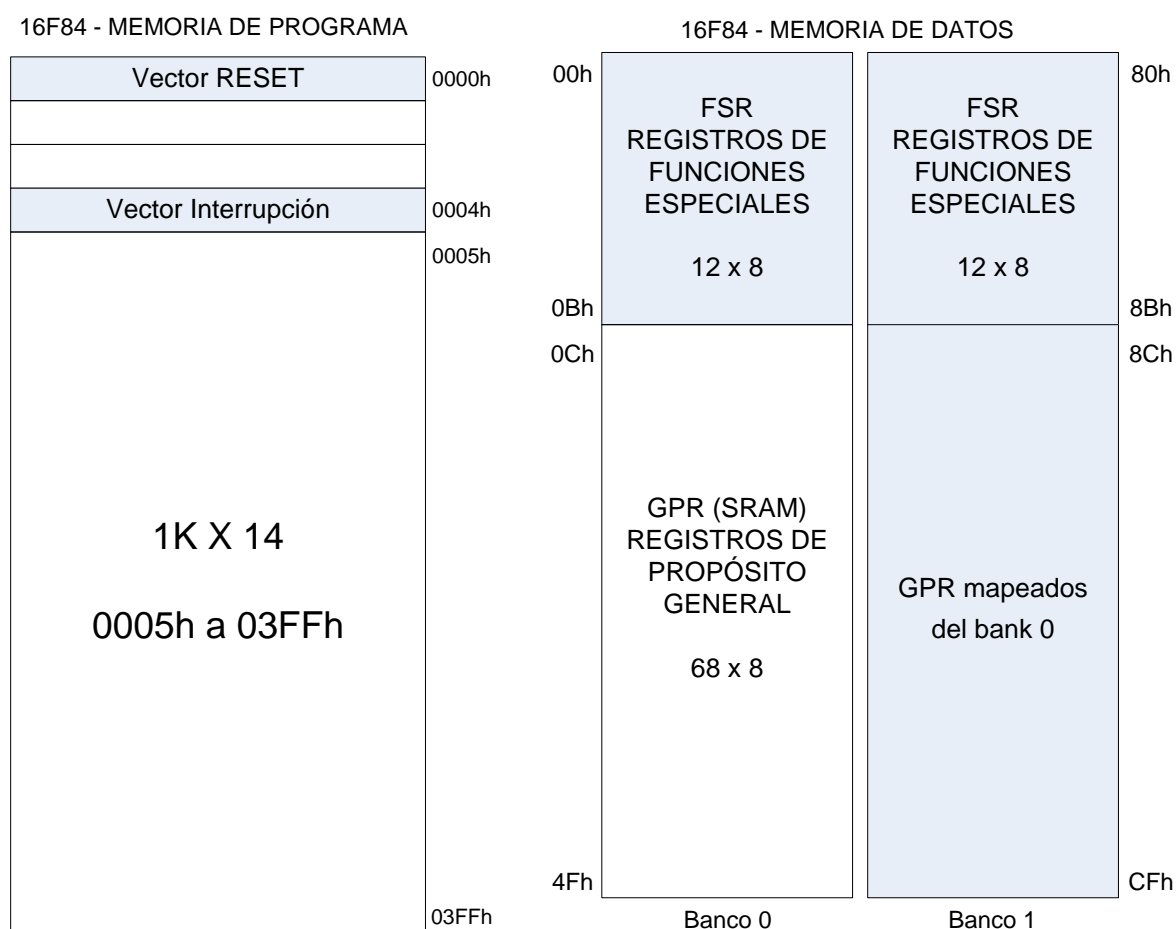


TRABAJO PRÁCTICO DE GABINETE N° 1:

PROGRAMACIÓN DEL PIC

Los ejercicios de este práctico se realizarán teniendo en cuenta un microcontrolador PIC 16F84A.

El esquema de memoria de este microcontrolador es el de la siguiente figura:

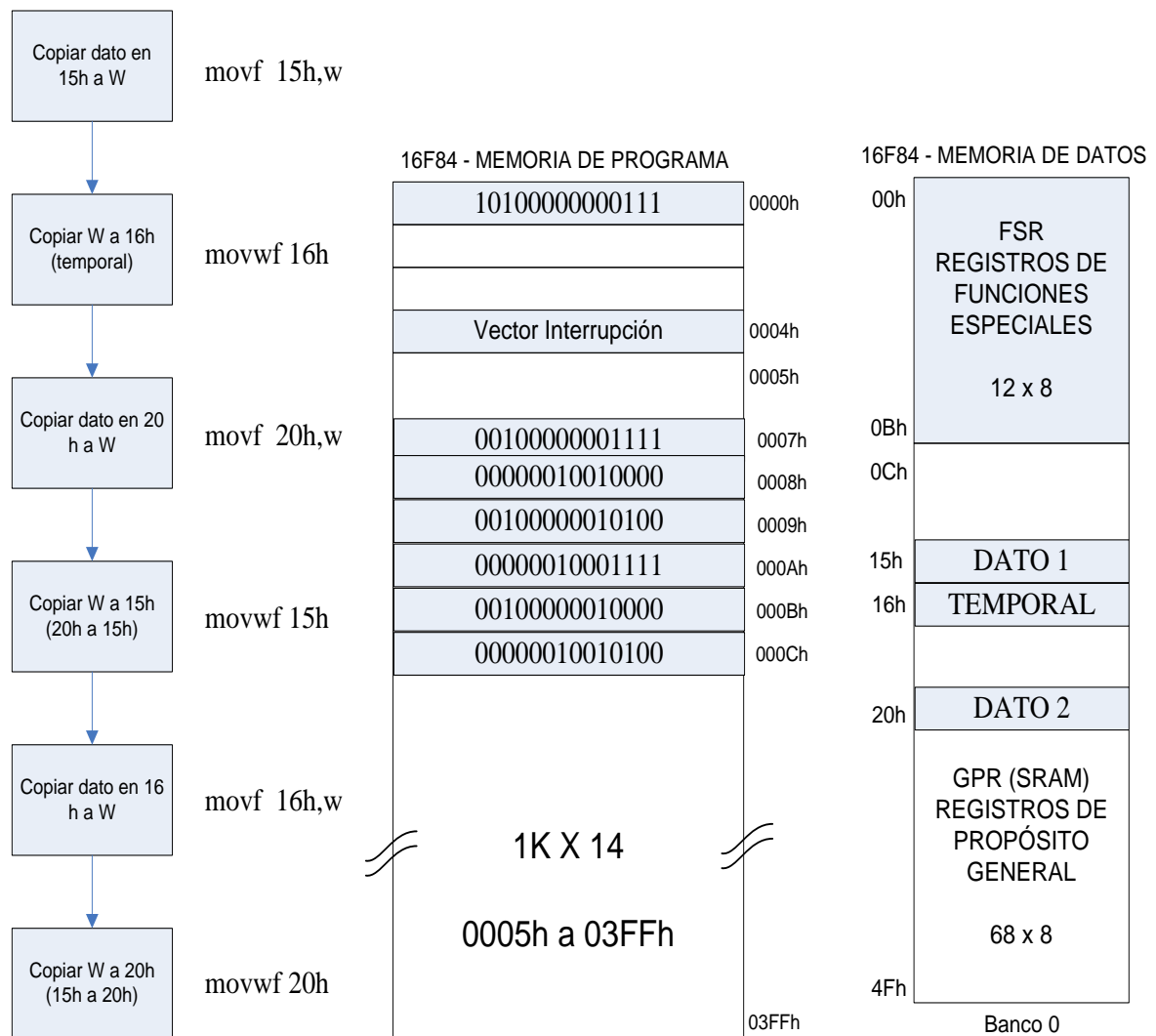


NOTA: si utilizamos el PIC 16f876, las posiciones de memoria deben modificarse por que no son las mismas que para el PIC 16f84.

Ejercicio n° 1:

Escriba un programa comenzando de la ubicación 7h que intercambie los datos contenidos de las ubicaciones 15h y 20h. Expresarlo en representación binaria, hexadecimal y mnemónica. Realizar un mapa de memoria de programa y de datos.

Ejemplificación de programa básico:



Dirección de memoria de programa	-----Instrucciones-----
----------------------------------	-------------------------

.... Dirección en hexadecimal	Mnemónica	Binaria	Hexadecimal
0x0000	goto 7h	10 1000 0000 0111	2 8 0 7
.....			
0x0007	movf 15h,w	00 1000 0000 1111	0 8 0 F
0x0008	movwf 16h	00 0000 1001 0000	0 0 9 0
0x0009	movf 20h,w	00 1000 0001 0100	0 8 1 4
0x000A	movwf 15h	00 0000 1000 1111	0 0 9 F
0x000B	movf 16h,w	00 1000 0001 0000	0 8 1 0
0x000C	movwf 20h	00 0000 1001 0100	0 0 9 4

Ejemplificación con variables:

```
; -----  
; Encabezado  
; -----  
    list    p=16f84a      ; list directive to define processor  
    #include <p16f84a.inc> ; processor specific variable definitions  
; -----  
; Definición de Variables  
; -----  
; acá van las definiciones de las variables con su posición  
primero    equ 0x15  
segundo    equ 0x20  
temp       equ 0x21  
; -----  
; Vectores  
; -----  
        ORG    0x000      ; processor reset vector  
        goto   main      ; go to beginning of program  
        ORG    0x004      ; interrupt vector location  
        goto   main      ;  
; =====  
; PROGRAMA PRINCIPAL  
; =====  
        org 0x007  
main  
        movlw 0x10  
        movwf primero    ;carga valor a primero  
        movlw 0x20  
        movwf segundo    ;carga valor a segundo  
        movf  primero,w  
        movwf temp  
        movf  segundo,w  
        movwf primero
```

```

movf temp,w
movwf segundo
clrf temp
;*****
goto $
END ; directive 'end of program'

```

Ejercicios obligatorios:

Ejercicio n° 2:

Proyecte una secuencia de instrucciones a partir de la dirección 20h que efectúe un salto a la ubicación 5Ah si el contenido de la dirección 1Bh es cero, y a la ubicación 12Ah si el contenido es distinto de cero. Represente el programa en lenguaje de máquina, assembler y hexadecimal. Realizar un mapa de memoria de programa y de datos.

Ejercicio n° 3:

Escriba un programa que comience en la ubicación 100h y sume tres números ubicados en las direcciones 25h, 26h, y 27h, y deposite el resultado en la dirección 28h. Represente el programa en lenguaje de máquina, assembler y hexadecimal. Realizar un mapa de memoria de programa y de datos.

Ejercicio n° 4:

Escriba un programa comenzando en la ubicación 300h que efectúe la operación lógica OR entre el contenido de la ubicación 46h y el literal A7h. Depositar el resultado en 46h. Represente el programa en lenguaje de máquina, assembler y hexadecimal. Realizar un mapa de memoria de programa y de datos.

Ejercicio n° 5:

Escriba un programa que efectúe la resta de dos números (con resultado positivo). Dichos números se encuentran en la ubicación 3Ah, y 3Bh y el resultado deberá depositarse en la dirección 3Ch. Comience el programa en la dirección 5h. Realizar un mapa de memoria de programa y de datos.

Ejercicio n° 6:

Escriba un programa que efectúe el intercambio del nibble más bajo con el más alto:

- a. Con la instrucción swap
- b. Con rotación de bits.

Ejercicio nº 7:

Realizar la multiplicación de dos números de 3 bits expresados en binario

.