

Welcome to [E-XFL.COM](https://www.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	SCI
Peripherals	LVD, POR, PWM
Number of I/O	13
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	192 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-SOIC (0.295", 7.50mm Width)
Supplier Device Package	16-SOIC
Purchase URL	<a href="https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mc908kx2cdwe">https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mc908kx2cdwe</a>



# **MC68HC908KX8**

# **MC68HC908KX2**

# **MC68HC08KX8**

## **Data Sheet**

---

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

<http://www.freescale.com>

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc.  
This product incorporates SuperFlash® technology licensed from SST.

© Freescale Semiconductor, Inc., 2005. All rights reserved.

## 2.4 Random-Access Memory (RAM)

Addresses \$0040–\$00FF are RAM locations. The location of the stack RAM is programmable. The 16-bit stack pointer allows the stack to be anywhere in the 64-Kbyte memory space.

### NOTE

*For correct operation, the stack pointer must point only to RAM locations.*

Before processing an interrupt, the CPU uses five bytes of the stack to save the contents of the CPU registers.

### NOTE

*For M6805, M146805 and M68HC05 compatibility, the H register is not stacked.*

During a subroutine call, the CPU uses two bytes of the stack to store the return address. The stack pointer decrements during pushes and increments during pulls.

### NOTE

*Be careful when using nested subroutines. The CPU could overwrite data in the RAM during a subroutine or during the interrupt stacking operation.*

## 2.5 FLASH Memory (FLASH)

The FLASH memory is an array of 7,680 bytes with an additional 36 bytes of user vectors and one byte used for block protection.

### NOTE

*An erased bit reads as 1 and a programmed bit reads as 0.*

The program and erase operations are facilitated through control bits in the FLASH control register (FLCR). See 2.6 FLASH Control Register.

The FLASH is organized internally as an 8192-word by 8-bit complementary metal-oxide semiconductor (CMOS) page erase, byte (8-bit) program embedded FLASH memory. Each page consists of 64 bytes. The page erase operation erases all words within a page. A page is composed of two adjacent rows.


A security feature prevents viewing of the FLASH contents.<sup>(1)</sup>

## 2.6 FLASH Control Register

The FLASH control register (FLCR) controls FLASH program and erase operations.

Address: \$FE08

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
Write:								
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

**Figure 2-3. FLASH Control Register (FLCR)**

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

# Chapter 4

## Configuration Register (CONFIG)

### 4.1 Introduction

This section describes the configuration registers, CONFIG1 and CONFIG2. The configuration registers control these options:

- Stop mode recovery time, 32 CGMXCLK cycles or 4096 CGMXCLK cycles
- Computer operating properly (COP) timeout period,  $2^{18}$ – $2^4$  or  $2^{13}$ – $2^4$  CGMXCLK cycles
- STOP instruction
- Computer operating properly (COP) module
- Low-voltage inhibit (LVI) module control and voltage trip point selection
- Enable/disable the oscillator (OSC) during stop mode
- Serial communications interface (SCI) clock source selection
- External clock/crystal source control
- Enable/disable for the FLASH charge-pump regulator

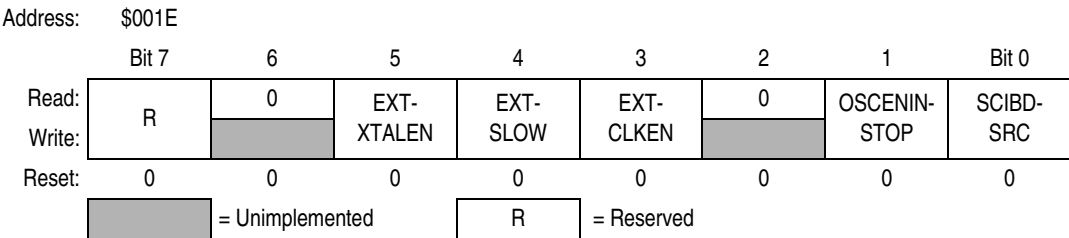
### 4.2 Functional Description

The configuration registers are used in the initialization of various options and can be written once after each reset. All of the configuration register bits are cleared during reset. Since the various options affect the operation of the microcontroller unit (MCU), it is recommended that these registers be written immediately after reset. The configuration registers are located at \$001E and \$001F. For compatibility, a write to a read-only memory (ROM) version of the MCU at this location will have no effect. The configuration register may be read at anytime.

#### NOTE

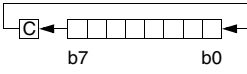
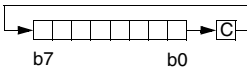
*The CONFIG module is known as an MOR (mask option register) on a ROM device. On a ROM device, the options are fixed at the time of device fabrication and are neither writable nor changeable by the user.*

*On a FLASH device, the CONFIG registers are special registers containing one-time writable latches after each reset. Upon a reset, the CONFIG registers default to predetermined settings as shown in Figure 4-1 and Figure 4-2.*

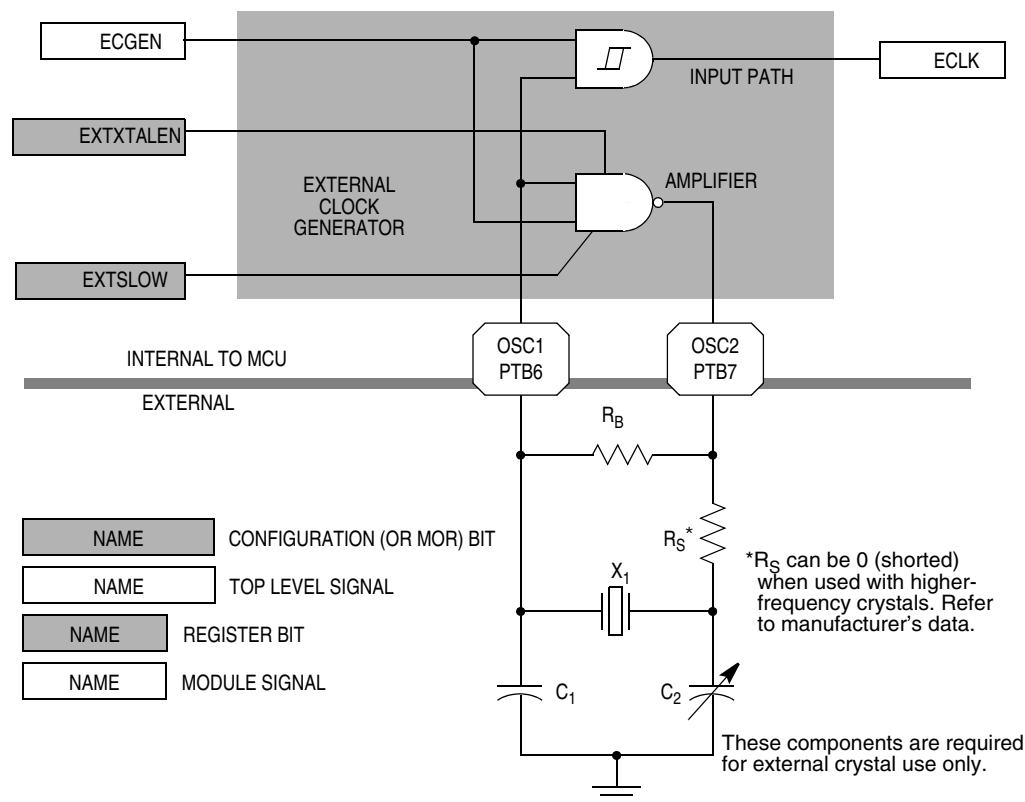


**Figure 4-1. Configuration Register 2 (CONFIG2)**

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

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z				
PULA	Pull A from Stack	$SP \leftarrow (SP + 1); \text{Pull (A)}$	–	–	–	–	–	INH	86		2
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		↑	–	–	–	↑	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		↑	–	–	–	↑	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	$SP \leftarrow (SP + 1); \text{Pull (CCR)}$ $SP \leftarrow (SP + 1); \text{Pull (A)}$ $SP \leftarrow (SP + 1); \text{Pull (X)}$ $SP \leftarrow (SP + 1); \text{Pull (PCH)}$ $SP \leftarrow (SP + 1); \text{Pull (PCL)}$	↑	↑	↑	↑	↑	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	$A \leftarrow (A) - (M) - (C)$	↑	–	–	–	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A2 B2 C2 D2 E2 F2 9EE2 9ED2	ii dd hh ll ee ff 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	–	–	–	↑	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh ll ee ff ff ff ff ee ff	3 4 4 3 2 4 5
STHX <i>opr</i>	Store H:X in M	$(M:M + 1) \leftarrow (H:X)$	0	–	–	–	↑	DIR	35	dd	4
STOP	Enable Interrupts, Stop Processing, Refer to MCU Documentation	$I \leftarrow 0$ ; Stop Processing	–	–	0	–	–	INH	8E		1
STX <i>opr</i> STX <i>opr</i> STX <i>opr</i> ,X STX <i>opr</i> ,X STX ,X STX <i>opr</i> ,SP STX <i>opr</i> ,SP	Store X in M	$M \leftarrow (X)$	0	–	–	–	↑	DIR EXT IX2 IX1 IX SP1 SP2	BF CF DF EF FF 9EEF 9EDF	dd hh ll ee ff ff ff ff ee ff	3 4 4 3 2 4 5
SUB # <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> ,X SUB <i>opr</i> ,X SUB ,X SUB <i>opr</i> ,SP SUB <i>opr</i> ,SP	Subtract	$A \leftarrow (A) - (M)$	↑	–	–	–	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A0 B0 C0 D0 E0 F0 9EE0 9ED0	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5

## Internal Clock Generator Module (ICG)



**Figure 7-3. External Clock Generator Block Diagram**

The amplifier is enabled when the external clock generator enable (ECGEN) signal is set and when the external crystal enable (EXTXTALEN) bit in the CONFIG (or MOR) register is set. ECGEN is controlled by the clock enable circuit (see 7.3.1 Clock Enable Circuit), and indicates that the external clock function is desired. When enabled, the amplifier will be connected between the PTB6/(OSC1) and PTB7/(OSC2)/RST pins. Otherwise, the PTB7/(OSC2)/RST pin reverts to its port function. In its typical configuration, the external oscillator requires five external components:

- Crystal,  $X_1$
- Fixed capacitor,  $C_1$
- Tuning capacitor,  $C_2$  (can also be a fixed capacitor)
- Feedback resistor,  $R_B$
- Series resistor,  $R_S$  (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.)

### 7.3.3.2 External Clock Input Path

The external clock input path is the means by which the microcontroller uses an external clock source. The input to the path is the PTB6/(OSC1) pin and the output is the external clock (ECLK). The path, which contains input buffering, is enabled when the external clock generator enable signal (ECGEN) is set. When not enabled, the PTB6/(OSC1) pin reverts to its port function.

The internal clock activity detector, shown in Figure 7-5, looks for at least one falling edge on the low-frequency base clock (IBASE) every time the external reference (EREF) is low. Since EREF is less than half the frequency of IBASE, this should occur every time. If it does not occur two consecutive times, the internal clock inactivity indicator (IOFF) is set. IOFF will be cleared the next time there is a falling edge of IBASE while EREF is low.

[illegible]

#### 7.3.4.3 External Clock Activity Detector

The external clock activity detector, shown in Figure 7-6, looks for at least one falling edge on the external clock (ECLK) every time the internal reference (IREF) is low. Since IREF is less than half the frequency of ECLK, this should occur every time. If it does not occur two consecutive times, the external clock inactivity indicator (EOFF) is set. EOFF will be cleared the next time there is a falling edge of ECLK while IREF is low.

The external clock stable bit (ECGS) is also generated in the external clock activity detector. ECGS is set on a falling edge of the external stabilization clock (ESTBCLK). This will be 4096 ECLK cycles after the external clock generator on bit is set or the MCU exits STOP (ECGEN = 1) if the external crystal enable (EXTXTALEN) in the CONFIG (or MOR) register is set, or 16 cycles when EXTXTALEN is clear. ECGS is cleared when the external clock generator is turned off or in STOP (ECGEN is clear) or when EOFF is set.



## Internal Clock Generator Module (ICG)

### ICGON — Internal Clock Generator On Bit

This read/write bit enables the internal clock generator. ICGON can be cleared when the CS bit has been set and the CMON bit has been clear for at least one bus cycle. ICGON is forced set when the CMON bit is set, the CS bit is clear, or during reset.

- 1 = Internal clock generator enabled
- 0 = Internal clock generator disabled

### ICGS — Internal Clock Generator Stable Bit

This read-only bit indicates when the internal clock generator has determined that the internal clock (ICLK) is within about 15% of the desired value. This bit is forced clear when the clock monitor determines the ICLK is inactive, when ICGON is clear, when the ICG multiplier register (ICGMR) is written, when the ICG trim register (ICGTR) is written, during STOP with OSCENINSTOP low, or during reset.

- 1 = Internal clock is within 15% of the desired value
- 0 = Internal clock may not be within 15% of the desired value

### ECGON — External Clock Generator On Bit

This read/write bit enables the external clock generator. ECGON can be cleared when the CS and CMON bits have been clear for at least one bus cycle. ECGON is forced set when the CMON bit or the CS bit is set. ECGON is forced clear during reset.

- 1 = External clock generator enabled
- 0 = External clock generator disabled

### ECGS — External Clock Generator Stable Bit


This read-only bit indicates when at least 4096 external clock (ECLK) cycles have elapsed since the external clock generator was enabled. This is not an assurance of the stability of ECLK but is meant to provide a start-up delay. This bit is forced clear when the clock monitor determines ECLK is inactive, when ECGON is clear, during STOP with OSCENINSTOP low, or during reset.

- 1 = 4096 ECLK cycles have elapsed since ECGON was set
- 0 = External clock is unstable, inactive, or disabled

## 7.7.2 ICG Multiplier Register

Address: \$0037

	Bit 7	6	5	4	3	2	1	Bit 0
Read:		N6	N5	N4	N3	N2	N1	N0
Write:								
Reset:	0	0	0	1	0	1	0	1

 = Unimplemented

**Figure 7-12. ICG Multiplier Register (ICGMR)**


### N6:N0 — ICG Multiplier Factor Bits

These read/write bits change the multiplier used by the internal clock generator. The internal clock (ICLK) will be  $(307.2 \text{ kHz} \pm 25\%) \times N$ . A value of \$00 in this register is interpreted the same as a value of \$01. This register cannot be written when the CMON bit is set. Reset sets this factor to \$15 (decimal 21) for default frequency of  $6.45 \text{ MHz} \pm 25\%$  ( $1.613 \text{ MHz} \pm 25\%$  bus).

### 7.7.3 ICG Trim Register

Address: \$0038

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
Write:								
Reset:	1	0	0	0	0	0	0	0

 = Unimplemented

**Figure 7-13. ICG Trim Register (ICGTR)**


#### TRIM7:TRIM0 — ICG Trim Factor Bits

These read/write bits change the size of the internal capacitor used by the internal clock generator. By testing the frequency of the internal clock and incrementing or decrementing this factor accordingly, the accuracy of the internal clock can be improved to  $\pm 2\%$ . Incrementing this register by one decreases the frequency by 0.195% of the unadjusted value. Decrementing this register by one increases the frequency by 0.195%. This register cannot be written when the CMON bit is set. Reset sets these bits to \$80, centering the range of possible adjustment.

### 7.7.4 ICG DCO Divider Register

Address: \$0039

	Bit 7	6	5	4	3	2	1	Bit 0
Read:					DDIV3	DDIV2	DDIV1	DDIV0
Write:								
Reset:	0	0	0	0	U	U	U	U

 = Unimplemented      U = Undefined

**Figure 7-14. ICG DCO Divider Control Register (ICGDVR)**


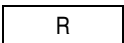
#### DDIV3:DDIV0 — ICG DCO Divider Control Bits

These bits indicate the number of divide-by-twos (DDIV) that follow the digitally controlled oscillator. When ICGON is set, DDIV is controlled by the digital loop filter. The range of valid values for DDIV is from \$0 to \$9. Values of \$A–\$F are interpreted the same as \$9. Since the DCO is active during reset, reset has no effect on DSTG and the value may vary.

### 7.7.5 ICG DCO Stage Register

Address: \$003A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	DSTG7	DSTG6	DSTG5	DSTG4	DSTG3	DSTG2	DSTG1	DSTG0
Write:	R	R	R	R	R	R	R	R
Reset:	U	U	U	U	U	U	U	U

 = Unimplemented       = Reserved      U = Unaffected

**Figure 7-15. ICG DCO Stage Control Register (ICGDSR)**

## External Interrupt (IRQ)

The vector fetch or software clear and the return of the  $\overline{\text{IRQ1}}$  pin to logic 1 can occur in any order. The interrupt request remains pending as long as the  $\overline{\text{IRQ1}}$  pin is at logic 0. A reset will clear the latch and the MODE1 control bit, thereby clearing the interrupt even if the pin stays low.

If the MODE1 bit is clear, the  $\overline{\text{IRQ1}}$  pin is falling-edge sensitive only. With MODE1 clear, a vector fetch or software clear immediately clears the IRQ1 latch.

The IRQF1 bit in the ISCR can be used to check for pending interrupts. The IRQF1 bit is not affected by the IMASK1 bit, which makes it useful in applications where polling is preferred.

Use the branch if interrupt pin is high (BIH) or branch if interrupt pin is low (BIL) instruction to read the logic level on the  $\overline{\text{IRQ1}}$  pin.

### NOTE

*When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine.*

## 8.5 IRQ Status and Control Register

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

- Shows the state of the IRQ1 interrupt flag
- Clears the IRQ1 interrupt latch
- Masks IRQ1 interrupt request
- Controls triggering sensitivity of the  $\overline{\text{IRQ1}}$  interrupt pin

Address: \$001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	IRQF1	0	IMASK1	MODE1
Write:	R	R	R	R	R	ACK1		
Reset:	0	0	0	0	U	0	0	0

R = Reserved      U = Unaffected

**Figure 8-3. IRQ Status and Control Register (ISCR)**

### IRQF1 — IRQ1 Flag Bit

This read-only status bit is high when the IRQ1 interrupt is pending.

1 = IRQ1 interrupt pending

0 = IRQ1 interrupt not pending

### ACK1 — IRQ1 Interrupt Request Acknowledge Bit

Writing a 1 to this write-only bit clears the IRQ1 latch. ACK1 always reads as 0. Reset clears ACK1.

### IMASK1 — IRQ1 Interrupt Mask Bit

Writing a 1 to this read/write bit disables IRQ1 interrupt requests. Reset clears IMASK1.

1 = IRQ1 interrupt requests disabled

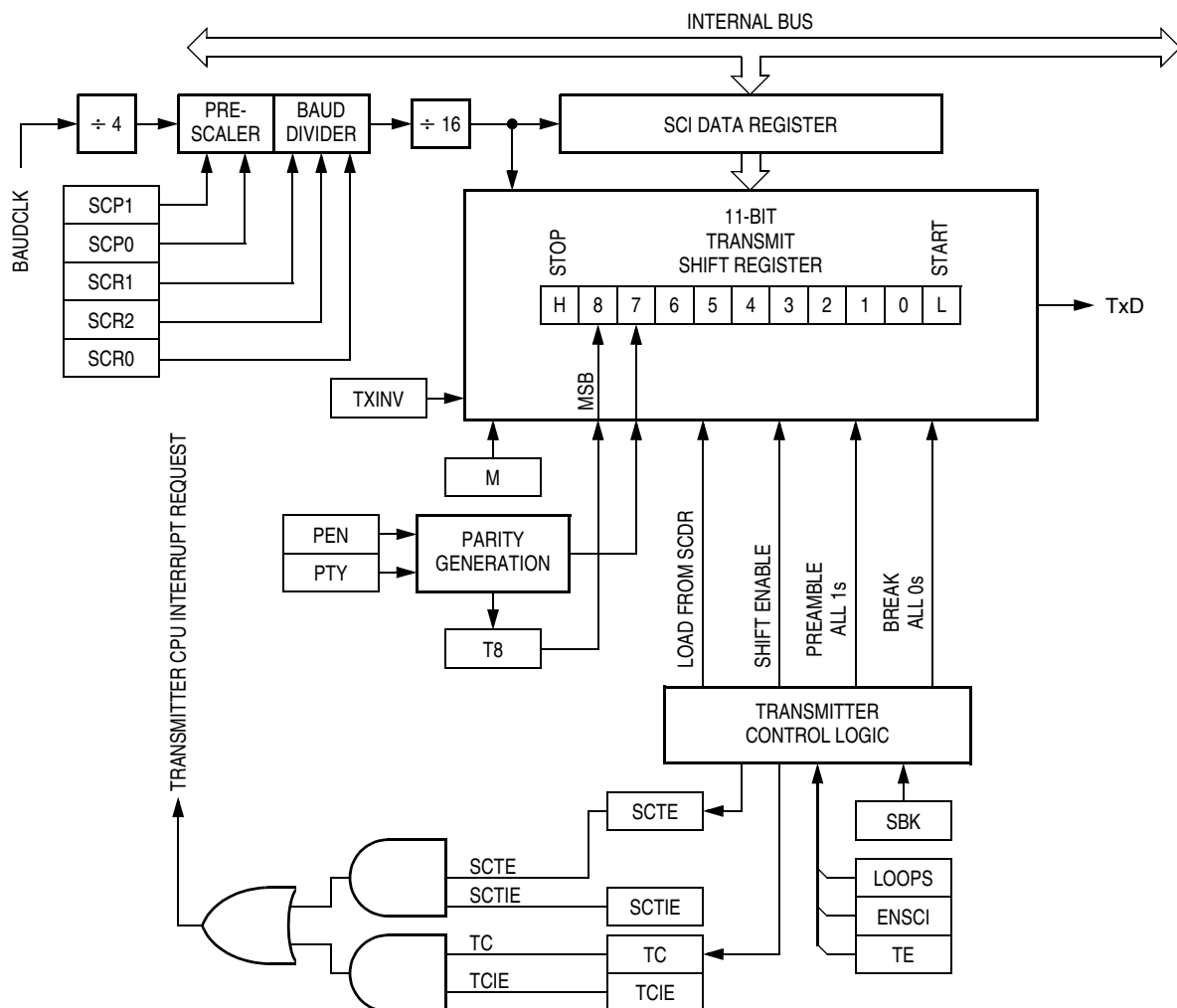
0 = IRQ1 interrupt requests enabled

### MODE1 — IRQ1 Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the  $\overline{\text{IRQ1}}$  pin. Reset clears MODE1.

1 = IRQ1 interrupt requests on falling edges and low levels

0 = IRQ1 interrupt requests on falling edges only



**Figure 12-5. SCI Transmitter Break Characters**

At the start of a transmission, transmitter control logic automatically loads the transmit shift register with a preamble of logic 1s. After the preamble shifts out, control logic transfers the SCDR data into the transmit shift register. A logic 0 start bit automatically goes into the least significant bit position of the transmit shift register. A logic 1 stop bit goes into the most significant bit position.

The SCI transmitter empty bit, SCTE, in SCS1 becomes set when the SCDR transfers a byte to the transmit shift register. The SCTE bit indicates that the SCDR can accept new data from the internal data bus. If the SCI transmit interrupt enable bit, SCTIE, in SCC2 is also set, the SCTE bit generates a transmitter CPU interrupt request.

When the transmit shift register is not transmitting a character, the TxD pin goes to the idle condition, logic 1. If at any time software clears the ENSCI bit in SCI control register 1 (SCC1), the transmitter and receiver relinquish control of the port B pins.

Writing a 1 to the send break bit, SBK, in SCC2 loads the transmit shift register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on the M bit in SCC1. As long as SBK is at 1, transmitter logic continuously loads break characters into the transmit shift register. After software clears the SBK bit, the shift register finishes transmitting the last

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7.

Table 12-2 summarizes the results of the start bit verification samples.

**Table 12-2. Start Bit Verification**

RT3, RT5, and RT7 Samples	Start Bit Verification	Noise Flag
000	Yes	0
001	Yes	1
010	Yes	1
011	No	0
100	Yes	1
101	No	0
110	No	0
111	No	0

If start bit verification is not successful, the RT clock is reset and a new search for a start bit begins.

To determine the value of a data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 12-3 summarizes the results of the data bit samples.

**Table 12-3. Data Bit Recovery**

RT8, RT9, and RT10 Samples	Data Bit Determination	Noise Flag
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

## NOTE

*The RT8, RT9, and RT10 samples do not affect start bit verification. If any or all of the RT8, RT9, and RT10 start bit samples are 1s following a successful start bit verification, the noise flag (NF) is set and the receiver assumes that the bit is a start bit.*

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

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

#### 12.4.3.6 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:

1. Address mark — An address mark is a 1 in the most significant bit position of a received character. When the WAKE bit is set, an address mark wakes the receiver from the standby state by clearing the RWU bit. The address mark also sets the SCI receiver full bit, SCRF. Software can then compare the character containing the address mark to the user-defined address of the receiver. If they are the same, the receiver remains awake and processes the characters that follow. If they are not the same, software can set the RWU bit and put the receiver back into the standby state.
2. Idle input line condition — When the WAKE bit is clear, an idle character on the RxD pin wakes the receiver from the standby state by clearing the RWU bit. The idle character that wakes the receiver does not set the receiver idle bit, IDLE, or the SCI receiver full bit, SCRF. The idle line type bit, ILTY, determines whether the receiver begins counting 1s as idle character bits after the start bit or after the stop bit.

#### NOTE

*With the WAKE bit clear, setting the RWU bit after the RxD pin has been idle may cause the receiver to wake up immediately.*

#### 12.4.3.7 Receiver Interrupts

These sources can generate CPU interrupt requests from the SCI receiver:

- SCI receiver full (SCRF) — The SCRF bit in SCS1 indicates that the receive shift register has transferred a character to the SCDR. SCRF can generate a receiver CPU interrupt request. Setting the SCI receive interrupt enable bit, SCRIE, in SCC2 enables the SCRF bit to generate receiver CPU interrupts.
- Idle input (IDLE) — The IDLE bit in SCS1 indicates that 10 or 11 consecutive 1s shifted in from the RxD pin. The idle line interrupt enable bit, ILIE, in SCC2 enables the IDLE bit to generate CPU interrupt requests.

#### 12.4.3.8 Error Interrupts

These receiver error flags in SCS1 can generate CPU interrupt requests:

- Receiver overrun (OR) — The OR bit indicates that the receive shift register shifted in a new character before the previous character was read from the SCDR. The previous character remains in the SCDR, and the new character is lost. The overrun interrupt enable bit, ORIE, in SCC3 enables OR to generate SCI error CPU interrupt requests.

## SCRIE — SCI Receive Interrupt Enable Bit

This read/write bit enables the SCRF bit to generate SCI receiver CPU interrupt requests. Setting the SCRIE bit in SCC3 enables the SCRF bit to generate 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

## TE — Transmitter Enable Bit

Setting this read/write bit begins the transmission by sending a preamble of 10 or 11 1s from the transmit shift register to the TxD pin. If software clears the TE bit, the transmitter completes any transmission in progress before the TxD returns to the idle condition (logic 1). Clearing and then setting TE during a transmission queues an idle character to be sent after the character currently being transmitted. Reset clears the TE bit.

- 1 = Transmitter enabled
- 0 = Transmitter disabled

### NOTE

*Writing to the TE bit is not allowed when the enable SCI bit (ENSCI) is clear. ENSCI is in SCI control register 1.*

## RE — Receiver Enable Bit

Setting this read/write bit enables the receiver. Clearing the RE bit disables the receiver but does not affect receiver interrupt flag bits. Reset clears the RE bit.

- 1 = Receiver enabled
- 0 = Receiver disabled

### NOTE

*Writing to the RE bit is not allowed when the enable SCI bit (ENSCI) is clear. ENSCI is in SCI control register 1.*

## RWU — Receiver Wakeup Bit

This read/write bit puts the receiver in a standby state during which receiver interrupts are disabled. The WAKE bit in SCC1 determines whether an idle input or an address mark brings the receiver out of the standby state and clears the RWU bit. Reset clears the RWU bit.

- 1 = Standby state
- 0 = Normal operation

## SBK — Send Break Bit

Setting and then clearing this read/write bit transmits a break character followed by a 1. The 1 after the break character guarantees recognition of a valid start bit. If SBK remains set, the transmitter continuously transmits break characters with no 1s between them. Reset clears the SBK bit.

- 1 = Transmit break characters
- 0 = No break characters being transmitted

### NOTE

*Do not toggle the SBK bit immediately after setting the SCTE bit. Toggling SBK before the preamble begins causes the SCI to send a break character instead of a preamble.*



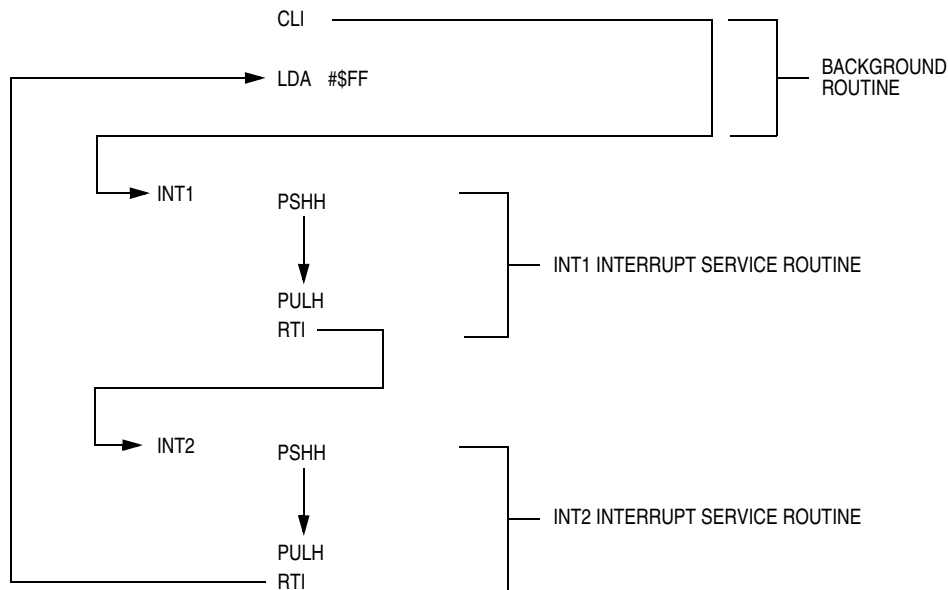


is pending upon exit from the original interrupt service routine, the pending interrupt is serviced before the load-accumulator- from-memory (LDA) instruction is executed.

The LDA opcode is prefetched by both the INT1 and INT2 RTI instructions. However, in the case of the INT1 RTI prefetch, this is a redundant operation.

**NOTE**

*To maintain compatibility with the M68HC05, M6805, and M146805 Families the H register is not pushed on the stack during interrupt entry. If the interrupt service routine modifies the H register or uses the indexed addressing mode, software should save the H register and then restore it prior to exiting the routine.*



**Figure 13-10. Interrupt Recognition Example**

### 13.5.1.2 SWI Instruction

The SWI instruction is a non-maskable instruction that causes an interrupt regardless of the state of the interrupt mask (I bit) in the condition code register.

**NOTE**

*A software interrupt pushes PC onto the stack. A software interrupt does **not** push PC – 1, as a hardware interrupt does.*

### 13.5.2 Reset

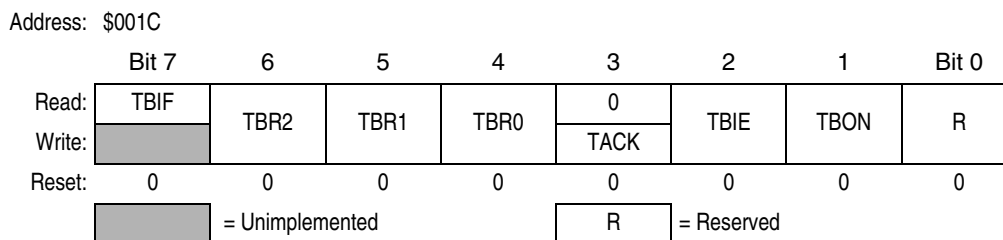
All reset sources always have higher priority than interrupts and cannot be arbitrated.

## 13.6 Low-Power Modes

Executing the WAIT or STOP instruction puts the MCU in a low power- consumption mode for standby situations. The SIM holds the CPU in a non-clocked state. Both STOP and WAIT clear the interrupt mask (I) in the condition code register, allowing interrupts to occur. Low-power modes are exited via an interrupt or reset.

## 14.7 Timebase Control Register

The timebase has one register, the timebase control register (TBCR), which is used to enable the timebase interrupts and set the rate.



**Figure 14-2. Timebase Control Register (TBCR)**

### TBIF — Timebase Interrupt Flag

This read-only flag bit is set when the timebase counter has rolled over.

1 = Timebase interrupt pending

0 = Timebase interrupt not pending

### TBR2–TBR0 — Timebase Divider Selection Bits

These read/write bits select the tap in the counter to be used for timebase interrupts as shown in Table 14-1.

#### NOTE

*Do not change TBR2–TBR0 bits while the timebase is enabled (TBON = 1).*

### TACK— Timebase ACKnowledge Bit

The TACK bit is a write-only bit and always reads as 0. Writing a 1 to this bit clears TBIF, the timebase interrupt flag bit. Writing a 0 to this bit has no effect.

1 = Clear timebase interrupt flag

0 = No effect

### TBIE — Timebase Interrupt Enabled Bit

This read/write bit enables the timebase interrupt when the TBIF bit becomes set. Reset clears the TBIE bit.

1 = Timebase interrupt is enabled.

0 = Timebase interrupt is disabled.

### TBON — Timebase Enabled Bit

This read/write bit enables the timebase. Timebase may be turned off to reduce power consumption when its function is not necessary. The counter can be initialized by clearing and then setting this bit. Reset clears the TBON bit.

1 = Timebase is enabled.

0 = Timebase is disabled and the counter initialized to 0s.

5. In the TIM status control register (TSC), clear the TIM stop bit, TSTOP.

Setting MS0B links channels 0 and 1 and configures them for buffered PWM operation. The TIM channel 0 registers (TCH0H and TCH0L) initially control the buffered PWM output. TIM status control register 0 (TSCR0) controls and monitors the PWM signal from the linked channels.

Clearing the toggle-on-overflow bit, TOVx, inhibits output toggles on TIM overflows. Subsequent output compares try to force the output to a state it is already in and have no effect. The result is a 0 percent duty cycle output.

Setting the channel x maximum duty cycle bit (CHxMAX) and setting the TOVx bit generates a 100 percent duty cycle output. See 15.8.4 TIM Channel Status and Control Registers.

## 15.5 Interrupts

These TIM sources can generate interrupt requests:

- TIM overflow flag (TOF) — The timer overflow flag (TOF) bit is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. The TIM overflow interrupt enable bit, TOIE, enables TIM overflow interrupt requests. TOF and TOIE are in the TIM status and control registers.
- TIM channel flags (CH1F and CH0F) — The CHxF bit is set when an input capture or output compare occurs on channel x. Channel x TIM CPU interrupt requests are controlled by the channel x interrupt enable bit, CHxIE. Channel x TIM CPU interrupt requests are enabled when CHxIE = 1. CHxF and CHxIE are in the TIM channel x status and control register.

## 15.6 Low-Power Modes

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

### 15.6.1 Wait Mode

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

If TIM functions are not required during wait mode, reduce power consumption by stopping the TIM before executing the WAIT instruction.

### 15.6.2 Stop Mode

The TIM is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions or the state of the TIM counter. TIM operation resumes when the MCU exits stop mode after an external interrupt.

## 15.7 I/O Signals

Port A shares two of its pins with the TIM, PTA3/ $\overline{\text{KBD3}}$ /TCH1 and PTA2/ $\overline{\text{KBD2}}$ /TCH0. Each channel input/output (I/O) pin is programmable independently as an input capture pin or an output compare pin. TCH0 can be configured as buffered output compare or buffered PWM pins.

## TRST — TIM Reset Bit

Setting this write-only bit resets the TIM counter and the TIM prescaler. Setting TRST has no effect on any other registers. Counting resumes from \$0000. TRST is cleared automatically after the TIM counter is reset and always reads as 0. Reset clears the TRST bit.

- 1 = Prescaler and TIM counter cleared
- 0 = No effect

### NOTE

*Setting the TSTOP and TRST bits simultaneously stops the TIM counter at a value of \$0000.*

## PS2–PS0 — Prescaler Select Bits

These read/write bits select one of the seven prescaler outputs as the input to the TIM counter as Table 15-2 shows. Reset clears the PS2–PS0 bits.

**Table 15-2. Prescaler Selection**

PS2–PS0	TIM Clock Source
000	Internal bus clock ÷ 1
001	Internal bus clock ÷ 2
010	Internal bus clock ÷ 4
011	Internal bus clock ÷ 8
100	Internal bus clock ÷ 16
101	Internal bus clock ÷ 32
110	Internal bus clock ÷ 64
111	Not available

## 15.8.2 TIM Counter Registers

The two read-only TIM counter registers (TCNTH and TCNTL) contain the high and low bytes of the value in the TIM counter. Reading the high byte (TCNTH) latches the contents of the low byte (TCNTL) into a buffer. Subsequent reads of TCNTH do not affect the latched TCNTL value until TCNTL is read. Reset clears the TIM counter registers. Setting the TIM reset bit (TRST) also clears the TIM counter registers.

Register name and address: TCNTH — \$0021

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

Register name and address: TCNTL — \$0022

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

= Unimplemented

**Figure 15-6. TIM Counter Registers (TCNTH and TCNTL)**

## ***How to Reach Us:***

### **Home Page:**

[www.freescale.com](http://www.freescale.com)

### **E-mail:**

[support@freescale.com](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto:LDCForFreescaleSemiconductor@hibbertgroup.com)

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2005. All rights reserved.