



SNS COLLEGE OF ENGINEERING

Coimbatore-35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

III MCT / V SEMESTER

UNIT 4 –PIC MICROCONTROLLER

TOPIC 1 –Instruction Set of PIC microcontroller



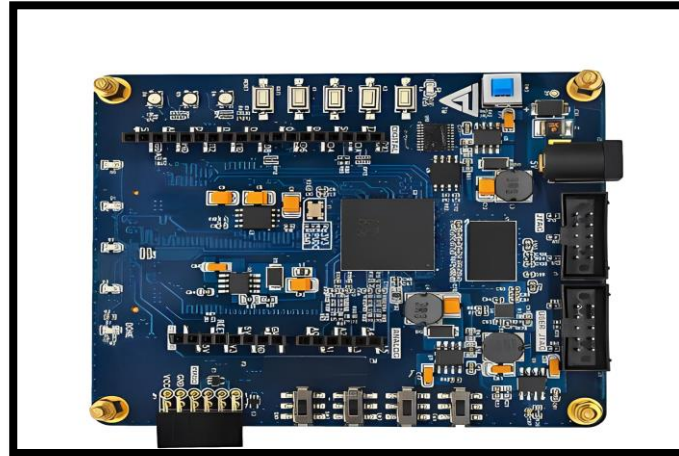
SNS COLLEGE OF ENGINEERING

(Autonomous)

DEPARTMENT OF MECHANICAL ENGINEERING



EMBEDDED SYSTEM FOR MECHATRONICS



Guess Today's Topic????



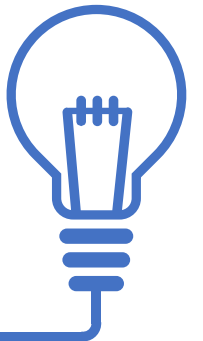


INSTRUCTION SET OF PIC MICROCONTROLLER





What is PIC Microcontroller ?





PIC Microcontroller:

PIC microcontrollers (Programmable Interface Controllers), are electronic circuits that can be programmed to carry out a vast range of tasks. They can be programmed to be timers or to control a production line and much more.





Instruction set of PIC Microcontroller

The PIC instruction set refers to the set of instructions that Microchip Technology PIC or ds PIC microcontroller supports. The instructions are usually programmed into the Flash memory of the processor, and automatically executed by the microcontroller on startup.





Types of Instruction set of PIC Microcontroller

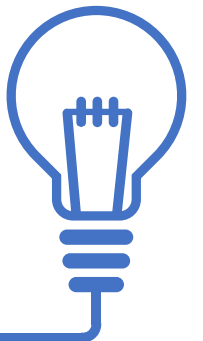
- Instruction Execution Time
- Program Control Instructions
- Bit-oriented Instructions
- Arithmetic-logic Instructions
- Data Transfer Instructions





Instruction Execution Time :

All instructions are single-cycle instructions. The only exception may be conditional branch instructions (if condition is met) or instructions being executed upon the program counter. In both cases, two cycles are required for instruction execution where the second cycle is executed as a NOP (**No Operation**). A single-cycle instruction consists of four clock cycles. If 4MHz oscillator is used, a nominal time for instruction execution is $1\mu\text{S}$. In case of jump, the instruction execution time is $2\mu\text{S}$.





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





Instruction Set in PIC16Cxx MC Family

- Complete set: 35 instructions.
- MC Architecture: RISC microcontroller.
- Instruction Types:
 1. **Data Processing Operations:**
 - Copy data between registers.
 - Manipulate data in a single register.
 - Arithmetic operations.
 - Logic operations.
 2. **Program Sequence Control Operations:**
 - Unconditional Jump.
 - Conditional Jump.
 - Call.
 - Control.



Word list

f any memory location in a microcontroller

W work register

b bit position in 'f' register

d destination bit

label group of eight characters which marks the beginning of a part of the program

TOS top of stack

[] option

< > bit position inside register

Example:

008C	movwf 0C
-------------	-----------------



Data transfer

Transfer of data in a MC is done between W register and an 'f' register.

Mnemonic	Description	Operation	Flag	Cycle	Notes
Data transfer					
MOVLW	k	Move constant to W	$k \rightarrow W$	1	
MOVWF	f	Move W to f	$W \rightarrow f$	1	
MOVF	f, d	Move f	$f \rightarrow d$	Z	1,2
CLRWF	-	Clear W	$0 \rightarrow W$	Z	1
CLRF	f	Clear f	$0 \rightarrow f$	Z	2
SWAPF	f, d	Swap nibbles in f	$f(7:4), (3:0) \rightarrow f(3:0), (7:4)$	1	1,2

These instructions provide for:

- a constant being written in W register (MOVLW)
- data to be copied from W register onto RAM.
- data from RAM to be copied onto W register (or on the same RAM location, at which point only the status of Z flag changes).
- Instruction CLRF writes constant 0 in 'f' register,
- Instruction CLRWF writes constant 0 in register W.
- SWAPF instruction exchanges places of the 4-bit nibbles field inside a register.





Arithmetic and logic



PIC like most MCs supports only subtraction and addition.

Flags C, DC and Z are set depending on a result of addition or subtraction.

Logic unit performs AND, OR, EX-OR, complement (COMF) and rotation (RLF & RRF).

Arithmetic and logic						
ADDLW	k	Add constant and W	$W+1 \rightarrow W$	C,DC,Z	1	
ADDWF	f, d	Add W and f	$W+f \rightarrow d$	C,DC,Z	1	1,2
SUBLW	k	Subtract W from constant	$W-k \rightarrow W$	C,DC,Z	1	
SUBWF	f, d	Subtract W from f	$W-f \rightarrow d$	C,DC,Z	1	1,2
ANDLW	k	AND constant with W	$W.AND.k \rightarrow W$	Z	1	
ANDWF	f, d	AND W with f	$W.AND.f \rightarrow d$	Z	1	1,2
IORLW	k	OR constant with W	$W.OR.k \rightarrow W$	Z	1	
IORWF	f, d	OR W with f	$W.OR.f \rightarrow d$	Z	1	1,2
XORLW	k	Exclusive OR constant with W	$W.XOR.k \rightarrow W$	Z	1	1,2
XORWF	f, d	Exclusive OR W with f	$W.XOR.f \rightarrow d$	Z	1	
INCF	f, d	Increment f	$f+1 \rightarrow f$	Z	1	1,2
DECF	f, d	Decrement f	$f-1 \rightarrow f$	Z	1	1,2
RLF	f, d	Rotate Left f through carry		C	1	1,2
RRF	f, d	Rotate Right f through carry		C	1	1,2
COMF	f, d	Complement f	$F \rightarrow d$	Z	1	1,2



Bit operations

Instructions BCF and BSF do setting or cleaning of one bit anywhere in the memory.

The CPU first reads the whole byte, changes one bit in it and then writes in the entire byte at the same place.

Bit operations

BCF	f, b	Bit Clear f	$0 \rightarrow f(b)$		1	1,2
BSF	f, b	Bit Set f	$1 \rightarrow f(b)$		1	1,2

Directing a program flow

- Instructions **GOTO**, **CALL** and **RETURN** are executed the same way as on all other microcontrollers, only stack is independent of internal RAM and limited to eight levels.
- '**RETLW k**' instruction is identical with **RETURN** instruction, except that before coming back from a subprogram a constant defined by instruction operand is written in W register.

Directing a program flow

BTFSC	f, b	Bit Test f, Skip if Clear	jump if f(b)=0		1 (2)	3
BTFSS	f, b	Bit Test f, Skip if Set	jump if f(b)=1		1 (2)	3
DECFSZ	f, d	Decrement f, Skip if 0	f-1 → d, jump if Z=1		1(2)	1,2,3
INCFSZ	f, d	Increment f, Skip if 0	f+1 → d, jump if Z=0		1(2)	1,2,3
GOTO	k	Go to address	W.AND.k → W		2	
CALL	k	Call subroutine	W.AND.f → d		2	
RETURN	-	Return from Subroutine	W.OR.k → W		2	
RETLW	k	Return with constant in W	W.OR.f → d		2	
RETFIE	-	Return from interrupt	W.XOR.k → W		2	



Look-up tables Design:

- This instruction 'RETLW k' enables us to design easily the Look-up tables (lists)
- We use them by determining data position on our table adding it to the address at which the table begins, and then we read data from that location (which is usually found in program memory).

- Table can be formed as a subprogram which consists of a series of 'RETLW k' instructions, where 'k' constants are members of the table.
- We write the position of a member of our table in W register, and using CALL instruction to call a subprogram which creates the table.
- The instruction **ADDWF PCL, f** adds the position of a W register member to the starting address of our table, found in PCL register, and so we get the real data address in program memory.
- When returning from a subprogram we will have in W register the contents of an addressed table member.

```

Main    molov 2
        call Lookup
Lookup  addwf PCL, f
        retlw k
        retlw k1
        retlw k2
        :
        :
        retlw kn

```

Other instructions

NOP	-	No Operation			1	
CLRWDT	-	Clear Watchdog Timer	0 → WDT, 1 → TO, 1 → PD	$\overline{TO}, \overline{PD}$	1	
SLEEP	-	Go into standby mode	0 → WDT, 1 → TO, 0 → PD	$\overline{TO}, \overline{PD}$	1	



Instruction Execution Period

- All instructions are executed in one cycle except for conditional branch instructions if condition was true, or if the contents of program counter was changed by some instruction. In that case, execution requires two instruction cycles, and the second cycle is executed as NOP (No Operation).
- Four oscillator clocks make up one instruction cycle. If we are using an oscillator with 4MHz frequency, the normal time for executing an instruction is 1 μ s, and in case of conditional branching, execution period is 2 μ s.



INSTRUCTIONS



```
Main    movlw 2      ;write number 2 to accumulator
call    Lookup   ;jump to the lookup table
Lookup  addwf PCL,f ;add accumulator and program cur
        ;rent address (PCL)

retlw   k0       ;return from subroutine (accumulator contains k0)
retlw   k1       ;...
retlw   k2       ;...
...     ;...
...     ;...
retlw   kn       ;return from subroutine (accumulator contains kn)
```





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.





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.





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).





Instructions of PIC Microcontroller

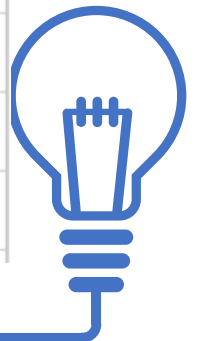


INSTRUCTION	DESCRIPTION	OPERATION	FLAG	CLK	*
Data Transfer Instructions					
MOVLW k	Move constant to W	k -> w		1	
MOVWF f	Move W to f	W -> f		1	
MOVF f,d	Move f to d	f -> d	Z	1	1, 2
CLRW	Clear W	0 -> W	Z	1	
CLRF f	Clear f	0 -> f	Z	1	2
SWAPF f,d	Swap nibbles in f	f(7:4),(3:0) -> f(3:0),(7:4)		1	1, 2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



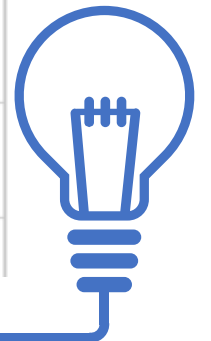


SUBWF f,d	Subtract W from f	f-W -> d	C, DC, Z	1	1, 2
ANDLW k	Logical AND with W with constant	W AND k -> W	Z	1	
ANDWF f,d	Logical AND with W with f	W AND f -> d	Z	1	1, 2
ANDWF f,d	Logical AND with W with f	W AND f -> d	Z	1	1, 2
IORLW k	Logical OR with W with constant	W OR k -> W	Z	1	
IORWF f,d	Logical 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 through CARRY bit		C	1	1, 2





RLF f,d	Rotate left f through CARRY bit		C	1	1, 2
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
INCFSSZ 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	





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	





Multiple Choice Question



- 1. Which instruction is applicable to set any bit while performing bitwise operation settings?**
 - a. bcf**
 - b. bsf**
 - c. both a & b**
 - d. none of the above**

- 2. Where is the result stored after an execution of increment and decrement operations over the special - purpose registers in PIC?**
 - a. File Register**
 - b. Working Register**
 - c. Both a & b**
 - d. none of the above**





3. Which flags of status register are most likely to get affected by the single-cycle increment and decrement instructions?

- a. P Flags**
- b. C Flags**
- c. OV Flags**
- d. Z Flags**

4. What is the execution speed of instructions in PIC especially while operating at the maximum value of clock rate?

- a. 0.1 μ s**
- b. 0.2 μ s**
- c. 0.4 μ s**
- d. 0.8 μ s**

5. Which flags are more likely to get affected in status registers by Arithmetic and Logical Unit (ALU) of PIC 16 CXX on the basis of instructions execution?

- a. Carry(C) Flags**
- b. Zero (Z) Flags**
- c. Digit Carry (DC) Flags**
- d. All of the above**





Answers:

1. b. bsf
2. c. Both a & b
3. d. Z Flags
4. b. $0.2 \mu\text{s}$
5. d. All of the above





Thank
you

