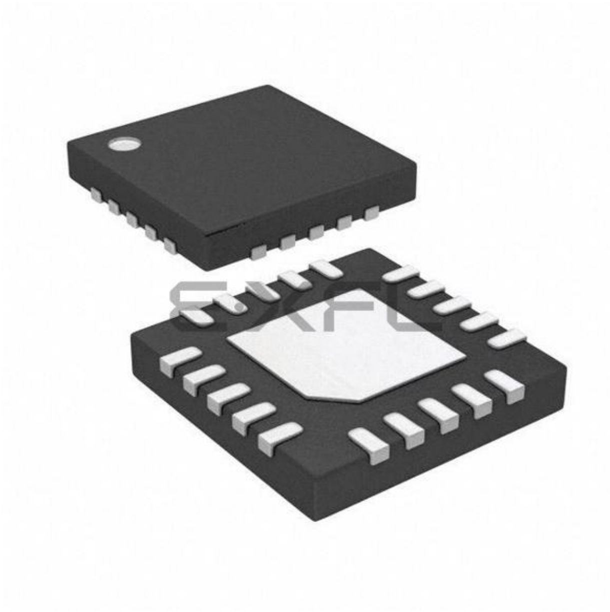
Microchip Technology - PIC16LF18345-I/GZ Datasheet



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	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 17x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-UFQFN Exposed Pad
Supplier Device Package	20-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf18345-i-gz

TABLE 1-2: PIC16(L)F18325 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
OUT(2)	C1	—	CMOS	Comparator C1 output.
	C2	—	CMOS	Comparator C2 output.
	NCO1	—	CMOS	Numerically Controlled Oscillator output.
	DSM	—	CMOS	Digital Signal Modulator output.
	TMR0	—	CMOS	TMR0 clock output.
	CCP1	—	CMOS	Capture/Compare/PWM 1 output.
	CCP2	—	CMOS	Capture/Compare/PWM 2 output.
	CCP3	—	CMOS	Capture/Compare/PWM 3 output.
	CCP4	—	CMOS	Capture/Compare/PWM 4 output.
	PWM5	—	CMOS	Pulse-Width Modulator 5 output.
	PWM6	—	CMOS	Pulse-Width Modulator 6 output.
	CWG1A	—	CMOS	Complementary Waveform Generator 1 output A.
	CWG2A	—	CMOS	Complementary Waveform Generator 2 output A.
	CWG1B	—	CMOS	Complementary Waveform Generator 1 output B.
	CWG2B	—	CMOS	Complementary Waveform Generator 2 output B.
	CWG1C	—	CMOS	Complementary Waveform Generator 1 output C.
	CWG2C	—	CMOS	Complementary Waveform Generator 2 output C.
	CWG1D	—	CMOS	Complementary Waveform Generator 1 output D.
	CWG2D	—	CMOS	Complementary Waveform Generator 2 output D.
	SDA1(3)	I^{2}C	OD	I^{2}C data output.
	SDA2(3)	I^{2}C	OD	I^{2}C data output.
	SCL1(3)	I^{2}C	OD	I^{2}C clock output.
	SCL2(3)	I^{2}C	OD	I^{2}C clock output.
	SDO1	—	CMOS	SPI1 data output.
	SD02	—	CMOS	SPI2 data output.
	SCK1	—	CMOS	SPI1 clock output.
	SCK2	—	CMOS	SPI2 clock output.
	TX/CK	—	CMOS	Asynchronous TX data/synchronous clock output.
	DT	—	CMOS	EUSART synchronous data output.
	CLC1OUT	—	CMOS	Configurable Logic Cell 1 source output.
	CLC2OUT	—	CMOS	Configurable Logic Cell 2 source output.
	CLC3OUT	—	CMOS	Configurable Logic Cell 3 source output.
	CLC4OUT	—	CMOS	Configurable Logic Cell 4 source output.
	CLKR	—	CMOS	Clock Reference output.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^{2}C = Schmitt Trigger input with I^{2}C
HV = High Voltage XTAL = Crystal levels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 13-1.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 13-2.

3: These I^{2}C functions are bidirectional. The output pin selections must be the same as the input pin selections.

TABLE 4-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31

Address	Name	PIC16(L)F18325	PIC16(L)F18345	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 0													
CPU CORE REGISTERS; see Table 4-2 **for specifics**													
00Ch	PORTA			—	—	RA5	RA4	RA3	RA2	RA1	RA0	`--xx xxxx`	`--uu uuuu`
00Dh	PORTB	X	—	Unimplemented								—	—
		—	X	RB7	RB6	RB5	RB4	—	—	—	—	`xxxx ----`	`uuuu ----`
00Eh	PORTC	X	—	—	—	RC5	RC4	RC3	RC2	RC1	RC0	`--xx xxxx`	`--uu uuuu`
		—	X	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	`xxxx xxxx`	`uuuu uuuu`
00Fh	—	—		Unimplemented								—	—
010h	PIR0			—	—	TMR0IF	IOCIF	—	—	—	INTF	`--00 ---0`	`--00 ---0`
011h	PIR1			TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	`0000 0000`	`0000 0000`
012h	PIR2			TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF	`0000 0000`	`0000 0000`
013h	PIR3			OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	`0000 0000`	`0000 0000`
014h	PIR4			CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	`0000 0000`	`0000 0000`
015h	TMR0L			TMR0L<7:0>								`xxxx xxxx`	`xxxx xxxx`
016h	TMR0H			TMR0H<7:0>								`1111 1111`	`1111 1111`
017h	T0CON0			T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>				`0-00 0000`	`0-00 0000`
018h	T0CON1			T0CS<2:0>			T0ASYNC	T0CKPS<3:0>				`0000 0000`	`0000 0000`
019h	TMR1L			TMR1L<7:0>								`xxxx xxxx`	`uuuu uuuu`
01Ah	TMR1H			TMR1H<7:0>								`xxxx xxxx`	`uuuu uuuu`
01Bh	T1CON			TMR1CS<1:0>		T1CKPS<1:0>		T1SOSC	$\overline{\text{T1SYNC}}$	—	TMR1ON	`0000 00-0`	`uuuu uu-u`
01Ch	T1GCON			TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ $\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		`0000 0x00`	`uuuu uxuu`
01Dh	TMR2			TMR2<7:0>								`0000 0000`	`0000 0000`
01Eh	PR2			PR2<7:0>								`1111 1111`	`1111 1111`
01Fh	T2CON			—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		`-000 0000`	`-000 0000`

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

TABLE 4-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	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 1													
CPU CORE REGISTERS; see Table 4-2 for specifics													
08Ch	TRISA			—	—	TRISA5	TRISA4	—	TRISA2	TRISA1	TRISA0	`--11 -111`	`--11 -111`
08Dh	TRISB	X	—	Unimplemented								—	—
		—	X	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	`1111 ----`	`1111 ----`
08Eh	TRISC	X	—	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	`--11 1111`	`--11 1111`
		—	X	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	`1111 1111`	`1111 1111`
08Fh	—	—		Unimplemented								—	—
090h	PIE0			—	—	TMR0IE	IOCIE	—	—	—	INTE	`--00 ---0`	`--00 ---0`
091h	PIE1			TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	`0000 0000`	`0000 0000`
092h	PIE2			TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE	`0000 0000`	`0000 0000`
093h	PIE3			OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	`0000 0000`	`0000 0000`
094h	PIE4			CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	`0000 0000`	`0000 0000`
095h	—	—		Unimplemented								—	—
096h	—	—		Unimplemented								—	—
097h	WDTCON			—	—	WDTPS<4:0>					SWDTEN	`--01 0110`	`--01 0110`
098h	—	—		Unimplemented								—	—
099h	—	—		Unimplemented								—	—
09Ah	—	—		Unimplemented								—	—
09Bh	ADRESL			ADRESL<7:0>								`xxxx xxxx`	`uuuu uuuu`
09Ch	ADRESH			ADRESH<7:0>								`xxxx xxxx`	`uuuu uuuu`
09Dh	ADCON0			CHS<5:0>						GO/$\overline{\text{DONE}}$	ADON	`0000 0000`	`0000 0000`
09Eh	ADCON1			ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		`0000 -000`	`0000 -000`
09Fh	ADACT			—	—	—	ADACT<4:0>					`---0 0000`	`---0 0000`

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

Note 1: Only on PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

5.3 Code Protection

Code protection allows the device to be protected from unauthorized access. Program memory protection and data memory are controlled independently. Internal access to the program memory is unaffected by any code protection setting.

5.3.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the $\overline{\text{CP}}$ bit in Configuration Words. When $\overline{\text{CP}}$ = `0`, external reads and writes of program memory are inhibited and a read will return all '`0`'s. The CPU can continue to read program memory, regardless of the protection bit settings. Self-write writing the program memory is dependent upon the write protection setting. See **Section 5.4 "Write Protection"** for more information.

5.3.2 DATA MEMORY PROTECTION

The entire data EEPROM is protected from external reads and writes by the $\overline{\text{CPD}}$ bit in the Configuration Words. When $\overline{\text{CPD}}$ = `0`, external reads and writes of EEPROM memory are inhibited and a read will return all '`0`'s. The CPU can continue to read and write EEPROM memory, regardless of the protection bit settings.

5.4 Write Protection

Write protection allows the device to be protected from unintended self-writes. Applications, such as boot loader software, can be protected while allowing other regions of the program memory to be modified.

The WRT<1:0> bits in Configuration Words define the size of the program memory block that is protected.

5.5 User ID

Four memory locations (8000h-8003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are readable and writable during normal execution. See **Section 11.4.7 "NVMREG EEPROM, User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations. For more information on checksum calculation, see the *"PIC16(L)F183XX Memory Programming Specification"* (DS40001738).

5.6 Device ID and Revision ID

The 14-bit device ID word is located at 8006h and the 14-bit revision ID is located at 8005h. These locations are read-only and cannot be erased or modified. See **Section 11.4.7 "NVMREG EEPROM, User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations.

Development tools, such as device programmers and debuggers, may be used to read the Device ID and Revision ID.

TABLE 7-3: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON1	—	NOSC<2:0>			NDIV<3:0>				90
OSCCON2	—	COSC<2:0>			CDIV<3:0>				90
OSCCON3	CWSHOLD	SOSCPWR	SOSCBE	ORDY	NOSCR	—	—	—	91
OSCSTAT1	EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLLR	92
OSCEN	EXTOEN	HFOEN	—	LFOEN	SOSCEN	ADOEN	—	—	93
OSCFRQ	—	—	—	—	HFFRQ<3:0>				94
OSCTUNE	—	—	HFTUN<5:0>						95

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.

TABLE 7-4: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	FCMEN	—	CSWEN	—	—	$\overline{\text{CLKOUTEN}}$	64
	7:0	—	RSTOSC2	RSTOSC1	RSTOSC0	—	FEXTOSC2	FEXTOSC1	FEXTOSC0	

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.

TABLE 9-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	101
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	103
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE	104
PIE3	OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	105
PIE4	CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	106
PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	107
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	108
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF	109
PIR3	OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	110
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	111
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	174
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	174
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	175
IOCBP(1)	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	175
IOCBN(1)	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	176
IOCBF(1)	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	176
IOCCP	IOCCP7(1)	IOCCP6(1)	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	177
IOCCN	IOCCN7(1)	IOCCN6(1)	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	177
IOCCF	IOCCF7(1)	IOCCF6(1)	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	178
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	30
VREGCON(2)	—	—	—	—	—	—	VREGPM	—	117
CPUDOZE	IDLEN	DOZEN	ROI	DOE	—	DOZE<2:0>			117
WDTCON	—	—	WDTPS<4:0>					SWDTEN	121

Legend: — = unimplemented location, read as '0'. Shaded cells are not used in Power-Down mode.

Note 1: PIC16(L)F18345 only.

2: PIC16F18325/18345 only.

12.0 I/O PORTS

TABLE 12-1: PORT AVAILABILITY PER DEVICE

Device	PORTA	PORTB	PORTC
PIC16(L)F18325	●		●
PIC16(L)F18345	●	●	●

Each port has ten standard registers for its operation. These registers are:

- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)
- TRISx registers (data direction)
- ANSELx registers (analog select)
- WPUx registers (weak pull-up)
- INLVLx (input level control)
- SLRCONx registers (slew rate)
- ODCONx registers (open-drain)

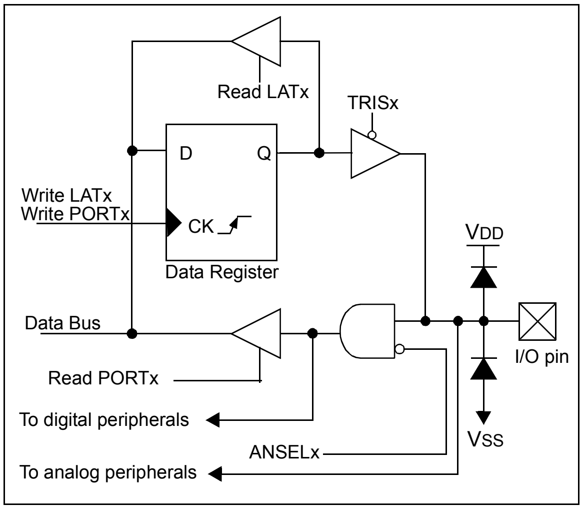
Most port pins share functions with device peripherals, both analog and digital. In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output; however, the pin can still be read.

The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSEL bit is set, the digital input buffer associated with that bit is disabled. Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 12-1.

FIGURE 12-1: GENERIC I/O PORT OPERATION



12.1 I/O Priorities

Each pin defaults to the PORT data latch after Reset. Other functions are selected with the peripheral pin select logic. See **Section 13.0 “Peripheral Pin Select (PPS) Module”** for more information.

Analog input functions, such as ADC and comparator inputs, are not shown in the peripheral pin select lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELx register. Digital output functions may continue to control the pin when it is in Analog mode.

Analog outputs, when enabled, take priority over the digital outputs and force the digital output driver to the high-impedance state.

REGISTER 12-11: LATB: PORTB DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0
LATB7	LATB6	LATB5	LATB4	—	—	—	—
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 7-4 **LATB<7:4>**: RB<5:4> Output Latch Value bits[(1)]

bit 3-0 **Unimplemented:** Read as ‘0’

Note 1: Writes to LATB are equivalent with writes to the corresponding PORTB register.Reads from LATB register return register values, not I/O pin values.

REGISTER 12-12: ANSELB: PORTB ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—
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 7-4 **ANSB<7:4>**: Analog Select between Analog or Digital Function
0 = Digital I/O. Pin is assigned to port or digital special function.
1 = Analog input. Pin is assigned as analog input[(1)]. Digital input buffer disabled.

bit 3-0 **Unimplemented:** Read as ‘0’

Note 1: Setting ANSB[n] = 1 disables the digital input circuitry. Weak pull-ups, if available, are unaffected. The corresponding TRIS bit must be set to Input mode by the user to allow external control of the voltage on the pin.

18.2 Comparator Control

Each comparator has two control registers: CMxCON0 and CMxCON1.

The CMxCON0 register (see Register 18-1) contains Control and Status bits for the following:

- Enable
- Output
- Output polarity
- Hysteresis enable
- Timer1 output synchronization

The CMxCON1 register (see Register 18-2) contains Control bits for the following:

- Interrupt on positive/negative edge enables
- Positive input channel selection
- Negative input channel selection

18.2.1 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.

18.2.2 COMPARATOR OUTPUT

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.

The comparator output can also be routed to an external pin through the RxyPPS register (Register 13-2). The corresponding TRIS bit must be clear to enable the pin as an output.

Note 1: The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched.

18.2.3 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 18-2 shows the output state versus input conditions, including polarity control.

TABLE 18-2: 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

18.3 Comparator Hysteresis

A selectable amount of separation voltage can be added to the input pins of each comparator to provide a hysteresis function to the overall operation. Hysteresis is enabled by setting the CxHYS bit of the CMxCON0 register.

See Comparator Specifications in Table 35-14 for more information.

18.4 Timer1 Gate Operation

The output resulting from a comparator operation can be used as a source for gate control of Timer1. See **Section 27.5 "Timer1 Gate"** for more information. This feature is useful for timing the duration or interval of an analog event.

It is recommended that the comparator output be synchronized to Timer1. This ensures that Timer1 does not increment while a change in the comparator is occurring.

18.4.1 COMPARATOR OUTPUT SYNCHRONIZATION

The output from a comparator can be synchronized with Timer1 by setting the CxSYNC bit of the CMxCON0 register.

Once enabled, the comparator output is latched on the falling edge of the Timer1 source clock. This allows the timer/counter to synchronize with the CxOUT bit so that the software sees no ambiguity due to timing. See the Comparator Block Diagram (Figure 18-2) and the Timer1 Block Diagram (Figure 27-1) for more information.

24.0 5-BIT DIGITAL-TO-ANALOG CONVERTER (DAC1) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 32 selectable output levels.

The input of the DAC can be connected to:

- External VREF pins
- VDD supply voltage
- FVR (Fixed Voltage Reference)

The output of the DAC can be configured to supply a reference voltage to the following:

- Comparator positive input
- ADC input channel
- DAC1OUT pin

The Digital-to-Analog Converter (DAC) is enabled by setting the DAC1EN bit of the DAC1CON0 register.

24.1 Output Voltage Selection

The DAC has 32 voltage level ranges. The 32 levels are set with the DAC1R<4:0> bits of the DAC1CON1 register.

The DAC output voltage is determined by Equation 24-1:

EQUATION 24-1: DAC OUTPUT VOLTAGE

$$V_{OUT} = \left((V_{SOURCE+} - V_{SOURCE-}) \times \frac{DAC1R\langle 4:0 \rangle}{2^5} \right) + (V_{SOURCE-})$$

$$V_{SOURCE+} = V_{DD} \text{ or } V_{REF+} \text{ or } FVR$$

$$V_{SOURCE-} = V_{SS} \text{ or } V_{REF-}$$

24.2 Ratiometric Output Level

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

The value of the individual resistors within the ladder can be found in Table 35-15.

24.3 DAC Voltage Reference Output

The DAC voltage can be output to the DAC1OUT pin by setting the DAC1OE bit of the DAC1CON0 register. Selecting the DAC reference voltage for output on the DAC1OUT pin automatically overrides the digital output buffer and digital input threshold detector functions, it disables the weak pull-up and the constant-current drive function of that pin. Reading the DAC1OUT pin when it has been configured for DAC reference voltage output will always return a ‘0’.

Due to the limited current drive capability, a buffer must be used on the DAC voltage reference output for external connections to the DAC1OUT pin. Figure 24-2 shows an example buffering technique.

29.2.2 TIMER1/3/5 MODE RESOURCE

Timer1/3/5 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

See **Section 27.0 “Timer1/3/5 Module with Gate Control”** for more information on configuring Timer1/3/5.

> **Note:** Clocking Timer1/3/5 from the system clock (FOSC) should not be used in Capture mode. In order for Capture mode to recognize the trigger event on the CCPx pin, Timer1/3/5 must be clocked from the instruction clock (FOSC/4) or from an external clock source.

29.2.3 SOFTWARE INTERRUPT MODE

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit of the PIE4 register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF interrupt flag bit of the PIR4 register following any change in Operating mode.

29.2.4 CCP PRESCALER

There are four prescaler settings specified by the CCPxMODE<3:0> bits of the CCPxCON register. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. Any Reset will clear the prescaler counter.

Switching from one capture prescaler to another does not clear the prescaler and may generate a false interrupt. To avoid this unexpected operation, turn the module off by clearing the CCPxCON register before changing the prescaler. Example 29-1 demonstrates the code to perform this function.

EXAMPLE 29-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
BANKSEL CCPxCON     ;Set Bank bits to point
                    ;to CCPxCON
CLRF    CCPxCON     ;Turn CCP module off
MOVLW   NEW_CAPT_PS ;Load the W reg with
                    ;the new prescaler
                    ;move value and CCP ON
MOVWF   CCPxCON     ;Load CCPxCON with this
                    ;value
```

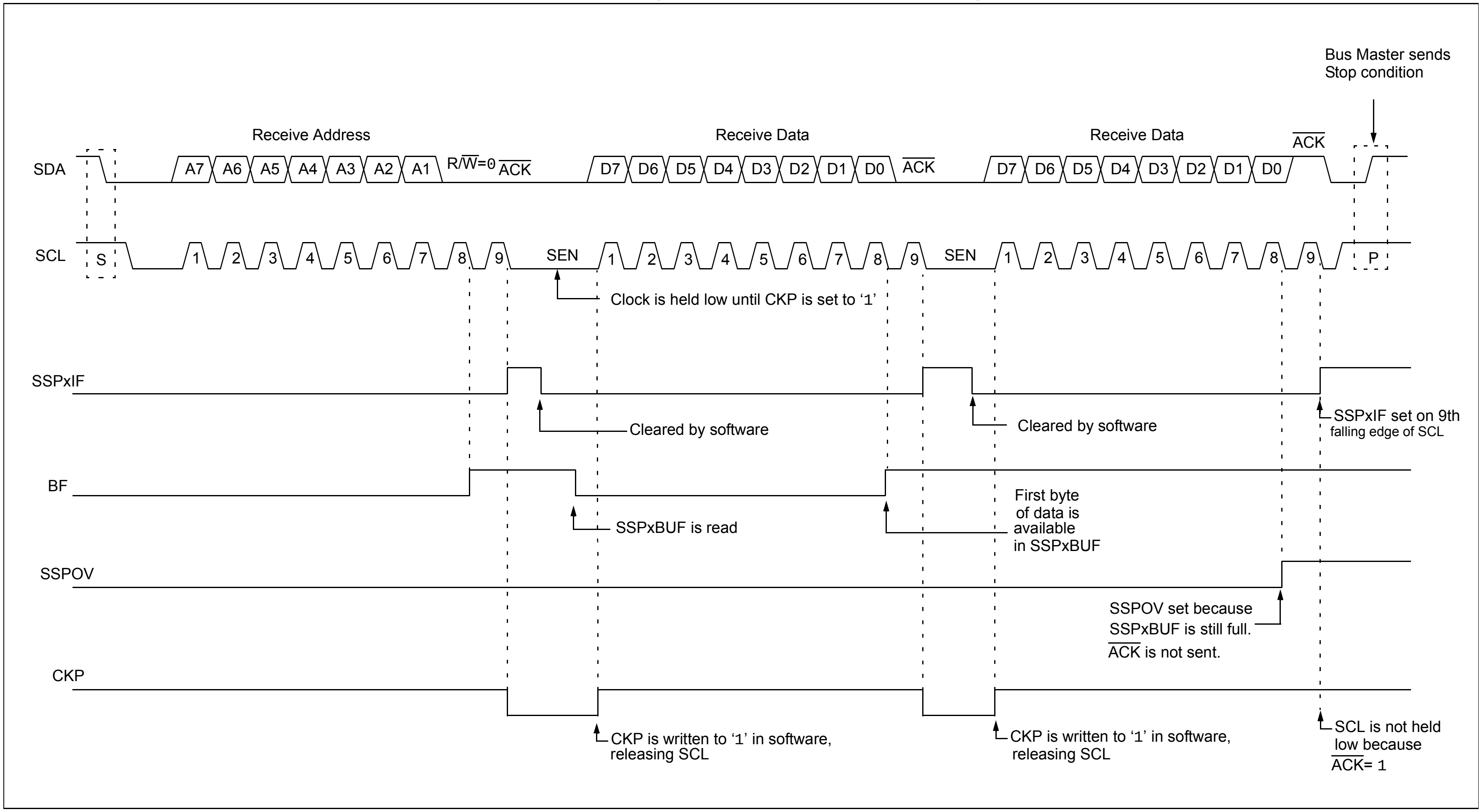
29.2.5 CAPTURE DURING SLEEP

Capture mode depends upon the Timer1/3/5 module for proper operation. There are two options for driving the Timer1/3/5 module in Capture mode. It can be driven by the instruction clock (FOSC/4), or by an external clock source.

When Timer1/3/5 is clocked by FOSC/4, Timer1/3/5 will not increment during Sleep. When the device wakes from Sleep, Timer1/3/5 will continue from its previous state.

Capture mode will operate during Sleep when Timer1/3/5 is clocked by an external clock source.

FIGURE 30-15: I^2C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)


Bus Master sends Stop condition
Receive Address
Receive Data
Receive Data
SDA
A7 A6 A5 A4 A3 A2 A1
R/W=0
ACK
D7 D6 D5 D4 D3 D2 D1 D0
ACK
D7 D6 D5 D4 D3 D2 D1 D0
ACK
SCL
S
1 2 3 4 5 6 7 8 9
SEN
1 2 3 4 5 6 7 8 9
SEN
1 2 3 4 5 6 7 8 9
P
Clock is held low until CKP is set to '1'
SSPxIF
Cleared by software
Cleared by software
SSPxIF set on 9th falling edge of SCL
BF
SSPxBUF is read
First byte of data is available in SSPxBUF
SSPOV
SSPOV set because SSPxBUF is still full. ACK is not sent.
CKP
CKP is written to '1' in software, releasing SCL
CKP is written to '1' in software, releasing SCL
SCL is not held low because ACK= 1

30.6.5 I^{2}C MASTER MODE REPEATED START CONDITION TIMING

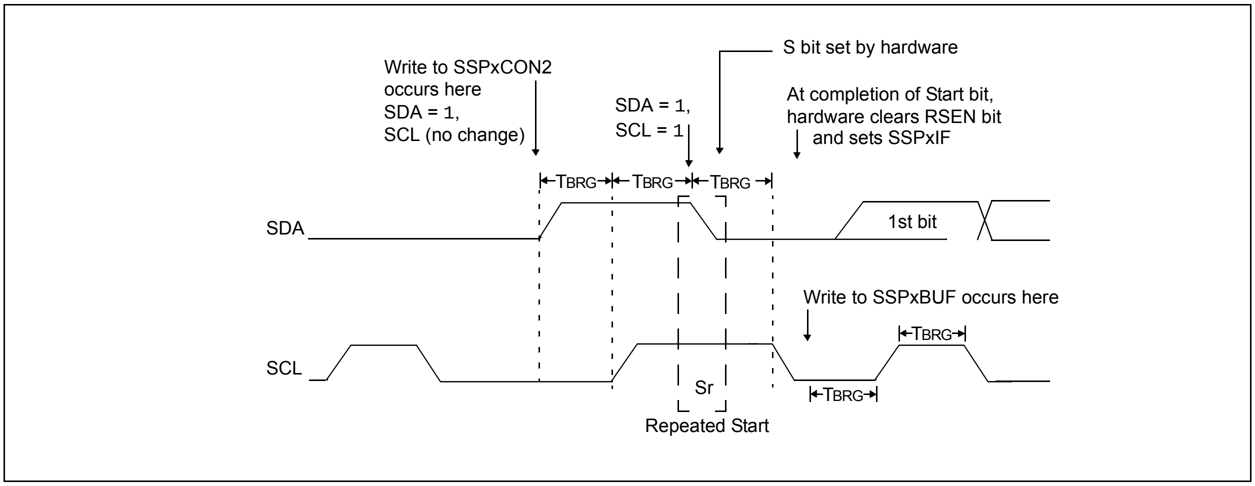
A Repeated Start condition (Figure 30-27) occurs when the RSEN bit of the SSPxCON2 register is programmed high and the master state machine is no longer active. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. SCL is asserted low. Following this, the RSEN bit of the SSPxCON2 register will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit of the SSPxSTAT register will be set. The SSPxIF bit will not be set until the Baud Rate Generator has timed out.

Note 1: If RSEN is programmed while any other event is in progress, it will not take effect.

2: A bus collision during the Repeated Start condition occurs if:

- SDA is sampled low when SCL goes from low-to-high.
- SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data ‘1’.

FIGURE 30-27: REPEATED START CONDITION WAVEFORM


S bit set by hardware
Write to SSPxCON2 occurs here
SDA = 1,
SCL (no change)
SDA = 1,
SCL = 1
At completion of Start bit, hardware clears RSEN bit and sets SSPxIF
TBRG
TBRG
TBRG
SDA
1st bit
Write to SSPxBUF occurs here
TBRG
SCL
Sr
TBRG
Repeated Start

REGISTER 30-4: SSPxCON3: SSP CONTROL REGISTER 3

R-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
ACKTIM[3]	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
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 **ACKTIM:** Acknowledge Time Status bit (I^2C mode only)[3]

1 = Indicates the I^2C bus is in an Acknowledge sequence, set on eighth falling edge of SCL clock
0 = Not an Acknowledge sequence, cleared on ninth rising edge of SCL clock

bit 6 **PCIE**: Stop Condition Interrupt Enable bit (I^2C mode only)

1 = Enable interrupt on detection of Stop condition
0 = Stop detection interrupts are disabled[2]

bit 5 **SCIE**: Start Condition Interrupt Enable bit (I^2C mode only)

1 = Enable interrupt on detection of Start or Restart conditions
0 = Start detection interrupts are disabled[2]

bit 4 **BOEN:** Buffer Overwrite Enable bit

In SPI Slave mode:[1]
1 = SSPBUF updates every time that a new data byte is shifted in ignoring the BF bit
0 = If new byte is received with BF bit of the SSPSTAT register already set, SSPOV bit of the SSPCON1 register is set, and the buffer is not updated

In I^2C Master mode and SPI Master mode:
This bit is ignored.

In I^2C Slave mode:
1 = SSPBUF is updated and $\overline{\text{ACK}}$ is generated for a received address/data byte, ignoring the state of the SSPOV bit only if the BF bit = 0.
0 = SSPBUF is only updated when SSPOV is clear

bit 3 **SDAHT:** SDA Hold Time Selection bit (I^2C mode only)

1 = Minimum of 300 ns hold time on SDA after the falling edge of SCL
0 = Minimum of 100 ns hold time on SDA after the falling edge of SCL

bit 2 **SBCDE:** Slave Mode Bus Collision Detect Enable bit (I^2C Slave mode only)

If, on the rising edge of SCL, SDA is sampled low when the module is outputting a high state, the BCL1IF bit of the PIR1 register is set, and bus goes idle

1 = Enable slave bus collision interrupts
0 = Slave bus collision interrupts are disabled

bit 1 **AHEN:** Address Hold Enable bit (I^2C Slave mode only)

1 = Following the eighth falling edge of SCL for a matching received address byte; CKP bit of the SSPCON1 register will be cleared and the SCL will be held low.
0 = Address holding is disabled

bit 0 **DHEN:** Data Hold Enable bit (I^2C Slave mode only)

1 = Following the eighth falling edge of SCL for a received data byte; slave hardware clears the CKP bit of the SSPCON1 register and SCL is held low.
0 = Data holding is disabled

Note 1: For daisy-chained SPI operation; allows the user to ignore all but the last received byte. SSPOV is still set when a new byte is received and BF = 1, but hardware continues to write the most recent byte to SSPBUF.

2: This bit has no effect in Slave modes that Start and Stop condition detection is explicitly listed as enabled.

3: The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is set.

REGISTER 30-5: SSPxMSK: SSP MASK REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SSPxMSK<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 7-1 **SSPxMSK<7:1>:** Mask bits
1 = The received address bit n is compared to SSPADD<n> to detect I^{2}C address match
0 = The received address bit n is not used to detect I^{2}C address match

bit 0 **SSPxMSK<0>:** Mask bit for I^{2}C Slave mode, 10-bit Address
I^{2}C Slave mode, 10-bit address (SSPM<3:0> = 0111 or 1111):
1 = The received address bit 0 is compared to SSPADD<0> to detect I^{2}C address match
0 = The received address bit 0 is not used to detect I^{2}C address match
I^{2}C Slave mode, 7-bit address:
MSK0 bit is ignored.

REGISTER 30-6: SSPxADD: MSSP ADDRESS AND BAUD RATE REGISTER (I^{2}C MODE)

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
SSPxADD<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	

Master mode:

bit 7-0 **SSPxADD<7:0>:** Baud Rate Clock Divider bits
SCL pin clock period = ((ADD<7:0> + 1) *4)/FOSC

10-bit Slave mode – Most Significant Address Byte:

bit 7-3 **Not used:** Unused for Most Significant Address Byte. Bit state of this register is a “don’t care”. Bit pattern sent by master is fixed by I^{2}C specification and must be equal to ‘11110’. However, those bits are compared by hardware and are not affected by the value in this register.

bit 2-1 **SSPxADD<2:1>:** Two Most Significant bits of 10-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a “don’t care”.

10-bit Slave mode – Least Significant Address Byte:

bit 7-0 **SSPxADD<7:0>:** Eight Least Significant bits of 10-bit address

7-bit Slave mode:

bit 7-1 **SSPxADD<7:1>:** 7-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a “don’t care”.

31.1.1.5 TSR Status

The TRMT bit of the TX1STA register indicates the status of the TSR register. This is a read-only bit. The TRMT bit is set when the TSR register is empty and is cleared when a character is transferred to the TSR register from the TX1REG. The TRMT bit remains clear until all bits have been shifted out of the TSR register. No interrupt logic is tied to this bit, so the user has to poll this bit to determine the TSR status.

Note:	The TSR register is not mapped in data memory, so it is not available to the user.

31.1.1.6 Transmitting 9-bit Characters

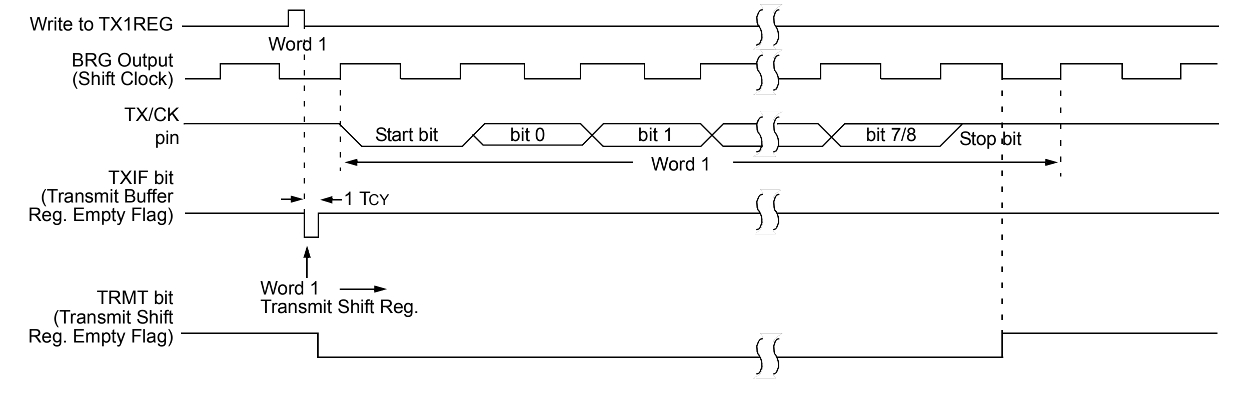
The EUSART1 supports 9-bit character transmissions. When the TX9 bit of the TX1STA register is set, the EUSART1 will shift nine bits out for each character transmitted. The TX9D bit of the TX1STA register is the ninth, and Most Significant data bit. When transmitting 9-bit data, the TX9D data bit must be written before writing the eight Least Significant bits into the TX1REG. All nine bits of data will be transferred to the TSR shift register immediately after the TX1REG is written.

A special 9-bit Address mode is available for use with multiple receivers. See **Section 31.1.2.7 "Address Detection"** for more information on the Address mode.

31.1.1.7 Asynchronous Transmission Set-up

1. Initialize the SP1BRGH, SP1BRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 31.3 "EUSART1 Baud Rate Generator (BRG)"**).
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If 9-bit transmission is desired, set the TX9 control bit. A set ninth data bit will indicate that the eight Least Significant data bits are an address when the receiver is set for address detection.
4. Set SCKP bit if inverted transmit is desired.
5. Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
6. If interrupts are desired, set the TXIE interrupt enable bit of the PIE1 register. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
7. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
8. Load 8-bit data into the TX1REG register. This will start the transmission.

FIGURE 31-3: ASYNCHRONOUS TRANSMISSION



37.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
  - MPLAB Xpress IDE Software
  - Microchip Code Configurator (MCC)
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/
    MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

37.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

37.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

37.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradeable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

37.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

37.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

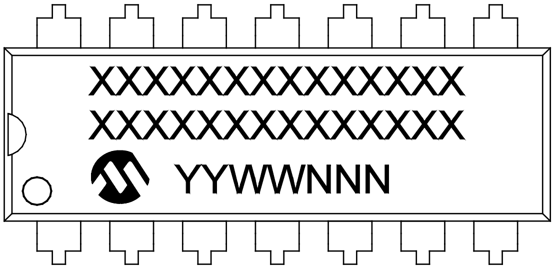
37.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

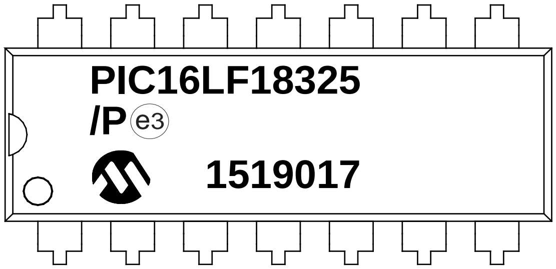
38.0 PACKAGING INFORMATION

38.1 Package Marking Information

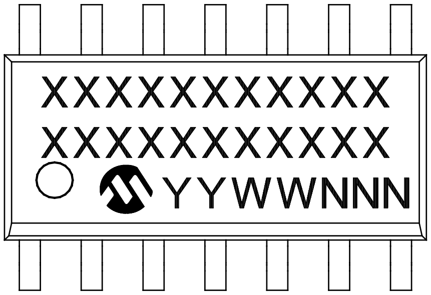
14-Lead PDIP (300 mil)



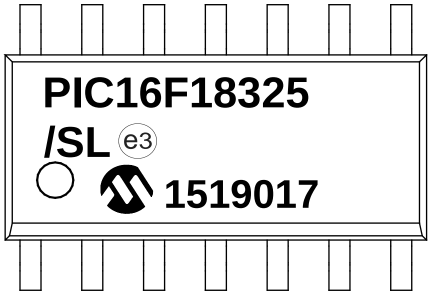
Example



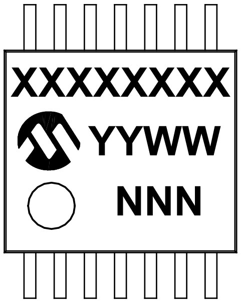
14-Lead SOIC (3.90 mm)



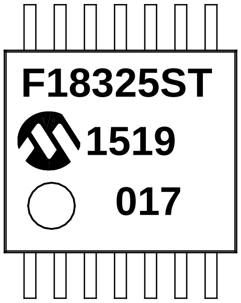
Example



14-Lead TSSOP (4.4 mm)



Example

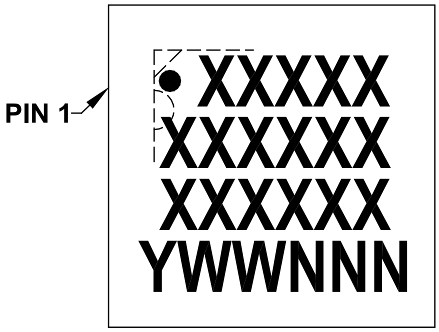


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	e3	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

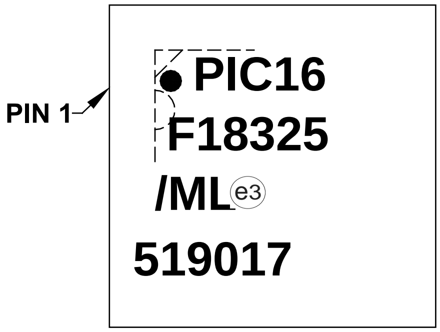
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

Package Marking Information (Continued)

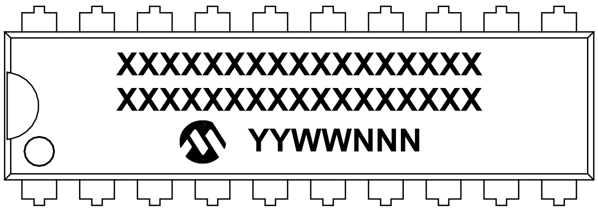
16-Lead UQFN (4x4x0.5mm)



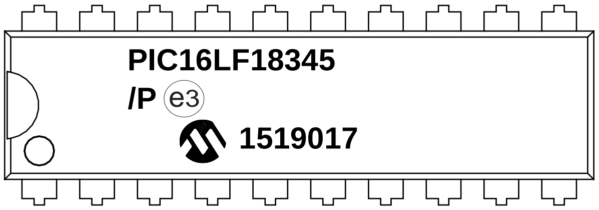
Example



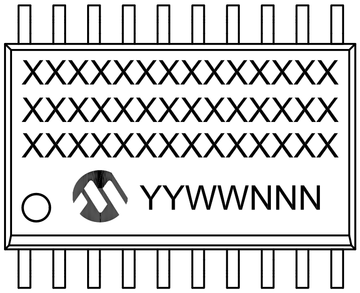
20-Lead PDIP (300 mil)



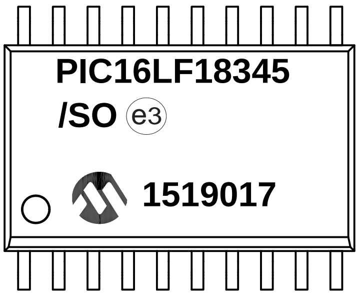
Example



20-Lead SOIC (7.50 mm)



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	e3	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.