Chapter 3: Instruction Sets

Instruction sets could be said of programmer's interface to hardware. The CPU of the PIC is responsible for using the instructions (or program code) stored in the program memory to execute the functions and operations the instructions intend to do. The instructions are stored in the program memory in a format of machine code, or hex code. Assembly language is the instruction mnemonics for the machine codes. Assembler generates machine codes from Assembly code.

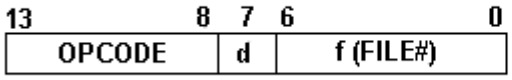
1. PIC16F877 Instruction

Each PIC16F877 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. There are three addressing modes in the 16F877 instruction: byte-oriented, bit-oriented, and literal and control operations. Byte-oriented instructions operate with a whole-byte data, like moving a byte of data in **W** register to a file register. Bit-oriented instructions are to check or change only a bit of a byte data. Literal operations involve with direct numerical value of loading or logical operation with **W** register. The literal operation is usually called 'an immediate addressing mode' in the more traditional microprocessor instructions.

Byte-oriented instructions

For byte-oriented instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator is used to specify which one of the file registers is to be used by the instruction. The destination designator specifies where the result of the operation is to be placed. If 'd' is '0', the result is placed in the **W** register. If 'd' is '1', the result is placed in the file register specified in the instruction. If we do not specify the destination, it is considered '1' (default value).

In the machine code level (in 14-bit word configuration), byte-oriented instructions are configured with 6 bits of Opcode, 1 bit for destination designator, followed by 7 bit file register address.



d=0 for desitination W
d=1 for destination f (default)
f= 7-bit file register address

The table below lists the instructions words of the byte-oriented operation. As we see at the last column of the table, the 6-bit Opcode portion is already given with a specific bit formation for each instruction mnemonic. The remaining 8 bits are determined by the destination of the operation and the file register the operation accesses to do the operation. The column with 'T' indicate the number instruction cycles needed to do the operation of each instruction.

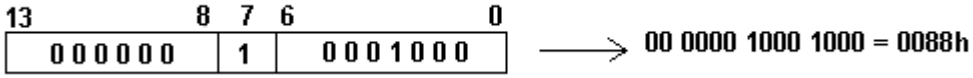
Table. Byte-oriented operation of 16F877 instructions.

Mnemonic	Description	T	Flag	Instruction word (OPCODE+operand)
addwf f,d	add W and f	1	C, Z	00 0111 dfff ffff
andwf f,d	and W with f	1	Z	00 0101 dfff ffff
clrf f	clear f (i.e., f=0)	1	Z	00 0001 1fff ffff
clrw	clear W	1	Z	00 0001 0xxx xxxx
comf f,d	complement f	1	Z	00 1001 dfff ffff

decf f,d	decrease f by 1	1	Z	00 0011 dfff ffff
decfsz f,d	decrease f by 1, skip if f=0	1(2)		00 1011 dfff ffff
incf f,d	increase f by 1	1	Z	00 1010 dfff ffff
incfsz f,d	increase f by 1, skip if f=0	1(2)		00 1111 dfff ffff
iorwf f,d	OR W with f	1	Z	00 0100 dfff ffff
movf f,d	move f	1	Z	00 1000 dfff ffff
movwf f	move W to f	1		00 0000 1fff ffff
nop	no operation	1		00 0000 0xx0 0000
rlf f,d	rotate left f through carry	1	C	00 1101 dfff ffff
rrf f,d	rotate right f through carry	1	C	00 1100 dfff ffff
subwf f,d	subtract W from f (i.e., f-W)	1	C, Z	00 0010 dfff ffff
swapf f,d	swap nibbles in f	1		00 1110 dfff ffff
xorwf f,d	XOR W with f	1	Z	00 0110 dfff ffff

Let's have an example of machine code generation from an instruction. Let's consider then the following instruction:
`movwf PORTD,` which moves a content in W register to PORTD register.

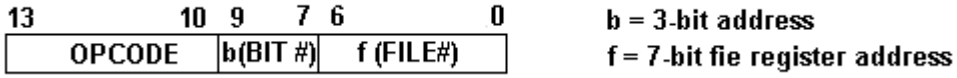
Since the OPCODE for `movwf` is set 000000 (see the byte-oriented operation table above), the destination is the default value of '1', and the file register address of PORTD is 0x08 (from the file register table in page 7), the corresponding machine code is 0088h:



If the file register is changed to PORTB, since the file register address of PORTB is 0x06, the corresponding machine code would be: 0086h.

Bit-oriented instructions

For bit-oriented instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the file register in which the bit is located. In the machine code level (in 14-bit word configuration), bit-oriented instructions are configured with 4 bits of Opcode, 3 bits for the bit number (between 000b and 111b), followed by 7 bit file register address.



The table below lists the instructions words of the bit-oriented operations. As before, at the last column of the table, the 4-bit Opcode portion is already given with a specific bit formation for each instruction mnemonic. The remaining 10 bits are determined by the 3-bit bit number and the 7-bit file register the operation accesses to do the operation.

Table. Bit-Oriented Instruction for 16F877

Mnemonic	Description	T	Flag	Instruction word (OPCODE+operand)
bcf f,b	clear f<b> bit (i.e., f<b)=0)	1		01 00bb bfff ffff
bsf f,b	set f<b> bit (i.e., f<b)=1)	1		01 01bb bfff ffff
btfsc f,b	test f<b> bit, skip if f<b>=0	1(2)		01 10bb bfff ffff
btfss f,b	test f<b> bit, skip if f<b>=1	1(2)		01 11bb bfff ffff