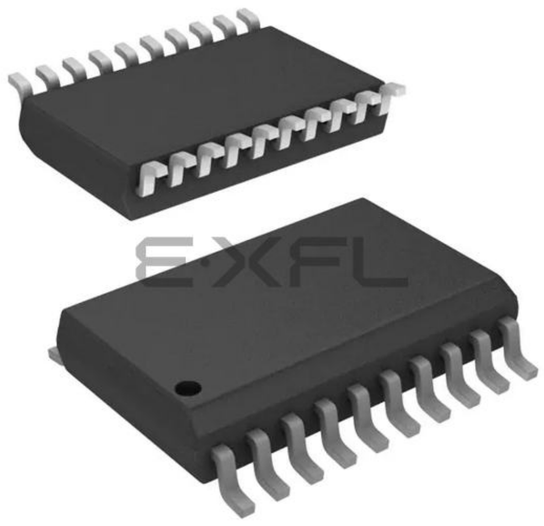
Microchip Technology - PIC16LF18446-I/SO Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

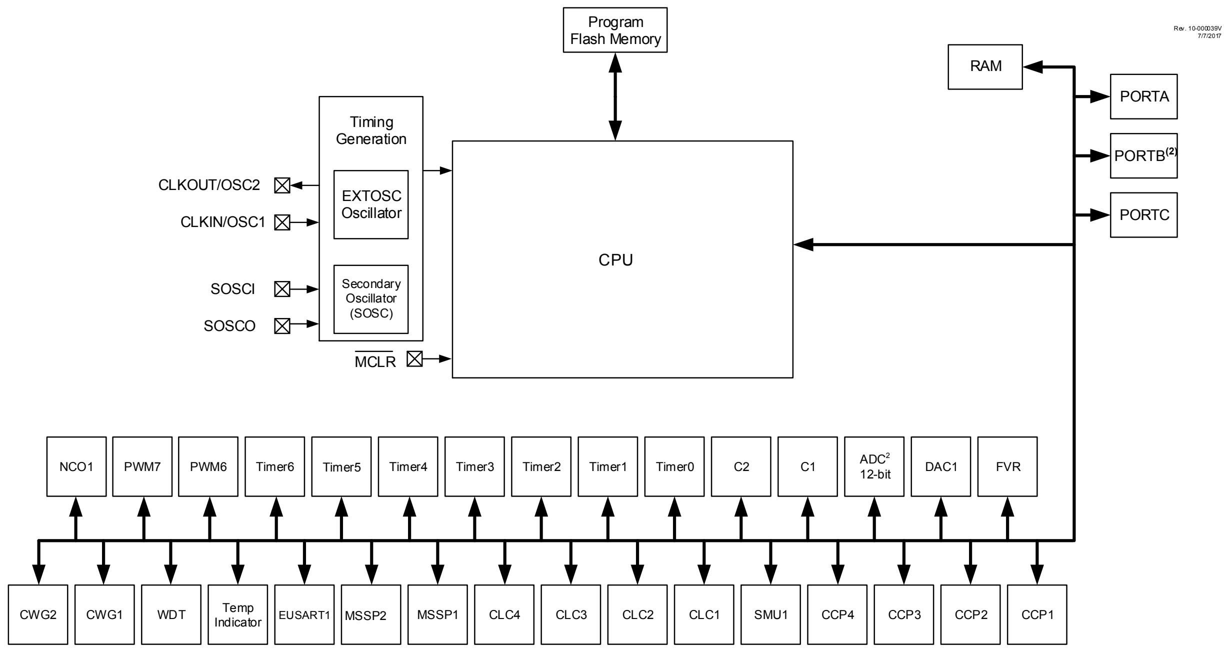
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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, PWM, WDT
Number of I/O	18
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 17x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf18446-i-so

Features	PIC16(L)F18426	PIC16(L)F18446
	16-levels hardware stack	16-levels hardware stack
Operating Frequency	DC – 32 MHz	DC – 32 MHz

Figure 1-1. PIC16(L)F18426/46 Device Block Diagram



Note:

1. See applicable chapters for more information on peripherals.
2. PORTB available only on 20-pin or higher pin-count devices.

1.4 Register and Bit naming conventions

1.4.1 Register Names

When there are multiple instances of the same peripheral in a device, the peripheral control registers will be depicted as the concatenation of a peripheral identifier, peripheral instance, and control identifier. The control registers section will show just one instance of all the register names with an ‘x’ in the place of the peripheral instance number. This naming convention may also be applied to peripherals when there is only one instance of that peripheral in the device to maintain compatibility with other devices in the family that contain more than one.

1.4.2 Bit Names

There are two variants for bit names:

- Short name: Bit function abbreviation
- Long name: Peripheral abbreviation + short name

10.7.7 PIE5

Name: PIE5
Address: 0x71B

Peripheral Interrupt Enable Register 5

Bit	7	6	5	4	3	2	1	0
	CLC4IE	CLC3IE	CLC2IE	CLC1IE		TMR5GIE	TMR3GIE	TMR1GIE
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 7 – CLC4IE CLC4 Interrupt Enable bit

Value	Description
1	Enabled
0	Disabled

Bit 6 – CLC3IE CLC3 Interrupt Enable bit

Value	Description
1	Enabled
0	Disabled

Bit 5 – CLC2IE CLC2 Interrupt Enable bit

Value	Description
1	Enabled
0	Disabled

Bit 4 – CLC1IE CLC1 Interrupt Enable bit

Value	Description
1	Enabled
0	Disabled

Bit 2 – TMR5GIE TMR5 Gate Interrupt Enable bit

Value	Description
1	Enabled
0	Disabled

Bit 1 – TMR3GIE TMR3 Gate Interrupt Enable bit

Value	Description
1	Enabled
0	Disabled

Bit 0 – TMR1GIE TMR1 Gate Interrupt Enable bit

10.7.8 PIE6

Name: PIE6
Address: 0x71C

Peripheral Interrupt Enable Register 6

Bit	7	6	5	4	3	2	1	0
					CCP4IE	CCP3IE	CCP2IE	CCP1IE
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – CCP4IE CCP4 Interrupt Enable bit

Value	Description
1	Enabled
0	Disabled

Bit 2 – CCP3IE CCP3 Interrupt Enable bit

Value	Description
1	Enabled
0	Disabled

Bit 1 – CCP2IE CCP2 Interrupt Enable bit

Value	Description
1	Enabled
0	Disabled

Bit 0 – CCP1IE CCP1 Interrupt Enable bit

Value	Description
1	Enabled
0	Disabled

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

Important: To modify only a portion of a previously programmed row, the contents of the entire row must be read. Then, the new data and retained data can be written into the write latches to reprogram the row of program memory. However, any unprogrammed locations can be written without first erasing the row. In this case, it is not necessary to save and rewrite the other previously programmed locations

Related Links
FSR and INDF Access
NVMREG Access

13.1.1 Program Memory Voltages

The program memory is readable and writable during normal operation over the full V_{DD} range.

13.1.1.1 Programming Externally

The program memory cell and control logic support write and Bulk Erase operations down to the minimum device operating voltage. Special BOR operation is enabled during Bulk Erase.

Related Links
BOR is Always Off

13.1.1.2 Self-programming

The program memory cell and control logic will support write and row erase operations across the entire V_{DD} range. Bulk Erase is not available when self-programming.

13.2 Data EEPROM

Data EEPROM consists of 256 bytes of user data memory. The EEPROM provides storage locations for 8-bit user defined data.

EEPROM can be read and/or written through:
- FSR/INDF indirect access
- NVMREG access
- External device programmer

Unlike program Flash memory, which must be written to by row, EEPROM can be written to byte by byte.

Related Links
FSR and INDF Access
NVMREG Access

13.3 FSR and INDF Access

The FSR and INDF registers allow indirect access to the program memory or EEPROM.

Related Links
FSR0

16.5.3 PMD2

Name: PMD2
Address: 0x798

PMD Control Register 2

Bit	7	6	5	4	3	2	1	0
	NCO1MD							
Access	R/W							
Reset	0							

Bit 7 – NCO1MD Disable Numerically Control Oscillator bit

Value	Description
1	NCO1 module disabled
0	NCO1 module enabled

16.5.4 PMD3

Name: PMD3
Address: 0x799

PMD Control Register 3

Bit	7	6	5	4	3	2	1	0
		DAC1MD	ADCMD			C2MD	C1MD	ZCDMD
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit 6 – DAC1MD Disable DAC1 bit

Value	Description
1	DAC module disabled
0	DAC module enabled

Bit 5 – ADCMD Disable ADC bit

Value	Description
1	ADC module disabled
0	ADC module enabled

Bit 2 – C2MD Disable Comparator C2 bit

Value	Description
1	C2 module disabled
0	C2 module enabled

Bit 1 – C1MD Disable Comparator C1 bit

Value	Description
1	C1 module disabled
0	C1 module enabled

Bit 0 – ZCDMD Disable Zero-Cross Detect module bit

Value	Description
1	ZCD module disabled
0	ZCD module enabled

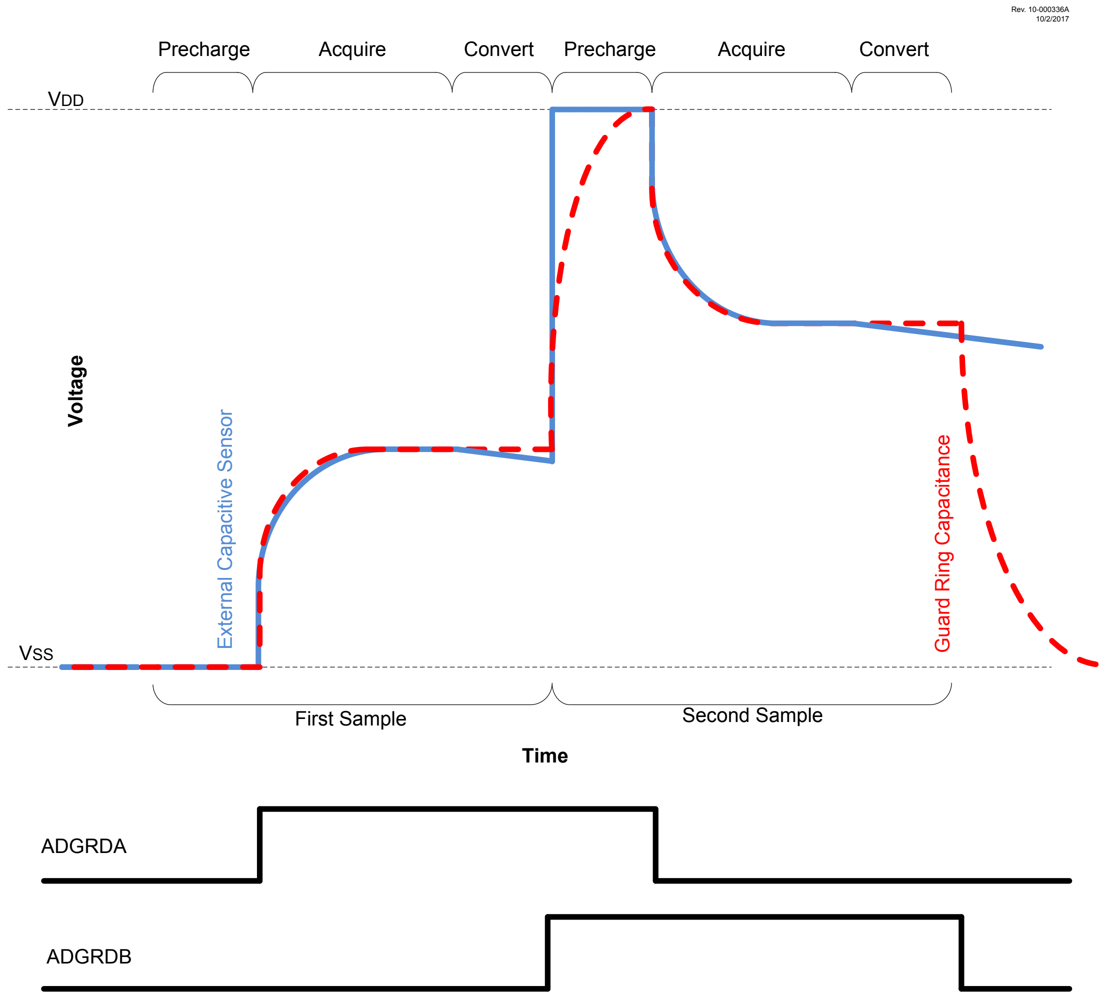
Table 20-3. ADC Auto-Conversion Trigger Sources

ACT	Auto-conversion Trigger Source
11111	Software write to ADPCH
11110	Reserved, do not use
11101	Software read of ADRESH
11100	Software read of ADERRH
11011 to 11000	Reserved, do not use
10111	CLC4_out
10110	CLC3_out
10101	CLC2_out
10100	CLC1_out
10011	Logical `OR` of all Interrupt-on-change Interrupt Flags
10010	C2_out
10001	C1_out
10000	NCO1_out
01111	PWM7_out
01110	PWM6_out
01101	CCP4_trigger
01100	CCP3_trigger
01011	CCP2_trigger
01010	CCP1_trigger
01001	SMT1_trigger
01000	TMR6_postscaled
00111	TMR5_overflow
00110	TMR4_postscaled
00101	TMR3_overflow
00100	TMR2_postscaled
00011	TMR1_overflow
00010	TMR0_overflow
00001	Pin selected by ADACTPPS
00000	External Trigger Disabled

20.2.7 ADC Conversion Procedure (Basic Mode)

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

Figure 20-9. Differential CVD with Guard Ring Output Waveform



Related Links

ADCON1

(PPS) Peripheral Pin Select Module

20.5.5 Additional Sample and Hold Capacitance

Additional capacitance can be added in parallel with the internal sample and hold capacitor (C_{HOLD}) by using the ADCAP register. This register selects a digitally programmable capacitance which is added to the ADC conversion bus, increasing the effective internal capacitance of the sample and hold capacitor in the ADC module. This is used to improve the match between internal and external capacitance for a better sensing performance. The additional capacitance does not affect analog performance of the ADC because it is not connected during conversion.

Related Links

Computation Operation

ADCAP

20.8.8 ADPCH

Name: ADPCH
Address: 0x09F

ADC Positive Channel Selection Register

Bit	7	6	5	4	3	2	1	0
	▨	▨	PCH[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – PCH[5:0] ADC Positive Input Channel Selection bits
See ADC Positive Input Channel Selections for input selection details.

20.8.16 ADPREV

Name: ADPREV
Address: 0x09B

ADC Previous Result Register

Bit	15	14	13	12	11	10	9	8
	PREVH[7:0]							
Access	RO	RO	RO	RO	RO	RO	RO	RO
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	PREVL[7:0]							
Access	RO	RO	RO	RO	RO	RO	RO	RO
Reset	x	x	x	x	x	x	x	x

Bits 15:8 – PREVH[7:0] Previous ADC Result Most Significant bits

Value	Condition	Description
0 to 0xFF	PSIS = 1	Upper byte of ADFLTR at the start of current ADC conversion
varies	PSIS = 0	Upper bits of ADRES at the start of current ADC conversion(1)

Bits 7:0 – PREVL[7:0] Previous ADC Result Least Significant bits

Value	Condition	Description
0 to 0xFF	PSIS = 1	Lower byte of ADFLTR at the start of current ADC conversion
varies	PSIS = 0	Lower bits of ADRES at the start of current ADC conversion(1)

Note: If PSIS = 0, PREVH and PREVL are formatted the same way as ADRES is, depending on the FRM bit.

33.8.7 CLCxGLS0

Name: CLCxGLS0
Address: 0x1E16,0x1E20,0x1E2A,0x1E34

CLCx Gate1 Logic Select Register

Bit	7	6	5	4	3	2	1	0
	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 1, 3, 5, 7 – G1DyT
dyT: Gate1 Data 'y' True (non-inverted) bit
Reset States: Default = xxxx
POR/BOR = x
All Other Resets = u

Value	Description
1	dyT is gated into g1
0	dyT is not gated into g1

Bits 0, 2, 4, 6 – G1DyN
dyN: Gate1 Data 'y' Negated (inverted) bit
Reset States: Default = xxxx
POR/BOR = x
All Other Resets = u

Value	Description
1	dyN is gated into g1
0	dyN is not gated into g1

35. (MSSP) Master Synchronous Serial Port Module

The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)

The SPI interface supports the following modes and features:

- Master mode
- Slave mode
- Clock Parity
- Slave Select Synchronization (Slave mode only)
- Daisy-chain connection of slave devices

The I^2C interface supports the following modes and features:

- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- Limited multi-master support
- 7-bit and 10-bit addressing
- Start and Stop interrupts
- Interrupt masking
- Clock stretching
- Bus collision detection
- General call address matching
- Address masking
- Address Hold and Data Hold modes
- Selectable SDA hold times

35.1 SPI Mode Overview

The Serial Peripheral Interface (SPI) bus is a synchronous serial data communication bus that operates in Full-Duplex mode. Devices communicate in a master/slave environment where the master device initiates the communication. A slave device is controlled through a Chip Select known as Slave Select.

The SPI bus specifies four signal connections:

- Serial Clock (SCK)
- Serial Data Out (SDO)
- Serial Data In (SDI)
- Slave Select ($\overline{SS}$)

The following figure shows the block diagram of the MSSP module when operating in SPI mode.

35.5.3 Slave Transmission

When the R/$\overline{\text{W}}$ bit of the incoming address byte is set and an address match occurs, the R/W bit is set. The received address is loaded into the SSPxBUF register, and an $\overline{\text{ACK}}$ pulse is sent by the slave on the ninth bit.

Following the $\overline{\text{ACK}}$, slave hardware clears the CKP bit and the SCL pin is held low (see Clock Stretching for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data.

The transmit data must be loaded into the SSPxBUF register which also loads the SSPSR register. Then the SCL pin should be released by setting the CKP bit. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time.

The $\overline{\text{ACK}}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. This $\overline{\text{ACK}}$ value is copied to the ACKSTAT bit. If ACKSTAT is set (not $\overline{\text{ACK}}$), then the data transfer is complete. In this case, when the not $\overline{\text{ACK}}$ is latched by the slave, the slave goes idle and waits for another occurrence of the Start bit. If the SDA line was low ($\overline{\text{ACK}}$), the next transmit data must be loaded into the SSPxBUF register. Again, the SCL pin must be released by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared by software and the SSPxSTAT register is used to determine the status of the byte. The SSPxIF bit is set on the falling edge of the ninth clock pulse.

35.5.3.1 Slave Mode Bus Collision

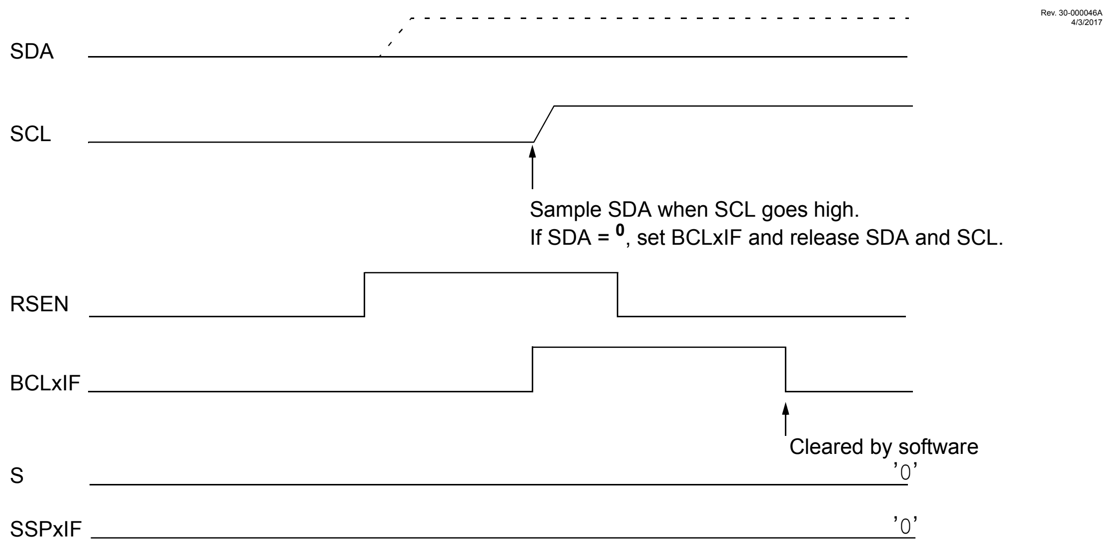
A slave receives a Read request and begins shifting data out on the SDA line. If a bus collision is detected and the SBCDE bit is set, the BCLxIF bit of the PIRx register is set. Once a bus collision is detected, the slave goes idle and waits to be addressed again. User software can use the BCLxIF bit to handle a slave bus collision.

35.5.3.2 7-bit Transmission

A master device can transmit a read request to a slave, and then clock data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 35-18 can be used as a reference to this list.

1. Master sends a Start condition on SDA and SCL.
2. S bit is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{\text{W}}$ bit set is received by the Slave setting SSPxIF bit.
4. Slave hardware generates an $\overline{\text{ACK}}$ and sets SSPxIF.
5. SSPxIF bit is cleared by user.
6. Software reads the received address from SSPxBUF, clearing BF.
7. R/$\overline{\text{W}}$ is set so CKP was automatically cleared after the $\overline{\text{ACK}}$.
8. The slave software loads the transmit data into SSPxBUF.
9. CKP bit is set releasing SCL, allowing the master to clock the data out of the slave.
10. SSPxIF is set after the $\overline{\text{ACK}}$ response from the master is loaded into the ACKSTAT register.
11. SSPxIF bit is cleared.
12. The slave software checks the ACKSTAT bit to see if the master wants to clock out more data.

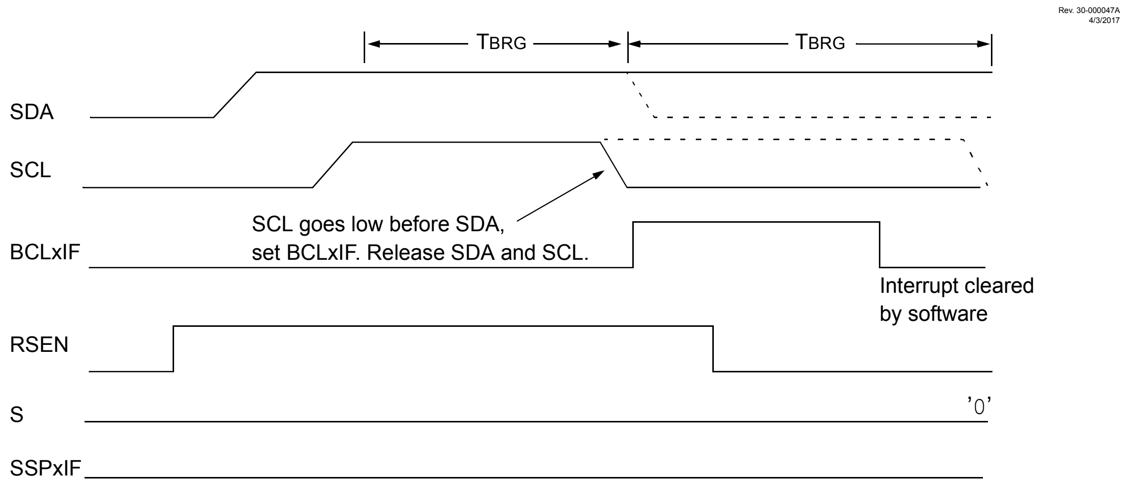
Figure 35-36. Bus Collision During a Repeated Start Condition (Case 1)


Rev. 30-000046A
4/3/2017
SDA
SCL
Sample SDA when SCL goes high.
If SDA = 0, set BCLxIF and release SDA and SCL.
RSEN
BCLxIF
Cleared by software
S
'0'
SSPxIF
'0'

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 35-37.

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 35-37. Bus Collision During Repeated Start Condition (Case 2)



35.6.13.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

1. After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out (Case 1).
2. After the SCL pin is deasserted, SCL is sampled low before SDA goes high (Case 2).

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPxADD

36.6.2 TXxSTA

Name: TXxSTA
Address: 0x011E

Transmit Status and Control Register

Bit	7	6	5	4	3	2	1	0
	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
Access	R/W	R/W	R/W	R/W	R/W	R/W	RO	R/W
Reset	0	0	0	0	0	0	1	0

Bit 7 – CSRC Clock Source Select bit

Value	Condition	Description
1	SYNC=1	Master mode (clock generated internally from BRG)
0	SYNC=1	Slave mode (clock from external source)
X	SYNC=0	Don't care

Bit 6 – TX9 9-bit Transmit Enable bit

Value	Description
1	Selects 9-bit transmission
0	Selects 8-bit transmission

Bit 5 – TXEN Transmit Enable bit
Enables transmitter[(1)]

Value	Description
1	Transmit enabled
0	Transmit disabled

Bit 4 – SYNC EUSART Mode Select bit

Value	Description
1	Synchronous mode
0	Asynchronous mode

Bit 3 – SENDB Send Break Character bit

Value	Condition	Description
1	SYNC=0	Send Sync Break on next transmission (cleared by hardware upon completion)
0	SYNC=0	Sync Break transmission disabled or completed
X	SYNC=1	Don't care

Bit 2 – BRGH High Baud Rate Select bit

Value	Condition	Description
1	SYNC=0	High speed, if BRG16 = 1, baud rate is baudclk/4; else baudclk/16
0	SYNC=0	Low speed
X	SYNC=1	Don't care

Figure 37-15. Capture Mode, Repeat Acquisition Timing Diagram

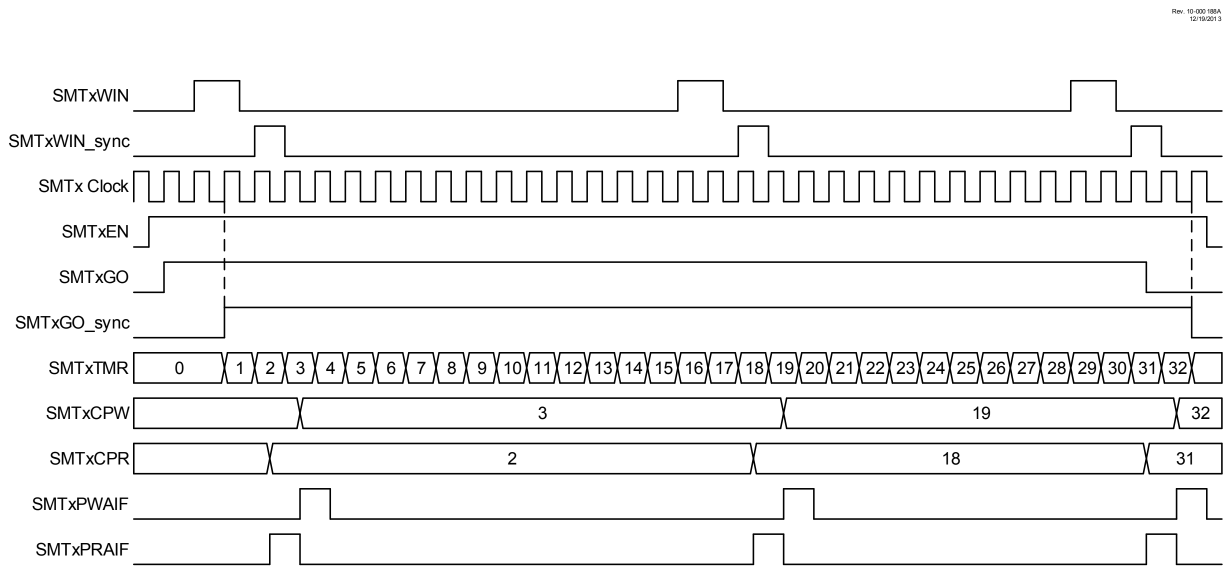
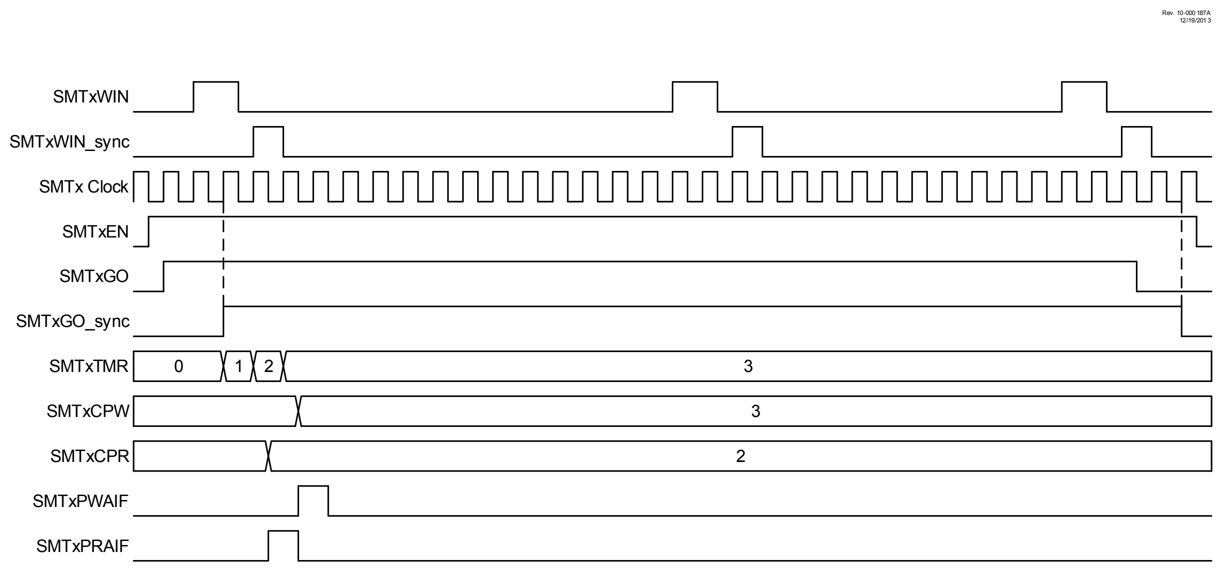


Figure 37-16. Capture Mode, Single Acquisition Timing Diagram



37.1.6.9 Counter Mode

This mode increments the timer on each pulse of the signal input. This mode is asynchronous to the SMT clock and uses the signal input as a time source. The SMTxCPW register will be updated with the current SMTxTMR value on the falling edge of the window input. See figure below.

Offset	Name	Bit Pos.								
0x0D8B	INTCON	7:0	GIE	PEIE						INTEDG
0x0D8C ... 0x0DFF	Reserved									
0x0E00	INDF0	7:0	INDF0[7:0]							
0x0E01	INDF1	7:0	INDF1[7:0]							
0x0E02	PCL	7:0	PCL[7:0]							
0x0E03	STATUS	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x0E04	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x0E06	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x0E08	BSR	7:0			BSR[5:0]					
0x0E09	WREG	7:0	WREG[7:0]							
0x0E0A	PCLATH	7:0		PCLATH[6:0]						
0x0E0B	INTCON	7:0	GIE	PEIE						INTEDG
0x0E0C ... 0x0E7F	Reserved									
0x0E80	INDF0	7:0	INDF0[7:0]							
0x0E81	INDF1	7:0	INDF1[7:0]							
0x0E82	PCL	7:0	PCL[7:0]							
0x0E83	STATUS	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x0E84	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x0E86	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x0E88	BSR	7:0			BSR[5:0]					
0x0E89	WREG	7:0	WREG[7:0]							
0x0E8A	PCLATH	7:0		PCLATH[6:0]						
0x0E8B	INTCON	7:0	GIE	PEIE						INTEDG
0x0E8C ... 0x0EFF	Reserved									
0x0F00	INDF0	7:0	INDF0[7:0]							
0x0F01	INDF1	7:0	INDF1[7:0]							
0x0F02	PCL	7:0	PCL[7:0]							
0x0F03	STATUS	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x0F04	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x0F06	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x0F08	BSR	7:0			BSR[5:0]					
0x0F09	WREG	7:0	WREG[7:0]							
0x0F0A	PCLATH	7:0		PCLATH[6:0]						
0x0F0B	INTCON	7:0	GIE	PEIE						INTEDG

42.4.9 Comparator Specifications

Table 42-15.

Standard Operating Conditions (unless otherwise stated) V_{DD} = 3.0V, T_A = 25°C							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ. †**	**Max.**	**Units**	**Conditions**
CM01	V_{IOFF}	Input Offset Voltage	—	±30	—	mV	$V_{ICM} = V_{DD}/2$
CM02	V_{ICM}	Input Common Mode Range	GND	—	V_{DD}	V	
CM03	CMRR	Common Mode Input Rejection Ratio	—	50	—	dB	
CM04	V_{HYST}	Comparator Hysteresis	15	25	35	mV	
CM05	T_{RESP}(1)	Response Time, Rising Edge	—	300	600	ns	
		Response Time, Falling Edge	—	220	500	ns	
CM06*	T_{MCV2VO}(2)	Mode Change to Valid Output	—	—	10	ns	

* - These parameters are characterized but not tested.

† Data in “Typ” column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note:
1. Response time measured with one comparator input at $V_{DD}/2$, while the other input transitions from V_{SS} to V_{DD}.
2. A mode change includes changing any of the control register values, including module enable.

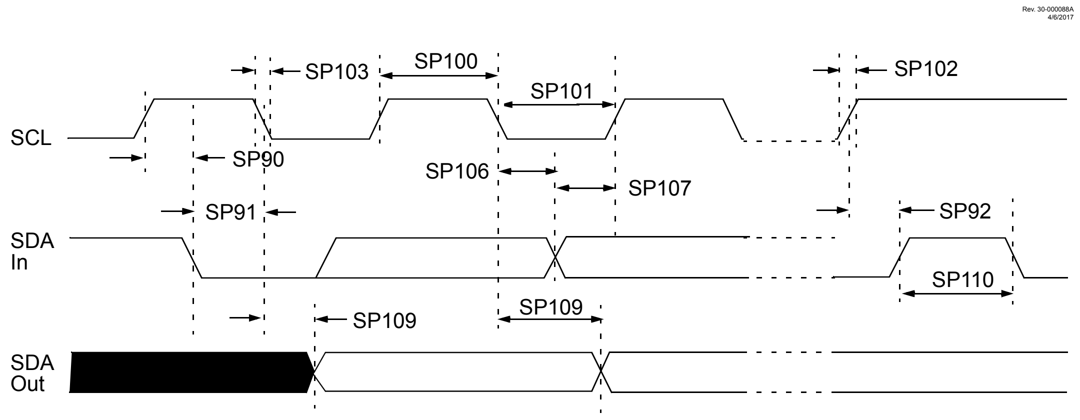
42.4.10 5-Bit DAC Specifications

Table 42-16.

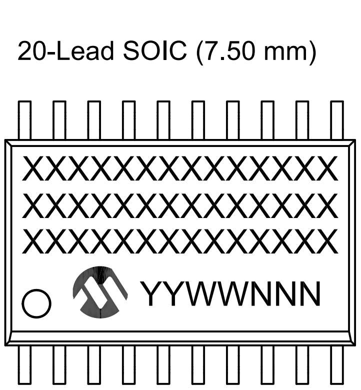
Standard Operating Conditions (unless otherwise stated) V_{DD} = 3.0V, T_A = 25°C							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ. †**	**Max.**	**Units**	**Conditions**
DSB01	V_{LSB}	Step Size	—	$(V_{DACREF}+ - V_{DACREF}-)/32$	—	V	
DSB02	V_{ACC}	Absolute Accuracy	—	—	±0.5	LSb	

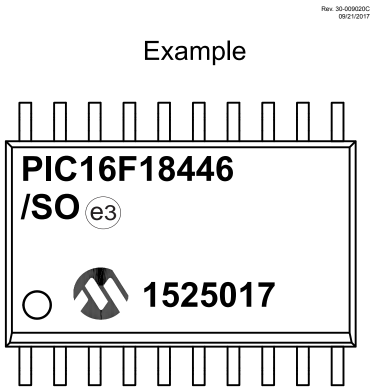
Standard Operating Conditions (unless otherwise stated)						
Param. No.	Sym.	Characteristic	Min.	Max.	Units	Conditions
if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line TR max. + $T_{SU:DAT}$ = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.						

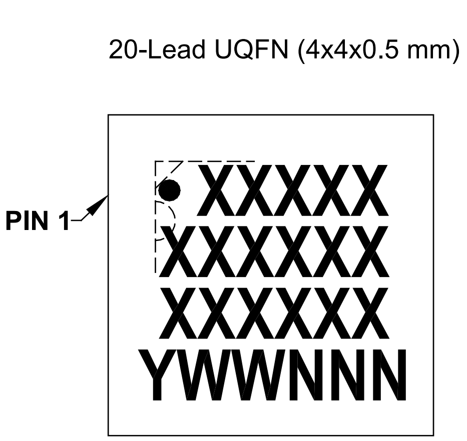
Figure 42-22. I²C Bus Data Timing

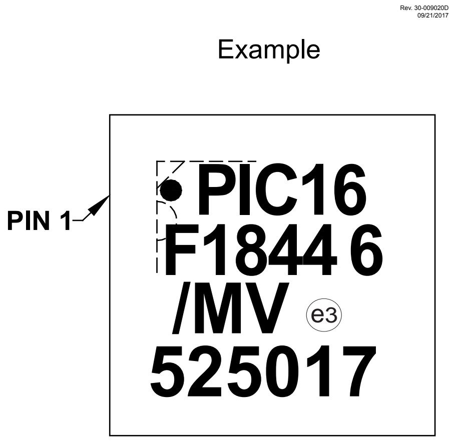


Note: Refer to Figure 42-4 for load conditions.









44.1 Package Details

The following sections give the technical details of the packages.