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	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 12x10b; D/A 1x8b
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/pic16f1618-i-so

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Website at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Website; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our website at **www.microchip.com** to receive the most current information on all of our products.

TABLE 1-2: PIC16(L)F1614 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/AN0/C1IN+/DAC1OUT1/ ICSPDAT	RA0	TTL/ST	CMOS/OD	General purpose I/O.
	AN0	AN	—	ADC Channel input.
	C1IN+	AN	—	Comparator positive input.
	DAC1OUT1	—	AN	Digital-to-Analog Converter output.
	ICSPDAT	ST	CMOS	ICSP™ Data I/O.
RA1/AN1/VREF+/C1IN0-/C2IN0-/ ICSPCLK	RA1	TTL/ST	CMOS/OD	General purpose I/O.
	AN1	AN	—	ADC Channel input.
	VREF+	AN	—	Voltage Reference input.
	C1IN0-	AN	—	Comparator negative input.
	C2IN0-	AN	CMOS/OD	Comparator negative input.
	ICSPCLK	ST	—	ICSP Programming Clock.
RA2/AN2/T0CKI(1)/CWG1IN(1) ZCD1IN/INT	RA2	TTL/ST	CMOS/OD	General purpose I/O.
	AN2	AN	—	ADC Channel input.
	T0CKI	TTL/ST	—	Timer0 clock input.
	CWG1IN	TTL/ST	—	CWG complementary input.
	ZCD1IN	AN	—	Zero-Cross Detect input.
	INT	TTL/ST	—	External interrupt.
RA3/VPP/T6IN(1)/SMTWIN2(1)/ $\overline{\text{MCLR}}$	RA3	TTL/ST	—	General purpose input with IOC and WPU.
	VPP	HV	—	Programming voltage.
	T6IN	TTL/ST	—	Timer6 input.
	SMTWIN2	TTL/ST	—	SMT2 window input.
	$\overline{\text{MCLR}}$	TTL/ST	—	Master Clear with internal pull-up.
RA4/AN3/T1G(1)/SMTSIG1(1)/ CLKOUT	RA4	TTL/ST	CMOS/OD	General purpose I/O.
	AN3	AN	—	ADC Channel input.
	T1G	TTL/ST	—	Timer1 Gate input.
	SMTSIG1	TTL/ST	—	SMT1 signal input.
	CLKOUT	—	CMOS	FOSC/4 output.
RA5/CLKIN/T1CKI(1)/T2IN(1)/ SMTWIN1(1)	RA5	TTL/ST	CMOS/OD	General purpose I/O.
	CLKIN	CMOS	—	External clock input (EC mode).
	T1CKI	TTL/ST	—	Timer1 clock input.
	T2IN	TTL/ST	—	Timer2 input.
	SMTWIN1	TTL/ST	—	SMT1 window input.
RC0/AN4/C2IN+/T5CKI(1)/ SCK(1)	RC0	TTL/ST	CMOS/OD	General purpose I/O.
	AN4	AN	—	ADC Channel input.
	C2IN+	AN	—	Comparator positive input.
	T5CKI	TTL/ST	—	Timer5 clock input.
	SCK	ST	CMOS	SPI clock.

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^2C = Schmitt Trigger input with I^2C levels
HV = High Voltage XTAL = Crystal

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

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 13-1.

TABLE 1-3: PIC16(L)F1618 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RB6/SCK(1, 3)	RB6	TTL/ST	CMOS/OD	General purpose I/O.
	SCK	ST	CMOS	SPI clock.
RB7/CK(1)	RB7	TTL/ST	CMOS/OD	General purpose I/O.
	CK	ST	CMOS	USART synchronous clock.
RC0/AN4/C2IN+/T5CKI(1)	RC0	TTL/ST	CMOS/OD	General purpose I/O.
	AN4	AN	—	ADC Channel input.
	C2IN+	AN	—	Comparator positive input.
	T5CKI	TTL/ST	—	Timer5 clock input.
RC1/AN5/C1IN1-/C2IN1-/ T4IN(1)/CLCIN(2)/SMTSIG2(1)	RC1	TTL/ST	CMOS/OD	General purpose I/O.
	AN5	AN	—	ADC Channel input.
	C1IN1-	AN	—	Comparator negative input.
	C2IN1-	AN	—	Comparator negative input.
	T4IN	TTL/ST	—	Timer4 input.
	CLCIN2	ST	—	Configurable Logic Cell source input.
	SMTSIG2	TTL/ST	—	SMT2 signal input.
RC2/AN6/C1IN2-/C2IN2-	RC2	TTL/ST	CMOS/OD	General purpose I/O.
	AN6	AN	—	ADC Channel input.
	C1IN2-	AN	—	Comparator negative input.
	C2IN2-	AN	—	Comparator negative input.
RC3/AN7/C1IN3-/C2IN3-/T5G(1)/ CCP2(1)/CLCIN0(1)/ATCC(1)	RC3	TTL/ST	—	General purpose input with IOC and WPU.
	AN7	AN	—	ADC Channel input.
	C1IN3-	AN	—	Comparator negative input.
	C2IN3-	AN	—	Comparator negative input.
	T5G	ST	—	Timer5 Gate input.
	CCP2	ST	CMOS	Capture/Compare/PWM2.
	CLCIN0	ST	—	Configurable Logic Cell source input.
	ATCC	ST	—	Angular Timer Capture/Compare input.
RC4/T3G(1)/CLCIN1(1)/HIC4	RC4	TTL/ST	CMOS/OD	General purpose I/O.
	T3G	ST	—	Timer3 Gate input.
	CLCIN1	ST	—	Configurable Logic Cell source input.
	HIC4	TTL	CMOS	High Current I/O.
RC5/T3CKI(1)/CCP2(1)/ATIN(1)/ HIC5	RC5	TTL/ST	CMOS/OD	General purpose I/O.
	T3CKI	TTL/ST	—	Timer3 clock input.
	CCP2	TTL/ST	CMOS/OD	Capture/Compare/PWM2.
	ATIN	TTL/ST	—	Angular Timer clock input.
	HIC5	TTL	CMOS	High Current I/O.

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^2C = Schmitt Trigger input with I^2C levels; HV = High Voltage; XTAL = Crystal

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

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 13-1.

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

TABLE 3-5: PIC16(L)F1614/8 MEMORY MAP, BANK 16-23

	BANK 16	BANK 17	BANK 18	BANK 19	BANK 20	BANK 21	BANK 22	BANK 23
800h	Core Registers (Table 3-1)	880h Core Registers (Table 3-1)	900h Core Registers (Table 3-1)	980h Core Registers (Table 3-1)	A00h Core Registers (Table 3-1)	A80h Core Registers (Table 3-1)	B00h Core Registers (Table 3-1)	B80h Core Registers (Table 3-1)
80Bh		88Bh	90Bh	98Bh	A0Bh	A8Bh	B0Bh	B8Bh
80Ch	AT1RESL	88Ch AT1CLK	90Ch Unimplemented Read as '0'	98Ch Unimplemented Read as '0'	A0Ch Unimplemented Read as '0'	A8Ch Unimplemented Read as '0'	B0Ch Unimplemented Read as '0'	B8Ch Unimplemented Read as '0'
80Dh	AT1RESH	88Dh AT1SIG						
80Eh	AT1MISSL	88Eh AT1CSEL1						
80Fh	AT1MISSH	88Fh AT1CC1L						
810h	AT1PERL	890h AT1CC1H						
811h	AT1PERH	891h AT1CCON1						
812h	AT1PHSL	892h AT1CSEL2						
813h	AT1PHSH	893h AT1CC2L						
814h	AT1CON0	894h AT1CC2H						
815h	AT1CON1	895h AT1CCON2						
816h	AT1IR0	896h AT1CSEL2						
817h	AT1IE0	897h AT1CC3L						
818h	AT1IR1	898h AT1CC3H						
819h	AT1IE1	899h AT1CCON3						
81Ah	AT1STPTL	89Ah						
81Bh	AT1STPTH							
81Ch	AT1ERRL							
81Dh	AT1ERRH							
86Fh		8EFh	96Fh	9EFh	A6Fh	AEFh	B6Fh	BEFh
870h	Accesses 70h – 7Fh	8F0h Accesses 70h – 7Fh	970h Accesses 70h – 7Fh	9F0h Accesses 70h – 7Fh	A70h Accesses 70h – 7Fh	AF0h Accesses 70h – 7Fh	B70h Accesses 70h – 7Fh	BF0h Accesses 70h – 7Fh
87Fh		8FFh	97Fh	9FFh	A7Fh	AFFh	B7Fh	BFFh

Legend: = Unimplemented data memory locations, read as '0'.

TABLE 3-14: 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
Banks 29											
E8Ch to E8Fh	—	Unimplemented								—	—
E90h	RA0PPS	—	—	—	RA0PPS<4:0>					---0 0000	---0 0000
E91h	RA1PPS	—	—	—	RA1PPS<4:0>					---0 0000	---0 0000
E92h	RA2PPS	—	—	—	RA2PPS<4:0>					---0 0000	---0 0000
E93h	—	Unimplemented								—	—
E94h	RA4PPS	—	—	—	RA4PPS<4:0>					---0 0000	---0 0000
E95h	RA5PPS	—	—	—	RA5PPS<4:0>					---0 0000	---0 0000
E96h to E9Bh	—	Unimplemented								—	—
E9Ch	RB4PPS(4)	—	—	—	RB4PPS<4:0>					---0 0000	---0 0000
E9Dh	RB5PPS(4)	—	—	—	RB5PPS<4:0>					---0 0000	---0 0000
E9Eh	RB6PPS(4)	—	—	—	RB6PPS<4:0>					---0 0000	---0 0000
E9Fh	RB7PPS(4)	—	—	—	RB7PPS<4:0>					---0 0000	---0 0000
EA0h	RC0PPS	—	—	—	RC0PPS<4:0>					---0 0000	---0 0000
EA1h	RC1PPS	—	—	—	RC1PPS<4:0>					---0 0000	---0 0000
EA2h	RC2PPS	—	—	—	RC2PPS<4:0>					---0 0000	---0 0000
EA3h	RC3PPS	—	—	—	RC3PPS<4:0>					---0 0000	---0 0000
EA4h	RC4PPS	—	—	—	RC4PPS<4:0>					---0 0000	---0 0000
EA5h	RC5PPS	—	—	—	RC5PPS<4:0>					---0 0000	---0 0000
EA6h	RC6PPS(4)	—	—	—	RC6PPS<4:0>					---0 0000	---0 0000
EA7h	RC7PPS(4)	—	—	—	RC7PPS<4:0>					---0 0000	---0 0000
EA8h to EEFh	—	Unimplemented								—	—

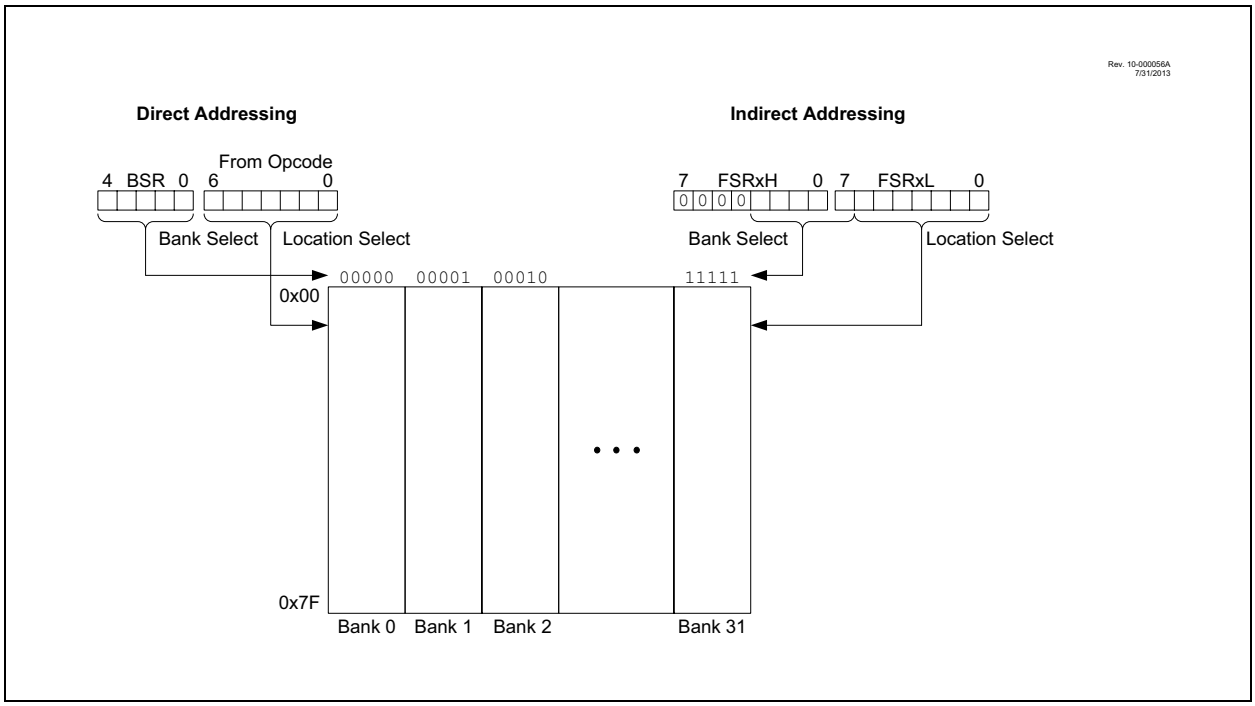
Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: PIC16F1614/8 only.
2: Unimplemented, read as '1'.
3: PIC16(L)F1614 only.
4: PIC16(L)F1618 only.

3.6.1 TRADITIONAL DATA MEMORY

The traditional data memory is a region from FSR address 0x000 to FSR address 0xFFF. The addresses correspond to the absolute addresses of all SFR, GPR and common registers.

FIGURE 3-9: TRADITIONAL DATA MEMORY MAP

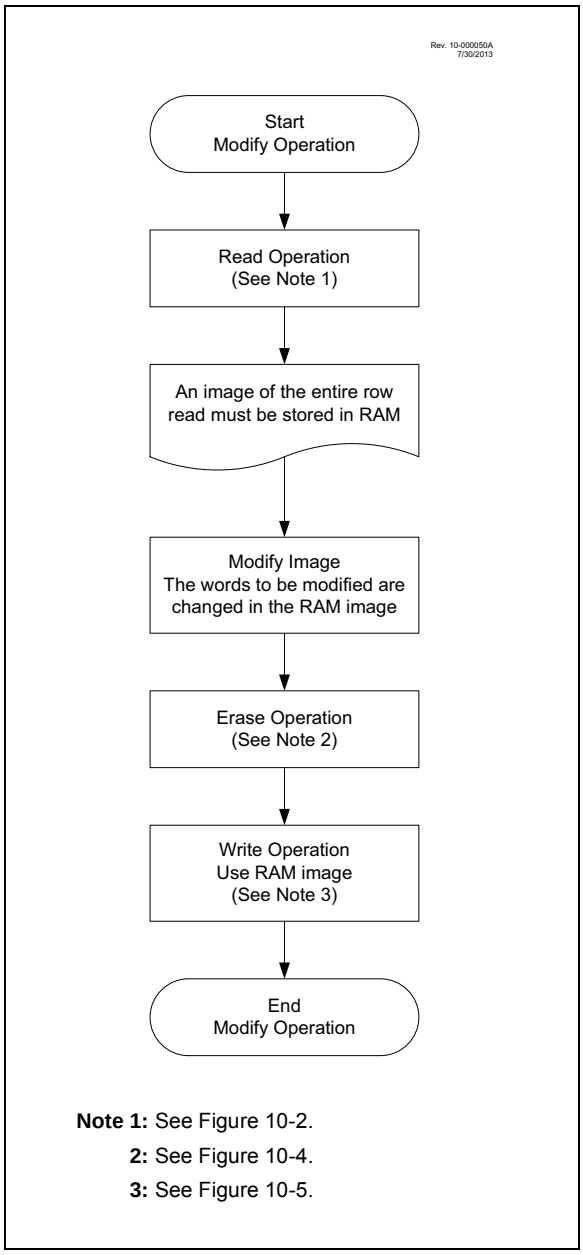


10.3 Modifying Flash Program Memory

When modifying existing data in a program memory row, and data within that row must be preserved, it must first be read and saved in a RAM image. Program memory is modified using the following steps:

1. Load the starting address of the row to be modified.
2. Read the existing data from the row into a RAM image.
3. Modify the RAM image to contain the new data to be written into program memory.
4. Load the starting address of the row to be rewritten.
5. Erase the program memory row.
6. Load the write latches with data from the RAM image.
7. Initiate a programming operation.

FIGURE 10-7: FLASH PROGRAM MEMORY MODIFY FLOWCHART



Note 1: See Figure 10-2.
2: See Figure 10-4.
3: See Figure 10-5.

11.11 Register Definitions: CRC and Scanner Control

REGISTER 11-1: CRCCON0: CRC CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R-0	R/W-0/0	U-0	U-0	R/W-0/0	R-0
EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL
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 **EN:** CRC Enable bit
1 = CRC module is released from Reset
0 = CRC is disabled and consumes no operating current

bit 6 **CRCGO:** CRC Start bit
1 = Start CRC serial shifter
0 = CRC serial shifter turned off

bit 5 **BUSY:** CRC Busy bit
1 = Shifting in progress or pending
0 = All valid bits in shifter have been shifted into accumulator and EMPTY = 1

bit 4 **ACCM:** Accumulator Mode bit
1 = Data is augmented with zeros
0 = Data is not augmented with zeros

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1 **SHIFTM:** Shift Mode bit
1 = Shift right (LSb)
0 = Shift left (MSb)

bit 0 **FULL:** Data Path Full Indicator bit
1 = CRCDATH/L registers are full
0 = CRCDATH/L registers have shifted their data into the shifter

REGISTER 11-2: CRCCON1: CRC CONTROL REGISTER 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	R/W-0/0
DLEN<3:0>				PLEN<3:0>			
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 7-4 **DLEN<3:0>:** Data Length bits
Denotes the length of the data word -1 (See Example 11-1)

bit 3-0 **PLEN<3:0>:** Polynomial Length bits
Denotes the length of the polynomial -1 (See Example 11-1)

TABLE 14-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-CHANGE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	152
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	180
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	180
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	180
IOCBF(2)	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	181
IOCBN(2)	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	181
IOCBP(2)	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	181
IOCCF	IOCCF7(2)	IOCCF6(2)	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	182
IOCCN	IOCCN7(2)	IOCCN6(2)	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	182
IOCCP	IOCCP7(2)	IOCCP6(2)	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	182
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	151
TRISC	TRISC7(2)	TRISC7(2)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	165

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by interrupt-on-change.

Note 1: Unimplemented, read as '1'.

2: PIC16(L)F1618 only.

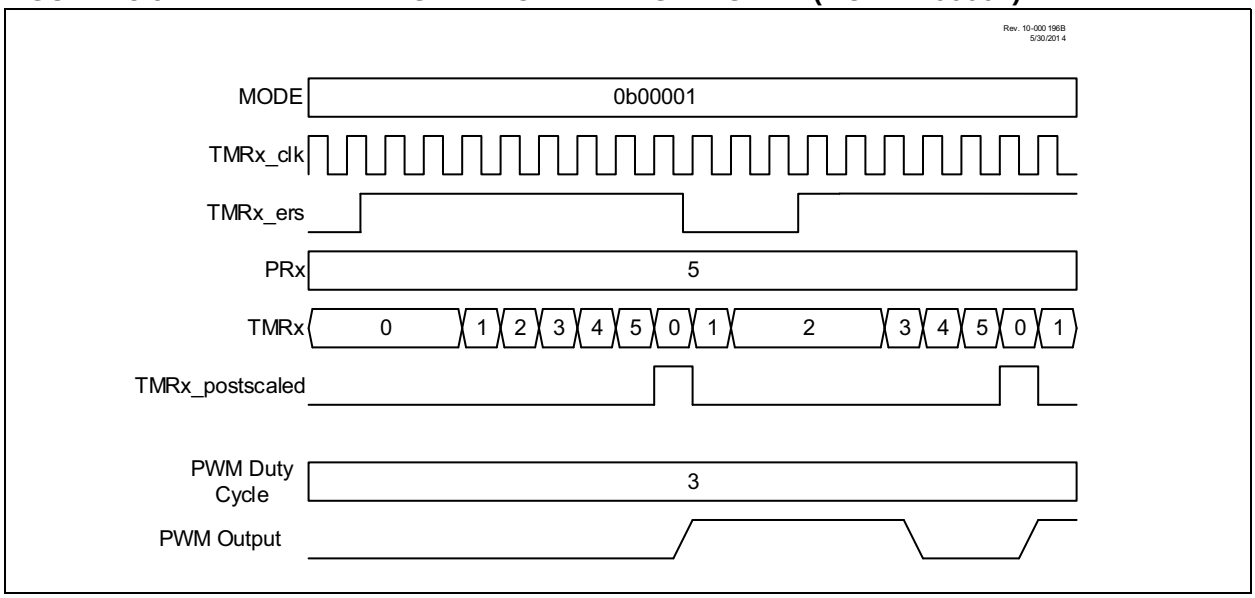
23.5.2 HARDWARE GATE MODE

The Hardware Gate modes operate the same as the Software Gate mode except the TMRx_ers external signal can also gate the timer. When used with the CCP the gating extends the PWM period. If the timer is stopped when the PWM output is high then the duty cycle is also extended.

When MODE<4:0> = 00001 then the timer is stopped when the external signal is high. When MODE<4:0> = 00010 then the timer is stopped when the external signal is low.

Figure 23-5 illustrates the Hardware Gating mode for MODE<4:0> = 00001 in which a high input level starts the counter.

FIGURE 23-5: HARDWARE GATE MODE TIMING DIAGRAM (MODE = 00001)

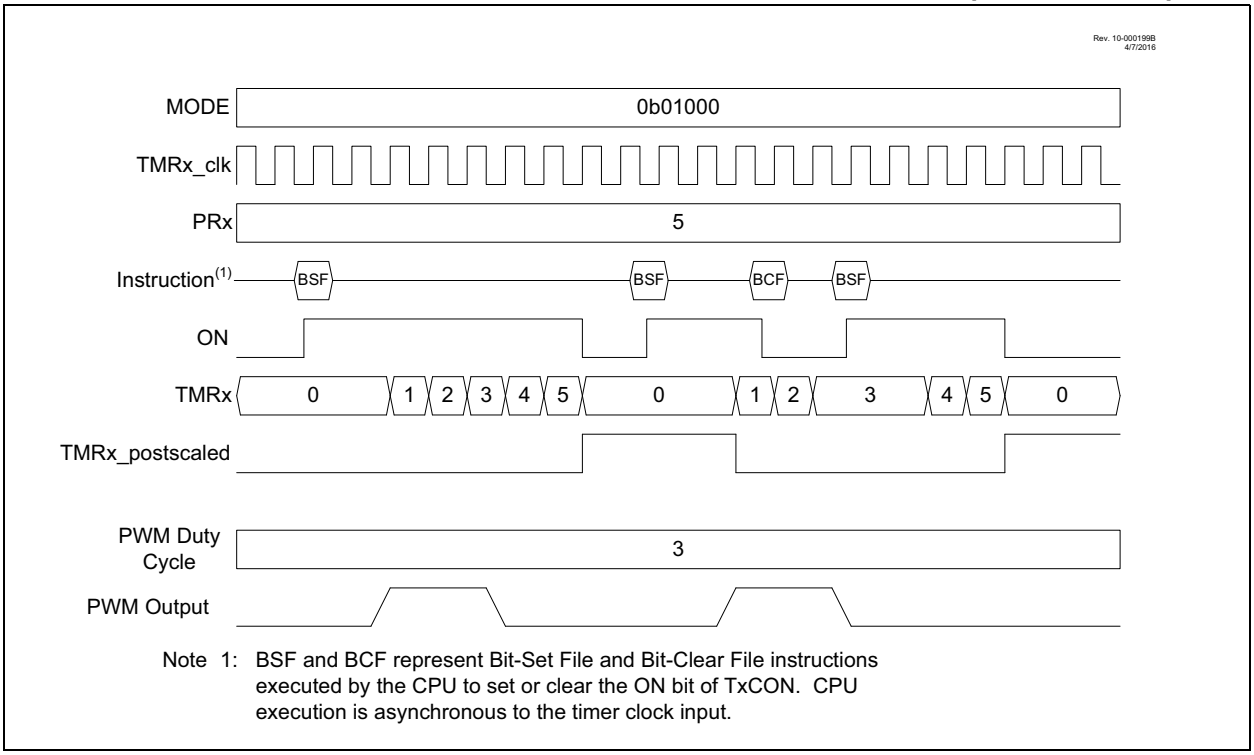


23.5.5 SOFTWARE START ONE-SHOT MODE

In One-Shot mode the timer resets and the ON bit is cleared when the timer value matches the PRx period value. The ON bit must be set by software to start another timer cycle. Setting MODE<4:0> = 01000 selects One-Shot mode which is illustrated in Figure 23-8. In the example, ON is controlled by BSF and BCF instructions. In the first case, a BSF instruction sets ON and the counter runs to completion and clears ON. In the second case, a BSF instruction starts the cycle, BCF/BSF instructions turn the counter off and on during the cycle, and then it runs to completion.

When One-Shot mode is used in conjunction with the CCP PWM operation the PWM pulse drive starts concurrent with setting the ON bit. Clearing the ON bit while the PWM drive is active will extend the PWM drive. The PWM drive will terminate when the timer value matches the CCPRx pulse width value. The PWM drive will remain off until software sets the ON bit to start another cycle. If software clears the ON bit after the CCPRx match but before the PRx match then the PWM drive will be extended by the length of time the ON bit remains cleared. Another timing cycle can only be initiated by setting the ON bit after it has been cleared by a PRx period count match.

FIGURE 23-8: SOFTWARE START ONE-SHOT MODE TIMING DIAGRAM (MODE = 01000)



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

24.6.6 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPxBUF register. This action will set the Buffer Full flag bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high. When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an $\overline{ACK}$ bit during the ninth bit time if an address match occurred, or if data was received properly. The status of $\overline{ACK}$ is written into the ACKSTAT bit on the rising edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPxIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPxBUF, leaving SCL low and SDA unchanged (Figure 24-28).

After the write to the SSPxBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/$\overline{W}$ bit are completed. On the falling edge of the eighth clock, the master will release the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the $\overline{ACK}$ bit is loaded into the ACKSTAT Status bit of the SSPxCON2 register. Following the falling edge of the ninth clock transmission of the address, the SSPxIF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPxBUF takes place, holding SCL low and allowing SDA to float.

24.6.6.1 BF Status Flag

In Transmit mode, the BF bit of the SSPxSTAT register is set when the CPU writes to SSPxBUF and is cleared when all eight bits are shifted out.

24.6.6.2 WCOL Status Flag

If the user writes the SSPxBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

WCOL must be cleared by software before the next transmission.

24.6.6.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit of the SSPxCON2 register is cleared when the slave has sent an Acknowledge ($\overline{ACK}$ = 0) and is set when the slave does not Acknowledge ($\overline{ACK}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

24.6.6.4 Typical Transmit Sequence:

1. The user generates a Start condition by setting the SEN bit of the SSPxCON2 register.
2. SSPxIF is set by hardware on completion of the Start.
3. SSPxIF is cleared by software.
4. The MSSP module will wait the required start time before any other operation takes place.
5. The user loads the SSPxBUF with the slave address to transmit.
6. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPxBUF is written to.
7. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPxCON2 register.
8. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
9. The user loads the SSPxBUF with eight bits of data.
10. Data is shifted out the SDA pin until all eight bits are transmitted.
11. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPxCON2 register.
12. Steps 8-11 are repeated for all transmitted data bytes.
13. The user generates a Stop or Restart condition by setting the PEN or RSEN bits of the SSPxCON2 register. Interrupt is generated once the Stop/Restart condition is complete.

24.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSPxCON2 register. When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 24-30).

24.6.8.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

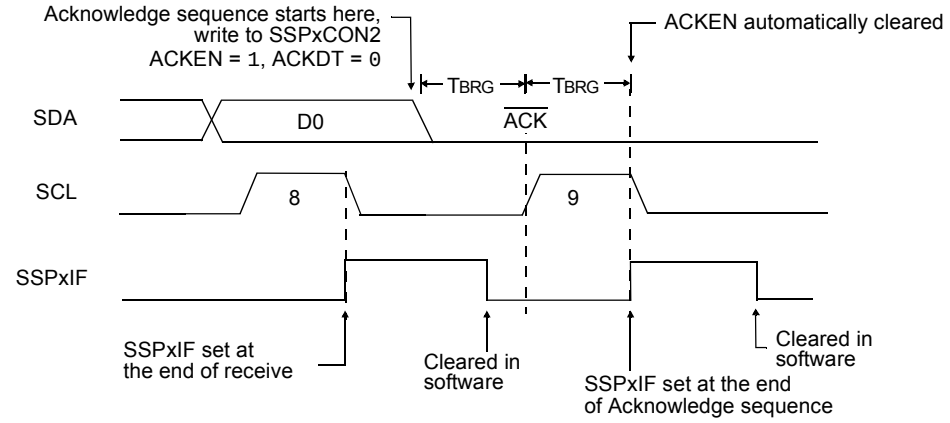
24.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN bit of the SSPxCON2 register. At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit of the SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 24-31).

24.6.9.1 WCOL Status Flag

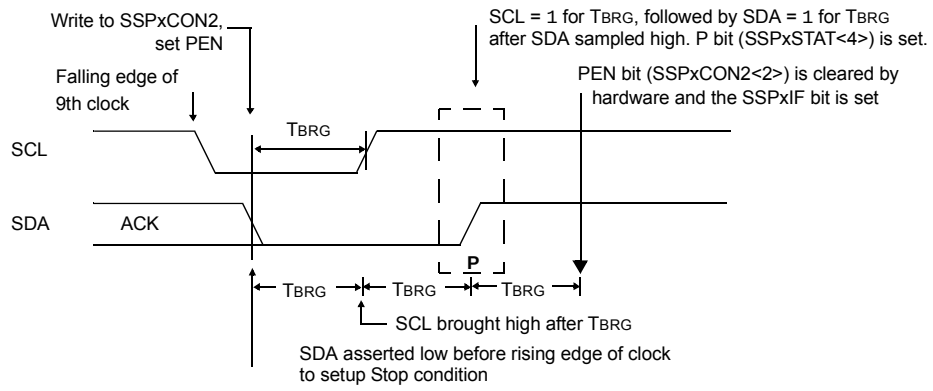
If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

FIGURE 24-30: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one Baud Rate Generator period.

FIGURE 24-31: STOP CONDITION RECEIVE OR TRANSMIT MODE



Note: TBRG = one Baud Rate Generator period.

27.2 Register Definitions: PWM Control

REGISTER 27-1: PWMxCON: PWM CONTROL REGISTER

R/W-0/0	U-0	R-0/0	R/W-0/0	U-0	U-0	U-0	U-0
PWMxEN	—	PWMxOUT	PWMxPOL	—	—	—	—
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 **PWMxEN:** PWM Module Enable bit
1 = PWM module is enabled
0 = PWM module is disabled

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **PWMxOUT:** PWM Module Output Value bit

bit 4 **PWMxPOL:** PWMx Output Polarity Select bit
1 = PWM output is active-low
0 = PWM output is active-high

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

REGISTER 27-2: PWMxDCH: PWM DUTY CYCLE HIGH BITS

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
PWMxDCH<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 **PWMxDCH<7:0>:** PWM Duty Cycle Most Significant bits
These bits are the MSbs of the PWM duty cycle. The two LSbs are found in the PWMxDCL register.

REGISTER 27-3: PWMxDCL: PWM DUTY CYCLE LOW BITS

R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0	U-0	U-0
PWMxDCL<7:6>		—	—	—	—	—	—
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	

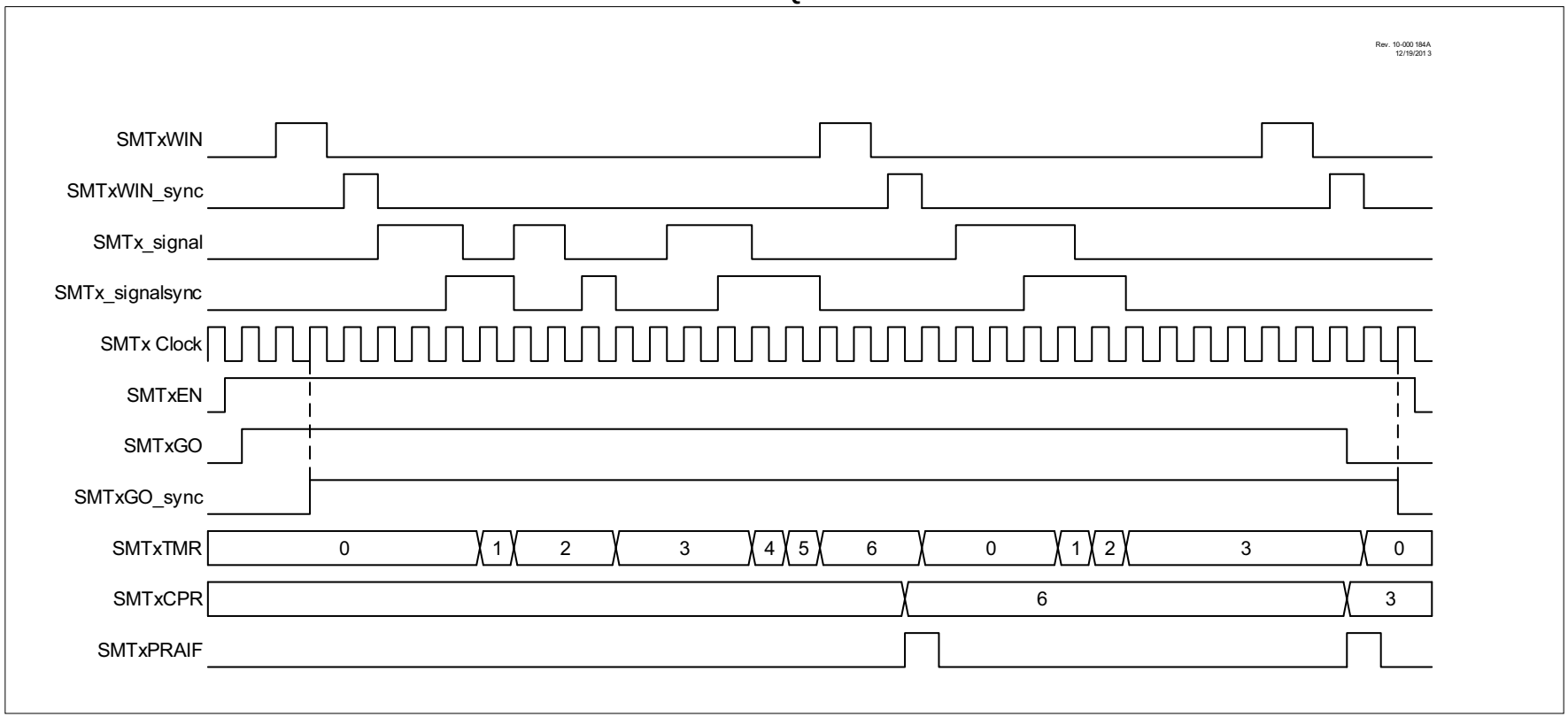
bit 7-6 **PWMxDCL<7:6>:** PWM Duty Cycle Least Significant bits
These bits are the LSbs of the PWM duty cycle. The MSbs are found in the PWMxDCH register.

bit 5-0 **Unimplemented:** Read as ‘0’

30.6.2 GATED TIMER MODE

Gated Timer mode uses the SMTSIGx input to control whether or not the SMTxTMR will increment. Upon a falling edge of the external signal, the SMTxCPW register will update to the current value of the SMTxTMR. Example waveforms for both repeated and single acquisitions are provided in Figure 30-4 and Figure 30-5.

FIGURE 30-12: GATED WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM



REGISTER 32-14: PIDxOUTHL: PID OUTPUT HIGH LOW 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
OUT<23:16>							
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	q = Value depends on condition

bit 7-0 **OUT<23:16>:** Bits <23:16> of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

REGISTER 32-15: PIDxOUTLH: PID OUTPUT LOW HIGH 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
OUT<15:8>							
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	q = Value depends on condition

bit 7-0 **OUT<15:8>:** Bits <15:8> of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

REGISTER 32-16: PIDxOUTLL: PID OUTPUT LOW LOW 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
OUT<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	q = Value depends on condition

bit 7-0 **OUT<7:0>:** Bits <7:0> of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

FIGURE 35-11: ADC CONVERSION TIMING (ADC CLOCK FOSC-BASED)

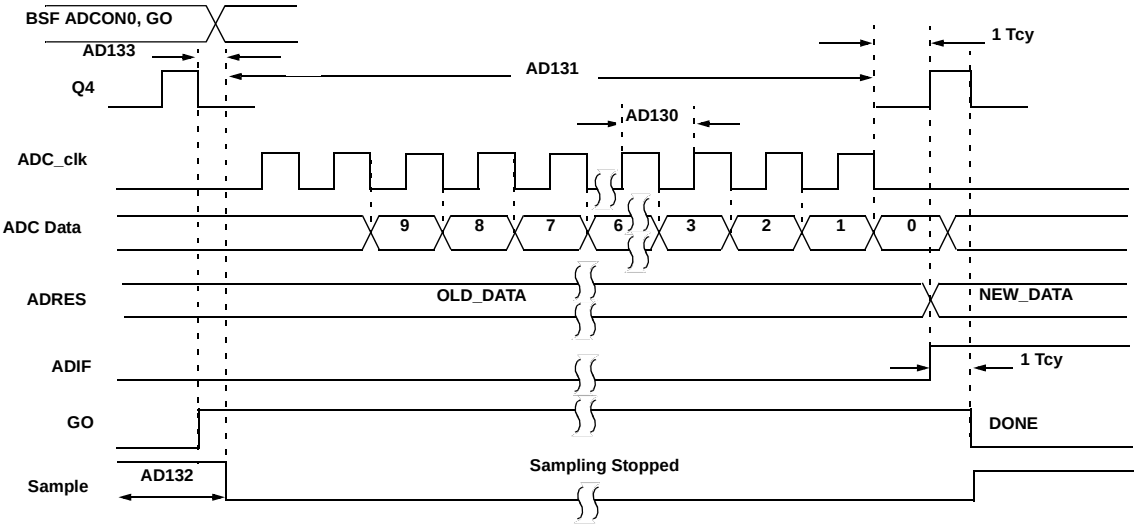
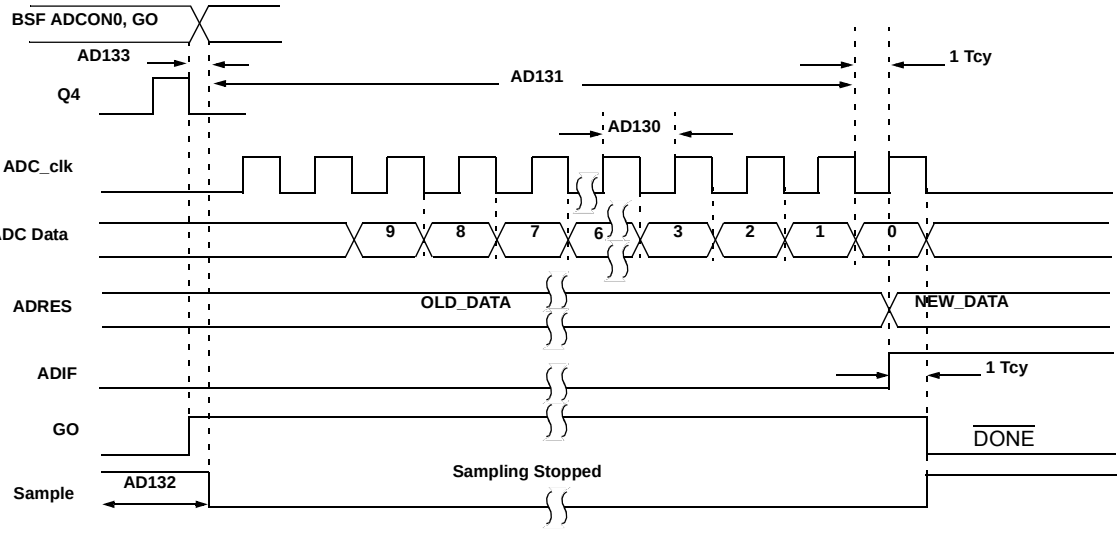


FIGURE 35-12: ADC CONVERSION TIMING (ADC CLOCK FROM FRC)



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

Note: Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

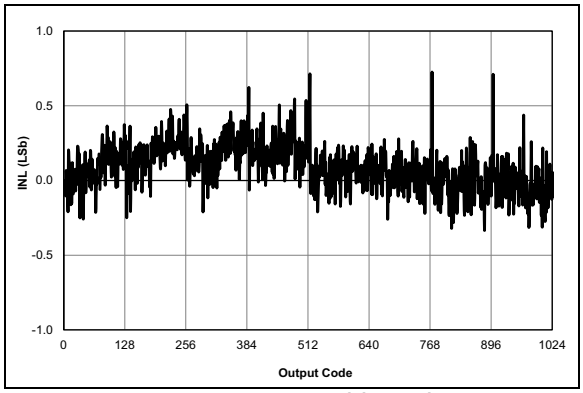


FIGURE 36-67: *ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 1 µS, 25°C.*

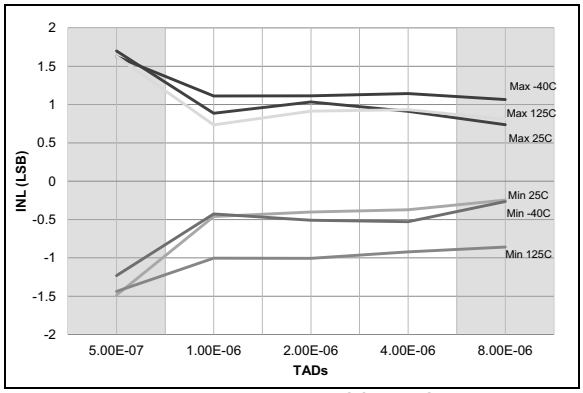


FIGURE 36-70: *ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, VREF = 3.0V.*

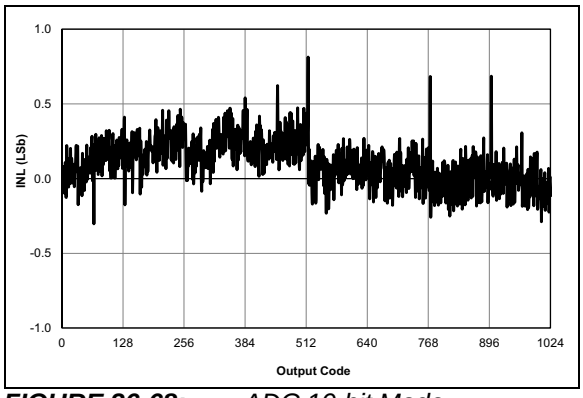


FIGURE 36-68: *ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 4 µS, 25°C.*

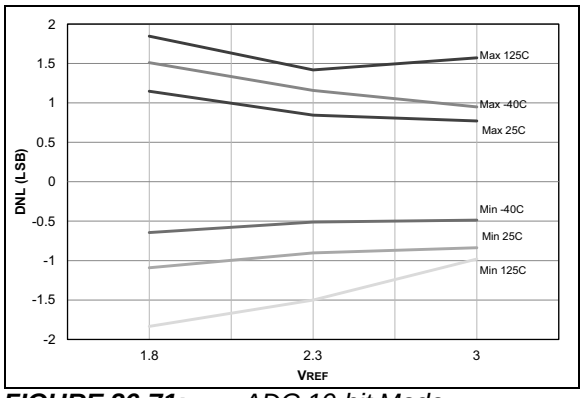


FIGURE 36-71: *ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, TAD = 1 µS.*

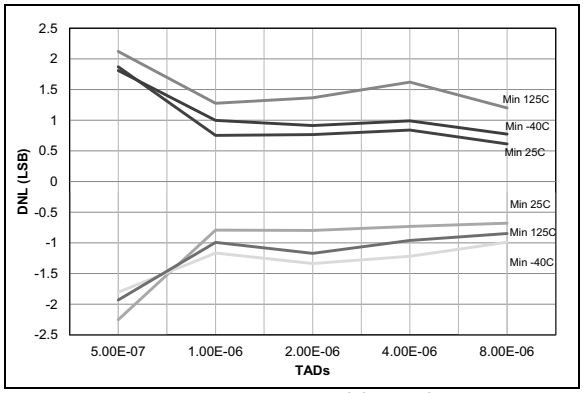


FIGURE 36-69: *ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, VREF = 3.0V.*

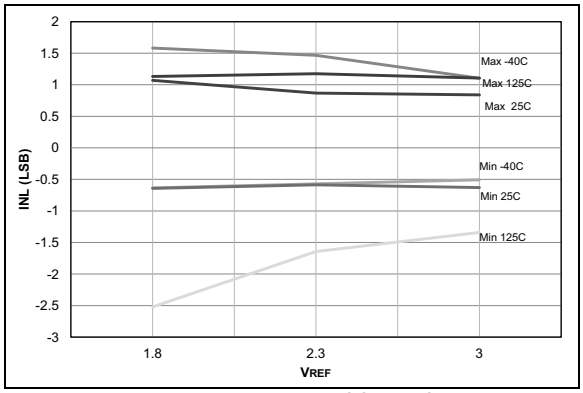


FIGURE 36-72: *ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 1 µS.*

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
- 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