Instruction Set:

It has been already mentioned that microcontrollers differs from other integrated circuits. Most of them are ready for installation into the target device just as they are, this is not the case with the microcontrollers. In order that the microcontroller may operate, it needs precise instructions on what to do. In other words, a program that the microcontroller should execute must be written and loaded into the microcontroller. This chapter covers the commands which the microcontroller "understands". The instruction set for the 16FXX includes 35 instructions in total. Such a small number of instructions is specific to the RISC microcontroller because they are well-optimized from the aspect of operating speed, simplicity in architecture and code compactness. The only disadvantage of RISC architecture is that the programmer is expected to cope with these instructions.

INSTRUCTION	**DESCRIPTION**	**OPERATION**	**FLAG**	**CLK**	*****
Data Transfer Instructions					
MOVLW k	Move constant to W	k -> w		1	
MOVWF f	Move W to f	W -> f		1	
MOVF f,d	Move f to d	f -> d	Z	1	1, 2
CLRW	Clear W	0 -> W	Z	1	
CLRF f	Clear f	0 -> f	Z	1	2
SWAPF f,d	Swap nibbles in f	f(7:4),(3:0) -> f(3:0),(7:4)		1	1, 2
Arithmetic-logic Instructions					
ADDLW k	Add W and	W+k -> W	C, DC, Z	1	

	constant				
ADDWF f,d	Add W and f	W+f -> d	C, DC ,Z	1	1, 2
SUBLW k	Subtract W from constant	k-W -> W	C, DC, Z	1	
SUBWF f,d	Subtract W from f	f-W -> d	C, DC, Z	1	1, 2
ANDLW k	Logical AND with W with constant	W AND k -> W	Z	1	
ANDWF f,d	Logical AND with W with f	W AND f -> d	Z	1	1, 2
ANDWF f,d	Logical AND with W with f	W AND f -> d	Z	1	1, 2
IORLW k	Logical OR with W with constant	W OR k -> W	Z	1	
IORWF f,d	Logical OR with W with f	W OR f -> d	Z	1	1, 2
XORLW k	Logical exclusive OR with W with constant	W XOR k -> W	Z	1	1, 2
XORWF f,d	Logical exclusive OR with W with f	W XOR f -> d	Z	1	
INCF f,d	Increment f by 1	f+1 -> f	Z	1	1, 2
DECF f,d	Decrement f by 1	f-1 -> f	Z	1	1, 2
RLF f,d	Rotate left f		C	1	1, 2

	through CARRY bit				
RRF f,d	Rotate right f through CARRY bit		C	1	1, 2
COMF f,d	Complement f	f -> d	Z	1	1, 2
Bit-oriented Instructions					
BCF f,b	Clear bit b in f	0 -> f(b)		1	1,2
BSF f,b	Set bit b in f	1 -> f(b)		1	1,2
Program Control Instructions					
BTFSC f,b	Test bit b of f. Skip the following instruction if clear.	Skip if f(b) = 0		1 (2)	3
BTFSS f,b	Test bit b of f. Skip the following instruction if set.	Skip if f(b) = 1		1 (2)	3
DECFSZ f,d	Decrement f. Skip the following instruction if clear.	f-1 -> d skip if Z = 1		1 (2)	1, 2, 3
INCFSZ f,d	Increment f. Skip the following instruction if set.	f+1 -> d skip if Z = 0		1 (2)	1, 2, 3
GOTO k	Go to address	k -> PC		2	
CALL k	Call subroutine	PC -> TOS, k -> PC		2	

RETURN	Return from subroutine	TOS -> PC		2	
RETLW k	Return with constant in W	k -> W, TOS -> PC		2	
RETFIE	Return from interrupt	TOS -> PC, 1 -> GIE		2	
Other instructions					
NOP	No operation	TOS -> PC, 1 -> GIE		1	
CLRWDT	Clear watchdog timer	0 -> WDT, 1 -> TO, 1 -> PD	TO, PD	1	
SLEEP	Go into sleep mode	0 -> WDT, 1 -> TO, 0 -> PD	TO, PD	1	

Table 9-1 16Fxx Instruction Set
*1 When an I/O register is modified as a function of itself, the value used will be that value present on the pins themselves. *2 If the instruction is executed on the TMR register and if d=1, the prescaler will be cleared. *3 If the PC is modified or test result is logic one (1), the instruction requires two cycles.

Data Transfer Instructions

Data Transfer within the microcontroller takes place between working register W (called accumulator) and a register which represents any location of internal RAM regardless of whether it is about special function or general purpose registers. First three instructions move literal to W register (MOVLW stands for **move Literal to W**), move data from W register to RAM and from RAM to W register (or to the same RAM location with change on flag Z only). Instruction CLRF clears f register, whereas CLRW clears W register. SWAPF instruction swaps nibbles within f register (one nibble contains four bits).

Arithmetic-logic Instructions

Similar to most microcontrollers, PIC supports only two arithmetic instructions- addition and subtraction. Flags C, DC, Z are automatically set depending on the results of addition or subtraction. The only exception is the flag C. Since subtraction is performed as addition with negative value, the flag C is inverted after subtraction. It means that the flag C is set if it is possible to perform operation and cleared if the larger number is subtracted from smaller one. Logic one (1) of the PIC is able to perform operations AND, OR, EX-OR, inverting (COMF) and rotation (RLF and RRF). Instructions which rotate a register actually rotate its bits through the flag C by one bit left (toward bit 7) or right (toward bit 0). The bit shifted from the register is moved to the flag C which is automatically moved to the bit on the opposite side of the register.

Bit-oriented Instructions

Instructions BCF and BSF clear or set any bit in memory. Although it seems to be a simple operation, it is not like that. CPU first reads the entire byte, changes one its bit and rewrites the whole byte to the same location.

Program Control Instructions

The PIC16F887 executes instructions GOTO, CALL, RETURN in the same way as all other microcontrollers do. A difference is that stack is independent from internal RAM and has 8 levels. The ‘RETLW k’ instruction is identical to RETURN instruction, with exception that a constant defined by instruction operand is written to the W register prior to return from subroutine. This instruction enables **Lookup** tables to be easily created by creating a table as a subroutine consisting of ‘RETLWk‘ instructions, where the literals ‘k’ belong to the table. The next step is to write the position of the literals k (0, 1, 2, 3...n) to W register and call the subroutine (table) using the CALL instruction. Table below consists of the following literals: k0, k1, k2...kn.