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	20MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	18
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	64 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f631t-i-ss

2.2.2.2 OPTION Register

The OPTION register, shown in Register 2-2, is a readable and writable register, which contains various control bits to configure:

- Timer0/WDT prescaler
- External RA2/INT interrupt
- Timer0
- Weak pull-ups on PORTA/PORTB

Note: To achieve a 1:1 prescaler assignment for Timer0, assign the prescaler to the WDT by setting PSA bit of the OPTION register to '1'. See **Section 6.3 "Timer1 Prescaler"**.

REGISTER 2-2: OPTION_REG: OPTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RABPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **$\overline{\text{RABPU}}$:** PORTA/PORTB Pull-up Enable bit
1 = PORTA/PORTB pull-ups are disabled
0 = PORTA/PORTB pull-ups are enabled by individual PORT latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of RA2/INT pin
0 = Interrupt on falling edge of RA2/INT pin

bit 5 **T0CS:** Timer0 Clock Source Select bit
1 = Transition on RA2/T0CKI pin
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **T0SE:** Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on RA2/T0CKI pin
0 = Increment on low-to-high transition on RA2/T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

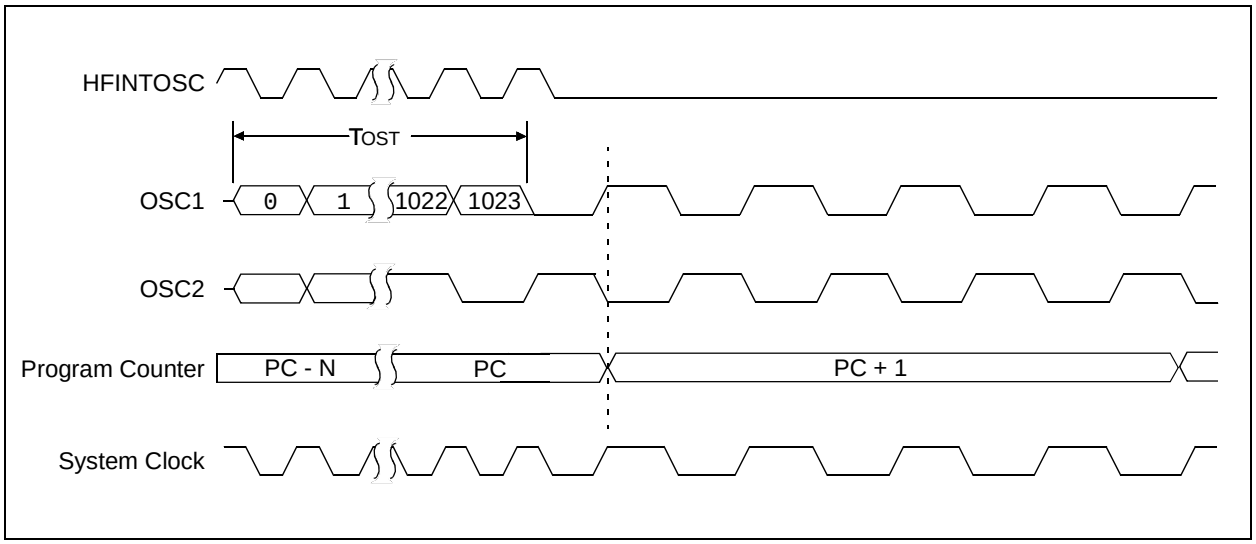
bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	Timer0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

3.7.3 CHECKING TWO-SPEED CLOCK STATUS

Checking the state of the OSTS bit of the OSCCON register will confirm if the microcontroller is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word register (CONFIG), or the internal oscillator.

FIGURE 3-7: TWO-SPEED START-UP



4.3 PORTB and TRISB Registers

PORTB is a 4-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 4-6). Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 4-3 shows how to initialize PORTB. Reading the PORTB register (Register 4-5) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch.

The TRISB register controls the PORTB pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISB register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

EXAMPLE 4-3: INITIALIZING PORTB

```
BCF     STATUS,RP0 ;Bank 0
BCF     STATUS,RP1 ;
CLRF    PORTB      ;Init PORTB
BSF     STATUS,RP0 ;Bank 1
MOVLW   FFh        ;Set RB<7:4> as inputs
MOVWF   TRISB      ;
BCF     STATUS,RP0 ;Bank 0
```

Note:	The ANSELH register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

4.4 Additional PORTB Pin Functions

PORTB pins RB<7:4> on the device family device have an interrupt-on-change option and a weak pull-up option. The following three sections describe these PORTB pin functions.

4.4.1 WEAK PULL-UPS

Each of the PORTB pins has an individually configurable internal weak pull-up. Control bits WPUB<7:4> enable or disable each pull-up (see Register 4-9). Each weak pull up is automatically turned off when the port pin is configured as an output. All pull-ups are disabled on a Power-on Reset by the RABPU bit of the OPTION register.

4.4.2 INTERRUPT-ON-CHANGE

Four of the PORTB pins are individually configurable as an interrupt-on-change pin. Control bits IOCB<7:4> enable or disable the interrupt function for each pin. Refer to Register 4-10. The interrupt-on-change feature is disabled on a Power-on Reset.

For enabled interrupt-on-change pins, the present value is compared with the old value latched on the last read of PORTB to determine which bits have changed or mismatch the old value. The 'mismatch' outputs are OR'd together to set the PORTB Change Interrupt flag bit (RABIF) in the INTCON register (Register 2-3).

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, clears the interrupt by:

a) Any read or write of PORTB. This will end the mismatch condition.
b) Clear the flag bit RABIF.

A mismatch condition will continue to set flag bit RABIF. Reading or writing PORTB will end the mismatch condition and allow flag bit RABIF to be cleared. The latch holding the last read value is not affected by a MCLR nor Brown-out Reset. After these Resets, the RABIF flag will continue to be set if a mismatch is present.

Note:	If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RABIF interrupt flag may not get set. Furthermore, since a read or write on a port affects all bits of that port, care must be taken when using multiple pins in Interrupt-on-Change mode. Changes on one pin may not be seen while servicing changes on another pin.

REGISTER 4-7: PORTB: PORTB REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	U-0	U-0	U-0	U-0
RB7	RB6	RB5	RB4	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-4 **RB<7:4>**: PORTB I/O Pin bit
1 = Port pin is > V_{IH}
0 = Port pin is < V_{IL}

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

TABLE 4-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

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
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—	0000 ----	0000 ----
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PORTB	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	uuuu ----
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	1111 ----	1111 ----

Legend: x = unknown, u = unchanged, – = unimplemented read as '0'. Shaded cells are not used by PORTB.

5.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (shared with Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt on overflow

Figure 5-1 is a block diagram of the Timer0 module.

5.1 Timer0 Operation

When used as a timer, the Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

5.1.1 8-BIT TIMER MODE

When used as a timer, the Timer0 module will increment every instruction cycle (without prescaler). Timer mode is selected by clearing the T0CS bit of the OPTION register to '0'.

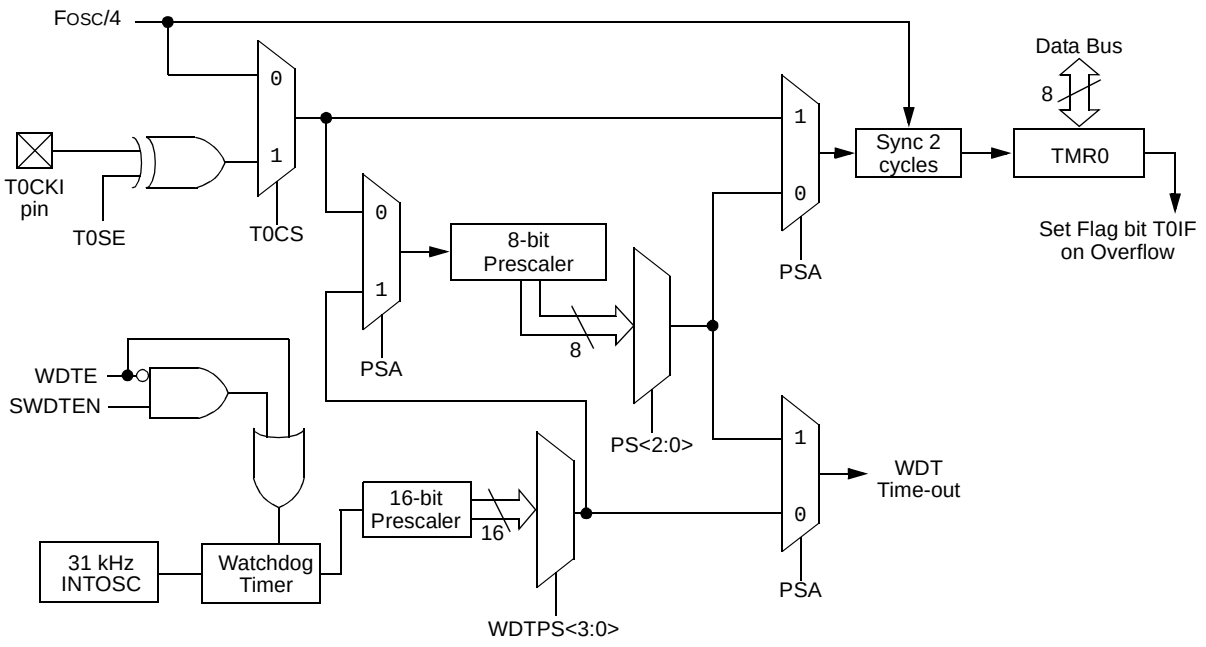
When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

Note:	The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

5.1.2 8-BIT COUNTER MODE

When used as a counter, the Timer0 module will increment on every rising or falling edge of the T0CKI pin. The incrementing edge is determined by the T0SE bit of the OPTION register. Counter mode is selected by setting the T0CS bit of the OPTION register to '1'.

FIGURE 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



Note 1: T0SE, T0CS, PSA, PS<2:0> are bits in the OPTION register.
2: SWDTEN and WDTPS<3:0> are bits in the WDTCON register.
3: WDTE bit is in the Configuration Word register.

9.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESL and ADRESH).

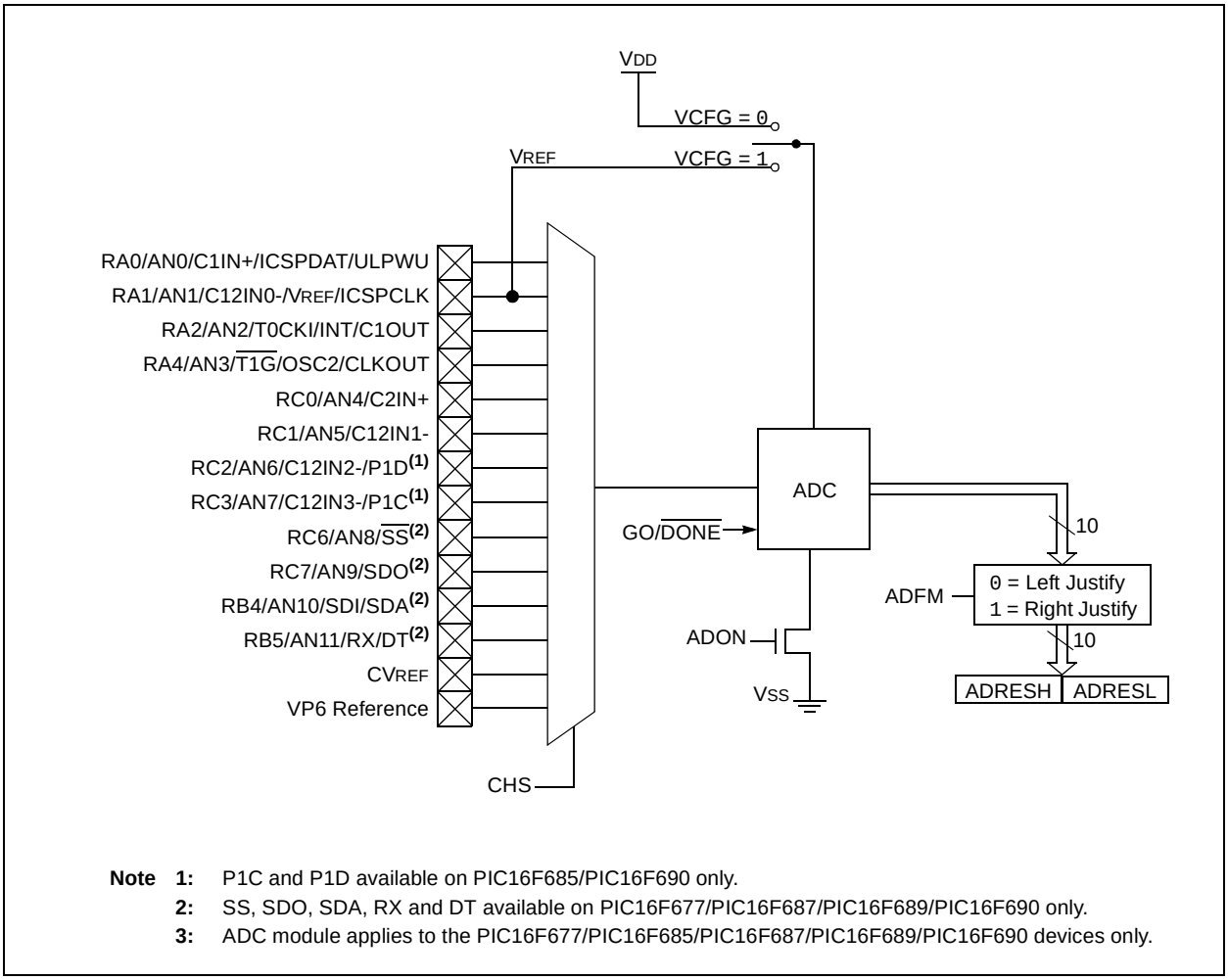
Note:	The ADC module applies to PIC16F677/ PIC16F685/PIC16F687/PIC16F689/ PIC16F690 devices only.

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.

Figure 9-1 shows the block diagram of the ADC.

FIGURE 9-1: ADC BLOCK DIAGRAM



Note 1: P1C and P1D available on PIC16F685/PIC16F690 only.
2: SS, SDO, SDA, RX and DT available on PIC16F677/PIC16F687/PIC16F689/PIC16F690 only.
3: ADC module applies to the PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690 devices only.

9.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Results formatting

9.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. See the corresponding port section for more information.

Note:	Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

9.1.2 CHANNEL SELECTION

The CHS bits of the ADCON0 register determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 9.2 "ADC Operation"** for more information.

9.1.3 ADC VOLTAGE REFERENCE

The VCFG bit of the ADCON0 register provides control of the positive voltage reference. The positive voltage reference can be either VDD or an external voltage source. The negative voltage reference is always connected to the ground reference.

9.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits of the ADCON1 register. There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC (dedicated internal oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11 TAD periods as shown in Figure 9-2.

For correct conversion, the appropriate TAD specification must be met. See A/D conversion requirements in **Section 17.0 "Electrical Specifications"** for more information. Table 9-1 gives examples of appropriate ADC clock selections.

Note:	Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

REGISTER 9-1: ADCON0: A/D CONTROL REGISTER 0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	VCFG	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **ADFM:** A/D Conversion Result Format Select bit
1 = Right justified
0 = Left justified

bit 6 **VCFG:** Voltage Reference bit
1 = V_{REF} pin
0 = V_{DD}

bit 5-2 **CHS<3:0>: Analog Channel Select bits**
0000 = AN0
0001 = AN1
0010 = AN2
0011 = AN3
0100 = AN4
0101 = AN5
0110 = AN6
0111 = AN7
1000 = AN8
1001 = AN9
1010 = AN10
1011 = AN11
1100 = CV_{REF}
1101 = 0.6V Fixed Voltage Reference
1110 = Reserved. Do not use.
1111 = Reserved. Do not use.

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

11.3.4 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the CCP1 pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

11.3.5 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 3.0 "Oscillator Module (With Fail-Safe Clock Monitor)"** for additional details.

11.3.6 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

11.3.7 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Disable the PWM pin (CCP1) output driver by setting the associated TRIS bit.
2. Set the PWM period by loading the PR2 register.
3. Configure the CCP module for the PWM mode by loading the CCP1CON register with the appropriate values.
4. Set the PWM duty cycle by loading the CCPR1L register and DC1B<1:0> bits of the CCP1CON register.
5. Configure and start Timer2:
   - •Clear the TMR2IF interrupt flag bit of the PIR1 register.
   - •Set the Timer2 prescale value by loading the T2CKPS bits of the T2CON register.
   - •Enable Timer2 by setting the TMR2ON bit of the T2CON register.
6. Enable PWM output after a new PWM cycle has started:
   - •Wait until Timer2 overflows (TMR2IF bit of the PIR1 register is set).
   - Enable the CCP1 pin output driver by clearing the associated TRIS bit.

11.4 PWM (Enhanced Mode)

The Enhanced PWM Mode can generate a PWM signal on up to four different output pins with up to ten bits of resolution. It can do this through four different PWM Output modes:

- Single PWM
- Half-Bridge PWM
- Full-Bridge PWM, Forward mode
- Full-Bridge PWM, Reverse mode

To select an Enhanced PWM mode, the P1M bits of the CCP1CON register must be set appropriately.

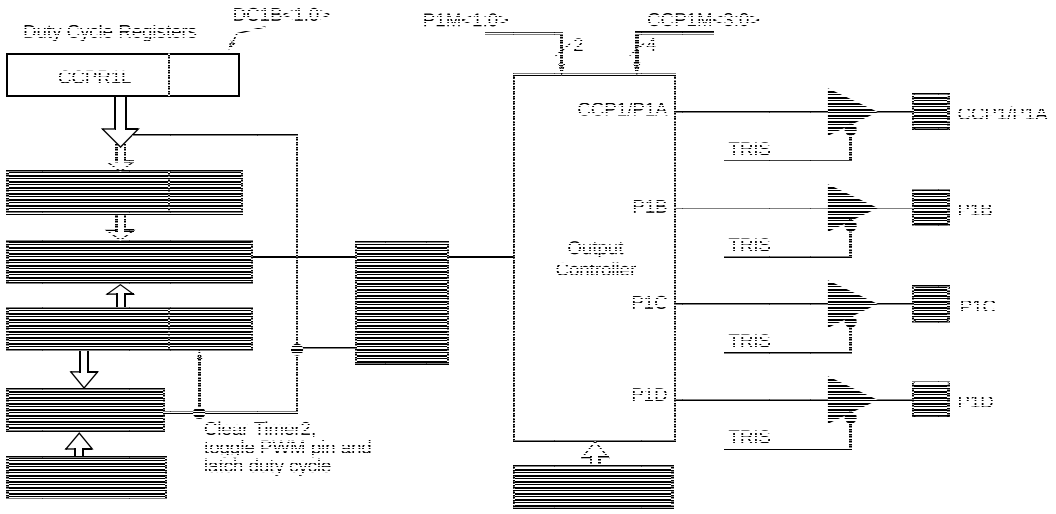
The PWM outputs are multiplexed with I/O pins and are designated P1A, P1B, P1C and P1D. The polarity of the PWM pins is configurable and is selected by setting the CCP1M bits in the CCP1CON register appropriately.

Table 11-4 shows the pin assignments for each Enhanced PWM mode.

Figure 11-5 shows an example of a simplified block diagram of the Enhanced PWM module.

Note:	To prevent the generation of an incomplete waveform when the PWM is first enabled, the ECCP module waits until the start of a new PWM period before generating a PWM signal.

FIGURE 11-5: EXAMPLE SIMPLIFIED BLOCK DIAGRAM OF THE ENHANCED PWM MODE



Note 1: The 8-bit timer TMR2 register is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler to create the 10-bit time base.

Note 1: The TRIS register value for each PWM output must be configured appropriately.

2: Clearing the CCP1CON register will relinquish ECCP control of all PWM output pins.

3: Any pin not used by an Enhanced PWM mode is available for alternate pin functions

TABLE 11-4: EXAMPLE PIN ASSIGNMENTS FOR VARIOUS PWM ENHANCED MODES

ECCP Mode	P1M<1:0>	CCP1/P1A	P1B	P1C	P1D
Single	00	Yes(1)	Yes(1)	Yes(1)	Yes(1)
Half-Bridge	10	Yes	Yes	No	No
Full-Bridge, Forward	01	Yes	Yes	Yes	Yes
Full-Bridge, Reverse	11	Yes	Yes	Yes	Yes

Note 1: Pulse Steering enables outputs in Single mode.

TABLE 12-5: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	FOSC = 20.000 MHz			FOSC = 18.432 MHz			FOSC = 11.0592 MHz			FOSC = 8.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	1221	1.73	255	1200	0.00	239	1200	0.00	143	1202	0.16	103
2400	2404	0.16	129	2400	0.00	119	2400	0.00	71	2404	0.16	51
9600	9470	-1.36	32	9600	0.00	29	9600	0.00	17	9615	0.16	12
10417	10417	0.00	29	10286	-1.26	27	10165	-2.42	16	10417	0.00	11
19.2k	19.53k	1.73	15	19.20k	0.00	14	19.20k	0.00	8	—	—	—
57.6k	—	—	—	57.60k	0.00	7	57.60k	0.00	2	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	FOSC = 4.000 MHz			FOSC = 3.6864 MHz			FOSC = 2.000 MHz			FOSC = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300	0.16	207	300	0.00	191	300	0.16	103	300	0.16	51
1200	1202	0.16	51	1200	0.00	47	1202	0.16	25	1202	0.16	12
2400	2404	0.16	25	2400	0.00	23	2404	0.16	12	—	—	—
9600	—	—	—	9600	0.00	5	—	—	—	—	—	—
10417	10417	0.00	5	—	—	—	10417	0.00	2	—	—	—
19.2k	—	—	—	19.20k	0.00	2	—	—	—	—	—	—
57.6k	—	—	—	57.60k	0.00	0	—	—	—	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	FOSC = 20.000 MHz			FOSC = 18.432 MHz			FOSC = 11.0592 MHz			FOSC = 8.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	2404	0.16	207
9600	9615	0.16	129	9600	0.00	119	9600	0.00	71	9615	0.16	51
10417	10417	0.00	119	10378	-0.37	110	10473	0.53	65	10417	0.00	47
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35	19231	0.16	25
57.6k	56.82k	-1.36	21	57.60k	0.00	19	57.60k	0.00	11	55556	-3.55	8
115.2k	113.64k	-1.36	10	115.2k	0.00	9	115.2k	0.00	5	—	—	—

13.12.3 SSP MASK REGISTER

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

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

This register must be initiated prior to setting SSPM<3:0> bits to select the I^2C Slave mode (7-bit or 10-bit address).

This register can only be accessed when the appropriate mode is selected by bits (SSPM<3:0> of SSPCON).

The SSP Mask register is active during:

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

REGISTER 13-3: SSPMSK: SSP MASK REGISTER(1)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0(2)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-1 **MSK<7:1>:** Mask bits

1 = The received address bit n is compared to SSPADD<n> to detect I^2C address match
0 = The received address bit n is not used to detect I^2C address match

bit 0 **MSK<0>:** Mask bit for I^2C Slave mode, 10-bit Address(2)

I^2C Slave mode, 10-bit Address (SSPM<3:0> = 0111):
1 = The received address bit 0 is compared to SSPADD<0> to detect I^2C address match
0 = The received address bit 0 is not used to detect I^2C address match

Note 1: When SSPCON bits SSPM<3:0> = 1001, any reads or writes to the SSPADD SFR address are accessed through the SSPMSK register. The SSPEN bit of the SSPCON register should be zero when accessing the SSPMSK register.

2: In all other SSP modes, this bit has no effect.

TABLE 14-4: INITIALIZATION CONDITION FOR REGISTER (CONTINUED)

Register	Address	Power-on Reset	$\overline{\text{MCLR}}$ Reset WDT Reset (Continued) Brown-out Reset(1)	Wake-up from Sleep through Interrupt Wake-up from Sleep through WDT Time-out
TRISB	86h/186h	1111 ----	1111 ----	uuuu ----
TRISC	87h/187h	1111 1111	1111 1111	uuuu uuuu
PIE1	8Ch	-000 0000	-000 0000	-uuu uuuu
PIE2	8Dh	0000 ----	0000 ----	uuuu uuuu
PCON	8Eh	--01 --0x	--0u --uq[1, 5]	--uu --uu
OSCCON	8Fh	-110 q000	-110 q000	-uuu uuuu
OSCTUNE	90h	---0 0000	---u uuuu	---u uuuu
PR2	92h	1111 1111	1111 1111	uuuu uuuu
SSPADD	93h	0000 0000	1111 1111	uuuu uuuu
SSPMSK(6)	93h	---- ----	1111 1111	uuuu uuuu
SSPSTAT	94h	0000 0000	1111 1111	uuuu uuuu
WPUA	95h	--11 -111	--11 -111	uuuu uuuu
IOCA	96h	--00 0000	--00 0000	--uu uuuu
WDTCON	97h	---0 1000	---0 1000	---u uuuu
TXSTA	98h	0000 0010	0000 0010	uuuu uuuu
SPBRG	99h	0000 0000	0000 0000	uuuu uuuu
SPBRGH	9Ah	0000 0000	0000 0000	uuuu uuuu
BAUDCTL	9Bh	01-0 0-00	01-0 0-00	uu-u u-uu
ADRESL	9Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	9Fh	-000 ----	-000 ----	-uuu ----
EEDAT	10Ch	0000 0000	0000 0000	uuuu uuuu
EEADR	10Dh	0000 0000	0000 0000	uuuu uuuu
EEDATH	10Eh	--00 0000	--00 0000	--uu uuuu
EEADRH	10Fh	---- 0000	---- 0000	---- uuuu
WPUB	115h	1111 ----	1111 ----	uuuu ----
IOCB	116h	0000 ----	0000 ----	uuuu ----
VRCON	118h	0000 0000	0000 0000	uuuu uuuu
CM1CON0	119h	0000 -000	0000 -000	uuuu -uuu
CM2CON0	11Ah	0000 -000	0000 -000	uuuu -uuu
CM2CON1	11Bh	00-- --00	00-- --10	uu-- --uu
ANSEL	11Eh	1111 1111	1111 1111	uuuu uuuu
ANSELH	11Fh	---- 1111	---- 1111	---- uuuu
EECON1	18Ch	x--- x000	u--- q000	---- uuuu
EECON2	18Dh	---- ----	---- ----	---- ----
PSTRCON	19Dh	---0 0001	---0 0001	---u uuuu
SRCON	19EH	0000 00--	0000 00--	uuuu uu--

Legend: u = unchanged, x = unknown, – = unimplemented bit, reads as '0', q = value depends on condition.

Note 1: If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

2: One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

3: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

4: See Table 14-5 for Reset value for specific condition.

5: If Reset was due to brown-out, then bit 0 = 0. All other Resets will cause bit 0 = u.

6: Accessible only when SSPM<3:0> = 1001.

TABLE 14-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS

14.4 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (e.g., W and STATUS registers). This must be implemented in software.

Since the upper 16 bytes of all GPR banks are common in the PIC16F631/677/685/687/689/690 (see Figures 2-2 and 2-3), temporary holding registers, W_TEMP and STATUS_TEMP, should be placed in here. These 16 locations do not require banking and therefore, make it easier to context save and restore. The same code shown in Example 14-1 can be used to:

- Store the W register
- Store the STATUS register
- Execute the ISR code
- Restore the Status (and Bank Select Bit register)
- Restore the W register

Note: The PIC16F631/677/685/687/689/690 normally does not require saving the PCLATH. However, if computed GOTO's are used in the ISR and the main code, the PCLATH must be saved and restored in the ISR.

EXAMPLE 14-1: SAVING STATUS AND W REGISTERS IN RAM

```
MOVWF  W_TEMP           ;Copy W to TEMP register
SWAPF  STATUS,W         ;Swap status to be saved into W
CLRF   STATUS           ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF  STATUS_TEMP      ;Save status to bank zero STATUS_TEMP register
:
:(ISR)                  ;Insert user code here
:
SWAPF  STATUS_TEMP,W    ;Swap STATUS_TEMP register into W
                        ;(sets bank to original state)
MOVWF  STATUS           ;Move W into STATUS register
SWAPF  W_TEMP,F         ;Swap W_TEMP
SWAPF  W_TEMP,W         ;Swap W_TEMP into W
```

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	$0 \le f \le 127$ $0 \le b < 7$
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.

CALL	Call Subroutine
Syntax:	[*label*] CALL k
Operands:	$0 \le k \le 2047$
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a 2-cycle instruction.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	$0 \le f \le 127$
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

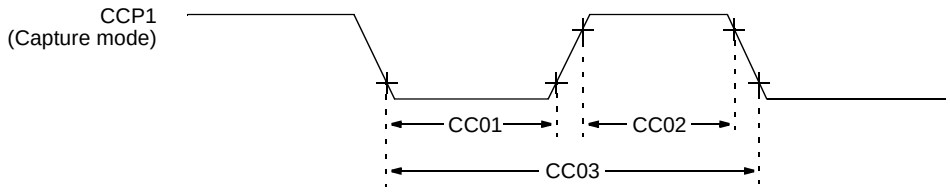
CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	$(\bar{f}) \rightarrow$ (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

FIGURE 17-9: CAPTURE/COMPARE/PWM TIMINGS (ECCP)



Note: Refer to Figure 17-3 for load conditions.

TABLE 17-6: CAPTURE/COMPARE/PWM REQUIREMENTS (ECCP)

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ TA ≤ +125°C								
Param No.	**Sym.**	**Characteristic**		**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
CC01*	TccL	CCP1 Input Low Time	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	20	—	—	ns	
CC02*	TccH	CCP1 Input High Time	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	20	—	—	ns	
CC03*	TccP	CCP1 Input Period		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 4 or 16)

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 17-14: I²C™ BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
100*	THIGH	Clock high time	100 kHz mode	4.0	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	µs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
101*	TLOW	Clock low time	100 kHz mode	4.7	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	µs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10-400 pF
103*	TF	SDA and SCL fall time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10-400 pF
90*	TSU:STA	Start condition setup time	100 kHz mode	4.7	—	µs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	µs	
91*	THD:STA	Start condition hold time	100 kHz mode	4.0	—	µs	After this period the first clock pulse is generated
			400 kHz mode	0.6	—	µs	
106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	µs	
107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
92*	TSU:STO	Stop condition setup time	100 kHz mode	4.7	—	µs	
			400 kHz mode	0.6	—	µs	
109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	**(Note 1)**
			400 kHz mode	—	—	ns	
110*	TBUF	Bus free time	100 kHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
	CB	Bus capacitive loading		—	400	pF	

* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

2: A Fast mode (400 kHz) I²C bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement TSU:DAT ≥ 250 ns must then be met. This will automatically be the case 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. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

FIGURE 18-4: MAXIMUM IDD vs. FOSC OVER VDD (HS MODE)

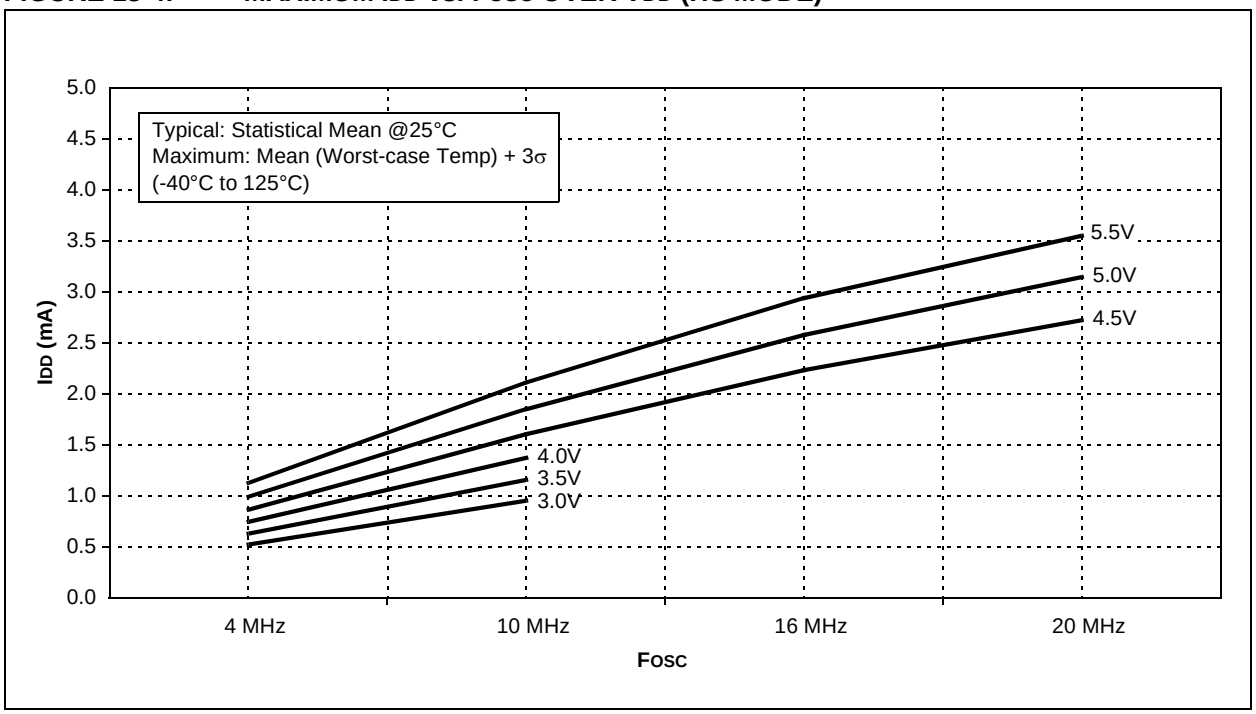


FIGURE 18-5: TYPICAL IDD vs. VDD OVER FOSC (XT MODE)

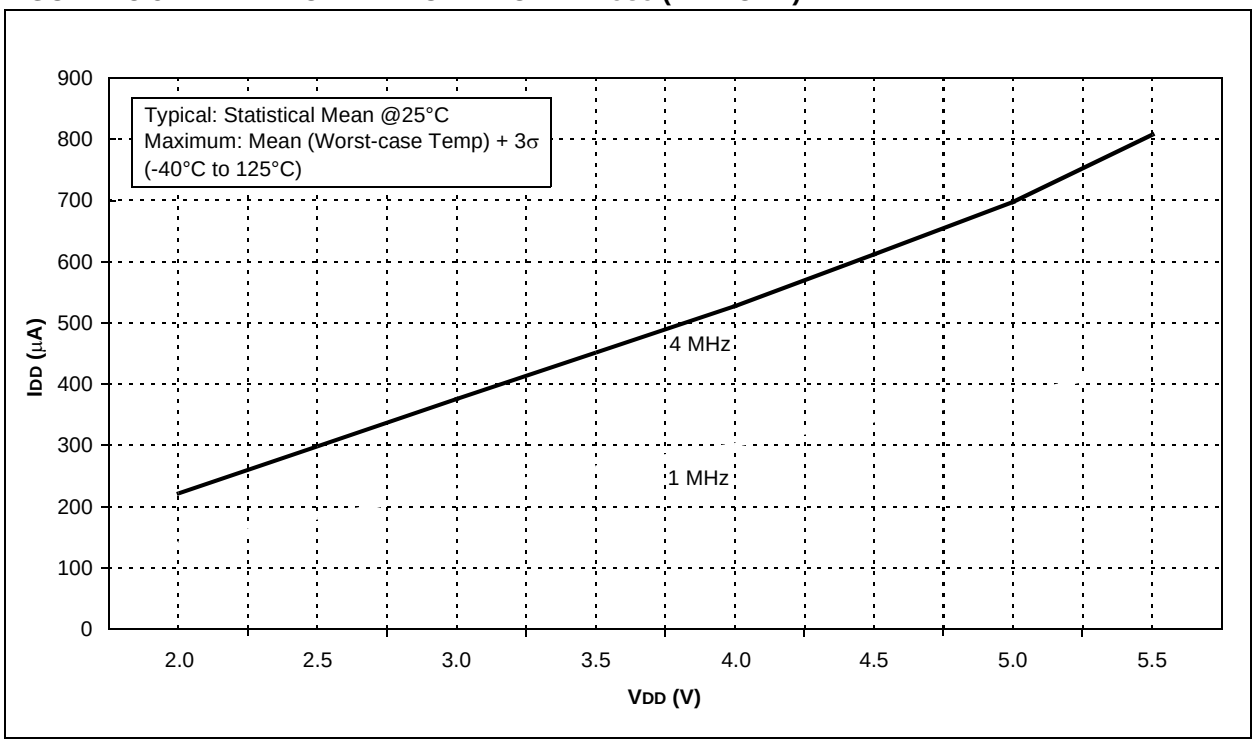


FIGURE 18-34: LFINTOSC FREQUENCY vs. VDD OVER TEMPERATURE (31 kHz)

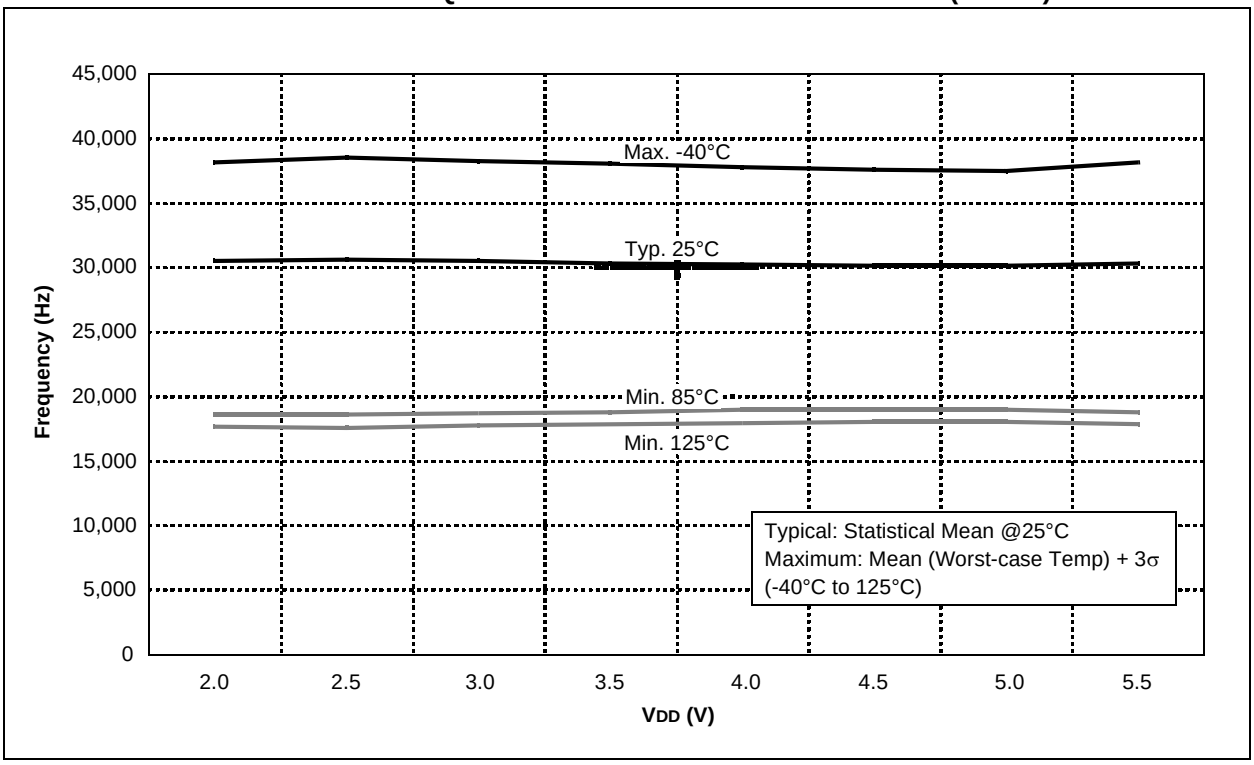
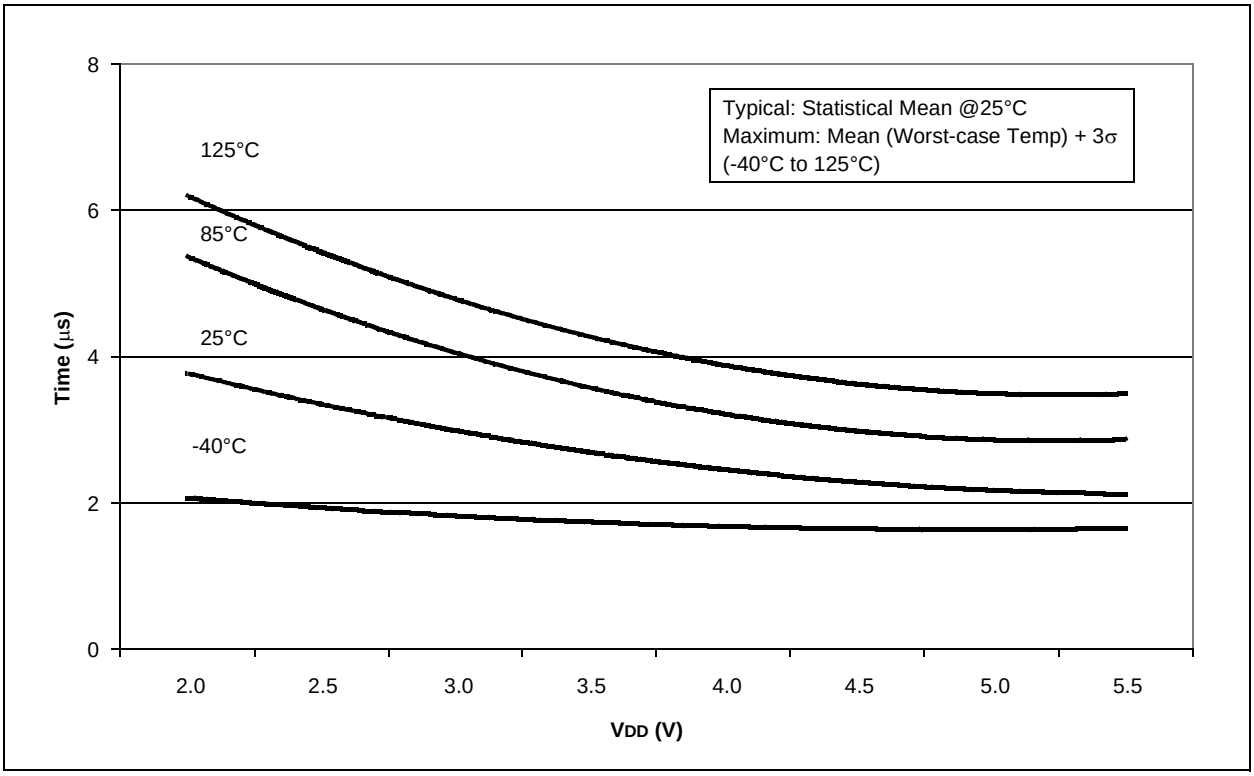


FIGURE 18-35: ADC CLOCK PERIOD vs. VDD OVER TEMPERATURE



PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX	XXX
Device	Temperature Range	Package	Pattern

Device:	PIC16F631[(1)], PIC16F677[(1)], PIC16F685[(1)], PIC16F687[(1)], PIC16F689[(1)], PIC16F690[(1)]; V_{DD} range 2.0V to 5.5V
Temperature Range:	I = -40°C to +85°C (Industrial) E = -40°C to +125°C (Extended)
Package:	ML = QFN (Quad Flat, no lead) P = PDIP SO = SOIC SS = SSOP
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)

Examples:

a) PIC16F685 - I/ML 301 = Industrial temp., QFN package, QTP pattern #301.

b) PIC16F689 - I/SO = Industrial temp., SOIC package.

c) PIC16F690T - E/SS = Extended temp., SSOP package.

Note 1: T = in tape and reel SSOP, SOIC and QFN packages only.