

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	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I ² C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08ac32cfge

Revision History

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://freescale.com/>

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Revision Number	Revision Date	Description of Changes
1	2/2008	Preliminary customer release.
2	3/2008	Market Launch Release.
3	8/2011	Added V_{BG} and I_{IC} in the Table A-6 . Added two figures of Figure 4-2 and Figure 4-3 to replace the old figure of “FLASH Program and Erase Flowchart”. Updated Table 15-8 . Updated R_{IDD} in the Table A-7 . Updated t_{RTI} in the Table A-12 .

Section Number	Title	Page
5.9.6	System Device Identification Register (SDIDH, SDIDL)	79
5.9.7	System Real-Time Interrupt Status and Control Register (SRTISC)	80
5.9.8	System Power Management Status and Control 1 Register (SPMSC1)	81
5.9.9	System Power Management Status and Control 2 Register (SPMSC2)	82
5.9.10	System Options Register 2 (SOPT2)	83

Chapter 6 Parallel Input/Output

6.1	Introduction	85
6.2	Pin Descriptions	85
6.3	Parallel I/O Control	85
6.4	Pin Control	86
6.4.1	Internal Pullup Enable	87
6.4.2	Output Slew Rate Control Enable	87
6.4.3	Output Drive Strength Select	87
6.5	Pin Behavior in Stop Modes	88
6.6	Parallel I/O and Pin Control Registers	88
6.6.1	Port A I/O Registers (PTAD and PTADD)	88
6.6.2	Port A Pin Control Registers (PTAPE, PTASE, PTADS)	89
6.6.3	Port B I/O Registers (PTBD and PTBDD)	91
6.6.4	Port B Pin Control Registers (PTBPE, PTBSE, PTBDS)	92
6.6.5	Port C I/O Registers (PCD and PCDD)	94
6.6.6	Port C Pin Control Registers (PCPE, PCSE, PCDS)	95
6.6.7	Port D I/O Registers (PTDD and PTDDD)	97
6.6.8	Port D Pin Control Registers (PTDPE, PTDSE, PTDDS)	98
6.6.9	Port E I/O Registers (PTED and PTEDD)	100
6.6.10	Port E Pin Control Registers (PTEPE, PTESE, PTEDS)	101
6.6.11	Port F I/O Registers (PTFD and PTFDD)	103
6.6.12	Port F Pin Control Registers (PTFPE, PTFSE, PTFDS)	104
6.6.13	Port G I/O Registers (PTGD and PTGDD)	106
6.6.14	Port G Pin Control Registers (PTGPE, PTGSE, PTGDS)	107

Chapter 7 Central Processor Unit (S08CPUV2)

7.1	Introduction	109
7.1.1	Features	109
7.2	Programmer's Model and CPU Registers	110
7.2.1	Accumulator (A)	110
7.2.2	Index Register (H:X)	110
7.2.3	Stack Pointer (SP)	111
7.2.4	Program Counter (PC)	111
7.2.5	Condition Code Register (CCR)	111

- An external clock source
- The output of the digitally-controlled oscillator (DCO) in the frequency-locked loop sub-module
- Control bits inside the ICG determine which source is connected.
- FFE is a control signal generated inside the ICG. If the frequency of ICGOUT $> 4 \times$ the frequency of ICGERCLK, this signal is a logic 1 and the fixed-frequency clock will be ICGERCLK/2. Otherwise the fixed-frequency clock will be BUSCLK.
- ICGLCLK — Development tools can select this internal self-clocked source (~ 8 MHz) to speed up BDC communications in systems where the bus clock is slow.
- ICGERCLK — External reference clock can be selected as the real-time interrupt clock source. Can also be used as the ALTCLK input to the ADC module.

writing to PPDACK, the pins will be controlled by their associated port control registers when the I/O latches are opened.

A separate self-clocked source (≈ 1 kHz) for the real-time interrupt allows a wakeup from stop2 or stop3 mode with no external components. When $RTIS2:RTIS1:RTIS0 = 0:0:0$, the real-time interrupt function and this 1-kHz source are disabled. Power consumption is lower when the 1-kHz source is disabled, but in that case the real-time interrupt cannot wake the MCU from stop.

3.6.2 Stop3 Mode

To enter stop3, the user must execute a STOP instruction with stop3 selected ($PPDC = 0$) and stop mode enabled ($STOPE = 1$). Upon entering the stop3 mode, all of the clocks in the MCU, including the oscillator itself, are halted. The ICG enters its standby state, as does the voltage regulator and the ADC. The states of all of the internal registers and logic, as well as the RAM content, are maintained. The I/O pin states are not latched at the pin as in stop2. Instead they are maintained by virtue of the states of the internal logic driving the pins being maintained.

Exit from stop3 is done by asserting \overline{RESET} , an asynchronous interrupt pin, or through the real-time interrupt (RTI). The asynchronous interrupt pins are the IRQ or KBI pins. Exit from stop3 can also be facilitated by the SCI receiver interrupt, the ADC, and LVI.

If stop3 is exited by means of the \overline{RESET} pin, then the MCU will be reset and operation will resume after taking the reset vector. Exit by means of an asynchronous interrupt or the real-time interrupt will result in the MCU taking the appropriate interrupt vector.

A separate self-clocked source (≈ 1 kHz) for the real-time interrupt allows a wakeup from stop2 or stop3 mode with no external components. When $RTIS2:RTIS1:RTIS0 = 0:0:0$, the real-time interrupt function and this 1-kHz source are disabled. Power consumption is lower when the 1-kHz source is disabled, but in that case the real-time interrupt cannot wake the MCU from stop.

3.6.3 Active BDM Enabled in Stop Mode

Entry into the active background mode from run mode is enabled if the ENBDM bit in BDCSCR is set. This register is described in [Chapter 16, “Development Support”](#) of this data sheet. If ENBDM is set when the CPU executes a STOP instruction, the system clocks to the background debug logic remain active when the MCU enters stop mode so background debug communication is still possible. In addition, the voltage regulator does not enter its low-power standby state but maintains full internal regulation. If the user attempts to enter stop2 with ENBDM set, the MCU will instead enter stop3.

Most background commands are not available in stop mode. The memory-access-with-status commands do not allow memory access, but they report an error indicating that the MCU is in either stop or wait mode. The BACKGROUND command can be used to wake the MCU from stop and enter active background mode if the ENBDM bit is set. After entering background debug mode, all background commands are available. [Table 3-2](#) summarizes the behavior of the MCU in stop when entry into the background debug mode is enabled.

5.5.2 External Interrupt Request (IRQ) Pin

External interrupts are managed by the IRQSC status and control register. When the IRQ function is enabled, synchronous logic monitors the pin for edge-only or edge-and-level events. When the MCU is in stop mode and system clocks are shut down, a separate asynchronous path is used so the IRQ (if enabled) can wake the MCU.

5.5.2.1 Pin Configuration Options

The IRQ pin enable (IRQPE) control bit in the IRQSC register must be 1 in order for the IRQ pin to act as the interrupt request (IRQ) input. As an IRQ input, the user can choose the polarity of edges or levels detected (IRQEDG), whether the pin detects edges-only or edges and levels (IRQMOD), and whether an event causes an interrupt or only sets the IRQF flag which can be polled by software.

The IRQ pin, when enabled, defaults to use an internal pull device (IRQPDD = 0), configured as a pull-up or pull-down depending on the polarity chosen. If the user desires to use an external pull-up or pull-down, the IRQPDD can be written to a 1 to turn off the internal device.

BIH and BIL instructions may be used to detect the level on the IRQ pin when the pin is configured to act as the IRQ input.

NOTE

- The voltage measured on the pulled up IRQ pin may be as low as $V_{DD}-0.7$ V. The internal gates connected to this pin are pulled all the way to V_{DD} . All other pins with the enabled pullup resistor will have an unloaded measurement of V_{DD} .
- When enabling the IRQ pin for use, the IRQF will be set, and should be cleared prior to enabling the interrupt. When configuring the pin for falling edge and level sensitivity in a 5V system, it is necessary to wait at least 6 cycles between clearing the flag and enabling the interrupt.

5.5.2.2 Edge and Level Sensitivity

The IRQMOD control bit reconfigures the detection logic so it detects edge events and pin levels. In this edge detection mode, the IRQF status flag becomes set when an edge is detected (when the IRQ pin changes from the deasserted to the asserted level), but the flag is continuously set (and cannot be cleared) as long as the IRQ pin remains at the asserted level.

5.5.3 Interrupt Vectors, Sources, and Local Masks

Table 5-2 provides a summary of all interrupt sources. Higher-priority sources are located toward the bottom of the table. The high-order byte of the address for the interrupt service routine is located at the first address in the vector address column, and the low-order byte of the address for the interrupt service routine is located at the next higher address.

When an interrupt condition occurs, an associated flag bit becomes set. If the associated local interrupt enable is 1, an interrupt request is sent to the CPU. Within the CPU, if the global interrupt mask (I bit in the CCR) is 0, the CPU will finish the current instruction, stack the PCL, PCH, X, A, and CCR CPU

7.2.3 Stack Pointer (SP)

This 16-bit address pointer register points at the next available location on the automatic last-in-first-out (LIFO) stack. The stack may be located anywhere in the 64-Kbyte address space that has RAM and can be any size up to the amount of available RAM. The stack is used to automatically save the return address for subroutine calls, the return address and CPU registers during interrupts, and for local variables. The AIS (add immediate to stack pointer) instruction adds an 8-bit signed immediate value to SP. This is most often used to allocate or deallocate space for local variables on the stack.

SP is forced to 0x00FF at reset for compatibility with the earlier M68HC05 Family. HCS08 programs normally change the value in SP to the address of the last location (highest address) in on-chip RAM during reset initialization to free up direct page RAM (from the end of the on-chip registers to 0x00FF).

The RSP (reset stack pointer) instruction was included for compatibility with the M68HC05 Family and is seldom used in new HCS08 programs because it only affects the low-order half of the stack pointer.

7.2.4 Program Counter (PC)

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched.

During normal program execution, the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, interrupt, and return operations load the program counter with an address other than that of the next sequential location. This is called a change-of-flow.

During reset, the program counter is loaded with the reset vector that is located at 0xFFFFE and 0xFFFFF. The vector stored there is the address of the first instruction that will be executed after exiting the reset state.

7.2.5 Condition Code Register (CCR)

The 8-bit condition code register contains the interrupt mask (I) and five flags that indicate the results of the instruction just executed. Bits 6 and 5 are set permanently to 1. The following paragraphs describe the functions of the condition code bits in general terms. For a more detailed explanation of how each instruction sets the CCR bits, refer to the *HCS08 Family Reference Manual, volume 1*.

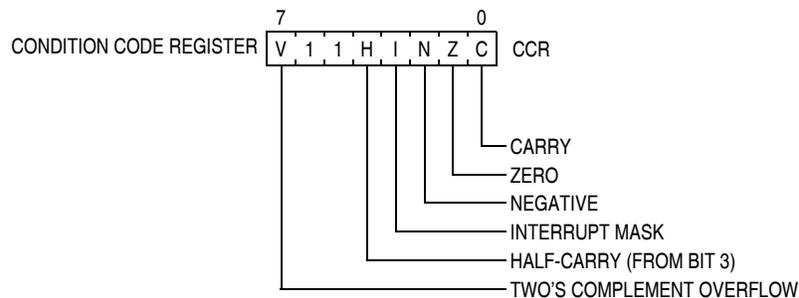
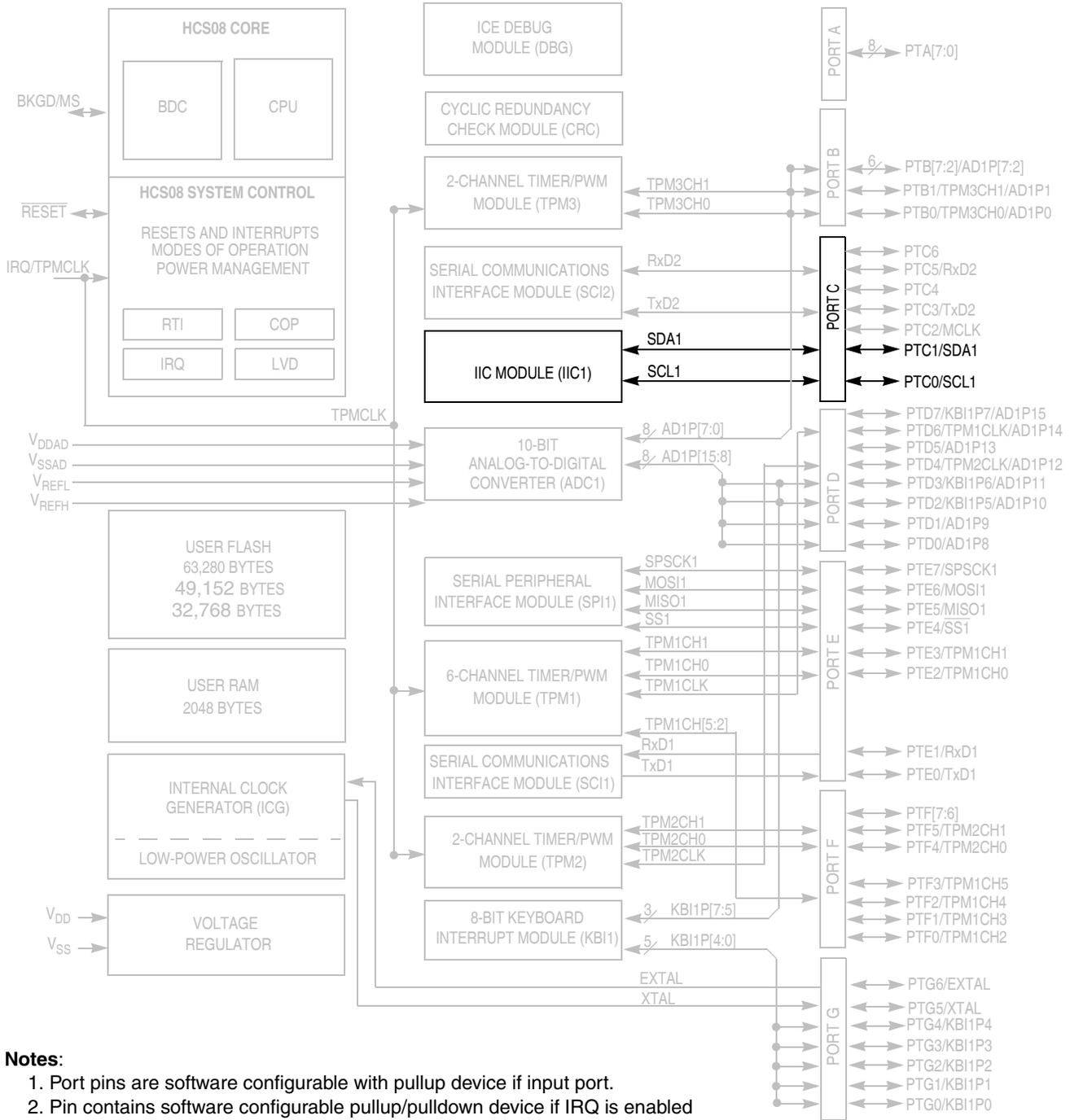


Figure 7-2. Condition Code Register

converter yields the lower code (and vice-versa). However, even very small amounts of system noise can cause the converter to be indeterminate (between two codes) for a range of input voltages around the transition voltage. This range is normally around $\pm 1/2$ LSB and will increase with noise. This error may be reduced by repeatedly sampling the input and averaging the result. Additionally the techniques discussed in [Section 9.7.2.3](#) will reduce this error.

Non-monotonicity is defined as when, except for code jitter, the converter converts to a lower code for a higher input voltage. Missing codes are those values which are never converted for any input value.

In 8-bit or 10-bit mode, the ADC is guaranteed to be monotonic and to have no missing codes.



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. Pin contains integrated pullup device.
4. PTD3, PTD2, PTD7, and PTG4 contain both pullup and pulldown devices. Pulldown enabled when KBI is enabled (KBIPEn = 1) and rising edge is selected (KBEDGn = 1).
5. TPMCLK, TPM1CLK, and TPM2CLK options are configured via software; out of reset, TPM1CLK, TPM2CLK, and TPMCLK are available to TPM1, TPM2, and TPM3 respectively.

Figure 11-1. Block Diagram Highlighting the IIC Module

Table 13-2. SCixBDL Field Descriptions

Field	Description
7:0 SBR[7:0]	Baud Rate Modulo Divisor — These 13 bits in SBR[12:0] are referred to collectively as BR, and they set the modulo divide rate for the SCI baud rate generator. When BR = 0, the SCI baud rate generator is disabled to reduce supply current. When BR = 1 to 8191, the SCI baud rate = BUSCLK/(16×BR). See also BR bits in Table 13-1 .

13.2.2 SCI Control Register 1 (SCixC1)

This read/write register is used to control various optional features of the SCI system.

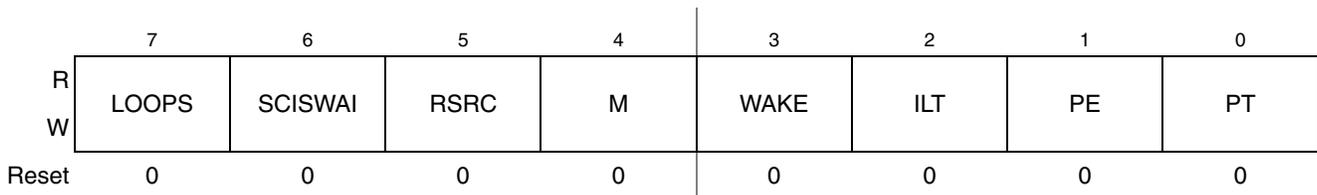


Figure 13-6. SCI Control Register 1 (SCixC1)

Table 13-3. SCixC1 Field Descriptions

Field	Description
7 LOOPS	Loop Mode Select — Selects between loop back modes and normal 2-pin full-duplex modes. When LOOPS = 1, the transmitter output is internally connected to the receiver input. 0 Normal operation — RxD and TxD use separate pins. 1 Loop mode or single-wire mode where transmitter outputs are internally connected to receiver input. (See RSRC bit.) RxD pin is not used by SCI.
6 SCISWAI	SCI Stops in Wait Mode 0 SCI clocks continue to run in wait mode so the SCI can be the source of an interrupt that wakes up the CPU. 1 SCI clocks freeze while CPU is in wait mode.
5 RSRC	Receiver Source Select — This bit has no meaning or effect unless the LOOPS bit is set to 1. When LOOPS = 1, the receiver input is internally connected to the TxD pin and RSRC determines whether this connection is also connected to the transmitter output. 0 Provided LOOPS = 1, RSRC = 0 selects internal loop back mode and the SCI does not use the RxD pins. 1 Single-wire SCI mode where the TxD pin is connected to the transmitter output and receiver input.
4 M	9-Bit or 8-Bit Mode Select 0 Normal — start + 8 data bits (LSB first) + stop. 1 Receiver and transmitter use 9-bit data characters start + 8 data bits (LSB first) + 9th data bit + stop.
3 WAKE	Receiver Wakeup Method Select — Refer to Section 13.3.3.2, “Receiver Wakeup Operation” for more information. 0 Idle-line wakeup. 1 Address-mark wakeup.
2 ILT	Idle Line Type Select — Setting this bit to 1 ensures that the stop bit and logic 1 bits at the end of a character do not count toward the 10 or 11 bit times of logic high level needed by the idle line detection logic. Refer to Section 13.3.3.2.1, “Idle-Line Wakeup” for more information. 0 Idle character bit count starts after start bit. 1 Idle character bit count starts after stop bit.

Table 13-7. SCIxC3 Field Descriptions (continued)

Field	Description
4 TXINV ¹	Transmit Data Inversion — Setting this bit reverses the polarity of the transmitted data output. 0 Transmit data not inverted 1 Transmit data inverted
3 ORIE	Overrun Interrupt Enable — 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	Noise Error Interrupt Enable — 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	Framing Error Interrupt Enable — 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	Parity Error Interrupt Enable — 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.

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

13.2.7 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.

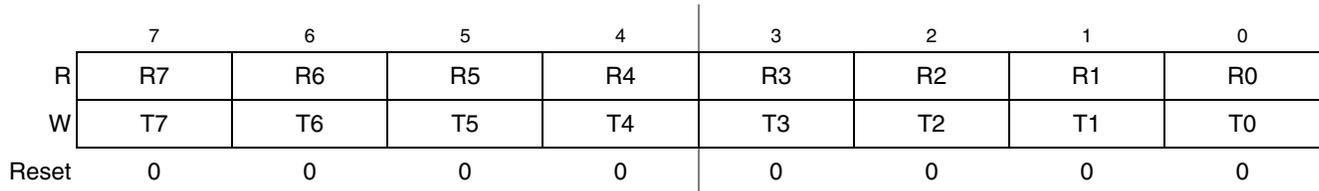


Figure 13-11. SCI Data Register (SCIxD)

13.3 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.

13.3.1 Baud Rate Generation

As shown in [Figure 13-12](#), the clock source for the SCI baud rate generator is the bus-rate clock.

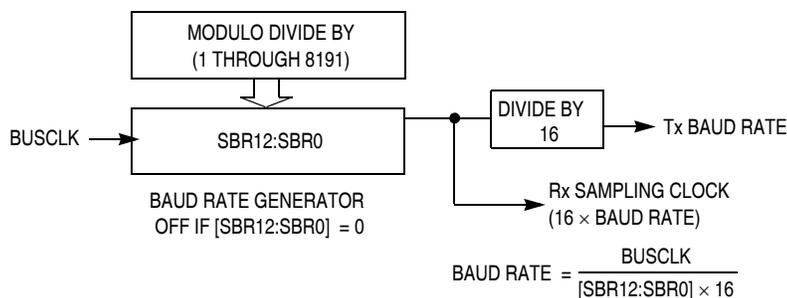


Figure 13-12. SCI Baud Rate Generation

SCI communications require the transmitter and receiver (which typically derive baud rates from independent clock sources) to use the same baud rate. Allowed tolerance on this baud frequency depends on the details of how the receiver synchronizes to the leading edge of the start bit and how bit sampling is performed.

The MCU resynchronizes to bit boundaries on every high-to-low transition, but in the worst case, there are no such transitions in the full 10- or 11-bit time character frame so any mismatch in baud rate is accumulated for the whole character time. For a SCI system whose bus frequency is driven by a crystal, the allowed baud rate mismatch is about ± 4.5 percent for 8-bit data format and about ± 4 percent for 9-bit data format. Although baud rate modulo divider settings do not always produce baud rates that exactly match standard rates, it is normally possible to get within a few percent, which is acceptable for reliable communications.

13.3.2 Transmitter Functional Description

This section describes the overall block diagram for the SCI transmitter, as well as specialized functions for sending break and idle characters. The transmitter block diagram is shown in [Figure 13-2](#).

The transmitter output (TxD) idle state defaults to logic high ($TXINV = 0$ following reset). The transmitter output is inverted by setting $TXINV = 1$. The transmitter is enabled by setting the TE bit in $SCIx C2$. This queues a preamble character that is one full character frame of the idle state. The transmitter then remains idle until data is available in the transmit data buffer. Programs store data into the transmit data buffer by writing to the SCI data register ($SCIxD$).

The central element of the SCI transmitter is the transmit shift register that is either 10 or 11 bits long depending on the setting in the M control bit. For the remainder of this section, we will assume $M = 0$, selecting the normal 8-bit data mode. In 8-bit data mode, the shift register holds a start bit, eight data bits, and a stop bit. When the transmit shift register is available for a new SCI character, the value waiting in the transmit data register is transferred to the shift register (synchronized with the baud rate clock) and the transmit data register empty (TDRE) status flag is set to indicate another character may be written to the transmit data buffer at $SCIxD$.

If no new character is waiting in the transmit data buffer after a stop bit is shifted out the TxD pin, the transmitter sets the transmit complete flag and enters an idle mode, with TxD high, waiting for more characters to transmit.



14.5 Functional Description

An SPI transfer is initiated by checking for the SPI transmit buffer empty flag (SPTEF = 1) and then writing a byte of data to the SPI data register (SPID) in the master SPI device. When the SPI shift register is available, this byte of data is moved from the transmit data buffer to the shifter, SPTEF is set to indicate there is room in the buffer to queue another transmit character if desired, and the SPI serial transfer starts.

During the SPI transfer, data is sampled (read) on the MISO pin at one SPSCCK edge and shifted, changing the bit value on the MOSI pin, one-half SPSCCK cycle later. After eight SPSCCK cycles, the data that was in the shift register of the master has been shifted out the MOSI pin to the slave while eight bits of data were shifted in the MISO pin into the master's shift register. At the end of this transfer, the received data byte is moved from the shifter into the receive data buffer and SPRF is set to indicate the data can be read by reading SPID. If another byte of data is waiting in the transmit buffer at the end of a transfer, it is moved into the shifter, SPTEF is set, and a new transfer is started.

Normally, SPI data is transferred most significant bit (MSB) first. If the least significant bit first enable (LSBFE) bit is set, SPI data is shifted LSB first.

When the SPI is configured as a slave, its \overline{SS} pin must be driven low before a transfer starts and \overline{SS} must stay low throughout the transfer. If a clock format where CPHA = 0 is selected, \overline{SS} must be driven to a logic 1 between successive transfers. If CPHA = 1, \overline{SS} may remain low between successive transfers. See [Section 14.5.1, "SPI Clock Formats"](#) for more details.

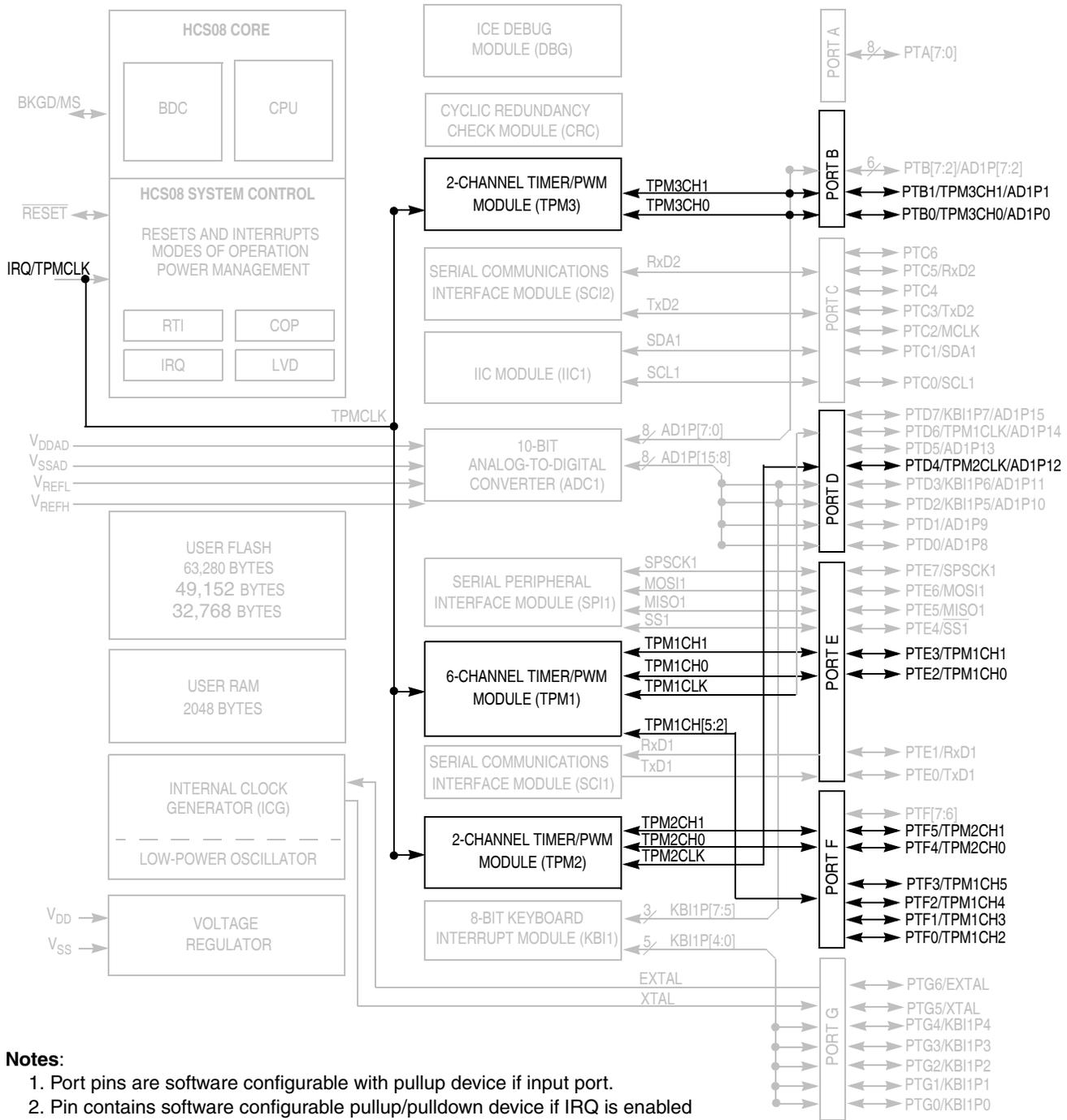
Because the transmitter and receiver are double buffered, a second byte, in addition to the byte currently being shifted out, can be queued into the transmit data buffer, and a previously received character can be in the receive data buffer while a new character is being shifted in. The SPTEF flag indicates when the transmit buffer has room for a new character. The SPRF flag indicates when a received character is available in the receive data buffer. The received character must be read out of the receive buffer (read SPID) before the next transfer is finished or a receive overrun error results.

In the case of a receive overrun, the new data is lost because the receive buffer still held the previous character and was not ready to accept the new data. There is no indication for such an overrun condition so the application system designer must ensure that previous data has been read from the receive buffer before a new transfer is initiated.

14.5.1 SPI Clock Formats

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the SPI system has a clock polarity (CPOL) bit and a clock phase (CPHA) control bit to select one of four clock formats for data transfers. CPOL selectively inserts an inverter in series with the clock. CPHA chooses between two different clock phase relationships between the clock and data.

[Figure 14-10](#) shows the clock formats when CPHA = 1. At the top of the figure, the eight bit times are shown for reference with bit 1 starting at the first SPSCCK edge and bit 8 ending one-half SPSCCK cycle after the sixteenth SPSCCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting in LSBFE. Both variations of SPSCCK polarity are shown, but only one of these waveforms applies for a specific transfer, depending on the value in CPOL. The SAMPLE IN waveform applies to the MOSI input of a slave or the MISO input of a master. The MOSI waveform applies to the



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. Pin contains integrated pullup device.
4. PTD3, PTD2, PTD7, and PTG4 contain both pullup and pulldown devices. Pulldown enabled when KBI is enabled (KBIPEn = 1) and rising edge is selected (KBEDGn = 1).
5. TPMCLK, TPM1CLK, and TPM2CLK options are configured via software; out of reset, TPM1CLK, TPM2CLK, and TPMCLK are available to TPM1, TPM2, and TPM3 respectively.

Figure 15-1. Block Diagram Highlighting the TPM Module

- BDM mode returns the latched value of TPMxCNTH:L from the read buffer instead of the frozen TPM counter value.
- This read coherency mechanism is cleared in TPM v3 in BDM mode if there is a write to TPMxSC, TPMxCNTH or TPMxCNTL. Instead, in these conditions the TPM v2 does not clear this read coherency mechanism.
3. Read of TPMxCnVH:L registers ([Section 15.5.5, “TPM Channel Value Registers \(TPMxCnVH:TPMxCnVL\)”](#))
 - In TPM v3, any read of TPMxCnVH:L registers during BDM mode returns the value of the TPMxCnVH:L register. In TPM v2, if only one byte of the TPMxCnVH:L registers was read before the BDM mode became active, then any read of TPMxCnVH:L registers during BDM mode returns the latched value of TPMxCNTH:L from the read buffer instead of the value in the TPMxCnVH:L registers.
 - This read coherency mechanism is cleared in TPM v3 in BDM mode if there is a write to TPMxCnSC. Instead, in this condition the TPM v2 does not clear this read coherency mechanism.
 4. Write to TPMxCnVH:L registers
 - Input Capture Mode ([Section 15.6.2.1, “Input Capture Mode”](#))

In this mode the TPM v3 does not allow the writes to TPMxCnVH:L registers. Instead, the TPM v2 allows these writes.
 - Output Compare Mode ([Section 15.6.2.2, “Output Compare Mode”](#))

In this mode and if (CLKSB:CLKSA not = 0:0), the TPM v3 updates the TPMxCnVH:L registers with the value of their write buffer at the next change of the TPM counter (end of the prescaler counting) after the second byte is written. Instead, the TPM v2 always updates these registers when their second byte is written.
 - Edge-Aligned PWM ([Section 15.6.2.3, “Edge-Aligned PWM Mode”](#))

In this mode and if (CLKSB:CLKSA not = 00), the TPM v3 updates the TPMxCnVH:L registers with the value of their write buffer after that the both bytes were written and when the TPM counter changes from (TPMxMODH:L - 1) to (TPMxMODH:L). If the TPM counter is a free-running counter, then this update is made when the TPM counter changes from \$FFFE to \$FFFF. Instead, the TPM v2 makes this update after that the both bytes were written and when the TPM counter changes from TPMxMODH:L to \$0000.
 - Center-Aligned PWM ([Section 15.6.2.4, “Center-Aligned PWM Mode”](#))

In this mode and if (CLKSB:CLKSA not = 00), the TPM v3 updates the TPMxCnVH:L registers with the value of their write buffer after that the both bytes were written and when the TPM counter changes from (TPMxMODH:L - 1) to (TPMxMODH:L). If the TPM counter is a free-running counter, then this update is made when the TPM counter changes from \$FFFE to \$FFFF. Instead, the TPM v2 makes this update after that the both bytes were written and when the TPM counter changes from TPMxMODH:L to (TPMxMODH:L - 1).
 5. Center-Aligned PWM ([Section 15.6.2.4, “Center-Aligned PWM Mode”](#))
 - $TPMxCnVH:L = TPMxMODH:L$ [SE110-TPM case 1]

In this case, the TPM v3 produces 100% duty cycle. Instead, the TPM v2 produces 0% duty cycle.

16.4.1.1 BDC Status and Control Register (BDCSCR)

This register can be read or written by serial BDC commands (READ_STATUS and WRITE_CONTROL) but is not accessible to user programs because it is not located in the normal memory map of the MCU.

	7	6	5	4	3	2	1	0
R	ENBDM	BDMACT	BKPTEN	FTS	CLKSW	WS	WSF	DVF
W								
Normal Reset	0	0	0	0	0	0	0	0
Reset in Active BDM:	1	1	0	0	1	0	0	0

= Unimplemented or Reserved

Figure 16-5. BDC Status and Control Register (BDCSCR)

Table 16-2. BDCSCR Register Field Descriptions

Field	Description
7 ENBDM	Enable BDM (Permit Active Background Mode) — Typically, this bit is written to 1 by the debug host shortly after the beginning of a debug session or whenever the debug host resets the target and remains 1 until a normal reset clears it. 0 BDM cannot be made active (non-intrusive commands still allowed) 1 BDM can be made active to allow active background mode commands
6 BDMACT	Background Mode Active Status — This is a read-only status bit. 0 BDM not active (user application program running) 1 BDM active and waiting for serial commands
5 BKPTEN	BDC Breakpoint Enable — If this bit is clear, the BDC breakpoint is disabled and the FTS (force tag select) control bit and BDCBKPT match register are ignored. 0 BDC breakpoint disabled 1 BDC breakpoint enabled
4 FTS	Force/Tag Select — When FTS = 1, a breakpoint is requested whenever the CPU address bus matches the BDCBKPT match register. When FTS = 0, a match between the CPU address bus and the BDCBKPT register causes the fetched opcode to be tagged. If this tagged opcode ever reaches the end of the instruction queue, the CPU enters active background mode rather than executing the tagged opcode. 0 Tag opcode at breakpoint address and enter active background mode if CPU attempts to execute that instruction 1 Breakpoint match forces active background mode at next instruction boundary (address need not be an opcode)
3 CLKSW	Select Source for BDC Communications Clock — CLKSW defaults to 0, which selects the alternate BDC clock source. 0 Alternate BDC clock source 1 MCU bus clock

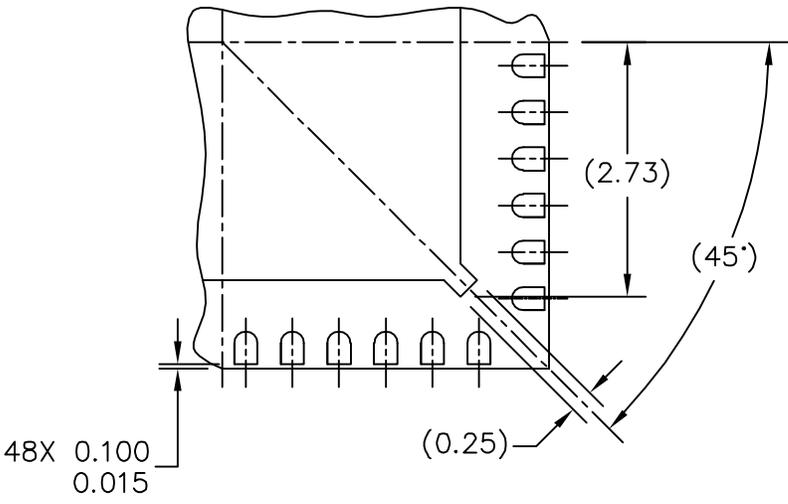
A.8 ADC Characteristics

Table A-8. 5 Volt 10-bit ADC Operating Conditions

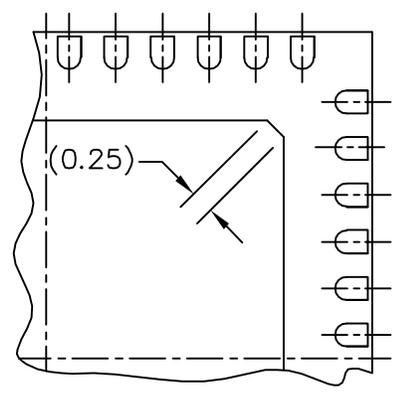
Characteristic	Conditions	Symb	Min	Typ ¹	Max	Unit
Supply voltage	Absolute	V_{DDAD}	2.7	—	5.5	V
	Delta to V_{DD} ($V_{DD}-V_{DDAD}$) ²	ΔV_{DDAD}	-100	0	+100	mV
Ground voltage	Delta to V_{SS} ($V_{SS}-V_{SSAD}$) ²	ΔV_{SSAD}	-100	0	+100	mV
Ref voltage high		V_{REFH}	2.7	V_{DDAD}	V_{DDAD}	V
Ref voltage low		V_{REFL}	V_{SSAD}	V_{SSAD}	V_{SSAD}	V
Supply current	Stop, reset, module off	I_{DDAD}	—	0.011	1	μ A
Input Voltage		V_{ADIN}	V_{REFL}	—	V_{REFH}	V
Input capacitance		C_{ADIN}	—	4.5	5.5	pF
Input resistance		R_{ADIN}	—	3	5	k Ω
Analog source resistance External to MCU	10-bit mode $f_{ADCK} > 4\text{MHz}$ $f_{ADCK} < 4\text{MHz}$	R_{AS}	—	—	5	k Ω
	8-bit mode (all valid f_{ADCK})		—	—	10	
ADC conversion clock frequency	High speed (ADLPC = 0)	f_{ADCK}	0.4	—	8.0	MHz
	Low power (ADLPC = 1)		0.4	—	4.0	
Temp Sensor Slope	-40°C– 25°C	m	—	3.266	—	mV/°C
	25°C– 125°C		—	3.638	—	
Temp Sensor Voltage	25°C	V_{TEMP25}	—	1.396	—	V

¹ Typical values assume $V_{DDAD} = 5.0\text{ V}$, Temp = 25°C, $f_{ADCK} = 1.0\text{MHz}$ unless otherwise stated. Typical values are for reference only and are not tested in production.

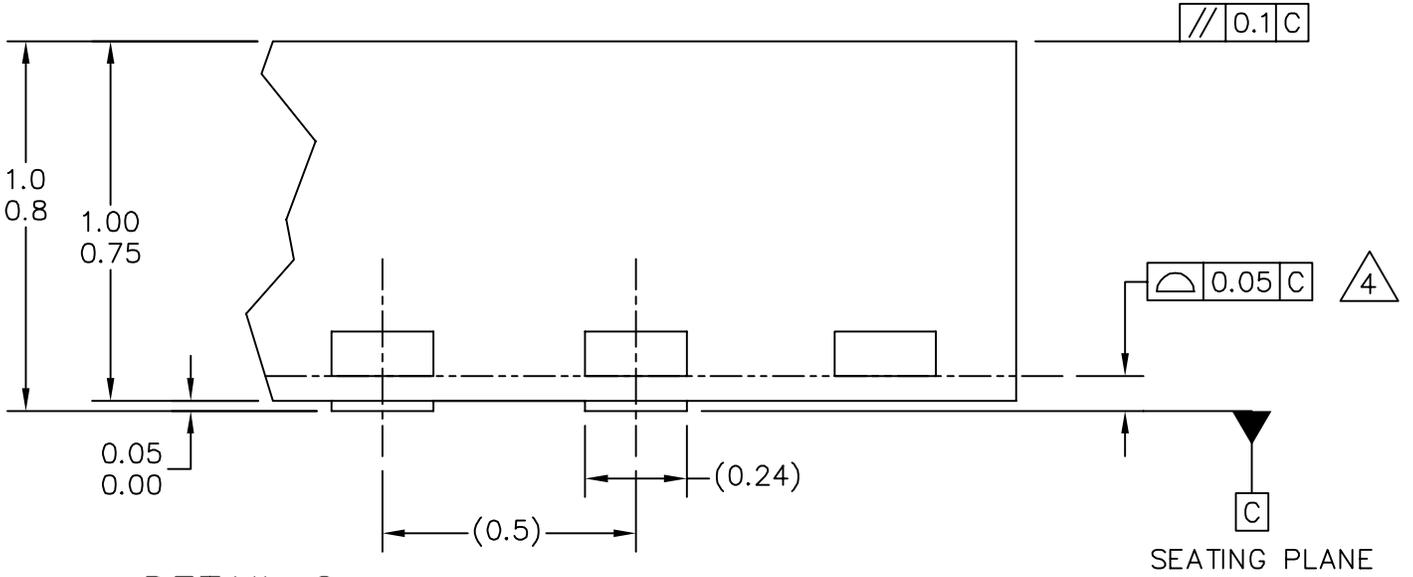
² dc potential difference.



DETAIL N
PREFERRED CORNER CONFIGURATION

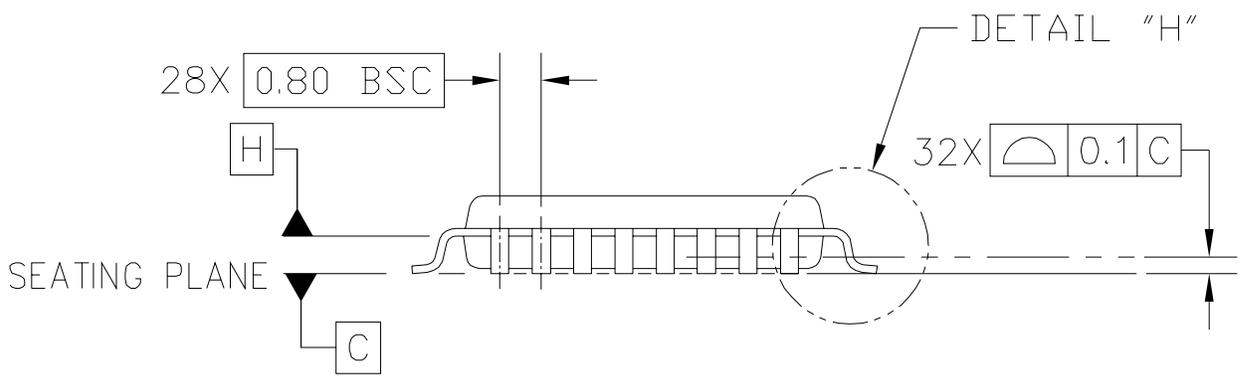
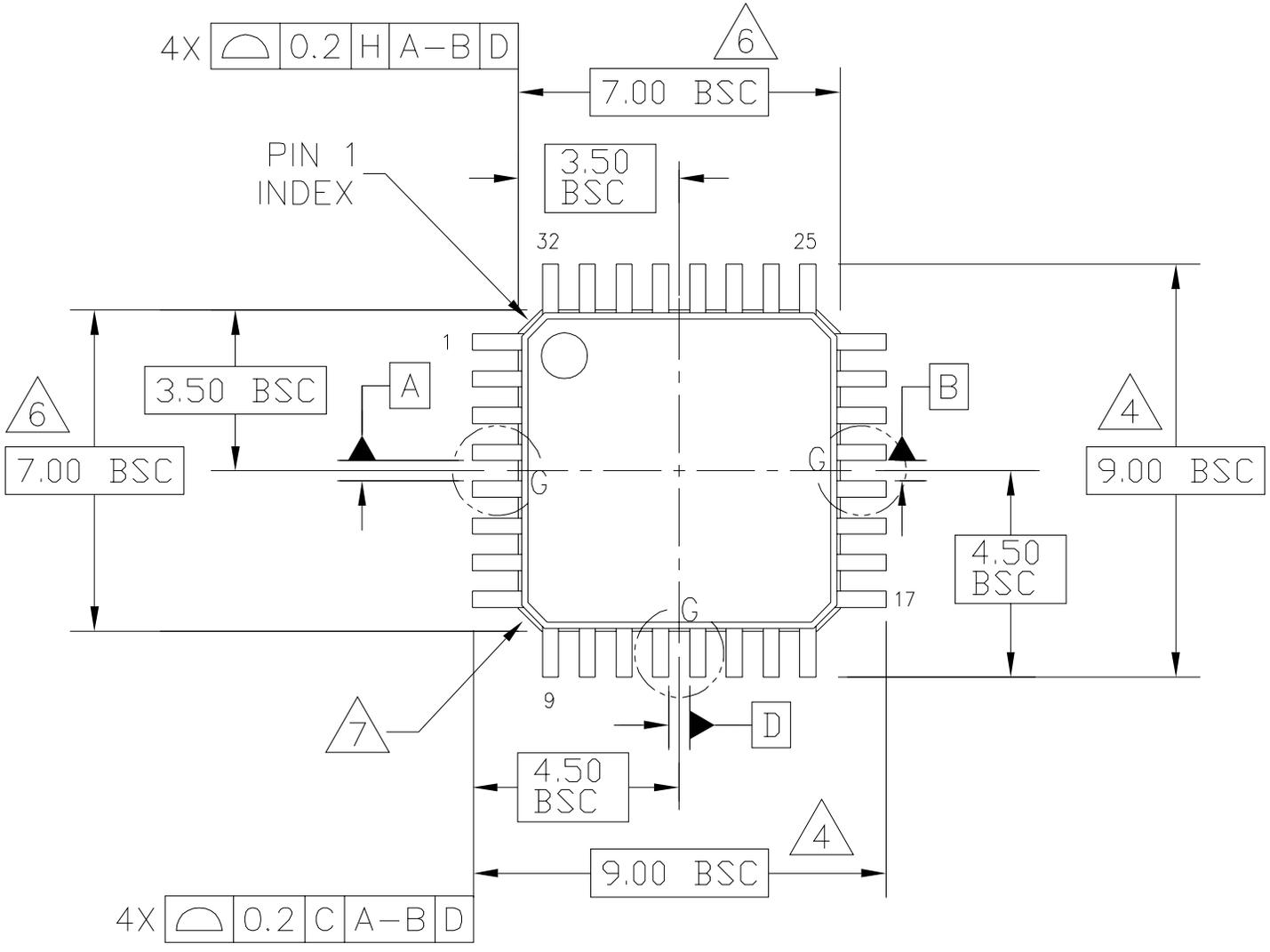


DETAIL M
PREFERRED PIN 1 BACKSIDE IDENTIFIER



DETAIL G
VIEW ROTATED 90° CW

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	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.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: LOW PROFILE QUAD FLAT PACK (LQFP) 32 LEAD, 0.8 PITCH (7 X 7 X 1.4)	DOCUMENT NO: 98ASH70029A	REV: D	
	CASE NUMBER: 873A-03	19 MAY 2005	
	STANDARD: JEDEC MS-026 BBA		