

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Tarea 1: Conjunto de Instrucción

Profesor: Rubén Anaya García

Alumnos: Reza Chavarría, Sergio Gabriel

Grupo: 1

Semestre: 2021-2

Tarea 1

Explicar el funcionamiento de cada una de las instrucciones, dar ejemplos de cada una de ellas y generar el código de la instrucción tanto en modo de direccionamiento directo como indirecto.

Para las instrucciones que tienen registro de destino, la selección será la siguiente

Registro F	F, f, REG, 1, [Sin parámetro]
Registro W	W, w, 0

- **ANDWF f,d**

- **Descripción**

Suma el contenido que se tiene en el registro W además del contenido de la misma instrucción

para guardarlo en la dirección o destino final.

```
00-0101-dfff-ffff
```

- **Ejemplo => Direccionamiento Directo**

```
processor 16f877a
include <p16f877a.inc>

VAR EQU H'60'
AUX EQU H'61'
ORG 0
GOTO INICIO
ORG 5

INICIO
MOVLW 0XE2
MOVWF VAR ;VAR->0x0XE2
```

```

MOVLW 0X44 ;W->0X44
ANDWF VAR,0 ;VAR->0XE2 AND 0X44=0X40
MOVWF AUX ;AUX->0X40
END

```

Código de instrucción obtenido

```
00-0101-0110-0000: 0560
```

- **Ejemplo => Direccionamiento indirecto**

```

processor 16f877a
include <p16f877a.inc>

ORG 0
GOTO INICIO
ORG 5

INICIO
    MOVLW 0X55
    MOVWF FSR ;FSR->0X55
    MOVLW 0X7E
    MOVWF INDF ;0X55->0X7E
    MOVLW 0X58 ;W->0X58
    ANDWF INDF,0 ;W->0X7E AND 0X58=0x58
    DECF FSR ;FSR->0X54
    MOVWF INDF ;0X54->0X58
END

```

Código de instrucción obtenido

```
00-0101-0000-0000: 0500
```

- **COMF f,d**

- **Descripción**

La instrucción realiza el complemento del registro asignado. El resultado lo guardará en el registro destino.

```
00 1001 dfff ffff
```

- **Ejemplo => Direccionamiento Directo**

```

processor 16f877a
include <p16f877a.inc>

VAR EQU H'20'
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0xB8
    MOVWF VAR ;VAR->0xB8
    COMF VAR,1 ;VAR->0x47
    END

```

Código de instrucción obtenido

```
00-1001-1010-0000: 09A0
```

- **Ejemplo => Direccionamiento indirecto**

```

processor 16f877a
include <p16f877a.inc>
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0x3A
    MOVWF FSR ;FSR->0x3A
    MOVLW 0x5F
    MOVWF INDF ;0x3A->0x5F
    COMF INDF,1 ;0x3A->0xA0
    END

```

Código de instrucción obtenido

```
00-1001-1000-0000:0980
```

- **DECF f,d**

- **Descripción**

Se realiza un decremento de una unidad en el registro seleccionado posteriormente este se guarda en el registro destino

```
00 0011 dfff ffff
```

- **Ejemplo => Direccionamiento Directo**

```

processor 16f877a
include <p16f877a.inc>

VALOR1 EQU H'2F'
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0XF0
    MOVWF VAR ;VAR->0XF0
    DECF VAR ;VAR->0XF0-0X01=0XEF
    END

```

Código de instrucción obtenido

00-0011-1010-1111: 03AF

- **Ejemplo => Direccionamiento indirecto**

```

processor 16f877a
include <p16f877a.inc>
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0X66
    MOVWF FSR ;FSR->0X66
    MOVLW 0X71
    MOVWF INDF ;<0X66>=0X71
    MOVLW 0X11
    ADDWF INDF,0 ;W->0X11 + 0X71 = 0X82
    DECF FSR,1 ;FSR=0X65
    MOVWF INDF ;<0X65>=0X82
    END

```

Código de instrucción obtenido

00-0011-1000-0100: 0584

- **INCF f,d**

- **Descripción**

Incrementa una unidad el registro establecido o seleccionado además de guardar el resultado en el registro destino

00-1010-dfff-ffff

◦ **Ejemplo => Direcccionamiento Directo**

```
processor 16f877a
include <p16f877a.inc>
VAR EQU H'52'
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0x00
    MOVWF VAR ;VAR->0x00

LOOP
    INCF VAR,F ;VAR->VAR+1
    GOTO LOOP
END
```

Código de instrucción obtenido

00-1010-1101-0010: 0AD2

◦ **Ejemplo => Direcccionamiento indirecto**

```
processor 16f877a
include <p16f877a.inc>
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0x3F
    MOVWF FSR ;FSR=0x3F
    MOVLW 0x7F
    MOVWF INDF ;<0x3F>=0x3F

LOOP
    MOVLW 0x40
    MOVWF FSR ;FSR=0x3F
    INCF INDF ;<0x3F>+1
    MOVLW 0xFF
    XORWF INDF,1 ;0x7F XOR 0xFF
    BTFSS STATUS,Z ;Z=0?
    GOTO LOOP ;Z=0
    GOTO INICIO ;Z=1
END
```

Código de instrucción obtenido

```
00-1010-1000-0000: 0A80
```

- **DECFSZ f,d**

- **Descripción**

Decrementa una unidad el registro seleccionado, posteriormente se revisa el bit z del registro status.

```
00-1011-dfff-ffff
```

- **Ejemplo => Direccionamiento Directo**

```
processor 16f877a
include <p16f877a.inc>

VAR EQU H'44'
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0x12
    MOVWF VAR ;VAR=0x12
    LOOP
    DECFSZ VAR ;VAR->VAR-1
    GOTO LOOP ;Z=0
    GOTO SUMA ;Z=1
SUMA
    DECF VAR,1 ;VAR-1
    GOTO LOOP
END
```

Código de instrucción obtenido

```
00-1011-1100-0100: 0BC4
```

- **Ejemplo => Direccionamiento indirecto**

```
processor 16f877a
include <p16f877a.inc>

    ORG 0TAREA 01 10
```

```

GOTO INICIO
ORG 5

INICIO
    MOVLW 0X5F
    MOVWF FSR ;FSR->0X5F
    MOVLW 0X10
    MOVWF INDF ;0X5F->0X10
    LOOP
    DECFSZ INDF,1 ;0X5F->0X5F-1
    GOTO LOOP ;Z->0
    GOTO INICIO ;Z->1
END

```

Código de instrucción obtenido

```
00-1011-1000-0000: 0B80
```

- **INCFSZ f,d**

- **Descripción**

Incrementa en uno el registro seleccionado además de revisar el bit z del registro del status.

```
00 1111 dfff ffff
```

- **Ejemplo => Direccionamiento Directo**

```

processor 16f877a
include <p16f877a.inc>

VAR1 EQU H'70'
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0X90
    MOVWF VAR ;VAR=0x90
    LOOP
    INCFSZ VAR ;VAR->VAR+1
    GOTO LOOP ;Z=0
    GOTO INICIO ;Z=1
END

```

Código de instrucción obtenido

```
00 -1111-1111-0000: 0FF0
```

- **Ejemplo => Direccionamiento indirecto**

```
processor 16f877a
include <p16f877a.inc>

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0X66
    MOVWF FSR ;FSR=0X66
    MOVLW 0X90
    MOVWF INDF ;0X66->0X90
    LOOP
    INCF SZ INDF,F ;0X66-><0X66>+1
    GOTO LOOP ;Z=0
    GOTO INICIO ;Z=1
    END
```

Código de instrucción obtenido

```
00-1111-1000-0000: 0F80
```

- **ADDWF f,d**

- **Descripción**

Suma el contenido que se tiene en el registro W además del contenido de la misma instrucción

para guardarlo en la dirección o destino final.

```
00-0111-dfff-ffff
```

- **Ejemplo => Direccionamiento Directo**

```
processor 16f877a
include <p16f877a.inc>

VAR EQU H'30'
    ORG 0
    GOTO INICIO
    ORG 5
```



```

INICIO
    MOVLW 0x25
    MOVWF VAR ;VAR->0x25
    MOVLW 0x30 ;W->0x30
    ADDWF VAR,0 ;W->W+VAR=0x25+0x30=0x55
    MOVWF H'31' ;H'31'->0x55
END

```

Código de instrucción obtenido

```
00-0111-0011-0000: 0730
```

- **Ejemplo => Direccionamiento indirecto**

```

processor 16f877a
include <p16f877a.inc>

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0x3F
    MOVWF FSR ;FSR->0x3F
    MOVLW 0x85
    MOVWF INDF ;0x3F->0x85
    MOVLW 0x50 ;W->0x50TAREA 01 3
    ADDWF INDF,1 ;0x3F->0x85+0x50=0xD5
END

```

Código de instrucción obtenido

```
00-0111-1000-0000: 0780
```

- **CLRF W**

- **Descripción**

La instrucción limpia todos los bits del registro W, por lo que no es necesario escribir un registro a limpiar

```
00-0001-0xxx-xxxx
```

- **Ejemplo => Direccionamiento Directo**

```

processor 16f877a
include <p16f877a.inc>

VAR EQU H'20'
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0X3B
    MOVWF VAR ;VAR->0X3B
    CLRF VAR ;VAR->0X00
    END

```

Código de instrucción obtenido

```
00-0001-0000-0000:0100
```

- **Ejemplo => Direccionamiento indirecto**

```

processor 16f877a
include <p16f877a.inc>
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0X30
    MOVWF FSR ;FSR->0X30
    MOVLW 0XFE
    MOVWF INDF ;0X30->0XFE
    MOVLW 0X2F
    MOVWF FSR ;FSR->0X2F
    CLRW ;W->0X00
    MOVWF INDF ;0X2F->0X00
    FIN
    GOTO FIN
    END

```

Código de instrucción obtenido

```
00-0001-0000-0000: 0100
```

- **CLRF f**

- **Descripción**

La instrucción limpia todos los bits del registro de la instrucción

```
00 0001 1fff ffff
```

- **Ejemplo => Direccionamiento Directo**

```
processor 16f877a
include <p16f877a.inc>

VAR EQU H'20'
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0X3B
    MOVWF VAR ;VAR->0X3B
    CLRF VAR ;VAR->0X00
    END
```

Código de instrucción obtenido

```
00-0001-1010-0000: 01A0
```

- **Ejemplo => Direccionamiento indirecto**

```
processor 16f877a
include <p16f877a.inc>
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0X34
    MOVWF FSR ;FSR->0X34
    MOVLW 0XA2
    MOVWF INDF ;0X34->0XA2
    COMF INDF ;0X34->COMP(0XA2)->0X5D
    CLRF INDF ;0X34->0X00
    END
```

Código de instrucción obtenido

```
00-0001-1000-0000: 0180
```

- **MOVF f,d**

- **Descripción**

Sirve para mover el contenido del registro seleccionado al registro destino

00-1000-dfff-ffff

◦ **Ejemplo => Direcccionamiento Directo**

```
processor 16f877a
include <p16f877a.inc>

VAR EQU H'63'
AUX EQU H'73'

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0XDD
    MOVWF VAR ;VAR=0xDDTAREA 01 15
    COMF VAR,1 ;VAR->COMP(0XDD)->0X22
    MOVF VAR,W ;W->0X22
    MOVWF AUX,1 ;AUX->0X22
    END
```

Código de instrucción obtenido

00-1000-0110-0011: 0863

◦ **Ejemplo => Direcccionamiento indirecto**

```
processor 16f877a
include <p16f877a.inc>

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0X33
    MOVWF FSR ;FSR->0X33
    MOVLW 0XAB
    MOVWF INDF ;0X33->0XAB
    MOVF INDF,W ;W=0X45
    MOVWF H'20' ;H'20'->0X45
    END
```

Código de instrucción obtenido

00-1000-0000-0000: 0800

- **IORWF f,d**

- **Descripción**

Hace una operación lógica OR inclusivo , donde si alguno de los bits a comparar o ambos bits son 1 el resultado será igual 1.

NOTA: se compara W respecto al registro seleccionado

```
00-0100-dfff-ffff
```

- **Ejemplo => Direccionamiento Directo**

```
processor 16f877a
include <p16f877a.inc>

VAR EQU H'55'
AUX EQU H'56'

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0x27
    MOVWF VAR ;VAR->0x27
    MOVLW 0x38
    IORWF VAR,W ;W->0x27 IOR 0x38=0x3F
    MOVWF AUX ;AUX->0x3F
    END
```

Código de instrucción obtenido

```
00-0100-0101-0101: 0455
```

- **Ejemplo => Direccionamiento indirecto**

```
processor 16f877a
include <p16f877a.inc>TAREA 01 14

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0x3E
    MOVWF FSR ;FSR->0x3E
    MOVLW 0x4A
    MOVWF INDF ;0x3E->0x4A
    MOVLW 0x81 ;W->0x81
    IORWF INDF ;0x3F-> 0x4A IOR 0x81= 0xCB
```

END

Código de instrucción obtenido

00-0100-1000-0000: 0480

- **MOVWF f,d**

- **Descripción**

Sirve para mover el contenido del registro W al registro destino

00 0000 1fff ffff

- **Ejemplo => Direccionamiento Directo**

```
processor 16f877a
include <p16f877a.inc>

VAR EQU H'50'
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0x40
    MOVWF VAR ;VAR->0x40
    MOVLW 0x30 ;W->0x30
    ADDWF VAR,F ;VAR->0x40+0x30=0x70
    END
```

Código de instrucción obtenido

00-0000-1101-0000: 00D0

- **Ejemplo => Direccionamiento indirecto**

```

processor 16f877a
include <p16f877a.inc>

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0x7F
    MOVWF FSR ;FSR->0x7F
    MOVLW 0x90
    MOVWF INDF ;0x7F->0x90
    END

```

Código de instrucción obtenido

```
00-0000-1000-0100: 0084
```

- **NOP**

- **Descripción**

La instrucción no realiza ninguna acción , sirve solo para hacer tiempo en código y reloj.

```
00 0000 0xx0 0000
```

- **Ejemplo => Direccionamiento Directo**

```

processor 16f877a
include <p16f877a.inc>

VALOR1 EQU H'40'
VALOR2 EQU H'41'
RESULT EQU H'42'

    ORG 0
    GOTO LOOP
    ORG 5

LOOP
    NOP
    NOP
    NOP
    GOTO LOOP

EN

```

Código de instrucción obtenido

```
00 0000 0000 0000: 0000
```

- **RLF f,d**

- **Descripción**

Recorre los bits del registro seleccionado hacia la izquierda además de pasar un bit de acarreo

```
00 1101 dfff ffff
```

- **Ejemplo => Direccionamiento Directo**

```
processor 16f877a
include <p16f877a.inc>
VAR EQU H'60'

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0XF8
    MOVWF VAR ;VAR->0XF8
    RLF VAR,1 ;VAR->0XF0 C=1
END
```

Código de instrucción obtenido

```
00-1101-1110-0000: 0DE0
```

- **Ejemplo => Direccionamiento indirecto**

```
processor 16f877a
include <p16f877a.inc>

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0X30
    MOVWF FSR ;FSR->0X30
    MOVLW 0X43
    MOVWF INDF ;0X30->0X43
    RLF INDF,1 ;INDF->RLF(0X43)->0X86
FIN
```



```
GOTO FIN
END
```

Código de instrucción obtenido

```
00-1101-1000-0000: 0D80
```

- **RRF f,d**

- **Descripción**

Recorre de posición los bits del registro seleccionado hacia la derecha. La información del bit 0, al hacer el recorrido a la derecha, se pasa al bit de acarreo.

```
00 1100 dfff ffff
```

- **Ejemplo => Direccionamiento Directo**

```
processor 16f877a
include <p16f877a.inc>
VAR EQU H'35'

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0XF8
    MOVWF VAR ;VAR->0XF8
    RRF VAR,1 ;VAR->0X7C
END
```

Código de instrucción obtenido

```
00-1100-1011-0101: 0CB5
```

- **Ejemplo => Direccionamiento indirecto**

```
processor 16f877a
include <p16f877a.inc>

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
```

```

MOVLW 0X40
MOVWF FSR ;FSR->0X40
MOVLW 0X43
MOVWF INDF ;0X40->0X43
RRF INDF,1 ;INDF->RRF(0X43)->0X21 C=1
END

```

Código de instrucción obtenido

```
00-1100-1000-000: 0c80
```

- **SUBWF f,d**

- **Descripción**

Resta el contenido del registro W al registro seleccionado. El resultado lo guarda en el registro destino.

```
00 0010 dfff ffff
```

- **Ejemplo => Direccionamiento Directo**

```

processor 16f877a
include <p16f877a.inc>

VAR EQU H'4C'
    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0X55
    MOVWF VAR1 ;VAR->0X55
    MOVLW 0X31
    SUBWF VAR,1 ;VAR->0X55-0X31=0X24
    END

```

Código de instrucción obtenido

```
00-0010-1100-1100: 02cc
```

- **Ejemplo => Direccionamiento indirecto**

```

processor 16f877a
include <p16f877a.inc>

    ORG 0
    GOTO INICIO

```

```
ORG 5
```

```
INICIO
```

```
    MOVLW 0X20
```

```
    MOVWF FSR ;FSR->0X20
```

```
    MOVLW 0X66
```

```
    MOVWF INDF ;0X20->0X66
```

```
    MOVLW 0X20 ;W->0X23
```

```
    SUBWF INDF,1 ;0X20->0X66-0X23=0X43
```

```
END
```

Código de instrucción obtenido

```
00-0010-1000-0000: 0280
```

- **SWAPF f,d**

- **Descripción**

Intercambia los 4 bits de mayor valor por los 4 de menor valor del registro seleccionado. El resultado lo guarda en el registro destino

```
00 1110 dfff ffff
```

- **Ejemplo => Direccionamiento Directo**

```
processor 16f877a
```

```
include <p16f877a.inc>
```

```
VAR EQU H'55'
```

```
    ORG 0
```

```
    GOTO INICIO
```

```
    ORG 5
```

```
INICIO
```

```
    MOVLW 0X5A
```

```
    MOVWF VAR ;VAR=0X5A
```

```
    SWAPF VAR,1 ;VAR->0XA5
```

```
END
```

Código de instrucción obtenido

```
00-1110-1101-0101: 0ED5
```

- **Ejemplo => Direccionamiento indirecto**

```

processor 16f877a
include <p16f877a.inc>
    ORG 0
    GOTO INICIOTAREA 01 22
    ORG 5

INICIO
    MOVLW 0X66
    MOVWF FSR ;FSR->0X66
    MOVLW 0X4D
    MOVWF INDF ;0X66->0X4D
    SWAPF INDF,1 ;<0X66>->0XD4
    END

```

Código de instrucción obtenido

```
00-1110-1000-0000: 0E80
```

- **XORWF f,d**

- **Descripción**

Realiza la operación lógica de OR exclusivo, solo si alguno de los bits a comparar es 1 el resultado será 1

```
00 1011 dfff ffff
```

- **Ejemplo => Direccionamiento Directo**

```

processor 16f877a
include <p16f877a.inc>
VAR EQU H'49'
AUX EQU H'50'

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0X23
    MOVWF VAR ;VAR->0x23
    MOVLW 0X85 ;W->0X85
    XORWF VAR,0 ;W->0X23 XOR 0X85=0XA6
    MOVWF AUX ;AUX->0XA6
    END

```

Código de instrucción obtenido

```
00 0110 0100 1001: 0649
```

- **Ejemplo => Direcccionamiento indirecto**

```
processor 16f877a
include <p16f877a.inc>

    ORG 0
    GOTO INICIO
    ORG 5

INICIO
    MOVLW 0X66
    MOVWF FSR ;FSR->0X66
    MOVLW 0X79
    MOVWF INDF ;0X66->0X79
    MOVLW 0X70
    XORWF INDF,1 ;0X66->0X79 XOR 0X70 = 0X09
    END
```

Código de instrucción obtenido

```
00 0110 1000 0000: 0680
```