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	Active
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	I²C, IRSCI, SCI, SPI
Peripherals	LED, LVD, POR, PWM
Number of I/O	32
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 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/pro/item?MUrl=&PartUrl=mc908ap32acfbe

2.5.7 FLASH Block Protect Register

The FLASH block protect register is implemented as an 8-bit I/O register. The value in this register determines the starting address of the protected range within the FLASH memory.

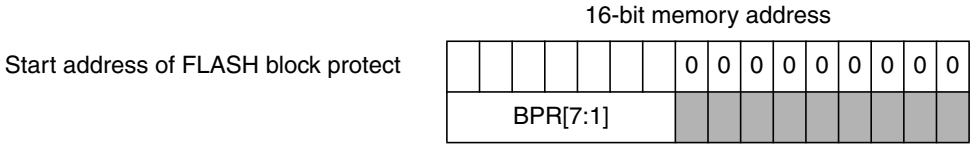
Address: $FE09

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

Figure 2-5. FLASH Block Protect Register (FLBPR)

BPR[7:0] — FLASH Block Protect Bits

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



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

The resultant 16-bit address is used for specifying the start address of the FLASH memory for block protection. The FLASH is protected from this start address to the end of FLASH memory, at $FFFF. With this mechanism, the protect start address can be X000, X200, X400, X0600, X800, XA00, XC00, or XE00 (at page boundaries — 512 bytes) within the FLASH memory.

Examples of protect start address:

Table 2-2 FLASH Block Protect Range

BPR[7:0]	Protected Range
$00 to $09	The entire FLASH memory is protected.
$0A or $0B (**0000 101x**)	$0A00 to $FFFF
$0C or $0D (**0000 110x**)	$0C00 to $FFFF
and so on...	
$FA or $FB (**1111 1101x**)	$FA00 to $FFFF
$FC or $FD or $FE	$FFCF to $FFFF
$FF	The entire FLASH memory is **NOT** protected.(1)

1. Except for the mask option register ($FFCF) and the 48-byte user vectors ($FFD0–$FFFF). These FLASH locations are always protected.

Chapter 3
Configuration & Mask Option Registers (CONFIG & 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 (262,128 or 8176 ICLK cycles)
- Low-voltage inhibit (LVI) on V_{DD}
- LVI on V_{REG}
- LVI module reset
- LVI module in stop mode
- STOP instruction
- Stop mode recovery time (32 ICLK or 4096 ICLK cycles)
- Oscillator (internal, RC, and crystal) during stop mode
- Serial communications interface clock source (CGMXCLK or f_{BUS})

The mask option register selects one of the following oscillator options:

- Internal oscillator
- RC oscillator
- Crystal oscillator

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001D	Configuration Register 2 (CONFIG2)†	Read: Write:	STOP_ ICLKDIS	STOP_ RCLKEN	STOP_ XCLKEN	OSCCLK1	OSCCLK0	0	0	SCIBDSRC
		Reset:	0	0	0	0	0	0	0	0
$001F	Configuration Register 1 (CONFIG1)†	Read: Write:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVIREGD	SSREC	STOP	COPD
		Reset:	0	0	0	0	0	0	0	0
$FFCF	Mask-Option-Register (MOR)#	Read: Write:	OSCSEL1	OSCSEL0	R	R	R	R	R	R
		Erased:	1	1	1	1	1	1	1	1

† One-time writable register after each reset.
# MOR is a non-volatile FLASH register; write by programming.

= Unimplemented R = Reserved

Figure 3-1. CONFIG and MOR Registers Summary

register disables the PLL and clears the BCS bit in the PLL control register (PCTL). (See 6.3.8 Base Clock Selector Circuit and 6.3.7 Special Programming Exceptions.). Reset initializes the register to $40 for a default range multiply value of 64.

NOTE

The VCO range select bits have built-in protection such that they cannot be written when the PLL is on (PLLON = 1) and such that the VCO clock cannot be selected as the source of the base clock (BCS = 1) if the VCO range select bits are all clear.

The PLL VCO range select register must be programmed correctly. Incorrect programming can result in failure of the PLL to achieve lock.

6.5.5 PLL Reference Divider Select Register

The PLL reference divider select register (PMDS) contains the programming information for the modulo reference divider.

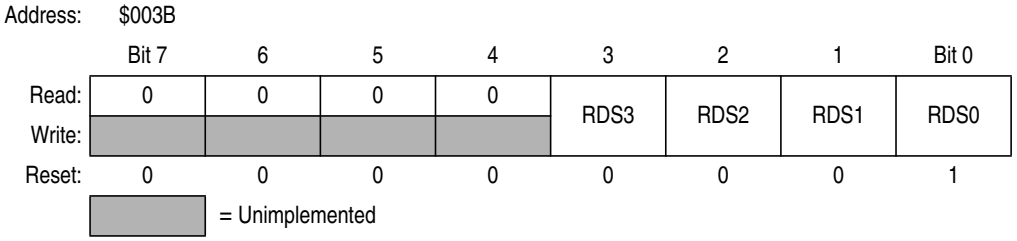


Figure 6-9. PLL Reference Divider Select Register (PMDS)

RDS[3:0] — Reference Divider Select Bits

These read/write bits control the modulo reference divider that selects the reference division factor, R. (See 6.3.3 PLL Circuits and 6.3.6 Programming the PLL.) RDS[3:0] cannot be written when the PLLON bit in the PCTL is set. A value of $00 in the reference divider select register configures the reference divider the same as a value of $01. (See 6.3.7 Special Programming Exceptions.) Reset initializes the register to $01 for a default divide value of 1.

NOTE

The reference divider select bits have built-in protection such that they cannot be written when the PLL is on (PLLON = 1).

NOTE

The default divide value of 1 is recommended for all applications.

6.6 Interrupts

When the AUTO bit is set in the PLL bandwidth control register (PBWC), the PLL can generate a CPU interrupt request every time the LOCK bit changes state. The PLLIE bit in the PLL control register (PCTL) enables CPU interrupts from the PLL. PLLF, the interrupt flag in the PCTL, becomes set whether interrupts are enabled or not. When the AUTO bit is clear, CPU interrupts from the PLL are disabled and PLLF reads as logic 0.

Software should read the LOCK bit after a PLL interrupt request to see if the request was due to an entry into lock or an exit from lock. When the PLL enters lock, the divided VCO clock, CGMPCLK, divided by two can be selected as the CGMOUT source by setting BCS in the PCTL. When the PLL exits lock, the

8.3.1 Entering Monitor Mode

Table 8-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a POR and will allow communication at 9600 baud provided one of the following sets of conditions is met:

1. If $FFFE and $FFFF do not contain $FF (programmed state):
   - The external clock is 4.9152 MHz with PTB0 low or 9.8304 MHz with PTB0 high
   - $\overline{IRQ1} = V_{TST}$
2. If $FFFE and $FFFF both contain $FF (erased state):
   - The external clock is 9.8304 MHz
   - $\overline{IRQ1} = V_{DD}$ (this can be implemented through the internal $\overline{IRQ1}$ pullup)

If V_{TST} is applied to $\overline{IRQ1}$ and PTB0 is low upon monitor mode entry (above condition set 1), the bus frequency is a divide-by-two of the input clock. If PTB0 is high with V_{TST} applied to $\overline{IRQ1}$ upon monitor mode entry, the bus frequency will be a divide-by-four of the input clock. Holding the PTB0 pin low when entering monitor mode causes a bypass of a divide-by-two stage at the oscillator only if V_{TST} is applied to $\overline{IRQ1}$. In this event, the CGMOUT frequency is equal to the CGMXCLK frequency, and the OSC1 input directly generates internal bus clocks. In this case, the OSC1 signal must have a 50% duty cycle at maximum bus frequency.

If entering monitor mode without high voltage on $\overline{IRQ1}$ (above condition set 2), then all port A pin requirements and conditions, including the PTB0 frequency divisor selection, are not in effect. This is to reduce circuit requirements when performing in-circuit programming.

NOTE

If the reset vector is blank and monitor mode is entered, the chip will see an additional reset cycle after the initial POR reset. Once the part has been programmed, the traditional method of applying a voltage, V_{TST}, to $\overline{IRQ1}$ must be used to enter monitor mode.

The COP module is disabled in monitor mode based on these conditions:

- If monitor mode was entered as a result of the reset vector being blank (above condition set 2), the COP is always disabled regardless of the state of $\overline{IRQ1}$ or $\overline{RST}$.
- If monitor mode was entered with V_{TST} on $\overline{IRQ1}$ (condition set 1), then the COP is disabled as long as V_{TST} is applied to either $\overline{IRQ1}$ or $\overline{RST}$.

The second condition states that as long as V_{TST} is maintained on the $\overline{IRQ1}$ pin after entering monitor mode, or if V_{TST} is applied to $\overline{RST}$ after the initial reset to get into monitor mode (when V_{TST} was applied to $\overline{IRQ1}$), then the COP will be disabled. In the latter situation, after V_{TST} is applied to the $\overline{RST}$ pin, V_{TST} can be removed from the $\overline{IRQ1}$ pin in the interest of freeing the $\overline{IRQ1}$ for normal functionality in monitor mode.

Figure 8-2 shows a simplified diagram of the monitor mode entry when the reset vector is blank and just V_{DD} voltage is applied to the $\overline{IRQ1}$ pin. An external oscillator of 9.8304 MHz is required for a baud rate of 9600, as the internal bus frequency is automatically set to the external frequency divided by four.

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.

8.5 ROM-Resident Routines

Seven routines stored in the monitor ROM area (thus ROM-resident) are provided for FLASH memory manipulation. Five of the seven routines are intended to simplify FLASH program, erase, and load operations. The other two routines are intended to simplify the use of the FLASH memory as EEPROM. Table 8-10 shows a summary of the ROM-resident routines.

Table 8-10. Summary of ROM-Resident Routines

Routine Name	Routine Description	Call Address	Stack Used (bytes)
PRGRNGE	Program a range of locations	$FC34	15
ERARNGE	Erase a page or the entire array	$FCE4	9
LDRNGE	Loads data from a range of locations	$FC00	7
MON_PRGRNGE	Program a range of locations in monitor mode	$FF24	17
MON_ERARNGE	Erase a page or the entire array in monitor mode	$FF28	11

The routines are designed to be called as stand-alone subroutines in the user program or monitor mode. The parameters that are passed to a routine are in the form of a contiguous data block, stored in RAM. The index register (H:X) is loaded with the address of the first byte of the data block (acting as a pointer), and the subroutine is called (JSR). Using the start address as a pointer, multiple data blocks can be used, any area of RAM be used. A data block has the control and data bytes in a defined order, as shown in Figure 8-9.

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.

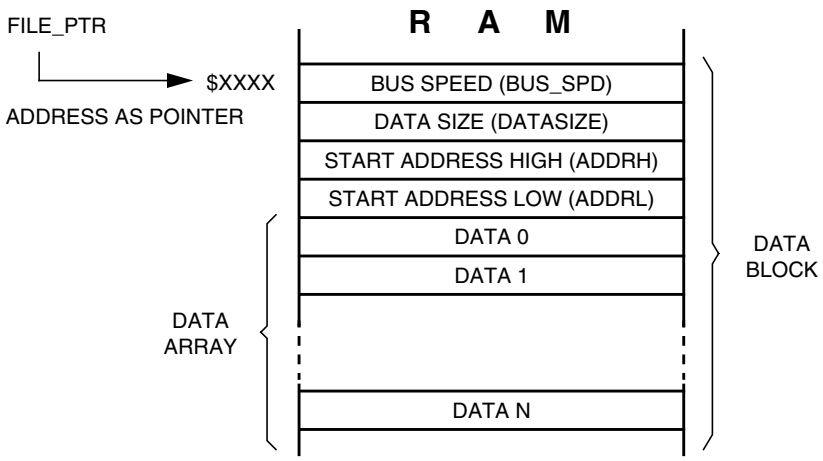


Figure 8-9. Data Block Format for ROM-Resident Routines

The control and data bytes are described below.

The coding example below is to perform a page erase, from $EE00–$EFFF. The Initialization subroutine is the same as the coding example for PRGRNGE (see 8.5.1 PRGRNGE).

```
ERARNGE         EQU     $FCE4
MAIN:
        BSR     INITIALISATION
        :
        :
        LDHX    #FILE_PTR
        JSR     ERARNGE
        :
```

8.5.3 LDRNGE

LDRNGE is used to load the data array in RAM with data from a range of FLASH locations.

Table 8-13. LDRNGE Routine

Routine Name	LDRNGE
Routine Description	Loads data from a range of locations
Calling Address	$FC00
Stack Used	7 bytes
Data Block Format	Bus speed (BUS_SPD) Data size (DATASIZE) Starting address (ADDRH) Starting address (ADDRL) Data 1 : Data N

The start location of FLASH from where data is retrieved is specified by the address ADDRH:ADDRL and the number of bytes from this location is specified by DATASIZE. The maximum number of bytes that can be retrieved in one routine call is 255 bytes. The data retrieved from FLASH is loaded into the data array in RAM. Previous data in the data array will be overwritten. User can use this routine to retrieve data from FLASH that was previously programmed.

The coding example below is to retrieve 64 bytes of data starting from $EE00 in FLASH. The Initialization subroutine is the same as the coding example for PRGRNGE (see 8.5.1 PRGRNGE).

```
LDRNGE          EQU     $FC00
MAIN:
        BSR     INITIALIZATION
        :
        :
        LDHX    #FILE_PTR
        JSR     LDRNGE
        :
```

11.3 Pin Name Conventions

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

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

SCI I/O (input/output) lines are implemented by sharing parallel I/O port pins. The full name of an SCI input or output reflects the name of the shared port pin. Table 11-1 shows the full names and the generic names of the SCI I/O pins. The generic pin names appear in the text of this section.

Table 11-1. Pin Name Conventions

Generic Pin Names:	RxD	TxD
Full Pin Names:	PTB3/RxD	PTB2/TxD

NOTE

When the SCI is enabled, the TxD pin is an open-drain output and requires a pullup resistor to be connected for proper SCI operation.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0013	SCI Control Register 1 (SCC1)	Read: Write:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Reset:	0	0	0	0	0	0	0	0
$0014	SCI Control Register 2 (SCC2)	Read: Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Reset:	0	0	0	0	0	0	0	0
$0015	SCI Control Register 3 (SCC3)	Read:	R8	T8	DMARE	DMATE	ORIE	NEIE	FEIE	PEIE
		Write:								
		Reset:	U	U	0	0	0	0	0	0
$0016	SCI Status Register 1 (SCS1)	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:								
		Reset:	1	1	0	0	0	0	0	0
$0017	SCI Status Register 2 (SCS2)	Read:							BKF	RPF
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0018	SCI Data Register (SCDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							
$0019	SCI Baud Rate Register (SCBR)	Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved U = Unaffected

Figure 11-1. SCI I/O Register Summary

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

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

Fast Data Tolerance

Figure 11-8 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.

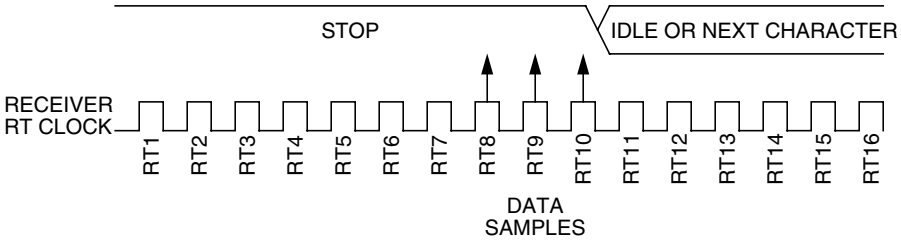


Figure 11-8. Fast Data

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 11-8, 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

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

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 11-8, 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

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

11.4.3.6 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.

11.8.2 SCI Control Register 2

SCI control register 2:

- Enables the following CPU interrupt requests:
  - Enables the SCTE bit to generate transmitter CPU interrupt requests
  - Enables the TC bit to generate transmitter CPU interrupt requests
  - Enables the SCRF bit to generate receiver CPU interrupt requests
  - Enables the IDLE bit to generate receiver CPU interrupt requests
- Enables the transmitter
- Enables the receiver
- Enables SCI wakeup
- Transmits SCI break characters

Address: $0014

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

Figure 11-10. SCI Control Register 2 (SCC2)

SCTIE — SCI Transmit Interrupt Enable Bit

This read/write bit enables the SCTE bit to generate SCI transmitter CPU interrupt requests. Reset clears the SCTIE bit.

1 = SCTE enabled to generate CPU interrupt
0 = SCTE not enabled to generate CPU interrupt

TCIE — Transmission Complete Interrupt Enable Bit

This read/write bit enables the TC bit to generate SCI transmitter CPU interrupt requests. Reset clears the TCIE bit.

1 = TC enabled to generate CPU interrupt requests
0 = TC not enabled to generate CPU interrupt requests

SCRIE — SCI Receive Interrupt Enable Bit

This read/write bit enables the SCRF bit to generate SCI receiver CPU interrupt requests. Reset clears the SCRIE bit.

1 = SCRF enabled to generate CPU interrupt
0 = SCRF not enabled to generate CPU interrupt

ILIE — Idle Line Interrupt Enable Bit

This read/write bit enables the IDLE bit to generate SCI receiver CPU interrupt requests. Reset clears the ILIE bit.

1 = IDLE enabled to generate CPU interrupt requests
0 = IDLE not enabled to generate CPU interrupt requests

12.3 IRSCI Module Overview

The IRSCI consists of a serial communications interface (SCI) and a infrared interface sub-module as shown in Figure 12-2.

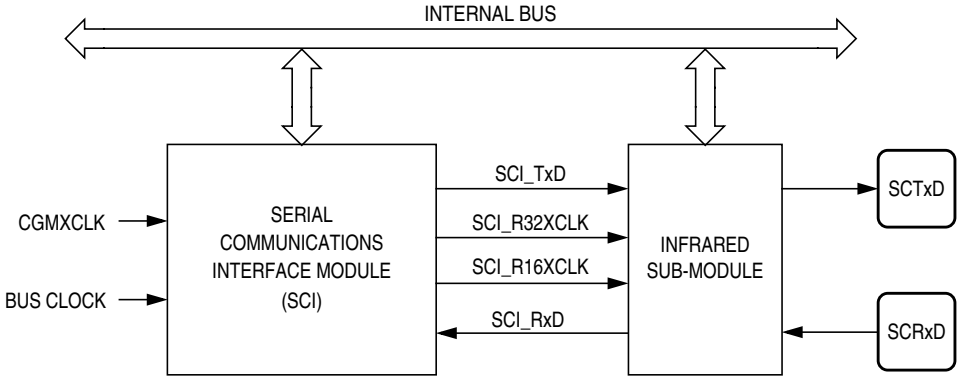


Figure 12-2. IRSCI Block Diagram

The SCI module provides serial data transmission and reception, with a programmable baud rate clock based on the bus clock or the CGMXCLK.

The infrared sub-module receives two clock sources from the SCI module: SCI_R16XCLK and SCI_R32XCLK. Both reference clocks are used to generate the narrow pulses during data transmission. The SCI_R16XCLK and SCI_R32XCLK are internal clocks with frequencies that are 16 and 32 times the baud rate respectively. Both SCI_R16XCLK and SCI_R32XCLK clocks are used for transmitting data. The SCI_R16XCLK clock is used only for receiving data.

NOTE

For proper SCI function (transmit or receive), the bus clock MUST be programmed to at least 32 times that of the selected baud rate.
When the infrared sub-module is disabled, signals on the TxD and RxD pins pass through unchanged to the SCI module.

12.4 Infrared Functional Description

Figure 12-3 shows the structure of the infrared sub-module.

The infrared sub-module provides the capability of transmitting narrow pulses to an infrared LED and receiving narrow pulses and transforming them to serial bits, which are sent to the SCI module. The infrared sub-module receives two clocks from the SCI. One of these two clocks is selected as the base clock to generate the 3/16, 1/16, or 1/32 bit width narrow pulses during transmission.

The sub-module consists of two main blocks: the transmit encoder and the receive decoder. When transmitting data, the SCI data stream is encoded by the infrared sub-module. For every "0" bit, a narrow "low" pulse is transmitted; no pulse is transmitted for "1" bits. When receiving data, the infrared pulses should be detected using an infrared photo diode for conversion to CMOS voltage levels before connecting to the RxD pin for the infrared decoder. The SCI data stream is reconstructed by stretching the "0" pulses.

13.12.4 $\overline{SS}$ (Slave Select)

The $\overline{SS}$ pin has various functions depending on the current state of the SPI. For an SPI configured as a slave, the $\overline{SS}$ is used to select a slave. For CPHA = 0, the $\overline{SS}$ is used to define the start of a transmission. (See 13.5 Transmission Formats.) Since it is used to indicate the start of a transmission, the $\overline{SS}$ must be toggled high and low between each byte transmitted for the CPHA = 0 format. However, it can remain low between transmissions for the CPHA = 1 format. See Figure 13-12.

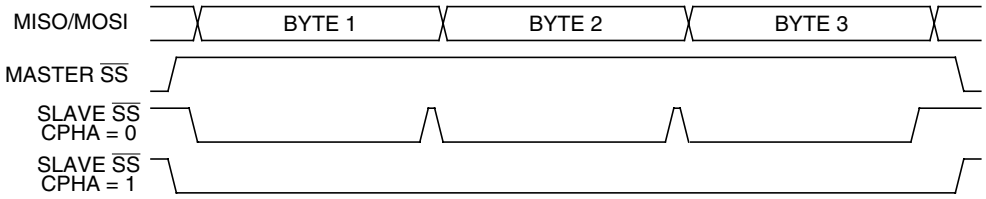


Figure 13-12. CPHA/$\overline{SS}$ Timing

When an SPI is configured as a slave, the $\overline{SS}$ pin is always configured as an input. It cannot be used as a general-purpose I/O regardless of the state of the MODFEN control bit. However, the MODFEN bit can still prevent the state of the $\overline{SS}$ from creating a MODF error. (See 13.13.2 SPI Status and Control Register.)

NOTE

A logic 1 voltage on the $\overline{SS}$ pin of a slave SPI puts the MISO pin in a high-impedance state. The slave SPI ignores all incoming SPSCK clocks, even if it was already in the middle of a transmission.

When an SPI is configured as a master, the $\overline{SS}$ input can be used in conjunction with the MODF flag to prevent multiple masters from driving MOSI and SPSCK. (See 13.7.2 Mode Fault Error.) For the state of the $\overline{SS}$ pin to set the MODF flag, the MODFEN bit in the SPSCK register must be set. If the MODFEN bit is low for an SPI master, the $\overline{SS}$ pin can be used as a general-purpose I/O under the control of the data direction register of the shared I/O port. With MODFEN high, it is an input-only pin to the SPI regardless of the state of the data direction register of the shared I/O port.

The CPU can always read the state of the $\overline{SS}$ pin by configuring the appropriate pin as an input and reading the port data register. (See Table 13-3.)

Table 13-3. SPI Configuration

SPE	SPMSTR	MODFEN	SPI Configuration	State of $\overline{SS}$ Logic
0	X(1)	X	Not enabled	General-purpose I/O; $\overline{SS}$ ignored by SPI
1	0	X	Slave	Input-only to SPI
1	1	0	Master without MODF	General-purpose I/O; $\overline{SS}$ ignored by SPI
1	1	1	Master with MODF	Input-only to SPI

Note 1. X = Don't care

Table 13-4. SPI Master Baud Rate Selection

SPR1 and SPR0	Baud Rate Divisor (BD)
00	2
01	8
10	32
11	128

Use this formula to calculate the SPI baud rate:

$$\text{Baud rate} = \frac{\text{CGMOUT}}{2 \times \text{BD}}$$

where:

CGMOUT = base clock output of the clock generator module (CGM)
BD = baud rate divisor

13.13.3 SPI Data Register

The SPI data register consists of the read-only receive data register and the write-only transmit data register. Writing to the SPI data register writes data into the transmit data register. Reading the SPI data register reads data from the receive data register. The transmit data and receive data registers are separate registers that can contain different values. (See Figure 13-2.)

Address: $0012

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R7	R6	R5	R4	R3	R2	R1	R0
Write:	T7	T6	T5	T4	T3	T2	T1	T0
Reset:	Unaffected by reset							

Figure 13-15. SPI Data Register (SPDR)

R7–R0/T7–T0 — Receive/Transmit Data Bits

NOTE

Do not use read-modify-write instructions on the SPI data register since the register read is not the same as the register written.

MMRW — MMIIC Master Read/Write

This bit is 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

MMCRCEF — MMIIC CRC Error Flag

This flag is set when a CRC error is detected, and cleared when no CRC error is detected. The MMCRCEF is only meaningful after receiving a PEC data. This flag is unaffected by reset.

1 = CRC error detected on PEC byte
0 = No CRC error detected on PEC byte

14.6.4 MMIIC Status Register (MMSR)

Address: $004B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MMRXIF	MMTXIF	MMATCH	MMSRW	MMRXAK	MMCRCBF	MMTXBE	MMRXBF
Write:	0	0						
Reset:	0	0	0	0	1	0	1	0

= Unimplemented

Figure 14-7. MMIIC Status Register (MMSR)

MMRXIF — MMIIC Receive Interrupt Flag

This flag is set after the data receive register (MMDRR) is loaded with a new received data. Once the MMDRR is loaded with received data, no more received data can be loaded to the MMDRR register until the CPU reads the data from the MMDRR to clear MMRXBF flag. MMRXIF 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; or when the MMEN = 0.

1 = New data in data receive register (MMDRR)
0 = No data received

MMTXIF — MMIIC Transmit Interrupt Flag

This flag is set when data in the data transmit register (MMDTR) is downloaded to the output circuit, and that new data can be written to the MMDTR. MMTXIF 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 when the MMEN = 0.

1 = Data transfer completed
0 = Data transfer in progress

MMATCH — MMIIC Address Match Flag

This flag is set when the received data in the data receive register (MMDRR) is a calling address which matches with the address or its extended addresses (MMEXTAD = 1) specified in the address register (MMADR). The MMATCH flag is set at the 9th clock of the calling address and will be cleared on the 9th clock of the next receiving data. Note: slave transmits do not clear MMATCH.

1 = Received address matches MMADR
0 = Received address does not match

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: 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: 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)

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

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

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

To determine the logic level on a keyboard interrupt pin, use the data direction register to configure the pin as an input and read the data register.

NOTE

Setting a keyboard interrupt enable bit (KBIEx) forces the corresponding keyboard interrupt pin to be an input, overriding the data direction register. However, the data direction register bit must be a logic 0 for software to read the pin.

18.4.1 Keyboard Initialization

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

To prevent a false interrupt on keyboard initialization:

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

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

Another way to avoid a false interrupt:

1. Configure the keyboard pins as outputs by setting the appropriate DDR bits in data direction register.
2. Write logic 1s to the appropriate data register bits.
3. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.

18.5 Keyboard Interrupt Registers

Two registers control the operation of the keyboard interrupt module:

- Keyboard Status and Control Register — $001A
- Keyboard Interrupt Enable Register — $001B

18.5.1 Keyboard Status and Control Register

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

20.3.4 Forced Reset Operation

In applications that require V_{DD} to remain above the V_{TRIPF1} level, enabling LVI resets allows the LVI module to reset the MCU when V_{DD} falls below the V_{TRIPF1} level. In the CONFIG1 register, the LVIPWRD and LVIRSTD bits must be at logic 0 to enable the LVI module and to enable LVI resets.

20.3.5 Voltage Hysteresis Protection

Once the LVI has triggered (by having V_{DD} fall below V_{TRIPF1}), the LVI will maintain a reset condition until V_{DD} rises above the rising trip point voltage, V_{TRIPR1}. This prevents a condition in which the MCU is continually entering and exiting reset if V_{DD} is approximately equal to V_{TRIPF1}. V_{TRIPR1} is greater than V_{TRIPF1} by the hysteresis voltage, V_{HYS}.

20.4 LVI Status Register

The LVI status register (LVISR) indicates if the V_{DD} voltage was detected below V_{TRIPF1} or V_{REG} voltage was detected below V_{TRIPF2}.

Address: $FE0F

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

= Unimplemented

Figure 20-3. LVI Status Register

LVIOUT — LVI Output Bit

This read-only flag becomes set when the V_{DD} or V_{REG} falls below their respective trip voltages. Reset clears the LVIOUT bit.

Table 20-1. LVIOUT Bit Indication

V_{DD}, V_{REG}	LVIOUT
$V_{DD} > V_{TRIPR1}$ and $V_{REG} > V_{TRIPR2}$	0
$V_{DD} < V_{TRIPF1}$ or $V_{DD} < V_{TRIPF2}$	1
$V_{TRIPF1} < V_{DD} < V_{TRIPR1}$ or $V_{TRIPF2} < V_{REG} < V_{TRIPR2}$	Previous value

20.5 LVI Interrupts

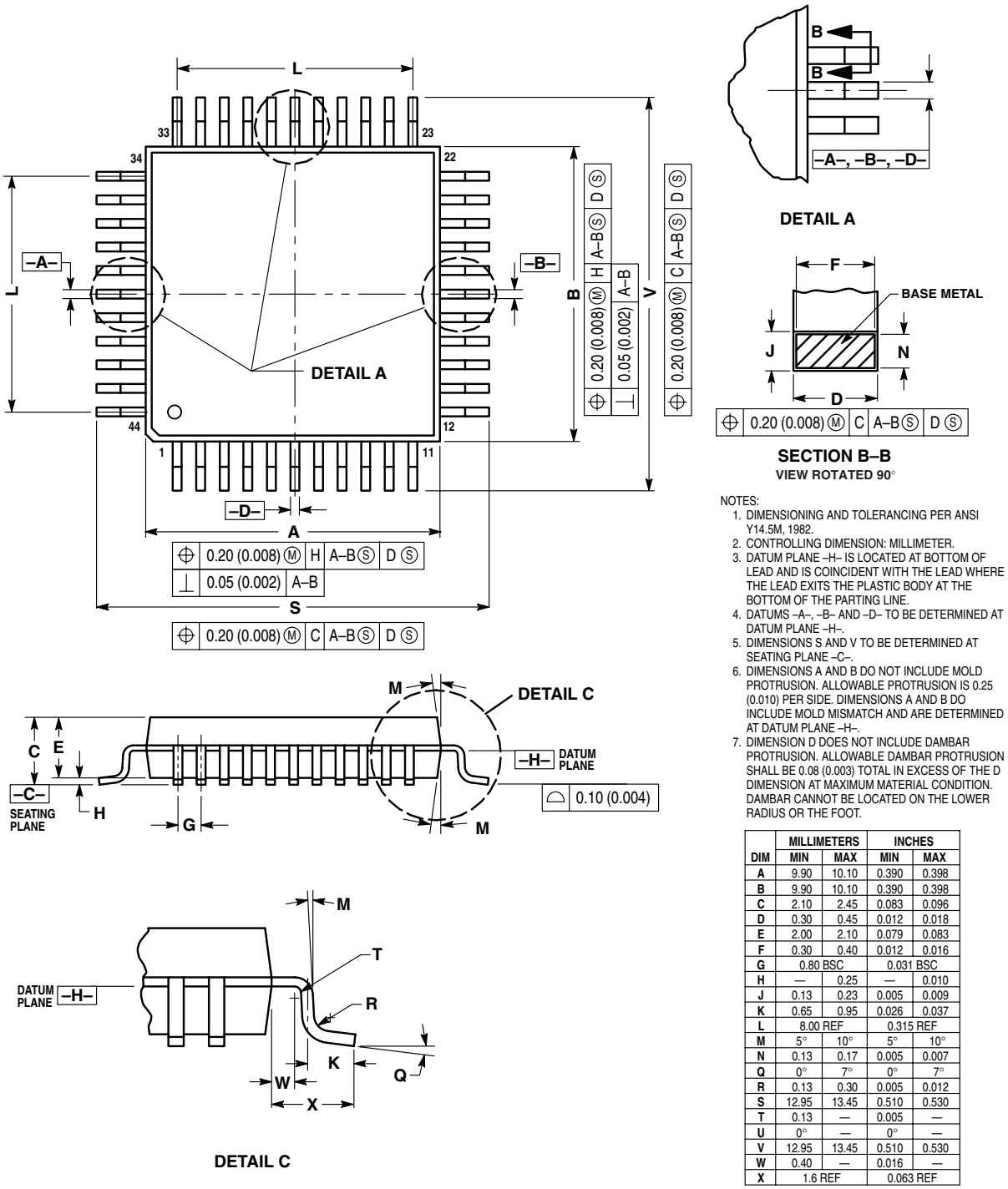
The LVI module does not generate interrupt requests.

22.5 5V DC Electrical Characteristics

Table 22-4. DC Electrical Characteristics (5V)

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage (I_{LOAD} = −12mA) PTA[0:7], PTB[4:7], PTC[0:5], PTD[0:7] (I_{LOAD} = −15mA) PTA[0:7], PTB[4:7], PTC[0:5], PTD[0:7]	 V_{OH} V_{OH}	 V_{DD}−0.8 V_{DD}−1.0	 — —	 — —	 V V
Output low voltage (I_{LOAD} = 6mA) PTA[0:7], PTB[4:7], PTC[0:5], PTD[0:7] (I_{LOAD} = 12mA) PTB[0:3], PTC[6:7] (I_{LOAD} = 15mA) PTA[0:7], PTB[4:7], PTC[0:5], PTD[0:7] (I_{LOAD} = 15mA) PTB[0:3], PTC[6:7] (I_{LOAD} = 15mA) as TxD, RxD, SCTxD, SCRxD (I_{LOAD} = see Table 22-8) as SDA, SCL	 V_{OL} V_{OL} V_{OL} V_{OL} V_{OLSCI} V_{OLIIC}	 — — — — — —	 — — — — — —	 0.4 0.4 0.8 0.6 0.4 0.4	 V V V V V V
LED sink current (V_{OL} = 3V) PTA[0:7]	I_{OL}	9	15	25	mA
Input high voltage PTA[0:7], PTB[0:7], PTC[0:7], PTD[0:7], $\overline{RST}$, $\overline{IRQ1}$ OSC1	V_{IH}	0.7 × V_{DD} 0.7 × V_{REG}	— —	V_{DD} V_{REG}	V V
Input low voltage PTA[0:7], PTB[0:7], PTC[0:7], PTD[0:7], $\overline{RST}$, $\overline{IRQ1}$ OSC1	V_{IL}	V_{SS} V_{SS}	— —	0.3 × V_{DD} 0.3 × V_{REG}	V V
V_{DD} supply current, f_{OP} = 8 MHz Run[3] Wait[4] Stop with OSC, TBM, and LVI modules on[5] with OSC and TBM modules on[5] all modules off[6]	 I_{DD}	 — — — — —	 10 2.5 1.8 1 20	 20 10 2.5 1.5 125	 mA mA mA mA µA
Digital I/O ports Hi-Z leakage current	I_{IL}	—	—	± 10	µA
Input current	I_{IN}	—	—	± 1	µA
Capacitance Ports (as input or output)	C_{OUT} C_{IN}	— —	— —	12 8	pF pF
POR rearm voltage[7]	V_{POR}	0	—	100	mV
POR rise time ramp rate[8]	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{TST}	1.4 × V_{DD}		8.5	V
Pullup resistors[9] PTD[0:7] RST, $\overline{IRQ1}$, $\overline{IRQ2}$	 R_{PU1} R_{PU2}	 21 21	 27 27	 39 39	 kΩ kΩ
Low-voltage inhibit, trip falling voltage1	V_{TRIPF1}	2.90	3.10	3.30	V
Low-voltage inhibit, trip rising voltage1	V_{TRIPR1}	3.25	3.45	3.65	V

23.3 44-Pin Quad Flat Pack (QFP)



DETAIL A

SECTION B–B
VIEW ROTATED 90°

DETAIL C

NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE –H– IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS –A–, –B– AND –D– TO BE DETERMINED AT DATUM PLANE –H–.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE –C–.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE –H–.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	9.90	10.10	0.390	0.398
B	9.90	10.10	0.390	0.398
C	2.10	2.45	0.083	0.096
D	0.30	0.45	0.012	0.018
E	2.00	2.10	0.079	0.083
F	0.30	0.40	0.012	0.016
G	0.80 BSC		0.031 BSC	
H	—	0.25	—	0.010
J	0.13	0.23	0.005	0.009
K	0.65	0.95	0.026	0.037
L	8.00 REF		0.315 REF	
M	5°	10°	5°	10°
N	0.13	0.17	0.005	0.007
Q	0°	7°	0°	7°
R	0.13	0.30	0.005	0.012
S	12.95	13.45	0.510	0.530
T	0.13	—	0.005	—
U	0°	—	0°	—
V	12.95	13.45	0.510	0.530
W	0.40	—	0.016	—
X	1.6 REF		0.063 REF	

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

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

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

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

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

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

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

MC68HC908AP64A
Rev. 3, 10/2007

freescale™
semiconductor