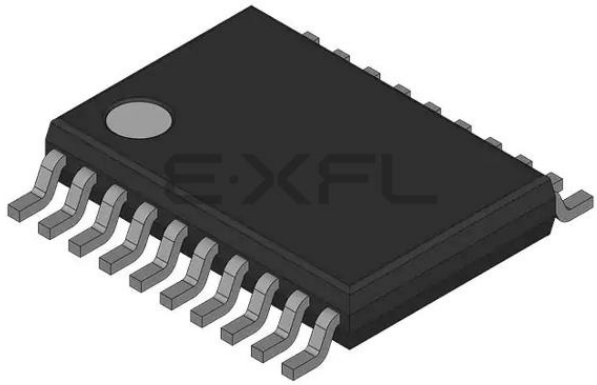

Welcome to E-XFL.COM

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	Obsolete
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	20-TSSOP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08sl16ctj

MC9S08EL32 Features

8-Bit HCS08 Central Processor Unit (CPU)

- 40-MHz HCS08 CPU (central processor unit)
- HC08 instruction set with added BGND instruction
- Support for up to 32 interrupt/reset sources

On-Chip Memory

- FLASH read/program/erase over full operating voltage and temperature
- EEPROM in-circuit programmable memory; program and erase while executing FLASH; erase abort
- Random-access memory (RAM)
- Security circuitry to prevent unauthorized access to RAM and NVM contents

Power-Saving Modes

- Two very low-power stop modes
- Reduced power wait mode
- Very low-power real-time interrupt for use in run, wait, and stop

Clock Source Options

- Oscillator (XOSC) — Loop-control Pierce oscillator; Crystal or ceramic resonator range of 31.25 kHz to 38.4 kHz or 1 MHz to 16 MHz
- Internal clock source (ICS) — Contains a frequency-locked loop (FLL) controlled by internal or external reference; precision trimming of internal reference allows 0.2% resolution and 2% deviation over temperature and voltage; supports bus frequencies from 2–20 MHz

System Protection

- Watchdog computer operating properly (COP) reset with option to run from dedicated 1-kHz internal clock source or bus clock
- Low-voltage detection with reset or interrupt; selectable trip points
- Illegal opcode detection with reset
- Illegal address detection with reset
- FLASH and EEPROM block protect

Development Support

- Single-wire background debug interface
- Breakpoint capability allows single breakpoint setting during in-circuit debugging (plus two more breakpoints in the on-chip debug module)
- In-circuit emulation (ICE) debug module — contains two comparators and nine trigger modes; eight-deep FIFO for storing change-of-flow address and event-only data; supports both tag and force breakpoints

Peripherals

- **ADC** — 16-channel, 10-bit resolution, 2.5 μs conversion time, automatic compare function, temperature sensor, internal bandgap reference channel; runs in stop3
- **ACMPx** — Two analog comparators with selectable interrupt on rising, falling, or either edge of comparator output; compare option to fixed internal bandgap reference voltage; output can optionally be routed to TPM module; runs in stop3
- **SCI** — Full duplex non-return to zero (NRZ); LIN master extended break generation; LIN slave extended break detection; wake-up on active edge
- **SLIC** — Supports LIN 2.0 and SAE J2602 protocols; up to 120 kbps, full LIN message buffering, automatic bit rate and frame synchronization, checksum generation and verification, UART-like byte transfer mode
- **SPI** — Full-duplex or single-wire bidirectional; double-buffered transmit and receive; master or slave mode; MSB-first or LSB-first shifting
- **IIC** — Up to 100 kbps with maximum bus loading; Multi-master operation; Programmable slave address; Interrupt driven byte-by-byte data transfer
- **TPMx** — One 4-channel (TPM1) and one 2-channel (TPM2); selectable input capture, output compare, or buffered edge- or center-aligned PWM on each channel
- **RTC** — 8-bit modulus real-time counter with binary or decimal based prescaler; external clock source for precise time base, time-of-day, calendar, or task scheduling functions; free running on-chip low power oscillator (1 kHz) for cyclic wake-up without external components

Input/Output

- 22 general purpose I/O pins
- 16 interrupt pins with selectable polarity
- Hysteresis and configurable pull up device on all input pins; Configurable slew rate and drive strength on all output pins.

Package Options

- 28-TSSOP
- 20-TSSOP

Chapter 1
Device Overview

The MC9S08EL32 Series and MC9S08SL16 Series are members of the low-cost, high-performance HCS08 Family of 8-bit microcontroller units (MCUs). All MCUs in the family use the enhanced HCS08 core and are available with a variety of modules, memory sizes, memory types, and package types.

1.1 Devices in the MC9S08EL32 Series and MC9S08SL16 Series

Table 1-1 summarizes the feature set available in the MC9S08EL32 Series and MC9S08SL16 Series of MCUs.

Table 1-1. MC9S08EL32 Series and MC9S08SL16 Series Features by MCU and Package

Feature	9S08EL32		9S08EL16		9S08SL16		9S08SL8	
FLASH size (bytes)	32768		16384		16384		8192	
RAM size (bytes)	1024				512			
EEPROM size (bytes)	512				256			
Pin quantity	28	20	28	20	28	20	28	20
Package type	TSSOP	TSSOP	TSSOP	TSSOP	TSSOP	TSSOP	TSSOP	TSSOP
Port Interrupts	16	12	16	12	16	12	16	12
ACMP1	yes				yes			
ACMP2	yes	no	yes	no	no			
ADC channels	16	12	16	12	16	12	16	12
DBG	yes				yes			
ICS	yes				yes			
IIC	yes				yes			
RTC	yes				yes			
SCI	yes				yes			
SLIC	yes				yes			
SPI	yes				yes			
TPM1 channels	4				2			
TPM2 channels	2				2			
XOSC	yes				yes			

for warning and detection. The LVD circuit is enabled when LVDE in SPMSC1 is set to 1. The LVD is disabled upon entering any of the stop modes unless LVDSE is set in SPMSC1. If LVDSE and LVDE are both set, then the MCU cannot enter stop2, and the current consumption in stop3 with the LVD enabled will be higher.

5.6.1 Power-On Reset Operation

When power is initially applied to the MCU, or when the supply voltage drops below the power-on reset rearm voltage level, V_{POR}, the POR circuit will cause a reset condition. As the supply voltage rises, the LVD circuit will hold the MCU in reset until the supply has risen above the low voltage detection low threshold, V_{LVDL}. Both the POR bit and the LVD bit in SRS are set following a POR.

5.6.2 Low-Voltage Detection (LVD) Reset Operation

The LVD can be configured to generate a reset upon detection of a low voltage condition by setting LVDRE to 1. The low voltage detection threshold is determined by the LVDV bit. After an LVD reset has occurred, the LVD system will hold the MCU in reset until the supply voltage has risen above the low voltage detection threshold. The LVD bit in the SRS register is set following either an LVD reset or POR.

5.6.3 Low-Voltage Warning (LVW) Interrupt Operation

The LVD system has a low voltage warning flag to indicate to the user that the supply voltage is approaching the low voltage condition. When a low voltage warning condition is detected and is configured for interrupt operation (LVWIE set to 1), LVWF in SPMSC1 will be set and an LVW interrupt request will occur.

5.7 Reset, Interrupt, and System Control Registers and Control Bits

One 8-bit register in the direct page register space and eight 8-bit registers in the high-page register space are related to reset and interrupt systems.

Refer to Table 4-2 and Table 4-3 in Chapter 4, “Memory,” of this data sheet for the absolute address assignments for all registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

Some control bits in the SOPT1 and SPMSC2 registers are related to modes of operation. Although brief descriptions of these bits are provided here, the related functions are discussed in greater detail in Chapter 3, “Modes of Operation.”

6.3 Pin Interrupts

Port A[3:0], port B[3:0] and port C pins can be configured as external interrupt inputs and as an external mean of waking the MCU from stop3 or wait low-power modes.

The block diagram for each port interrupt logic is shown Figure 6-2.

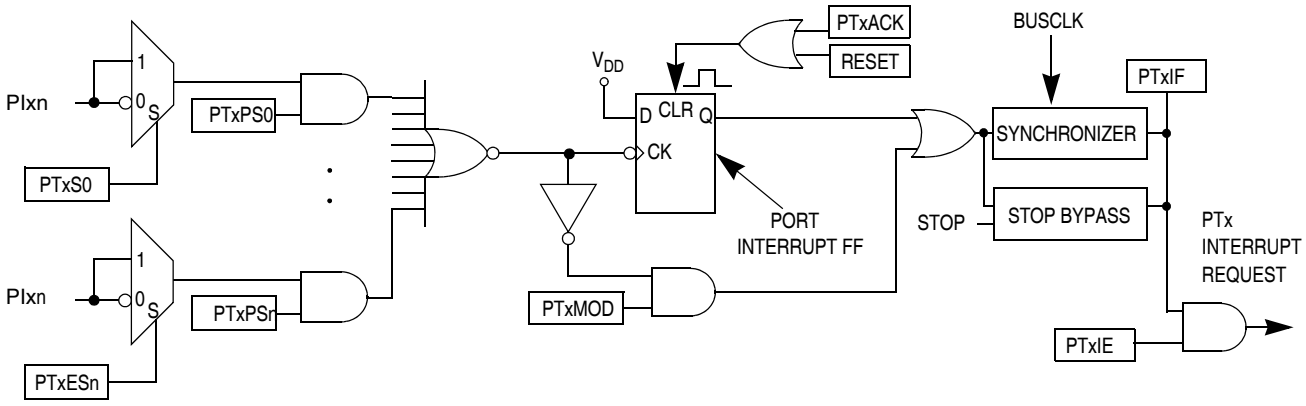

Figure 6-2. Port Interrupt Block Diagram

Writing to the PTxPSn bits in the port interrupt pin select register (PTxPS) independently enables or disables each port pin. Each port can be configured as edge sensitive or edge and level sensitive based on the PTxMOD bit in the port interrupt status and control register (PTxSC). Edge sensitivity can be software programmed to be either falling or rising; the level can be either low or high. The polarity of the edge or edge and level sensitivity is selected using the PTxESn bits in the port interrupt edge select register (PTxES).

Synchronous logic is used to detect edges. Prior to detecting an edge, enabled port inputs must be at the deasserted logic level. A falling edge is detected when an enabled port input signal is seen as a logic 1 (the deasserted level) during one bus cycle and then a logic 0 (the asserted level) during the next cycle. A rising edge is detected when the input signal is seen as a logic 0 during one bus cycle and then a logic 1 during the next cycle.

6.3.1 Edge Only Sensitivity

A valid edge on an enabled port pin will set PTxIF in PTxSC. If PTxIE in PTxSC is set, an interrupt request will be presented to the CPU. Clearing of PTxIF is accomplished by writing a 1 to PTxACK in PTxSC.

6.3.2 Edge and Level Sensitivity

A valid edge or level on an enabled port pin will set PTxIF in PTxSC. If PTxIE in PTxSC is set, an interrupt request will be presented to the CPU. Clearing of PTxIF is accomplished by writing a 1 to PTxACK in PTxSC provided all enabled port inputs are at their deasserted levels. PTxIF will remain set if any enabled port pin is asserted while attempting to clear by writing a 1 to PTxACK.

6.5.1.7 Port A Interrupt Pin Select Register (PTAPS)

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTAPS3	PTAPS2	PTAPS1	PTAPS0
W								
Reset:	0	0	0	0	0	0	0	0

Figure 6-9. Port A Interrupt Pin Select Register (PTAPS)

Table 6-7. PTAPS Register Field Descriptions

Field	Description
3:0 PTAPS[3:0]	**Port A Interrupt Pin Selects** — Each of the PTAPSn bits enable the corresponding port A interrupt pin. 0 Pin not enabled as interrupt. 1 Pin enabled as interrupt.

6.5.1.8 Port A Interrupt Edge Select Register (PTAES)

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTAES3	PTAES2	PTAES1	PTAES0
W								
Reset:	0	0	0	0	0	0	0	0

Figure 6-10. Port A Edge Select Register (PTAES)

Table 6-8. PTAES Register Field Descriptions

Field	Description
3:0 PTAES[3:0]	**Port A Edge Selects** — Each of the PTAESn bits serves a dual purpose by selecting the polarity of the active interrupt edge as well as selecting a pull-up or pull-down device if enabled. 0 A pull-up device is connected to the associated pin interrupt and detects falling edge/low level for interrupt generation. 1 A pull-down device is connected to the associated pin interrupt and detects rising edge/high level for interrupt generation.

6.5.2.3 Port B Pull Enable Register (PTBPE)

	7	6	5	4	3	2	1	0
R W	PTBPE7	PTBPE6	PTBPE5	PTBPE4	PTBPE3	PTBPE2	PTBPE1	PTBPE0
Reset:	0	0	0	0	0	0	0	0

Figure 6-13. Internal Pull Enable for Port B Register (PTBPE)

Table 6-11. PTBPE Register Field Descriptions

Field	Description
7:0 PTBPE[7:0]	**Internal Pull Enable for Port B Bits** — Each of these control bits determines if the internal pull-up or internal (pin interrupt only) pull-down device is enabled for the associated PTB pin. For port B pins that are configured as outputs, these bits have no effect and the internal pull devices are disabled. 0 Internal pull-up/pull-down device disabled for port B bit n. 1 Internal pull-up/pull-down device enabled for port B bit n.

6.5.2.4 Port B Slew Rate Enable Register (PTBSE)

	7	6	5	4	3	2	1	0
R W	PTBSE7	PTBSE6	PTBSE5	PTBSE4	PTBSE3	PTBSE2	PTBSE1	PTBSE0
Reset:	0	0	0	0	0	0	0	0

Figure 6-14. Slew Rate Enable for Port B Register (PTBSE)

Table 6-12. PTBSE Register Field Descriptions

Field	Description
7:0 PTBSE[7:0]	**Output Slew Rate Enable for Port B Bits** — Each of these control bits determines if the output slew rate control is enabled for the associated PTB pin. For port B pins that are configured as inputs, these bits have no effect. 0 Output slew rate control disabled for port B bit n. 1 Output slew rate control enabled for port B bit n.

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.

8.4.3 Bus Frequency Divider

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

8.4.4 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.

8.4.5 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.

8.4.6 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 ICSERCLK, 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 ICSERCLK. In these modes, the frequency can be equal to the maximum frequency the chip-level timing specifications will support (see the Device Overview chapter).

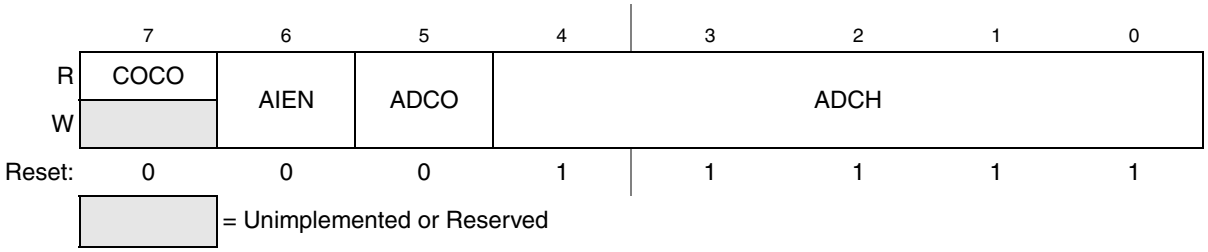

Figure 10-3. Status and Control Register (ADCSC1)

Table 10-3. ADCSC1 Register Field Descriptions

Field	Description
7 COCO	**Conversion Complete Flag** — The COCO flag is a read-only bit which is set each time a conversion is completed when the compare function is disabled (ACFE = 0). When the compare function is enabled (ACFE = 1) the COCO flag is set upon completion of a conversion only if the compare result is true. This bit is cleared whenever ADCSC1 is written or whenever ADCRL is read. 0 Conversion not completed 1 Conversion completed
6 AIEN	**Interrupt Enable** — AIEN is used to enable conversion complete interrupts. When COCO becomes set while AIEN is high, an interrupt is asserted. 0 Conversion complete interrupt disabled 1 Conversion complete interrupt enabled
5 ADCO	**Continuous Conversion Enable** — ADCO is used to enable continuous conversions. 0 One conversion following a write to the ADCSC1 when software triggered operation is selected, or one conversion following assertion of ADHWT when hardware triggered operation is selected. 1 Continuous conversions initiated following a write to ADCSC1 when software triggered operation is selected. Continuous conversions are initiated by an ADHWT event when hardware triggered operation is selected.
4:0 ADCH	**Input Channel Select** — The ADCH bits form a 5-bit field which is used to select one of the input channels. The input channels are detailed in Figure 10-4. The successive approximation converter subsystem is turned off when the channel select bits are all set to 1. This feature allows for explicit disabling of the ADC and isolation of the input channel from all sources. Terminating continuous conversions this way will prevent an additional, single conversion from being performed. It is not necessary to set the channel select bits to all 1s to place the ADC in a low-power state when continuous conversions are not enabled because the module automatically enters a low-power state when a conversion completes.

Figure 10-4. Input Channel Select

ADCH	Input Select
00000	AD0
00001	AD1
00010	AD2
00011	AD3
00100	AD4
00101	AD5
00110	AD6
00111	AD7

ADCH	Input Select
10000	AD16
10001	AD17
10010	AD18
10011	AD19
10100	AD20
10101	AD21
10110	AD22
10111	AD23

Table 10-8. Input Clock Select

ADICLK	Selected Clock Source
00	Bus clock
01	Bus clock divided by 2
10	Alternate clock (ALTCLK)
11	Asynchronous clock (ADACK)

10.3.8 Pin Control 1 Register (APCTL1)

The pin control registers are used to disable the I/O port control of MCU pins used as analog inputs. APCTL1 is used to control the pins associated with channels 0–7 of the ADC module.

	7	6	5	4	3	2	1	0
R W	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
Reset:	0	0	0	0	0	0	0	0

Figure 10-11. Pin Control 1 Register (APCTL1)

Table 10-9. APCTL1 Register Field Descriptions

Field	Description
7 ADPC7	**ADC Pin Control 7** — ADPC7 is used to control the pin associated with channel AD7. 0 AD7 pin I/O control enabled 1 AD7 pin I/O control disabled
6 ADPC6	**ADC Pin Control 6** — ADPC6 is used to control the pin associated with channel AD6. 0 AD6 pin I/O control enabled 1 AD6 pin I/O control disabled
5 ADPC5	**ADC Pin Control 5** — ADPC5 is used to control the pin associated with channel AD5. 0 AD5 pin I/O control enabled 1 AD5 pin I/O control disabled
4 ADPC4	**ADC Pin Control 4** — ADPC4 is used to control the pin associated with channel AD4. 0 AD4 pin I/O control enabled 1 AD4 pin I/O control disabled
3 ADPC3	**ADC Pin Control 3** — ADPC3 is used to control the pin associated with channel AD3. 0 AD3 pin I/O control enabled 1 AD3 pin I/O control disabled
2 ADPC2	**ADC Pin Control 2** — ADPC2 is used to control the pin associated with channel AD2. 0 AD2 pin I/O control enabled 1 AD2 pin I/O control disabled

Table 11-3. IICF Field Descriptions

Field	Description
7–6 MULT	**IIC Multiplier Factor**. The MULT bits define the multiplier factor, mul. This factor, along with the SCL divider, generates the IIC baud rate. The multiplier factor mul as defined by the MULT bits is provided below. 00 mul = 01 01 mul = 02 10 mul = 04 11 Reserved
5–0 ICR	**IIC Clock Rate**. The ICR bits are used to prescale the bus clock for bit rate selection. These bits and the MULT bits determine the IIC baud rate, the SDA hold time, the SCL Start hold time, and the SCL Stop hold time. Table 11-5 provides the SCL divider and hold values for corresponding values of the ICR. The SCL divider multiplied by multiplier factor mul generates IIC baud rate. $\text{IIC baud rate} = \frac{\text{bus speed (Hz)}}{\text{mul} \times \text{SCLdivider}}$ ***Eqn. 11-1*** SDA hold time is the delay from the falling edge of SCL (IIC clock) to the changing of SDA (IIC data). SDA hold time = bus period (s) × mul × SDA hold value ***Eqn. 11-2*** SCL start hold time is the delay from the falling edge of SDA (IIC data) while SCL is high (Start condition) to the falling edge of SCL (IIC clock). SCL Start hold time = bus period (s) × mul × SCL Start hold value ***Eqn. 11-3*** SCL stop hold time is the delay from the rising edge of SCL (IIC clock) to the rising edge of SDA SDA (IIC data) while SCL is high (Stop condition). SCL Stop hold time = bus period (s) × mul × SCL Stop hold value ***Eqn. 11-4***

For example, if the bus speed is 8 MHz, the table below shows the possible hold time values with different ICR and MULT selections to achieve an IIC baud rate of 100kbps.

Table 11-4. Hold Time Values for 8 MHz Bus Speed

MULT	ICR	Hold Times (μs)		
		SDA	SCL Start	SCL Stop
0x2	0x00	3.500	3.000	5.500
0x1	0x07	2.500	4.000	5.250
0x1	0x0B	2.250	4.000	5.250
0x0	0x14	2.125	4.250	5.125
0x0	0x18	1.125	4.750	5.125

11.4.1.2 Slave Address Transmission

The first byte of data transferred immediately after the start signal is the slave address transmitted by the master. This is a seven-bit calling address followed by a R/$\overline{W}$ bit. The R/$\overline{W}$ bit tells the slave the desired direction of data transfer.

1 = Read transfer, the slave transmits data to the master.
0 = Write transfer, the master transmits data to the slave.

Only the slave with a calling address that matches the one transmitted by the master responds by sending back an acknowledge bit. This is done by pulling the SDA low at the ninth clock (see Figure 11-9).

No two slaves in the system may have the same address. If the IIC module is the master, it must not transmit an address equal to its own slave address. The IIC cannot be master and slave at the same time. However, if arbitration is lost during an address cycle, the IIC reverts to slave mode and operates correctly even if it is being addressed by another master.

11.4.1.3 Data Transfer

Before successful slave addressing is achieved, the data transfer can proceed byte-by-byte in a direction specified by the R/$\overline{W}$ bit sent by the calling master.

All transfers that come after an address cycle are referred to as data transfers, even if they carry sub-address information for the slave device

Each data byte is 8 bits long. Data may be changed only while SCL is low and must be held stable while SCL is high as shown in Figure 11-9. There is one clock pulse on SCL for each data bit, the msb being transferred first. Each data byte is followed by a 9th (acknowledge) bit, which is signalled from the receiving device. An acknowledge is signalled by pulling the SDA low at the ninth clock. In summary, one complete data transfer needs nine clock pulses.

If the slave receiver does not acknowledge the master in the ninth bit time, the SDA line must be left high by the slave. The master interprets the failed acknowledge as an unsuccessful data transfer.

If the master receiver does not acknowledge the slave transmitter after a data byte transmission, the slave interprets this as an end of data transfer and releases the SDA line.

In either case, the data transfer is aborted and the master does one of two things:

- Relinquishes the bus by generating a stop signal.
- Commences a new calling by generating a repeated start signal.

11.4.1.4 Stop Signal

The master can terminate the communication by generating a stop signal to free the bus. However, the master may generate a start signal followed by a calling command without generating a stop signal first. This is called repeated start. A stop signal is defined as a low-to-high transition of SDA while SCL at logical 1 (see Figure 11-9).

The master can generate a stop even if the slave has generated an acknowledge at which point the slave must release the bus.

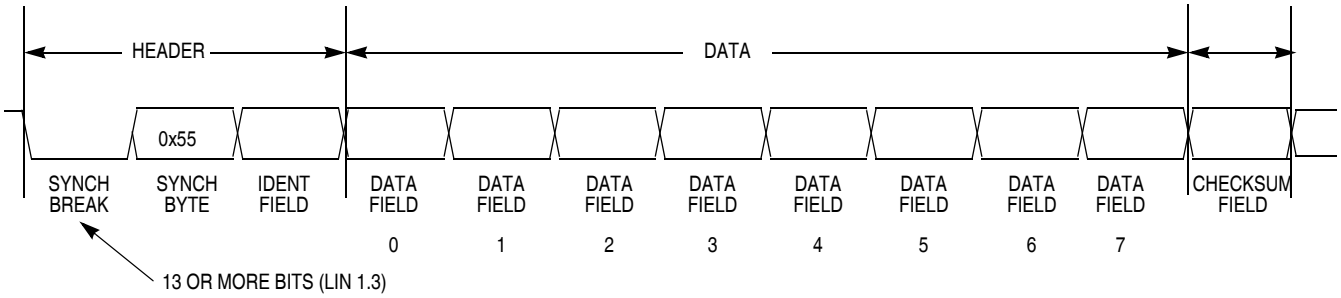

Figure 12-13. Typical LIN MESSAGE FRAME

12.6.1 LIN Message Frame Header

The HEADER section of all LIN messages is transmitted by the master node in the network and contains synchronization data, as well as the identifier to define what information is to be contained in the message frame. Formally, the header is comprised of three parts:

1. SYNCH BREAK
2. SYNCH BYTE (0x55)
3. IDENTIFIER FIELD

The first two components are present to allow the LIN slave nodes to recognize the beginning of the message frame and derive the bit rate of the master module.

The SYNCH BREAK allows the slave to see the beginning of a message frame on the bus. The SLIC module can receive a standard 10-bit break character for the SYNCH BREAK, or any break symbol 10 or more bit times in length. This encompasses the LIN requirement of 13 or more bits of length for the SYNCH BREAK character.

The SYNCH BYTE is always a 0x55 data byte, providing five falling edges for the slave to derive the bit rate of the master node.

The identifier byte indicates to the slave what is the nature of the data in the message frame. This data might be supplied from either the master node or the slave node, as determined at system design time. The slave node must read this identifier, check for parity errors, and determine whether it is to send or receive data in the data field.

More information on the HEADER is contained in Section 12.6.7.1, “LIN Message Headers.”

12.6.2 LIN Data Field

The data field is comprised of standard bytes (eight data bits, no parity, one stop bit) of data, from 0–8 bytes for normal LIN frames and greater than eight bytes for extended LIN frames. The SLIC module will either transmit or receive these bytes, depending upon the user code interpretation of the identifier byte. Data is always transmitted into the data field least significant byte (LSB) first.

The SLIC module can automatically handle up to 64 bytes in extended LIN message frames without significantly changing program execution.

Table 13-1. SPIC1 Field Descriptions (continued)

Field	Description
4 MSTR	**Master/Slave Mode Select** 0 SPI module configured as a slave SPI device 1 SPI module configured as a master SPI device
3 CPOL	**Clock Polarity** — This bit effectively places an inverter in series with the clock signal from a master SPI or to a slave SPI device. Refer to Section 13.5.1, "SPI Clock Formats" for more details. 0 Active-high SPI clock (idles low) 1 Active-low SPI clock (idles high)
2 CPHA	**Clock Phase** — This bit selects one of two clock formats for different kinds of synchronous serial peripheral devices. Refer to Section 13.5.1, "SPI Clock Formats" for more details. 0 First edge on SPSCK occurs at the middle of the first cycle of an 8-cycle data transfer 1 First edge on SPSCK occurs at the start of the first cycle of an 8-cycle data transfer
1 SSOE	**Slave Select Output Enable** — This bit is used in combination with the mode fault enable (MODFEN) bit in SPCR2 and the master/slave (MSTR) control bit to determine the function of the $\overline{SS}$ pin as shown in Table 13-2.
0 LSBFE	**LSB First (Shifter Direction)** 0 SPI serial data transfers start with most significant bit 1 SPI serial data transfers start with least significant bit

Table 13-2. $\overline{SS}$ Pin Function

MODFEN	SSOE	Master Mode	Slave Mode
0	0	General-purpose I/O (not SPI)	Slave select input
0	1	General-purpose I/O (not SPI)	Slave select input
1	0	$\overline{SS}$ input for mode fault	Slave select input
1	1	Automatic $\overline{SS}$ output	Slave select input

NOTE

Ensure that the SPI should not be disabled (SPE=0) at the same time as a bit change to the CPHA bit. These changes should be performed as separate operations or unexpected behavior may occur.

13.4.2 SPI Control Register 2 (SPIC2)

This read/write register is used to control optional features of the SPI system. Bits 7, 6, 5, and 2 are not implemented and always read 0.

	7	6	5	4	3	2	1	0
R	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 13-6. SPI Control Register 2 (SPIC2)

(becomes unlatched) when the TPMxCnSC register is written (whether BDM mode is active or not). Any write to the channel registers will be ignored during the input capture mode.

When BDM is active, the coherency mechanism is frozen (unless reset by writing to TPMxCnSC register) such that the buffer latches remain in the state they were in when the BDM became active, even if one or both halves of the channel register are read while BDM is active. This assures that if the user was in the middle of reading a 16-bit register when BDM became active, it will read the appropriate value from the other half of the 16-bit value after returning to normal execution. The value read from the TPMxCnVH and TPMxCnVL registers in BDM mode is the value of these registers and not the value of their read buffer.

In output compare or PWM modes, writing to either byte (TPMxCnVH or TPMxCnVL) latches the value into a buffer. After both bytes are written, they are transferred as a coherent 16-bit value into the timer-channel registers according to the value of CLKSB:CLKSA bits and the selected mode, so:

- If (CLKSB:CLKSA = 0:0), then the registers are updated when the second byte is written.
- If (CLKSB:CLKSA not = 0:0 and in output compare mode) then the registers are updated after the second byte is written and on the next change of the TPM counter (end of the prescaler counting).
- If (CLKSB:CLKSA not = 0:0 and in EPWM or CPWM modes), then the registers are updated after the both bytes were written, and the TPM counter changes from (TPMxMODH:TPMxMODL - 1) to (TPMxMODH:TPMxMODL). If the TPM counter is a free-running counter then the update is made when the TPM counter changes from 0xFFFE to 0xFFFF.

The latching mechanism may be manually reset by writing to the TPMxCnSC register (whether BDM mode is active or not). This latching mechanism allows coherent 16-bit writes in either big-endian or little-endian order which is friendly to various compiler implementations.

When BDM is active, the coherency mechanism is frozen such that the buffer latches remain in the state they were in when the BDM became active even if one or both halves of the channel register are written while BDM is active. Any write to the channel registers bypasses the buffer latches and directly write to the channel register while BDM is active. The values written to the channel register while BDM is active 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.)

- Non-intrusive commands can be executed at any time even while the user's program is running. Non-intrusive commands allow a user to read or write MCU memory locations or access status and control registers within the background debug controller.

Typically, a relatively simple interface pod is used to translate commands from a host computer into commands for the custom serial interface to the single-wire background debug system. Depending on the development tool vendor, this interface pod may use a standard RS-232 serial port, a parallel printer port, or some other type of communications such as a universal serial bus (USB) to communicate between the host PC and the pod. The pod typically connects to the target system with ground, the BKGD pin, $\overline{\text{RESET}}$, and sometimes V_{DD}. An open-drain connection to reset allows the host to force a target system reset, which is useful to regain control of a lost target system or to control startup of a target system before the on-chip nonvolatile memory has been programmed. Sometimes V_{DD} can be used to allow the pod to use power from the target system to avoid the need for a separate power supply. However, if the pod is powered separately, it can be connected to a running target system without forcing a target system reset or otherwise disturbing the running application program.

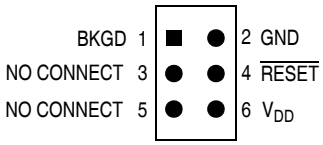

Figure 17-2. BDM Tool Connector

17.2.1 BKGD Pin Description

BKGD is the single-wire background debug interface pin. The primary function of this pin is for bidirectional serial communication of active background mode commands and data. During reset, this pin is used to select between starting in active background mode or starting the user's application program. This pin is also used to request a timed sync response pulse to allow a host development tool to determine the correct clock frequency for background debug serial communications.

BDC serial communications use a custom serial protocol first introduced on the M68HC12 Family of microcontrollers. This protocol assumes the host knows the communication clock rate that is determined by the target BDC clock rate. All communication is initiated and controlled by the host that drives a high-to-low edge to signal the beginning of each bit time. Commands and data are sent most significant bit first (MSB first). For a detailed description of the communications protocol, refer to Section 17.2.2, "Communication Details."

If a host is attempting to communicate with a target MCU that has an unknown BDC clock rate, a SYNC command may be sent to the target MCU to request a timed sync response signal from which the host can determine the correct communication speed.

BKGD is a pseudo-open-drain pin and there is an on-chip pullup so no external pullup resistor is required. Unlike typical open-drain pins, the external RC time constant on this pin, which is influenced by external capacitance, plays almost no role in signal rise time. The custom protocol provides for brief, actively driven speedup pulses to force rapid rise times on this pin without risking harmful drive level conflicts. Refer to Section 17.2.2, "Communication Details," for more detail.

The SYNC command is unlike other BDC commands because the host does not necessarily know the correct communications speed to use for BDC communications until after it has analyzed the response to the SYNC command.

To issue a SYNC command, the host:

- Drives the BKGD pin low for at least 128 cycles of the slowest possible BDC clock (The slowest clock is normally the reference oscillator/64 or the self-clocked rate/64.)
- Drives BKGD high for a brief speedup pulse to get a fast rise time (This speedup pulse is typically one cycle of the fastest clock in the system.)
- Removes all drive to the BKGD pin so it reverts to high impedance
- Monitors the BKGD pin for the sync response pulse

The target, upon detecting the SYNC request from the host (which is a much longer low time than would ever occur during normal BDC communications):

- Waits for BKGD to return to a logic high
- Delays 16 cycles to allow the host to stop driving the high speedup pulse
- Drives BKGD low for 128 BDC clock cycles
- Drives a 1-cycle high speedup pulse to force a fast rise time on BKGD
- Removes all drive to the BKGD pin so it reverts to high impedance

The host measures the low time of this 128-cycle sync response pulse and determines the correct speed for subsequent BDC communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

17.2.4 BDC Hardware Breakpoint

The BDC includes one relatively simple hardware breakpoint that compares the CPU address bus to a 16-bit match value in the BDCBKPT register. This breakpoint can generate a forced breakpoint or a tagged breakpoint. A forced breakpoint causes the CPU to enter active background mode at the first instruction boundary following any access to the breakpoint address. The tagged breakpoint causes the instruction opcode at the breakpoint address to be tagged so that the CPU will enter active background mode rather than executing that instruction if and when it reaches the end of the instruction queue. This implies that tagged breakpoints can only be placed at the address of an instruction opcode while forced breakpoints can be set at any address.

The breakpoint enable (BKPTEN) control bit in the BDC status and control register (BDCSCR) is used to enable the breakpoint logic (BKPTEN = 1). When BKPTEN = 0, its default value after reset, the breakpoint logic is disabled and no BDC breakpoints are requested regardless of the values in other BDC breakpoint registers and control bits. The force/tag select (FTS) control bit in BDCSCR is used to select forced (FTS = 1) or tagged (FTS = 0) type breakpoints.

The on-chip debug module (DBG) includes circuitry for two additional hardware breakpoints that are more flexible than the simple breakpoint in the BDC module.

Table A-10. Analog Comparator Electrical Specifications (continued)

Num	C	Rating	Symbol	Min	Typical	Max	Unit
4	D	Analog input offset voltage	V_{AIO}		20	40	mV
5	D	Analog Comparator hysteresis	V_H	3.0	6.0	20.0	mV
6	D	Analog input leakage current	I_{ALKG}	—	—	1.0	μA
7	D	Analog Comparator initialization delay	t_{AINIT}	—	—	1.0	μs

A.11 ADC Characteristics

Table A-11. ADC Operating Conditions

Num	Characteristic	Conditions	Symb	Min	Typ[1]	Max	Unit	Comment
1	Supply voltage	Absolute	V_{DDAD}	2.7	—	5.5	V	
2	Input Voltage		V_{ADIN}	V_{REFL}	—	V_{REFH}	V	
3	Input Capacitance		C_{ADIN}	—	4.5	5.5	pF	
4	Input Resistance		R_{ADIN}	—	3	5	kΩ	
5	Analog Source Resistance	10 bit mode f_{ADCK} > 4MHz f_{ADCK} < 4MHz	R_{AS}	 — —	 — —	 5 10	kΩ	External to MCU
6		8 bit mode (all valid f_{ADCK})		—	—	10		
7	ADC Conversion Clock Freq.	High Speed (ADLPC=0)	f_{ADCK}	0.4	—	8.0	MHz	
8		Low Power (ADLPC=1)		0.4	—	4.0		

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

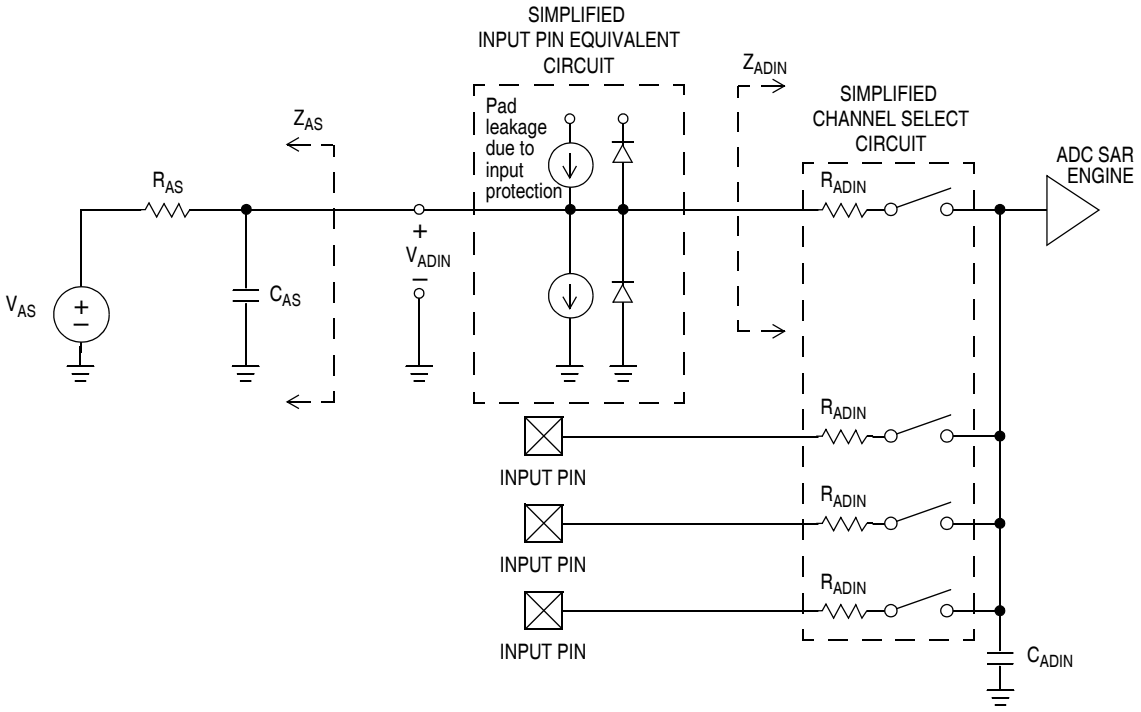

Figure A-9. ADC Input Impedance Equivalency Diagram

Table A-12. ADC Characteristics

Characteristic	Conditions	C	Symb	Min	Typ[1]	Max	Unit	Comment
Supply current	ADLPC=1 ADLSMP=1 ADCO=1	T	I_{DD} + I_{DDAD}	—	133	—	μA	ADC current only
	ADLPC=1 ADLSMP=0 ADCO=1	T	I_{DD} + I_{DDAD}	—	218	—	μA	ADC current only
	ADLPC=0 ADLSMP=1 ADCO=1	T	I_{DD} + I_{DDAD}	—	327	—	μA	ADC current only
	ADLPC=0 ADLSMP=0 ADCO=1	P	I_{DD} + I_{DDAD}	—	0.582	1	mA	ADC current only
ADC asynchronous clock source	High speed (ADLPC=0)	P	f_{ADACK}	2	3.3	5	MHz	t_{ADACK} = $1/f_{ADACK}$
	Low power (ADLPC=1)			1.25	2	3.3		