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	I²C, SCI
Peripherals	LED, LVD, POR, PWM
Number of I/O	26
Program Memory Size	16KB (16K 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 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	32-SDIP (0.400", 10.16mm)
Supplier Device Package	32-SDIP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908jl16cspe

Chapter 14
Low-Voltage Inhibit (LVI)

Chapter 15
Central Processor Unit (CPU)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA)	Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PTB)	Read: Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Reset:	Unaffected by reset							
$0002	Unimplemented	Read: Write:								
$0003	Port D Data Register (PTD)	Read: Write:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Reset:	Unaffected by reset							
$0004	Data Direction Register A (DDRA)	Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB)	Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Reset:	0	0	0	0	0	0	0	0
$0006	Unimplemented									
$0007	Data Direction Register D (DDRD)	Read: Write:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Reset:	0	0	0	0	0	0	0	0
$0008	Port E Data Register (PTE)	Read: Write:							PTE1	PTE0
		Reset:	Unaffected by reset							
$0009	Unimplemented									
$000A	Port D Control Register (PDCR)	Read: Write:	0	0	0	0	SLOWD7	SLOWD6	PTDPU7	PTDPU6
		Reset:	0	0	0	0	0	0	0	0
$000B	Unimplemented									
$000C	Data Direction Register E (DDRE)	Read: Write:							DDRE1	DDRE0
		Reset:	0	0	0	0	0	0	0	0

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

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

Chapter 3
Configuration and Mask Option Registers (CONFIG and MOR)

3.1 Introduction

This section describes the configuration registers, CONFIG1 and CONFIG2; and the mask option register (MOR).

The configuration registers enable or disable these options:

- Computer operating properly module (COP)
- COP timeout period ($2^{13}–2^4$ or $2^{18}–2^4$ ICLK cycles)
- Internal oscillator during stop mode
- Low voltage inhibit (LVI) module
- LVI module voltage trip point selection
- STOP instruction
- Stop mode recovery time (32 or 4096 ICLK cycles)
- Pull-up on $\overline{\text{IRQ}}$ pin
- MMIIC pin selection

The mask option register selects the oscillator option:

- Crystal or RC

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001E	Configuration Register 2 (CONFIG2)(1)	Read: Write:	IRQPUD	R	R	LVIT1	LVIT0	R	IICSEL	STOP_ ICLKDIS
		Reset:	0	0	0	0(2)	0(2)	0	0	0
$001F	Configuration Register 1 (CONFIG1)(1)	Read: Write:	COPRS	R	R	LVID	R	SSREC	STOP	COPD
		Reset:	0	0	0	0	0	0	0	0
$FFD0	Mask Option Register (MOR)(3)	Read: Write:	OSCSEL	R	R	R	R	R	R	R
		Reset:	Unaffected by reset; $FF when blank							

1. One-time writable register after each reset.
2. LVIT1 and LVIT0 reset to 0 by a power-on reset (POR) only.
3. Non-volatile FLASH register; write by programming.

R = Reserved

Figure 3-1. CONFIG Registers Summary

Address:	$FFD0							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	OSCSEL	R	R	R	R	R	R	R
Erased:	1	1	1	1	1	1	1	1
Reset:	Unaffected by reset							

Non-volatile FLASH register; write by programming.

R = Reserved

Figure 5-1. Mask Option Register (MOR)

OSCSEL — Oscillator Select Bit

OSCSEL selects the oscillator type for the MCU. The erased or unprogrammed state of this bit is logic 1, selecting the crystal oscillator option. This bit is unaffected by reset.

1 = Crystal oscillator
0 = RC oscillator

Bits 6–0 — Should be left as logic 1's.

NOTE

When Crystal oscillator is selected, the OSC2/RCCLK/PTA6/KBI6 pin is used as OSC2; other functions such as PTA6/KBI6 will not be available.

5.2.1 XTAL Oscillator

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

In its typical configuration, the XTAL oscillator is connected in a Pierce oscillator configuration, as shown in Figure 5-2. 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
- Fixed capacitor, C_1
- Tuning capacitor, C_2 (can also be a fixed capacitor)
- Feedback resistor, R_B
- Series resistor, R_S (optional)

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.

5.2.2 RC Oscillator

The RC oscillator circuit is designed for use with external resistor and capacitor to provide a clock source with tolerance less than 10%. See Figure 5-3.

In its typical configuration, the RC oscillator requires two external components, one R and one C. Component values should have a tolerance of 1% or less, to obtain a clock source with less than 10% tolerance. The oscillator configuration uses two components:

- C_{EXT}
- R_{EXT}

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
  - 7-frequency internal bus clock prescaler selection
  - External clock input on timer 2 (bus frequency ÷2 maximum)
- 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 external clock input for TIM2 is shared with the an ADC channel pin. 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	T2CLK
Full TIM Pin Names:	TIM1	PTD4/T1CH0	PTD5/T1CH1	—
	TIM2	PTE0/T2CH0	PTE1/T2CH1	ADC12/T2CLK

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.

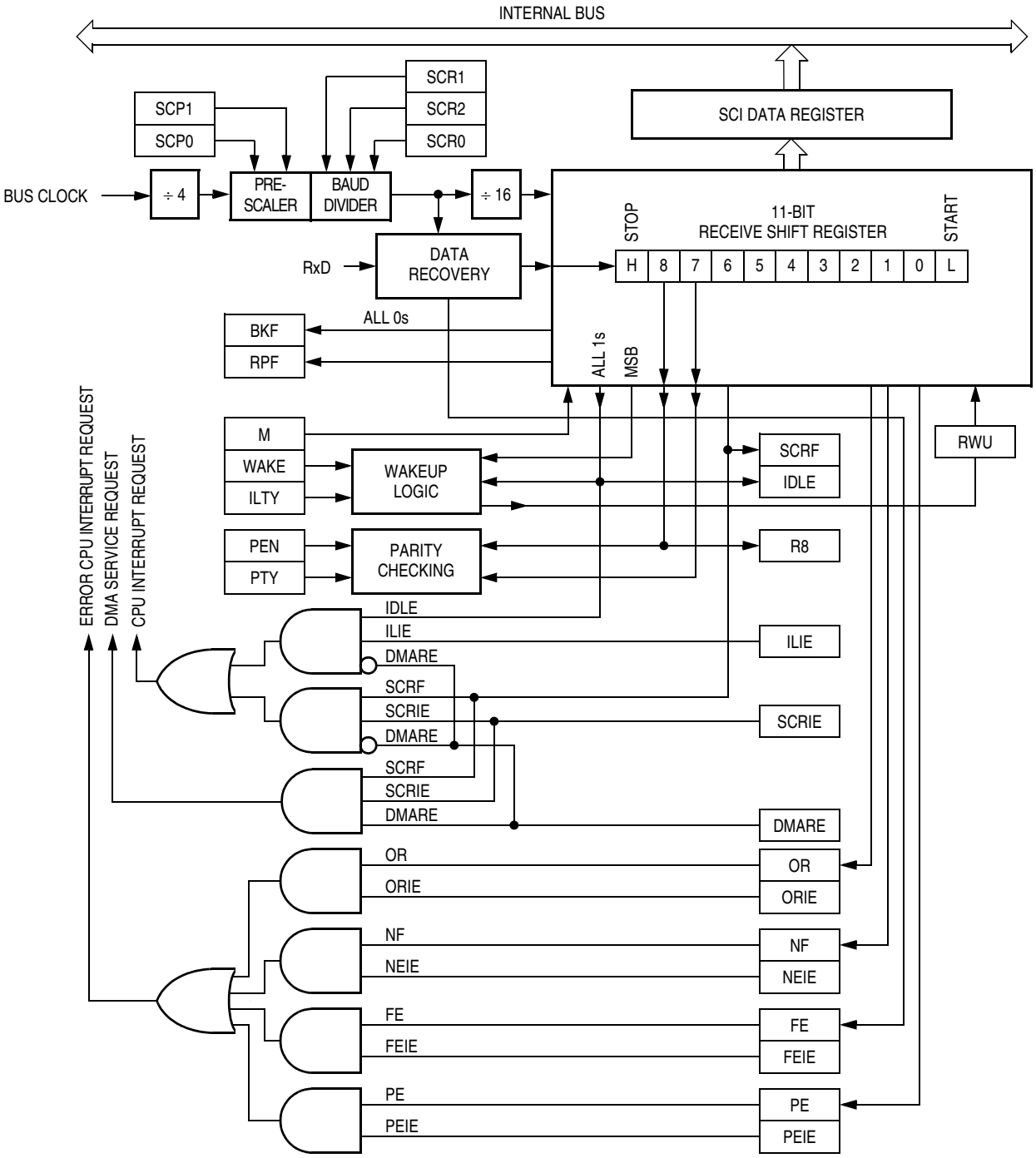

Figure 7-5. SCI Receiver Block Diagram

Fast Data Tolerance

Figure 7-8 shows how much a fast received character can be misaligned without causing a noise error or a framing error. The fast stop bit ends at RT10 instead of RT16 but is still there for the stop bit data samples at RT8, RT9, and RT10.

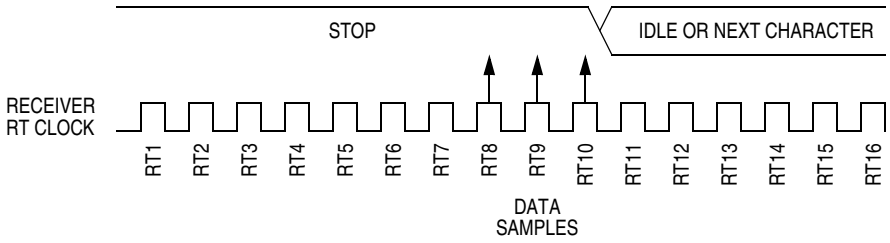

Figure 7-8. Fast Data

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

With the misaligned character shown in Figure 7-8, the receiver counts 154 RT cycles at the point when the count of the transmitting device is
10 bit times × 16 RT cycles = 160 RT cycles.

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

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

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

With the misaligned character shown in Figure 7-8, the receiver counts 170 RT cycles at the point when the count of the transmitting device is
11 bit times × 16 RT cycles = 176 RT cycles.

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

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

7.4.3.6 Receiver Wakeup

So that the MCU can ignore transmissions intended only for other receivers in multiple-receiver systems, the receiver can be put into a standby state. Setting the receiver wakeup bit, RWU, in SCC2 puts the receiver into a standby state during which receiver interrupts are disabled.

Depending on the state of the WAKE bit in SCC1, either of two conditions on the RxD pin can bring the receiver out of the standby state:

M — Mode (Character Length) Bit

This read/write bit determines whether SCI characters are eight or nine bits long. (See Table 7-5.) The ninth bit can serve as an extra stop bit, as a receiver wakeup signal, or as a parity bit. Reset clears the M bit.

1 = 9-bit SCI characters
0 = 8-bit SCI characters

WAKE — Wakeup Condition Bit

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

1 = Address mark wakeup
0 = Idle line wakeup

ILTY — Idle Line Type Bit

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

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

PEN — Parity Enable Bit

This read/write bit enables the SCI parity function. (See Table 7-5.) When enabled, the parity function inserts a parity bit in the most significant bit position. (See Figure 7-3.) Reset clears the PEN bit.

1 = Parity function enabled
0 = Parity function disabled

PTY — Parity Bit

This read/write bit determines whether the SCI generates and checks for odd parity or even parity. (See Table 7-5.) Reset clears the PTY bit.

1 = Odd parity
0 = Even parity

NOTE

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

Table 7-5. Character Format Selection

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

7.8.7 SCI Baud Rate Register

The baud rate register (SCBR) selects the baud rate for both the receiver and the transmitter.

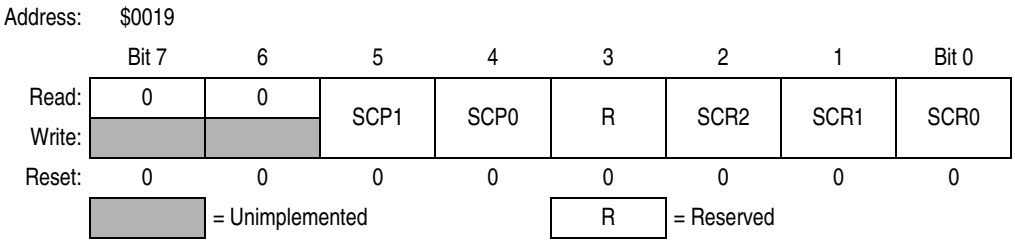

Figure 7-16. SCI Baud Rate Register (SCBR)

SCP1 and SCP0 — SCI Baud Rate Prescaler Bits

These read/write bits select the baud rate prescaler divisor as shown in Table 7-6. Reset clears SCP1 and SCP0.

Table 7-6. SCI Baud Rate Prescaling

SCP1 and SCP0	Prescaler Divisor (PD)
00	1
01	3
10	4
11	13

SCR2–SCR0 — SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate divisor as shown in Table 7-7. Reset clears SCR2–SCR0.

Table 7-7. SCI Baud Rate Selection

SCR2, SCR1, and SCR0	Baud Rate Divisor (BD)
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

Use this formula to calculate the SCI baud rate:

$$\text{baud rate} = \frac{\text{SCI clock source}}{64 \times PD \times BD}$$

where:

SCI clock source = bus clock
PD = prescaler divisor
BD = baud rate divisor

Table 7-8 shows the SCI baud rates that can be generated with a 4.9152 MHz bus clock.

8.3 I/O Pins

The MMIIC module uses two I/O pins, shared with standard port I/O pins. The full name of the MMIIC I/O pins are listed in Table 8-1. The generic pin name appear in the text that follows.

Table 8-1. Pin Name Conventions

MMIIC Generic Pin Names:	Full MCU Pin Names:
SDA	PTA2/KBI2/SDA(1)
	PTD7/RxD/SDA
SCL	PTA3/KBI3/SCL(1)
	PTD6/TxD/SCL

1. Position of MMIIC module pins is user selectable using CONFIG2 option bit. Refer to Chapter 3 Configuration and Mask Option Registers (CONFIG and MOR) for additional information.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0040	Multi-Master IIC Master Control Register (MIMCR)	Read:	MMALIF	MMNAKIF	MMBB	MMAST	MMRW	MMBR2	MMBR1	MMBR0
		Write:	0	0						
		Reset:	0	0	0	0	0	0	0	0
$0041	Multi-Master IIC Address Register (MMADR)	Read:	MMAD7	MMAD6	MMAD5	MMAD4	MMAD3	MMAD2	MMAD1	MMEXTAD
		Write:								
		Reset:	1	0	1	0	0	0	0	0
$0042	Multi-Master IIC Control Register (MMCR)	Read:	MMEN	MMIEN	0	0	MMTXAK	REPSEN	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0043	Multi-Master IIC Status Register (MMSR)	Read:	MMRXIF	MMTXIF	MMATCH	MMSRW	MMRXAK	0	MMTXBE	MMRXBF
		Write:	0	0						
		Reset:	0	0	0	0	1	0	1	0
$0044	Multi-Master IIC Data Transmit Register (MMDTR)	Read:	MMTD7	MMTD6	MMTD5	MMTD4	MMTD3	MMTD2	MMTD1	MMTD0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0045	Multi-Master IIC Data Receive Register (MMDRR)	Read:	MMRD7	MMRD6	MMRD5	MMRD4	MMRD3	MMRD2	MMRD1	MMRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 8-1. MMIIC I/O Register Summary

8.8.3 Multi-Master IIC Master Control Register (MIMCR)

Address: $0040

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MMALIF	MMNAKIF	MMBB	MMAST	MMRW	MMBR2	MMBR1	MMBR0
Write:	0	0						
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 8-6. Multi-Master IIC Master Control Register (MIMCR)

MMALIF — Multi-Master Arbitration Lost Interrupt Flag

This flag is set when software attempt to set MMAST but the MMBB has been set by detecting the start condition on the lines or when the MMIIC is transmitting a "1" to SDA line but detected a "0" from SDA line in master mode – an arbitration loss. This bit generates an interrupt request to the CPU if the MMIEN bit in MMCR is also set. This bit is cleared by writing "0" to it or by reset.

1 = Lost arbitration in master mode
0 = No arbitration lost

MMNAKIF — No Acknowledge Interrupt Flag

This flag is only set in master mode (MMAST = 1) when there is no acknowledge bit detected after one data byte or calling address is transferred. This flag also clears MMAST. MMNAKIF generates an interrupt request to CPU if the MMIEN bit in MMCR is also set. This bit is cleared by writing "0" to it or by reset.

1 = No acknowledge bit detected
0 = Acknowledge bit detected

MMBB — Bus Busy Flag

This flag is set after a start condition is detected (bus busy), and is cleared when a stop condition (bus idle) is detected. Reset clears this bit.

1 = Start condition detected
0 = Stop condition detected or MMIIC is disabled

MMAST — Master Control Bit

This bit is set to initiate a master mode transfer. In master mode, the module generates a start condition to the SDA and SCL lines, followed by sending the calling address stored in MMADR. When the MMAST bit is cleared by MMNAKIF set (no acknowledge) or by software, the module generates the stop condition to the lines after the current byte is transmitted. If an arbitration loss occurs (MMALIF = 1), the module reverts to slave mode by clearing MMAST, and releasing SDA and SCL lines immediately. This bit is cleared by writing “0” to it or by reset.

1 = Master mode operation
0 = Slave mode operation

MMRW — Master Read/Write

This bit will be transmitted out as bit 0 of the calling address when the module sets the MMAST bit to enter master mode. The MMRW bit determines the transfer direction of the data bytes that follows. When it is "1", the module is in master receive mode. When it is "0", the module is in master transmit mode. Reset clears this bit.

1 = Master mode receive
0 = Master mode transmit

COCO — Conversion Complete Bit

The COCO bit is a read-only bit which is set each time a conversion is completed. This bit is cleared whenever the status and control register is written or whenever the data register (low) is read.

1 = Conversion completed
0 = Conversion not completed

AIEN — ADC10 Interrupt Enable Bit

When this bit is set, an interrupt is generated at the end of a conversion. The interrupt signal is cleared when the data register is read or the status/control register is written. Reset clears the AIEN bit.

1 = ADC10 interrupt enabled
0 = ADC10 interrupt disabled

ADCO — ADC10 Continuous Conversion Bit

When written high, the ADC10 will begin to convert samples continuously (continuous conversion mode) and update the result registers at the end of each conversion, provided the ADCH[4:0] bits do not decode to all 1s. The ADC10 will continue to convert until the MCU enters reset, the MCU enters stop mode (if ACLKEN is clear), the ADCLK register is written, or until the ADCSC is written again. If Stop is entered (with ACLKEN low), continuous conversions will cease and can only be restarted with a write to the ADCSC. Any write to the ADCSC with the ADCO bit set and the ADCH bits not all 1s will abort the current conversion and begin continuous conversions.

If the bus frequency is less than the ADCK frequency, precise sample time for continuous conversions cannot be guaranteed in short-sample mode (ADLSMP = 0). If the bus frequency is less than 1/11th of the ADCK frequency, precise sample time for continuous conversions cannot be guaranteed in long-sample mode (ADLSMP = 1).

When clear, the ADC10 will perform a single conversion (single conversion mode) each time the ADCSC is written (assuming the ADCH[4:0] bits do not decode all 1s). Reset clears the ADCO bit.

1 = Continuous conversion following a write to the ADCSC
0 = One conversion following a write to the ADCSC

ADCH[4:0] — Channel Select Bits

ADCH4, ADCH3, ADCH2, ADCH1, and ADCH0 form a 5-bit field which is used to select one of the input channels. The input channels are detailed in Table 9-2.

The successive approximation converter subsystem is turned off when the channel select bits are all set to 1. This feature allows for explicit disabling of the ADC10 and isolation of the input channel from the I/O pad. Terminating continuous convert mode this way will prevent an additional, single conversion from being performed. It is not necessary to set the channel select bits to all 1s to place the ADC10 in a low-power state, however, because the module is automatically placed in a low-power state when a conversion completes.

Table 10-1. Port Control Register Bits Summary

Port	Bit	DDR	Module Control			Pin
			Module	Register	Control Bit	
A	0	DDRA0	KBI	KBIER ($001B)	KBIE0	PTA0/KBI0
	1	DDRA1	KBI	KBIER ($001B)	KBIE1	PTA1/KBI1
	2	DDRA2	KBI	KBIER ($001B)	KBIE2	PTA2/KBI2
			MMIIC	MMCR	MMEN	PTA2/KBI2/SDA(1)(2)
	3	DDRA3	KBI	KBIER ($001B)	KBIE3	PTA3/KBI3
			MMIIC	MMCR	MMEN	PTA3/KBI3/SCL(1)(2)
	4	DDRA4	KBI	KBIER ($001B)	KBIE4	PTA4/KBI4
	5	DDRA5	KBI	KBIER ($001B)	KBIE5	PTA5/KBI5
	6	DDRA6	OSC KBI	PTAPUE ($000D) KBIER ($001B)	PTA6EN KBIE6	RCCLK/PTA6/KBI6(3)
	7	DDRA7	KBI	KBIER ($001B)	KBIE7	PTA7/KBI7
B	0	DDRB0	ADC	ADSCR ($003C)	ADCH[4:0]	PTB0/ADC0
	1	DDRB1	ADC	ADSCR ($003C)	ADCH[4:0]	PTB1/ADC1
	2	DDRB2	ADC	ADSCR ($003C)	ADCH[4:0]	PTB2/ADC2
	3	DDRB3	ADC	ADSCR ($003C)	ADCH[4:0]	PTB3/ADC3
	4	DDRB4	ADC	ADSCR ($003C)	ADCH[4:0]	PTB4/ADC4
	5	DDRB5	ADC	ADSCR ($003C)	ADCH[4:0]	PTB5/ADC5
	6	DDRB6	ADC	ADSCR ($003C)	ADCH[4:0]	PTB6/ADC6
	7	DDRB7	ADC	ADSCR ($003C)	ADCH[4:0]	PTB7/ADC7
D	0	DDRD0	ADC	ADSCR ($003C)	ADCH[4:0]	PTD0/ADC11
	1	DDRD1	ADC	ADSCR ($003C)	ADCH[4:0]	PTD1/ADC10
	2	DDRD2	ADC	ADSCR ($003C)	ADCH[4:0]	PTD2/ADC9
	3	DDRD3	ADC	ADSCR ($003C)	ADCH[4:0]	PTD3/ADC8
	4	DDRD4	TIM1	T1SC0 ($0025)	ELS/0B:ELS0A	PTD4/T1CH0
	5	DDRD5	TIM1	T1SC1 ($0028)	ELS1B:ELS1A	PTD5/T1CH1
	6	DDRD6	SCI	SCC1 ($0013)	ENSCI	PTD6/TxD
			MMIIC	MMCR	MMEN	PTD6/TxD/SCL(1)(4)
	7	DDRD7	SCI	SCI	ENSCI	PTD7/RxD
			MMIIC	MMCR	MMEN	PTD7/RxD/SDA(1)(4)
E	0	DDRE0	TIM2	T2SC0 ($0035)	ELS0B:ELS0A	PTE0/T2CH0
	1	DDRE1	TIM2	T2SC1 ($0038)	ELS1B:ELS1A	PTE1/T2CH1

1. Position of MMIIC module pins is user selectable using CONFIG2 option bit.
2. If MMIIC module is using the PTA2/PTA3 pairs for IIC (CONFIG2 – IICSEL = 1, MMEN = 1), the MMIIC module will have priority over the KBI module.
3. RCCLK/PTA6/KBI6 pin is only available when OSCSEL=0 (RC option);
   PTAPUE register has priority control over the port pin.
   RCCLK/PTA6/KBI6 is the OSC2 pin when OSCSEL=1 (XTAL option).
4. If ESCI module is enabled (ENSCI = 1), the ESCI will have priority over the PTD6/PTD7 pins regardless of the state of the MMIIC module.

Table 15-1. Instruction Set Summary (Sheet 2 of 6)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
BHS *rel*	Branch if Higher or Same (Same as BCC)	PC ← (PC) + 2 + *rel* ? (C) = 0	–	–	–	–	–	–	REL	24	rr	3
BIH *rel*	Branch if $\overline{\text{IRQ}}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 1	–	–	–	–	–	–	REL	2F	rr	3
BIL *rel*	Branch if $\overline{\text{IRQ}}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 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

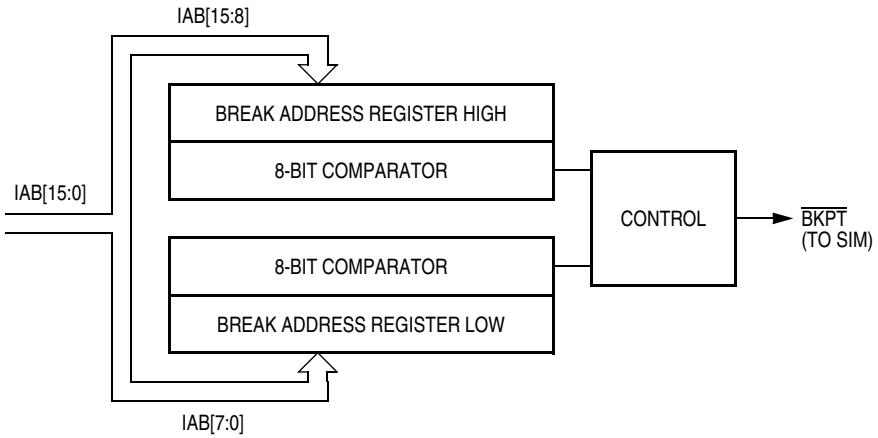

Figure 16-1. Break Module Block Diagram

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:							See note	
		Reset:							0	
$FE03	Break Flag Control Register (BFCR)	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE0C	Break Address High Register (BRKH)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0D	Break Address low Register (BRKL)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0E	Break Status and Control Register (BRKSCR)	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

Note: Writing a logic 0 clears SBSW.

= Unimplemented R = Reserved

Figure 16-2. Break I/O Register Summary

16.2.2 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 16.2.6.4 Break Flag Control Register (BFCR) and see the **Break Interrupts** subsection for each module.)

16.3 Monitor Module (MON)

This section describes the monitor ROM (MON) and the monitor mode entry methods. The monitor ROM allows complete testing of the MCU through a single-wire interface with a host computer. This mode is also used for programming and erasing of FLASH memory in the MCU. Monitor mode entry can be achieved without use of the higher test voltage, V_{TST}, as long as vector addresses $FFFE and $FFFF are blank, thus reducing the hardware requirements for in-circuit programming.

Features include:

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

16.3.1 Functional Description

The monitor ROM receives and executes commands from a host computer. Figure 16-8 shows a example circuit used to enter monitor mode and communicate with a host computer via a standard RS-232 interface.

Simple monitor commands can access any memory address. In monitor mode, the MCU can execute host-computer code in RAM while most MCU pins retain normal operating mode functions. All communication between the host computer and the MCU is through the PTB0 pin. A level-shifting and multiplexing interface is required between PTB0 and the host computer. PTB0 is used in a wired-OR configuration and requires a pull-up resistor.

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