

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, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	20-TSSOP
Purchase URL	<a href="https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=s9s08sg8e2vtjr">https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=s9s08sg8e2vtjr</a>

Section Number	Title	Page
4.5.3	Program and Erase Command Execution .....	50
4.5.4	Burst Program Execution .....	51
4.5.5	Access Errors .....	53
4.5.6	FLASH Block Protection .....	53
4.5.7	Vector Redirection .....	54
4.6	Security .....	54
4.7	FLASH Registers and Control Bits .....	55
4.7.1	FLASH Clock Divider Register (FCDIV) .....	56
4.7.2	FLASH Options Register (FOPT and NVOPT) .....	57
4.7.3	FLASH Configuration Register (FCNFG) .....	58
4.7.4	FLASH Protection Register (FPROT and NVPROT) .....	58
4.7.5	FLASH Status Register (FSTAT) .....	59
4.7.6	FLASH Command Register (FCMD) .....	60

## Chapter 5

### Resets, Interrupts, and General System Control

5.1	Introduction .....	61
5.2	Features .....	61
5.3	MCU Reset .....	61
5.4	Computer Operating Properly (COP) Watchdog .....	62
5.5	Interrupts .....	63
5.5.1	Interrupt Stack Frame .....	64
5.5.2	Interrupt Vectors, Sources, and Local Masks .....	65
5.6	Low-Voltage Detect (LVD) System .....	67
5.6.1	Power-On Reset Operation .....	67
5.6.2	Low-Voltage Detection (LVD) Reset Operation .....	67
5.6.3	Low-Voltage Warning (LVW) Interrupt Operation .....	67
5.7	Reset, Interrupt, and System Control Registers and Control Bits .....	67
5.7.1	System Reset Status Register (SRS) .....	68
5.7.2	System Background Debug Force Reset Register (SBDFFR) .....	69
5.7.3	System Options Register 1 (SOPT1) .....	70
5.7.4	System Options Register 2 (SOPT2) .....	71
5.7.5	System Device Identification Register (SDIDH, SDIDL) .....	72
5.7.6	System Power Management Status and Control 1 Register (SPMSC1) .....	73
5.7.7	System Power Management Status and Control 2 Register (SPMSC2) .....	74

## Chapter 6

### Parallel Input/Output Control

6.1	Port Data and Data Direction .....	75
6.2	Pull-up, Slew Rate, and Drive Strength .....	76
6.3	Ganged Output .....	77
6.4	Pin Interrupts .....	78

## 1.2 MCU Block Diagram

The block diagram in Figure 1-1 shows the structure of the MC9S08SG8 MCU.

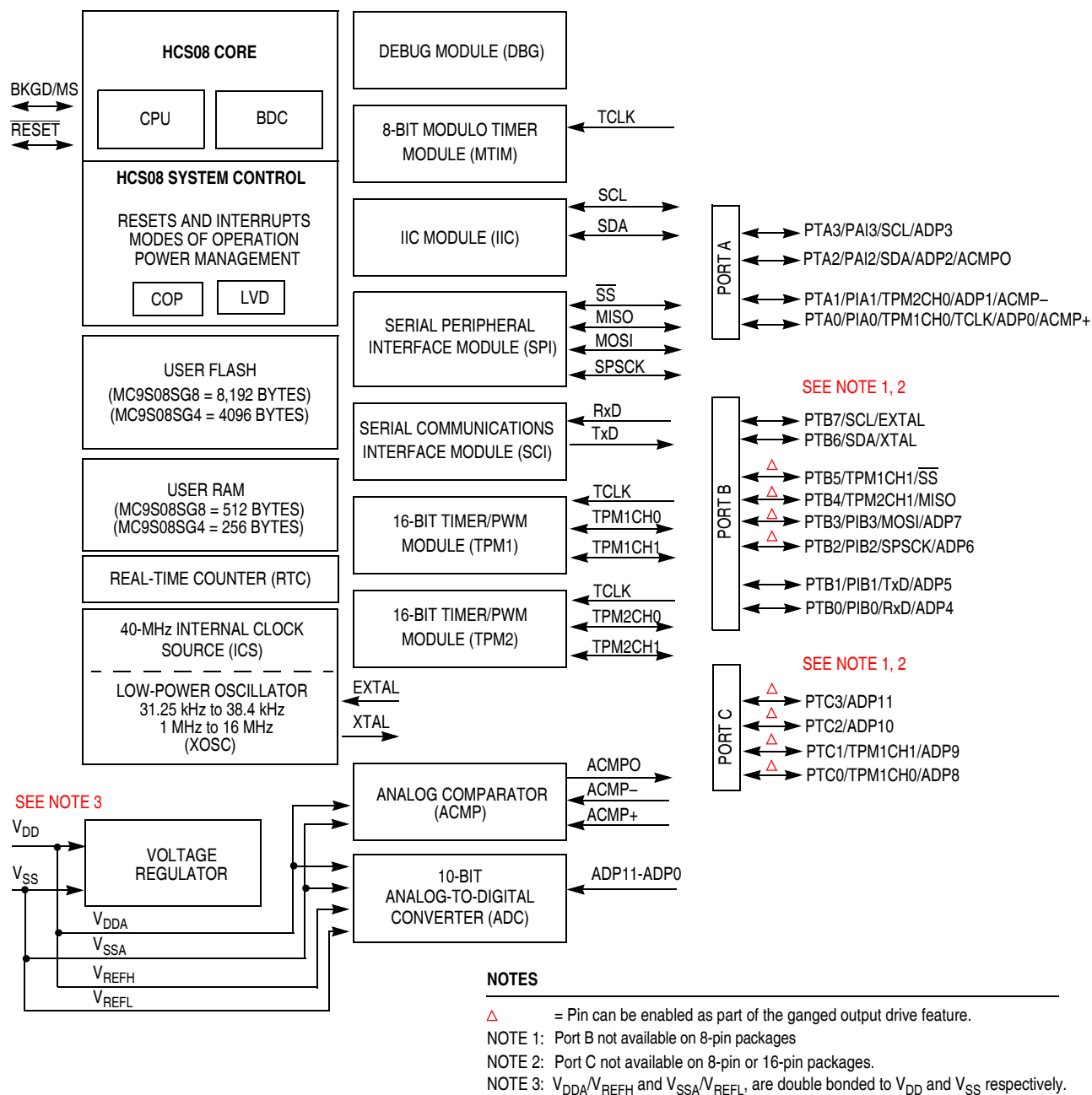


Figure 1-1. MC9S08SG8 Block Diagram

Table 2-1. Pin Availability by Package Pin-Count

Pin Number			Priority					
			← Lowest Highest →					
20-pin	16-pin	8-pin	Port Pin	Alt 1	Alt 2	Alt 3	Alt 4	Alt5
1	1	1						$\overline{\text{RESET}}$
2	2	2					BKGD	MS
3	3	3						$V_{DD}$
4	4	4						$V_{SS}$
5	5	—	PTB7	SCL <sup>1</sup>	EXTAL			
6	6	—	PTB6	SDA <sup>1</sup>	XTAL			
7	7	—	PTB5	TPM1CH1 <sup>2</sup>	$\overline{SS}$	PTC0 <sup>3</sup>		
8	8	—	PTB4	TPM2CH1	MISO	PTC0 <sup>3</sup>		
9	—	—	PTC3			PTC0 <sup>3</sup>	ADP11	
10	—	—	PTC2			PTC0 <sup>3</sup>	ADP10	
11	—	—	PTC1		TPM1CH1 <sup>2</sup>	PTC0 <sup>3</sup>	ADP9	
12	—	—	PTC0		TPM1CH0 <sup>2</sup>	PTC0 <sup>3</sup>	ADP8	
13	9	—	PTB3	PIB3	MOSI	PTC0 <sup>3</sup>	ADP7	
14	10	—	PTB2	PIB2	SPSCK	PTC0 <sup>3</sup>	ADP6	
15	11	—	PTB1	PIB1	TxD		ADP5	
16	12	—	PTB0	PIB0	RxD		ADP4	
17	13	5	PTA3	PIA3	SCL <sup>1</sup>		ADP3	
18	14	6	PTA2	PIA2	SDA <sup>1</sup>		ADP2	ACMPO
19	15	7	PTA1	PIA1	TPM2CH0		ADP1 <sup>4</sup>	ACMP <sup>−4</sup>
20	16	8	PTA0	PIA0	TPM1CH0 <sup>2</sup>	TCLK	ADP0 <sup>4</sup>	ACMP <sup>+</sup> <sup>4</sup>

<sup>1</sup> IIC pins can be repositioned using IICPS in SOPT2, default reset locations are on PTA2 and PTA3.

<sup>2</sup> TPM1CHx pins can be repositioned using TPM1PS in SOPT2, default reset locations are on PTA0 and PTB5.

<sup>3</sup> This port pin is part of the ganged output feature. When pin is enabled for ganged output, it will have priority over all digital modules. The output data, drive strength and slew-rate control of this port pin will follow the configuration for the PTC0 pin, even in 16-pin packages where PTC0 doesn't bond out. Ganged output not available in 8-pin packages.

<sup>4</sup> If ACMP and ADC are both enabled, both will have access to the pin.

## Chapter 5

# Resets, Interrupts, and General System Control

### 5.1 Introduction

This section discusses basic reset and interrupt mechanisms and the various sources of reset and interrupt in the MC9S08SG8. Some interrupt sources from peripheral modules are discussed in greater detail within other sections of this data sheet. This section 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 are not part of on-chip peripheral systems with their own chapters.

### 5.2 Features

Reset and interrupt features include:

- Multiple sources of reset for flexible system configuration and reliable operation
- Reset status register (SRS) to indicate source of most recent reset
- Separate interrupt vector for each module (reduces polling overhead) (see [Table 5-2](#))

### 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 pull-up 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 MC9S08SG8 has the following sources for reset:

- Power-on reset (POR)
- External pin reset (PIN)
- Low-voltage detect (LVD)
- Computer operating properly (COP) timer
- Illegal opcode detect (IOP)
- Illegal address detect (ILAD)
- Background debug forced reset

Each of these sources, with the exception of the background debug forced reset, has an associated bit in the system reset status register (SRS).

### 7.4.3 Wait Mode Operation

The WAIT instruction enables interrupts by clearing the I bit in the CCR. It then halts the clocks to the CPU to reduce overall power consumption while the CPU is waiting for the interrupt or reset event that will wake the CPU from wait mode. When an interrupt or reset event occurs, the CPU clocks will resume and the interrupt or reset event will be processed normally.

If a serial BACKGROUND command is issued to the MCU through the background debug interface while the CPU is in wait mode, CPU clocks will resume and the CPU will enter active background mode where other serial background commands can be processed. This ensures that a host development system can still gain access to a target MCU even if it is in wait mode.

### 7.4.4 Stop Mode Operation

Usually, all system clocks, including the crystal oscillator (when used), are halted during stop mode to minimize power consumption. In such systems, external circuitry is needed to control the time spent in stop mode and to issue a signal to wake the target MCU when it is time to resume processing. Unlike the earlier M68HC05 and M68HC08 MCUs, the HCS08 can be configured to keep a minimum set of clocks running in stop mode. This optionally allows an internal periodic signal to wake the target MCU from stop mode.

When a host debug system is connected to the background debug pin (BKGD) and the ENBDM control bit has been set by a serial command through the background interface (or because the MCU was reset into active background mode), the oscillator is forced to remain active when the MCU enters stop mode. In this case, if a serial BACKGROUND command is issued to the MCU through the background debug interface while the CPU is in stop mode, CPU clocks will resume and the CPU will enter active background mode where other serial background commands can be processed. This ensures that a host development system can still gain access to a target MCU even if it is in stop mode.

Recovery from stop mode depends on the particular HCS08 and whether the oscillator was stopped in stop mode. Refer to the [Modes of Operation](#) chapter for more details.

### 7.4.5 BGND Instruction

The BGND instruction is new to the HCS08 compared to the M68HC08. BGND would not be used in normal user programs because it forces the CPU to stop processing user instructions and enter the active background mode. The only way to resume execution of the user program is through reset or by a host debug system issuing a GO, TRACE1, or TAGGO serial command through the background debug interface.

Software-based breakpoints can be set by replacing an opcode at the desired breakpoint address with the BGND opcode. When the program reaches this breakpoint address, the CPU is forced to active background mode rather than continuing the user program.

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

Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR				
						VH	I	N	Z	C
TXS	Transfer Index Reg. to SP $SP \leftarrow (H:X) - 0x0001$	INH	94	2	f <sub>p</sub>	--	--	--	--	--
WAIT	Enable Interrupts; Wait for Interrupt I bit $\leftarrow$ 0; Halt CPU	INH	8F	2+	f <sub>p</sub> . . .	--	0	--	--	--

**Source Form:** Everything in the source forms columns, *except expressions in italic characters*, is literal information which must appear in the assembly source file exactly as shown. The initial 3- to 5-letter mnemonic and the characters (#, ( ) and +) are always a literal characters.

- n* Any label or expression that evaluates to a single integer in the range 0-7.
- opr8i* Any label or expression that evaluates to an 8-bit immediate value.
- opr16i* Any label or expression that evaluates to a 16-bit immediate value.
- opr8a* Any label or expression that evaluates to an 8-bit direct-page address (0x00xx).
- opr16a* Any label or expression that evaluates to a 16-bit address.
- opr8* Any label or expression that evaluates to an unsigned 8-bit value, used for indexed addressing.
- opr16* Any label or expression that evaluates to a 16-bit value, used for indexed addressing.
- rel* Any label or expression that refers to an address that is within -128 to +127 locations from the start of the next instruction.

#### Operation Symbols:

- A Accumulator
- CCR Condition code register
- H Index register high byte
- M Memory location
- n* Any bit
- opr* Operand (one or two bytes)
- PC Program counter
- PCH Program counter high byte
- PCL Program counter low byte
- rel* Relative program counter offset byte
- SP Stack pointer
- SPL Stack pointer low byte
- X Index register low byte
- & Logical AND
- | Logical OR
- ⊕ Logical EXCLUSIVE OR
- ( ) Contents of
- + Add
- Subtract, Negation (two's complement)
- × Multiply
- ÷ Divide
- # Immediate value
- ← Loaded with
- :

#### CCR Bits:

- V Overflow bit
- H Half-carry bit
- I Interrupt mask
- N Negative bit
- Z Zero bit
- C Carry/borrow bit

#### Addressing Modes:

- DIR Direct addressing mode
- EXT Extended addressing mode
- IMM Immediate addressing mode
- INH Inherent addressing mode
- IX Indexed, no offset addressing mode
- IX1 Indexed, 8-bit offset addressing mode
- IX2 Indexed, 16-bit offset addressing mode
- IX+ Indexed, no offset, post increment addressing mode
- IX1+ Indexed, 8-bit offset, post increment addressing mode
- REL Relative addressing mode
- SP1 Stack pointer, 8-bit offset addressing mode
- SP2 Stack pointer 16-bit offset addressing mode

#### Cycle-by-Cycle Codes:

- f Free cycle. This indicates a cycle where the CPU does not require use of the system buses. An f cycle is always one cycle of the system bus clock and is always a read cycle.
- p Program fetch; read from next consecutive location in program memory
- r Read 8-bit operand
- s Push (write) one byte onto stack
- u Pop (read) one byte from stack
- v Read vector from 0xFFxx (high byte first)
- w Write 8-bit operand

#### CCR Effects:

- b Set or cleared
- Not affected
- U Undefined

### 9.1.5 Features

Features of the ADC module include:

- Linear successive approximation algorithm with 10 bits resolution.
- Up to 28 analog inputs.
- Output formatted in 10- or 8-bit right-justified format.
- Single or continuous conversion (automatic return to idle after single conversion).
- Configurable sample time and conversion speed/power.
- Conversion complete flag and interrupt.
- Input clock selectable from up to four sources.
- Operation in wait or stop3 modes for lower noise operation.
- Asynchronous clock source for lower noise operation.
- Selectable asynchronous hardware conversion trigger.
- Automatic compare with interrupt for less-than, or greater-than or equal-to, programmable value.

### 9.1.6 Block Diagram

Figure 9-2 provides a block diagram of the ADC module



#### 10.1.4.4 FLL Bypassed Internal Low Power (FBILP)

In FLL bypassed internal low power mode, the FLL is disabled and bypassed, and the ICS supplies a clock derived from the internal reference clock. The BDC clock is not available.

#### 10.1.4.5 FLL Bypassed External (FBE)

In FLL bypassed external mode, the FLL is enabled and controlled by an external reference clock, but is bypassed. The ICS supplies a clock derived from the external reference clock. The external reference clock can be an external crystal/resonator supplied by an OSC controlled by the ICS, or it can be another external clock source. The BDC clock is supplied from the FLL.

#### 10.1.4.6 FLL Bypassed External Low Power (FBELP)

In FLL bypassed external low power mode, the FLL is disabled and bypassed, and the ICS supplies a clock derived from the external reference clock. The external reference clock can be an external crystal/resonator supplied by an OSC controlled by the ICS, or it can be another external clock source. The BDC clock is not available.

#### 10.1.4.7 Stop (STOP)

In stop mode the FLL is disabled and the internal or external reference clocks can be selected to be enabled or disabled. The BDC clock is not available and the ICS does not provide an MCU clock source.

## 10.2 External Signal Description

There are no ICS signals that connect off chip.

## 10.3 Register Definition

Figure 10-1 is a summary of ICS registers.

**Table 10-1. ICS Register Summary**

Name		7	6	5	4	3	2	1	0
ICSC1	R	CLKS		RDIV			IREFS	IRCLKEN	IREFSTEN
	W								
ICSC2	R	BDIV		RANGE	HGO	LP	EREFS	ERCLKEN	EREFSTEN
	W								
ICSTRM	R	TRIM							
	W								
ICSSC	R	0	0	0	IREFST	CLKST		OSCINIT	FTRIM
	W								

The CLKS bits can also be changed at anytime, but the RDIV bits must be changed simultaneously so that the resulting frequency stays in the range of 31.25 kHz to 39.0625 kHz. The actual switch to the newly selected clock will not occur until after a few full cycles of the new clock. If the newly selected clock is not available, the previous clock will remain selected.

## 10.4.3 Bus Frequency Divider

The BDIV bits can be changed at anytime and the actual switch to the new frequency will occur immediately.

## 10.4.4 Low Power Bit Usage

The low power bit (LP) is provided to allow the FLL to be disabled and thus conserve power when it is not being used. However, in some applications it may be desirable to enable the FLL and allow it to lock for maximum accuracy before switching to an FLL engaged mode. Do this by writing the LP bit to 0.

## 10.4.5 Internal Reference Clock

When IRCLKEN is set the internal reference clock signal will be presented as ICSIRCLK, which can be used as an additional clock source. The ICSIRCLK frequency can be re-targeted by trimming the period of the internal reference clock. This can be done by writing a new value to the TRIM bits in the ICSTRM register. Writing a larger value will slow down the ICSIRCLK frequency, and writing a smaller value to the ICSTRM register will speed up the ICSIRCLK frequency. The TRIM bits will effect the ICSOUT frequency if the ICS is in FLL engaged internal (FEI), FLL bypassed internal (FBI), or FLL bypassed internal low power (FBILP) mode. The TRIM and FTRIM value will not be affected by a reset.

Until ICSIRCLK is trimmed, programming low reference divider (RDIV) factors may result in ICSOUT frequencies that exceed the maximum chip-level frequency and violate the chip-level clock timing specifications (see the [Device Overview](#) chapter).

If IREFSTEN is set and the IRCLKEN bit is written to 1, the internal reference clock will keep running during stop mode in order to provide a fast recovery upon exiting stop.

All MCU devices are factory programmed with a trim value in a reserved memory location. This value can be copied to the ICSTRM register during reset initialization. The factory trim value does not include the FTRIM bit. For finer precision, the user can trim the internal oscillator in the application and set the FTRIM bit accordingly.

## 10.4.6 Optional External Reference Clock

The ICS module can support an external reference clock with frequencies between 31.25 kHz to 5 MHz in all modes. When the ERCLKEN is set, the external reference clock signal will be presented as ICSECLK, which can be used as an additional clock source. When IREFS = 1, the external reference clock will not be used by the FLL and will only be used as ICSECLK. In these modes, the frequency can be equal to the maximum frequency the chip-level timing specifications will support (see the [Device Overview](#) chapter).

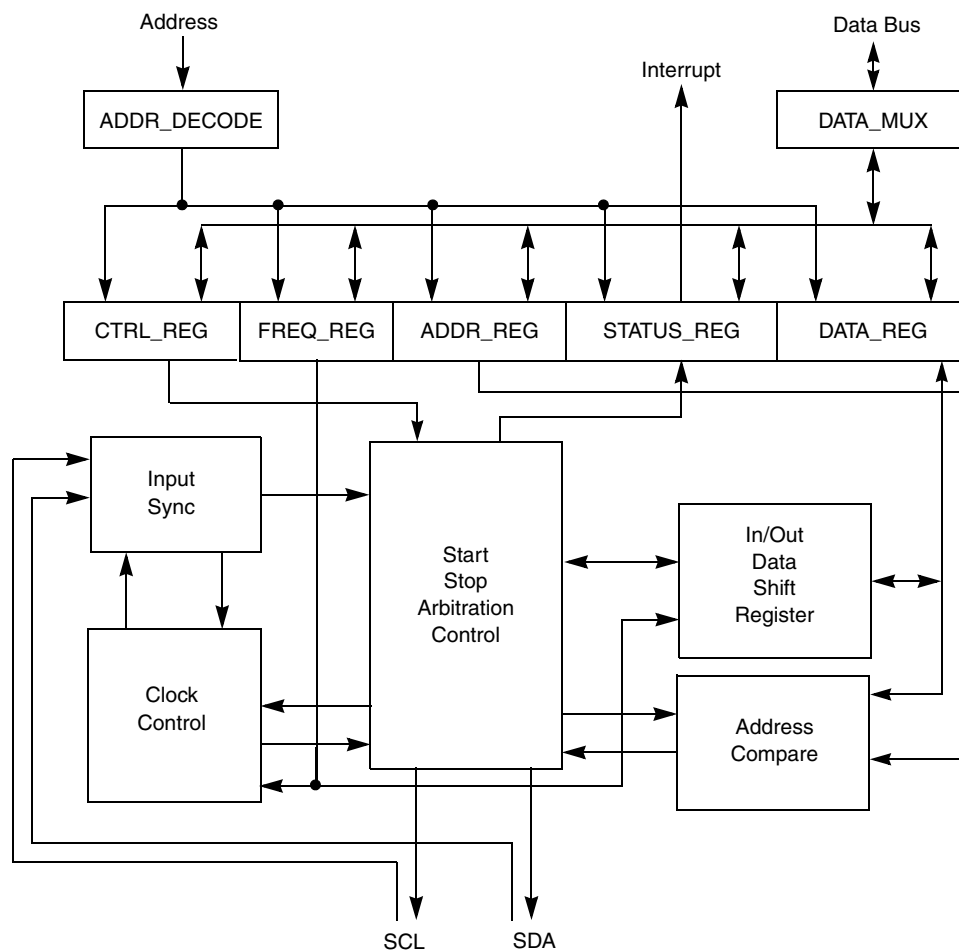


Figure 11-2. IIC Functional Block Diagram

## 11.2 External Signal Description

This section describes each user-accessible pin signal.

### 11.2.1 SCL — Serial Clock Line

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

### 11.2.2 SDA — Serial Data Line

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

## 11.3 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

### 11.3.3 IIC Control Register (IICC1)

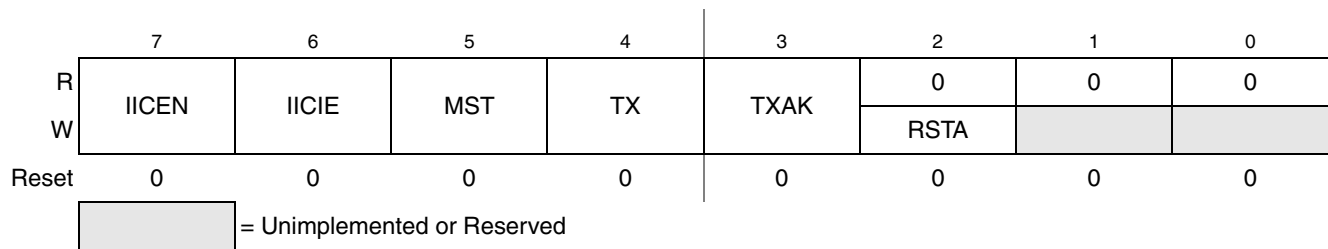


Figure 11-5. IIC Control Register (IICC1)

Table 11-6. IICC1 Field Descriptions

Field	Description
7 IICEN	<b>IIC Enable.</b> The IICEN bit determines whether the IIC module is enabled. 0 IIC is not enabled 1 IIC is enabled
6 IICIE	<b>IIC Interrupt Enable.</b> The IICIE bit determines whether an IIC interrupt is requested. 0 IIC interrupt request not enabled 1 IIC interrupt request enabled
5 MST	<b>Master Mode Select.</b> The MST bit changes from a 0 to a 1 when a start signal is generated on the bus and master mode is selected. When this bit changes from a 1 to a 0 a stop signal is generated and the mode of operation changes from master to slave. 0 Slave mode 1 Master mode
4 TX	<b>Transmit Mode Select.</b> The TX bit selects the direction of master and slave transfers. In master mode, this bit should be set according to the type of transfer required. Therefore, for address cycles, this bit is always high. When addressed as a slave, this bit should be set by software according to the SRW bit in the status register. 0 Receive 1 Transmit
3 TXAK	<b>Transmit Acknowledge Enable.</b> This bit specifies the value driven onto the SDA during data acknowledge cycles for master and slave receivers. 0 An acknowledge signal is sent out to the bus after receiving one data byte 1 No acknowledge signal response is sent
2 RSTA	<b>Repeat start.</b> Writing a 1 to this bit generates a repeated start condition provided it is the current master. This bit is always read as cleared. Attempting a repeat at the wrong time results in loss of arbitration.

### 11.3.4 IIC Status Register (IICS)

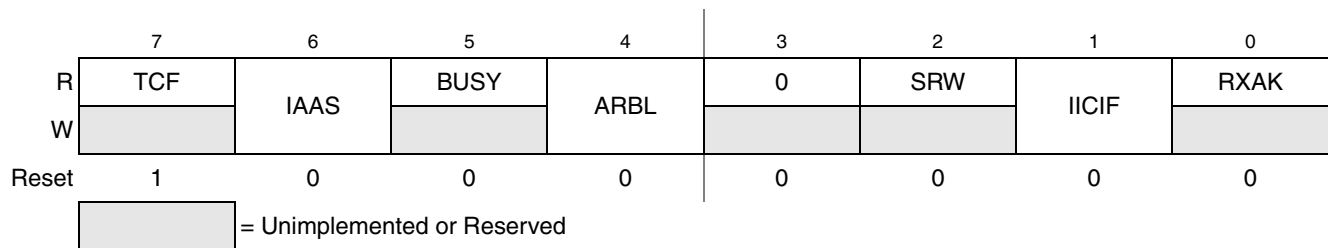
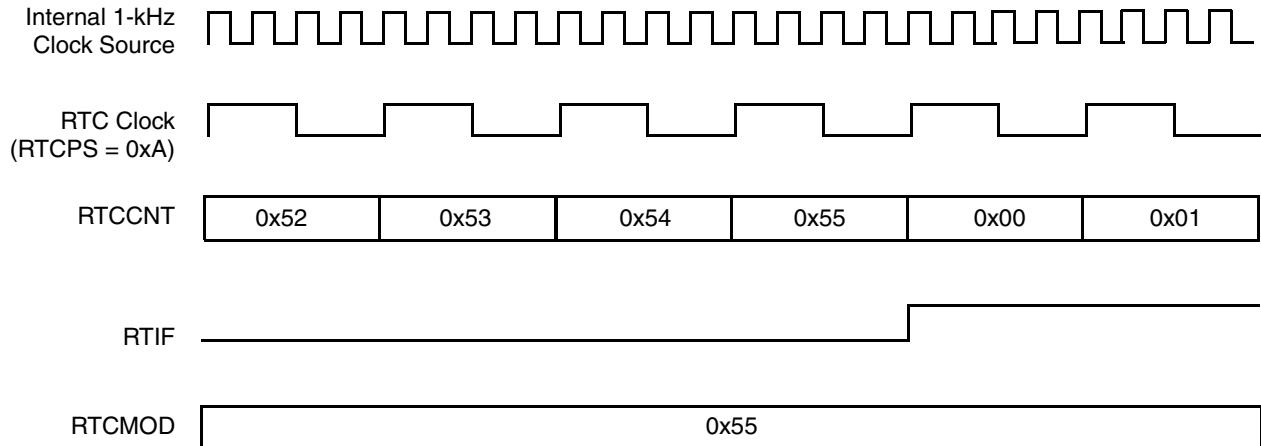


Figure 11-6. IIC Status Register (IICS)



**Figure 13-6. RTC Counter Overflow Example**

In the example of [Figure 13-6](#), the selected clock source is the 1-kHz internal oscillator clock source. The prescaler (RTCPs) is set to 0xA or divide-by-4. The modulo value in the RTCMOD register is set to 0x55. When the counter, RTCCNT, reaches the modulo value of 0x55, the counter overflows to 0x00 and continues counting. The real-time interrupt flag, RTIF, sets when the counter value changes from 0x55 to 0x00. A real-time interrupt is generated when RTIF is set, if RTIE is set.

## 13.5 Initialization/Application Information

This section provides example code to give some basic direction to a user on how to initialize and configure the RTC module. The example software is implemented in C language.

The example below shows how to implement time of day with the RTC using the 1-kHz clock source to achieve the lowest possible power consumption. Because the 1-kHz clock source is not as accurate as a crystal, software can be added for any adjustments. For accuracy without adjustments at the expense of additional power consumption, the external clock (ERCLK) or the internal clock (IRCLK) can be selected with appropriate prescaler and modulo values.

```
/* Initialize the elapsed time counters */
Seconds = 0;
Minutes = 0;
Hours = 0;
Days=0;

/* Configure RTC to interrupt every 1 second from 1-kHz clock source */
RTCMOD.byte = 0x00;
RTCSC.byte = 0x1F;

/*****
Function Name : RTC_ISR
Notes : Interrupt service routine for RTC module.
*****/
```

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.

### 14.3.2.1 Send Break and Queued Idle

The SBK control bit in SCIC2 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 Freescale Semiconductor 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 14-8. 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

### 14.3.3 Receiver Functional Description

In this section, the receiver block diagram ([Figure 14-3](#)) 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 SCIC2. 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 14.3.5.1, “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)

The most common uses of the SPI system include connecting simple shift registers for adding input or output ports or connecting small peripheral devices such as serial A/D or D/A converters. Although [Figure 15-2](#) shows a system where data is exchanged between two MCUs, many practical systems involve simpler connections where data is unidirectionally transferred from the master MCU to a slave or from a slave to the master MCU.

## 15.1.2.2 SPI Module Block Diagram

[Figure 15-3](#) is a block diagram of the SPI module. The central element of the SPI is the SPI shift register. Data is written to the double-buffered transmitter (write to SPID) and gets transferred to the SPI shift register at the start of a data transfer. After shifting in a byte of data, the data is transferred into the double-buffered receiver where it can be read (read from SPID). Pin multiplexing logic controls connections between MCU pins and the SPI module.

When the SPI is configured as a master, the clock output is routed to the SPSCCK pin, the shifter output is routed to MOSI, and the shifter input is routed from the MISO pin.

When the SPI is configured as a slave, the SPSCCK pin is routed to the clock input of the SPI, the shifter output is routed to MISO, and the shifter input is routed from the MOSI pin.

In the external SPI system, simply connect all SPSCCK pins to each other, all MISO pins together, and all MOSI pins together. Peripheral devices often use slightly different names for these pins.

## 15.3 Modes of Operation

### 15.3.1 SPI in Stop Modes

The SPI is disabled in all stop modes, regardless of the settings before executing the STOP instruction. During either stop1 or stop2 mode, the SPI module will be fully powered down. Upon wake-up from stop1 or stop2 mode, the SPI module will be in the reset state. During stop3 mode, clocks to the SPI module are halted. No registers are affected. If stop3 is exited with a reset, the SPI will be put into its reset state. If stop3 is exited with an interrupt, the SPI continues from the state it was in when stop3 was entered.

## 15.4 Register Definition

The SPI has five 8-bit registers to select SPI options, control baud rate, report SPI status, and for transmit/receive data.

Refer to the direct-page register summary in the [Memory](#) chapter of this data sheet for the absolute address assignments for all SPI registers. This section refers to registers and control bits only by their names, and a Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

### 15.4.1 SPI Control Register 1 (SPIC1)

This read/write register includes the SPI enable control, interrupt enables, and configuration options.

	7	6	5	4	3	2	1	0
R								
W								
Reset	0	0	0	0	0	1	0	0

Figure 15-5. SPI Control Register 1 (SPIC1)

Table 15-1. SPIC1 Field Descriptions

Field	Description
7 SPIE	<b>SPI Interrupt Enable (for SPRF and MODF)</b> — This is the interrupt enable for SPI receive buffer full (SPRF) and mode fault (MODF) events. 0 Interrupts from SPRF and MODF inhibited (use polling) 1 When SPRF or MODF is 1, request a hardware interrupt
6 SPE	<b>SPI System Enable</b> — Disabling the SPI halts any transfer that is in progress, clears data buffers, and initializes internal state machines. SPRF is cleared and SPTEF is set to indicate the SPI transmit data buffer is empty. 0 SPI system inactive 1 SPI system enabled
5 SPTIE	<b>SPI Transmit Interrupt Enable</b> — This is the interrupt enable bit for SPI transmit buffer empty (SPTEF). 0 Interrupts from SPTEF inhibited (use polling) 1 When SPTEF is 1, hardware interrupt requested



are used for PWM & output compare operation once normal execution resumes. Writes to the channel registers while BDM is active do not interfere with partial completion of a coherency sequence. After the coherency mechanism has been fully exercised, the channel registers are updated using the buffered values written (while BDM was not active) by the user.

## 16.4 Functional Description

All TPM functions are associated with a central 16-bit counter which allows flexible selection of the clock source and prescale factor. There is also a 16-bit modulo register associated with the main counter.

The CPWMS control bit chooses between center-aligned PWM operation for all channels in the TPM (CPWMS=1) or general purpose timing functions (CPWMS=0) where each channel can independently be configured to operate in input capture, output compare, or edge-aligned PWM mode. The CPWMS control bit is located in the main TPM status and control register because it affects all channels within the TPM and influences the way the main counter operates. (In CPWM mode, the counter changes to an up/down mode rather than the up-counting mode used for general purpose timer functions.)

The following sections describe the main counter and each of the timer operating modes (input capture, output compare, edge-aligned PWM, and center-aligned PWM). Because details of pin operation and interrupt activity depend upon the operating mode, these topics will be covered in the associated mode explanation sections.

### 16.4.1 Counter

All timer functions are based on the main 16-bit counter (TPMxCNTH:TPMxCNTL). This section discusses selection of the clock source, end-of-count overflow, up-counting vs. up/down counting, and manual counter reset.

#### 16.4.1.1 Counter Clock Source

The 2-bit field, CLKS<sub>B</sub>:CLKS<sub>A</sub>, in the timer status and control register (TPMxSC) selects one of three possible clock sources or OFF (which effectively disables the TPM). See [Table 16-6](#). After any MCU reset, CLKS<sub>B</sub>:CLKS<sub>A</sub>=0:0 so no clock source is selected, and the TPM is in a very low power state. These control bits may be read or written at any time and disabling the timer (writing 00 to the CLKS<sub>B</sub>:CLKS<sub>A</sub> field) does not affect the values in the counter or other timer registers.

### 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.

	Commands begin with an 8-bit hexadecimal command code in the host-to-target direction (most significant bit first)
/	= separates parts of the command
d	= delay 16 target BDC clock cycles
AAAA	= a 16-bit address in the host-to-target direction
RD	= 8 bits of read data in the target-to-host direction
WD	= 8 bits of write data in the host-to-target direction
RD16	= 16 bits of read data in the target-to-host direction
WD16	= 16 bits of write data in the host-to-target direction
SS	= the contents of BDCSCR in the target-to-host direction (STATUS)
CC	= 8 bits of write data for BDCSCR in the host-to-target direction (CONTROL)
RBKP	= 16 bits of read data in the target-to-host direction (from BDCBKPT breakpoint register)
WBKP	= 16 bits of write data in the host-to-target direction (for BDCBKPT breakpoint register)

Table A-6. DC Characteristics (continued)

Num	C	Characteristic		Symbol	Condition	Min	Typ <sup>1</sup>	Max	Unit	Temp Rated <sup>2</sup>	
										Stand ard	AEC Grade 0
11	P	Pullup or Pulldown <sup>3</sup> resistors; when enabled		R <sub>PU</sub> , R <sub>PD</sub>		17	37	52	kΩ	x	x
	C	RESET <sup>4</sup>				R <sub>PU</sub>	17	37	52	kΩ	x
12	D	DC injection current <sup>5, 6, 7, 8</sup>		I <sub>IC</sub>	V <sub>IN</sub> > V <sub>DD</sub>	0	—	2	mA	x	x
		Single pin limit	0			—	−0.2	mA	x	x	
		Total MCU limit, includes sum of all stressed pins			V <sub>IN</sub> > V <sub>DD</sub>	0	—	25	mA	x	x
					V <sub>IN</sub> < V <sub>SS</sub>	0	—	−5	mA	x	x
13	D	Input Capacitance, all pins		C <sub>In</sub>		—	—	8	pF	x	x
14	D	RAM retention voltage		V <sub>RAM</sub>		—	0.6	1.0	V	x	x
15	D	POR re-arm voltage <sup>9</sup>		V <sub>POR</sub>		0.9	1.4	2.0	V	x	x
16	D	POR re-arm time <sup>10</sup>		t <sub>POR</sub>		10	—	—	μs	x	x
17	P	Low-voltage detection threshold — high range	V <sub>DD</sub> falling	V <sub>LVD1</sub>		3.9 3.88	4.0 4.0	4.1 4.12	V	x	x
			V <sub>DD</sub> rising			4.0 3.98	4.1 4.1	4.2 4.22		x	x
18	P	Low-voltage detection threshold — low range <sup>11, 12</sup>		V <sub>LVD0</sub>		2.48 2.54	2.56 2.62	2.64 2.70	V	x x	x x
		V <sub>DD</sub> falling V <sub>DD</sub> rising									
19	P	Low-voltage warning threshold — high range 1	V <sub>DD</sub> falling	V <sub>LVW3</sub>		4.5 4.48	4.6 4.6	4.7 4.72	V	x	x
			V <sub>DD</sub> rising			4.6 4.58	4.7 4.7	4.8 4.82		x	x
20	P	Low-voltage warning threshold — high range 0	V <sub>DD</sub> falling	V <sub>LVW2</sub>		4.2 4.18	4.3 4.3	4.4 4.42	V	x	x
			V <sub>DD</sub> rising			4.3 4.28	4.4 4.4	4.5 4.52		x	x
21	P	Low-voltage warning threshold low range 1		V <sub>LVW1</sub>		2.84 2.90	2.92 2.98	3.00 3.06	V	x x	x x
		V <sub>DD</sub> falling V <sub>DD</sub> rising									
22	P	Low-voltage warning threshold — low range 0		V <sub>LVW0</sub>		2.66 2.72	2.74 2.80	2.82 2.88	V	x x	x x
		V <sub>DD</sub> falling V <sub>DD</sub> rising									

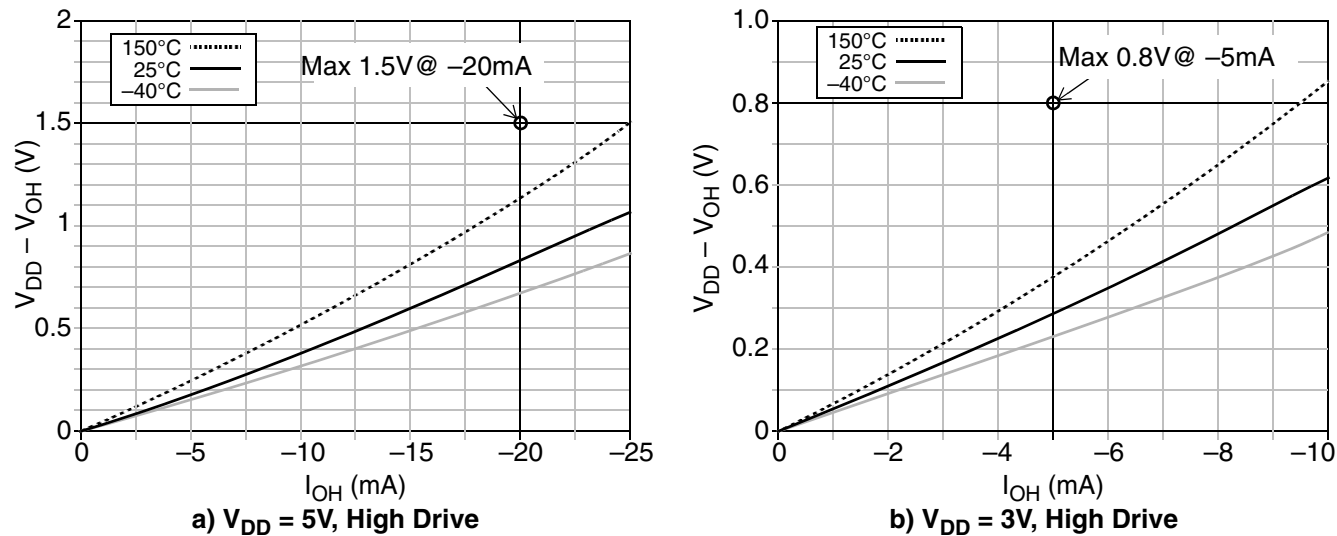


Figure A-3. Typical  $V_{DD} - V_{OH}$  vs  $I_{OH}$ , High Drive Strength

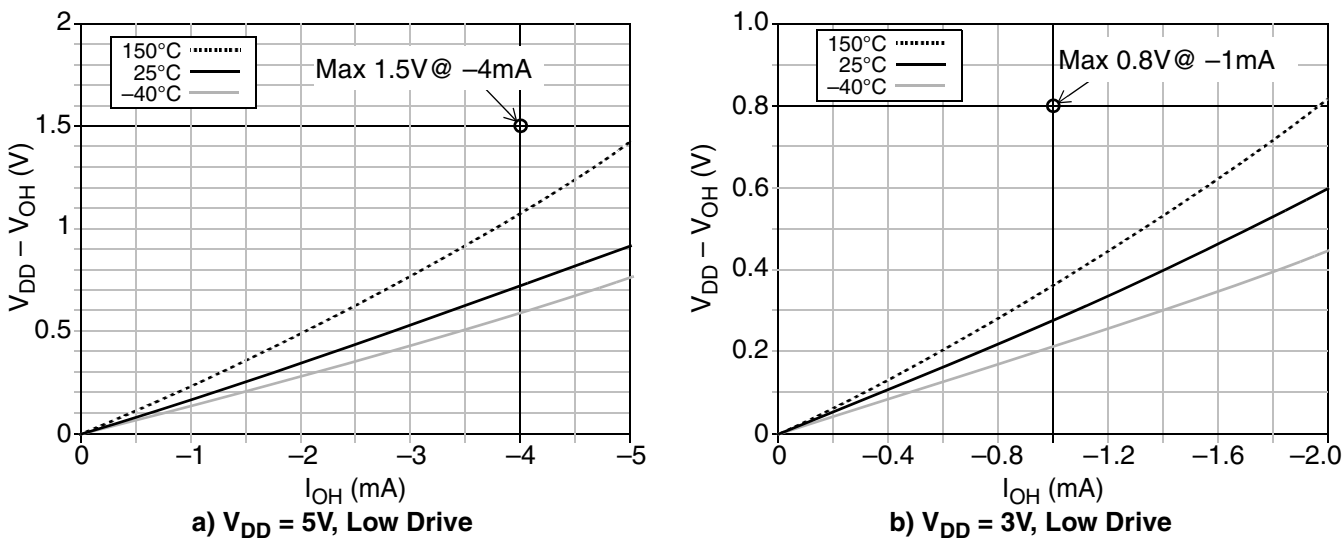


Figure A-4. Typical  $V_{DD} - V_{OH}$  vs  $I_{OH}$ , Low Drive Strength

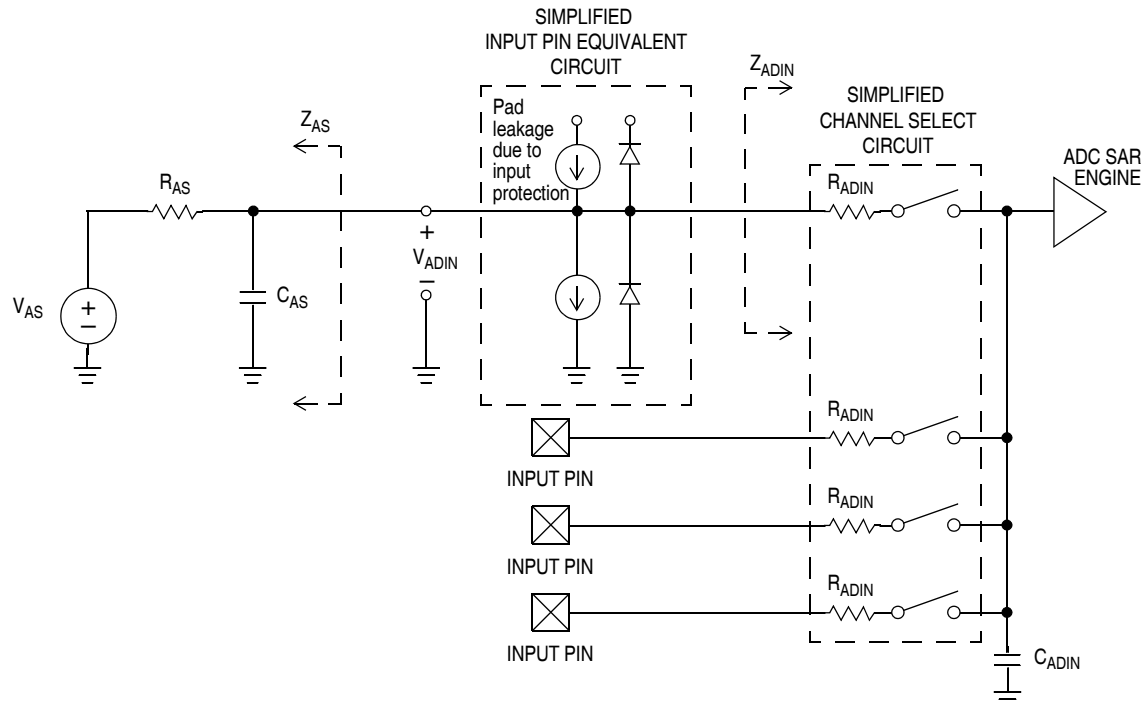


Figure A-9. ADC Input Impedance Equivalency Diagram

Table A-12. ADC Characteristics

Characteristic	Conditions	C	Symb	Min	Typ <sup>1</sup>	Max	Unit	Temp Rated <sup>2</sup>		Comment
								Stand ard	AEC Grade 0	
Supply current	ADLPC=1 ADLSMP=1 ADCO=1	T	$I_{DD} + I_{DDAD}$	—	133	—	$\mu A$	x	x	ADC current only
Supply current	ADLPC=1 ADLSMP=0 ADCO=1	T	$I_{DD} + I_{DDAD}$	—	218	—	$\mu A$	x	x	ADC current only
Supply current	ADLPC=0 ADLSMP=1 ADCO=1	T	$I_{DD} + I_{DDAD}$	—	327	—	$\mu A$	x	x	ADC current only
Supply current	ADLPC=0 ADLSMP=0 ADCO=1	P	$I_{DD} + I_{DDAD}$	—	0.582	1	mA	x	x	ADC current only
ADC asynchronous clock source	High speed (ADLPC=0)	P	$f_{ADACK}$	2	3.3	5	MHz	x	x	$t_{ADACK} = 1/f_{ADACK}$
	Low power (ADLPC=1)			1.25	2	3.3		x	x	