



SNS COLLEGE OF TECHNOLOGY

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 AND COMMUNICATION ENGINEERING

19ECB211 – MICROCONTROLLER PROGRAMMING & INTERFACING

II YEAR IV SEM

UNIT I – PIC MICROCONTROLLER : HISTORY , FEATURES & ARCHITECTURE

TOPIC 8 – PIC Instructions



PIC Instructions

- a program that the microcontroller should execute must be written and loaded into the microcontroller.
- 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.



PIC Instructions

OPCODE FIELD DISCRPTIONS

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
TO	Time-out bit
PD	Power-down bit



PIC – Data Transfer 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



PIC – Arithmetic & Logical Instructions



Arithmetic-logic Instructions					
ADDLW k	Add W and constant	$W+k \rightarrow W$	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	$k-W \rightarrow W$	C, DC, Z	1	
SUBWF f,d	Subtract W from f	$f-W \rightarrow d$	C, DC, Z	1	1, 2
ANDLW k	Logical AND with W with constant	$W \text{ AND } k \rightarrow W$	Z	1	
ANDWF f,d	Logical AND with W with f	$W \text{ AND } f \rightarrow d$	Z	1	1, 2
ANDWF f,d	Logical AND with W with f	$W \text{ AND } f \rightarrow d$	Z	1	1, 2
IORLW k	Logical OR with W with constant	$W \text{ OR } k \rightarrow W$	Z	1	
IORWF f,d	Logical OR with W with f	$W \text{ OR } f \rightarrow d$	Z	1	1, 2



PIC – Arithmetic & Logical Instructions



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
RRF f,d	Rotate right f through CARRY bit		C	1	1, 2
COMF f,d	Complement f	f -> d	Z	1	1, 2



PIC – 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.

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



PIC – Program Control Instructions



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	



PIC – 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	



References



<https://www.embedded.com/the-evolution-of-embedded-devices-addressing-complex-design-challenges/>

<http://iamtechnical.com/org-origin-end-list-include-config-radix-directives>

<https://www.electronicsspecifier.com/products/design-automation/embedded-systems-the-evolution-of-embedded-system-design>

Mazidi M. A., McKinlay R. D., Causey D. "PIC Microcontroller And Embedded Systems" Pearson Education International, 2008(Unit I,II,III, IV & V)

Thank You