

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	Not For New Designs
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	I ² C, IRSCI, SCI, SPI
Peripherals	LED, LVD, POR, PWM
Number of I/O	32
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K 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	44-QFP
Supplier Device Package	44-QFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908ap16cfber

12.5.3.8	Error Interrupts	194
12.6	Low-Power Modes	195
12.6.1	Wait Mode	195
12.6.2	Stop Mode	195
12.7	SCI During Break Module Interrupts	195
12.8	I/O Signals	196
12.8.1	PTC6/SCTxD (Transmit Data)	196
12.8.2	PTC7/SCRxD (Receive Data)	196
12.9	I/O Registers	197
12.9.1	IRSCI Control Register 1	197
12.9.2	IRSCI Control Register 2	199
12.9.3	IRSCI Control Register 3	200
12.9.4	IRSCI Status Register 1	202
12.9.5	IRSCI Status Register 2	204
12.9.6	IRSCI Data Register	205
12.9.7	IRSCI Baud Rate Register	205
12.9.8	IRSCI Infrared Control Register	208

Chapter 13

Serial Peripheral Interface Module (SPI)

13.1	Introduction	209
13.2	Features	209
13.3	Pin Name Conventions and I/O Register Addresses	209
13.4	Functional Description	210
13.4.1	Master Mode	210
13.4.2	Slave Mode	212
13.5	Transmission Formats	212
13.5.1	Clock Phase and Polarity Controls	212
13.5.2	Transmission Format When CPHA = 0	213
13.5.3	Transmission Format When CPHA = 1	214
13.5.4	Transmission Initiation Latency	214
13.6	Queuing Transmission Data	215
13.7	Error Conditions	216
13.7.1	Overflow Error	217
13.7.2	Mode Fault Error	218
13.8	Interrupts	219
13.9	Resetting the SPI	220
13.10	Low-Power Modes	221
13.10.1	Wait Mode	221
13.10.2	Stop Mode	221
13.11	SPI During Break Interrupts	221
13.12	I/O Signals	222
13.12.1	MISO (Master In/Slave Out)	222
13.12.2	MOSI (Master Out/Slave In)	222
13.12.3	SPSCK (Serial Clock)	222
13.12.4	\overline{SS} (Slave Select)	223

20.2	Features	287
20.3	Functional Description	287
20.3.1	Low V_{DD} Detector	288
20.3.2	Low V_{REG} Detector	288
20.3.3	Polled LVI Operation	288
20.3.4	Forced Reset Operation	289
20.3.5	Voltage Hysteresis Protection	289
20.4	LVI Status Register	289
20.5	LVI Interrupts	289
20.6	Low-Power Modes	290
20.6.1	Wait Mode	290
20.6.2	Stop Mode	290

Chapter 21 Break Module (BRK)

21.1	Introduction	291
21.2	Features	291
21.3	Functional Description	291
21.3.1	Flag Protection During Break Interrupts	292
21.3.2	CPU During Break Interrupts	292
21.3.3	TIM1 and TIM2 During Break Interrupts	292
21.3.4	COP During Break Interrupts	292
21.4	Low-Power Modes	292
21.4.1	Wait Mode	293
21.4.2	Stop Mode	293
21.5	Break Module Registers	293
21.5.1	Break Status and Control Register	293
21.5.2	Break Address Registers	294
21.5.3	SIM Break Status Register	294
21.5.4	SIM Break Flag Control Register	295

Chapter 22 Electrical Specifications

22.1	Introduction	297
22.2	Absolute Maximum Ratings	297
22.3	Functional Operating Range	298
22.4	Thermal Characteristics	298
22.5	5V DC Electrical Characteristics	299
22.6	5V Control Timing	300
22.7	5V Oscillator Characteristics	301
22.8	5V ADC Electrical Characteristics	302
22.9	3V DC Electrical Characteristics	303
22.10	3V Control Timing	304
22.11	3V Oscillator Characteristics	305
22.12	3V ADC Electrical Characteristics	306

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

4.4 Arithmetic/Logic Unit (ALU)

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

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

4.5 Low-Power Modes

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

4.5.1 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

4.5.2 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.

Table 4-1. Instruction Set Summary

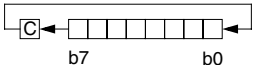
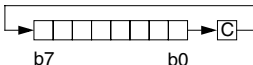
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
PULH	Pull H from Stack	$SP \leftarrow (SP + 1); \text{Pull (H)}$	-	-	-	-	-	-	INH	8A		2
PULX	Pull X from Stack	$SP \leftarrow (SP + 1); \text{Pull (X)}$	-	-	-	-	-	-	INH	88		2
ROL <i>opr</i> ROLA ROLX ROL <i>opr</i> ,X ROL ,X ROL <i>opr</i> ,SP	Rotate Left through Carry		0	-	-	0	0	0	DIR INH INH IX1 IX SP1	39 49 59 69 79 9E69	dd ff ff	4 1 1 4 3 5
ROR <i>opr</i> RORA RORX ROR <i>opr</i> ,X ROR ,X ROR <i>opr</i> ,SP	Rotate Right through Carry		0	-	-	0	0	0	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	4 1 1 4 3 5
RSP	Reset Stack Pointer	$SP \leftarrow \$FF$	-	-	-	-	-	-	INH	9C		1
RTI	Return from Interrupt	$SP \leftarrow (SP + 1); \text{Pull (CCR)}$ $SP \leftarrow (SP + 1); \text{Pull (A)}$ $SP \leftarrow (SP + 1); \text{Pull (X)}$ $SP \leftarrow (SP + 1); \text{Pull (PCH)}$ $SP \leftarrow (SP + 1); \text{Pull (PCL)}$	0	0	0	0	0	0	INH	80		7
RTS	Return from Subroutine	$SP \leftarrow SP + 1; \text{Pull (PCH)}$ $SP \leftarrow SP + 1; \text{Pull (PCL)}$	-	-	-	-	-	-	INH	81		4
SBC # <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> ,X SBC <i>opr</i> ,X SBC ,X SBC <i>opr</i> ,SP SBC <i>opr</i> ,SP	Subtract with Carry	$A \leftarrow (A) - (M) - (C)$	0	-	-	0	0	0	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	$C \leftarrow 1$	-	-	-	-	-	1	INH	99		1
SEI	Set Interrupt Mask	$I \leftarrow 1$	-	-	1	-	-	-	INH	9B		2
STA <i>opr</i> STA <i>opr</i> STA <i>opr</i> ,X STA <i>opr</i> ,X STA ,X STA <i>opr</i> ,SP STA <i>opr</i> ,SP	Store A in M	$M \leftarrow (A)$	0	-	-	0	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 4 3 2 4 5
STHX <i>opr</i>	Store H:X in M	$(M:M + 1) \leftarrow (H:X)$	0	-	-	0	0	-	DIR	35	dd	4
STOP	Enable \overline{IRQ} Pin; Stop Processing	$I \leftarrow 0$; Stop Processing	-	-	0	-	-	-	INH	8E		1

Table 4-1. Instruction Set Summary

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
A	Accumulator	<i>n</i>										
C	Carry/borrow bit	<i>opr</i>										
CCR	Condition code register	PC										
dd	Direct address of operand	PCH										
dd rr	Direct address of operand and relative offset of branch instruction	PCL										
DD	Direct to direct addressing mode	REL										
DIR	Direct addressing mode	<i>rel</i>										
DIX+	Direct to indexed with post increment addressing mode	rr										
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	SP1										
EXT	Extended addressing mode	SP2										
ff	Offset byte in indexed, 8-bit offset addressing	SP										
H	Half-carry bit	U										
H	Index register high byte	V										
hh ll	High and low bytes of operand address in extended addressing	X										
I	Interrupt mask	Z										
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	o										
N	Negative bit	—										



Chapter 5

Oscillator (OSC)

5.1 Introduction

The oscillator module consist of three types of oscillator circuits:

- Internal oscillator
- RC oscillator
- 32.768kHz crystal (x-tal) oscillator

The reference clock for the CGM and other MCU sub-systems is selected by programming the mask option register located at \$FFCF.

The reference clock for the timebase module (TBM) is selected by the two bits, OSCCLK1 and OSCCLK0, in the CONFIG2 register.

The internal oscillator runs continuously after a POR or reset, and is always available. The RC and crystal oscillator cannot run concurrently; one is disabled while the other is selected; because the RC and x-tal circuits share the same OSC1 pin.

NOTE

The oscillator circuits are powered by the on-chip V_{REG} regulator, therefore, the output swing on OSC1 and OSC2 is from V_{SS} to V_{REG} .

Figure 5-1. shows the block diagram of the oscillator module.

5.2 Clock Selection

Reference clocks are selectable for the following sub-systems:

- CGMXCLK and CGMRCLK — Reference clock for clock generator module (CGM) and other MCU sub-systems other than TBM and COP. This is the main reference clock for the MCU.
- OSCCLK — Reference clock for timebase module (TBM).

Enter monitor mode with pin configuration shown in [Figure 8-1](#) by pulling \overline{RST} low and then high. The rising edge of \overline{RST} latches monitor mode. Once monitor mode is latched, the values on the specified pins can change.

Once out of reset, the MCU waits for the host to send eight security bytes. (See [8.4 Security](#).) After the security bytes, the MCU sends a break signal (10 consecutive logic 0's) to the host, indicating that it is ready to receive a command.

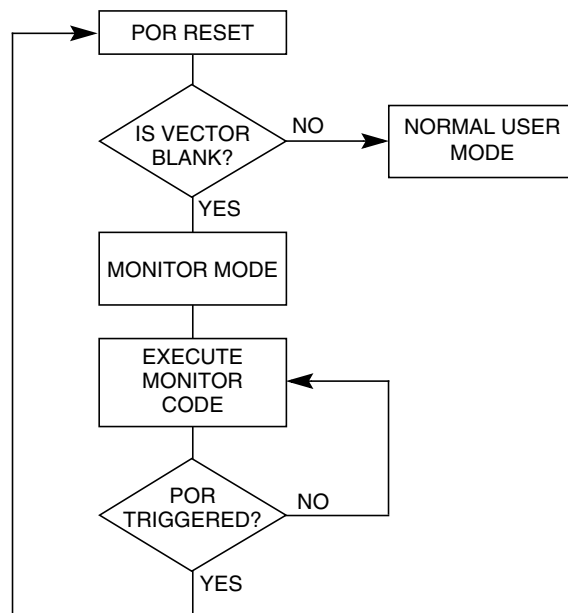


Figure 8-2. Low-Voltage Monitor Mode Entry Flowchart

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

NOTE

Exiting monitor mode after it has been initiated by having a blank reset vector requires a power-on reset (POR). Pulling \overline{RST} low will not exit monitor mode in this situation.

[Table 8-2](#) summarizes the differences between user mode and monitor mode vectors.

Table 8-2. Mode Differences (Vectors)

Modes	Functions					
	Reset Vector High	Reset Vector Low	Break Vector High	Break Vector Low	SWI Vector High	SWI Vector Low
User	\$FFFE	\$FFFF	\$FFFC	\$FFFD	\$FFFC	\$FFFD
Monitor	\$FEFE	\$FEFF	\$FEFC	\$FEFD	\$FEFC	\$FEFD

During the software execution, it does not consume any dedicated RAM location, the run-time heap will extend the system stack, all other RAM location will not be affected.

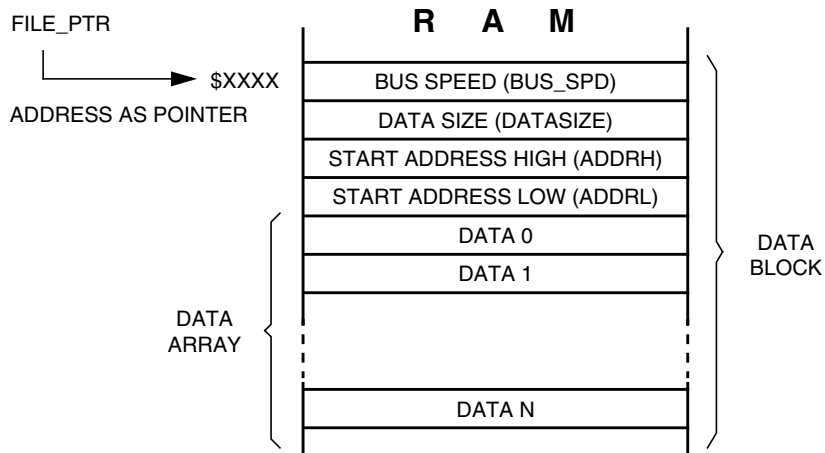


Figure 8-9. Data Block Format for ROM-Resident Routines

The control and data bytes are described below.

- **Bus speed** — This one byte indicates the operating bus speed of the MCU. The value of this byte should be equal to 4 times the bus speed. E.g., for a 4MHz bus, the value is 16 (\$10). This control byte is useful where the MCU clock source is switched between the PLL clock and the crystal clock.
- **Data size** — This one byte indicates the number of bytes in the data array that are to be manipulated. The maximum data array size is 255. Routines EE_WRITE and EE_READ are restricted to manipulate a data array between 7 to 15 bytes. Whereas routines ERARNGE and MON_ERARNGE do not manipulate a data array, thus, this data size byte has no meaning.
- **Start address** — These two bytes, high byte followed by low byte, indicate the start address of the FLASH memory to be manipulated.
- **Data array** — This data array contains data that are to be manipulated. Data in this array are programmed to FLASH memory by the programming routines: PRGRNGE, MON_PRGRNGE, EE_WRITE. For the read routines: LDRNGE and EE_READ, data is read from FLASH and stored in this array.

8.5.1 PRGRNGE

PRGRNGE is used to program a range of FLASH locations with data loaded into the data array.

Table 8-11. PRGRNGE Routine

Routine Name	PRGRNGE
Routine Description	Program a range of locations
Calling Address	\$FC34
Stack Used	15 bytes
Data Block Format	Bus speed (BUS_SPD) Data size (DATASIZE) Start address high (ADDRH) Start address (ADDRL) Data 1 (DATA1) : Data N (DATAN)

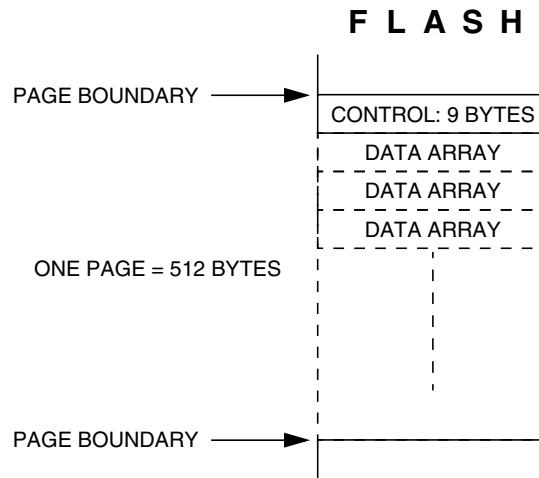


Figure 8-10. EE_WRITE FLASH Memory Usage

The coding example below uses the \$EE00–\$EFFF page for data storage. The data array size is 15 bytes, and the bus speed is 4.9152 MHz. The coding assumes the data block is already loaded in RAM, with the address pointer, FILE_PTR, pointing to the first byte of the data block.

```

                ORG      RAM
:
FILE_PTR:
BUS_SPD        DS.B    1      ; Indicates 4x bus frequency
DATASIZE        DS.B    1      ; Data size to be programmed
START_ADDR     DS.W    1      ; FLASH starting address
DATAARRAY      DS.B    15     ; Reserved data array

EE_WRITE       EQU      $FF36
FLASH_START    EQU      $EE00

                ORG      FLASH
INITIALISATION:
    MOV        #20,     BUS_SPD
    MOV        #15,     DATASIZE
    LDHX       #FLASH_START
    STHX       START_ADDR
    RTS

MAIN:
    BSR        INITIALISATION
:
:
    LHDX       #FILE_PTR
    JSR        EE_WRITE

```

NOTE

The EE_WRITE routine is unable to check for incorrect data blocks, such as the FLASH page boundary address and data size. It is the responsibility of the user to ensure the starting address indicated in the data block is at the FLASH page boundary and the data size is 7 to 15. If the FLASH page is already programmed with a data array with a different size, the EE_WRITE call will be ignored.

11.8.3 SCI Control Register 3

SCI control register 3:

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

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

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

R8 — Received Bit 8

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

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

T8 — Transmitted Bit 8

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

DMARE — DMA Receive Enable Bit

CAUTION

The DMA module is not included on this MCU. Writing a logic 1 to DMARE or DMATE may adversely affect MCU performance.

1 = DMA not enabled to service SCI receiver DMA service requests generated by the SCRF bit (SCI receiver CPU interrupt requests enabled)

0 = DMA not enabled to service SCI receiver DMA service requests generated by the SCRF bit (SCI receiver CPU interrupt requests enabled)

DMATE — DMA Transfer Enable Bit

CAUTION

The DMA module is not included on this MCU. Writing a logic 1 to DMARE or DMATE may adversely affect MCU performance.

1 = SCTE DMA service requests enabled; SCTE CPU interrupt requests disabled

0 = SCTE DMA service requests disabled; SCTE CPU interrupt requests enabled

ORIE — Receiver Overrun Interrupt Enable Bit

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

1 = SCI error CPU interrupt requests from OR bit enabled

0 = SCI error CPU interrupt requests from OR bit disabled

- Framing error (FE) — The FE bit in IRSCS1 is set when a logic 0 occurs where the receiver expects a stop bit. The framing error interrupt enable bit, FEIE, in IRSCC3 enables FE to generate SCI error CPU interrupt requests.
- Parity error (PE) — The PE bit in IRSCS1 is set when the SCI detects a parity error in incoming data. The parity error interrupt enable bit, PEIE, in IRSCC3 enables PE to generate SCI error CPU interrupt requests.

12.6 Low-Power Modes

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

12.6.1 Wait Mode

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

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

Refer to [7.6 Low-Power Modes](#) for information on exiting wait mode.

12.6.2 Stop Mode

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

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

Refer to [7.6 Low-Power Modes](#) for information on exiting stop mode.

12.7 SCI During Break Module Interrupts

The system integration module (SIM) controls whether status bits in other modules can be cleared during interrupts generated by the break module. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state.

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

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

12.8 I/O Signals

The two IRSCI I/O pins are:

- PTC6/SCTxD — Transmit data
- PTC7/SCRxD — Receive data

12.8.1 PTC6/SCTxD (Transmit Data)

The PTC6/SCTxD pin is the serial data (standard or infrared) output from the SCI transmitter. The IRSCI shares the PTC6/SCTxD pin with port C. When the IRSCI is enabled, the PTC6/SCTxD pin is an output regardless of the state of the DDRC6 bit in data direction register C (DDRC).

NOTE

The PTC6/SCTxD pin is an open-drain pin when configured as an output. Therefore, when configured as SCTxD or a general purpose output pin (PTC6), a pullup resistor must be connected to this pin.

12.8.2 PTC7/SCRxD (Receive Data)

The PTC7/SCRxD pin is the serial data input to the IRSCI receiver. The IRSCI shares the PTC7/SCRxD pin with port C. When the IRSCI is enabled, the PTC7/SCRxD pin is an input regardless of the state of the DDRC7 bit in data direction register C (DDRC).

NOTE

The PTC7/SCRxD pin is an open-drain pin when configured as an output. Therefore, when configured as a general purpose output pin (PTC7), a pullup resistor must be connected to this pin.

Table 12-5 shows a summary of I/O pin functions when the SCI is enabled.

Table 12-5. SCI Pin Functions (Standard and Infrared)

IRSCC1 [ENSCI]	IRSCIRCR [IREN]	IRSCC2 [TE]	IRSCC2 [RE]	TxD Pin	RxD Pin
1	0	0	0	Hi-Z ⁽¹⁾	Input ignored (terminate externally)
1	0	0	1	Hi-Z ⁽¹⁾	Input sampled, pin should idle high
1	0	1	0	Output SCI (idle high)	Input ignored (terminate externally)
1	0	1	1	Output SCI (idle high)	Input sampled, pin should idle high
1	1	0	0	Hi-Z ⁽¹⁾	Input ignored (terminate externally)
1	1	0	1	Hi-Z ⁽¹⁾	Input sampled, pin should idle high
1	1	1	0	Output IR SCI (idle high)	Input ignored (terminate externally)
1	1	1	1	Output IR SCI (idle high)	Input sampled, pin should idle high
0	X	X	X	Pins under port control (standard I/O port)	

1. After completion of transmission in progress.

12.9 I/O Registers

The following I/O registers control and monitor SCI operation:

- IRSCI control register 1 (IRSCC1)
- IRSCI control register 2 (IRSCC2)
- IRSCI control register 3 (IRSCC3)

Infrared Serial Communications Interface Module (IRSCI)

- IRSCI status register 1 (IRSCS1)
- IRSCI status register 2 (IRSCS2)
- IRSCI data register (IRSCDR)
- IRSCI baud rate register (IRSCBR)
- IRSCI infrared control register (IRSCIRCR)

12.9.1 IRSCI Control Register 1

SCI control register 1:

- Enables loop mode operation
- Enables the SCI
- Controls output polarity
- Controls character length
- Controls SCI wakeup method
- Controls idle character detection
- Enables parity function
- Controls parity type

Address:	\$0040							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	LOOPS	ENSCI	0	M	WAKE	ILTY	PEN	PTY
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 12-12. IRSCI Control Register 1 (IRSCC1)

LOOPS — Loop Mode Select Bit

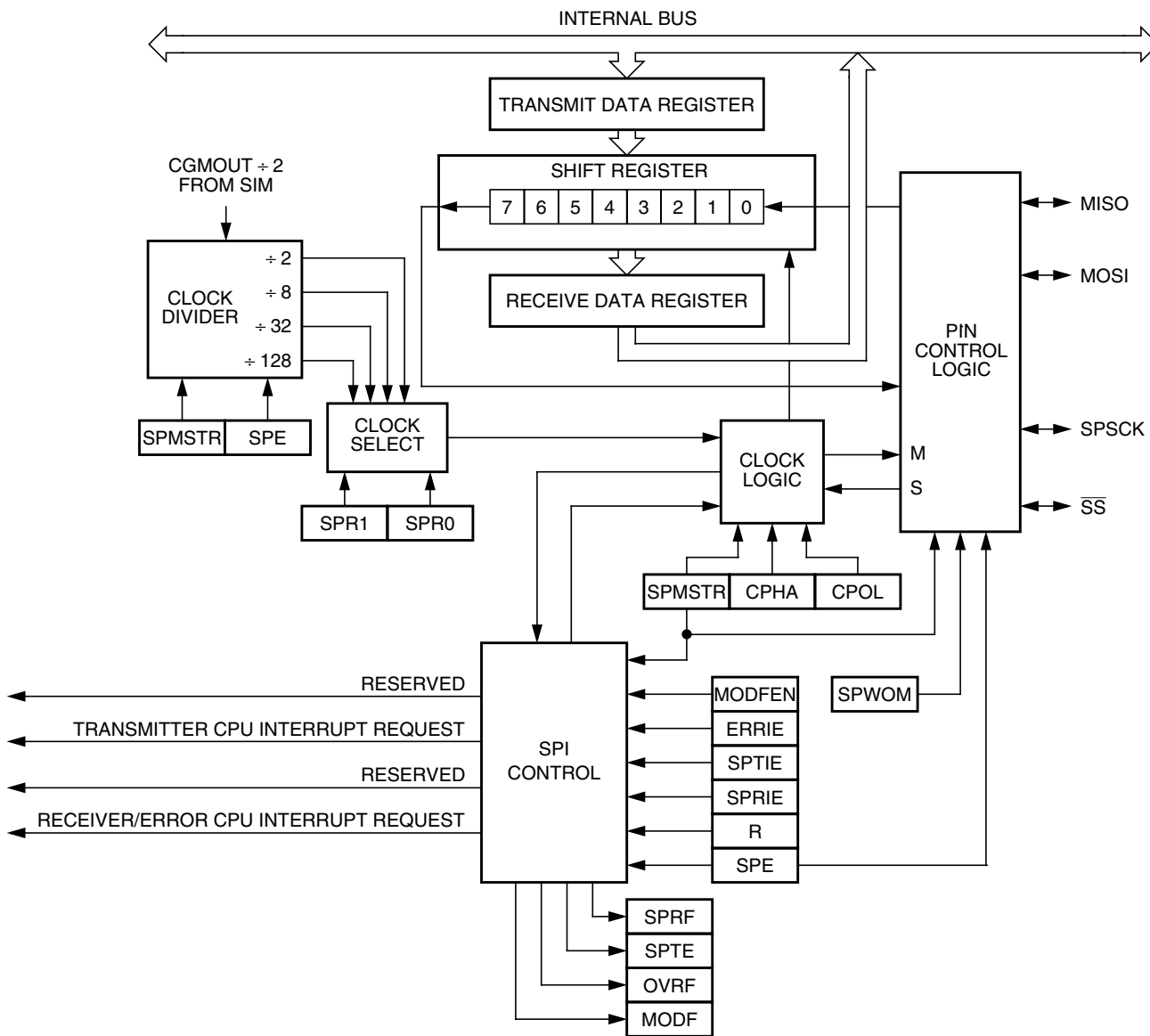
This read/write bit enables loop mode operation for the SCI only. In loop mode the RxD pin is disconnected from the SCI, and the transmitter output goes into the receiver input. Both the transmitter and the receiver must be enabled to use loop mode. The infrared encoder/decoder is not in the loop. Reset clears the LOOPS bit.

- 1 = Loop mode enabled
- 0 = Normal operation enabled

ENSCI — Enable SCI Bit

This read/write bit enables the SCI and the SCI baud rate generator. Clearing ENSCI sets the SCTE and TC bits in SCI status register 1 and disables transmitter interrupts. Reset clears the ENSCI bit.

- 1 = SCI enabled
- 0 = SCI disabled



13.4.2 Slave Mode

The SPI operates in slave mode when the SPMSTR bit is clear. In slave mode, the SPSCCK pin is the input for the serial clock from the master MCU. Before a data transmission occurs, the \overline{SS} pin of the slave SPI must be at logic 0. \overline{SS} must remain low until the transmission is complete. (See [13.7.2 Mode Fault Error](#).)

In a slave SPI module, data enters the shift register under the control of the serial clock from the master SPI module. After a byte enters the shift register of a slave SPI, it transfers to the receive data register, and the SPRF bit is set. To prevent an overflow condition, slave software then must read the receive data register before another full byte enters the shift register.

The maximum frequency of the SPSCCK for an SPI configured as a slave is the bus clock speed (which is twice as fast as the fastest master SPSCCK clock that can be generated). The frequency of the SPSCCK for an SPI configured as a slave does not have to correspond to any SPI baud rate. The baud rate only controls the speed of the SPSCCK generated by an SPI configured as a master. Therefore, the frequency of the SPSCCK for an SPI configured as a slave can be any frequency less than or equal to the bus speed.

When the master SPI starts a transmission, the data in the slave shift register begins shifting out on the MISO pin. The slave can load its shift register with a new byte for the next transmission by writing to its transmit data register. The slave must write to its transmit data register at least one bus cycle before the master starts the next transmission. Otherwise, the byte already in the slave shift register shifts out on the MISO pin. Data written to the slave shift register during a transmission remains in a buffer until the end of the transmission.

When the clock phase bit (CPHA) is set, the first edge of SPSCCK starts a transmission. When CPHA is clear, the falling edge of \overline{SS} starts a transmission. (See [13.5 Transmission Formats](#).)

NOTE

SPSCCK must be in the proper idle state before the slave is enabled to prevent SPSCCK from appearing as a clock edge.

13.5 Transmission Formats

During an SPI transmission, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). A serial clock synchronizes shifting and sampling on the two serial data lines. A slave select line allows selection of an individual slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. On a master SPI device, the slave select line can optionally be used to indicate multiple-master bus contention.

13.5.1 Clock Phase and Polarity Controls

Software can select any of four combinations of serial clock (SPSCCK) phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, which selects an active high or low clock and has no significant effect on the transmission format.

The clock phase (CPHA) control bit selects one of two fundamentally different transmission formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

NOTE

Before writing to the CPOL bit or the CPHA bit, disable the SPI by clearing the SPI enable bit (SPE).

16.2 Port A

Port A is an 8-bit special-function port that shares all of its pins with the analog-to-digital converter (ADC) module. Port A pins also have LED direct drive capability.

16.2.1 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:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Reset:	Unaffected by reset							
Alternative Function:	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
Additional Function:	LED drive	LED drive	LED drive	LED drive	LED drive	LED drive	LED drive	LED drive

Figure 16-2. 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.

ADC7–ADC0 — ADC Channels 7 to 0

ADC7–ADC0 are pins used for the input channels to the analog-to-digital converter module. The channel select bits, ADCH[4:0], in the ADC status and control register define which port pin will be used as an ADC input and overrides any control from the port I/O logic.

NOTE

Care must be taken when reading port A while applying analog voltages to ADC7–ADC0 pins. If the appropriate ADC channel is not enabled, excessive current drain may occur if analog voltages are applied to the PTAx/ADCx pin, while PTA is read as a digital input. Those ports not selected as analog input channels are considered digital I/O ports.

LED drive — Direct LED drive pins

PTA7–PTA0 pins can be configured for direct LED drive. See [16.2.3 Port-A LED Control Register \(LEDA\)](#).

16.2.2 Data Direction Register (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:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset:	0	0	0	0	0	0	0	0

Figure 16-3. Data Direction Register A (DDRA)

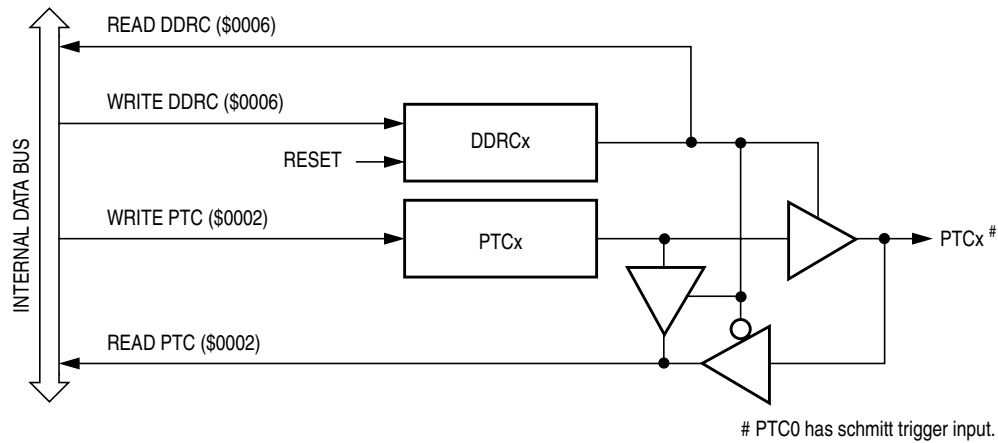


Figure 16-11. Port C I/O Circuit

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 16-4 summarizes the operation of the port C pins.

Table 16-4. 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.

16.5 Port D

Port D is an 8-bit special function port that shares all of its pins with the keyboard interrupt module.

16.5.1 Port D Data Register (PTD)

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

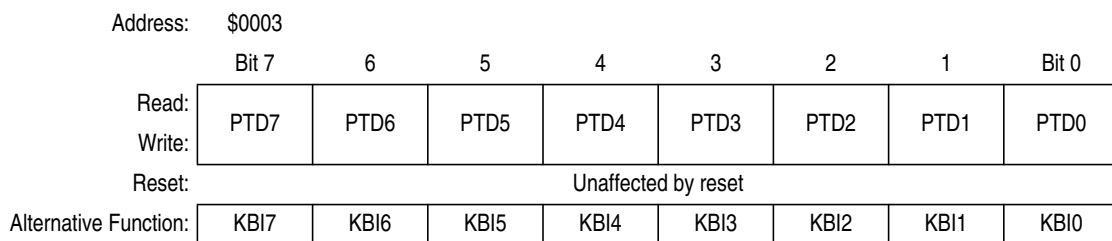


Figure 16-12. Port D Data Register (PTD)

Chapter 21

Break Module (BRK)

21.1 Introduction

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

21.2 Features

Features of the break module include:

- Accessible input/output (I/O) registers during the break interrupt
- CPU-generated break interrupts
- Software-generated break interrupts
- COP disabling during break interrupts

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$FE00	SIM Break Status Register (SBSR)	Read:	R	R	R	R	R	R	SBSW	R
		Write:							Note	
		Reset:	0							
\$FE03	SIM Break Flag Control Register (SBFCR)	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
\$FE0C	Break Address Register High (BRKH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE0D	Break Address Register Low (BRKL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE0E	Break Status and Control Register (BRKSCR)	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

Note: Writing a logic 0 clears BW.


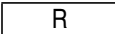
 = Unimplemented  = Reserved

Figure 21-1. Break Module I/O Register Summary

21.3 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal to the CPU. The CPU then loads the instruction register with a software interrupt instruction (SWI) after completion of the current CPU instruction. The program counter vectors to \$FFFC and \$FFFD (\$FEFC and \$FEFD in monitor mode).

23.3 44-Pin Quad Flat Pack (QFP)

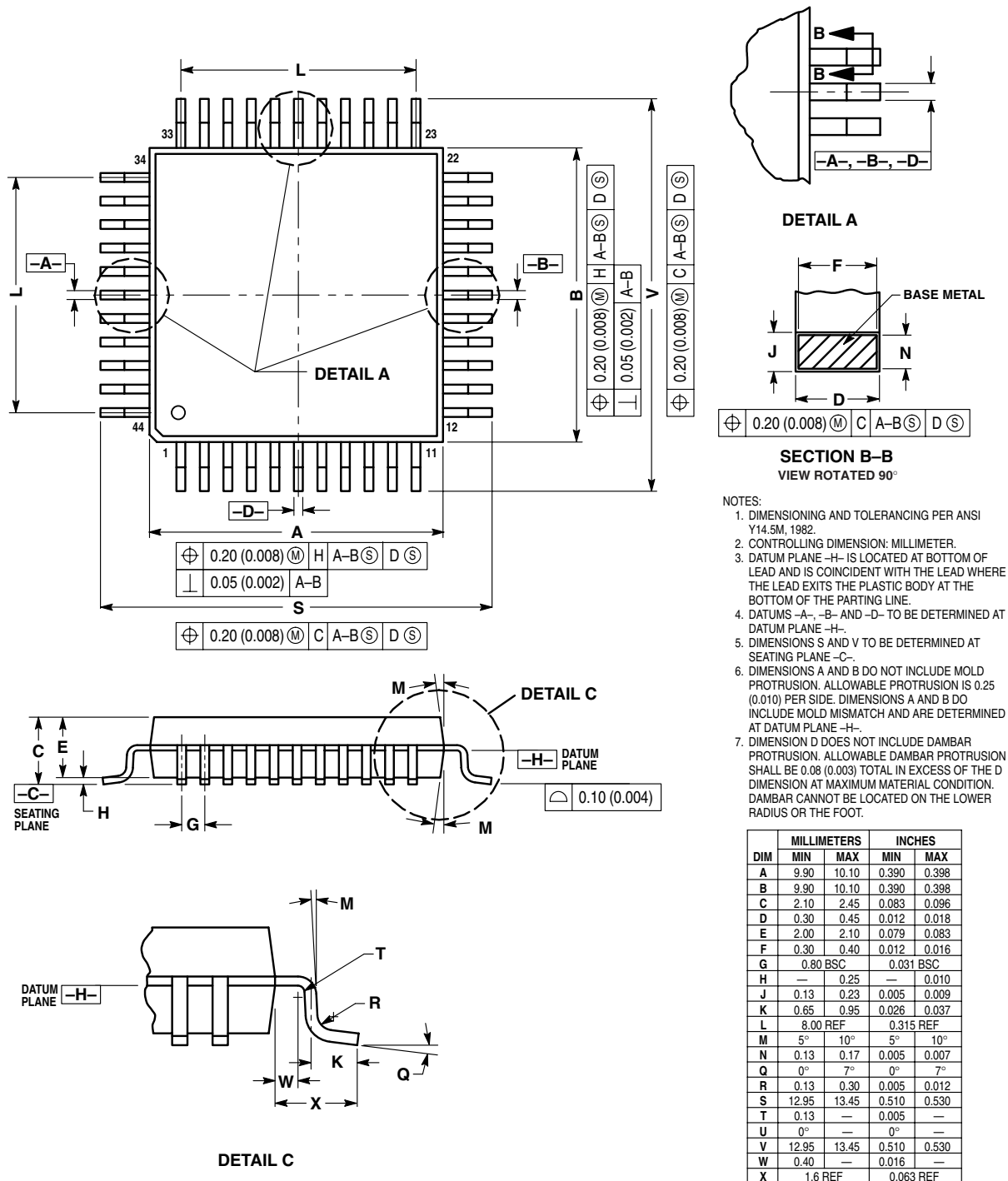


Figure 23-2. 44-Pin QFP (Case #824A)