

Facultad de ingeniería

Materia: Laboratorio de Microcomputadoras

Previo 2

Título: Programación En Ensamblador direccionamiento indirecto

Integrantes:

- Martínez Pérez Brian Erik - 319049792
- Nuñez Rodas Abraham Enrique - 114003546
- Vicenteño Maldonado Jennifer Michel - 317207251

Profesor: Moises Melendez Reyes

Grupo: 1

Fecha de Entrega: 23 de febrero de 2025

Semestre: 2025-2



1. Explique el modo de direccionamiento directo para los microcontroladores PIC16F877A.

Para acceder una posición de memoria mediante direccionamiento directo, la CPU simplemente usa la dirección indicada en los 7 bits menos significativos del código de operación y la selección de banco de los bits RP1:RP0 como se ilustra en la siguiente figura.

ejemplo: MOVLW 0x55 ; Cargar el valor 0x55 en el registro W

2. Explique el modo de direccionamiento indirecto para los microcontroladores PIC16F877A.

Este modo de direccionamiento permite acceder una localidad de memoria de datos usando una dirección de memoria variable a diferencia del direccionamiento directo, en que la dirección es fija. Esto puede ser útil para el manejo de tablas de datos.

ejemplo: MOVWF INDF ; Guardar en la dirección apuntada por FSR (0x20)

3. Explique el funcionamiento de los registros FSR e INDF en el modo de direccionamiento indirecto.

FSR (File Select Register): Actúa como un puntero que contiene la dirección de memoria RAM a la que se desea acceder. Cuando se modifica su valor, apunta dinámicamente a diferentes posiciones de la memoria.

INDF (Indirect File Register): No es un registro físico real, sino un alias que accede al contenido de la dirección almacenada en FSR.

4. Para el siguiente código y utilizando MPLab:

```
5
6     processor 16f877a
7     include <p16f877a.inc>
8
9     org 0
10    goto inicio
11
12    org 5
13 inicio: movlw 0x20      ; W  <--  0x20
14          movwf FSR      ; inicializa el puntero
15 next:   clrf INDF       ; borra el contenido de lo que apunta FSR, (INDF) = 0
16          incf FSR,f     ; incrementa el apuntador, FSR = FSR + 1
17          btfss FSR,4     ; ya hemos terminado?
18          goto next      ; no, ve a borrar la siguiente localidad
19          goto inicio    ; si, ve a la etiqueta "inicio"
20          end
```

Evidencia de ejecución:

```
processor 16f877a
include <pl6f877a.inc>
```

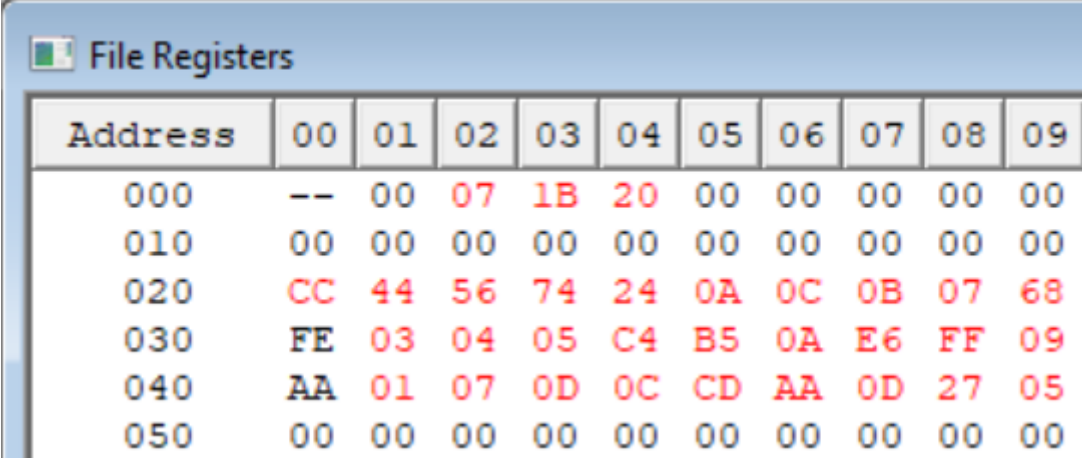
```
ORG 0
GOTO INICIO
ORG 5
```

```
INICIO: MOVLW 0x20
        MOVWF FSR
NEXT:   CLRF INDF
        INCF FSR, f
        BTFSS FSR, 4
        GOTO NEXT
        GOTO INICIO
END
```

1	0000	2805	GOTO 0x5
2	0001	3FFF	
3	0002	3FFF	
4	0003	3FFF	
5	0004	3FFF	
6	0005	3020	MOVLW 0x20
7	0006	0084	MOVWF 0x4
8	0007	0180	CLRF 0
9	0008	0A84	INCF 0x4, F
10	0009	1E04	BTFSS 0x4, 0x4
11	000A	2807	GOTO 0x7
12	000B	2805	GOTO 0x5

a) Escriba en la venta de memoria RAM (File Registers) valores aleatorios desde la localidad 0x20 hasta la localidad 0x40.

/*



Address	00	01	02	03	04	05	06	07	08	09
000	--	00	07	1B	20	00	00	00	00	00
010	00	00	00	00	00	00	00	00	00	00
020	CC	44	56	74	24	0A	0C	0B	07	68
030	FE	03	04	05	C4	B5	0A	E6	FF	09
040	AA	01	07	0D	0C	CD	AA	0D	27	05
050	00	00	00	00	00	00	00	00	00	00

b) Ejecute el programa paso a paso (use el control "ANIMATE") *Incluya una imagen con la simulación del programa como evidencia.

File Registers										
Address	00	01	02	03	04	05	06	07	08	09
000	--	00	07	1B	22	00	00	00	00	00
010	00	00	00	00	00	00	00	00	00	00
020	00	00	00	00	00	00	00	00	00	00
030	FE	03	04	05	C4	B5	0A	E6	FF	09
040	AA	01	07	0D	0C	CD	AA	0D	27	05
050	00	00	00	00	00	00	00	00	00	00

5. Explique qué es lo que hace el programa.

El programa utiliza el puntero FSR en el que almacena la dirección 0x20, después vacía el contenido de 0x20, después incrementa el valor en FSR en 1, volviendolo 0x21, después revisa el valor de bit 4 de FSR comparando si su valor es mayor a 0x10, repite el proceso hasta encontrar la siguiente dirección y borra de nuevo su contenido,.

6 ¿Por qué el programa no modifica los valores desde las localidades 0x30?

Esto es debido a la instrucción BTFSS, FSR 4, ya que cuando llega a 0x30 el valor de ese bit se vuelve 1 termina el programa.

Referencias

del PIC, 2. 1-La Familia. (s/f). 2.- Descripción General del PIC16F877. Edu.ar. Recuperado el 23 de febrero de 2025, de [https://exa.unne.edu.ar/ingenieria/sistemas/public_html/Archi_pdf/HojaDatos/Micro controlador es/PIC16F877.pdf](https://exa.unne.edu.ar/ingenieria/sistemas/public_html/Archi_pdf/HojaDatos/Micro%20controlador%20es/PIC16F877.pdf)

GARCÍA, Rubén, SAVAGE, Jesús, MUNIVE, Carlos. Prácticas de laboratorio de microcomputadoras, México, Universidad Nacional Autónoma de México Facultad de Ingeniería.