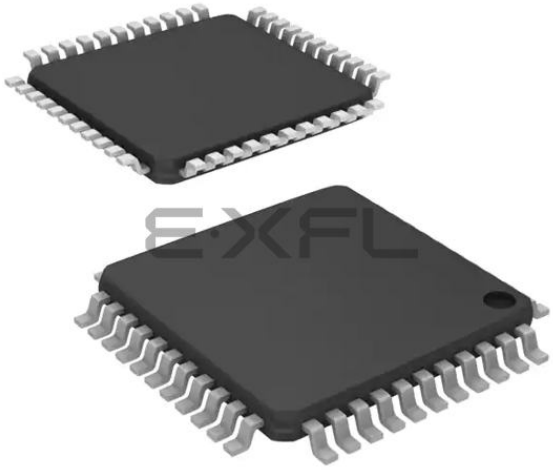


Welcome to E-XFL.COM

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	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PSMC, PWM, WDT
Number of I/O	35
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 14x12b; D/A 1x8b, 3x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1789-e-pt

1.0 DEVICE OVERVIEW

The PIC16(L)F1788/9 are described within this data sheet. The block diagram of these devices are shown in Figure 1-1. The available peripherals are shown in Table 1-1, and the pin out descriptions are shown in Tables 1-2 and 1-3.

TABLE 1-1: DEVICE PERIPHERAL SUMMARY

Peripheral		PIC16(L)F1782	PIC16(L)F1783	PIC16(L)F1784	PIC16(L)F1786	PIC16(L)F1787	PIC16(L)F1788	PIC16(L)F1789
Analog-to-Digital Converter (ADC)		●	●	●	●	●	●	●
Fixed Voltage Reference (FVR)		●	●	●	●	●	●	●
Reference Clock Module		●	●	●	●	●	●	●
Temperature Indicator		●	●	●	●	●	●	●
Capture/Compare/PWM (CCP/ECCP) Modules								
	CCP1	●	●	●	●	●	●	●
	CCP2	●	●	●	●	●	●	●
	CCP3			●	●	●	●	●
Comparators								
	C1	●	●	●	●	●	●	●
	C2	●	●	●	●	●	●	●
	C3	●	●	●	●	●	●	●
	C4			●	●	●	●	●
Digital-to-Analog Converter (DAC)								
	(8-bit DAC) D1	●	●	●	●	●	●	●
	(5-bit DAC) D2						●	●
	(5-bit DAC) D3						●	●
	(5-bit DAC) D4						●	●
Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter (EUSART)								
	EUSART	●	●	●	●	●	●	●
Master Synchronous Serial Ports								
	MSSP	●	●	●	●	●	●	●
Op Amp								
	Op Amp 1	●	●	●	●	●	●	●
	Op Amp 2	●	●	●	●	●	●	●
	Op Amp 3			●		●		●
Programmable Switch Mode Controller (PSMC)								
	PSMC1	●	●	●	●	●	●	●
	PSMC2	●	●	●	●	●	●	●
	PSMC3			●	●	●	●	●
	PSMC4						●	●
Timers								
	Timer0	●	●	●	●	●	●	●
	Timer1	●	●	●	●	●	●	●
	Timer2	●	●	●	●	●	●	●

TABLE 3-12: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 7											
38Ch	INLVLA	Input Type Control for PORTA								0000 0000	0000 0000
38Dh	INLVLB	Input Type Control for PORTB								0000 0000	0000 0000
38Eh	INLVLC	Input Type Control for PORTC								1111 1111	1111 1111
38Fh	INLVLD(3)	Input Type Control for PORTD								1111 1111	1111 1111
390h	INLVLE	—	—	—	—	INLVLE3	INLVLE2(3)	INLVLE1(3)	INLVLE0(3)	---- 1111	---- 1111
391h	IOCAP	IOCAP<7:0>								0000 0000	0000 0000
392h	IOCAN	IOCAN<7:0>								0000 0000	0000 0000
393h	IOCAF	IOCAF<7:0>								0000 0000	0000 0000
394h	IOCBP	IOCBP<7:0>								0000 0000	0000 0000
395h	IOCBN	IOCBN<7:0>								0000 0000	0000 0000
396h	IOCBF	IOCBF<7:0>								0000 0000	0000 0000
397h	IOCCP	IOCCP<7:0>								0000 0000	0000 0000
398h	IOCCN	IOCCN<7:0>								0000 0000	0000 0000
399h	IOCCF	IOCCF<7:0>								0000 0000	0000 0000
39Ah — 39Ch	—	Unimplemented								—	—
39Dh	IOCEP	—	—	—	—	IOCEP3	—	—	—	---- 0---	---- 0---
39Eh	IOCEN	—	—	—	—	IOCEN3	—	—	—	---- 0---	---- 0---
39Fh	IOCEF	—	—	—	—	IOCEF3	—	—	—	---- 0---	---- 0---
Bank 8-9											
40Ch or 41Fh and 48Ch or 49Fh	—	Unimplemented								—	—
Bank 10											
50Ch — 510h	—	Unimplemented								—	—
511h	OPA1CON	OPA1EN	OPA1SP	—	—	—	—	OPA1PCH<1:0>		00-- --00	00-- --00
512h	—	Unimplemented								—	—
513h	OPA2CON	OPA2EN	OPA2SP	—	—	—	—	OPA2PCH<1:0>		00-- --00	00-- --00
514h	—	Unimplemented								—	—
515h	OPA3CON(3)	OPA3EN	OPA3SP	—	—	—	OPA3PCH<2:0>			00-- -000	00-- -000
51Ah	CLKRCON	CLKREN	CLKROE	CLKRSLR	CLKRDC<1:0>		CLKRDIV<2:0>			0011 0000	0011 0000
51Bh — 51Fh	—	Unimplemented								—	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: These registers can be addressed from any bank.
2: Unimplemented, read as '1'.
3: PIC16(L)F1789 only.
4: PIC16F1788/9 only.

12.7 Register Definitions: EEPROM and Flash Control

REGISTER 12-1: EEDATL: EEPROM DATA LOW BYTE REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
EEDAT<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 **EEDAT<7:0>**: Read/write value for EEPROM data byte or Least Significant bits of program memory

REGISTER 12-2: EEDATH: EEPROM DATA HIGH BYTE REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	EEDAT<13:8>					
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **EEDAT<13:8>**: Read/write value for Most Significant bits of program memory

REGISTER 12-3: EEADRL: EEPROM ADDRESS REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EEADR<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 **EEADR<7:0>**: Specifies the Least Significant bits for program memory address or EEPROM address

REGISTER 12-4: EEADRH: EEPROM ADDRESS HIGH BYTE REGISTER

U-1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—(1)	EEADR<14:8>						
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **Unimplemented:** Read as ‘1’

bit 6-0 **EEADR<14:8>**: Specifies the Most Significant bits for program memory address or EEPROM address

Note 1: Unimplemented, read as ‘1’.

13.7 PORTC Registers

13.7.1 DATA REGISTER

PORTC is an 8-bit wide bidirectional port. The corresponding data direction register is TRISC (Register 13-20). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 13-1 shows how to initialize an I/O port.

Reading the PORTC register (Register 13-19) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATC).

13.7.2 DIRECTION CONTROL

The TRISC register (Register 13-20) controls the PORTC pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read '0'.

13.7.3 OPEN-DRAIN CONTROL

The ODCONC register (Register 13-23) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCONC bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCONC bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

13.7.4 SLEW RATE CONTROL

The SLRCONC register (Register 13-24) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONC bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONC bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

13.7.5 INPUT THRESHOLD CONTROL

The INLVLC register (Register 13-25) controls the input voltage threshold for each of the available PORTC input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTC register and also the level at which an interrupt-on-change occurs, if that feature is enabled. See **Section TABLE 31-1: "Supply Voltage"** for more information on threshold levels.

Note:	Changing the input threshold selection should be performed while all peripheral modules are disabled. Changing the threshold level during the time a module is active may inadvertently generate a transition associated with an input pin, regardless of the actual voltage level on that pin.

13.7.6 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each PORTC pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 13-7.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input and some digital input functions are not included in the list below. These input functions can remain active when the pin is configured as an output. Certain digital input functions override other port functions and are included in the priority list.

TABLE 13-7: PORTC OUTPUT PRIORITY

Pin Name	Function Priority(1)
RC0	T1OSO PSMC1A RC0
RC1	PSMC1B CCP2 RC1
RC2	PSMC1C CCP1 RC2
RC3	PSMC1D SCL SCK RC3
RC4	PSMC1E SDA RC4
RC5	PSMC1F SDO RC5
RC6	PSMC2A TX/CK CCP3 RC6
RC7	PSMC2B DT RC7

Note 1: Priority listed from highest to lowest.

13.9 PORTD Registers (PIC16(L)F1789 only)

13.9.1 DATA REGISTER

PORTD is an 8-bit wide bidirectional port. The corresponding data direction register is TRISD (Register 13-27). Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 13-1 shows how to initialize an I/O port.

Reading the PORTD register (Register 13-26) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATD).

13.9.2 DIRECTION CONTROL

The TRISD register (Register 13-27) controls the PORTD pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISD register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read '0'.

13.9.3 OPEN-DRAIN CONTROL

The ODCOND register (Register 13-31) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCOND bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCOND bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

13.9.4 SLEW RATE CONTROL

The SLRCOND register (Register 13-32) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCOND bit is set, the corresponding port pin drive is slew rate limited. When an SLRCOND bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

13.9.5 INPUT THRESHOLD CONTROL

The INLVLD register (Register 13-33) controls the input voltage threshold for each of the available PORTD input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTD register and also the level at which an interrupt-on-change occurs, if that feature is enabled. See **Section 31.3 "DC Characteristics"** for more information on threshold levels.

Note:	Changing the input threshold selection should be performed while all peripheral modules are disabled. Changing the threshold level during the time a module is active may inadvertently generate a transition associated with an input pin, regardless of the actual voltage level on that pin.

13.9.6 PORTD FUNCTIONS AND OUTPUT PRIORITIES

Each PORTD pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 13-9.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input and some digital input functions are not included in the list below. These input functions can remain active when the pin is configured as an output. Certain digital input functions override other port functions and are included in the priority list.

TABLE 13-9: PORTD OUTPUT PRIORITY

Pin Name	Function Priority(1)
RD0	RD0
RD1	OPA3OUT RD1
RD2	RD2
RD3	PSMC4A RD3
RD4	PSMC3F RD4
RD5	PSMC3E RD5
RD6	PSMC3D C3OUT RD6
RD7	PSMC3C C4OUT RD7

Note 1: Priority listed from highest to lowest.

14.6 Register Definitions: Interrupt-on-Change Control

REGISTER 14-1: IOCxP: INTERRUPT-ON-CHANGE POSITIVE EDGE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCxP7	IOCxP6	IOCxP5	IOCxP4	IOCxP3	IOCxP2	IOCxP1	IOCxP0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **IOCx<7:0>:** Interrupt-on-Change Positive Edge Enable bits(1)

1 = Interrupt-on-Change enabled on the pin for a positive going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.

0 = Interrupt-on-Change disabled for the associated pin.

Note 1: For IOCEP register, bit 3 (IOCEP3) is the only implemented bit in the register.

REGISTER 14-2: IOCxN: INTERRUPT-ON-CHANGE NEGATIVE EDGE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCxN7	IOCxN6	IOCxN5	IOCxN4	IOCxN3	IOCxN2	IOCxN1	IOCxN0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **IOCxN<7:0>:** Interrupt-on-Change Negative Edge Enable bits(1)

1 = Interrupt-on-Change enabled on the pin for a negative going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.

0 = Interrupt-on-Change disabled for the associated pin.

Note 1: For IOCEN register, bit 3 (IOCEN3) is the only implemented bit in the register.

FIGURE 15-1: VOLTAGE REFERENCE BLOCK DIAGRAM

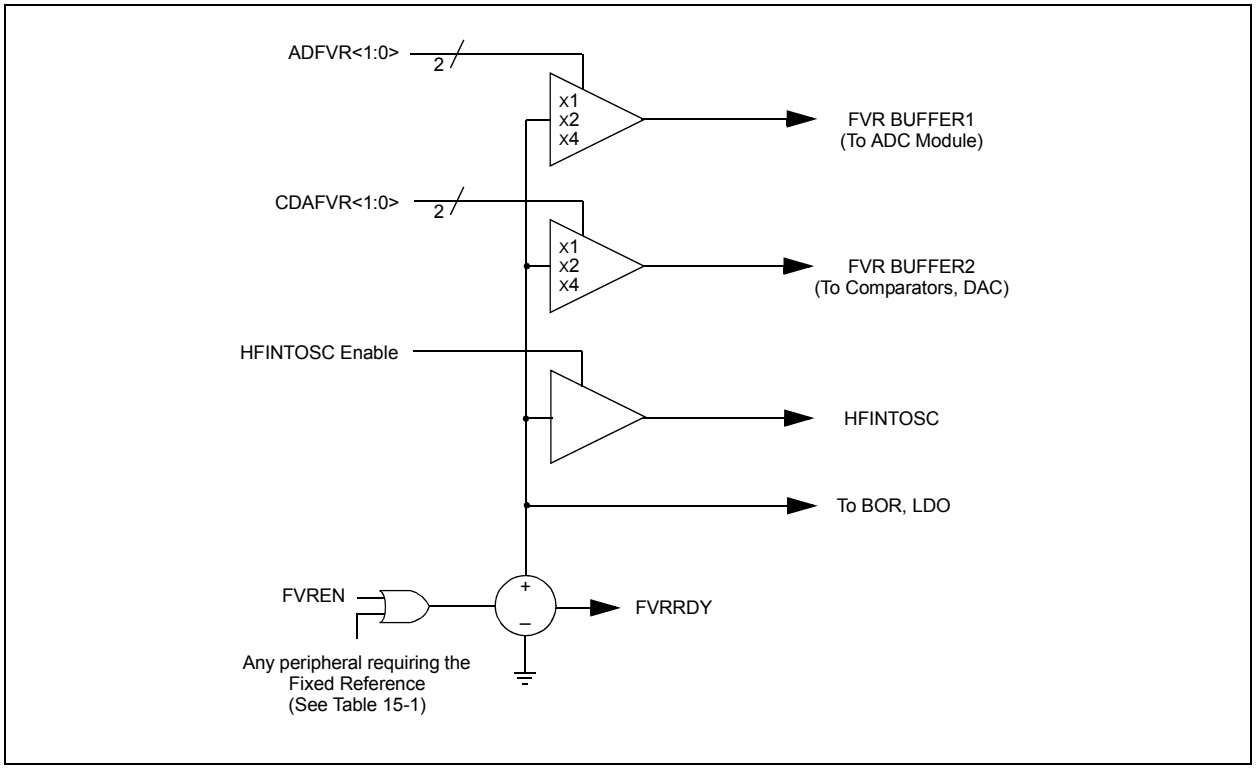


TABLE 15-1: PERIPHERALS REQUIRING THE FIXED VOLTAGE REFERENCE (FVR)

Peripheral	Conditions	Description
HFINTOSC	FOSC<2:0> = 100 and IRCF<3:0> ≠ 000x	INTOSC is active and device is not in Sleep
BOR	BOREN<1:0> = 11	BOR always enabled
	BOREN<1:0> = 10 and BORFS = 1	BOR disabled in Sleep mode, BOR Fast Start enabled.
	BOREN<1:0> = 01 and BORFS = 1	BOR under software control, BOR Fast Start enabled
LDO	All PIC16F1788/9 devices, when VREGPM = 1 and not in Sleep	The device runs off of the ULP regulator when in Sleep mode.
PSMC 64 MHz	PxSRC<1:0>	64 MHz clock forces HFINTOSC on during Sleep.

REGISTER 17-2: ADCON1: ADC CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>	
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **ADFM:** ADC Result Format Select bit (see Figure 17-3)
1 = 2’s complement format.
0 = Sign-magnitude result format.

bit 6-4 **ADCS<2:0>:** ADC Conversion Clock Select bits
111 = FRC (clock supplied from a dedicated FRC oscillator)
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC (clock supplied from a dedicated FRC oscillator)
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **ADNREF:** ADC Negative Voltage Reference Configuration bit
1 = VREF- is connected to external VREF- pin(1)
0 = VREF- is connected to VSS

bit 1-0 **ADPREF<1:0>:** ADC Positive Voltage Reference Configuration bits
11 = VREF+ is connected internally to FVR Buffer 1
10 = Reserved
01 = VREF+ is connected to VREF+ pin
00 = VREF+ is connected to VDD

Note 1: When selecting the FVR or VREF+ pin as the source of the positive reference, be aware that a minimum voltage specification exists. See **Section 31.0 “Electrical Specifications”** for details.

21.2 Comparator Control

Each comparator has two control registers: CMxCON0 and CMxCON1.

The CMxCON0 register (see Register 21-1) contains Control and Status bits for the following:

- Enable
- Output selection
- Output polarity
- Speed/Power selection
- Hysteresis enable
- Output synchronization

The CMxCON1 register (see Register 21-2) contains Control bits for the following:

- Interrupt enable
- Interrupt edge polarity
- Positive input channel selection
- Negative input channel selection

21.2.1 COMPARATOR ENABLE

Setting the CxON bit of the CMxCON0 register enables the comparator for operation. Clearing the CxON bit disables the comparator resulting in minimum current consumption.

21.2.2 COMPARATOR OUTPUT SELECTION

The output of the comparator can be monitored by reading either the CxOUT bit of the CMxCON0 register or the MCxOUT bit of the CMOUT register. In order to make the output available for an external connection, the following conditions must be true:

- CxOE bit of the CMxCON0 register must be set
- Corresponding TRIS bit must be cleared
- CxON bit of the CMxCON0 register must be set

Note 1:	The CxOE bit of the CMxCON0 register overrides the PORT data latch. Setting the CxON bit of the CMxCON0 register has no impact on the port override.
2:	The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched.

21.2.3 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the CxPOL bit of the CMxCON0 register. Clearing the CxPOL bit results in a non-inverted output.

Table 21-2 shows the output state versus input conditions, including polarity control.

TABLE 21-2: COMPARATOR OUTPUT STATE VS. INPUT CONDITIONS

Input Condition	CxPOL	CxOUT
CxVN > CxVP	0	0
CxVN < CxVP	0	1
CxVN > CxVP	1	1
CxVN < CxVP	1	0

21.2.4 COMPARATOR SPEED/POWER SELECTION

The trade-off between speed or power can be optimized during program execution with the CxSP control bit. The default state for this bit is '1' which selects the normal speed mode. Device power consumption can be optimized at the cost of slower comparator propagation delay by clearing the CxSP bit to '0'.

26.2 Event Sources

There are two main sources for the period, rising edge and falling edge events:

- Synchronous input
  - Time base
- Asynchronous Inputs
  - Digital Inputs
  - Analog inputs

26.2.1 TIME BASE

The Time Base section consists of several smaller pieces.

- 16-bit time base counter
- 16-bit Period register
- 16-bit Phase register (rising edge event)
- 16-bit Duty Cycle register (falling edge event)
- Clock control
- Interrupt Generator

An example of a fully synchronous PWM waveform generated with the time base is shown in Figure 26-2.

The PSMCxLD bit of the PSMCxCON register is provided to synchronize changes to the event Count registers. Changes are withheld from taking action until the first period event Reset after the PSMCxLD bit is set. For example, to change the PWM frequency, while maintaining the same effective duty cycle, the Period and Duty Cycle registers need to be changed. The changes to all four registers take effect simultaneously on the period event Reset after the PSMCxLD bit is set.

26.2.1.1 16-bit Counter (Time Base)

The PSMCxTMR is the counter used as a timing reference for each synchronous PWM period. The counter starts at 0000h and increments to FFFFh on the rising edge of the psmc_clk signal.

When the counter rolls over from FFFFh to 0000h without a period event occurring, the overflow interrupt will be generated, thereby setting the PxTOVIF bit of the PSMC Time Base Interrupt Control (PSMCxINT) register (Register 26-34).

The PSMCxTMR counter is reset on both synchronous and asynchronous period events.

The PSMCxTMR is accessible to software as two 8-bit registers:

- PSMC Time Base Counter Low (PSMCxTMRL) register (Register 26-19)
- PSMC PSMC Time Base Counter High (PSMCxTMRH) register (Register 26-20)

PSMCxTMR is reset to the default POR value when the PSMCxEN bit is cleared.

26.2.1.2 16-bit Period Register

The PSMCxPR Period register is used to determine a synchronous period event referenced to the 16-bit PSMCxTMR digital counter. A match between the PSMCxTMR and PSMCxPR register values will generate a period event.

The match will generate a period match interrupt, thereby setting the PxTPRIF bit of the PSMC Time Base Interrupt Control (PSMCxINT) register (Register 26-34).

The 16-bit period value is accessible to software as two 8-bit registers:

- PSMC Period Count Low Byte (PSMCxPRL) register (Register 26-25)
- PSMC Period Count High Byte (PSMCxPRH) register (Register 26-26)

The 16-bit period value is double-buffered before it is presented to the 16-bit time base for comparison. The buffered registers are updated on the first period event Reset after the PSMCxLD bit of the PSMCxCON register is set.

The synchronous PWM period time can be determined from Equation 26-1.

EQUATION 26-1: PWM PERIOD

$$Period = \frac{\text{PSMCxPR}[15:0] + 1}{F_{\text{psmc\_clk}}}$$

26.2.1.3 16-bit Phase Register

The PSMCxPH Phase register is used to determine a synchronous rising edge event referenced to the 16-bit PSMCxTMR digital counter. A match between the PSMCxTMR and the PSMCxPH register values will generate a rising edge event.

The match will generate a phase match interrupt, thereby setting the PxTPHIF bit of the PSMC Time Base Interrupt Control (PSMCxINT) register (Register 26-34).

The 16-bit phase value is accessible to software as two 8-bit registers:

- PSMC Phase Count Low Byte (PSMCxPHL) register (Register 26-34)
- PSMC Phase Count High Byte (PSMCxPHH) register (Register 26-34)

The 16-bit phase value is double-buffered before it is presented to the 16-bit PSMCxTMR for comparison. The buffered registers are updated on the first period event Reset after the PSMCxLD bit of the PSMCxCON register is set.

26.3.11 VARIABLE FREQUENCY - FIXED DUTY CYCLE PWM WITH COMPLEMENTARY OUTPUTS

This mode is the same as the single output Fixed Duty Cycle mode except a complementary output with dead-band control is generated.

The rising edge and falling edge events are unused in this mode. Therefore, a different triggering mechanism is required for the dead-band counters.

A period events that generate a rising edge on PSMCxA use the rising edge dead-band counters.

A period events that generate a falling edge on PSMCxA use the falling edge dead-band counters.

26.3.11.1 Mode Features

- Dead-band control is available
- No steering control available
- Fractional Frequency Adjust
  - Fine period adjustments are made with the PSMC Fractional Frequency Adjust (PSMCxFFA) register (Register 26-29)
- Primary PWM is output to the following pin:
  - PSMCxA
- Complementary PWM is output to the following pin:
  - PSMCxB

26.3.11.2 Waveform Generation

Period Event

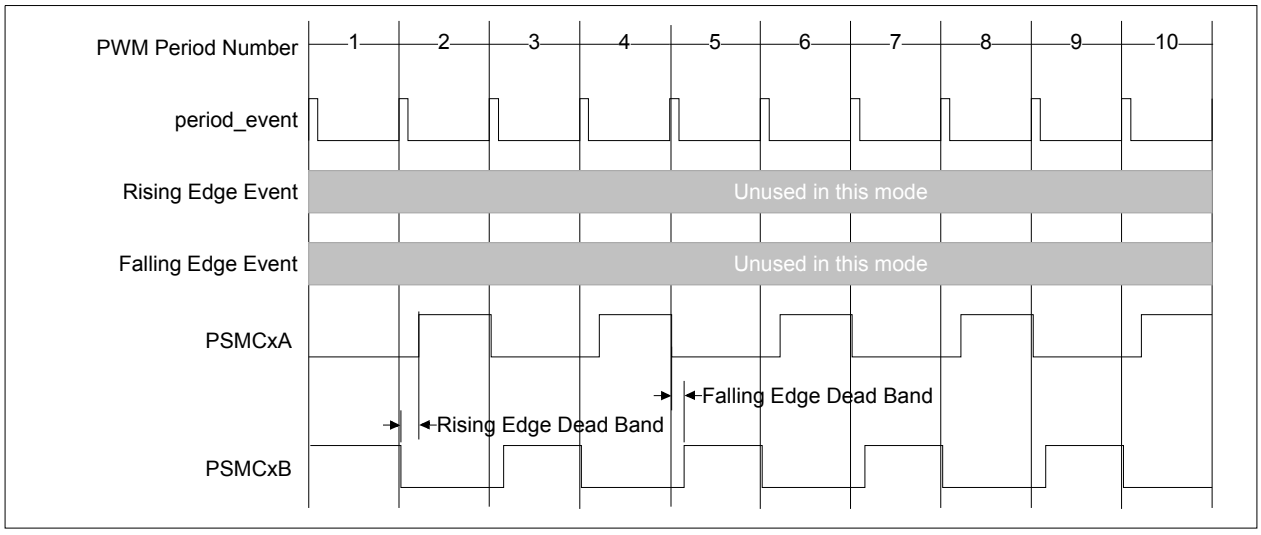
When output is going inactive to active:

- Complementary output is set inactive
- FFA counter is incremented by the 4-bit value in PSMCFFA register.
- Dead-band rising is activated (if enabled)
- Primary output is set active

When output is going active to inactive:

- Primary output is set inactive
- FFA counter is incremented by the 4-bit value in PSMCFFA register
- Dead-band falling is activated (if enabled)
- Complementary output is set active

FIGURE 26-14: VARIABLE FREQUENCY – FIXED DUTY CYCLE PWM WITH COMPLEMENTARY OUTPUTS WAVEFORM



REGISTER 26-9: PSMCxPOL: PSMC POLARITY CONTROL REGISTER

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	PxPOLIN	PxPOLF(1)	PxPOLE(1)	PxPOLD(1)	PxPOLC(1)	PxPOLB	PxPOLA
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **Unimplemented:** Read as '0'

bit 6 **PxPOLIN:** PSMCxIN Polarity bit
- 1 = PSMCxIN input is active-low
- 0 = PSMCxIN input is active-high

bit 5-0 **PxPOLy:** PSMCx Output y Polarity bit(1)
- 1 = PWM PSMCx output y is active-low
- 0 = PWM PSMCx output y is active-high

Note 1: These bits are not implemented on PSMC2.

REGISTER 26-10: PSMCxBLNK: PSMC BLANKING CONTROL REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	PxFEBM1	PxFEBM0	—	—	PxREBM1	PxREBM0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **PxFEBM<1:0>** PSMC Falling Edge Blanking Mode bits
- 11 = Reserved – do not use
- 10 = Reserved – do not use
- 01 = Immediate blanking
- 00 = No blanking

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **PxREBM<1:0>** PSMC Rising Edge Blanking Mode bits
- 11 = Reserved – do not use
- 10 = Reserved – do not use
- 01 = Immediate blanking
- 00 = No blanking

REGISTER 26-17: PSMCxASDL: PSMC AUTO-SHUTDOWN OUTPUT LEVEL REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	PxASDLF(1)	PxASDLE(1)	PxASDLD(1)	PxASDLC(1)	PxASDLB	PxASDLA
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **PxASDLF:** PSMCx Output F Auto-Shutdown Pin Level bit(1)
- 1 = When auto-shutdown is asserted, pin PSMCxF will drive logic '1'
- 0 = When auto-shutdown is asserted, pin PSMCxF will drive logic '0'

bit 4 **PxASDLE:** PSMCx Output E Auto-Shutdown Pin Level bit(1)
- 1 = When auto-shutdown is asserted, pin PSMCxE will drive logic '1'
- 0 = When auto-shutdown is asserted, pin PSMCxE will drive logic '0'

bit 3 **PxASDLD:** PSMCx Output D Auto-Shutdown Pin Level bit(1)
- 1 = When auto-shutdown is asserted, pin PSMCxD will drive logic '1'
- 0 = When auto-shutdown is asserted, pin PSMCxD will drive logic '0'

bit 2 **PxASDLC:** PSMCx Output C Auto-Shutdown Pin Level bit(1)
- 1 = When auto-shutdown is asserted, pin PSMCxC will drive logic '1'
- 0 = When auto-shutdown is asserted, pin PSMCxC will drive logic '0'

bit 1 **PxASDLB:** PSMCx Output B Auto-Shutdown Pin Level bit
- 1 = When auto-shutdown is asserted, pin PSMCxB will drive logic '1'
- 0 = When auto-shutdown is asserted, pin PSMCxB will drive logic '0'

bit 0 **PxASDLA:** PSMCx Output A Auto-Shutdown Pin Level bit
- 1 = When auto-shutdown is asserted, pin PSMCxA will drive logic '1'
- 0 = When auto-shutdown is asserted, pin PSMCxA will drive logic '0'

Note 1: These bits are not implemented on PSMC2.

REGISTER 26-18: PSMCxASDS: PSMC AUTO-SHUTDOWN SOURCE REGISTER

R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0
PxASDSIN	—	—	PxASDSC4	PxASDSC3	PxASDSC2	PxASDSC1	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **PxASDSIN:** Auto-shutdown occurs on PSMCxIN pin
1 = Auto-shutdown will occur when PSMCxIN pin goes true
0 = PSMCxIN pin will not cause auto-shutdown

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **PxASDSC4:** Auto-shutdown occurs on sync_C4OUT output
1 = Auto-shutdown will occur when sync_C4OUT output goes true
0 = sync_C4OUT will not cause auto-shutdown

bit 3 **PxASDSC3:** Auto-shutdown occurs on sync_C3OUT output
1 = Auto-shutdown will occur when sync_C3OUT output goes true
0 = sync_C3OUT will not cause auto-shutdown

bit 2 **PxASDSC2:** Auto-shutdown occurs on sync_C2OUT output
1 = Auto-shutdown will occur when sync_C2OUT output goes true
0 = sync_C2OUT will not cause auto-shutdown

bit 1 **PxASDSC1:** Auto-shutdown occurs on sync_C1OUT output
1 = Auto-shutdown will occur when sync_C1OU output goes true
0 = sync_C1OU will not cause auto-shutdown

bit 0 **Unimplemented:** Read as '0'

27.2.2 SPI MODE OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

To enable the serial port, SSP Enable bit, SSPEN of the SSPCON1 register, must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPCONx registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

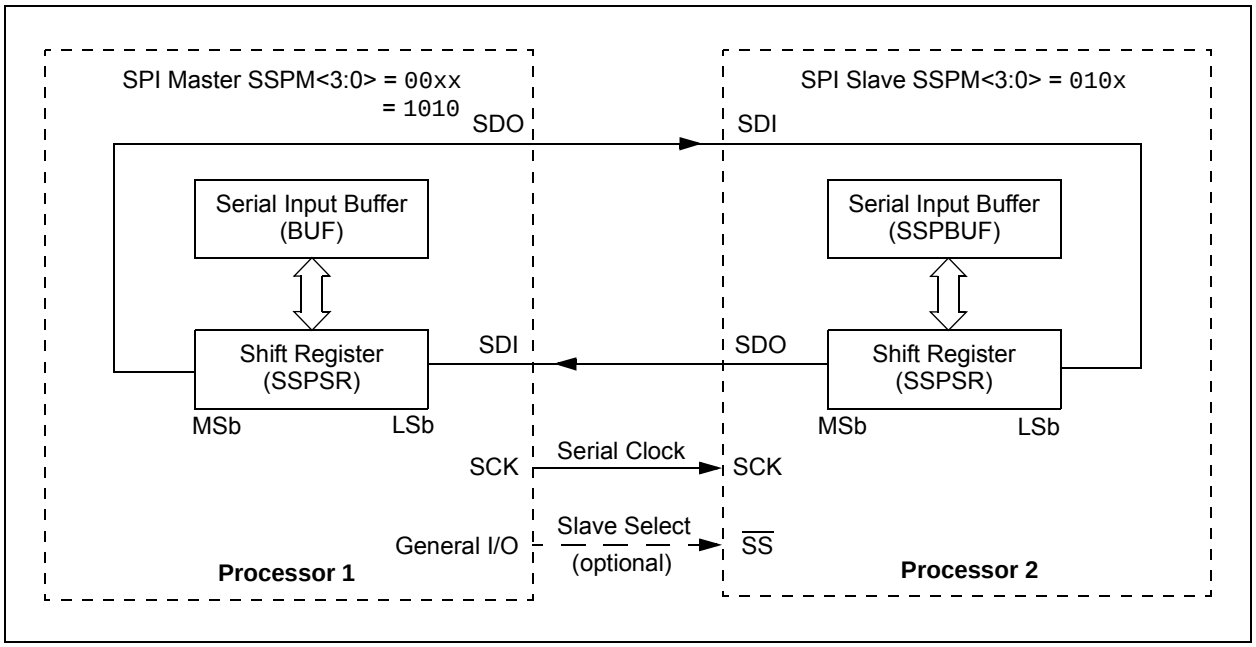
- SDI must have corresponding TRIS bit set
- SDO must have corresponding TRIS bit cleared
- SCK (Master mode) must have corresponding TRIS bit cleared
- SCK (Slave mode) must have corresponding TRIS bit set
- $\overline{SS}$ must have corresponding TRIS bit set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the eight bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full Detect bit, BF of the SSPSTAT register, and the interrupt flag bit, SSP1IF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the write collision detect bit WCOL of the SSPCON1 register, will be set. User software must clear the WCOL bit to allow the following write(s) to the SSPBUF register to complete successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The Buffer Full bit, BF of the SSPSTAT register, indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur.

FIGURE 27-5: SPI MASTER/SLAVE CONNECTION



27.5.8 GENERAL CALL ADDRESS SUPPORT

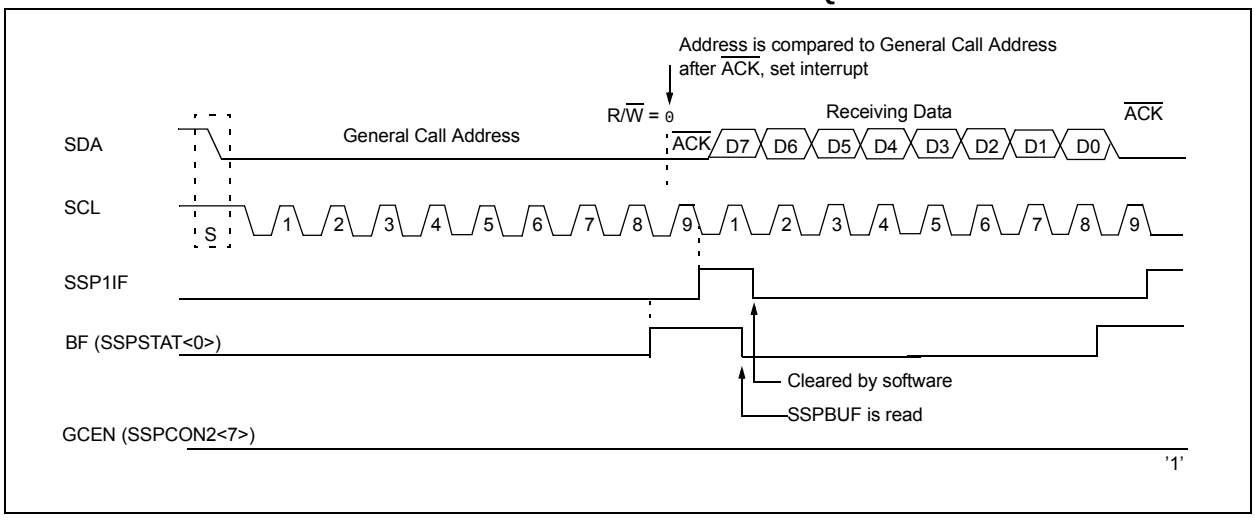
The addressing procedure for the I^2C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is a reserved address in the I^2C protocol, defined as address 0x00. When the GCEN bit of the SSPCON2 register is set, the slave module will automatically $\overline{ACK}$ the reception of this address regardless of the value stored in SSPADD. After the slave clocks in an address of all zeros with the R/$\overline{W}$ bit clear, an interrupt is generated and slave software can read SSPBUF and respond. Figure 27-23 shows a general call reception sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit of the SSPCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the 8th falling edge of SCL. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

FIGURE 27-24: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE



27.5.9 SSP MASK REGISTER

An SSP Mask (SSPMSK) register (Register 27-5) is available in I^2C Slave mode as a mask for the value held in the SSPSR register during an address comparison operation. A zero ('0') bit in the SSPMSK register has the effect of making the corresponding bit of the received address a "don't care".

This register is reset to all '1's upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

The SSP Mask register is active during:

- 7-bit Address mode: address compare of A<7:1>.
- 10-bit Address mode: address compare of A<7:0> only. The SSP mask has no effect during the reception of the first (high) byte of the address.

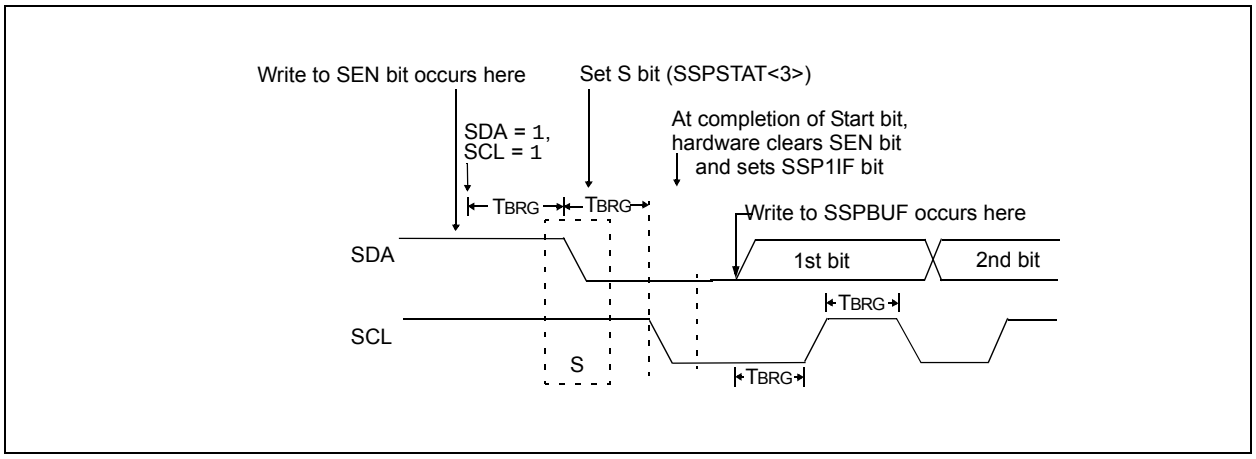
27.6.4 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start Enable bit, SEN bit of the SSPCON2 register. If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<7:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the Start condition and causes the S bit of the SSPSTAT1 register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPADD<7:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit of the SSPCON2 register will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

Note 1: If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCL1IF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.

2: The Philips I²C specification states that a bus collision cannot occur on a Start.

FIGURE 27-26: FIRST START BIT TIMING



27.6.13.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

a) A low level is sampled on SDA when SCL goes from low level to high level.
b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

When the user releases SDA and the pin is allowed to float high, the BRG is loaded with SSPADD and counts down to zero. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 27-35). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, see Figure 27-36.

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

FIGURE 27-36: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

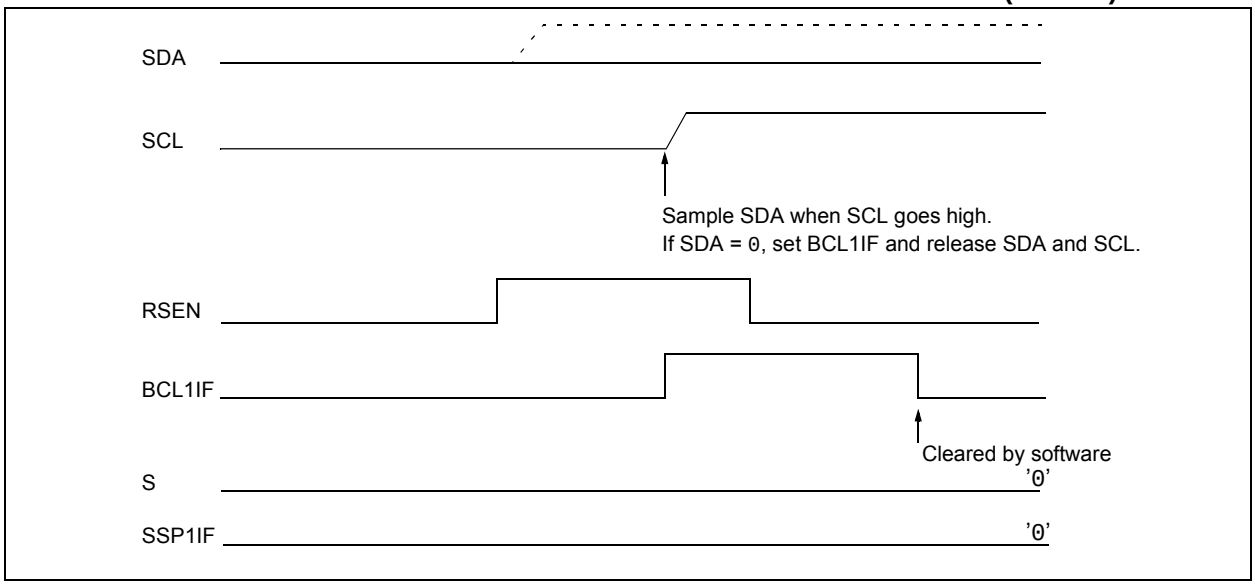
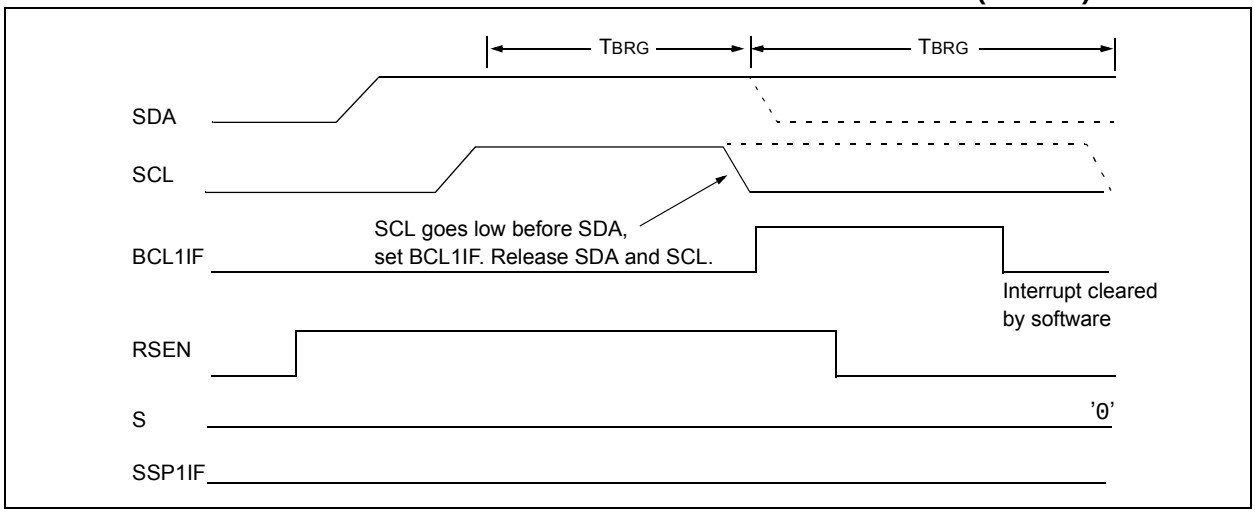


FIGURE 27-37: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



28.4 EUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUDCON register selects 16-bit mode.

The SPBRGH, SPBRGL register pair determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by both the BRGH bit of the TXSTA register and the BRG16 bit of the BAUDCON register. In Synchronous mode, the BRGH bit is ignored.

Table 28-3 contains the formulas for determining the baud rate. Example 28-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various asynchronous modes have been computed for your convenience and are shown in Table 28-3. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SPBRGH, SPBRGL register pair causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is idle before changing the system clock.

EXAMPLE 28-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$Desired\ Baud\ Rate = \frac{FOSC}{64([SPBRGH:SPBRGL] + 1)}$$

Solving for SPBRGH:SPBRGL:

$$X = \frac{\frac{FOSC}{Desired\ Baud\ Rate}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

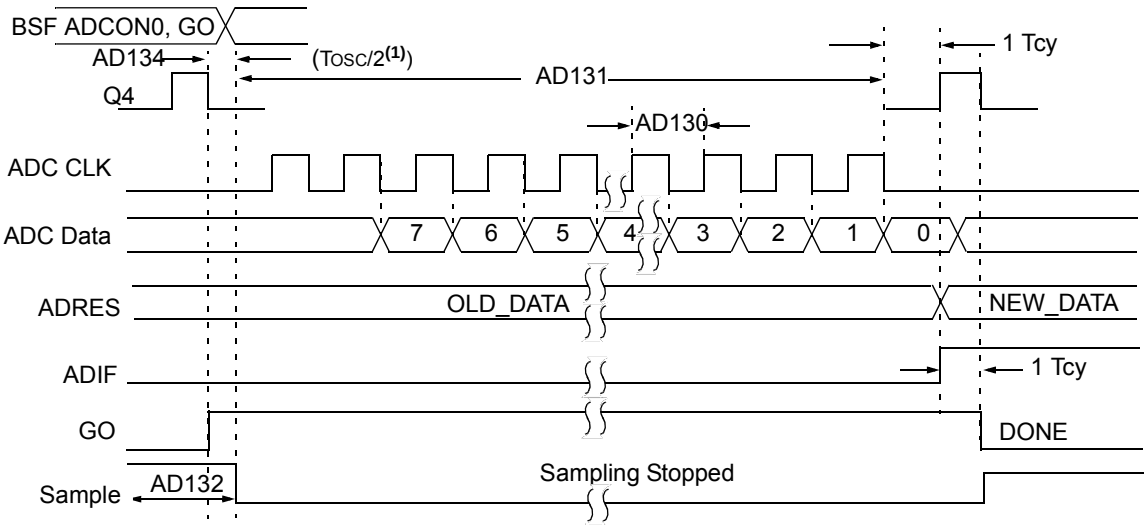
$$Calculated\ Baud\ Rate = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$Error = \frac{Calc.\ Baud\ Rate - Desired\ Baud\ Rate}{Desired\ Baud\ Rate}$$

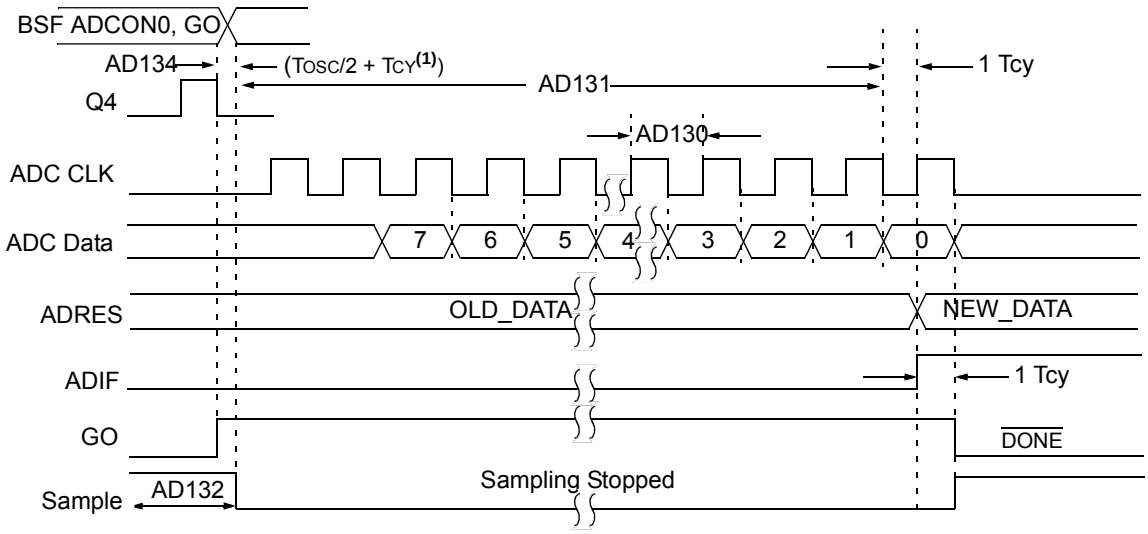
$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

FIGURE 31-12: ADC CONVERSION TIMING (NORMAL MODE)



Note 1: If the ADC clock source is selected as RC, a time of TCY is added before the ADC clock starts. This allows the SLEEP instruction to be executed.

FIGURE 31-13: ADC CONVERSION TIMING (SLEEP MODE)



Note 1: If the ADC clock source is selected as RC, a time of TCY is added before the ADC clock starts. This allows the SLEEP instruction to be executed.