



Welcome to [E-XFL.COM](http://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](#)"

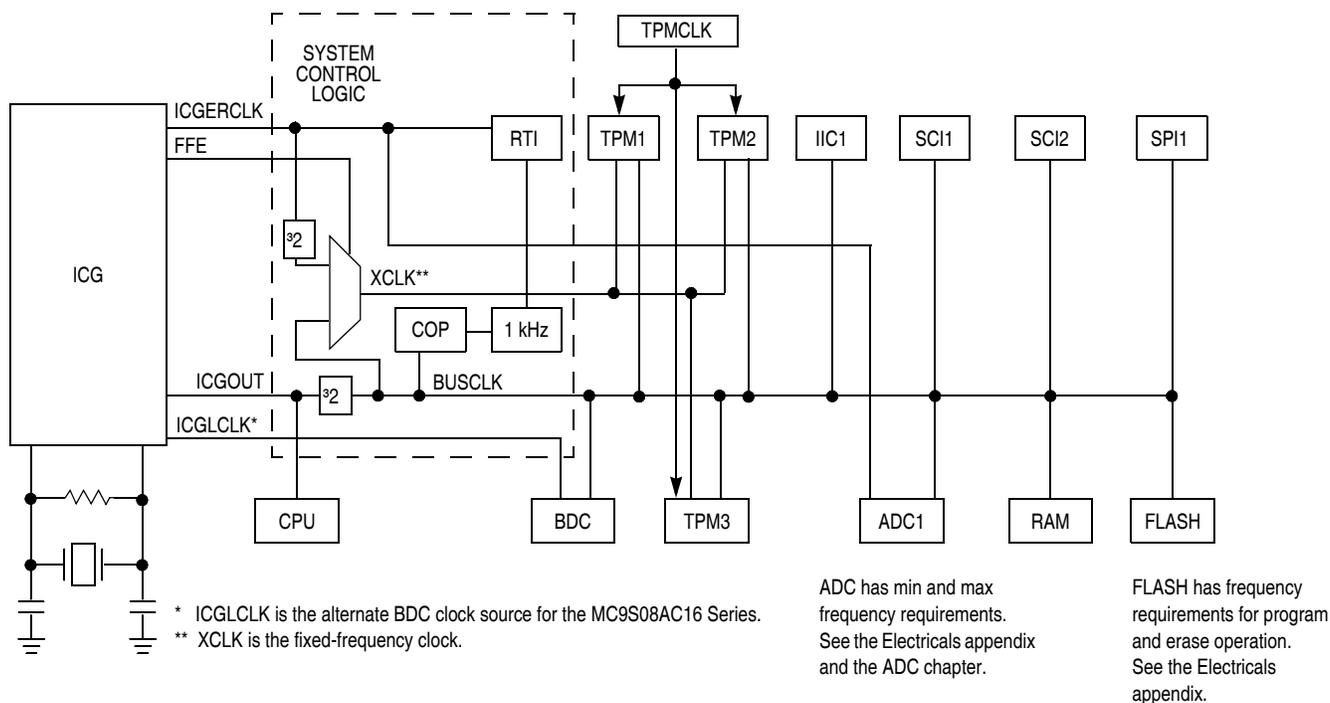
Details	
Product Status	Active
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I <sup>2</sup> C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mc9s08ac8mfge">https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mc9s08ac8mfge</a>

Table 1-2 lists the functional versions of the on-chip modules.

**Table 1-2. Versions of On-Chip Modules**

Module	Version
Analog-to-Digital Converter (ADC)	1
Internal Clock Generator (ICG)	4
Inter-Integrated Circuit (IIC)	2
Keyboard Interrupt (KBI)	1
Serial Communications Interface (SCI)	4
Serial Peripheral Interface (SPI)	3
Timer Pulse-Width Modulator (TPM)	3
Central Processing Unit (CPU)	2

### 1.3 System Clock Distribution



**Figure 1-2. System Clock Distribution Diagram**

Some of the modules inside the MCU have clock source choices. Figure 1-2 shows a simplified clock connection diagram. The ICG supplies the clock sources:

- ICGOUT is an output of the ICG module. It is one of the following:
  - The external crystal oscillator
  - An external clock source
  - The output of the digitally-controlled oscillator (DCO) in the frequency-locked loop sub-module

Figure 2-3 shows the 42-pin SDIP pin assignments for the MC9S08AC16 Series device.

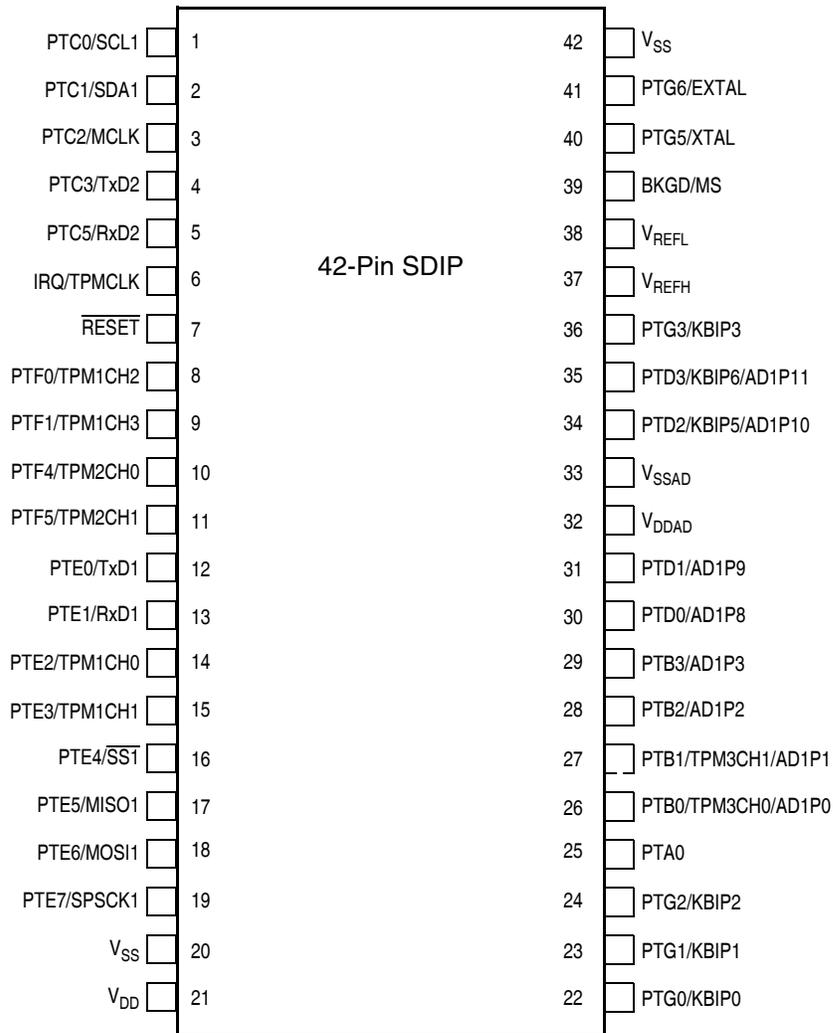


Figure 2-3. MC9S08AC16 Series in 42-Pin SDIP Package

**Table 4-2. Direct-Page Register Summary (Sheet 1 of 3)**

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x0000	PTAD	PTAD7	R	R	R	R	PTAD2	PTAD1	PTAD0
0x0001	PTADD	PTADD7	R	R	R	R	PTADD2	PTADD1	PTADD0
0x0002	PTBD	R	R	R	R	PTBD3	PTBD2	PTBD1	PTBD0
0x0003	PTBDD	R	R	R	R	PTBDD3	PTBDD2	PTBDD1	PTBDD0
0x0004	PTCD	0	R	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
0x0005	PTCDD	0	R	PTCDD5	PTCDD4	PTCDD3	PTCDD2	PTCDD1	PTCDD0
0x0006	PTDD	R	R	R	R	PTDD3	PTDD2	PTDD1	PTDD0
0x0007	PTDDD	R	R	R	R	PTDDD3	PTDDD2	PTDDD1	PTDDD0
0x0008	PTED	PTED7	PTED6	PTED5	PTED4	PTED3	PTED2	PTED1	PTED0
0x0009	PTEDD	PTEDD7	PTEDD6	PTEDD5	PTEDD4	PTEDD3	PTEDD2	PTEDD1	PTEDD0
0x000A	PTFD	R	PTFD6	PTFD5	PTFD4	R	R	PTFD1	PTFD0
0x000B	PTFDD	R	PTFDD6	PTFDD5	PTFDD4	R	R	PTFDD1	PTFDD0
0x000C	PTGD	0	PTGD6	PTGD5	PTGD4	PTGD3	PTGD2	PTGD1	PTGD0
0x000D	PTGDD	0	PTGDD6	PTGDD5	PTGDD4	PTGDD3	PTGDD2	PTGDD1	PTGDD0
0x000E– 0x000F	Reserved	—	—	—	—	—	—	—	—
0x0010	ADC1SC1	COCO	AIEN	ADCO	ADCH				
0x0011	ADC1SC2	ADACT	ADTRG	ACFE	ACFGT	0	0	R	R
0x0012	ADC1RH	0	0	0	0	0	0	ADR9	ADR8
0x0013	ADC1RL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
0x0014	ADC1CVH	0	0	0	0	0	0	ADCV9	ADCV8
0x0015	ADC1CVL	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
0x0016	ADC1CFG	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
0x0017	APCTL1	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
0x0018	APCTL2	ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8
0x0019	APCTL3	ADPC23	ADPC22	ADPC21	ADPC20	ADPC19	ADPC18	ADPC17	ADPC16
0x001A– 0x001B	Reserved	—	—	—	—	—	—	—	—
0x001C	IRQSC	0	IRQPDD	IRQEDG	IRQPE	IRQF	IRQACK	IRQIE	IRQMOD
0x001D	Reserved	—	—	—	—	—	—	—	—
0x001E	KBISC	0	KBEDG6	KBEDG5	KBEDG4	KBF	KBACK	KBIE	KBIMOD
0x001F	KBIPE	0	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
0x0020	TPM1SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0x0021	TPM1CNTH	Bit 15	14	13	12	11	10	9	Bit 8
0x0022	TPM1CNTL	Bit 7	6	5	4	3	2	1	Bit 0
0x0023	TPM1MODH	Bit 15	14	13	12	11	10	9	Bit 8
0x0024	TPM1MODL	Bit 7	6	5	4	3	2	1	Bit 0
0x0025	TPM1C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
0x0026	TPM1C0VH	Bit 15	14	13	12	11	10	9	Bit 8
0x0027	TPM1C0VL	Bit 7	6	5	4	3	2	1	Bit 0

## Chapter 5

# Resets, Interrupts, and System Configuration

### 5.1 Introduction

This chapter discusses basic reset and interrupt mechanisms and the various sources of reset and interrupts in the MC9S08AC16 Series. Some interrupt sources from peripheral modules are discussed in greater detail within other chapters of this data manual. This chapter gathers basic information about all reset and interrupt sources in one place for easy reference. A few reset and interrupt sources, including the computer operating properly (COP) watchdog and real-time interrupt (RTI), are not part of on-chip peripheral systems with their own sections but are part of the system control logic.

### 5.2 Features

Reset and interrupt features include:

- Multiple sources of reset for flexible system configuration and reliable operation:
  - Power-on detection (POR)
  - Low voltage detection (LVD) with enable
  - External  $\overline{\text{RESET}}$  pin
  - COP watchdog with enable and two timeout choices
  - Illegal opcode
  - Illegal address
  - Serial command from a background debug host
- Reset status register (SRS) to indicate source of most recent reset
- Separate interrupt vectors for each module (reduces polling overhead) (see Table 5-11)

### 5.3 MCU Reset

Resetting the MCU provides a way to start processing from a known set of initial conditions. During reset, most control and status registers are forced to initial values and the program counter is loaded from the reset vector (0xFFFF:0xFFFF). On-chip peripheral modules are disabled and I/O pins are initially configured as general-purpose high-impedance inputs with pullup devices disabled. The I bit in the condition code register (CCR) is set to block maskable interrupts so the user program has a chance to initialize the stack pointer (SP) and system control settings. SP is forced to 0x00FF at reset.

The following sources of reset are available on the MC9S08AC16 Series:

- Power-on reset (POR)
- Low-voltage detect (LVD)

# Chapter 7

## Central Processor Unit (S08CPUV2)

### 7.1 Introduction

This section provides summary information about the registers, addressing modes, and instruction set of the CPU of the HCS08 Family. For a more detailed discussion, refer to the *HCS08 Family Reference Manual, volume 1*.

The HCS08 CPU is fully source- and object-code-compatible with the M68HC08 CPU. Several instructions and enhanced addressing modes were added to improve C compiler efficiency and to support a new background debug system which replaces the monitor mode of earlier M68HC08 microcontrollers (MCU).

#### 7.1.1 Features

Features of the HCS08 CPU include:

- Object code fully upward-compatible with M68HC05 and M68HC08 Families
- All registers and memory are mapped to a single 64-Kbyte address space
- 16-bit stack pointer (any size stack anywhere in 64-Kbyte address space)
- 16-bit index register (H:X) with powerful indexed addressing modes
- 8-bit accumulator (A)
- Many instructions treat X as a second general-purpose 8-bit register
- Seven addressing modes:
  - Inherent — Operands in internal registers
  - Relative — 8-bit signed offset to branch destination
  - Immediate — Operand in next object code byte(s)
  - Direct — Operand in memory at 0x0000–0x00FF
  - Extended — Operand anywhere in 64-Kbyte address space
  - Indexed relative to H:X — Five submodes including auto increment
  - Indexed relative to SP — Improves C efficiency dramatically
- Memory-to-memory data move instructions with four address mode combinations
- Overflow, half-carry, negative, zero, and carry condition codes support conditional branching on the results of signed, unsigned, and binary-coded decimal (BCD) operations
- Efficient bit manipulation instructions
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- STOP and WAIT instructions to invoke low-power operating modes

## 7.4.1 Reset Sequence

Reset can be caused by a power-on-reset (POR) event, internal conditions such as the COP (computer operating properly) watchdog, or by assertion of an external active-low reset pin. When a reset event occurs, the CPU immediately stops whatever it is doing (the MCU does not wait for an instruction boundary before responding to a reset event). For a more detailed discussion about how the MCU recognizes resets and determines the source, refer to the Resets, Interrupts, and System Configuration chapter.

The reset event is considered concluded when the sequence to determine whether the reset came from an internal source is done and when the reset pin is no longer asserted. At the conclusion of a reset event, the CPU performs a 6-cycle sequence to fetch the reset vector from 0xFFFFE and 0xFFFF and to fill the instruction queue in preparation for execution of the first program instruction.

## 7.4.2 Interrupt Sequence

When an interrupt is requested, the CPU completes the current instruction before responding to the interrupt. At this point, the program counter is pointing at the start of the next instruction, which is where the CPU should return after servicing the interrupt. The CPU responds to an interrupt by performing the same sequence of operations as for a software interrupt (SWI) instruction, except the address used for the vector fetch is determined by the highest priority interrupt that is pending when the interrupt sequence started.

The CPU sequence for an interrupt is:

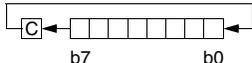
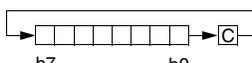
1. Store the contents of PCL, PCH, X, A, and CCR on the stack, in that order.
2. Set the I bit in the CCR.
3. Fetch the high-order half of the interrupt vector.
4. Fetch the low-order half of the interrupt vector.
5. Delay for one free bus cycle.
6. Fetch three bytes of program information starting at the address indicated by the interrupt vector to fill the instruction queue in preparation for execution of the first instruction in the interrupt service routine.

After the CCR contents are pushed onto the stack, the I bit in the CCR is set to prevent other interrupts while in the interrupt service routine. Although it is possible to clear the I bit with an instruction in the interrupt service routine, this would allow nesting of interrupts (which is not recommended because it leads to programs that are difficult to debug and maintain).

For compatibility with the earlier M68HC05 MCUs, the high-order half of the H:X index register pair (H) is not saved on the stack as part of the interrupt sequence. The user must use a PSHH instruction at the beginning of the service routine to save H and then use a PULH instruction just before the RTI that ends the interrupt service routine. It is not necessary to save H if you are certain that the interrupt service routine does not use any instructions or auto-increment addressing modes that might change the value of H.

The software interrupt (SWI) instruction is like a hardware interrupt except that it is not masked by the global I bit in the CCR and it is associated with an instruction opcode within the program so it is not asynchronous to program execution.

Table 7-2. . Instruction Set Summary (Sheet 6 of 9)

Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR	
						V I 1 H	I N Z C
MOV <i>opr8a,opr8a</i> MOV <i>opr8a,X+</i> MOV <i>#opr8i,opr8a</i> MOV <i>,X+,opr8a</i>	Move $(M)_{\text{destination}} \leftarrow (M)_{\text{source}}$ In IX+/DIR and DIR/IX+ Modes, $H:X \leftarrow (H:X) + \$0001$	DIR/DIR DIR/IX+ IMM/DIR IX+/DIR	4E dd dd 5E dd 6E ii dd 7E dd	5 5 4 5	rpwpp rfwpp pwpp rfwpp	0 1 1 -	- $\uparrow\downarrow$ -
MUL	Unsigned multiply $X:A \leftarrow (X) \times (A)$	INH	42	5	ffffp	- 1 1 0	- - - - 0
NEG <i>opr8a</i> NEGA NEGX NEG <i>opr8,X</i> NEG <i>,X</i> NEG <i>opr8,SP</i>	Negate $M \leftarrow -(M) = \$00 - (M)$ (Two's Complement) $A \leftarrow -(A) = \$00 - (A)$ $X \leftarrow -(X) = \$00 - (X)$ $M \leftarrow -(M) = \$00 - (M)$ $M \leftarrow -(M) = \$00 - (M)$ $M \leftarrow -(M) = \$00 - (M)$	DIR INH INH IX1 IX SP1	30 dd 40 50 60 ff 70 9E 60 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	$\uparrow$ 1 1 -	- $\uparrow\downarrow\downarrow$
NOP	No Operation — Uses 1 Bus Cycle	INH	9D	1	p	- 1 1 -	- - - - -
NSA	Nibble Swap Accumulator $A \leftarrow (A[3:0]:A[7:4])$	INH	62	1	p	- 1 1 -	- - - - -
ORA <i>#opr8i</i> ORA <i>opr8a</i> ORA <i>opr16a</i> ORA <i>opr8,X</i> ORA <i>opr8,X</i> ORA <i>,X</i> ORA <i>opr8,SP</i> ORA <i>opr16,SP</i> ORA <i>opr8,SP</i>	Inclusive OR Accumulator and Memory $A \leftarrow (A)   (M)$	IMM DIR EXT IX2 IX1 IX SP2 SP1	AA ii BA dd CA hh ll DA ee ff EA ff FA 9E DA ee ff 9E EA ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 -	- $\uparrow\downarrow$ -
PSHA	Push Accumulator onto Stack $\text{Push } (A); SP \leftarrow (SP) - \$0001$	INH	87	2	sp	- 1 1 -	- - - - -
PSHH	Push H (Index Register High) onto Stack $\text{Push } (H); SP \leftarrow (SP) - \$0001$	INH	8B	2	sp	- 1 1 -	- - - - -
PSHX	Push X (Index Register Low) onto Stack $\text{Push } (X); SP \leftarrow (SP) - \$0001$	INH	89	2	sp	- 1 1 -	- - - - -
PULA	Pull Accumulator from Stack $SP \leftarrow (SP + \$0001); \text{Pull } (A)$	INH	86	3	ufp	- 1 1 -	- - - - -
PULH	Pull H (Index Register High) from Stack $SP \leftarrow (SP + \$0001); \text{Pull } (H)$	INH	8A	3	ufp	- 1 1 -	- - - - -
PULX	Pull X (Index Register Low) from Stack $SP \leftarrow (SP + \$0001); \text{Pull } (X)$	INH	88	3	ufp	- 1 1 -	- - - - -
ROL <i>opr8a</i> ROLA ROLX ROL <i>opr8,X</i> ROL <i>,X</i> ROL <i>opr8,SP</i>	Rotate Left through Carry 	DIR INH INH IX1 IX SP1	39 dd 49 59 69 ff 79 9E 69 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	$\uparrow$ 1 1 -	- $\uparrow\downarrow\downarrow$
ROR <i>opr8a</i> RORA RORX ROR <i>opr8,X</i> ROR <i>,X</i> ROR <i>opr8,SP</i>	Rotate Right through Carry 	DIR INH INH IX1 IX SP1	36 dd 46 56 66 ff 76 9E 66 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	$\uparrow$ 1 1 -	- $\uparrow\downarrow\downarrow$

**Table 8-12. MFD and RFD Decode Table**

101	14	101	÷32
110	16	110	÷64
111	18	111	÷128

## 8.5.2 Example #1: External Crystal = 32 kHz, Bus Frequency = 4.19 MHz

In this example, the FLL will be used (in FEE mode) to multiply the external 32 kHz oscillator up to 8.38 MHz to achieve 4.19 MHz bus frequency.

After the MCU is released from reset, the ICG is in self-clocked mode (SCM) and supplies approximately 8 MHz on ICGOUT, which corresponds to a 4 MHz bus frequency ( $f_{\text{Bus}}$ ).

The clock scheme will be FLL engaged, external (FEE). So

$$f_{\text{ICGOUT}} = f_{\text{ext}} * P * N / R ; P = 64, f_{\text{ext}} = 32 \text{ kHz} \quad \text{Eqn. 8-1}$$

Solving for N / R gives:

$$N / R = 8.38 \text{ MHz} / (32 \text{ kHz} * 64) = 4 ; \text{ we can choose } N = 4 \text{ and } R = 1 \quad \text{Eqn. 8-2}$$

The values needed in each register to set up the desired operation are:

**ICGC1 = \$38 (%00111000)**

Bit 7	HGO	0	Configures oscillator for low power
Bit 6	RANGE	0	Configures oscillator for low-frequency range; FLL prescale factor is 64
Bit 5	REFS	1	Oscillator using crystal or resonator is requested
Bits 4:3	CLKS	11	FLL engaged, external reference clock mode
Bit 2	OSCSTEN	0	Oscillator disabled
Bit 1	LOCD	0	Loss-of-clock detection enabled
Bit 0		0	Unimplemented or reserved, always reads zero

**ICGC2 = \$00 (%00000000)**

Bit 7	LOLRE	0	Generates an interrupt request on loss of lock
Bits 6:4	MFD	000	Sets the MFD multiplication factor to 4
Bit 3	LOCRES	0	Generates an interrupt request on loss of clock
Bits 2:0	RFD	000	Sets the RFD division factor to ÷1

**ICGS1 = \$xx**

This is read only except for clearing interrupt flag

**ICGS2 = \$xx**

This is read only; should read DCOS = 1 before performing any time critical tasks

**ICGFLTLU/L = \$xx**

Only needed in self-clocked mode; FLT will be adjusted by loop to give 8.38 MHz DCO clock  
 Bits 15:12 unused 0000

# Chapter 10

## Timer/PWM (S08TPMV3)

### 10.1 Introduction

The MC9S08AC16 Series includes three independent timer/PWM (TPM) modules which support traditional input capture, output compare, or buffered edge-aligned pulse-width modulation (PWM) on each channel. A control bit in each TPM configures all channels in that timer to operate as center-aligned PWM functions. In each TPM, timing functions are based on a separate 16-bit counter with prescaler and modulo features to control frequency and range (period between overflows) of the time reference. This timing system is ideally suited for a wide range of control applications, and the center-aligned PWM capability on the TPM extends the field of applications to motor control in small appliances.

The use of the fixed system clock, XCLK, as the clock source for any of the TPM modules allows the TPM prescaler to run using the oscillator rate divided by two (ICGERCLK/2). This option is only available if the ICG is configured in FEE mode and the proper conditions are met (see Section 8.4.11, “Fixed Frequency Clock”). In all other ICG modes this selection is redundant because XCLK is the same as BUSCLK.

An external clock source can be connected to the TPMCLK pin. The maximum frequency for TPMCLK is the bus clock frequency divided by 4. All three TPM modules can independently select TPMCLK as the clock source.

### 10.2 Features

The timer system in the MC9S08AC16 Series includes a 4-channel TPM1, a separate 2-channel TPM2 and a separate 2-channel TPM3. Timer system features include:

- A total of up to eight channels:
  - Each channel may be input capture, output compare, or buffered edge-aligned PWM
  - Rising-edge, falling-edge, or any-edge input capture trigger
  - Set, clear, or toggle output compare action
  - Selectable polarity on PWM outputs
- Each TPM may be configured for buffered, center-aligned pulse-width modulation (CPWM) on all channels
- Clock source to prescaler for each TPM is independently selectable as bus clock, fixed system clock, or an external pin:
  - Prescale taps for divide by 1, 2, 4, 8, 16, 32, 64, or 128
  - External clock input: TPMCLK for use by TPM1, TPM2, and/or TPM3
- 16-bit free-running or up/down (CPWM) count operation
- 16-bit modulus register to control counter range
- Timer system enable
- One interrupt per channel plus a terminal count interrupt for each TPM module

**Table 11-7. SCIxC3 Field Descriptions (continued)**

Field	Description
4 TXINV <sup>1</sup>	<b>Transmit Data Inversion</b> — Setting this bit reverses the polarity of the transmitted data output. 0 Transmit data not inverted 1 Transmit data inverted
3 ORIE	<b>Overrun Interrupt Enable</b> — This bit enables the overrun flag (OR) to generate hardware interrupt requests. 0 OR interrupts disabled (use polling). 1 Hardware interrupt requested when OR = 1.
2 NEIE	<b>Noise Error Interrupt Enable</b> — This bit enables the noise flag (NF) to generate hardware interrupt requests. 0 NF interrupts disabled (use polling). 1 Hardware interrupt requested when NF = 1.
1 FEIE	<b>Framing Error Interrupt Enable</b> — This bit enables the framing error flag (FE) to generate hardware interrupt requests. 0 FE interrupts disabled (use polling). 1 Hardware interrupt requested when FE = 1.
0 PEIE	<b>Parity Error Interrupt Enable</b> — This bit enables the parity error flag (PF) to generate hardware interrupt requests. 0 PF interrupts disabled (use polling). 1 Hardware interrupt requested when PF = 1.

<sup>1</sup> Setting TXINV inverts the TxD output for all cases: data bits, start and stop bits, break, and idle.

## 11.2.7 SCI Data Register (SCIxD)

This register is actually two separate registers. Reads return the contents of the read-only receive data buffer and writes go to the write-only transmit data buffer. Reads and writes of this register are also involved in the automatic flag clearing mechanisms for the SCI status flags.

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

**Figure 11-11. SCI Data Register (SCIxD)**

## 11.3 Functional Description

The SCI allows full-duplex, asynchronous, NRZ serial communication among the MCU and remote devices, including other MCUs. The SCI comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. During normal operation, the MCU monitors the status of the SCI, writes the data to be transmitted, and processes received data. The following describes each of the blocks of the SCI.

### 11.3.1 Baud Rate Generation

As shown in Figure 11-12, the clock source for the SCI baud rate generator is the bus-rate clock.

Writing 0 to TE does not immediately release the pin to be a general-purpose I/O pin. Any transmit activity that is in progress must first be completed. This includes data characters in progress, queued idle characters, and queued break characters.

### 11.3.2.1 Send Break and Queued Idle

The SBK control bit in SCIxC2 is used to send break characters which were originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0 (10 bit times including the start and stop bits). A longer break of 13 bit times can be enabled by setting BRK13 = 1. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 1 and then write 0 to the SBK bit. This action queues a break character to be sent as soon as the shifter is available. If SBK is still 1 when the queued break moves into the shifter (synchronized to the baud rate clock), an additional break character is queued. If the receiving device is another SCI, the break characters will be received as 0s in all eight data bits and a framing error (FE = 1) occurs.

When idle-line wakeup is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 0 and then write 1 to the TE bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while TE = 0, the SCI transmitter never actually releases control of the TxD pin. If there is a possibility of the shifter finishing while TE = 0, set the general-purpose I/O controls so the pin that is shared with TxD is an output driving a logic 1. This ensures that the TxD line will look like a normal idle line even if the SCI loses control of the port pin between writing 0 and then 1 to TE.

The length of the break character is affected by the BRK13 and M bits as shown below.

**Table 11-8. Break Character Length**

BRK13	M	Break Character Length
0	0	10 bit times
0	1	11 bit times
1	0	13 bit times
1	1	14 bit times

### 11.3.3 Receiver Functional Description

In this section, the receiver block diagram (Figure 11-3) is used as a guide for the overall receiver functional description. Next, the data sampling technique used to reconstruct receiver data is described in more detail. Finally, two variations of the receiver wakeup function are explained.

The receiver input is inverted by setting RXINV = 1. The receiver is enabled by setting the RE bit in SCIxC2. Character frames consist of a start bit of logic 0, eight (or nine) data bits (LSB first), and a stop bit of logic 1. For information about 9-bit data mode, refer to Section 11.3.5.1, “8- and 9-Bit Data Modes.” For the remainder of this discussion, we assume the SCI is configured for normal 8-bit data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full, the data character is transferred to the receive data register and the receive data register full (RDRF)

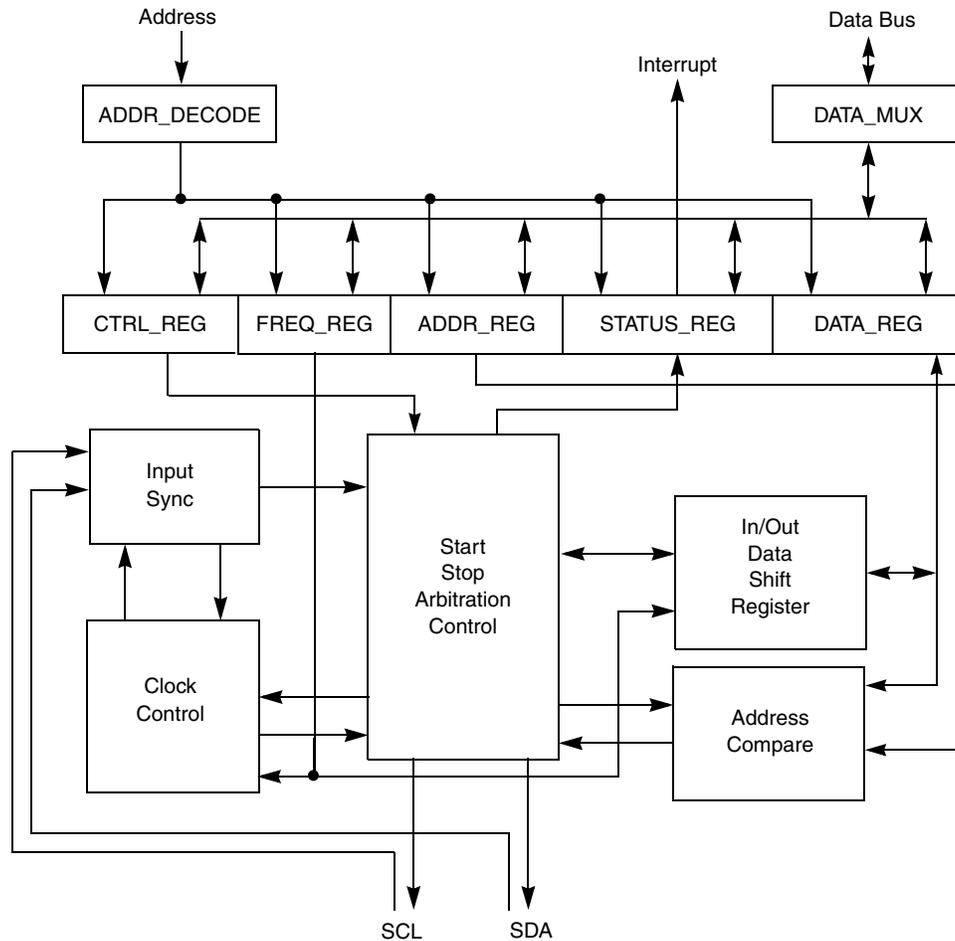


Figure 13-2. IIC Functional Block Diagram

## 13.2 External Signal Description

This section describes each user-accessible pin signal.

### 13.2.1 SCL — Serial Clock Line

The bidirectional SCL is the serial clock line of the IIC system.

### 13.2.2 SDA — Serial Data Line

The bidirectional SDA is the serial data line of the IIC system.

## 13.3 Register Definition

This section consists of the IIC register descriptions in address order.

Refer to the direct-page register summary in the memory chapter of this document for the absolute address assignments for all IIC registers. This section refers to registers and control bits only by their names. A

After a repeated start condition (Sr), all other slave devices also compare the first seven bits of the first byte of the slave address with their own addresses and test the eighth ( $\overline{R/\overline{W}}$ ) bit. However, none of them are addressed because  $\overline{R/\overline{W}} = 1$  (for 10-bit devices) or the 11110XX slave address (for 7-bit devices) does not match.

S	Slave Address 1st 7 bits 11110 + AD10 + AD9	R/W 0	A1	Slave Address 2nd byte AD[8:1]	A2	Sr	Slave Address 1st 7 bits 11110 + AD10 + AD9	R/W 1	A3	Data	A	...	Data	A	P
---	---	----------	----	--------------------------------------	----	----	---	----------	----	------	---	-----	------	---	---

**Table 13-10. Master-Receiver Addresses a Slave-Transmitter with a 10-bit Address**

After the master-receiver has sent the first byte of the 10-bit address, the slave-transmitter sees an IIC interrupt. Software must ensure the contents of IICD are ignored and not treated as valid data for this interrupt.

### 13.4.3 General Call Address

General calls can be requested in 7-bit address or 10-bit address. If the GCAEN bit is set, the IIC matches the general call address as well as its own slave address. When the IIC responds to a general call, it acts as a slave-receiver and the IAAS bit is set after the address cycle. Software must read the IICD register after the first byte transfer to determine whether the address matches its own slave address or a general call. If the value is 00, the match is a general call. If the GCAEN bit is clear, the IIC ignores any data supplied from a general call address by not issuing an acknowledgement.

## 13.5 Resets

The IIC is disabled after reset. The IIC cannot cause an MCU reset.

## 13.6 Interrupts

The IIC generates a single interrupt.

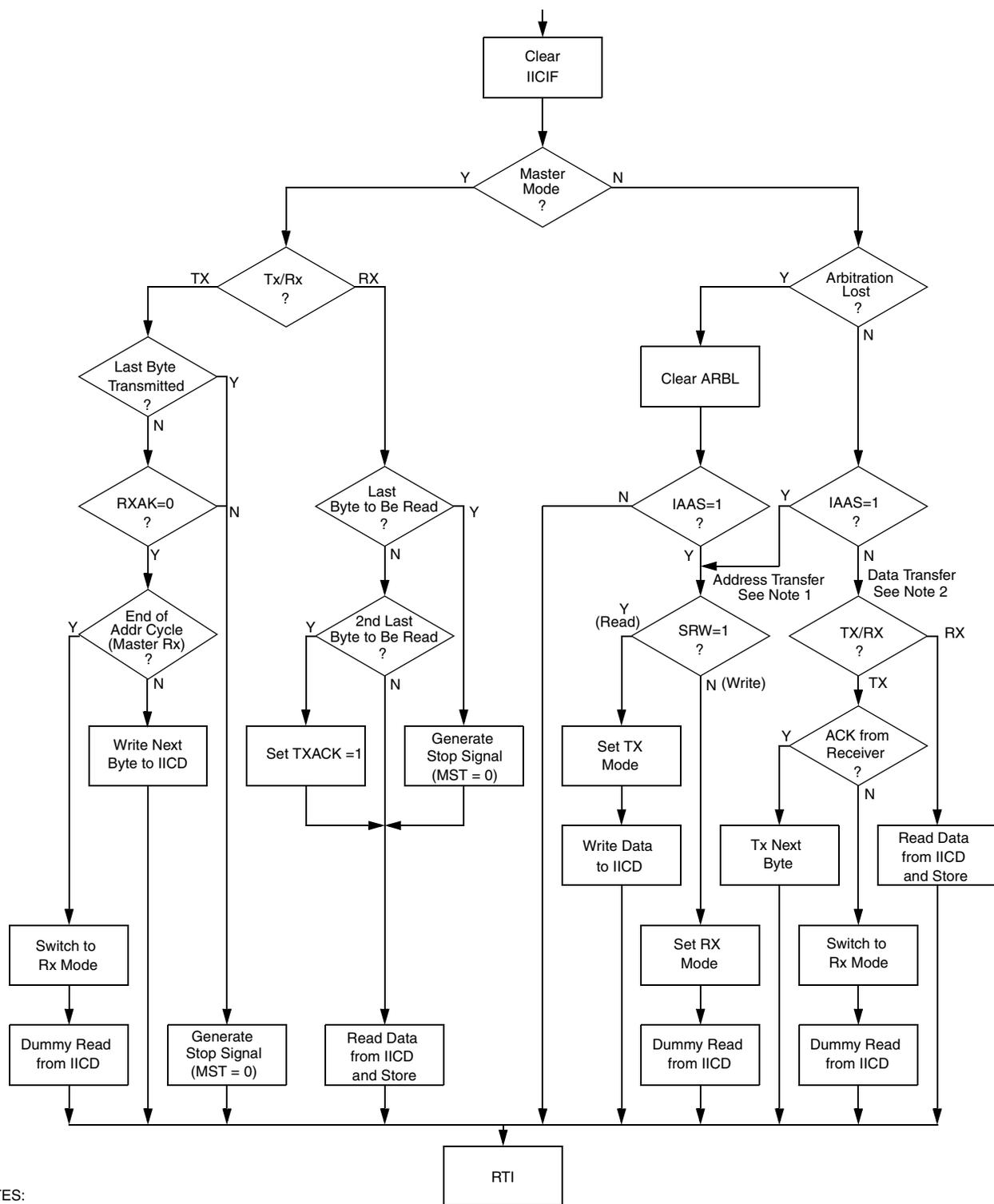
An interrupt from the IIC is generated when any of the events in Table 13-11 occur, provided the IICIE bit is set. The interrupt is driven by bit IICIF (of the IIC status register) and masked with bit IICIE (of the IIC control register). The IICIF bit must be cleared by software by writing a 1 to it in the interrupt routine. You can determine the interrupt type by reading the status register.

**Table 13-11. Interrupt Summary**

Interrupt Source	Status	Flag	Local Enable
Complete 1-byte transfer	TCF	IICIF	IICIE
Match of received calling address	IAAS	IICIF	IICIE
Arbitration Lost	ARBL	IICIF	IICIE

### 13.6.1 Byte Transfer Interrupt

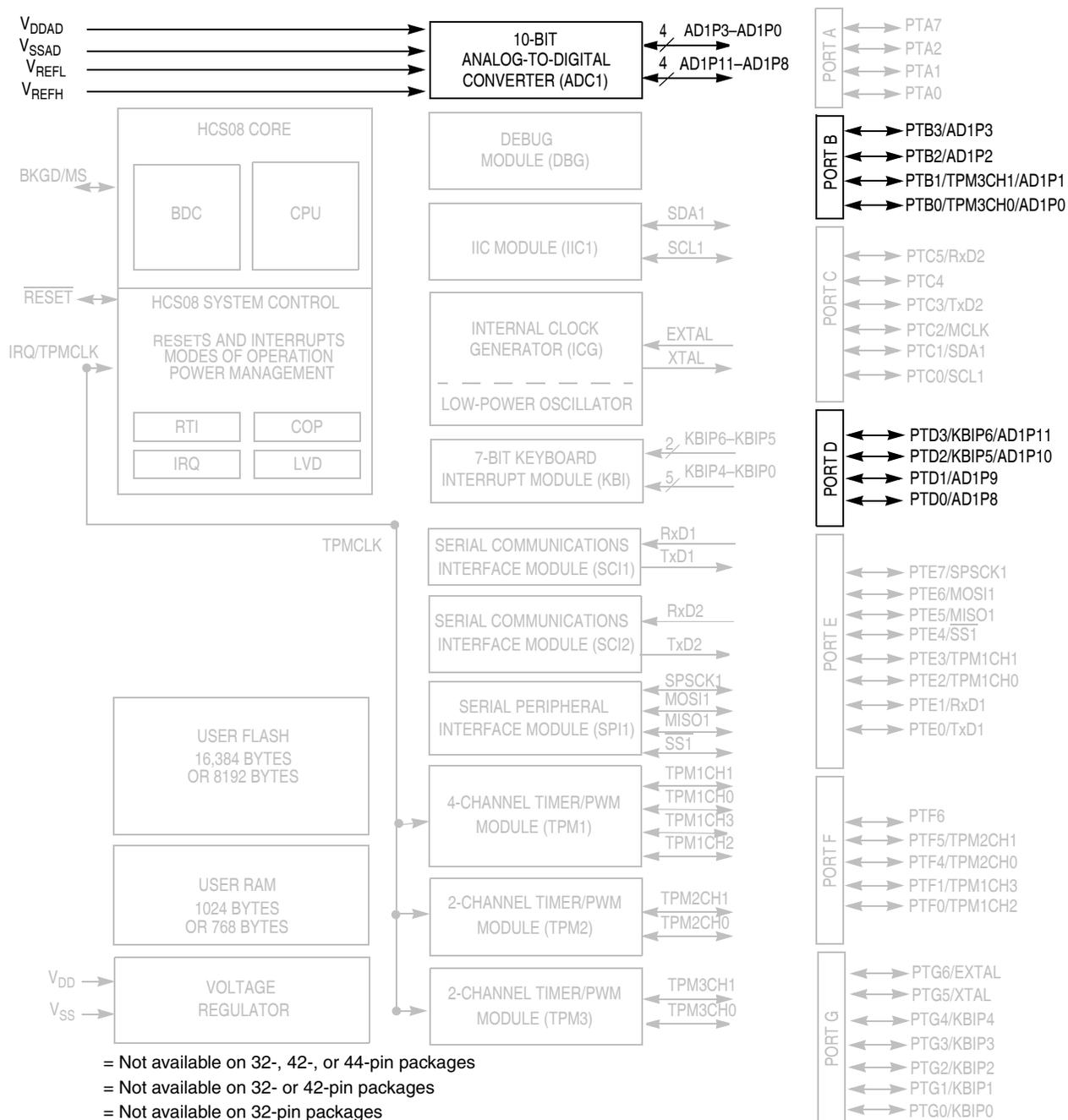
The TCF (transfer complete flag) bit is set at the falling edge of the ninth clock to indicate the completion of byte transfer.



NOTES:

1. If general call is enabled, a check must be done to determine whether the received address was a general call address (0x00). If the received address was a general call address, then the general call must be handled by user software.
2. When 10-bit addressing is used to address a slave, the slave sees an interrupt following the first byte of the extended address. User software must ensure that for this interrupt, the contents of IICD are ignored and not treated as a valid data transfer.

Figure 13-12. Typical IIC Interrupt Routine



**Notes:**

1. Port pins are software configurable with pullup device if input port.
2. Pin contains software configurable pullup/pulldown device if IRQ is enabled (IRQPE = 1). Pulldown is enabled if rising edge detect is selected (IRQEDG = 1)
3. IRQ does not have a clamp diode to  $V_{DD}$ . IRQ should not be driven above  $V_{DD}$ .
4. Pin contains integrated pullup device.
5. PTD3, PTD2, and PTG4 contain both pullup and pulldown devices. Pulldown enabled when KBI is enabled (KBIPEn = 1) and rising edge is selected (KBEDGn = 1).

**Figure 14-1. MC9S08AC16 Block Diagram Highlighting the ADC**

- Average the result by converting the analog input many times in succession and dividing the sum of the results. Four samples are required to eliminate the effect of a 1LSB, one-time error.
- Reduce the effect of synchronous noise by operating off the asynchronous clock (ADACK) and averaging. Noise that is synchronous to ADCK cannot be averaged out.

#### 14.7.2.4 Code Width and Quantization Error

The ADC quantizes the ideal straight-line transfer function into 1024 steps (in 10-bit mode). Each step ideally has the same height (1 code) and width. The width is defined as the delta between the transition points to one code and the next. The ideal code width for an N bit converter (in this case N can be 8 or 10), defined as 1LSB, is:

$$1\text{LSB} = (V_{\text{REFH}} - V_{\text{REFL}}) / 2^N \quad \text{Eqn. 14-2}$$

There is an inherent quantization error due to the digitization of the result. For 8-bit or 10-bit conversions the code will transition when the voltage is at the midpoint between the points where the straight line transfer function is exactly represented by the actual transfer function. Therefore, the quantization error will be  $\pm 1/2\text{LSB}$  in 8- or 10-bit mode. As a consequence, however, the code width of the first (\$000) conversion is only  $1/2\text{LSB}$  and the code width of the last (\$FF or \$3FF) is  $1.5\text{LSB}$ .

#### 14.7.2.5 Linearity Errors

The ADC may also exhibit non-linearity of several forms. Every effort has been made to reduce these errors but the system should be aware of them because they affect overall accuracy. These errors are:

- Zero-scale error ( $E_{\text{ZS}}$ ) (sometimes called offset) — This error is defined as the difference between the actual code width of the first conversion and the ideal code width ( $1/2\text{LSB}$ ). Note, if the first conversion is \$001, then the difference between the actual \$001 code width and its ideal ( $1\text{LSB}$ ) is used.
- Full-scale error ( $E_{\text{FS}}$ ) — This error is defined as the difference between the actual code width of the last conversion and the ideal code width ( $1.5\text{LSB}$ ). Note, if the last conversion is \$3FE, then the difference between the actual \$3FE code width and its ideal ( $1\text{LSB}$ ) is used.
- Differential non-linearity (DNL) — This error is defined as the worst-case difference between the actual code width and the ideal code width for all conversions.
- Integral non-linearity (INL) — This error is defined as the highest-value the (absolute value of the) running sum of DNL achieves. More simply, this is the worst-case difference of the actual transition voltage to a given code and its corresponding ideal transition voltage, for all codes.
- Total unadjusted error (TUE) — This error is defined as the difference between the actual transfer function and the ideal straight-line transfer function, and therefore includes all forms of error.

#### 14.7.2.6 Code Jitter, Non-Monotonicity and Missing Codes

Analog-to-digital converters are susceptible to three special forms of error. These are code jitter, non-monotonicity, and missing codes.

Code jitter is when, at certain points, a given input voltage converts to one of two values when sampled repeatedly. Ideally, when the input voltage is infinitesimally smaller than the transition voltage, the

**Table A-11. ICG Frequency Specifications (continued)**  
 ( $V_{DDA} = V_{DDA} \text{ (min) to } V_{DDA} \text{ (max)}$ , Temperature Range =  $-40$  to  $125^\circ\text{C}$  Ambient)

Num	C	Characteristic	Symbol	Min	Typ <sup>1</sup>	Max	Unit
18		<b>MC9S08ACxx:</b> Internal oscillator deviation from trimmed frequency <sup>9</sup> $V_{DD} = 2.7 - 5.5 \text{ V}$ , (constant temperature) $V_{DD} = 5.0 \text{ V} \pm 10\%$ , $-40^\circ\text{C}$ to $125^\circ\text{C}$	$ACC_{int}$	— —	$\pm 0.5$ $\pm 0.5$	$\pm 2$ $\pm 2$	%
	C	<b>S9S08AWxxA:</b> Internal oscillator deviation from trimmed frequency <sup>10</sup> $V_{DD} = 2.7 - 5.5 \text{ V}$ , (constant temperature)	$ACC_{int}$	—	$\pm 0.5$	$\pm 1.5$	%
	P	$V_{DD} = 5.0 \text{ V} \pm 10\%$ , $-40^\circ\text{C}$ to $85^\circ\text{C}$		—	$\pm 0.5$	$\pm 1.5$	
	P	$V_{DD} = 5.0 \text{ V} \pm 10\%$ , $-40^\circ\text{C}$ to $125^\circ\text{C}$		—	$\pm 0.5$	$\pm 2.0$	

<sup>1</sup> Typical values are based on characterization data at  $V_{DD} = 5.0\text{V}$ ,  $25^\circ\text{C}$  unless otherwise stated.

<sup>2</sup> Self-clocked mode frequency is the frequency that the DCO generates when the FLL is open-loop.

<sup>3</sup> Loss of reference frequency is the reference frequency detected internally, which transitions the ICG into self-clocked mode if it is not in the desired range.

<sup>4</sup> Loss of DCO frequency is the DCO frequency detected internally, which transitions the ICG into FLL bypassed external mode (if an external reference exists) if it is not in the desired range.

<sup>5</sup> This parameter is characterized before qualification rather than 100% tested.

<sup>6</sup> Proper PC board layout procedures must be followed to achieve specifications.

<sup>7</sup> This specification applies to the period of time required for the FLL to lock after entering FLL engaged internal or external modes. If a crystal/resonator is being used as the reference, this specification assumes it is already running.

<sup>8</sup> Jitter is the average deviation from the programmed frequency measured over the specified interval at maximum  $f_{ICGOUT}$ . Measurements are made with the device powered by filtered supplies and clocked by a stable external clock signal. Noise injected into the FLL circuitry via  $V_{DDA}$  and  $V_{SSA}$  and variation in crystal oscillator frequency increase the  $C_{Jitter}$  percentage for a given interval.

<sup>9</sup> See Figure A-9.

<sup>10</sup> See Figure A-9.

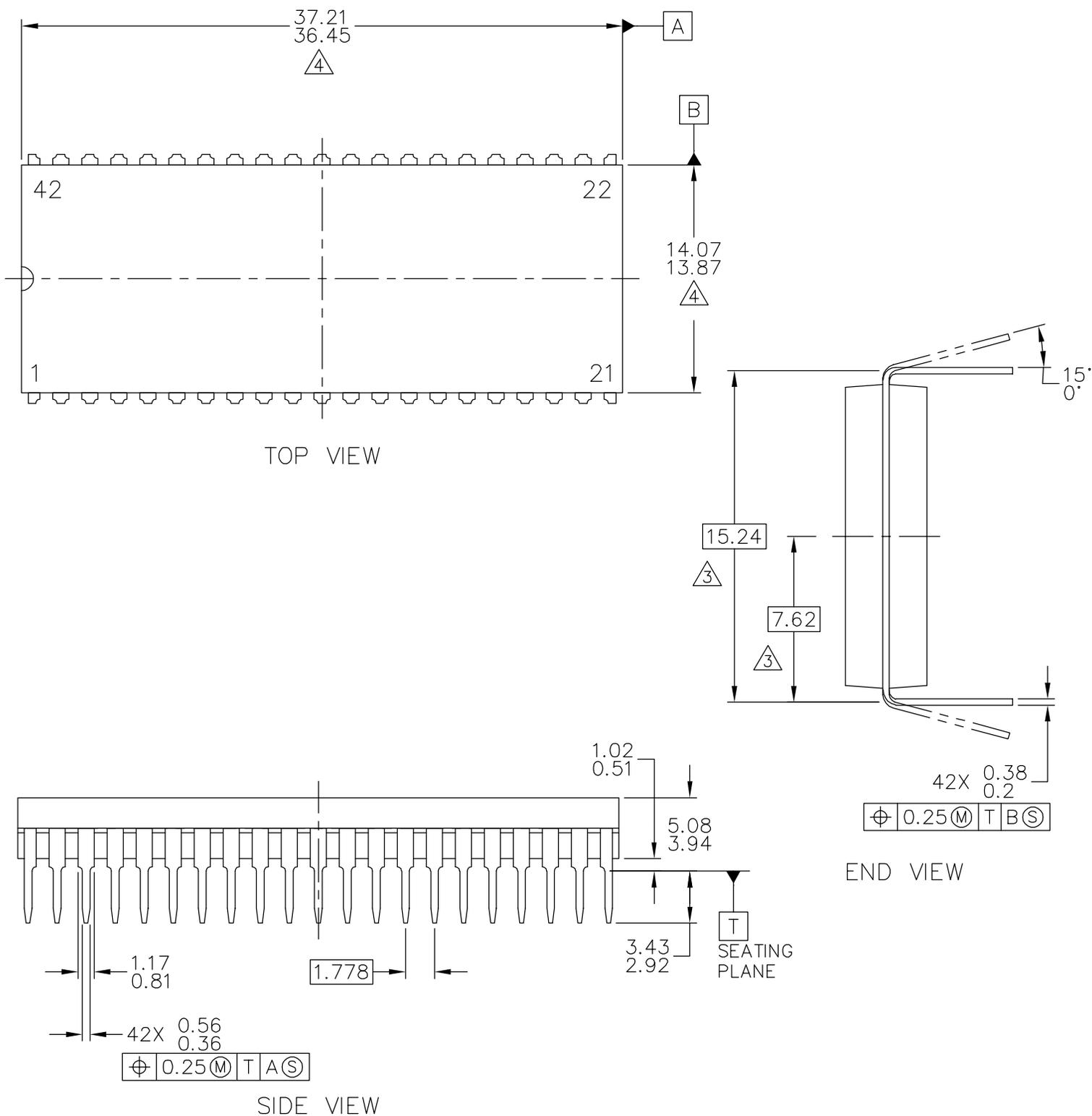




NOTES:

1. DIMENSIONS ARE IN MILLIMETERS.
2. INTERPRET DIMENSIONS AND TOLERANCES PER ASME Y14.5M-1994.
3. THE COMPLETE JEDEC DESIGNATOR FOR THIS PACKAGE IS: HF-PQFN.
4.  COPLANARITY APPLIES TO LEADS, CORNER LEADS, AND DIE ATTACH PAD.
5. MIN METAL GAP SHOULD BE 0.2MM.

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	<b>MECHANICAL OUTLINE</b>	PRINT VERSION NOT TO SCALE	
TITLE: THERMALLY ENHANCED QUAD FLAT NON-LEADED PACKAGE (QFN) 48 TERMINAL, 0.5 PITCH (7 X 7 X 1)	DOCUMENT NO: 98ARH99048A	REV: F	
	CASE NUMBER: 1314-05	05 DEC 2005	
	STANDARD: JEDEC-MO-220 VKKD-2		



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	<b>MECHANICAL OUTLINE</b>	PRINT VERSION NOT TO SCALE	
TITLE:  42 LD PDIP	DOCUMENT NO: 98ASB42767B		REV: A
	CASE NUMBER: 858-01		24 OCT 2005
	STANDARD: NON-JEDEC		