

## EJERCICIOS

1. Después de ejecutar el siguiente programa en la MS:

```
MOV 12,13  
MOV 13,14  
ADD 15,13
```

a) ¿Cuál es el contenido de las posiciones de la tabla?

<b>12</b>	0000 0000 1111 1111	255
<b>13</b>	1111 1111 0000 0000	65280
<b>14</b>	1010 1010 1010 1010	43690
<b>15</b>	0101 0101 0101 0101	21845

Este es el contenido de las posiciones de la tabla.

255
255
22100
21845

b) ¿Cuál es el valor de FZ?

FZ es 0.

## HOJA DE EJERCICIOS DE LA CPU

2. Sabiendo que los contenidos en la figura se obtienen después de haber ejecutado el programa en la MS:

MOV 36, 37

ADD 37, 38

MOV 38, 39

CMP 39, 3A

36	00000000 11111111	255
37	00000000 11111111	255
38	11111111 11111111	65535
39	11111111 11111111	65535
40	00000000 11111111	255

a) Averiguar el valor inicial que tenía la posición 38 antes de ejecutar el programa.

Restamos 37 a 38 y nos da el resultado, 65280.

$11111111\ 11111111 - 00000000\ 11111111 = 11111111\ 00000000$

3. Obtener los formatos binarios que corresponden a las instrucciones del siguiente programa de la MS.

MOV 105,106 → 1000011110010000

CMP 106,107 → 0100100000010001

BEQ 100 → 1100010100001010

## HOJA DE EJERCICIOS DE LA CPU

4. Escribir un programa que realice la siguiente operación:

```
If ( a == b)
  c = a+1;
else
  c = b+1;
```

```
.ver 1 ;If ( a != b ); c=b+1; else; c=a+1

.data 12 ;dirección de inicio de los datos
dw var1 1
dw varA 2
dw varB 2
dw varC 0

.code
    cmp varA, varB                ;Comparamos A, B
    beq igual                    ;Si A,B son iguales salto a igual:
    mov varB, varC                ;Movemos B a C
    add var1, varC                ;Sumamos 1 a C
    cmp varC, varC                ;Comparamos la misma variable para poner FZ a 0
    beq END                      ;Salto a fin de programa
igual: mov varA, varC            ;Movemos A a C
    add var1, varC                ;Sumamos 1 a C
    END: cmp varA, varB           ;Fin de programa ()
```

→ Supones que en la posición 101 hay un 1, que a está en la 102, que b está en la 103 y que el resultado c debe quedar en la 104.

5. Escribir un programa que realice la siguiente operación:

```
i = 0;
while ( i < a)
  i = i + 1;
```

```
.ver 1 ;i=0 while (i<a) i=i+1

.data 15 ;dirección de inicio de los datos
dw varCero 0
dw varUno 1
dw varA 4
dw varI ?

.code
    mov varCero, varI            ;i=0
bucle: cmp varA, varI            ;comparo varA y varI
    beq final                    ;salta a final: si varA y varI son iguales
    add varUno, varI             ;sumar 1 a I
    cmp varUno, varUno           ;FZ=1
    beq bucle                    ;salto a bucle:
final: cmp varI, varI
```

→ La sentencia while hace que se ejecute la instrucción i=i+1 mientras se cumple la condición, es decir, mientras i sea menor que a.

## HOJA DE EJERCICIOS DE LA CPU

→ Asumir que en la posición 101 hay un 0, que en la 102 hay un 1, que a está en la 103 y que el resultado i debe quedar en la posición 104.

6. Se pide un programa en lenguaje ensamblador de la MS para realizar una multiplicación, sabiendo que el multiplicando (a) está depositado en la posición 100 de memoria y el multiplicador (b) está en la 101. El resultado (c) debe guardarse en la posición 102 de la memoria. El programa va a requerir un contador (i) que lleve la cuenta del número de veces que se va sumando el multiplicando. Para este fin se destina la posición de memoria 103, que inicialmente tendrá un 0.

```
.ver 1

.data 15 ;dirección de inicio de los datos
dw multiplicando      2
dw multiplicador      3
dw resultado          0
dw varI 0
dw varUno             1

.code
bucle:  cmp multiplicador, varI      ;comparo multiplicador y varI
        beq final                  ;salta a final: si multiplicador y varI son iguales
        add varUno, varI            ;Añadimos 1 al contador varI
        add multiplicando, resultado ;sumar multiplicando al resultado
        cmp varI, varI             ;FZ=1
        beq bucle                  ;salto a bucle:
final:  cmp varI, varI             ;Fin de programa
```

## HOJA DE EJERCICIOS DE LA CPU

7. El factorial de un número  $a$  es el resultado de la siguiente operación:

**Factorial de  $n = 1*2*3*...*(a-1)*a$**

```
.ver 1

.data 20 ;dirección de inicio de los datos
dw multiplicando      2
dw multiplicador      1
dw resultado          0
dw tope 5
dw varI 0
dw varCero            0
dw varUno             1

.code
while: cmp multiplicador, tope
      beq end
bucle: cmp multiplicador, varI      ;comparo multiplicador y varI
      beq final                    ;salta a final: si multiplicador y varI son iguales
      add varUno, varI              ;Añadimos 1 al contador varI
      add multiplicando, resultado ;sumar multiplicando al resultado
      cmp varI, varI               ;FZ=1
      beq bucle                    ;salto a bucle:
final: mov varCero, varI
      add varUno, multiplicador
      mov varCero, multiplicando
      add resultado, multiplicando
      cmp varCero, varCero
      beq while
end:   cmp varCero, varCero
```

→