



Welcome to [E-XFL.COM](#)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I ² C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	22
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	28-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08el32f1mtl

Section Number	Title	Page
8.4.1	Operational Modes	123
8.4.2	Mode Switching	125
8.4.3	Bus Frequency Divider	126
8.4.4	Low Power Bit Usage	126
8.4.5	Internal Reference Clock	126
8.4.6	Optional External Reference Clock	126
8.4.7	Fixed Frequency Clock	127

Chapter 9

5-V Analog Comparator (S08ACMPV2)

9.1	Introduction	129
9.1.1	ACMPx Configuration Information	129
9.1.2	ACMP1/TPM1 Configuration Information	129
9.1.3	Features	131
9.1.4	Modes of Operation	131
9.1.5	Block Diagram	132
9.2	External Signal Description	133
9.3	Memory Map	133
9.3.1	Register Descriptions	133
9.4	Functional Description	135

Chapter 10

Analog-to-Digital Converter (S08ADC1)

10.1	Introduction	137
10.1.1	Channel Assignments	137
10.1.2	Alternate Clock	138
10.1.3	Hardware Trigger	138
10.1.4	Temperature Sensor	138
10.1.5	Features	141
10.1.6	Block Diagram	141
10.2	External Signal Description	142
10.2.1	Analog Power (V_{DDAD})	143
10.2.2	Analog Ground (V_{SSAD})	143
10.2.3	Voltage Reference High (V_{REFH})	143
10.2.4	Voltage Reference Low (V_{REFL})	143
10.2.5	Analog Channel Inputs (ADx)	143
10.3	Register Definition	143
10.3.1	Status and Control Register 1 (ADCSC1)	143
10.3.2	Status and Control Register 2 (ADCSC2)	145
10.3.3	Data Result High Register (ADCRH)	146
10.3.4	Data Result Low Register (ADCRL)	146
10.3.5	Compare Value High Register (ADCCVH)	147

Section Number	Title	Page
11.6	Interrupts	179
11.6.1	Byte Transfer Interrupt	179
11.6.2	Address Detect Interrupt	180
11.6.3	Arbitration Lost Interrupt	180
11.7	Initialization/Application Information	181

Chapter 12

Slave LIN Interface Controller (S08SLICV1)

12.1	Introduction	185
12.1.1	Features	187
12.1.2	Modes of Operation	188
12.1.3	Block Diagram	191
12.2	External Signal Description	191
12.2.1	SLCTx — SLIC Transmit Pin	191
12.2.2	SLCRx — SLIC Receive Pin	191
12.3	Register Definition	191
12.3.1	SLIC Control Register 1 (SLCC1)	191
12.3.2	SLIC Control Register 2 (SLCC2)	193
12.3.3	SLIC Bit Time Registers (SLCBTH, SLCBTL)	195
12.3.4	SLIC Status Register (SLCS)	196
12.3.5	SLIC State Vector Register (SLCSV)	197
12.3.6	SLIC Data Length Code Register (SLCDLC)	202
12.3.7	SLIC Identifier and Data Registers (SLCID, SLCD7-SLCD0)	203
12.4	Functional Description	204
12.5	Interrupts	204
12.5.1	SLIC During Break Interrupts	204
12.6	Initialization/Application Information	204
12.6.1	LIN Message Frame Header	205
12.6.2	LIN Data Field	205
12.6.3	LIN Checksum Field	206
12.6.4	SLIC Module Constraints	206
12.6.5	SLCSV Interrupt Handling	206
12.6.6	SLIC Module Initialization Procedure	206
12.6.7	Handling LIN Message Headers	208
12.6.8	Handling Command Message Frames	211
12.6.9	Handling Request LIN Message Frames	214
12.6.10	Handling MSG to Minimize Interrupts	218
12.6.11	Sleep and Wakeup Operation	219
12.6.12	Polling Operation	219
12.6.13	LIN Data Integrity Checking Methods	219
12.6.14	High-Speed LIN Operation	220
12.6.15	Bit Error Detection and Physical Layer Delay	223

Section Number	Title	Page
16.6	Interrupts	300
16.6.1	General	300
16.6.2	Description of Interrupt Operation	301
16.7	The Differences from TPM v2 to TPM v3	302

Chapter 17 Development Support

17.1	Introduction	307
17.1.1	Forcing Active Background	307
17.1.2	Features	310
17.2	Background Debug Controller (BDC)	310
17.2.1	BKGD Pin Description	311
17.2.2	Communication Details	312
17.2.3	BDC Commands	316
17.2.4	BDC Hardware Breakpoint	318
17.3	On-Chip Debug System (DBG)	319
17.3.1	Comparators A and B	319
17.3.2	Bus Capture Information and FIFO Operation	319
17.3.3	Change-of-Flow Information	320
17.3.4	Tag vs. Force Breakpoints and Triggers	320
17.3.5	Trigger Modes	321
17.3.6	Hardware Breakpoints	323
17.4	Register Definition	323
17.4.1	BDC Registers and Control Bits	323
17.4.2	System Background Debug Force Reset Register (SBDFFR)	325
17.4.3	DBG Registers and Control Bits	326

Appendix A Electrical Characteristics

A.1	Introduction	331
A.2	Parameter Classification	331
A.3	Absolute Maximum Ratings	331
A.4	Thermal Characteristics	332
A.5	ESD Protection and Latch-Up Immunity	333
A.6	DC Characteristics	334
A.7	Supply Current Characteristics	338
A.8	External Oscillator (XOSC) Characteristics	341
A.9	Internal Clock Source (ICS) Characteristics	342
A.10	Analog Comparator (ACMP) Electricals	343
A.11	ADC Characteristics	344
A.12	AC Characteristics	347
A.12.1	Control Timing	347

5.4 Computer Operating Properly (COP) Watchdog

The COP watchdog is intended to force a system reset when the application software fails to execute as expected. To prevent a system reset from the COP timer (when it is enabled), application software must reset the COP counter periodically. If the application program gets lost and fails to reset the COP counter before it times out, a system reset is generated to force the system back to a known starting point.

After any reset, the COP watchdog is enabled (see [Section 5.7.3, “System Options Register 1 \(SOPT1\),”](#) for additional information). If the COP watchdog is not used in an application, it can be disabled by clearing COPT bits in SOPT1.

The COP counter is reset by writing 0x0055 and 0x00AA (in this order) to the address of SRS during the selected timeout period. Writes do not affect the data in the read-only SRS. As soon as the write sequence is done, the COP timeout period is restarted. If the program fails to do this during the time-out period, the MCU will reset. Also, if any value other than 0x0055 or 0x00AA is written to SRS, the MCU is immediately reset.

The COPCLKS bit in SOPT2 (see [Section 5.7.4, “System Options Register 2 \(SOPT2\),”](#) for additional information) selects the clock source used for the COP timer. The clock source options are either the bus clock or an internal 1-kHz clock source. With each clock source, there are three associated time-outs controlled by the COPT bits in SOPT1. [Table 5-1](#) summarizes the control functions of the COPCLKS and COPT bits. The COP watchdog defaults to operation from the 1-kHz clock source and the longest time-out (2^{10} cycles).

Table 5-1. COP Configuration Options

Control Bits		Clock Source	COP Overflow Count
COPCLKS	COPT[1:0]		
N/A	0:0	N/A	COP is disabled
0	0:1	1 kHz	2^5 cycles (32 ms ¹)
0	1:0	1 kHz	2^8 cycles (256 ms ¹)
0	1:1	1 kHz	2^{10} cycles (1.024 s ¹)
1	0:1	Bus	2^{13} cycles
1	1:0	Bus	2^{16} cycles
1	1:1	Bus	2^{18} cycles

¹ Values are shown in this column based on $t_{RTI} = 1$ ms. See t_{RTI} in the appendix [Section A.12.1, “Control Timing,”](#) for the tolerance of this value.

When the bus clock source is selected, windowed COP operation is available by setting COPW in the SOPT2 register. In this mode, writes to the SRS register to clear the COP timer must occur in the last 25% of the selected timeout period. A premature write immediately resets the MCU. When the 1-kHz clock source is selected, windowed COP operation is not available.

5.7.3 System Options Register 1 (SOPT1)

This high page register is a write-once register so only the first write after reset is honored. It can be read at any time. Any subsequent attempt to write to SOPT1 (intentionally or unintentionally) is ignored to avoid accidental changes to these sensitive settings. SOPT1 should be written during the user's reset initialization program to set the desired controls even if the desired settings are the same as the reset settings.

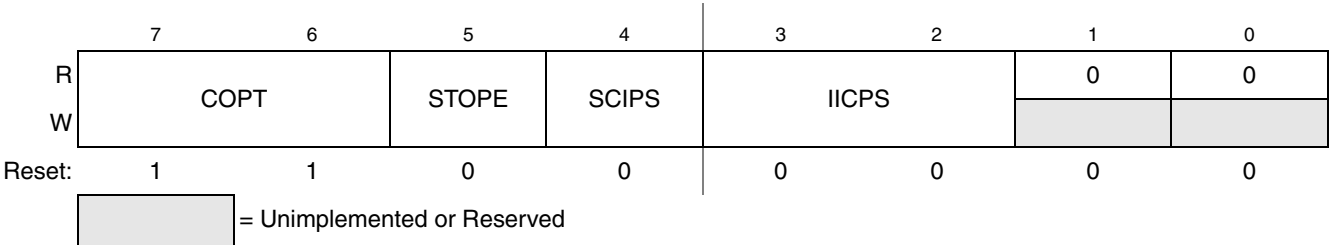


Figure 5-4. System Options Register 1 (SOPT1)

Table 5-5. SOPT1 Register Field Descriptions

Field	Description
7:6 COPT[1:0]	COP Watchdog Timeout — These write-once bits select the timeout period of the COP. COPT along with COPCLKS in SOPT2 defines the COP timeout period. See Table 5-1 .
5 STOPE	Stop Mode Enable — This write-once bit is used to enable stop mode. If stop mode is disabled and a user program attempts to execute a STOP instruction, an illegal opcode reset is forced. 0 Stop mode disabled. 1 Stop mode enabled.
4 SCIPS	SCI Pin Select — This write-once bit selects the location of the RxD and TxD pins of the SCI module. 0 RxD on PTB0, TxD on PTB1. 1 RxD on PTA2, TxD on PTA3.
3:2 IICPS	IIC Pin Select — These write-once bits select the location of the SCL and SDA pins of the IIC module. 00 SDA on PTA2, SCL on PTA3. 01 SDA on PTB6, SCL on PTB7. 1x SDA on PTB2, SCL on PTB3.

It is a good programming practice to write to the port data register before changing the direction of a port pin to become an output. This ensures that the pin will not be driven momentarily with an old data value that happened to be in the port data register.

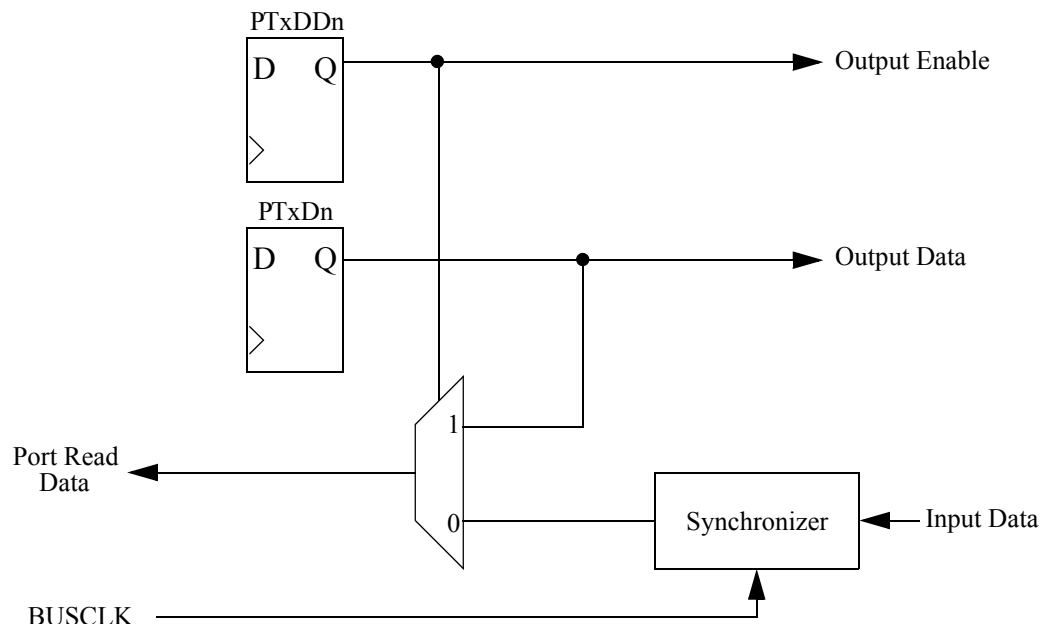


Figure 6-1. Parallel I/O Block Diagram

6.2 Pull-up, Slew Rate, and Drive Strength

Associated with the parallel I/O ports is a set of registers located in the high page register space that operate independently of the parallel I/O registers. These registers are used to control pull-ups, slew rate, and drive strength for the pins.

An internal pull-up device can be enabled for each port pin by setting the corresponding bit in the pull-up enable register (PTxPEN). The pull-up 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 pull-up enable register bit. The pull-up device is also disabled if the pin is controlled by an analog function.

Slew rate control can be enabled for each port pin by setting the corresponding bit in the slew rate control register (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.

An output pin can be selected to have high output drive strength by setting the corresponding bit in the drive strength select register (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 MCU 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.

6.5.1.5 Port A Drive Strength Selection Register (PTADS)

	7	6	5	4	3	2	1	0
R	PTADS7	PTADS6	0	0	PTADS3	PTADS2	PTADS1	PTADS0
W								
Reset:	0	0	0	0	0	0	0	0

Figure 6-7. Drive Strength Selection for Port A Register (PTADS)

Table 6-5. PTADS Register Field Descriptions

Field	Description
7:6 PTADS[7:6]	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. 0 Low output drive strength selected for port A bit n. 1 High output drive strength selected for port A bit n.
3:0 PTADS[3:0]	

6.5.1.6 Port A Interrupt Status and Control Register (PTASC)

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTAIF	0	PTAIE	PTAMOD
W						PTAACK		
Reset:	0	0	0	0	0	0	0	0

Figure 6-8. Port A Interrupt Status and Control Register (PTASC)

Table 6-6. PTASC Register Field Descriptions

Field	Description
3 PTAIF	Port A Interrupt Flag — PTAIF indicates when a port A interrupt is detected. Writes have no effect on PTAIF. 0 No port A interrupt detected. 1 Port A interrupt detected.
2 PTAACK	Port A Interrupt Acknowledge — Writing a 1 to PTAACK is part of the flag clearing mechanism. PTAACK always reads as 0.
1 PTAIE	Port A Interrupt Enable — PTAIE determines whether a port A interrupt is requested. 0 Port A interrupt request not enabled. 1 Port A interrupt request enabled.
0 PTAMOD	Port A Detection Mode — PTAMOD (along with the PTAES bits) controls the detection mode of the port A interrupt pins. 0 Port A pins detect edges only. 1 Port A pins detect both edges and levels.

6.5.2.3 Port B Pull Enable Register (PTBPE)

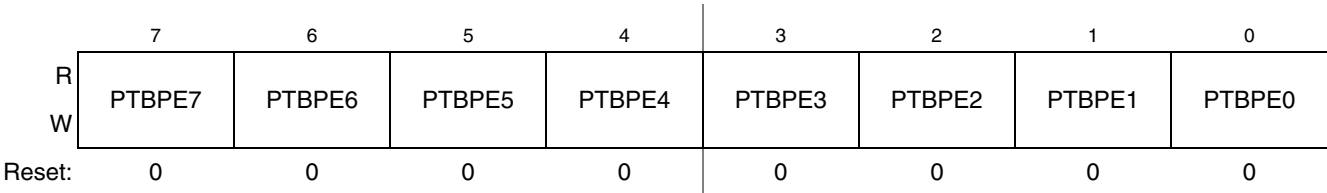


Figure 6-13. Internal Pull Enable for Port B Register (PTBPE)

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)

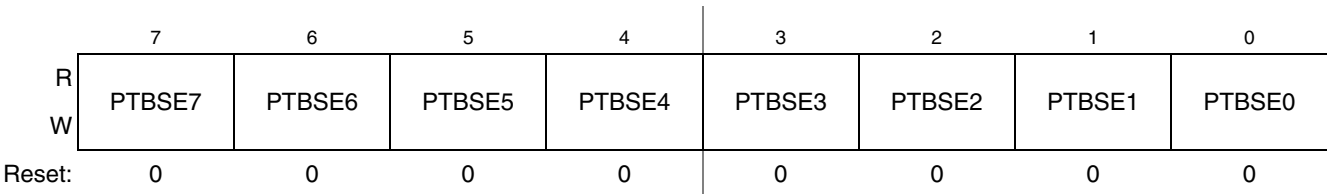


Figure 6-14. Slew Rate Enable for Port B Register (PTBSE)

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.

8.3.3 ICS Trim Register (ICSTRM)

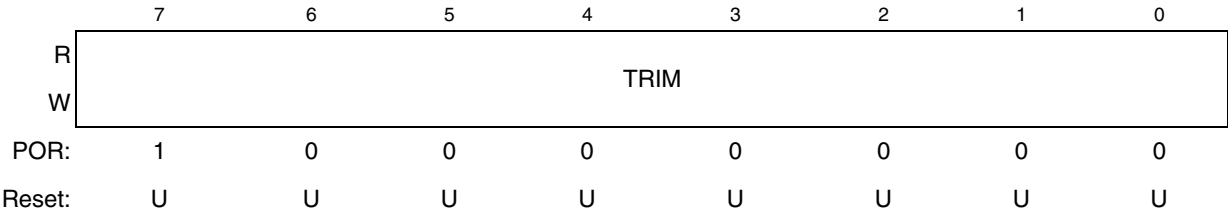


Figure 8-5. ICS Trim Register (ICSTRM)

Table 8-4. ICS Trim Register Field Descriptions

Field	Description
7:0 TRIM	<p>ICS Trim Setting — The TRIM bits control the internal reference clock frequency by controlling the internal reference clock period. The bits' effect are binary weighted (i.e., bit 1 will adjust twice as much as bit 0). Increasing the binary value in TRIM will increase the period, and decreasing the value will decrease the period.</p> <p>An additional fine trim bit is available in ICSSC as the FTRIM bit.</p>

8.3.4 ICS Status and Control (ICSSC)

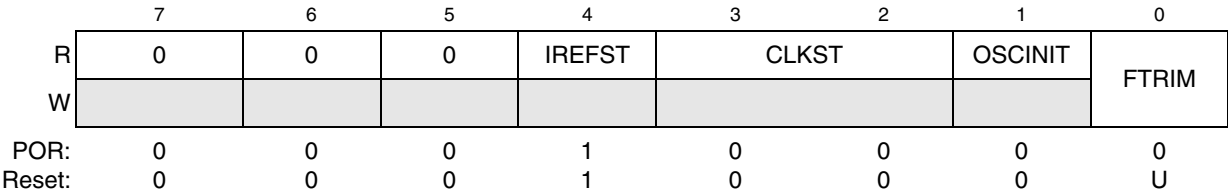


Figure 8-6. ICS Status and Control Register (ICSSC)

Table 8-5. ICS Status and Control Register Field Descriptions

Field	Description
7:5	Reserved, should be cleared.
4 IREFST	<p>Internal Reference Status — The IREFST bit indicates the current source for the reference clock. The IREFST bit does not update immediately after a write to the IREFS bit due to internal synchronization between clock domains.</p> <p>0 Source of reference clock is external clock. 1 Source of reference clock is internal clock.</p>
3-2 CLKST	<p>Clock Mode Status — The CLKST bits indicate the current clock mode. The CLKST bits don't update immediately after a write to the CLKST bits due to internal synchronization between clock domains.</p> <p>00 Output of FLL is selected. 01 FLL Bypassed, Internal reference clock is selected. 10 FLL Bypassed, External reference clock is selected. 11 Reserved.</p>

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.

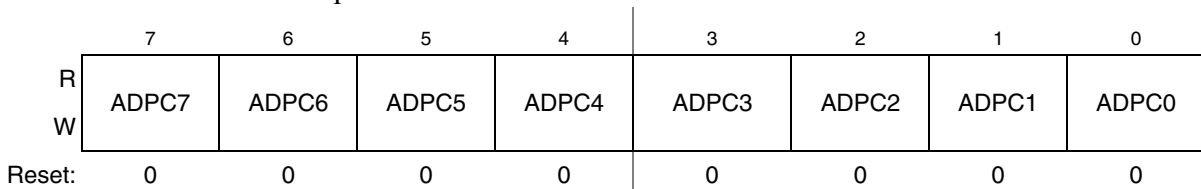


Figure 10-11. Pin Control 1 Register (APCTL1)

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 11-8. IICD Field Descriptions

Field	Description
7–0 DATA	Data — In master transmit mode, when data is written to the IICD, a data transfer is initiated. The most significant bit is sent first. In master receive mode, reading this register initiates receiving of the next byte of data.

NOTE

When transitioning out of master receive mode, the IIC mode should be switched before reading the IICD register to prevent an inadvertent initiation of a master receive data transfer.

In slave mode, the same functions are available after an address match has occurred.

The TX bit in IICC must correctly reflect the desired direction of transfer in master and slave modes for the transmission to begin. For instance, if the IIC is configured for master transmit but a master receive is desired, reading the IICD does not initiate the receive.

Reading the IICD returns the last byte received while the IIC is configured in master receive or slave receive modes. The IICD does not reflect every byte transmitted on the IIC bus, nor can software verify that a byte has been written to the IICD correctly by reading it back.

In master transmit mode, the first byte of data written to IICD following assertion of MST is used for the address transfer and should comprise of the calling address (in bit 7 to bit 1) concatenated with the required R/W bit (in position bit 0).

11.3.6 IIC Control Register 2 (IICC2)

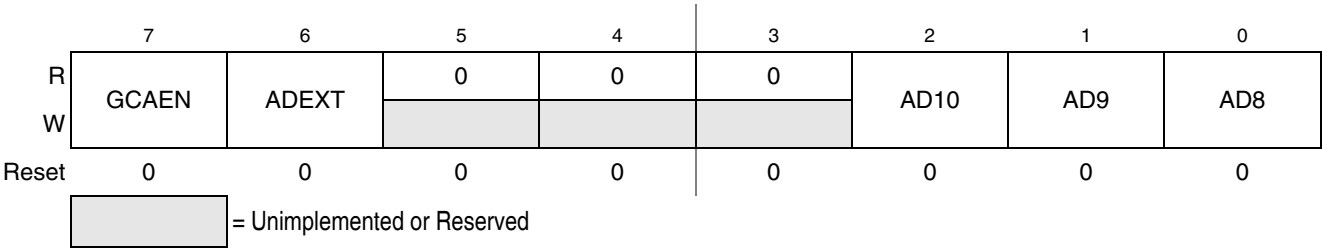


Figure 11-8. IIC Control Register (IICC2)

Table 11-9. IICC2 Field Descriptions

Field	Description
7 GCAEN	General Call Address Enable. The GCAEN bit enables or disables general call address. 0 General call address is disabled 1 General call address is enabled
6 ADEXT	Address Extension. The ADEXT bit controls the number of bits used for the slave address. 0 7-bit address scheme 1 10-bit address scheme
2–0 AD[10:8]	Slave Address. The AD[10:8] field contains the upper three bits of the slave address in the 10-bit address scheme. This field is only valid when the ADEXT bit is set.

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

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

Table 11-11. Master-Receiver Addresses a Slave-Transmitter with a 10-bit Address

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

11.4.3 General Call Address

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

11.5 Resets

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

11.6 Interrupts

The IIC generates a single interrupt.

An interrupt from the IIC is generated when any of the events in [Table 11-12](#) occur, provided the IICIE bit is set. The interrupt is driven by bit IICIF (of the IIC status register) and masked with bit IICIE (of the IIC control register). The IICIF bit must be cleared by software by writing a 1 to it in the interrupt routine. You can determine the interrupt type by reading the status register.

Table 11-12. Interrupt Summary

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

11.6.1 Byte Transfer Interrupt

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

entering SLIC stop mode, any activity on the network will cause the SLIC module to exit SLIC stop mode and generate an unmaskable interrupt of the CPU. This wakeup interrupt state is reflected in the SLCSV, encoded as the highest priority interrupt. This interrupt can be cleared by the CPU with a read of the SLCSV and clearing of the SLCF interrupt flag. Depending upon which low-power mode instruction the CPU executes to cause the SLIC module to enter SLIC stop, the message which wakes up the SLIC module (and the CPU) may or may not be received.

There are two different possibilities:

1. **Wakeup from SLIC Stop with CPU in STOP**
When the CPU executes the STOP instruction, all clocks in the MCU, including clocks to the SLIC module, are turned off. Therefore, the message which wakes up the SLIC module and the CPU from stop mode will not be received. This is due primarily to the amount of time required for the MCU's oscillator to stabilize before the clocks can be applied internally to the other MCU modules, including the SLIC module.
2. **Wakeup from SLIC Stop with CPU in WAIT.** If the CPU executes the WAIT instruction and the SLIC module enters the stop mode (SLCWCM = 1), the clocks to the SLIC module are turned off, but the clocks in the MCU continue to run. Therefore, the message which wakes up the SLIC module from stop and the CPU from wait mode will be received correctly by the SLIC module. This is because very little time is required for the CPU to turn the clocks to the SLIC module back on after the wakeup interrupt occurs.

NOTE

While the SLIC module will correctly receive a message which arrives when the SLIC module is in stop or wait mode and the MCU is in wait mode, if the user enters this mode while a message is being received, the data in the message will become corrupted. This is due to the steps required for the SLIC module to resume operation upon exiting stop or wait mode, and its subsequent resynchronization with the LIN bus.

12.1.2.8 Normal and Emulation Mode Operation

The SLIC module operates in the same manner in all normal and emulation modes. All SLIC module registers can be read and written except those that are reserved, unimplemented, or write once. The user must be careful not to unintentionally change reserved bits to avoid unexpected SLIC module behavior.

12.1.2.9 Special Mode Operation

Some aspects of SLIC module operation can be modified in special test mode. This mode is reserved for internal use only.

12.1.2.10 Low-Power Options

The SLIC module can save power in disabled, wait, and stop modes.

method was employed for this message frame. Refer to the LIN specification for more details on the calculations.

- **Byte Framing Error**

This error comes from the standard UART definition for byte encoding and occurs when the STOP bit is sampled and reads back as a 090. STOP should always read as 1.

NOTE

A byte framing error can also be an indication that the number of data bytes received in a LIN message frame does not match the value written to the SLCDLC register. See [Section 12.6.7, “Handling LIN Message Headers,”](#) for more details.

- **Identifier Received Successfully**

This interrupt source indicates that a LIN identifier byte has been received with correct parity and is waiting in the LIN identifier buffer (SLCID). Upon reading this interrupt source from SLCSV, the user can then decode the identifier in software to determine the nature of the LIN message frame. To clear this source, SLCID must be read.

- **Identifier-Parity-Error**

A parity error in the identifier (i.e., corrupted identifier) will be flagged. Typical LIN slave applications do not distinguish between an unknown but valid identifier, and a corrupted identifier. However, it is mandatory for all slave nodes to evaluate in case of a known identifier all eight bits of the ID-Field and distinguish between a known and a corrupted identifier. The received identifier value is reported in SLCID so that the user software can choose to acknowledge or ignore the parity error message. Once the ID parity error has been detected, the SLIC will begin looking for another LIN header and will not receive message data, even if it appears on the bus.

- **Wakeup**

The wakeup interrupt source indicates that the SLIC module has entered SLIC run mode from SLIC stop mode.

12.3.5.2 Byte Transfer Mode Operation

When byte transfer mode is enabled (BTM = 1), many of the interrupt sources for the SLCSV no longer apply, as they are specific to LIN operations. [Table 12-9](#) shows those interrupt sources which are applicable to BTM operations. The value of the SLCSV for each interrupt source remains the same, as well as the priority of the interrupt source.

Table 12-9. Interrupt Sources Summary (BTM = 1)

SLCSV	I3	I2	I1	I0	Interrupt Source	Priority
0x00	0	0	0	0	No Interrupts Pending	0 (Lowest)
0x0C	0	0	1	1	TX Message Buffer Empty	3
0x14	0	1	0	1	RX Data Buffer Full No Errors	5
0x18	0	1	1	0	Bit-Error	6
0x1C	0	1	1	1	Receiver Buffer Overrun	7

12.3.7 SLIC Identifier and Data Registers (SLCID, SLCD7-SLCD0)

The SLIC identifier (SLCID) and eight data registers (SLCD7–SLCD0) comprise the transmit and receive buffer and are used to read/write the identifier and message buffer 8 data bytes. In BTM mode (BTM = 1), only SLCID is used to send and receive bytes, as only one byte is handled at any one time. The number of bytes to be read from or written to these registers is determined by the user software and written to SLCDLC. To obtain proper data, reads and writes to these registers must be made based on the proper length corresponding to a particular message. It is the responsibility of the user software to keep track of this value to prevent data corruption. For example, it is possible to read data from locations in the message buffer which contain erroneous or old data if the user software reads more data registers than were updated by the incoming message, as indicated in SLCDLC.

NOTE

An incorrect length value written to SLCDLC can result in the user software misreading or miswriting data in the message buffer. An incorrect length value might also result in SLIC error messages. For example, if a 4-byte message is to be received, but the user software incorrectly reports a 3-byte length to the DLC, the SLIC will assume the 4th data byte is actually a checksum value and attempt to validate it as such. If this value doesn't match the calculated value, an incorrect checksum error will occur. If it does happen to match the expected value, then the message would be received as a 3-byte message with valid checksum. Either case is incorrect behavior for the application and can be avoided by ensuring that the correct length code is used for each identifier.

The first data byte received after the LIN identifier in a LIN message frame will be loaded into SLCD0. The next byte (if applicable) will be loaded into SLCD1, and so forth.

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

Figure 12-11. SLIC Identifier Register (SLCID)

The SLIC identifier register is used to capture the incoming LIN identifier and when the SLCSV value indicates that the identifier has been received successfully, this register contains the received identifier value. If the incoming identifier contained a parity error, this register value will not contain valid data.

In byte transfer mode (BTM = 1), this register is used for sending and receiving each byte of data. When transmitting bytes, the data is loaded into this register, then TXGO in SLCDLC is set to initiate the transmission. When receiving bytes, they are read from this register only.

14.1.1 Features

Features of SCI module include:

- Full-duplex, standard non-return-to-zero (NRZ) format
- Double-buffered transmitter and receiver with separate enables
- Programmable baud rates (13-bit modulo divider)
- Interrupt-driven or polled operation:
 - Transmit data register empty and transmission complete
 - Receive data register full
 - Receive overrun, parity error, framing error, and noise error
 - Idle receiver detect
 - Active edge on receive pin
 - Break detect supporting LIN
- Hardware parity generation and checking
- Programmable 8-bit or 9-bit character length
- Receiver wakeup by idle-line or address-mark
- Optional 13-bit break character generation / 11-bit break character detection
- Selectable transmitter output polarity

14.1.2 Modes of Operation

See [Section 14.3, “Functional Description,”](#) For details concerning SCI operation in these modes:

- 8- and 9-bit data modes
- Stop mode operation
- Loop mode
- Single-wire mode

EPWM mode
TPMxMODH:TPMxMODL = 0x0007
TPMxMODH:TPMxMODL = 0x0005

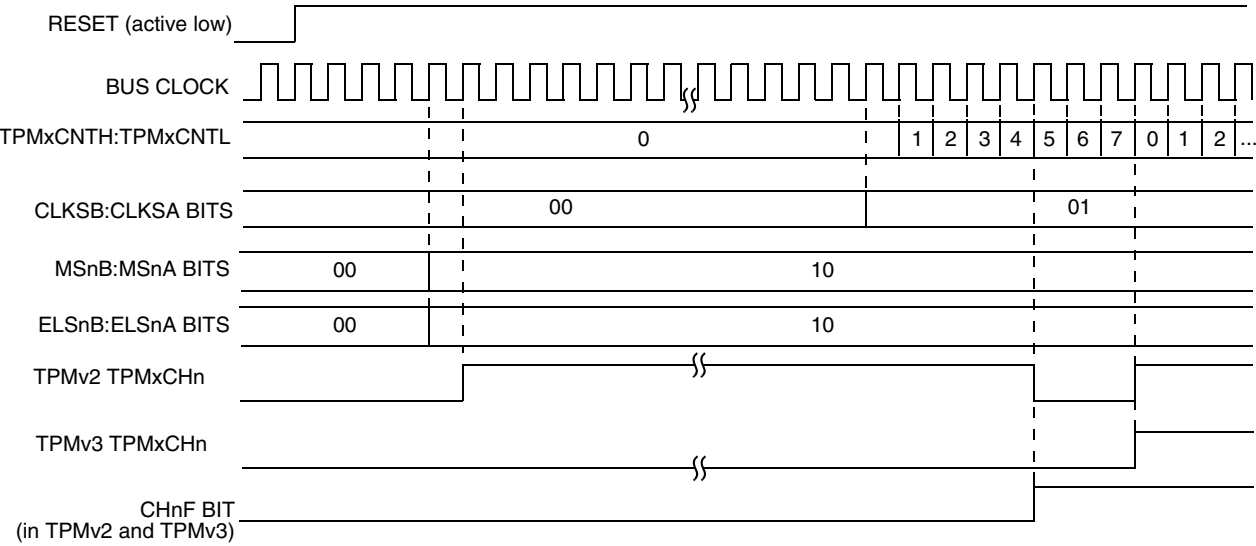
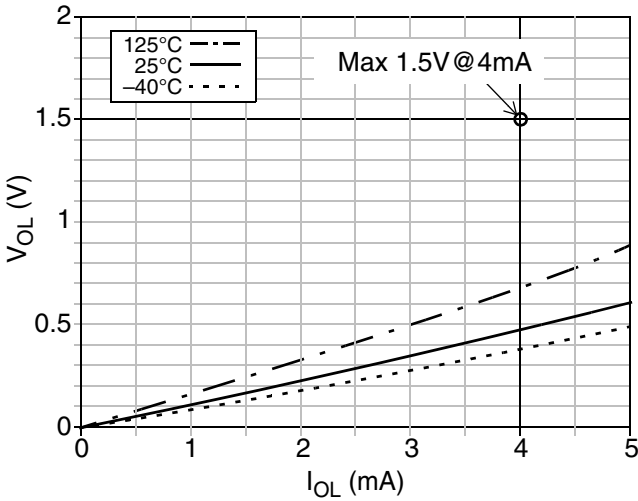
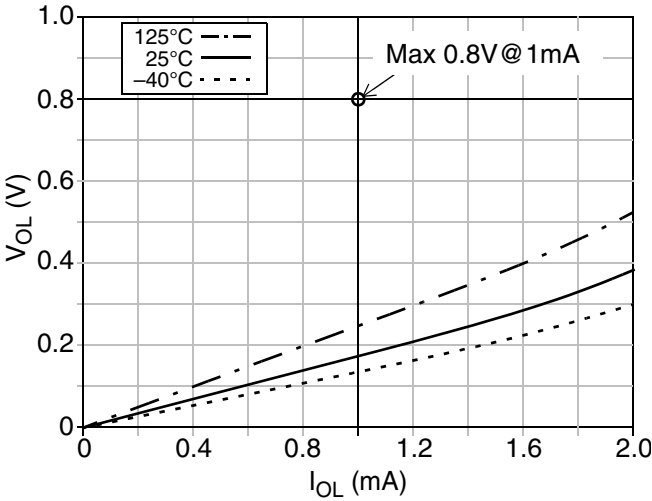


Figure 0-1. Generation of high-true EPWM signal by TPM v2 and v3 after the reset

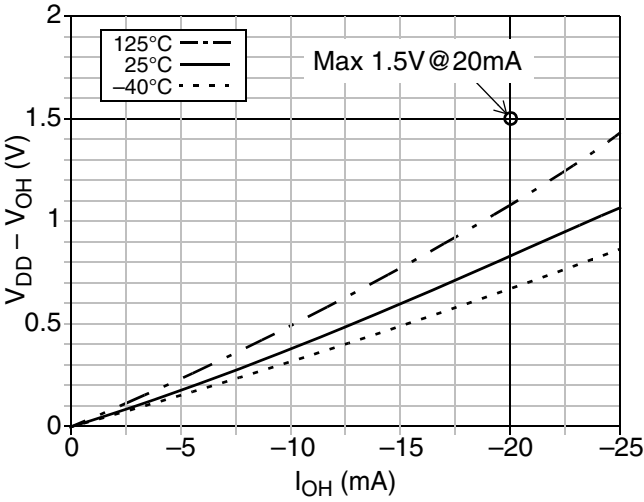


a) $V_{DD} = 5V$, Low Drive

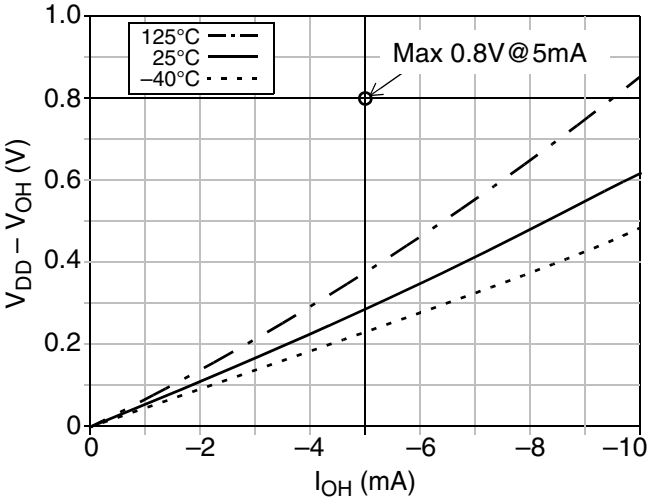


b) $V_{DD} = 3V$, Low Drive

Figure A-2. Typical V_{OL} vs I_{OL} , Low Drive Strength



a) $V_{DD} = 5V$, High Drive



b) $V_{DD} = 3V$, High Drive

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

Table A-15. SPI Electrical Characteristic (continued)

Num ¹	C	Rating ²	Symbol	Min	Max	Unit
11	D	Data hold time (outputs)				
		Master	t_{HO}	-10	—	ns
		Slave	t_{HO}	-10	—	ns
12	D	Operating frequency				
		Master	f_{op}	$f_{Bus}/2048$	5^5	MHz
		Slave	f_{op}	dc	$f_{Bus}/4$	

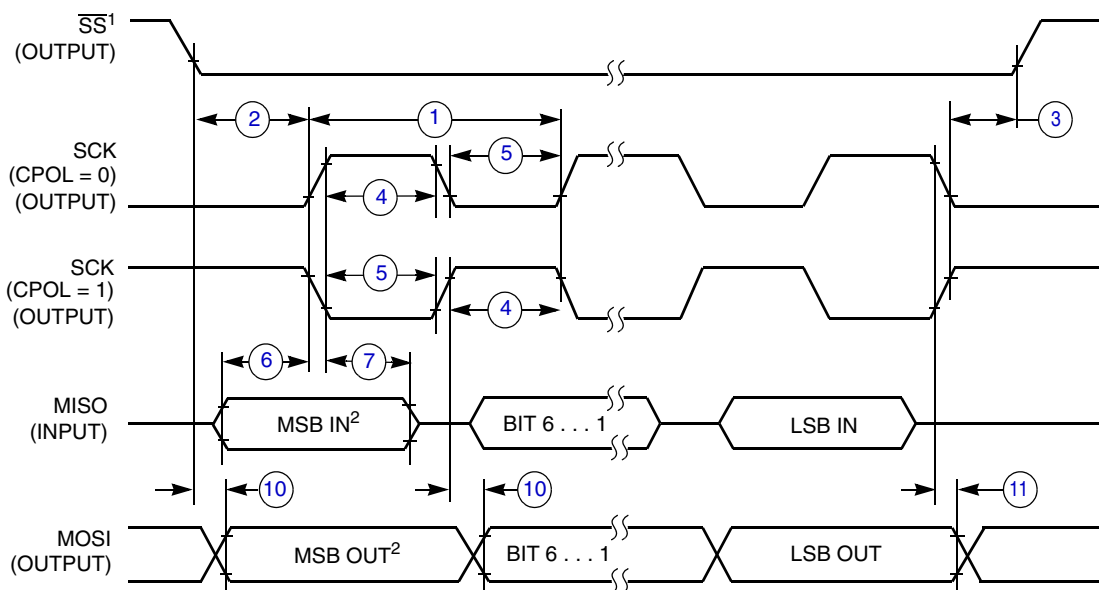
¹ Refer to Figure A-14 through Figure A-17.

² All timing is shown with respect to 20% V_{DD} and 70% V_{DD} , unless noted; 100 pF load on all SPI pins. All timing assumes slew rate control disabled and high drive strength enabled for SPI output pins.

³ Time to data active from high-impedance state.

⁴ Hold time to high-impedance state.

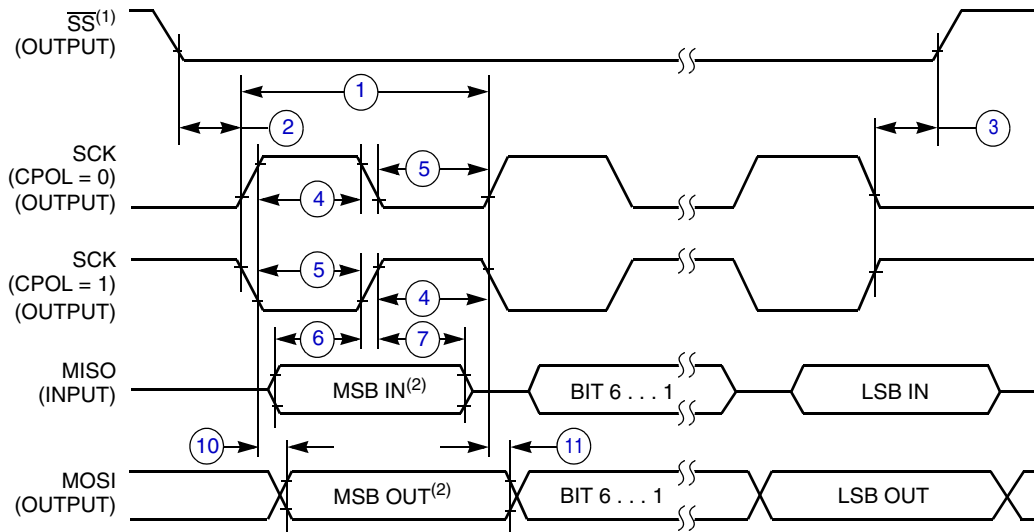
⁵ Maximum baud rate must be limited to 5 MHz due to input filter characteristics.



NOTES:

1. \overline{SS} output mode (MODFEN = 1, SSOE = 1).
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

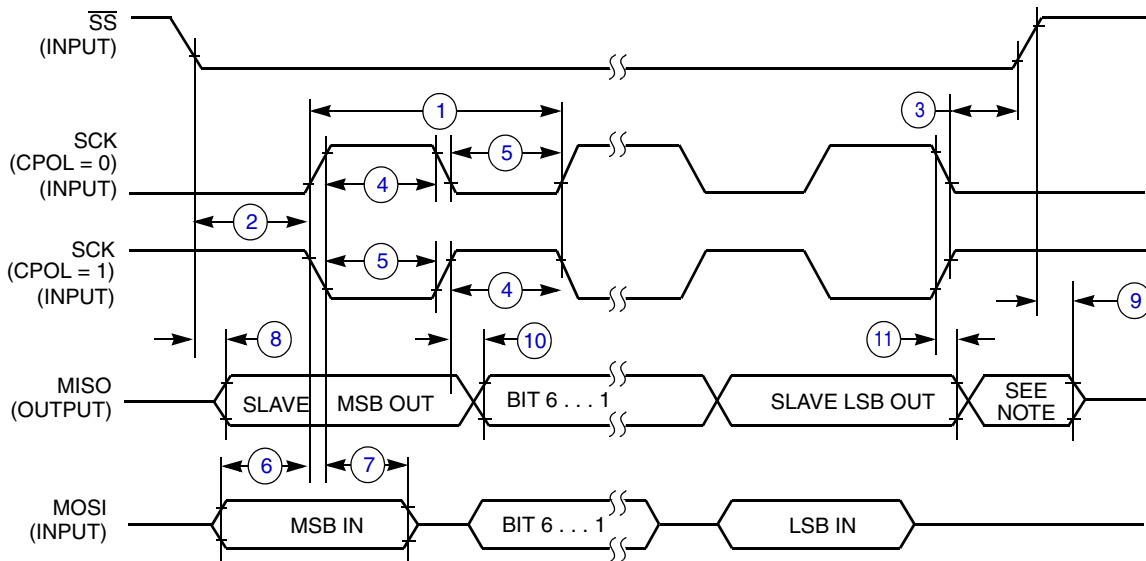
Figure A-14. SPI Master Timing (CPHA = 0)



NOTES:

1. \overline{SS} output mode (MODFEN = 1, SSOE = 1).
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

Figure A-15. SPI Master Timing (CPHA = 1)



NOTE:

1. Not defined but normally MSB of character just received

Figure A-16. SPI Slave Timing (CPHA = 0)