PIC INSTRUCTION SET

These are the required symbols that are used with instruction set.

- -f file: It represents the used register. Used on byte oriented instructions.
- -d destination: If this variable is 0, the result will be stored on W register. If this variable is 1, the result will be stored on desired file.
- -b bit: It represents the bit on the register which is used on the process.
- -k literal: It is the constant value that will be treated with W register.

Every instruction is 14 bits long and they are separated into two parts. OPCODE defines the instruction type; second part is OPERAND and it can be constituted from one or more words. All instructions are executed in one instruction cycle except jump (or go to) instructions. In 4 MHz operating frequency, one instruction cycle is 1 μs.

ADDLW Add Literal to W

Usage : ADDLW k
Code : 11 111x kkkk kkkk
Process : W + k → W
Period : 1
Status Bit : C, DC, Z
Description : k literal is summed with W register and the result is stored in W register.

ADDWF Add W to File

Usage : ADDWF f, d
Code : 00 0111 dfff ffff
Process : W + f → d
Period : 1
Status Bit : C, DC, Z
Description : W register and f register is summed and the result is stored in d.

ANDLW AND Literal with W

Usage : ANDLW k
Code : 11 1001 kkkk kkkk
Process : (W AND K) → W
Period : 1
Status Bit : Z
Description : The content of W register and k value are processed with AND logic. The result is stored in W register.

ANDWF AND W with File

Usage : ANDWF f, d
Code : 00 0101 dfff ffff
Process : (W AND f) → d
Period : 1
Status Bit : Z
Description : The content of W register and f register are processed with AND logic. The result is stored in d.

BCF Bit Clear File

Usage : BCF f, b
Code : 01 00bb bfff ffff
Process : 0 → f (b)
Period : 1
Status Bit : None
Description : The 'b' bit in the f register is reset.

BSF Bit Set File

Usage : BSF f, b
Code : 01 01bb bfff ffff
Process : 1 → f (b)
Period : 1
Status Bit : None
Description : The 'b' bit in the f register is set.

BTFSC Bit Test File Skip If Clear

Usage : BTFSC f, b
Code : 01 10bb bfff ffff
Process : If f (b) = 0 jump to next line
Period : 1 (2)
Status Bit : None
Description : If the related bit in file register is 0, jump to next instruction. If it jumps, the instruction cycle will be 2 periods.

BTFSS Bit Test File Skip If Set

Usage : BTFSS f, b
Code : 01 11bb bfff ffff
Process : If f (b) = 1 jump to next line
Period : 1 (2)
Status Bit : None
Description : If the related bit in file register is 1, jump to next instruction. If it jumps, the instruction cycle will be 2 periods.

CALL Call a sub-program

Usage : CALL k
Code : 10 0kkk kkkk kkkk
Process : PC + 1→ Stack, k (address) → PC
Period : 2
Status Bit : None
Description : Calls a sub-program named with 'k'. Return address is stored in stack.

CLRF Clear File

Usage : CLRF f
Code : 00 0001 1fff ffff
Process : 00h → f, 1 → Z
Period : 1
Status Bit : Z
Description : F register is cleared and Z flag is set to 1.

CLRW Clear W

Usage : CLRW
Code : 00 0001 0xxx xxxx
Process : 00h → W, 1 → Z
Period : 1
Status Bit : Z
Description : W register is cleared and Z flag is set to 1.

CLRWDT Clear Watchdog Timer

Usage : CLRWDT
Code : 00 0000 0110 0100
Process : 00h → WDT, 1 → TO, 1 → PD
Period : 1
Status Bit : TO, PD
Description : Watchdog timer is reset. Moreover, WDT prescaler is reset. TO and PD bits of Status register is set.

COMF Complement File

Usage : COMF f, d
Code : 00 1001 dfff ffff
Process : (f)' → d
Period : 1
Status Bit : Z
Description : Calculate the complement of f register. The result is stored in d.

DECF Decrement File

Usage : DECF f, d
Code : 00 0011 dfff ffff
Process : f - 1 → d
Period : 1
Status Bit : Z
Description : Decrease the value of f register by 1. The result is stored in d.

DECFSZ Decrement File Skip If Zero

Usage : DECFSZ f, d
Code : 00 1011 dfff ffff
Process : If (f – 1) = 0 jump to next line, result → d
Period : 1 (2)
Status Bit : None
Description : Decrease the value of f register by 1. If the result is zero, jump to next instruction. The result is stored in d. If it jumps, the instruction cycle will be 2 periods.

GOTO Go to a label (address)

Usage : GOTO k
Code : 10 1kkk kkkk kkkk
Process : k → PC
Period : 2
Status Bit : None
Description : Jumps to the label named with 'k'. The content of PC changes.

INCF Increment File

Usage : INCF f, d
Code : 00 1010 dfff ffff
Process : f + 1 → d
Period : 1
Status Bit : Z
Description : Increase the value of f register by 1. The result is stored in d.

INCFSZ Increment File Skip If Zero

Usage : INCFSZ f, d
Code : 00 1111 dfff ffff
Process : If (f + 1) = 0 jump to next line, result → d
Period : 1 (2)
Status Bit : None
Description : Increase the value of f register by 1. If the result is zero, jump to next instruction. The result is stored in d. If it jumps, the instruction cycle will be 2 periods.

IORLW OR Literal with W

Usage : IORW k
Code : 11 1000 kkkk kkkk
Process : (W OR k) → W
Period : 1
Status Bit : Z
Description : The content of W register and k value are processed with OR logic. The result is stored in W register.

IORWF OR W with File

Usage : IORWF f, d
Code : 00 0100 dfff ffff
Process : (W OR k) → d
Period : 1
Status Bit : Z
Description : The content of W register and f register are processed with OR logic. The result is stored in d.

MOVLW Move Literal to W

Usage : MOVLW k
Code : 11 00xx kkkk kkkk
Process : k → W
Period : 1
Status Bit : None
Description : Move k literal to W register.

MOVF Move File

Usage : MOVF f, d
Code : 00 1000 dfff ffff
Process : f → d
Period : 1
Status Bit : Z
Description : Move the content of f register to destination.

MOVWF Move W to File

Usage : MOVWF f
Code : 00 0000 1fff ffff
Process : W → f
Period : 1
Status Bit : None
Description : Move the content of W register to f register.

NOP **No Operation**

Usage : NOP
Code : 00 0000 0xx0 0000
Process : None
Period : 1
Status Bit : None
Description : Wait for 1 instruction cycle

RETFIE **Return From Interrupt**

Usage : RETFIE
Code : 00 0000 0000 1001
Process : Stack → PC, 1 → GIE
Period : 2
Status Bit : None
Description : The content of stack is loaded to PC. GIE bit in INTCON register is set. Program returns to the line when it has been called before.

RETLW **Return Literal with W**

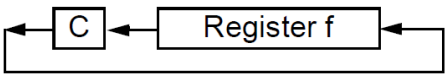
Usage : RETLW k
Code : 11 01xx kkkk kkkk
Process : k → W, Stack → PC
Period : 2
Status Bit : None
Description : W register is loaded with constant literal k. Top value (return address) on stack is loaded to PC.

RETURN **Return from a sub-program**

Usage : RETURN
Code : 00 0000 0000 1000
Process : Stack → PC
Period : 2
Status Bit : None
Description : Returns from a sub-program. Top value on stack is loaded to PC.

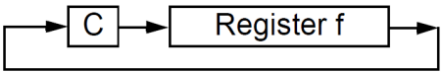
RLF **Rotate Left File**

Usage : RLF f, d
Code : 00 1101 dfff ffff
Process : ----------
Period : 1
Status Bit : C
Description : The content of f register is rotated one bit to the left through the carry flag. The result is stored in d.



RRF Rotate Right File

Usage : RRF f, d
Code : 00 1100 dfff ffff
Process : ----------
Period : 1
Status Bit : C
Description : The content of f register is rotated one bit to the right through the carry flag. The result is stored in d.



SLEEP Power-Down Mode

Usage : SLEEP
Code : 00 0000 0110 0011
Process : 00h → WDT, 0 → WDT prescaler, 1 → TO, 0→PD
Period : 1
Status Bit : TO, PD
Description : PD bit is cleared. TO bit is set. WDT and its prescaler are cleared. The processor is put into sleep mode with the oscillator stopped.

SUBLW Subtract W from Literal

Usage : SUBLW k
Code : 11 110x kkkk kkkk
Process : k – W → W
Period : 1
Status Bit : C, DC, Z
Description : The content of W register is subtracted from k literal. The result is stored in W register.

SUBWF Subtract W from File

Usage : SUBWF f, d
Code : 00 0010 dfff ffff
Process : f – W → d
Period : 1
Status Bit : C, DC, Z
Description : The content of f register is subtracted from W register. The result is stored in d.

SWAPF Swap File

Usage : SWAPF f, d
Code : 00 1110 dfff ffff
Process : f <3:0> → d <7:4>, f <7:4> → d <3:0>
Period : 1
Status Bit : None
Description : The lower 4 bits and higher 4 bits of f register is swapped. The result is stored in d.

XORLW XOR Literal with W

Usage : XORLW k
Code : 11 1010 kkkk kkkk
Process : (W XOR k) → W
Period : 1
Status Bit : Z
Description : The content of W register and k value are processed with XOR logic. The result is stored in W register.

XORWF XOR W with File

Usage : XORWF f,d
Code : 00 0110 dfff ffff
Process : (W XOR f) → d
Period : 1
Status Bit : Z
Description : The content of W register and f register are processed with XOR logic. The result is stored in d.

PIC16F877A INSTRUCTION SET SUMMARY

Mnemonic, Operands		Description	Cycles	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS					
ADDWF	f, d	Add W and f	1	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	Z	1,2
CLRF	f	Clear f	1	Z	2
CLRW	-	Clear W	1	Z	
COMF	f, d	Complement f	1	Z	1,2
DECF	f, d	Decrement f	1	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)		1,2,3
INCF	f, d	Increment f	1	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)		1,2,3
IORWF	f, d	Inclusive OR W with f	1	Z	1,2
MOVF	f, d	Move f	1	Z	1,2
MOVWF	f	Move W to f	1		
NOP	-	No Operation	1		
RLF	f, d	Rotate Left f through Carry	1	C	1,2
RRF	f, d	Rotate Right f through Carry	1	C	1,2
SUBWF	f, d	Subtract W from f	1	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1		1,2
XORWF	f, d	Exclusive OR W with f	1	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS					
BCF	f, b	Bit Clear f	1		1,2
BSF	f, b	Bit Set f	1		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)		3
LITERAL AND CONTROL OPERATIONS					
ADDLW	k	Add literal and W	1	C,DC,Z	
ANDLW	k	AND literal with W	1	Z	
CALL	k	Call subroutine	2		
CLRWDT	-	Clear Watchdog Timer	1	TO,PD	
GOTO	k	Go to address	2		
IORLW	k	Inclusive OR literal with W	1	Z	
MOVLW	k	Move literal to W	1		
RETFIE	-	Return from interrupt	2		
RETLW	k	Return with literal in W	2		
RETURN	-	Return from Subroutine	2		
SLEEP	-	Go into standby mode	1	TO,PD	
SUBLW	k	Subtract W from literal	1	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	Z	

1 - When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2 - If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.

3 - If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.