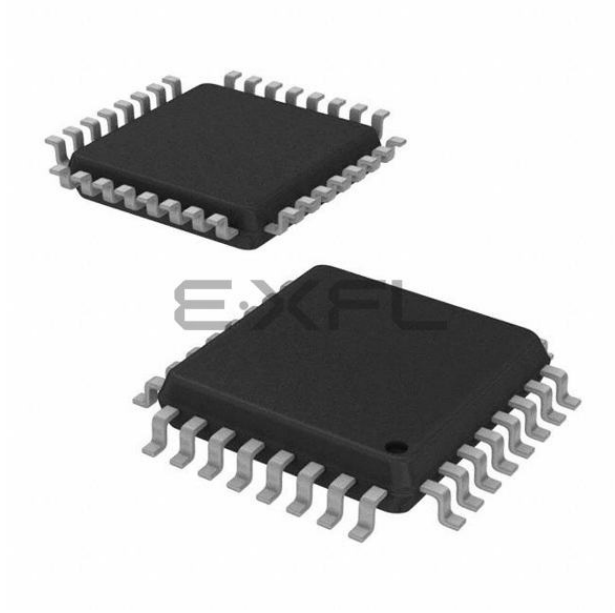


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	LINbus, SCI, SPI
Peripherals	POR, PWM
Number of I/O	24
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908ey8acfje

Chapter 14
System Integration Module (SIM)

See Chapter 12 Input/Output (I/O) Ports (PORTS) and Chapter 13 Enhanced Serial Communications Interface (ESCI) Module.

NOTE

Any unused inputs and I/O ports should be tied to an appropriate logic level (either V_{DD} or V_{SS}). Although the I/O ports of the MC68HC908EY16A do not require termination, termination is recommended to reduce the possibility of electro-static discharge damage.

1.6 Pin Summary

Table 1-1. External Pin Summary

Pin Name	Function	Driver Type	Hysteresis[1]	Reset State
PTA6/$\overline{SS}$	General-Purpose I/O SPI Slave Select	Dual State	Yes	Input Hi-Z
PTA5/SPSCK	General-Purpose I/O SPI Clock	Dual State	Yes	Input Hi-Z
PTA4/KBD4	General-Purpose I/O Keyboard Wakeup Pin	Dual State	Yes	Input Hi-Z
PTA3/KBD3/RxD	General-Purpose I/O Keyboard Wakeup Pin SCI Receive Data	Dual State	Yes	Input Hi-Z
PTA2/KBD2/TxD	General-Purpose I/O Keyboard Wakeup Pin SCI Transmit Data	Dual State	Yes	Input Hi-Z
PTA1/KBD1	General-Purpose I/O Keyboard Wakeup Pin	Dual State	Yes	Input Hi-Z
PTA0/KBD0	General-Purpose I/O Keyboard Wakeup Pin	Dual State	Yes	Input Hi-Z
PTB7/ATD7/TBCH1	General-Purpose I/O ADC Channel Timer B Channel 1	Dual State	Yes	Input Hi-Z
PTB6/ATD6/TBCH0	General-Purpose I/O ADC Channel Timer B Channel 0	Dual State	Yes	Input Hi-Z
PTB5/ATD5/SPSCK	General-Purpose I/O ADC Channel SPI Clock	Dual State	Yes	Input Hi-Z
PTB4/ATD4/MOSI	General-Purpose I/O ADC Channel SPI Data Path	Dual State	Yes	Input Hi-Z
PTB3/ATD3/MISO	General-Purpose I/O ADC Channel SPI Data Path	Dual State	Yes	Input Hi-Z
PTB2/ATD2	General-Purpose I/O ADC Channel	Dual State	Yes	Input Hi-Z

When a conversion is aborted, the contents of the data registers, ADRH and ADRL, are not altered but continue to be the values transferred after the completion of the last successful conversion. In the case that the conversion was aborted by a reset, ADRH and ADRL return to their reset states.

Upon reset or when a conversion is otherwise aborted, the ADC10 module will enter a low power, inactive state. In this state, all internal clocks and references are disabled. This state is entered asynchronously and immediately upon aborting of a conversion.

3.3.3.4 Total Conversion Time

The total conversion time depends on many factors such as sample time, bus frequency, whether ACLKEN is set, and synchronization time. The total conversion time is summarized in Table 3-1.

Table 3-1. Total Conversion Time versus Control Conditions

Conversion Mode	ACLKEN	Maximum Conversion Time
8-Bit Mode (short sample — ADLSMP = 0):		
Single or 1st continuous	0	18 ADCK + 3 bus clock
Single or 1st continuous	1	18 ADCK + 3 bus clock + 5 μs
Subsequent continuous ($f_{Bus} \geq f_{ADCK}$)	x	16 ADCK
8-Bit Mode (long sample — ADLSMP = 1):		
Single or 1st continuous	0	38 ADCK + 3 bus clock
Single or 1st continuous	1	38 ADCK + 3 bus clock + 5 μs
Subsequent continuous ($f_{Bus} \geq f_{ADCK}$)	x	36 ADCK
10-Bit Mode (short sample — ADLSMP = 0):		
Single or 1st continuous	0	21 ADCK + 3 bus clock
Single or 1st continuous	1	21 ADCK + 3 bus clock + 5 μs
Subsequent continuous ($f_{Bus} \geq f_{ADCK}$)	x	19 ADCK
10-Bit Mode (long sample — ADLSMP = 1):		
Single or 1st continuous	0	41 ADCK + 3 bus clock
Single or 1st continuous	1	41 ADCK + 3 bus clock + 5 μs
Subsequent continuous ($f_{Bus} \geq f_{ADCK}$)	x	39 ADCK

The maximum total conversion time for a single conversion or the first conversion in continuous conversion mode is determined by the clock source chosen and the divide ratio selected. The clock source is selectable by the ADICLK and ACLKEN bits, and the divide ratio is specified by the ADIV bits. For example, if the alternate clock source is 16 MHz and is selected as the input clock source, the input clock divide-by-8 ratio is selected and the bus frequency is 4 MHz, then the conversion time for a single 10-bit conversion is:

$$\text{Maximum Conversion time} = \frac{21 \text{ ADCK cycles}}{16 \text{ MHz}/8} + \frac{3 \text{ bus cycles}}{4 \text{ MHz}} = 11.25 \ \mu\text{s}$$

$$\text{Number of bus cycles} = 11.25 \ \mu\text{s} \times 4 \text{ MHz} = 45 \text{ cycles}$$

NOTE

The ADCK frequency must be between f_{ADCK} minimum and f_{ADCK} maximum to meet A/D specifications.

Chapter 8
Internal Clock Generator (ICG) Module

8.1 Introduction

The internal clock generator (ICG) module is used to create a stable clock source for the microcontroller without using any external components. The ICG generates the oscillator output clock (CGMXCLK), which is used by the computer operating properly (COP), low-voltage inhibit (LVI), and other modules. The ICG also generates the clock generator output (CGMOUT), which is fed to the system integration module (SIM) to create the bus clocks. The bus frequency will be one-fourth the frequency of CGMXCLK and one-half the frequency of CGMOUT. Finally, the ICG generates the timebase clock (TBMCLK), which is used in the timebase module (TBM).

8.2 Features

The ICG has these features:

- Selectable external clock generator, either 1-pin external source or 2-pin crystal, multiplexed with port pins
- Internal clock generator with programmable frequency output in integer multiples of a nominal frequency (307.2 kHz ± 25 percent)
- Internal oscillator trimmed accuracy of ±3.5 percent
- Bus clock software selectable from either internal or external clock (bus frequency range from 76.8 kHz ± 25 percent to 9.75 MHz ± 25 percent in 76.8-kHz increments)

  NOTE

  For the MC68HC908EY16A, do not exceed the maximum bus frequency of 8 MHz at 5.0 V.

- Timebase clock automatically selected from external clock if external clock is available
- Clock monitor for both internal and external clocks

8.3 Functional Description

The ICG, shown in Figure 8-2, contains these major submodules:

- Clock enable circuit
- Internal clock generator
- External clock generator
- Clock monitor circuit
- Clock selection circuit

12.3 Port B

Port B is an 8-bit special-function port that shares all of its pins with the analog-to-digital converter (ADC) and some pin functions with TIMB.

Port B is designed so that the ADC function will take priority over the timer functionality on PTB6 and PTB7. If the ADC is selected for a conversion on a previously enabled timer pin, the port pin will be connected to the ADC and disconnected from the timer. If both the timer input capture and ADC functions are being used on the same port pin, it is recommended that the timer channel be disabled before the pin is enabled as an ADC input to avoid glitches. If both the timer output compare (or PWM) and ADC functions are being used on the same port pin, it is recommended that the timer channel be disabled before the pin is enabled as an ADC input.

12.3.1 Port B Data Register

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: Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Reset:	Unaffected by reset							
Alternative Function:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Alternative Function:	TBCH1	TBCH0	SPSCK	MOSI	MISO			

Figure 12-4. Port B Data Register (PTB)

PTB[7:0] — 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.

12.3.2 Data Direction Register B

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

Address:	$0005							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Reset:	0	0	0	0	0	0	0	0

Figure 12-5. Data Direction Register B (DDRB)

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

These read/write bits control port B data direction. Reset clears DDRB[7:0], 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.

12.6 Port E

Port E is a 2-bit special function port that shares its pins with the enhanced serial communications interface module (ESCI).

12.6.1 Port E Data Register

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

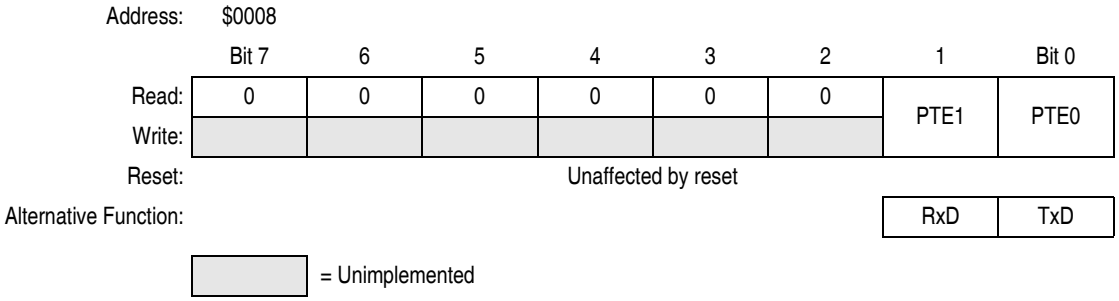


Figure 12-13. Port E Data Register (PTE)

PTE[1:0] — 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 PTE[1:0].

12.6.2 Data Direction Register E

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

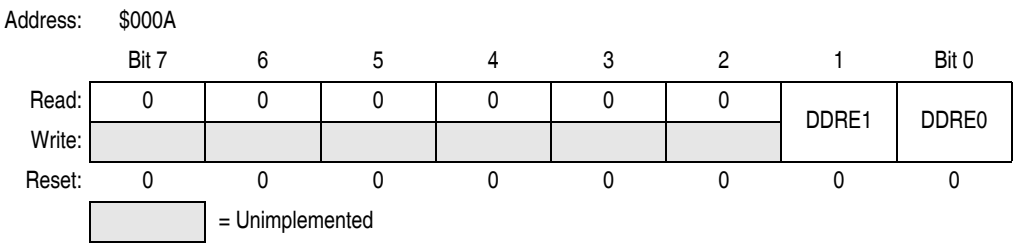


Figure 12-14. Data Direction Register E (DDRE)

DDRE[1:0] — Data Direction Register E Bits

These read/write bits control port E data direction. Reset clears DDRE[1:0], 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 12-15 shows the port E I/O logic.

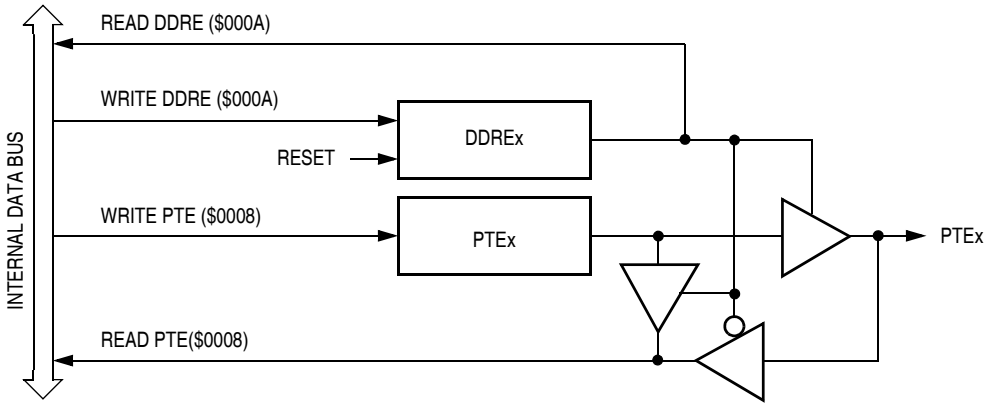


Figure 12-15. Port E I/O Circuit

When bit DDREx is a 1, reading address $0008 reads the PTEx data latch. When bit DDREx is a 0, reading address $0008 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 12-5 summarizes the operation of the port E pins.

Table 12-5. Port E Pin Functions

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

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

Receiving a break character has these effects on ESCI registers:

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

13.4.2.4 Idle Characters

For TXINV = 0 (output not inverted), a transmitted idle character contains all 1s and has no start, stop, or parity bit. Idle character length depends on the M bit in SCC1. The preamble is a synchronizing idle character that begins every transmission.

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

13.4.2.5 Inversion of Transmitted Output

The transmit inversion bit (TXINV) in ESCI control register 1 (SCC1) reverses the polarity of transmitted data. All transmitted values including idle, break, start, and stop bits, are inverted when TXINV is at 1. See 13.8.1 ESCI Control Register 1.

13.4.2.6 Transmitter Interrupts

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

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

13.4.3 Receiver

Figure 13-5 shows the structure of the ESCI receiver.

13.4.3.1 Character Length

The receiver can accommodate either 8-bit or 9-bit data. The state of the M bit in ESCI control register 1 (SCC1) determines character length. When receiving 9-bit data, bit R8 in ESCI control register 3 (SCC3) is the ninth bit (bit 8). When receiving 8-bit data, bit R8 is a copy of the eighth bit (bit 7).

13.8.3 ESCI Control Register 3

ESCI control register 3 (SCC3):

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

Address: $0012

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
Write:								
Reset:	U	0	0	0	0	0	0	0

= Unimplemented R = Reserved U = Unaffected

Figure 13-11. ESCI Control Register 3 (SCC3)

R8 — Received Bit 8

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

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

T8 — Transmitted Bit 8

When the ESCI is transmitting 9-bit characters, T8 is the read/write ninth bit (bit 8) of the transmitted character. T8 is loaded into the transmit shift register at the same time that the SCDR is loaded into the transmit shift register. Reset clears the T8 bit.

ORIE — Receiver Overrun Interrupt Enable Bit

This read/write bit enables ESCI error CPU interrupt requests generated by the receiver overrun bit, OR. Reset clears ORIE.

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

NEIE — Receiver Noise Error Interrupt Enable Bit

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

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

FEIE — Receiver Framing Error Interrupt Enable Bit

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

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

PEIE — Receiver Parity Error Interrupt Enable Bit

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

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

Chapter 15
Serial Peripheral Interface (SPI) Module

15.1 Introduction

This section describes the serial peripheral interface (SPI) module, which allows full-duplex, synchronous, serial communications with peripheral devices.

15.2 Features

Features of the SPI module include:

- Full-duplex operation
- Master and slave modes
- Double-buffered operation with separate transmit and receive registers
- Four master mode frequencies (maximum = bus frequency ÷ 2)
- Maximum slave mode frequency = bus frequency
- Serial clock with programmable polarity and phase
- Two separately enabled interrupts with CPU service:
  - SPRF (SPI receiver full)
  - SPTE (SPI transmitter empty)
- Mode fault error flag with CPU interrupt capability
- Overflow error flag with CPU interrupt capability
- Programmable wired-OR mode
- I^2C (inter-integrated circuit) compatibility

15.3 Pin Name and Register Name Conventions

The generic names of the SPI input/output (I/O) pins are:

- $\overline{SS}$ (slave select)
- SPSCK (SPI serial clock)
- MOSI (master out slave in)
- MISO (master in slave out)

The SPI shares four I/O pins with a parallel I/O port. The full name of an SPI pin reflects the name of the shared port pin. Table 15-1 shows the full names of the SPI I/O pins. The generic pin names appear in the text that follows.

Table 15-1. Pin Name Conventions

SPI Generic Pin Name	MISO	MOSI	$\overline{SS}$	SPSCK
Full SPI Pin Name	PTC0/MISO	PTC1/MOSI	PTA6/$\overline{SS}$	PTA5/SPSCK
Alternative Pin	PTB3	PTB4	PTC2	PTB5

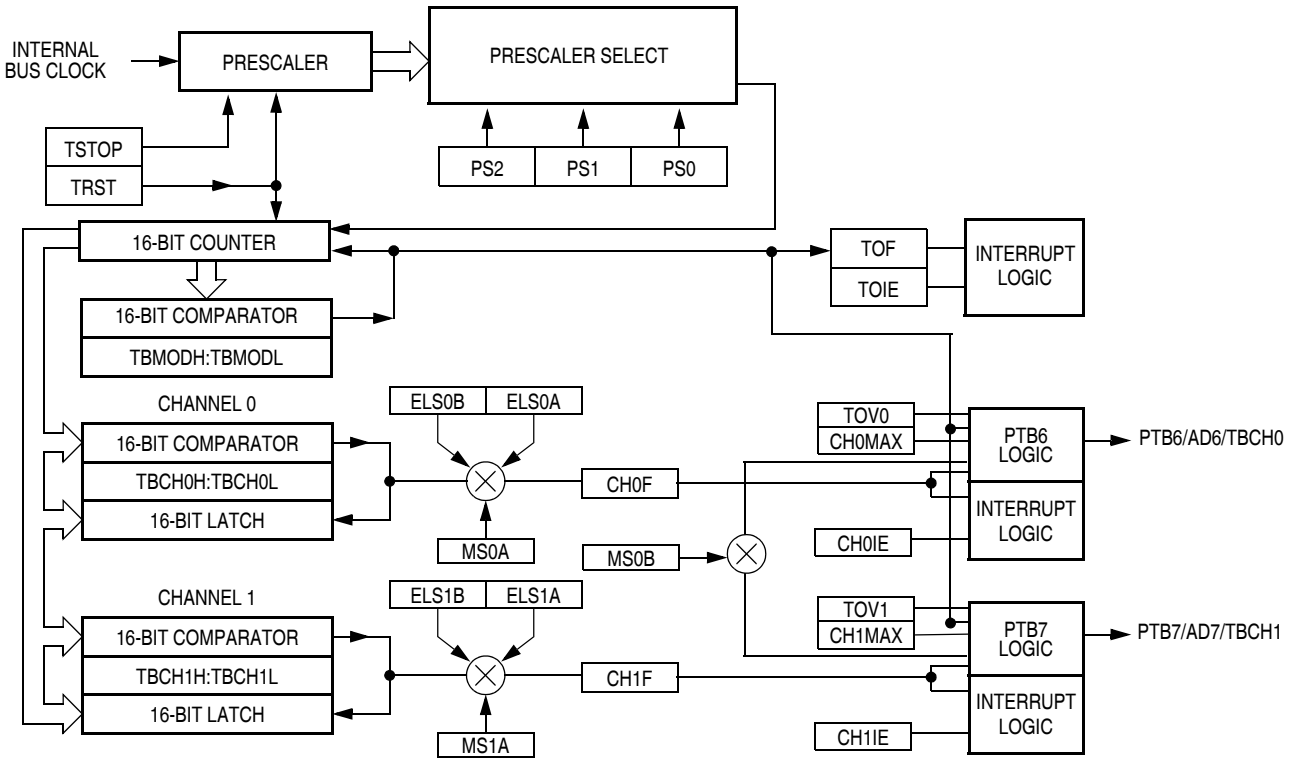


Figure 18-2. TIMB Block Diagram

18.3.2 Input Capture

An input capture function has three basic parts: edge select logic, an input capture latch, and a 16-bit counter. Two 8-bit registers, which make up the 16-bit input capture register, are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The polarity of the active edge is programmable. The level transition which triggers the counter transfer is defined by the corresponding input edge bits (ELSxB and ELSxA in TBSC0 through TBSC1 control registers with x referring to the active channel number). When an active edge occurs on the pin of an input capture channel, the TIMB latches the contents of the TIMB counter into the TIMB channel registers, TBCHxH–TBCHxL. Input captures can generate TIMB CPU interrupt requests. Software can determine that an input capture event has occurred by enabling input capture interrupts or by polling the status flag bit.

The free-running counter contents are transferred to the TIMB channel status and control register (TBCHxH–TBCHxL, see 18.8.5 TIMB Channel Registers) on each proper signal transition regardless of whether the TIMB channel flag (CH0F–CH1F in TBSC0–TBSC1 registers) is set or clear. When the status flag is set, a CPU interrupt is generated if enabled. The value of the count latched or “captured” is the time of the event. Because this value is stored in the input capture register 2 bus cycles after the actual event occurs, user software can respond to this event at a later time and determine the actual time of the event. However, this must be done prior to another input capture on the same pin; otherwise, the previous time value will be lost.

Chapter 19
Development Support

19.1 Introduction

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

19.2 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 of the break module 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

19.2.1 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 system integration module (SIM). The SIM then causes the CPU to load the instruction register with a software interrupt instruction (SWI). 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 1 to the BRKA bit in the break status and control register.

When a CPU generated address matches the contents of the break address registers, the break interrupt is generated. A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the microcontroller unit (MCU) to normal operation.

Figure 19-2 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)

19.3 Monitor Module (MON)

The monitor module allows debugging and programming 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 FLASH
- FLASH memory security feature(1)
- FLASH memory programming interface
- Use of external 9.8304 MHz oscillator or ICG to generate internal operating frequency of 2.4576 MHz
- Monitor mode entry without high voltage, V_{TST}, if reset vector is blank ($FFFE and $FFFF contain $FF)
- Normal monitor mode entry if V_{TST} is applied to $\overline{IRQ}$

19.3.1 Functional Description

Figure 19-8 shows a simplified diagram of the monitor mode.

The monitor module receives and executes commands from a host computer. Figure 19-9, Figure 19-10, and Figure 19-11 show example circuits used to enter monitor mode and communicate with a host computer via a standard RS-232 interface.

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

Table 19-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a power-on reset (POR) and will allow communication at 9600 baud provided one of the following sets of conditions is met:

- If $FFFE and $FFFF are erased or programmed:
  - The external clock is 9.8304 MHz (9600 baud)
  - $\overline{IRQ} = V_{TST}$
- If $FFFE and $FFFF contain $FF (erased state):
  - The external clock is 9.8304 MHz (9600 baud)
  - $\overline{IRQ} = V_{DD}$ (this can be implemented through the internal $\overline{IRQ}$ pullup)
- If $FFFE and $FFFF contain $FF (erased state):
  - The ICG clock is nominal 2.45 MHz (nominal 9600 baud)
  - $\overline{IRQ} = V_{SS}$

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

Table 19-1. Monitor Mode Signal Requirements and Options

Mode	$\overline{IRQ}$	$\overline{RST}$	Reset Vector	Serial Communication		Mode Selection		ICG	COP	Communication Speed		
				PTA0	PTA1	PTB4	PTB3			External Clock	f_{OP}	Baud Rate
Normal Monitor	V_{TST}	V_{DD} or V_{TST}	X	1	0	1	0	OFF	Disabled	9.8304 MHz	2.4576 MHz	9600
Forced Monitor	V_{DD}	V_{DD}	$FFFF (blank)	1	0	X	X	OFF	Disabled	9.8304 MHz	2.4576 MHz	9600
	V_{SS}	V_{DD}		1	0	X	X	ON	Disabled	—	Nominal 2.45 MHz	Nominal 9600
User	V_{DD} or V_{SS}	V_{DD} or V_{TST}	Not $FFFF	X	X	X	X	ON	Enabled	—	Nominal 1.6 MHz	X
MON08 Function [Pin No.]	V_{TST} [6]	$\overline{RST}$ [4]	—	COM [8]	SSEL [10]	MOD0 [12]	MOD1 [14]	—	—	OSC1 [13]	—	—

1. PTA0 must have a pullup resistor to V_{DD} in monitor mode.
2. Communication speed in the table is an example to obtain a baud rate of 9600 except the forced monitor $\overline{IRQ} = V_{SS}$ case. Baud rate using external oscillator is bus frequency / 256.
4. X = don't care
5. $\overline{RST}$ column indicates the state of $\overline{RST}$ after the monitor entry.
6. MON08 pin refers to P&E Microcomputer Systems' MON08-Cyclone 2 by 8-pin connector.

NC	1	2	GND
NC	3	4	$\overline{RST}$
NC	5	6	$\overline{IRQ}$
NC	7	8	PTA0
NC	9	10	PTA1
NC	11	12	PTB3
OSC1	13	14	PTB4
V_{DD}	15	16	NC

19.3.1.2 Forced Monitor Mode

If entering monitor mode without high voltage on $\overline{IRQ}$, then PTA1, PTB3, and PTB4 pin requirements and conditions are not in effect. This is to reduce circuit requirements when performing in-circuit programming.

If the reset vector is blank and monitor mode is entered without V_{TST} on $\overline{IRQ}$, the MCU will see an additional reset cycle after the initial power-on reset (POR). The MCU will initially come out of reset in user mode. Internal circuitry monitors the reset vector fetches and will assert an internal reset if it detects the reset vector is erased ($FFFF).

Once the MCU enters this mode any reset other than a POR will automatically force the MCU to come back to the forced monitor mode. Exiting the forced monitor mode requires a POR. Pulling $\overline{RST}$ low will

A brief description of each monitor mode command is given in Table 19-3 through Table 19-8.

Table 19-3. READ (Read Memory) Command

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

Command Sequence

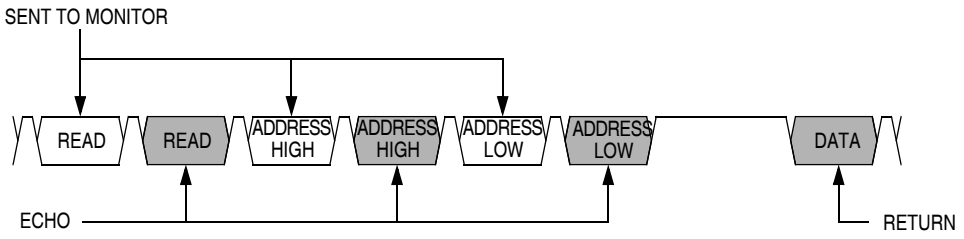


Table 19-4. WRITE (Write Memory) Command

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

Command Sequence

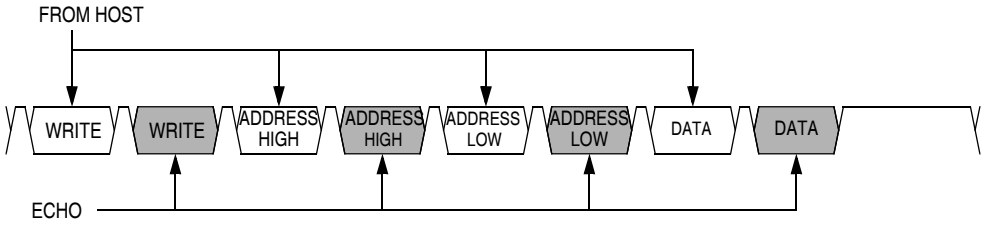
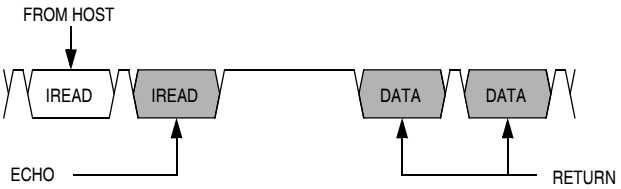


Table 19-5. IREAD (Indexed Read) Command

Description	Read next 2 bytes in memory from last address accessed
Operand	None
Data Returned	Returns contents of next two addresses
Opcode	$1A

Command Sequence



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

NOTE

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

To determine whether the security code entered is correct, check to see if bit 6 of RAM address $40 is set. If it is, then the correct security code has been entered and FLASH can be accessed.

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

19.3.3 Extended Security

In addition to the above security, a more secure feature called extended security is implemented in the MCU to further protect FLASH contents. Once this extended security is enabled, the MCU does not allow any user to enter the monitor mode even when all 8 security bytes are matched correctly. The extended security feature can be enabled by programming address $FDFF located in the user FLASH memory with data $00.

To unlock the extended security feature, the MCU must enter the monitor mode by failing the 8 byte security check. Then the FLASH must be mass-erased. This unlock process will erase the FLASH contents completely.

NOTE

To avoid enabling the extended security unintentionally, the user must make sure that the user software does not contain data $00 at address $FDFF.

19.4 Routines Supported in ROM

In the ROM, five routines are supported. Because the ROM has a jump table, the user does not call the routines with direct addresses. Therefore, the calling addresses will not change—even when the ROM code is updated in the future.

This section introduces each routine briefly. Details are discussed in later sections.

- **GetByte** — This routine is used to receive a byte serially on the general-purpose I/O PTA0. The receiving baud rate is the same as the baud rate used in monitor mode. In the GetByte routine, the GetBit routine is called to generate baud rates.
- **PutByte** — This routine is used to send a byte serially on the general-purpose I/O PTA0. The sending baud rate is the same as the baud rate specified in monitor mode.
- **Verify** — This routine is used to perform one of two options. Using the send-out option, this routine reads FLASH locations and sends the data out serially on the general-purpose I/O PTA0. Using the compare option, this routine compares the FLASH data against data in a specific RAM location, which is referred to as a DATA array. The DATA array locations and the variable locations required for this routine are at fixed memory addresses.

Table 19-11. Verify Routine Options

Option	Description
Send-out option	Used to read a range of FLASH locations and to send the read data to a host through PTA0 by using the PutByte routine.
Compare option	Used to read a range of FLASH locations and to compare the read data against the DATA array

Verify Routine Options

- Send-Out Option — If the accumulator (A) is initialized with $00 at the routine entry, the read data will be sent out serially through PTA0. The communication baud rate is the same as the baud rate described in the PutByte routine. When this option is selected, the PTA0 must be pulled up and configured as an input and the PTA0 data bit must be initialized to 0.
- Compare Option — If A is initialized with a non-zero value, the read data is compared against the DATA array for each byte of FLASH and the DATA array is replaced by the data read from FLASH. If the data does not match the corresponding value, the data read from FLASH can be confirmed in the DATA array. All data in the DATA array must be in the zero page, but a range can be beyond a row size or a page size.

Carry (C) Bit and Checksum

The first and last addresses of the range to be read and/or compared are specified as parameters in registers H:X and LADDR, respectively. In the compare option, the carry (C) bit of the condition code register (CCR) is set if the data in the specified range is verified successfully against the data in the DATA array. However when the send-out option is selected, the status of the C bit is meaningless because this function does not include the compare operation. Both options calculate a checksum on data read in the range. This checksum, which is the LSB of the sum of all bytes in the entire data collection, is stored in A upon return from the function.

Interrupts are not masked. The COP is serviced in Verify. The first COP is serviced at 24 bus cycles after this routine is called in the user software. However, the COP timeout might still occur in the send-out option if the COP is configured for a short timeout period.

Entry Condition

H:X — Contains the beginning address in a range.

LADDR — Contains the last address in a range.

A — When A contains $00, read data is sent out via PTA0 (send-out option is selected). When A contains a non-zero value, read data is verified against the DATA array (compare option is selected).

DATA array — Contains data to be verified against FLASH data. For the send-out option, the DATA array is not used.

PTA0 — When the send-out option is selected, this pin must be configured as an input and pulled up in hardware and PTA0 must be initialized to 0.

Exit Condition

A — Contains a checksum value.

Package drawing 98ASH70029A page 1 of 3