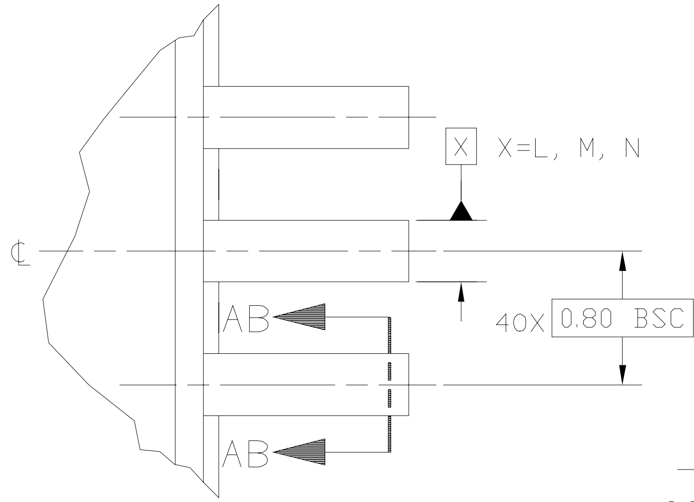
MC Order Number	Operating Temperature Range	Package	RoHS Compliant
MC68HC08LT8CFGE	–40 to +85 °C	44-pin LQFP	Yes

17.3 Package Dimensions

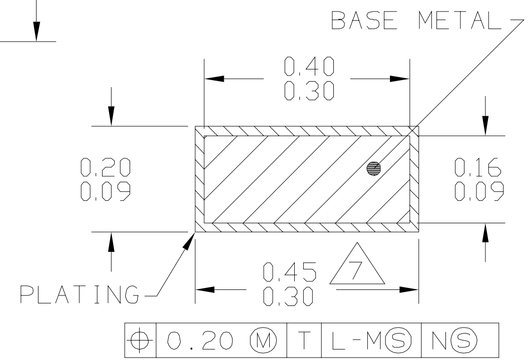
Refer to the following pages for detailed package dimensions.



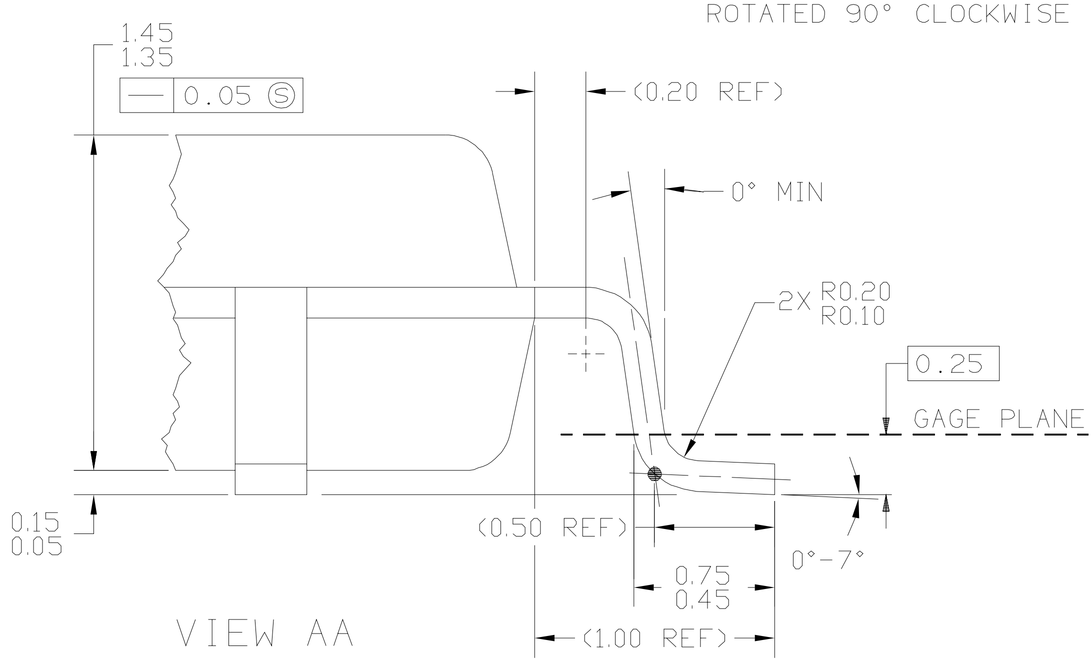
	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: 44 LD TQFP, 10 X 10 PKG, 0.8 PITCH, 1.4 THICK	DOCUMENT NO: 98ASS23225W	REV: C
	CASE NUMBER: 824D-02	19 MAY 2005
	STANDARD: JEDEC MS-026-BCB	



VIEW Y



SECTION AB-AB
ROTATED 90° CLOCKWISE



VIEW AA

	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: 44 LD TQFP, 10 X 10 PKG, 0.8 PITCH, 1.4 THICK	DOCUMENT NO: 98ASS23225W	REV: C
	CASE NUMBER: 824D-02	19 MAY 2005
	STANDARD: JEDEC MS-026 BCB	

NOTES:

1. DIMENSIONS AND TOLERANCING PER ASME Y14.5M-1994.

2. CONTROLLING DIMENSION: MILLIMETER

3. DATUM PLANE H 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 L, M AND N TO BE DETERMINED AT DATUM PLANE H.

△5. DIMENSIONS TO BE DETERMINED AT SEATING PLANE T.

△6. DIMENSIONS DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 PER SIDE. DIMENSIONS DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE H.

△7. DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE DIMENSION TO EXCEED 0.53. MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD OR PROTRUSION 0.07.

	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: 44 LD TQFP, 10 X 10 PKG, 0.8 PITCH, 1.4 THICK	DOCUMENT NO: 98ASS23225W	REV: C
	CASE NUMBER: 824D-02	19 MAY 2005
	STANDARD: JEDEC MS-026 BCB	

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

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

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

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

MC68HC08LT8
Rev. 1, 3/2006

freescale™
semiconductor

AMEYA360

Components Supply Platform

Authorized Distribution Brand：

Website：

Welcome to visit www.ameya360.com

Contact Us：

➢ Address：

401 Building No.5, JiuGe Business Center, Lane 2301, Yishan Rd
Minhang District, Shanghai , China

➢ Sales：

Direct +86 (21) 6401-6692

Email amall@ameya360.com

QQ 800077892

Skype ameyasales1 ameyasales2

➢ Customer Service：

Email service@ameya360.com

➢ Partnership：

Tel +86 (21) 64016692-8333

Email mkt@ameya360.com