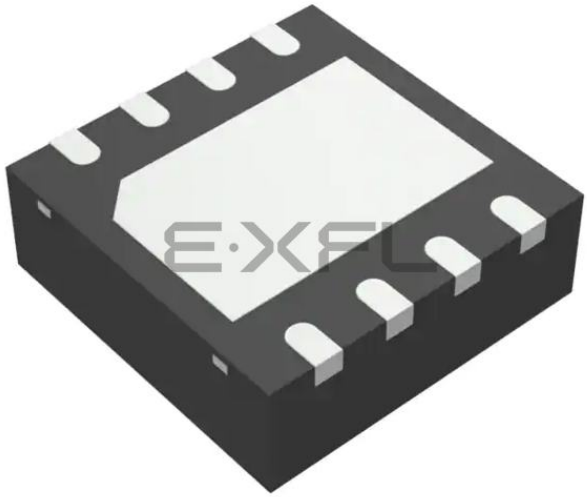


**Welcome to E-XFL.COM**

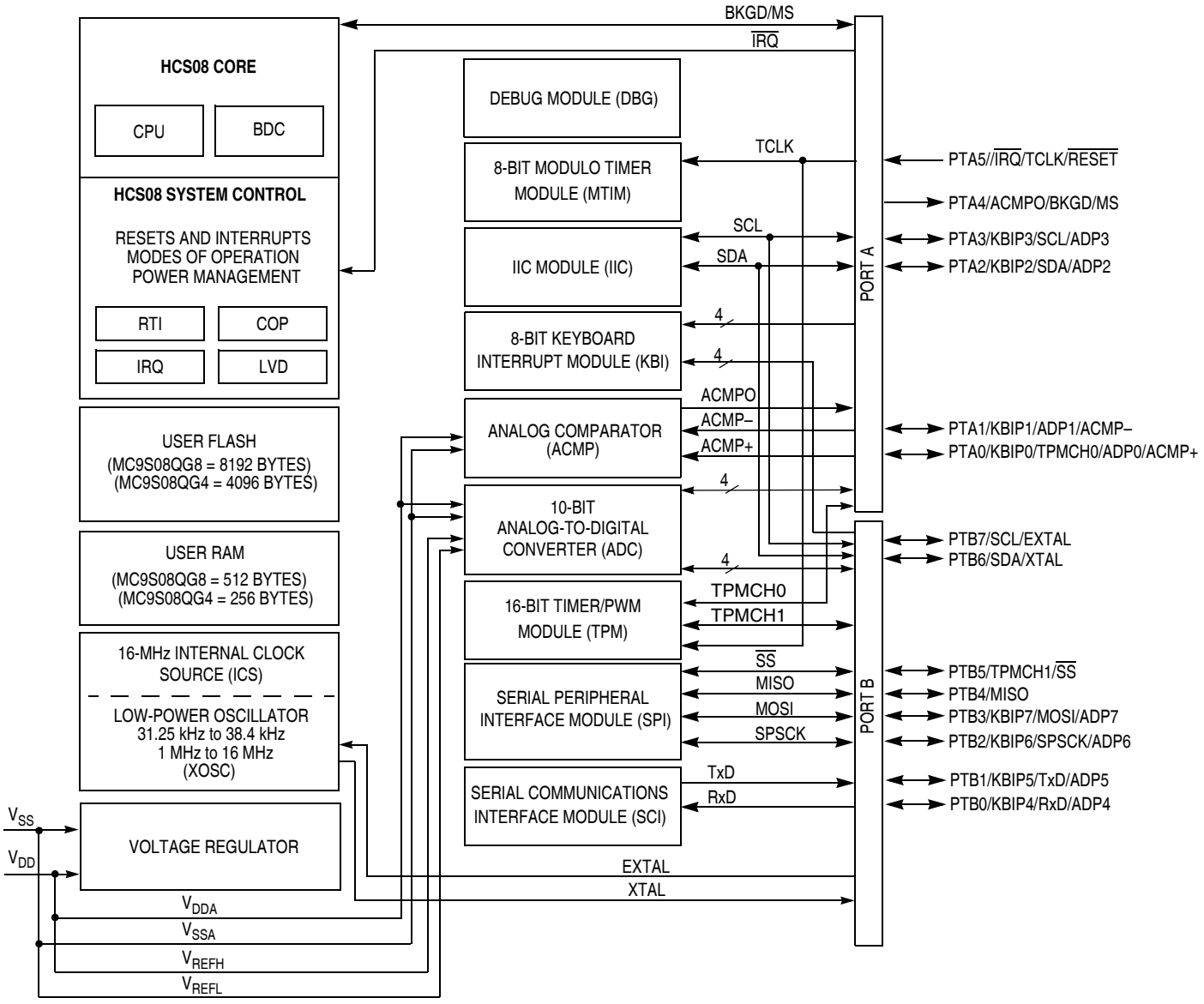
## What is "Embedded - Microcontrollers"?

"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 | 20MHz |
| Connectivity | I²C, SCI, SPI |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 4 |
| Program Memory Size | 4KB (4K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 256 x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.6V |
| Data Converters | A/D 4x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 8-VDFN Exposed Pad |
| Supplier Device Package | 8-DFN-EP (4x4) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08qg4cfqe |

## 1.1.2 MCU Block Diagram




NOTES:
1. Not all pins or pin functions are available on all devices; see Table 1-1 for available functions on each device.
2. Port pins are software configurable with pullup device if input port.
3. Port pins are software configurable for output drive strength.
4. Port pins are software configurable for output slew rate control.
5. $\overline{IRQ}$ contains a software configurable (IRQPDD) pullup device if PTA5 enabled as $\overline{IRQ}$ pin function (IRQPE = 1).
6. $\overline{RESET}$ contains integrated pullup device if PTA5 enabled as reset pin function (RSTPE = 1).
7. PTA4 contains integrated pullup device if BKGD enabled (BKGDPE = 1).
8. SDA and SCL pin locations can be repositioned under software control (IICPS), defaults on PTA2 and PTA3.
9. When pin functions as KBI (KBIPEn = 1) and associated pin is configured to enable the pullup device, KBEDGn can be used to reconfigure the pullup as a pulldown device.

**Figure 1-1. MC9S08QG8/4 Block Diagram**

### 2.2.2 Oscillator (XOSC)

Out of reset, the MCU uses an internally generated clock provided by the internal clock source (ICS) module. The internal frequency is nominally 16-MHz and the default ICS settings will provide for a 8-MHz bus out of reset. For more information on the ICS, see Chapter 10, "Internal Clock Source (S08ICSV1)."

The oscillator module (XOSC) in this MCU is a Pierce oscillator that can accommodate a crystal or ceramic resonator in either of two frequency ranges selected by the RANGE bit in ICSC2. Rather than a crystal or ceramic resonator, an external clock source can be connected to the EXTAL input pin.

Refer to Figure 2-4 for the following discussion. $R_S$ (when used) and $R_F$ should be low-inductance resistors such as carbon composition resistors. Wire-wound resistors, and some metal film resistors, have too much inductance. C1 and C2 normally should be high-quality ceramic capacitors that are specifically designed for high-frequency applications.

$R_F$ is used to provide a bias path to keep the EXTAL input in its linear range during crystal startup, and its value is not generally critical. Typical systems use 1 MΩ to 10 MΩ. Higher values are sensitive to humidity, and lower values reduce gain and (in extreme cases) could prevent startup.

C1 and C2 are typically in the 5-pF to 25-pF range and are chosen to match the requirements of a specific crystal or resonator. Be sure to take into account printed circuit board (PCB) capacitance and MCU pin capacitance when sizing C1 and C2. The crystal manufacturer typically specifies a load capacitance which is the series combination of C1 and C2, which are usually the same size. As a first-order approximation, use 10 pF as an estimate of combined pin and PCB capacitance for each oscillator pin (EXTAL and XTAL).

### 2.2.3 Reset (Input Only)

After a power-on reset (POR), the PTA5/$\overline{\text{IRQ}}$/TCLK/$\overline{\text{RESET}}$ pin defaults to a general-purpose input port pin, PTA5. Setting RSTPE in SOPT1 configures the pin to be the $\overline{\text{RESET}}$ input pin. After configured as $\overline{\text{RESET}}$, the pin will remain $\overline{\text{RESET}}$ until the next POR. The $\overline{\text{RESET}}$ pin can be used to reset the MCU from an external source when the pin is driven low. When enabled as the $\overline{\text{RESET}}$ pin (RSTPE = 1), an internal pullup device is automatically enabled.

**NOTE**

This pin does not contain a clamp diode to $V_{DD}$ and should not be driven above $V_{DD}$.

The voltage measured on the internally pulled-up $\overline{\text{RESET}}$ pin will not be pulled to $V_{DD}$. The internal gates connected to this pin are pulled to $V_{DD}$. The $\overline{\text{RESET}}$ pullup should not be used to pull up components external to the MCU.

**NOTE**

In EMC-sensitive applications, an external RC filter is recommended on the $\overline{\text{RESET}}$ pin, if enabled. See Figure 2-4 for an example.

**NOTE**

To avoid extra current drain from floating input pins, the reset initialization routine in the application program must either enable on-chip pullup devices or change the direction of unused pins to outputs so the pins do not float.

When using the 8-pin devices, the user must either enable on-chip pullup devices or change the direction of non-bonded out port B pins to outputs so the pins do not float.

#### 2.2.5.1 Pin Control Registers

To select drive strength or enable slew rate control or pullup devices, the user writes to the appropriate pin control register located in the high page register block of the memory map. The pin control registers operate independently of the parallel I/O registers and allow control of a port on an individual pin basis.

##### 2.2.5.1.1 Internal Pullup Enable

An internal pullup device can be enabled for each port pin by setting the corresponding bit in one of the pullup enable registers (PTxPEn). The pullup device is disabled if the pin is configured as an output by the parallel I/O control logic or any shared peripheral function, regardless of the state of the corresponding pullup enable register bit. The pullup device is also disabled if the pin is controlled by an analog function.

The KBI module, when enabled for rising edge detection, causes an enabled internal pull device to be configured as a pulldown.

#### 2.2.5.2 Output Slew Rate Control

Slew rate control can be enabled for each port pin by setting the corresponding bit in one of the slew rate control registers (PTxSEn). When enabled, slew control limits the rate at which an output can transition in order to reduce EMC emissions. Slew rate control has no effect on pins that are configured as inputs.

#### 2.2.5.3 Output Drive Strength Select

An output pin can be selected to have high output drive strength by setting the corresponding bit in one of the drive strength select registers (PTxDSn). When high drive is selected, a pin is capable of sourcing and sinking greater current. Even though every I/O pin can be selected as high drive, the user must ensure that the total current source and sink limits for the chip are not exceeded. Drive strength selection is intended to affect the DC behavior of I/O pins. However, the AC behavior is also affected. High drive allows a pin to drive a greater load with the same switching speed as a low drive enabled pin into a smaller load. Because of this, the EMC emissions may be affected by enabling pins as high drive.

# Chapter 3
# Modes of Operation

## 3.1 Introduction

The operating modes of the MC9S08QG8/4 are described in this section. Entry into each mode, exit from each mode, and functionality while in each mode are described.

## 3.2 Features

- Active background mode for code development
- Wait mode:
  - CPU halts operation to conserve power
  - System clocks running
  - Full voltage regulation is maintained
- Stop modes: CPU and bus clocks stopped
  - Stop1: Full powerdown of internal circuits for maximum power savings
  - Stop2: Partial powerdown of internal circuits; RAM contents retained
  - Stop3: All internal circuits powered for fast recovery; RAM and register contents are retained

## 3.3 Run Mode

Run is the normal operating mode for the MC9S08QG8/4. This mode is selected upon the MCU exiting reset if the BKGD/MS pin is high. In this mode, the CPU executes code from internal memory with execution beginning at the address fetched from memory at 0xFFFE:0xFFFF after reset.

## 3.4 Active Background Mode

The active background mode functions are managed through the background debug controller (BDC) in the HCS08 core. The BDC, together with the on-chip debug module (DBG), provides the means for analyzing MCU operation during software development.

Active background mode is entered in any of five ways:

- When the BKGD/MS pin is low during POR or immediately after issuing a background debug force reset (see 5.8.3, “System Background Debug Force Reset Register (SBDFR)”)
- When a BACKGROUND command is received through the BKGD pin
- When a BGND instruction is executed
- When encountering a BDC breakpoint
- When encountering a DBG breakpoint

## 5.8.9 System Power Management Status and Control 2 Register (SPMSC2)

This high page register contains status and control bits to configure the stop mode behavior of the MCU. See Section 3.6, "Stop Modes," for more information on stop modes.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | PDF | PPDF | 0 | PDC[1] | PPDC[1] |
| W | | | | | | PPDACK | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 5-11. System Power Management Status and Control 2 Register (SPMSC2)**

1 This bit can be written only one time after reset. Additional writes are ignored.

**Table 5-13. SPMSC2 Register Field Descriptions**

| Field | Description |
|---|---|
| 4<br>PDF | **Power Down Flag** — This read-only status bit indicates the MCU has recovered from stop1 mode.<br>0 MCU has not recovered from stop1 mode.<br>1 MCU recovered from stop1 mode. |
| 3<br>PPDF | **Partial Power Down Flag** — This read-only status bit indicates that the MCU has recovered from stop2 mode.<br>0 MCU has not recovered from stop2 mode.<br>1 MCU recovered from stop2 mode. |
| 2<br>PPDACK | **Partial Power Down Acknowledge** — Writing a 1 to PPDACK clears the PPDF and the PDF bits. |
| 1<br>PDC | **Power Down Control** — The PDC bit controls entry into the power down (stop2 and stop1) modes.<br>0 Power down modes are disabled.<br>1 Power down modes are enabled. |
| 0<br>PPDC | **Partial Power Down Control** — The PPDC bit controls which power down mode is selected.<br>0 Stop1 full power down mode enabled if PDC set.<br>1 Stop2 partial power down mode enabled if PDC set. |

### 6.4.2.3 Port A Drive Strength Select (PTADS)

An output pin can be selected to have high output drive strength by setting the corresponding bit in the drive strength select register (PTADS). When high drive is selected, a pin is capable of sourcing and sinking greater current. Even though every I/O pin can be selected as high drive, the user must ensure that the total current source and sink limits for the chip are not exceeded. Drive strength selection is intended to affect the DC behavior of I/O pins. However, the AC behavior is also affected. High drive allows a pin to drive a greater load with the same switching speed as a low drive enabled pin into a smaller load. Because of this the EMC emissions may be affected by enabling pins as high drive.

### 6.4.2.4 Port A Drive Strength Select (PTADS)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | PTADS5[1] | PTADS4 | PTADS3 | PTADS2 | PTADS1 | PTADS0 |
| W | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 6-8. Drive Strength Selection for Port A Register (PTADS)**

1 PTADS5 has no effect on the input-only PTA5 pin.

**Table 6-5. PTADS Register Field Descriptions**

| Field | Description |
|---|---|
| 5:0<br>PTADS[5:0] | **Output Drive Strength Selection for Port A Bits** — Each of these control bits selects between low and high output drive for the associated PTA pin. For port A pins that are configured as inputs, these bits have no effect.<br>0 Low output drive strength selected for port A bit n.<br>1 High output drive strength selected for port A bit n. |

### 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 0xFFFE and 0xFFFF. 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,* Freescale Semiconductor document order number HCS08RMv1.

#### 7.3.6.7 SP-Relative, 16-Bit Offset (SP2)

This variation of indexed addressing uses the 16-bit value in the stack pointer (SP) plus a 16-bit offset included in the instruction as the address of the operand needed to complete the instruction.

## 7.4 Special Operations

The CPU performs a few special operations that are similar to instructions but do not have opcodes like other CPU instructions. In addition, a few instructions such as STOP and WAIT directly affect other MCU circuitry. This section provides additional information about these operations.

### 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 0xFFFE 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

KBISC provided all enabled keyboard inputs are at their deasserted levels. KBF will remain set if any enabled KBI pin is asserted while attempting to clear by writing a 1 to KBACK.

### 12.4.3 KBI Pullup/Pulldown Resistors

The KBI pins can be configured to use an internal pullup/pulldown resistor using the associated I/O port pullup enable register. If an internal resistor is enabled, the KBIES register is used to select whether the resistor is a pullup (KBEDGn = 0) or a pulldown (KBEDGn = 1).

### 12.4.4 KBI Initialization

When a keyboard interrupt pin is first enabled it is possible to get a false keyboard interrupt flag. To prevent a false interrupt request during keyboard initialization, the user should do the following:

1. Mask keyboard interrupts by clearing KBIE in KBISC.
2. Enable the KBI polarity by setting the appropriate KBEDGn bits in KBIES.
3. If using internal pullup/pulldown device, configure the associated pullup enable bits in PTxPE.
4. Enable the KBI pins by setting the appropriate KBIPEn bits in KBIPE.
5. Write to KBACK in KBISC to clear any false interrupts.
6. Set KBIE in KBISC to enable interrupts.

### 13.1.4 Block Diagram

The block diagram for the modulo timer module is shown Figure 13-2.

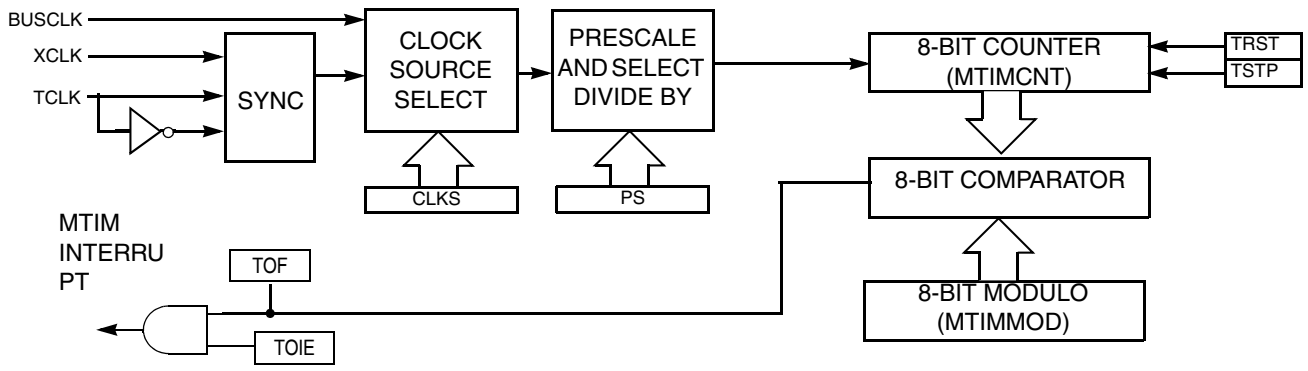



**Figure 13-2. Modulo Timer (MTIM) Block Diagram**

## 13.2 External Signal Description

The MTIM includes one external signal, TCLK, used to input an external clock when selected as the MTIM clock source. The signal properties of TCLK are shown in Table 13-1.

**Table 13-1. Signal Properties**

| Signal | Function | I/O |
|---|---|---|
| TCLK | External clock source input into MTIM | I |

The TCLK input must be synchronized by the bus clock. Also, variations in duty cycle and clock jitter must be accommodated. Therefore, the TCLK signal must be limited to one-fourth of the bus frequency.

The TCLK pin can be muxed with a general-purpose port pin. See the Pins and Connections chapter for the pin location and priority of this function.

## 13.3 Register Definition

Figure 13-3 is a summary of MTIM registers.

masked by local interrupt enable masks. The flags can still be polled by software when the local masks are cleared to disable generation of hardware interrupt requests.

The SCI transmitter has two status flags that optionally can generate hardware interrupt requests. Transmit data register empty (TDRE) indicates when there is room in the transmit data buffer to write another transmit character to SCID. If the transmit interrupt enable (TIE) bit is set, a hardware interrupt will be requested whenever TDRE = 1. Transmit complete (TC) indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with TxD1 high. This flag is often used in systems with modems to determine when it is safe to turn off the modem. If the transmit complete interrupt enable (TCIE) bit is set, a hardware interrupt will be requested whenever TC = 1. Instead of hardware interrupts, software polling may be used to monitor the TDRE and TC status flags if the corresponding TIE or TCIE local interrupt masks are 0s.

When a program detects that the receive data register is full (RDRF = 1), it gets the data from the receive data register by reading SCID. The RDRF flag is cleared by reading SCIS1 while RDRF = 1 and then reading SCID.

When polling is used, this sequence is naturally satisfied in the normal course of the user program. If hardware interrupts are used, SCIS1 must be read in the interrupt service routine (ISR). Normally, this is done in the ISR anyway to check for receive errors, so the sequence is automatically satisfied.

The IDLE status flag includes logic that prevents it from getting set repeatedly when the RxD1 line remains idle for an extended period of time. IDLE is cleared by reading SCIS1 while IDLE = 1 and then reading SCID. After IDLE has been cleared, it cannot become set again until the receiver has received at least one new character and has set RDRF.

If the associated error was detected in the received character that caused RDRF to be set, the error flags — noise flag (NF), framing error (FE), and parity error flag (PF) — get set at the same time as RDRF. These flags are not set in overrun cases.

If RDRF was already set when a new character is ready to be transferred from the receive shifter to the receive data buffer, the overrun (OR) flag gets set instead and the data and any associated NF, FE, or PF condition is lost.

## 14.4 Additional SCI Functions

The following sections describe additional SCI functions.

### 14.4.1 8- and 9-Bit Data Modes

The SCI system (transmitter and receiver) can be configured to operate in 9-bit data mode by setting the M control bit in SCIC1. In 9-bit mode, there is a ninth data bit to the left of the MSB of the SCI data register. For the transmit data buffer, this bit is stored in T8 in SCIC3. For the receiver, the ninth bit is held in R8 in SCIC3.

For coherent writes to the transmit data buffer, write to the T8 bit before writing to SCID.

MOSI output pin from a master and the MISO waveform applies to the MISO output from a slave. The $\overline{SS}$ OUT waveform applies to the slave select output from a master (provided MODFEN and SSOE = 1). The master $\overline{SS}$ output goes to active low one-half SPSCK cycle before the start of the transfer and goes back high at the end of the eighth bit time of the transfer. The $\overline{SS}$ IN waveform applies to the slave select input of a slave.

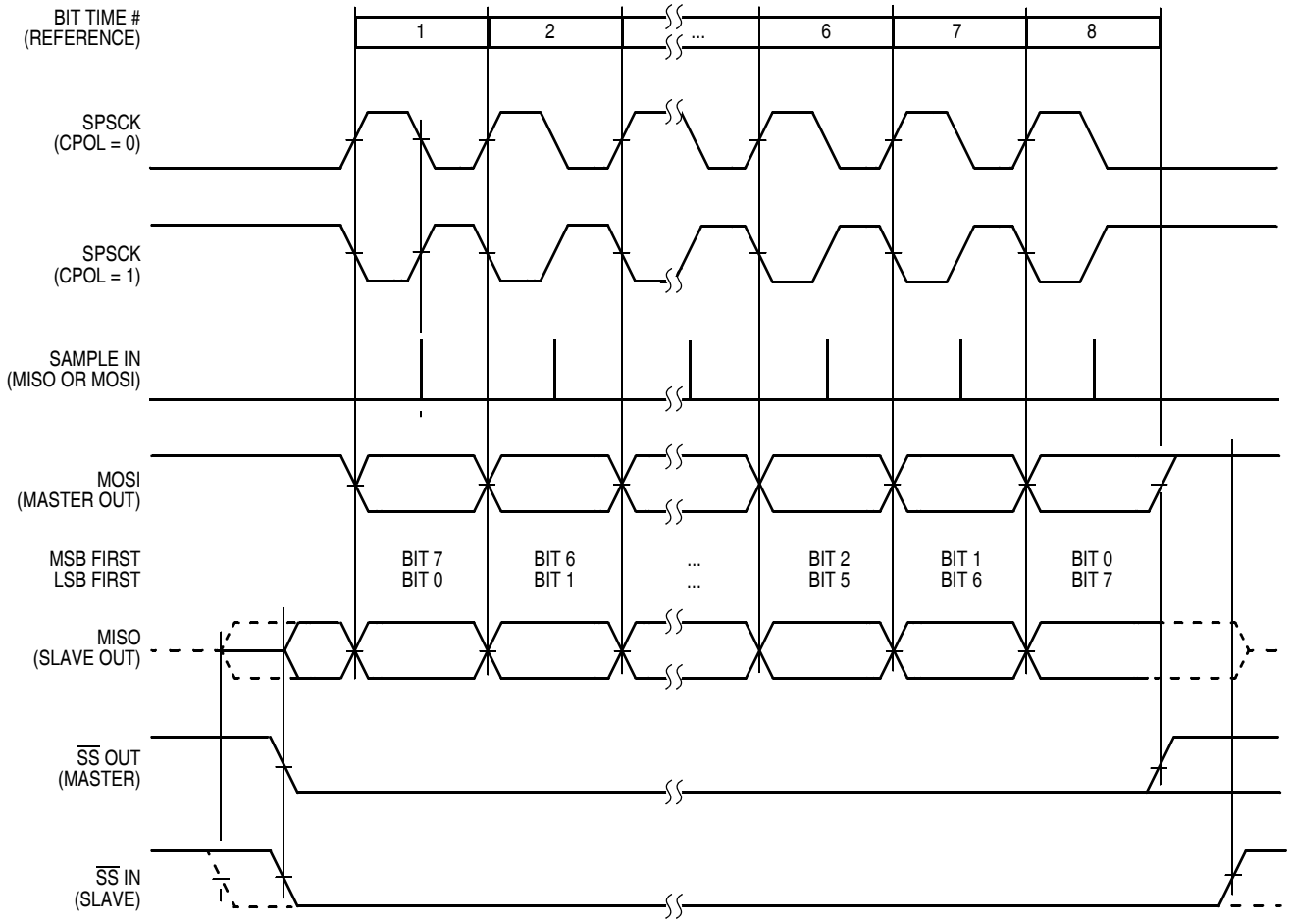



**Figure 15-10. SPI Clock Formats (CPHA = 1)**

When CPHA = 1, the slave begins to drive its MISO output when $\overline{SS}$ goes to active low, but the data is not defined until the first SPSCK edge. The first SPSCK edge shifts the first bit of data from the shifter onto the MOSI output of the master and the MISO output of the slave. The next SPSCK edge causes both the master and the slave to sample the data bit values on their MISO and MOSI inputs, respectively. At the third SPSCK edge, the SPI shifter shifts one bit position which shifts in the bit value that was just sampled, and shifts the second data bit value out the other end of the shifter to the MOSI and MISO outputs of the master and slave, respectively. When CHPA = 1, the slave's $\overline{SS}$ input is not required to go to its inactive high level between transfers.

Figure 15-11 shows the clock formats when CPHA = 0. At the top of the figure, the eight bit times are shown for reference with bit 1 starting as the slave is selected ($\overline{SS}$ IN goes low), and bit 8 ends at the last SPSCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting

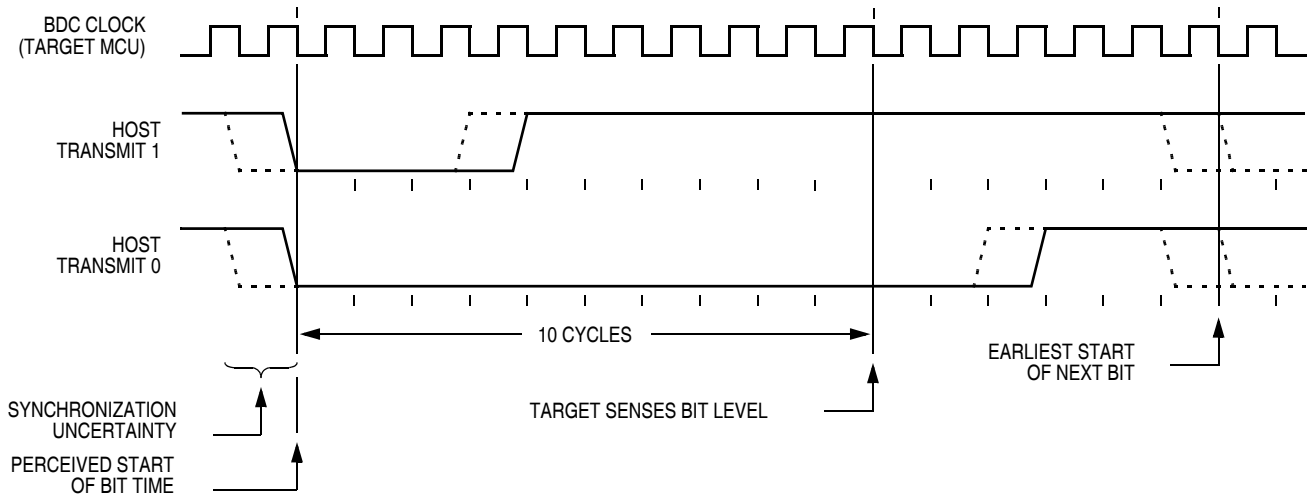



**Figure 17-2. BDC Host-to-Target Serial Bit Timing**

Figure 17-3 shows the host receiving a logic 1 from the target HCS08 MCU. Because the host is asynchronous to the target MCU, there is a 0-to-1 cycle delay from the host-generated falling edge on BKGD to the perceived start of the bit time in the target MCU. The host holds the BKGD pin low long enough for the target to recognize it (at least two target BDC cycles). The host must release the low drive before the target MCU drives a brief active-high speedup pulse seven cycles after the perceived start of the bit time. The host should sample the bit level about 10 cycles after it started the bit time.

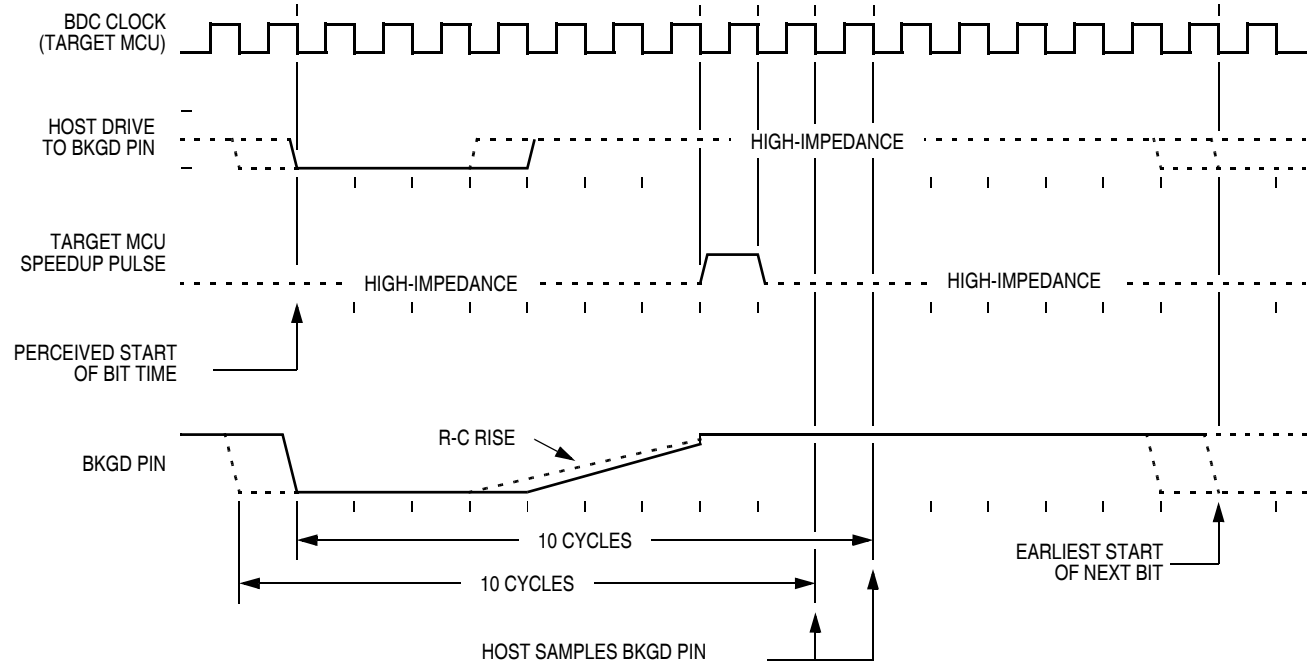



**Figure 17-3. BDC Target-to-Host Serial Bit Timing (Logic 1)**

Figure 17-4 shows the host receiving a logic 0 from the target HCS08 MCU. Because the host is asynchronous to the target MCU, there is a 0-to-1 cycle delay from the host-generated falling edge on BKGD to the start of the bit time as perceived by the target MCU. The host initiates the bit time but the target HCS08 finishes it. Because the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 BDC clock cycles, then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 cycles after starting the bit time.

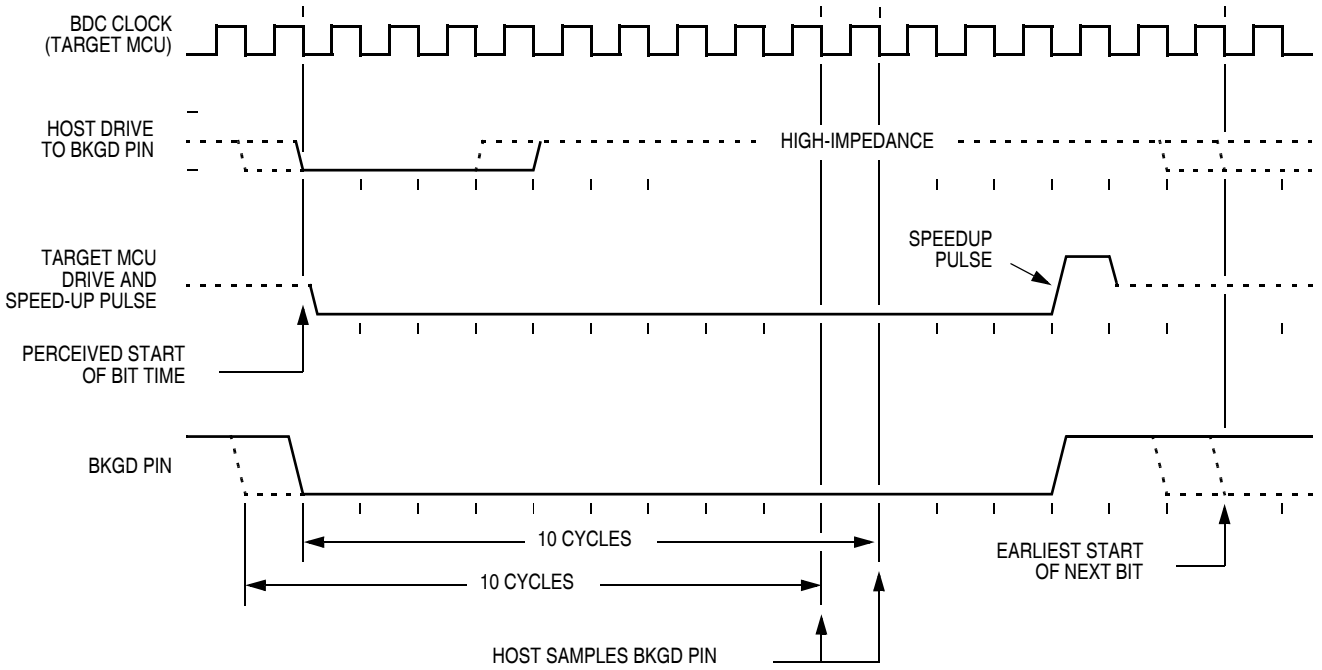



**Figure 17-4. BDM Target-to-Host Serial Bit Timing (Logic 0)**

## 17.2.3 BDC Commands

BDC commands are sent serially from a host computer to the BKGD pin of the target HCS08 MCU. All commands and data are sent MSB-first using a custom BDC communications protocol. Active background mode commands require that the target MCU is currently in the active background mode while non-intrusive commands may be issued at any time whether the target MCU is in active background mode or running a user application program.

Table 17-1 shows all HCS08 BDC commands, a shorthand description of their coding structure, and the meaning of each command.

### Coding Structure Nomenclature

This nomenclature is used in Table 17-1 to describe the coding structure of the BDC commands.

$P_{int} = I_{DD} \times V_{DD}$, Watts — chip internal power

$P_{I/O}$ = Power dissipation on input and output pins — user determined

For most applications, $P_{I/O} << P_{int}$ and can be neglected. An approximate relationship between $P_D$ and $T_J$ (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \qquad \textit{Eqn. A-2}$$

Solving Equation A-1 and Equation A-2 for K gives:

$$K = P_D \times (T_A + 273°C) + \theta_{JA} \times (P_D)^2 \qquad \textit{Eqn. A-3}$$

where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring $P_D$ (at equilibrium) for a known $T_A$. Using this value of K, the values of $P_D$ and $T_J$ can be obtained by solving Equation A-1 and Equation A-2 iteratively for any value of $T_A$.

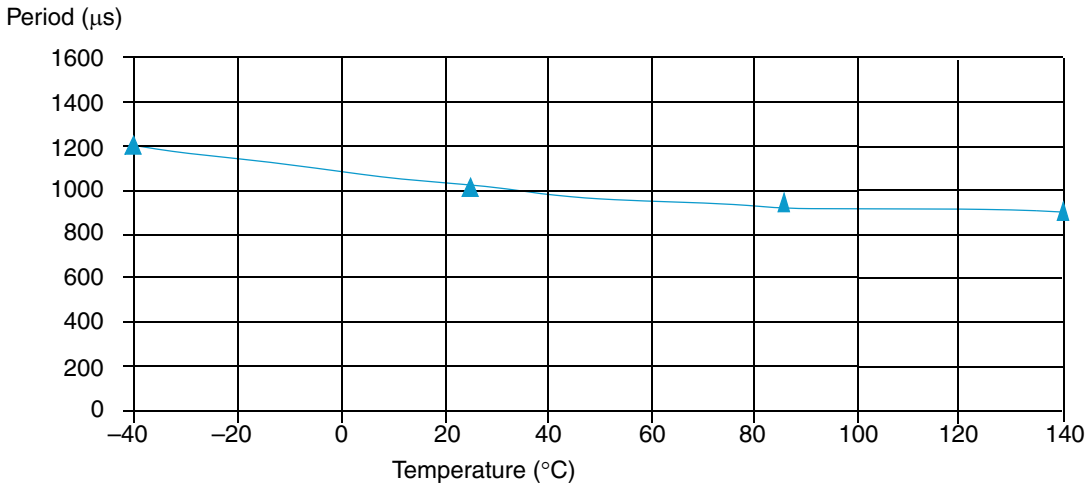



**Figure A-8. Typical RTI Clock Period vs. Temperature**

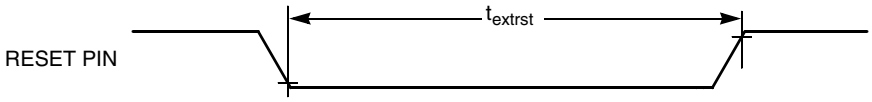



**Figure A-9. Reset Timing**

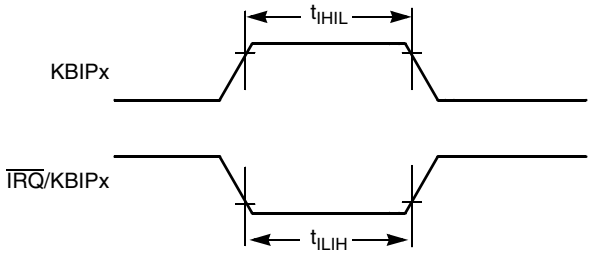



**Figure A-10. IRQ/KBIPx Timing**

## A.8.2 TPM/MTIM Module Timing

Synchronizer circuits determine the shortest input pulses that can be recognized or the fastest clock that can be used as the optional external source to the timer counter. These synchronizers operate from the current bus rate clock.

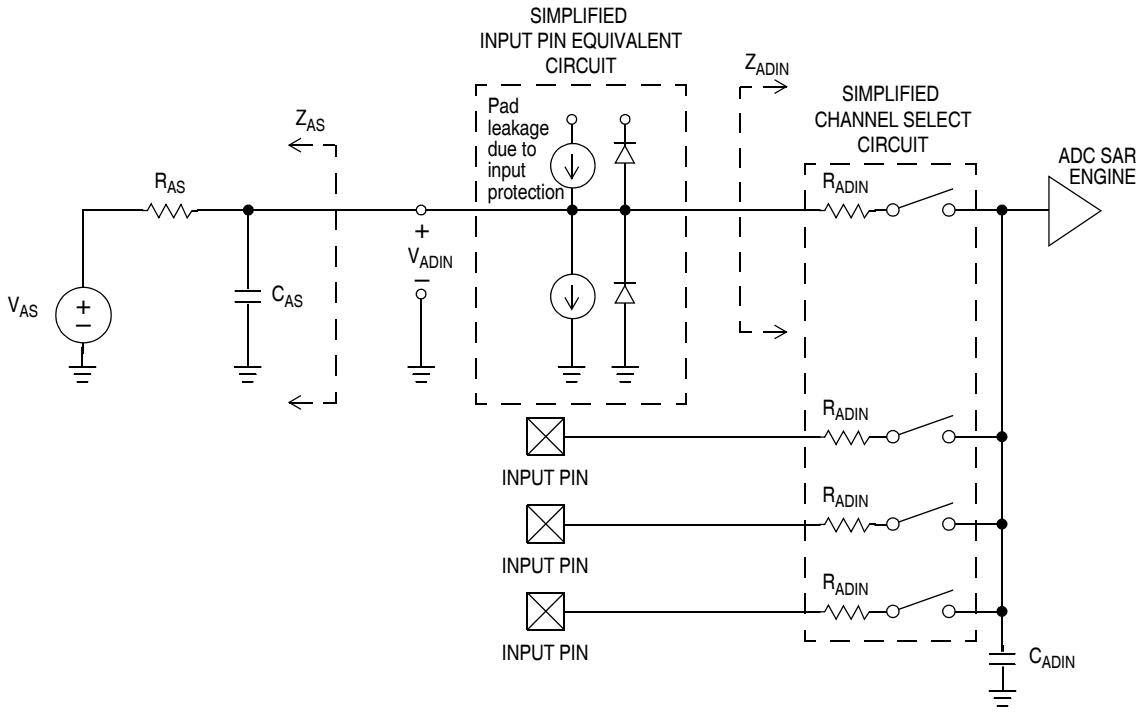
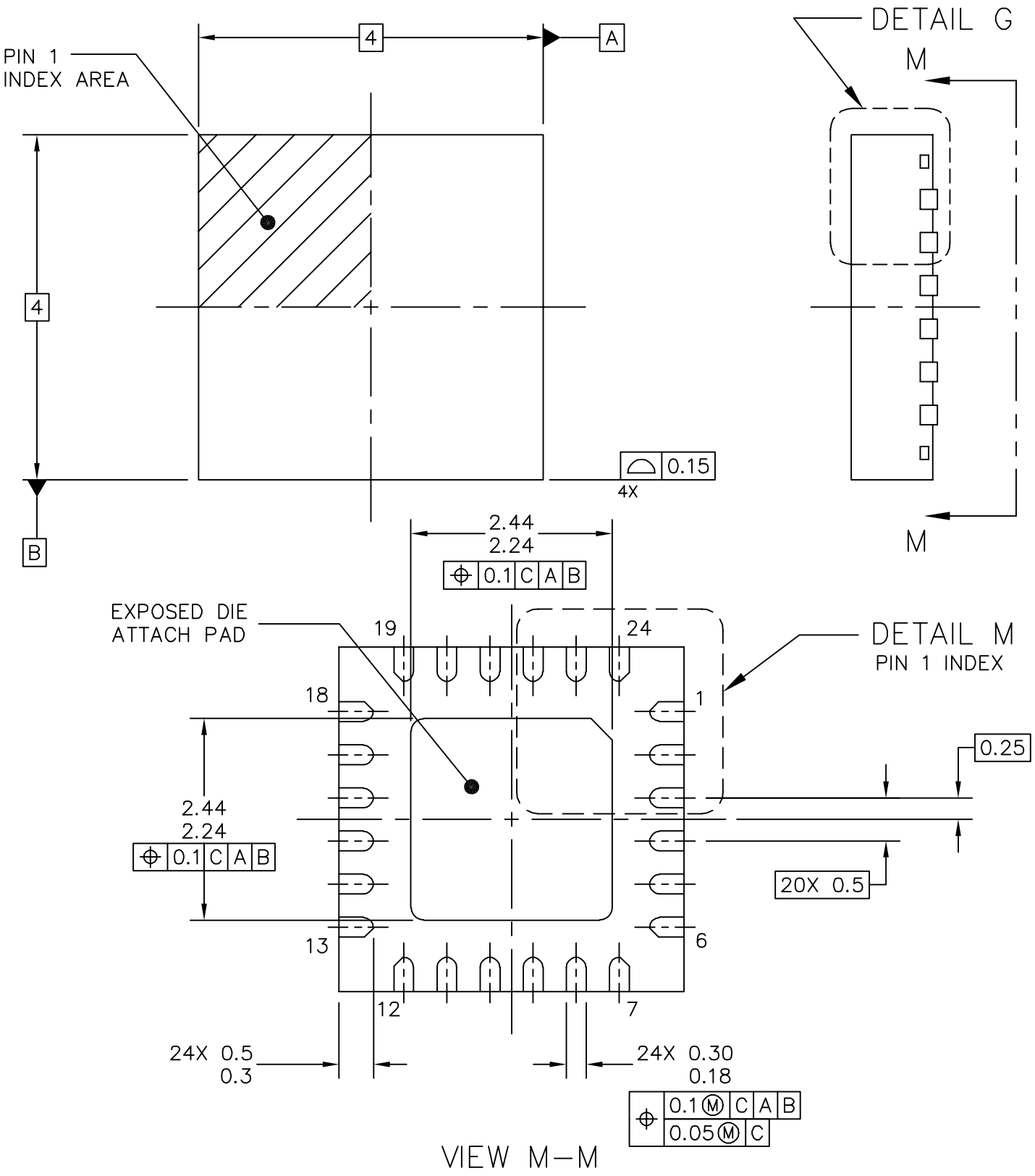



**Figure A-17. ADC Input Impedance Equivalency Diagram**

**Table A-14. 3 Volt 10-bit ADC Characteristics**

| Characteristic | Conditions | Symb | Min | Typ[1] | Max | Unit | Comment |
|---|---|---|---|---|---|---|---|
| Supply current<br>ADLPC=1<br>ADLSMP=1<br>ADCO=1 | | $I_{DDAD}$ | — | 120 | — | μA | |
| Supply current<br>ADLPC=1<br>ADLSMP=0<br>ADCO=1 | | $I_{DDAD}$ | — | 202 | — | μA | |
| Supply current<br>ADLPC=0<br>ADLSMP=1<br>ADCO=1 | | $I_{DDAD}$ | — | 288 | — | μA | |
| Supply current<br>ADLPC=0<br>ADLSMP=0<br>ADCO=1 | | $I_{DDAD}$ | — | 532 | 646 | μA | |
| ADC asynchronous clock source | High speed (ADLPC=0) | $f_{ADACK}$ | 2 | 3.3 | 5 | MHz | $t_{ADACK} = 1/f_{ADACK}$ |
| | Low power (ADLPC=1) | | 1.25 | 2 | 3.3 | | |






| MECHANICAL OUTLINE | PRINT VERSION NOT TO SCALE |
|---|---|

| TITLE: THERMALLY ENHANCED QUAD FLAT NON–LEADED PACKAGE (QFN) 24 TERMINAL, 0.5 PITCH (4 X 4 X 1) | DOCUMENT NO: 98ARL10605D | REV: O |
|---|---|---|
| | CASE NUMBER: 1897–01 | 08 SEP 2006 |
| | STANDARD: JEDEC MO–220 VGGD–8 | |

STYLE 1:

| PIN | |
|---|---|
| 1. | CATHODE |
| 2. | CATHODE |
| 3. | CATHODE |
| 4. | CATHODE |
| 5. | CATHODE |
| 6. | CATHODE |
| 7. | CATHODE |
| 8. | CATHODE |
| 9. | ANODE |
| 10. | ANODE |
| 11. | ANODE |
| 12. | ANODE |
| 13. | ANODE |
| 14. | ANODE |
| 15. | ANODE |
| 16. | ANODE |

STYLE 2:

| PIN | |
|---|---|
| 1. | COMMON DRAIN |
| 2. | COMMON DRAIN |
| 3. | COMMON DRAIN |
| 4. | COMMON DRAIN |
| 5. | COMMON DRAIN |
| 6. | COMMON DRAIN |
| 7. | COMMON DRAIN |
| 8. | COMMON DRAIN |
| 9. | GATE |
| 10. | SOURCE |
| 11. | GATE |
| 12. | SOURCE |
| 13. | GATE |
| 14. | SOURCE |
| 15. | GATE |
| 16. | SOURCE |



| MECHANICAL OUTLINE | PRINT VERSION NOT TO SCALE |
|---|---|

| TITLE: | DOCUMENT NO: 98ASB42431B | REV: T |
|---|---|---|
| 16 LD PDIP | CASE NUMBER: 648-08 | 19 MAY 2005 |
| | STANDARD: NON-JEDEC | |