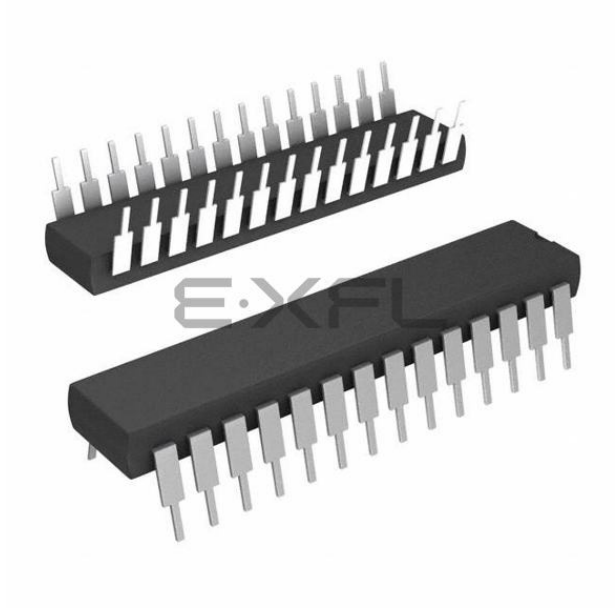


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

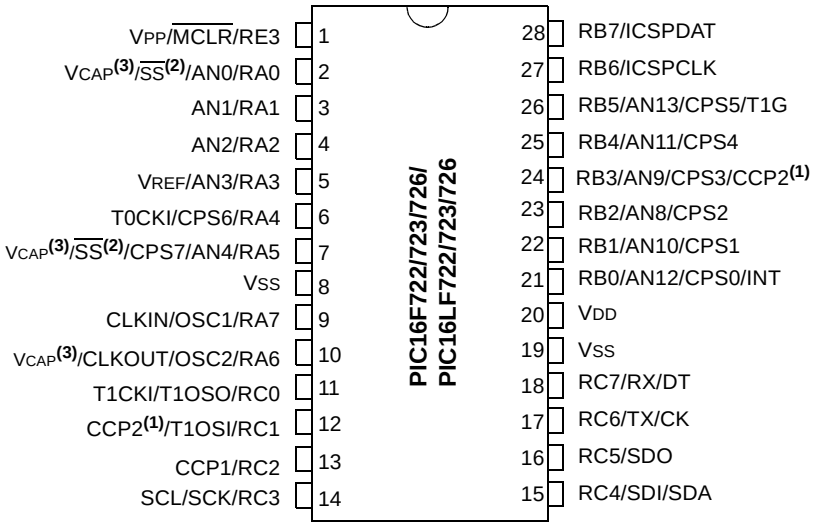
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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"

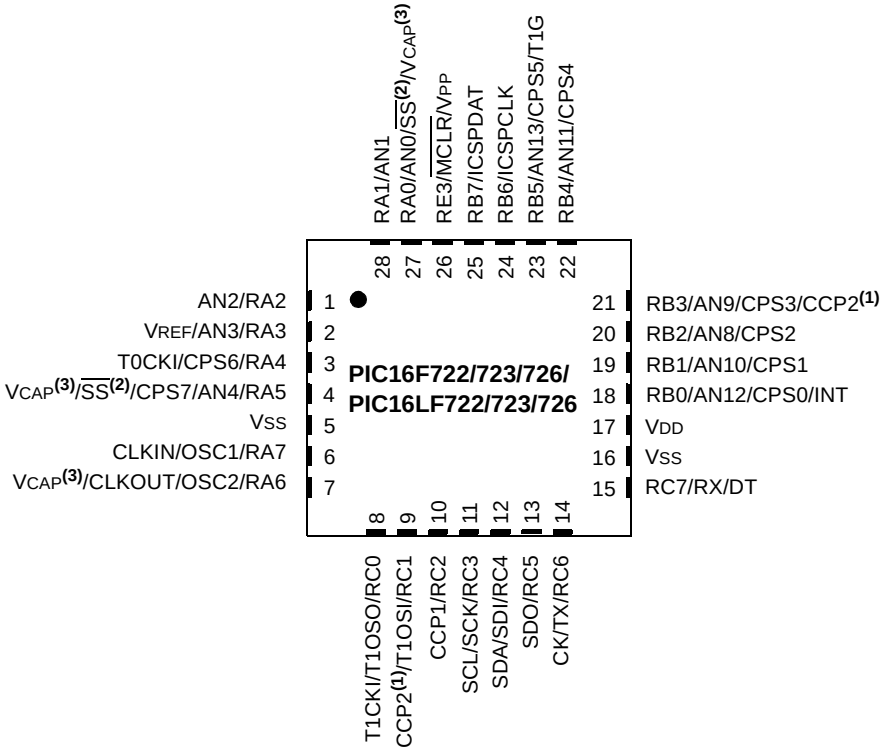
Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	368 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 11x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf726-e-sp

Pin Diagrams – 28-PIN PDIP/SOIC/SSOP/QFN/UQFN (PIC16F722/723/726/PIC16LF722/723/726)

PDIP, SOIC, SSOP



QFN, UQFN



Note 1: CCP2 pin location may be selected as RB3 or RC1.
2: SS pin location may be selected as RA5 or RA0.
3: PIC16F722/723/726 devices only.

REGISTER 6-5: PORTB: PORTB REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
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 7-0 **RB<7:0>**: PORTB I/O Pin bit
1 = Port pin is > V_{IH}
0 = Port pin is < V_{IL}

REGISTER 6-6: TRISB: PORTB TRI-STATE 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
TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
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 7-0 **TRISB<7:0>:** PORTB Tri-State Control bit
1 = PORTB pin configured as an input (tri-stated)
0 = PORTB pin configured as an output

FIGURE 6-11: BLOCK DIAGRAM OF RB6

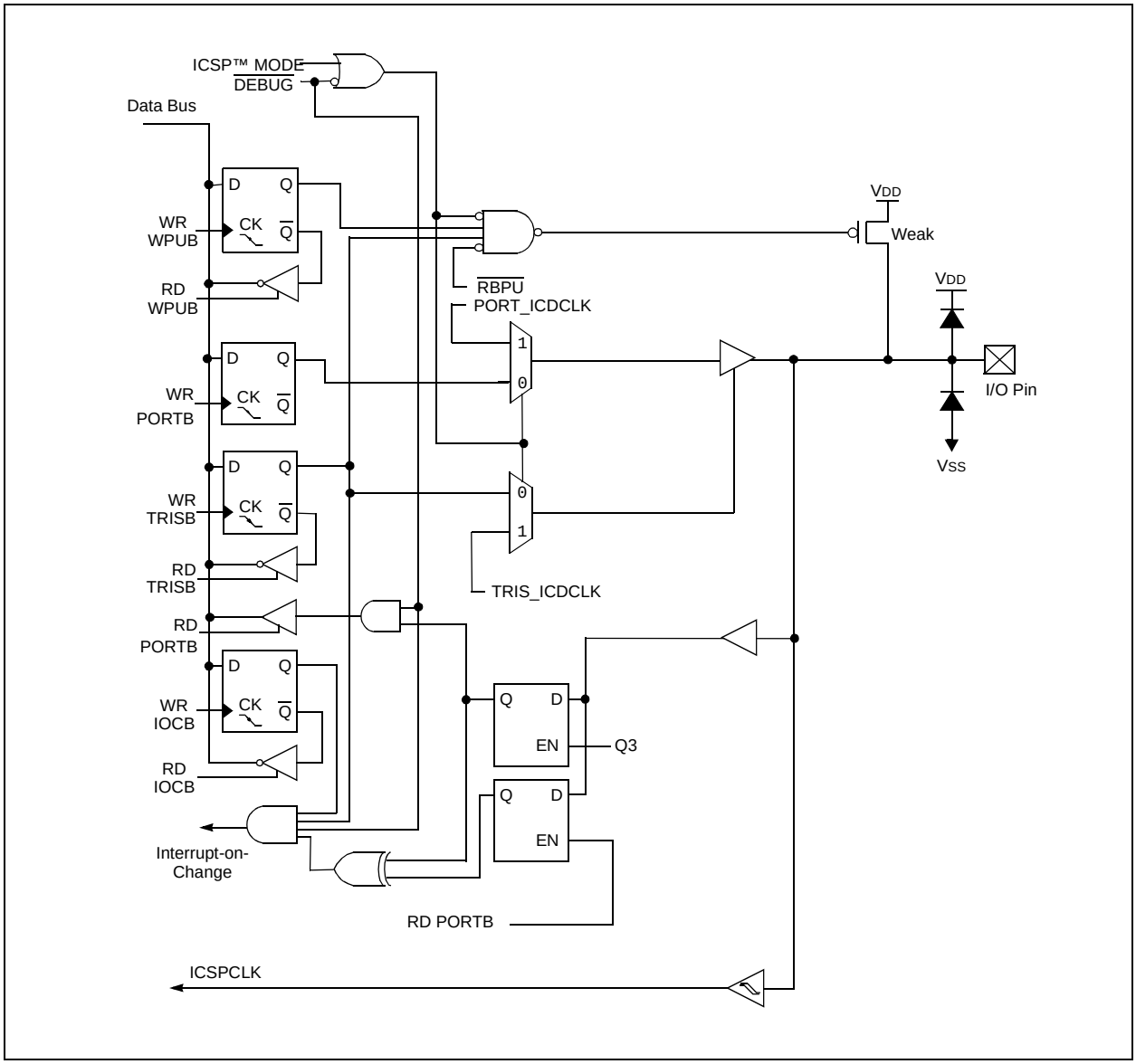
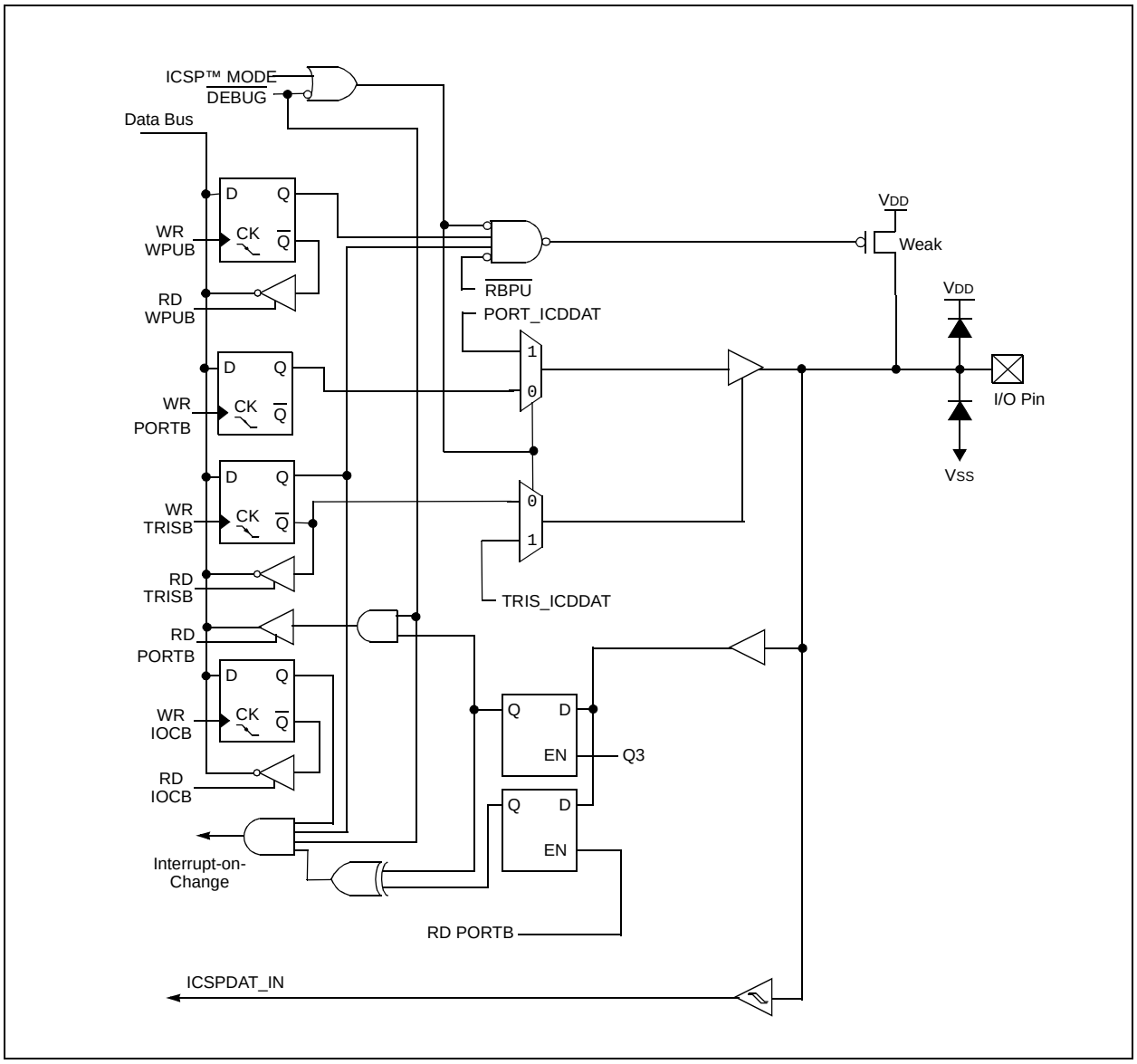


FIGURE 6-12: BLOCK DIAGRAM OF RB7



12.11 Timer1 Control Register

The Timer1 Control register (T1CON), shown in Register 12-1, is used to control Timer1 and select the various features of the Timer1 module.

REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON
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 7-6 **TMR1CS<1:0>:** Timer1 Clock Source Select bits
11 = Timer1 clock source is Capacitive Sensing Oscillator (CAPOSC)
10 = Timer1 clock source is pin or oscillator:
If T1OSCEN = 0:
External clock from T1CKI pin (on the rising edge)
If T1OSCEN = 1:
Crystal oscillator on T1OSI/T1OSO pins
01 = Timer1 clock source is system clock (FOSC)
00 = Timer1 clock source is instruction clock (FOSC/4)

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN:** LP Oscillator Enable Control bit
1 = Dedicated Timer1 oscillator circuit enabled
0 = Dedicated Timer1 oscillator circuit disabled

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Control bit
TMR1CS<1:0> = 1X
1 = Do not synchronize external clock input
0 = Synchronize external clock input with system clock (FOSC)

TMR1CS<1:0> = 0X
This bit is ignored. Timer1 uses the internal clock when TMR1CS<1:0> = 1X.

bit 1 **Unimplemented:** Read as '0'

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1
Clears Timer1 Gate flip-flop

TABLE 12-6: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1

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
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	--11 1111	--11 1111
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	xxxx xxxx
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	0000 00-0	uuuu uu-u
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1$\overline{\text{GGO}}$/$\overline{\text{DONE}}$	T1GVAL	T1GSS1	T1GSS0	0000 0x00	uuuu uxuu

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.

REGISTER 14-1: CPSCON0: CAPACITIVE SENSING CONTROL REGISTER 0

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R-0	R/W-0
CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS
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 **CPSON:** Capacitive Sensing Module Enable bit
1 = Capacitive sensing module is operating
0 = Capacitive sensing module is shut off and consumes no operating current

bit 6-4 **Unimplemented:** Read as '0'

bit 3-2 **CPSRNG<1:0>:** Capacitive Sensing Oscillator Range bits
00 = Oscillator is Off.
01 = Oscillator is in low range. Charge/discharge current is nominally 0.1 µA.
10 = Oscillator is in medium range. Charge/discharge current is nominally 1.2 µA.
11 = Oscillator is in high range. Charge/discharge current is nominally 18 µA.

bit 1 **CPSOUT:** Capacitive Sensing Oscillator Status bit
1 = Oscillator is sourcing current (Current flowing out the pin)
0 = Oscillator is sinking current (Current flowing into the pin)

bit 0 **T0XCS:** Timer0 External Clock Source Select bit
If T0CS = 1
The T0XCS bit controls which clock external to the core/Timer0 module supplies Timer0:
1 = Timer0 Clock Source is the capacitive sensing oscillator
0 = Timer0 Clock Source is the T0CKI pin
If T0CS = 0
Timer0 clock source is controlled by the core/Timer0 module and is FOSC/4.

15.3 PWM Mode

The PWM mode generates a Pulse-Width Modulated signal on the CCPx pin. The duty cycle, period and resolution are determined by the following registers:

- PR2
- T2CON
- CCPRxL
- CCPxCON

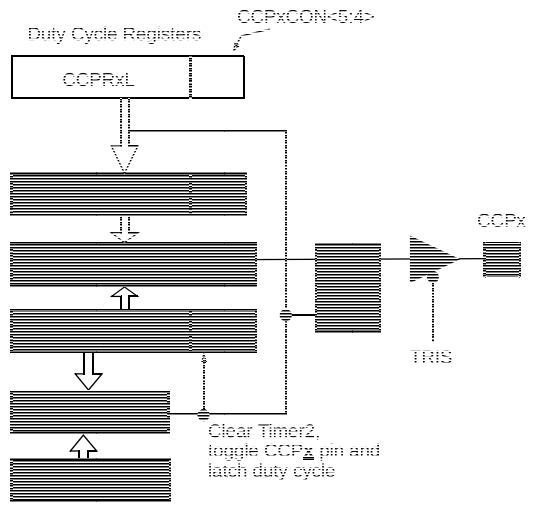
In Pulse-Width Modulation (PWM) mode, the CCP module produces up to a 10-bit resolution PWM output on the CCPx pin.

Figure 15-3 shows a simplified block diagram of PWM operation.

Figure 15-4 shows a typical waveform of the PWM signal.

For a step-by-step procedure on how to set up the CCP module for PWM operation, refer to **Section 15.3.8 "Setup for PWM Operation"**.

FIGURE 15-3: SIMPLIFIED PWM BLOCK DIAGRAM

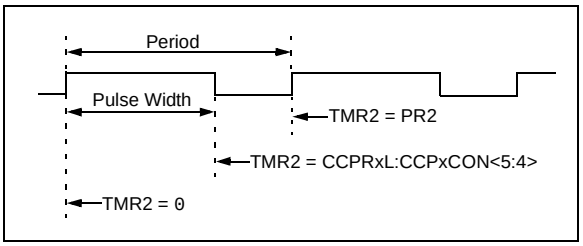


Note 1: The 8-bit timer TMR2 register is concatenated with the 2-bit internal system clock (Fosc), or 2 bits of the prescaler, to create the 10-bit time base.

2: In PWM mode, CCPRxH is a read-only register.

The PWM output (Figure 15-4) has a time base (period) and a time that the output stays high (duty cycle).

FIGURE 15-4: CCP PWM OUTPUT



15.3.1 CCPx PIN CONFIGURATION

In PWM mode, the CCPx pin is multiplexed with the PORT data latch. The user must configure the CCPx pin as an output by clearing the associated TRIS bit.

Either RC1 or RB3 can be selected as the CCP2 pin. Refer to **Section 6.1 "Alternate Pin Function"** for more information.

Note:	Clearing the CCPxCON register will relinquish CCPx control of the CCPx pin.

18.0 PROGRAM MEMORY READ

The Flash program memory is readable during normal operation over the full VDD range of the device. To read data from Program Memory, five Special Function Registers (SFRs) are used:

- PMCON1
- PMDATL
- PMDATH
- PMADRL
- PMADRH

The value written to the PMADRH:PMADRL register pair determines which program memory location is read. The read operation will be initiated by setting the RD bit of the PMCON1 register. The program memory flash controller takes two instructions to complete the read, causing the second instruction after the setting the RD bit will be ignored. To avoid conflict with program execution, it is recommended that the two instructions following the setting of the RD bit are NOP. When the read completes, the result is placed in the PMDATLH:PMDATL register pair. Refer to Example 18-1 for sample code.

Note:	Code-protect does not effect the CPU from performing a read operation on the program memory. For more information, refer to **Section 8.2 “Code Protection”**.

EXAMPLE 18-1: PROGRAM MEMORY READ

```
                BANKSEL  PMADRL          ;
                MOVF     MS_PROG_ADDR, W;
                MOVWF    PMADRH          ;MS Byte of Program Address to read
                MOVF     LS_PROG_ADDR, W;
                MOVWF    PMADRL          ;LS Byte of Program Address to read
                BANKSEL  PMCON1          ;
Required        BSF      PMCON1, RD      ;Initiate Read
Sequence        NOP
                NOP                      ;Any instructions here are ignored as program
                                         ;memory is read in second cycle after BSF
                BANKSEL  PMDATL          ;
                MOVF     PMDATL, W       ;W = LS Byte of Program Memory Read
                MOVWF    LOWPMBYTE       ;
                MOVF     PMDATH, W       ;W = MS Byte of Program Memory Read
                MOVWF    HIGHPMBYTE      ;
```

19.2 Wake-up Using Interrupts

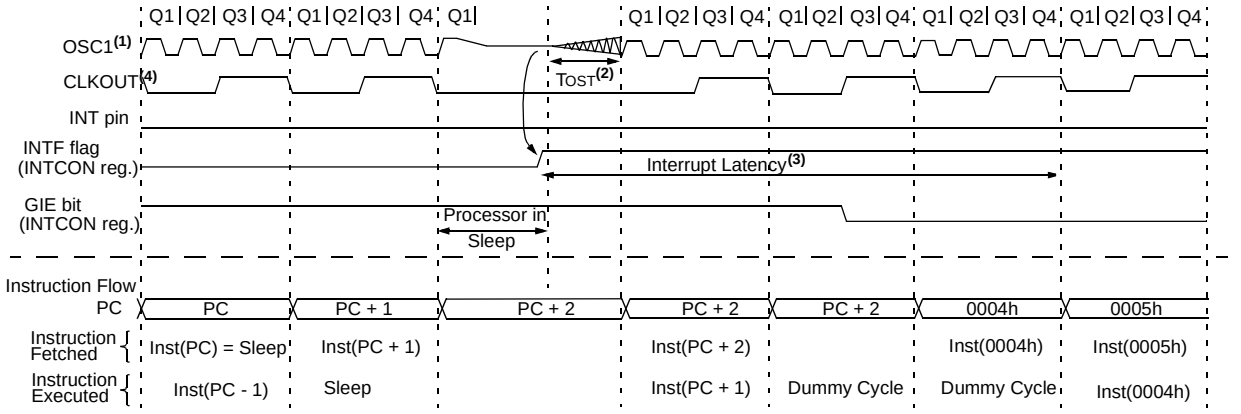
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will not be cleared, the $\overline{TO}$ bit will not be set and the $\overline{PD}$ bit will not be cleared.
- If the interrupt occurs **during or after** the execution of a SLEEP instruction, the device will immediately wake-up from Sleep. The SLEEP instruction will be completely executed before the wake-up. Therefore, the WDT and WDT prescaler and postscaler (if enabled) will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the SLEEP instruction was executed as a NOP.

To ensure that the WDT is cleared, a CLRWDT instruction should be executed before a SLEEP instruction.

FIGURE 19-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: XT, HS or LP Oscillator mode assumed.
2: TOST = 1024 TOSC (drawing not to scale). This delay does not apply to EC and RC Oscillator modes.
3: GIE = 1 assumed. In this case after wake-up, the processor jumps to 0004h. If GIE = 0, execution will continue in-line.
4: CLKOUT is not available in XT, HS, LP or EC Oscillator modes, but shown here for timing reference.

TABLE 19-1: 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	Value on POR, BOR	Value on all other Resets
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0	0000 0000	0000 0000
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 0000	0000 0000
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIE2	—	—	—	—	—	—	—	CCP2IE	---- ---0	---- ---0
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIR2	—	—	—	—	—	—	—	CCP2IF	---- ---0	---- ---0

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used in Power-Down mode.

21.0 INSTRUCTION SET SUMMARY

The PIC16(L)F722/3/4/6/7 instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 21-1, while the various opcode fields are summarized in Table 21-1.

Table 21-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution time of 1 µs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

21.1 Read-Modify-Write Operations

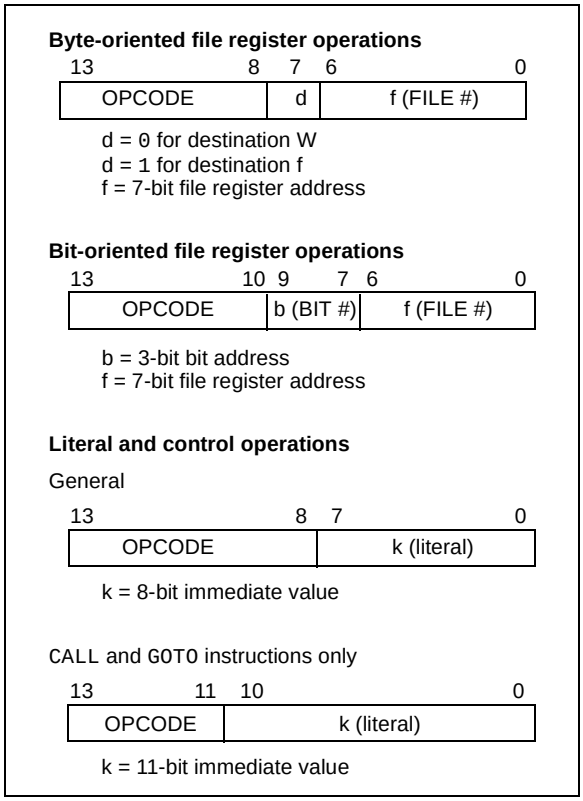
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a CLRF PORTB instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended consequence of clearing the condition that set the RBIF flag.

TABLE 21-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
$\overline{TO}$	Time-out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
$\overline{PD}$	Power-down bit

FIGURE 21-1: GENERAL FORMAT FOR INSTRUCTIONS



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'.

MOVF	Move f
Syntax:	[*label*] MOVF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) → (dest)
Status Affected:	Z
Description:	The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.
Words:	1
Cycles:	1
Example:	MOVF FSR, 0 After Instruction W = value in FSR register Z = 1

MOVWF	Move W to f
Syntax:	[*label*] MOVWF f
Operands:	$0 \le f \le 127$
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from W register to register 'f'.
Words:	1
Cycles:	1
Example:	MOVW OPTION F Before Instruction OPTION = 0xFF W = 0x4F After Instruction OPTION = 0x4F W = 0x4F

MOVLW	Move literal to W
Syntax:	[*label*] MOVLW k
Operands:	$0 \le k \le 255$
Operation:	k → (W)
Status Affected:	None
Description:	The 8-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.
Words:	1
Cycles:	1
Example:	MOVLW 0x5A After Instruction W = 0x5A

NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.
Words:	1
Cycles:	1
Example:	NOP

RETFIE	Return from Interrupt
Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction.
Words:	1
Cycles:	2
Example:	`RETFIE` After Interrupt PC = TOS GIE = 1

RETLW	Return with literal in W
Syntax:	[*label*] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W); TOS → PC
Status Affected:	None
Description:	The W register is loaded with the 8-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.
Words:	1
Cycles:	2

Example:	`CALL TABLE;W contains table`
	`;offset value`
	`TABLE ;W now has table value`
	`•`
	`•`
	`ADDWF PC ;W = offset`
	`RETLW k1 ;Begin table`
	`RETLW k2 ;`
	`•`
	`•`
	`•`
	`RETLW kn ; End of table`
	
	Before Instruction
	W = 0x07
	After Instruction
	W = value of k8

RETURN	Return from Subroutine
Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a 2-cycle instruction.

23.3 DC Characteristics: PIC16(L)F722/3/4/6/7-I/E (Power-Down) (Continued)

PIC16LF722/3/4/6/7			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
PIC16F722/3/4/6/7			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	Device Characteristics	Min.	Typ†	Max. +85°C	Max. +125°C	Units	Conditions	
							VDD	Note
	Power-down Base Current (IPD)(2)							
D027		—	0.06	0.7	5.0	µA	1.8	A/D Current **(Note 1, Note 4)**, no conversion in progress
		—	0.08	1.0	5.5	µA	3.0	
D027		—	6	10.7	18	µA	1.8	A/D Current **(Note 1, Note 4)**, no conversion in progress
		—	7	10.6	20	µA	3.0	
		—	7.2	11.9	22	µA	5.0	
D027A		—	250	400	—	µA	1.8	A/D Current **(Note 1, Note 4)**, conversion in progress
		—	250	400	—	µA	3.0	
D027A		—	280	430	—	µA	1.8	A/D Current **(Note 1, Note 4, Note 5)**, conversion in progress
		—	280	430	—	µA	3.0	
		—	280	430	—	µA	5.0	
D028		—	2.2	3.2	14.4	µA	1.8	Cap Sense Low Power Oscillator mode
		—	3.3	4.4	15.6	µA	3.0	
D028		—	6.5	13	21	µA	1.8	Cap Sense Low Power Oscillator mode
		—	8	14	23	µA	3.0	
		—	8	14	25	µA	5.0	
D028A		—	4.2	6	17	µA	1.8	Cap Sense Medium Power Oscillator mode
		—	6	7	18	µA	3.0	
D028A		—	8.5	15.5	23	µA	1.8	Cap Sense Medium Power Oscillator mode
		—	11	17	24	µA	3.0	
		—	11	18	27	µA	5.0	
D028B		—	12	14	25	µA	1.8	Cap Sense High Power Oscillator mode
		—	32	35	44	µA	3.0	
D028B		—	16	20	31	µA	1.8	Cap Sense High Power Oscillator mode
		—	36	41	50	µA	3.0	
		—	42	49	58	µA	5.0	

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

Note 1: The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.

3: Fixed Voltage Reference is automatically enabled whenever the BOR is enabled

4: A/D oscillator source is FRC

5: 0.1 µF capacitor on VCAP (RA0).

TABLE 23-7: PIC16F722/3/4/6/7 A/D CONVERTER (ADC) CHARACTERISTICS:

Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
AD01	NR	Resolution	—	—	8	bit	
AD02	EIL	Integral Error	—	—	±1.7	LSb	VREF = 3.0V
AD03	EDL	Differential Error	—	—	±1	LSb	No missing codes VREF = 3.0V
AD04	EOFF	Offset Error	—	—	±2.2	LSb	VREF = 3.0V
AD05	EGN	Gain Error	—	—	±1.5	LSb	VREF = 3.0V
AD06	VREF	Reference Voltage(3)	1.8	—	VDD	V	
AD07	VAIN	Full-Scale Range	VSS	—	VREF	V	
AD08	ZAIN	Recommended Impedance of Analog Voltage Source	—	—	50	kΩ	Can go higher if external 0.01µF capacitor is present on input pin.

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

Note 1: Total Absolute Error includes integral, differential, offset and gain errors.

2: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

3: When ADC is off, it will not consume any current other than leakage current. The power-down current specification includes any such leakage from the ADC module.

TABLE 23-8: PIC16F722/3/4/6/7 A/D CONVERSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
AD130*	TAD	A/D Clock Period	1.0	—	9.0	µs	TOSC-based
		A/D Internal RC Oscillator Period	1.0	2.0	6.0	µs	ADCS<1:0> = 11 (ADRC mode)
AD131	TCNV	Conversion Time (not including Acquisition Time)(1)	—	10.5	—	TAD	Set GO/$\overline{\text{DONE}}$ bit to conversion complete
AD132*	TACQ	Acquisition Time	—	1.0	—	µs	

* These parameters are characterized but not tested.

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

Note 1: The ADRES register may be read on the following TCY cycle.

FIGURE 24-7: PIC16F722/3/4/6/7 TYPICAL IDD vs. VDD OVER FOSC, EXTRC MODE, VCAP = 0.1 µF

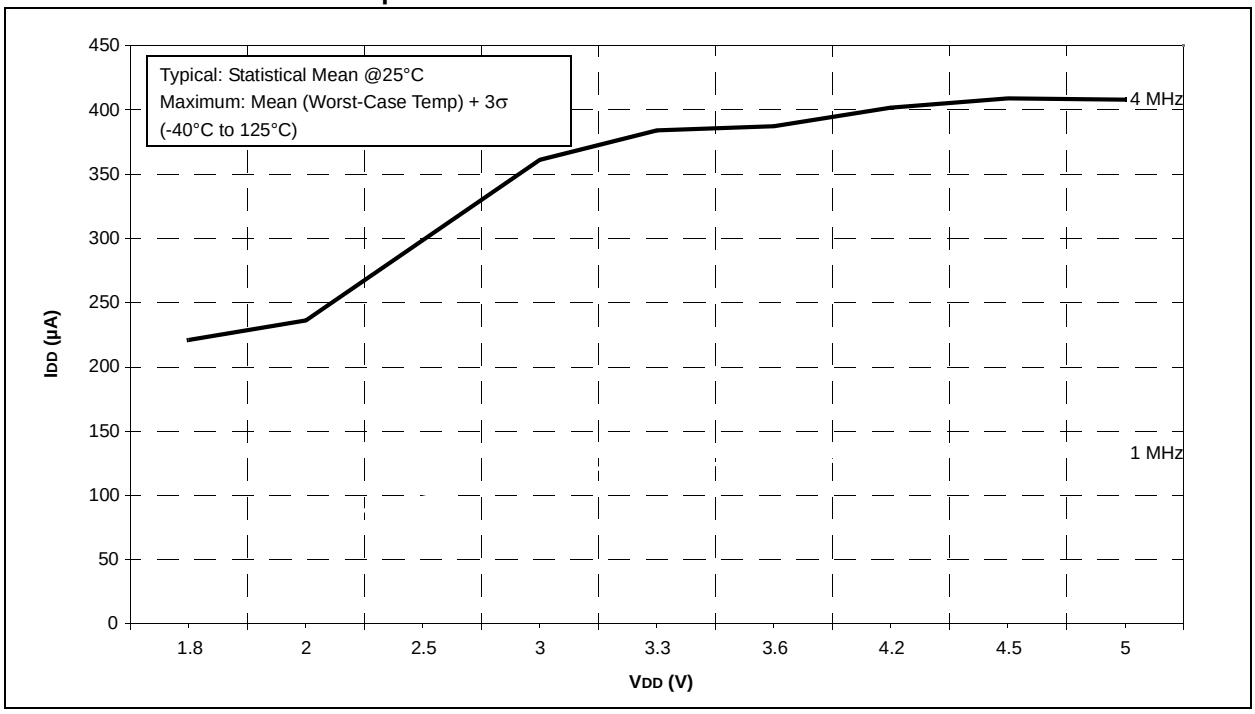


FIGURE 24-8: PIC16LF722/3/4/6/7 TYPICAL IDD vs. VDD OVER FOSC, EXTRC MODE

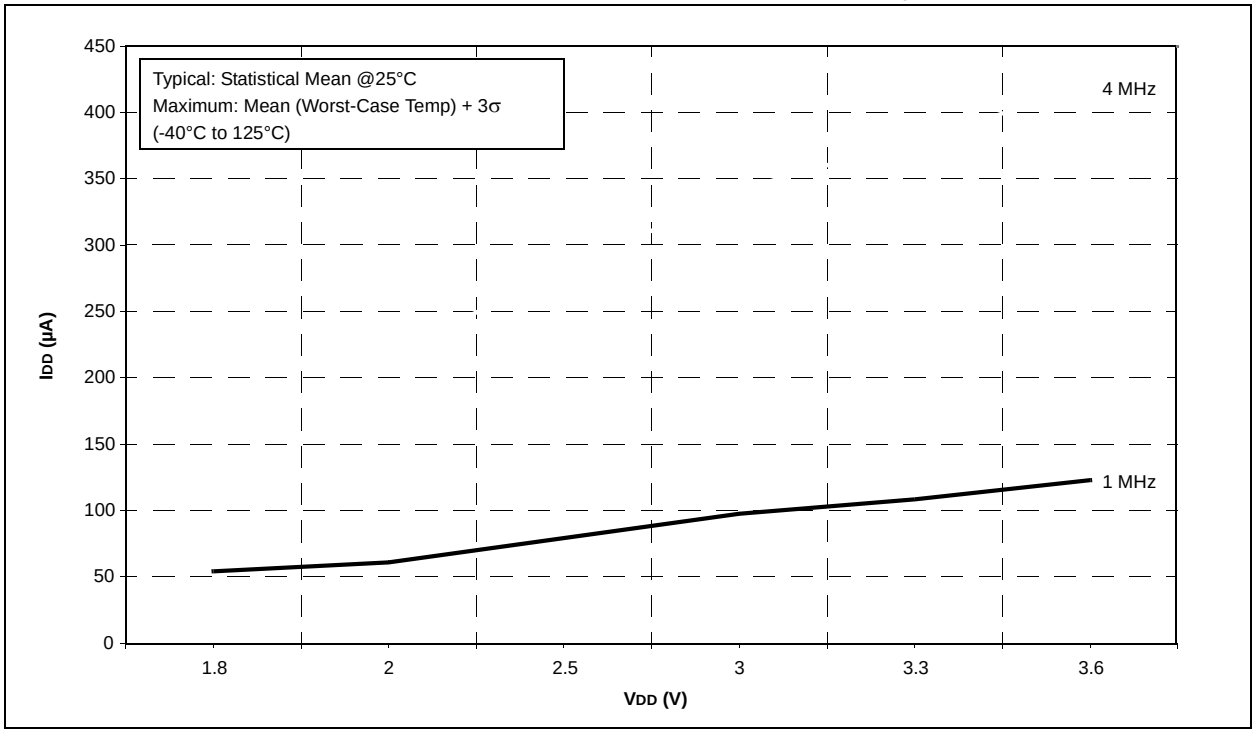


FIGURE 24-23: PIC16F722/3/4/6/7 TYPICAL IDD vs. FOSC OVER VDD, INTOSC MODE, VCAP =0.1µF

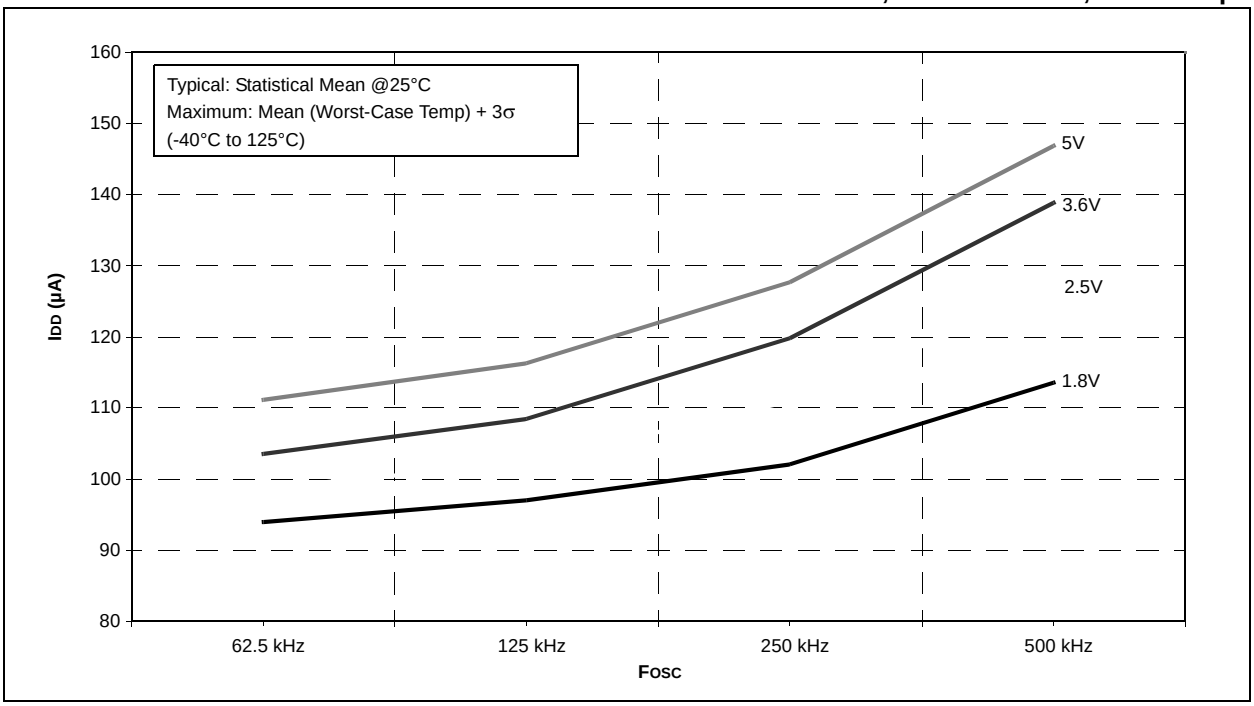


FIGURE 24-24: PIC16LF722/3/4/6/7 TYPICAL IDD vs. FOSC OVER VDD, INTOSC MODE

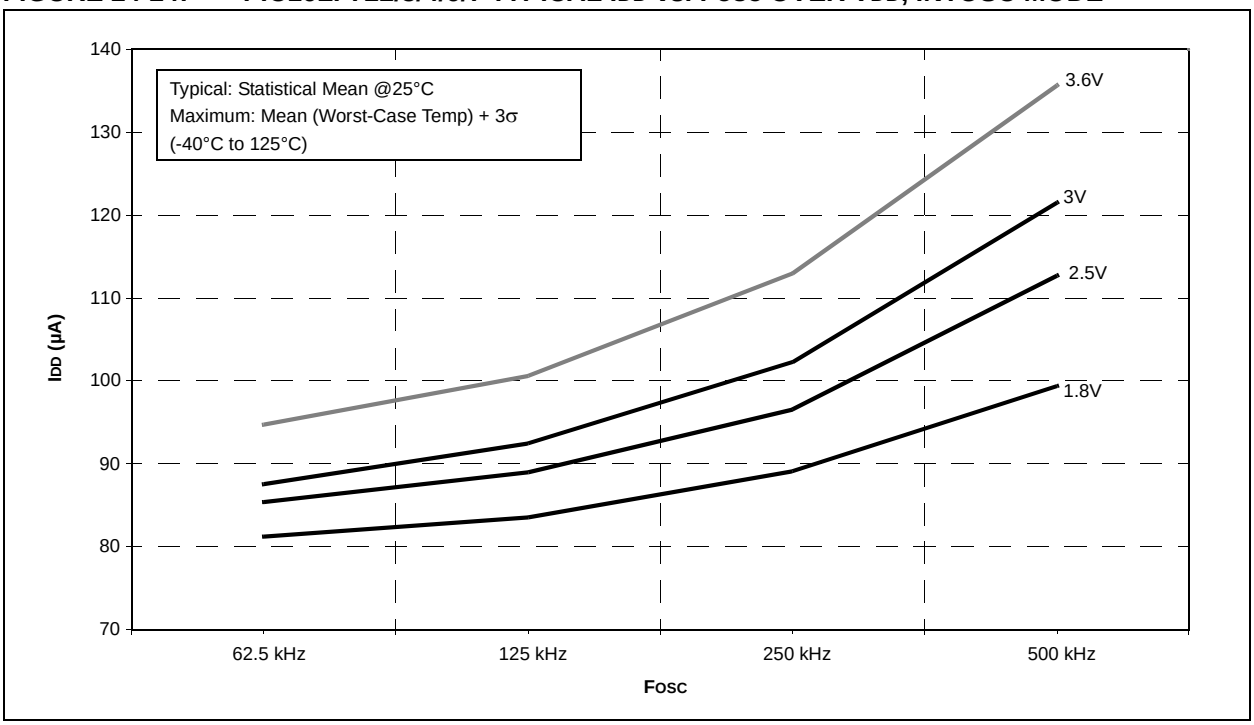


FIGURE 24-57: V_{OL} vs. I_{OL} OVER TEMPERATURE, V_{DD} = 1.8V

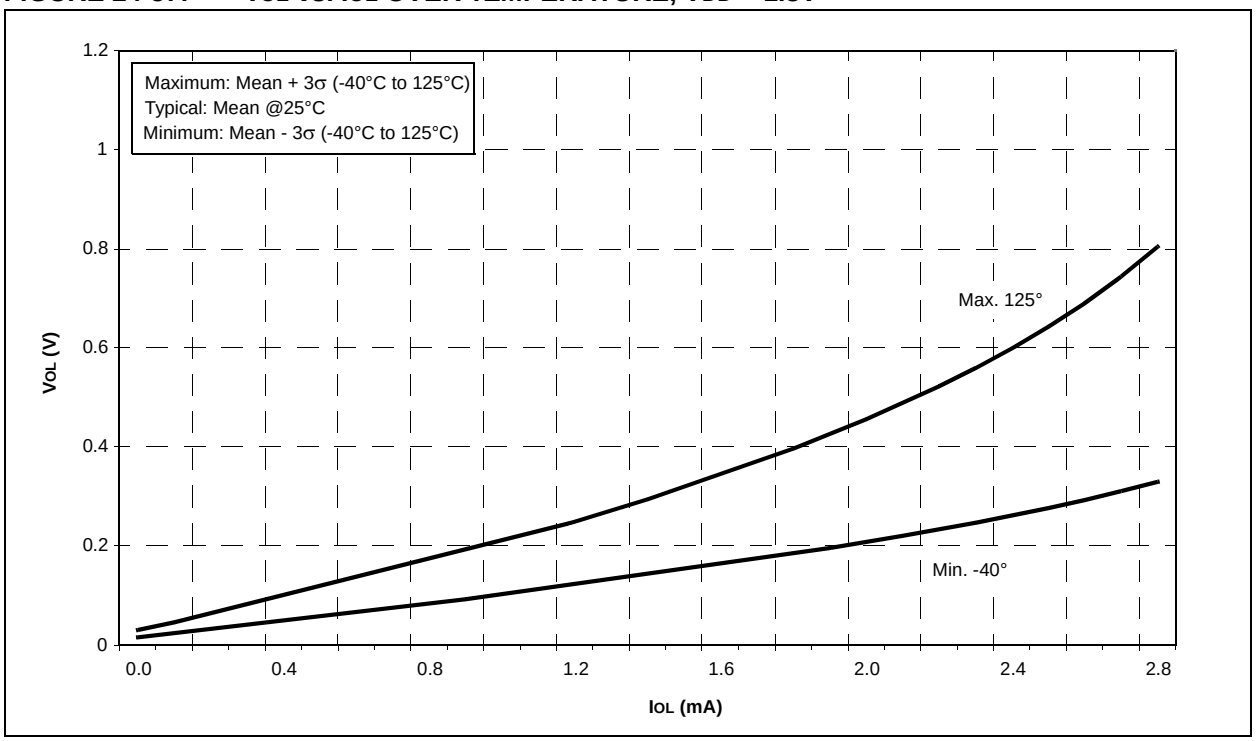


FIGURE 24-58: PIC16F722/3/4/6/7 PWRT PERIOD

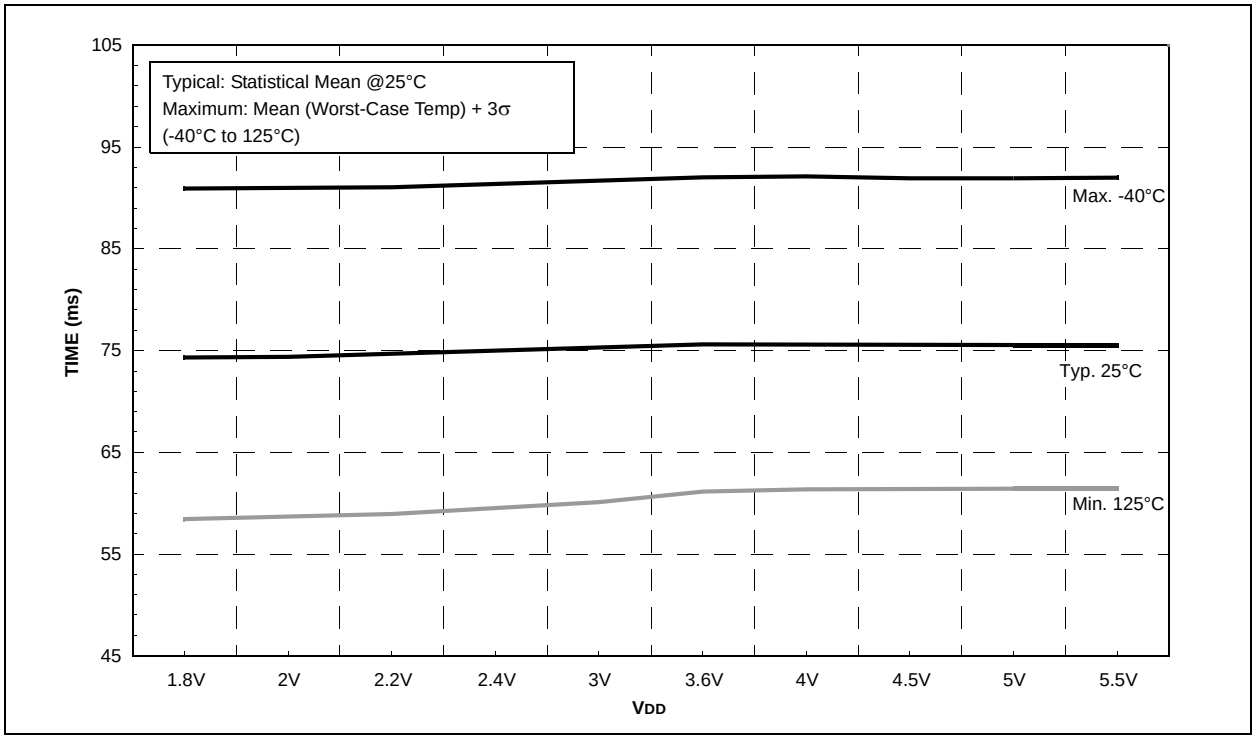


FIGURE 24-61: PIC16F722/3/4/6/7 A/D INTERNAL RC OSCILLATOR PERIOD

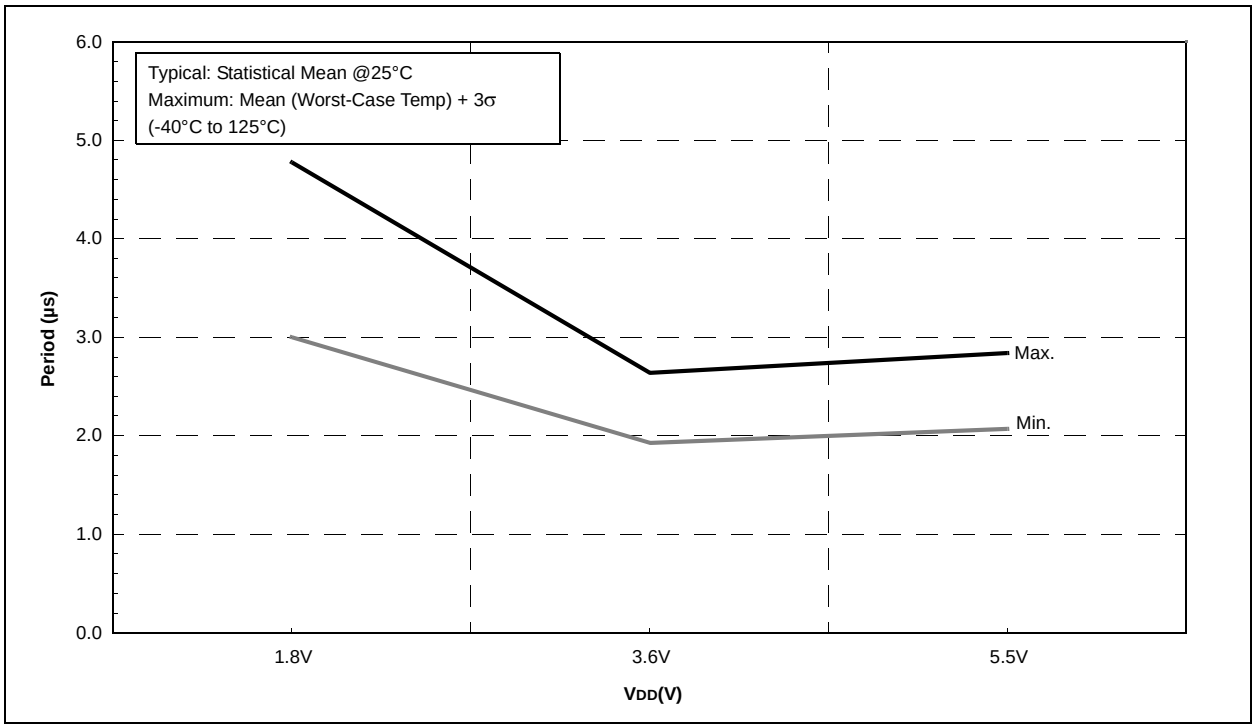


FIGURE 24-62: PIC16F722/3/4/6/7 CAP SENSE OUTPUT CURRENT, POWER MODE = HIGH