



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO FACULTAD DE INGENIERÍA  
DIVISIÓN DE INGENIERÍA ELÉCTRICA  
DEPARTAMENTO DE INGENIERÍA EN  
COMPUTACIÓN**

**LABORATORIO DE MICROCOMPUTADORAS**

Jose Francisco Ugalde Vivo Gpo teoría  
Hector Montoya Perez Gpo teoría 5  
Lara Sala Kevin Arturo Gpo teoría

# Práctica #1

# Introducción General a un Microcontrolador PIC16F877

**Objetivo.** Familiarizar al alumno en el conocimiento del ensamblador, del simulador, el conjunto de instrucciones de un microcontrolador y ejecutar programas en tiempo de simulación.

3.- Modificar el programa anterior, para que ahora los datos que operará se encuentren en las localidades **J** y **K** respectivamente, el resultado almacenarlo en otras direcciones, reservadas para **C1** y **R1**; C1 representa el valor de la bandera de acarreo y R1 el resultado.

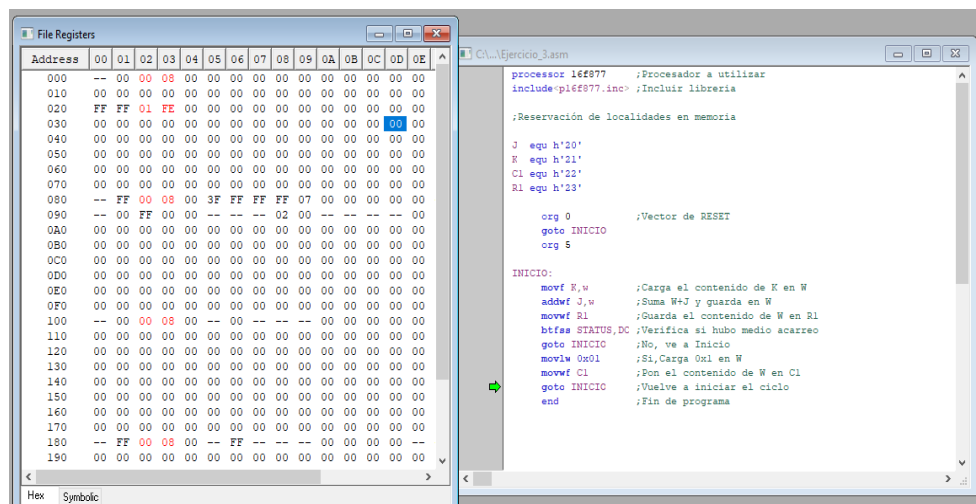
```
processor 16f877    ;Procesador a utilizar
include<p16f877.inc> ;Incluir libreria
```

;Reservación de localidades en memoria

J equ h'20'  
K equ h'21'  
C1 equ h'22'  
R1 equ h'23'

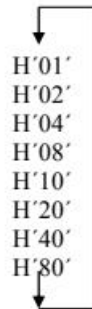
```
org 0
goto INICIO
org 5
```

INICIO:	
movlw h'0'	;Guardamos en w el 0
movwf C1	;Limpiamos C1
movwf R1	;Limpiamos R1
movf K,w	;Carga el contenido de K en W
addwf J,w	;Suma W+J y guarda en W
movwf R1	;Guarda el contenido de W en R1
btfss STATUS,DC	;Verifica si hubo medio acarreo
goto INICIO	;No, ve a Inicio
movlw 0x01	;Si,Carga 0x1 en W
movwf C1	;Pon el contenido de W en C1
goto INICIO	;Vuelve a iniciar el ciclo
end	;Fin de programa



4.- Realice un programa que ejecute la siguiente secuencia, misma que deberá ver en la dirección de memoria (registro) de su elección.

Secuencia:



Donde H'01' indica que el dato esta dado en hexadecimal.

```
processor 16f877      ;Procesador a utilizar
include<p16f877.inc> ;Incluir libreria
```

```
org 0
goto INICIO
org 5
```

INICIO:

```
movlw h'1'      ;Guardamos en W el valor de 1
movwf h'20'     ;Guardamos en espacio h'20' el valor de W
```

CICLO:

```
rlf h'20',1      ;Hacemos un recorrimiento de bit a la izquierda
btfss h'20',7    ;Validamos el bit de la posición 7, Si es 1 salta
goto CICLO       ;No salto, vamos a CICLO
goto INICIO      ;Si salto, vamos a INICIO
end
```

**File Registers Table:**

Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E
000	--	00	0A	18	00	00	00	00	00	00	00	00	00	00	00
010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
020	80	00	00	00	00	00	00	00	00	00	00	00	00	00	00
030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
080	--	FF	0A	18	00	3F	FF	FF	FF	07	00	00	00	00	00
090	--	00	FF	00	00	--	--	--	02	00	--	--	--	--	00
0A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
100	--	00	0A	18	00	--	00	--	--	--	00	00	00	00	00
110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
180	--	FF	0A	18	00	--	FF	--	--	--	00	00	00	00	--
190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

**Assembly Code (Ejercicio\_3.asm):**

```

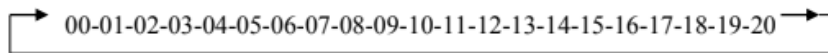
processor 16f877      ;Procesador a utilizar
include<p16f877.inc> ;Incluir libreria

org 0
goto INICIO
org 5

INICIO:
movlw h'1'      ;Guardamos en W el valor de 1
movwf h'20'     ;Guardamos en espacio h'20' el valor de W

CICLO:
rlf h'20',1      ;Hacemos un recorrimiento de bit a la izquierda
btfss h'20',7    ;Validamos el bit de la posición 7, Si es 1 salta
goto CICLO       ;No salto, vamos a CICLO
goto INICIO      ;Si salto, vamos a INICIO
end
  
```

5.- Desarrollar un programa que presente la cuenta en numeración decimal en la localidad de memoria de su elección, como se indica a continuación.



```
processor 16f877    ;Procesador a utilizar
include<p16f877.inc> ;Libreria
```

```
org 0
goto INICIO
org 5
```

INICIO

```
    movlw h'0'      ;Guardamos en W el 0
    movwf h'20'     ;Guardamos en la localidad 20 lo que hay en W
```

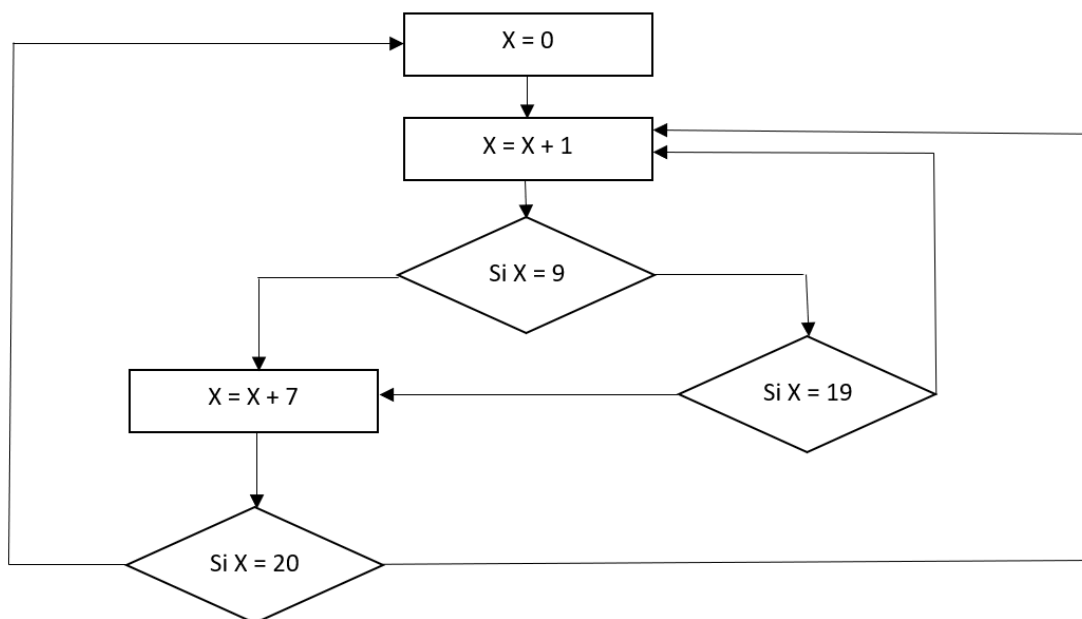
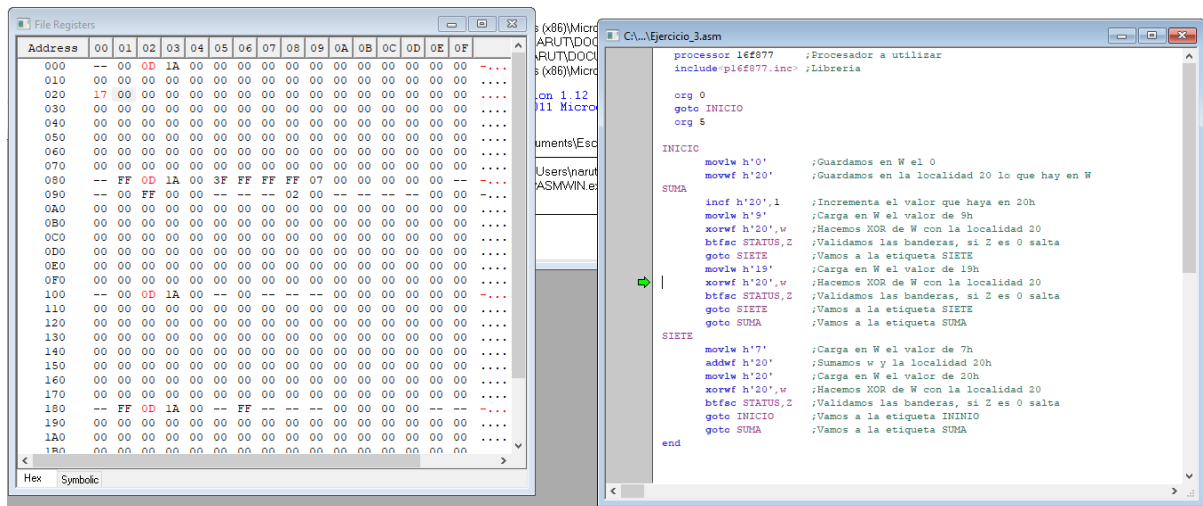
SUMA

```
    incf h'20',1    ;Incrementa el valor que haya en 20h
    movlw h'9'      ;Carga en W el valor de 9h
    xorwf h'20',w    ;Hacemos XOR de W con la localidad 20
    btfsc STATUS,Z  ;Validamos las banderas, si Z es 0 salta
        goto SIETE  ;Vamos a la etiqueta SIETE
    movlw h'19'     ;Carga en W el valor de 19h
    xorwf h'20',w    ;Hacemos XOR de W con la localidad 20
    btfsc STATUS,Z  ;Validamos las banderas, si Z es 0 salta
        goto SIETE  ;Vamos a la etiqueta SIETE
    goto SUMA       ;Vamos a la etiqueta SUMA
```

SIETE

```
    movlw h'7'      ;Carga en W el valor de 7h
    addwf h'20'     ;Sumamos w y la localidad 20h
    movlw h'20'     ;Carga en W el valor de 20h
    xorwf h'20',w    ;Hacemos XOR de W con la localidad 20
    btfsc STATUS,Z  ;Validamos las banderas, si Z es 0 salta
        goto INICIO ;Vamos a la etiqueta INICIO
    goto SUMA       ;Vamos a la etiqueta SUMA
```

end



## Conclusiones

Kevin Lara:

Los resultados obtenidos en la práctica, así como la realización de esta misma, nos ayudaron a familiarizarnos en gran medida con el simulador, lenguaje ensamblador y el set de instrucciones del microcontrolador.

Los ejercicios se trataron de hacer lo más parecido a las sugerencias del profesor durante la realización de la práctica a excepción de del ejercicio 4. La diferencia entre la solución presentada en clase a la que nosotros elaboramos fue únicamente que al momento de comprar los valores (para la cuestión de cambio de ciclo o reset del contador) en lugar de usar subwf utilizamos xorwf para comprobar si se trataba del mismo valor.

Montoya Pérez Héctor:

Después de la realización de la práctica, he empezado a entender más lenguaje ensamblador, el funcionamiento del microcontrolador y en especial he entendido muy bien cómo usar MPLAB. Personalmente, aún con la ayuda del set de instrucciones, foros de ayuda y ayuda de otros compañeros, me sigue costando un poco de trabajo entender bien el funcionamiento de los registros al igual que la sintaxis del set de instrucciones del microcontrolador. Es cuestión de práctica para perfeccionar los conocimientos adquiridos en esta sesión.

Con lo anterior dicho, podemos decir que los objetivos de la práctica fueron alcanzados completamente.

José Francisco Ugalde Vivó

Con la realización de los ejercicios contenidos en la práctica, hemos logrado resolver problemas básicos de la programación de un microcontrolador. De esta manera, hemos cumplido con los objetivos propuestos y repasado los conceptos vistos durante la explicación teórica de las clases. Asimismo, esto nos ha permitido conocer el entorno de desarrollo del microcontrolador y la manera de simular dicho elemento con las herramientas disponibles dentro del entorno de desarrollo. Finalmente, podemos decir que podremos continuar revisando estos conceptos en el futuro en cuanto al desarrollo y resolución de problemas con un microcontrolador.

### **Referencias Bibliográficas**

Microchip.(2003). PIC16F87XA Data Sheet. Recuperado de <https://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf>  
Explaining the Instruction Set for the PIC12c508A. Recuperado de [http://users.tpg.com.au/users/talking/explaining\\_instruction\\_set.html](http://users.tpg.com.au/users/talking/explaining_instruction_set.html)