

Actividad 6

- a) MOV AX, [3423h]
- b) MOV AX, [BX]
- c) MOV AX, BX
- d) MOV AX, 324
- e) MOV AX, [BX + SI + 21]
- f) MOV AX, [BX + 15]
- g) MOV AX, [SI - 14]
- h) MOV AX, [BX + SI]

A partir de una descripción de una operación, identifica el modo de direccionamiento utilizado:

- a) Mover el valor almacenado en la dirección de memoria calculada por BX + SI + 31 al registro AX.
 - b) Mover el valor almacenado en la dirección de memoria calculada por BX + SI al registro AX.
 - c) Mover el valor almacenado en la dirección de memoria 5423h al registro AX.
 - d) Mover el valor almacenado en la dirección de memoria apuntada por BX al registro AX.
 - e) Mover el valor 45 al registro AX.
 - f) Mover el valor almacenado en la dirección de memoria calculada por BX + 23 al registro AX.
 - g) Mover el valor almacenado en la dirección de memoria calculada por SI - 56 al registro AX.
 - h) Mover el contenido del registro BX al registro AX.
-
- ☐ Escribe un programa en ensamblador que tome dos valores de 8 bits y realice una operación AND bit a bit entre ellos, almacenando el resultado en un tercer registro.
 - ☐ Escribe un programa en ensamblador que tome dos valores de 8 bits y realice una operación OR bit a bit entre ellos, almacenando el resultado en un tercer registro.
 - ☐ Escribe un programa en ensamblador que tome dos valores de 8 bits y realice una operación XOR bit a bit entre ellos, almacenando el resultado en un tercer registro.

Para cada uno de estos programas es necesario desarrollar los bloques .data, .text y _start o main

Identificación del Modo de Direcccionamiento:

a) MOV AX, [3423h]

Modo de direccionamiento directo (inmediato). Se carga el valor almacenado en la dirección de memoria 3423h en el registro AX.

b) MOV AX, [BX]

Modo de direccionamiento indirecto. Se carga el valor almacenado en la dirección de memoria apuntada por el registro BX en el registro AX.

c) MOV AX, BX

Modo de direccionamiento de registro a registro. Se mueve el contenido del registro BX al registro AX.

d) MOV AX, 324

Modo de direccionamiento directo (inmediato). Se carga el valor inmediato 324 en el registro AX.

e) MOV AX, [BX + SI + 21]

Modo de direccionamiento indexado. Se suma el contenido de los registros BX, SI y el desplazamiento 21 para calcular la dirección de memoria y se carga el valor almacenado en esa dirección en el registro AX.

f) MOV AX, [BX + 15]

Modo de direccionamiento indexado. Se suma el contenido del registro BX y el desplazamiento 15 para calcular la dirección de memoria y se carga el valor almacenado en esa dirección en el registro AX.

g) MOV AX, [SI - 14]

Modo de direccionamiento indexado. Se resta el desplazamiento 14 del contenido del registro SI para calcular la dirección de memoria y se carga el valor almacenado en esa dirección en el registro AX.

h) MOV AX, [BX + SI]

Modo de direccionamiento indexado. Se suma el contenido de los registros BX y SI para calcular la dirección de memoria y se carga el valor almacenado en esa dirección en el registro AX.

Descripción de Operaciones:

a) Mover el valor almacenado en la dirección de memoria calculada por BX + SI + 31 al registro AX.

Modo de direccionamiento indexado.

b) Mover el valor almacenado en la dirección de memoria calculada por BX + SI al registro AX.

Modo de direccionamiento indexado.

c) Mover el valor almacenado en la dirección de memoria 5423h al registro AX.
Modo de direccionamiento directo (inmediato).

d) Mover el valor almacenado en la dirección de memoria apuntada por BX al registro AX.

Modo de direccionamiento indirecto.

e) Mover el valor 45 al registro AX.

Modo de direccionamiento directo (inmediato).

f) Mover el valor almacenado en la dirección de memoria calculada por BX + 23 al registro AX.

Modo de direccionamiento indexado.


g) Mover el valor almacenado en la dirección de memoria calculada por SI - 56 al registro AX.

Modo de direccionamiento indexado.


h) Mover el contenido del registro BX al registro AX.

Modo de direccionamiento de registro a registro.

Programa en Ensamblador para AND Bit a Bit:


```
HelloWorld.asm 429ee58vq   
1 section .data  
2     value1 db 0x3F ; Primer valor de 8 bits  
3     value2 db 0xA5 ; Segundo valor de 8 bits  
4     result db 0 ; Resultado del AND bit a bit  
5  
6 section .text  
7 global _start  
8  
9 _start:  
10     mov al, [value1] ; Cargar primer valor en AL  
11     mov bl, [value2] ; Cargar segundo valor en BL  
12     and al, bl ; Realizar operación AND bit a bit  
13     mov [result], al ; Almacenar resultado en 'result'
```

Programa en Ensamblador para OR Bit a Bit

HelloWorld.asm 429ee58vq 

```
1 section .data
2     value1 db 0x3F    ; Primer valor de 8 bits
3     value2 db 0xA5    ; Segundo valor de 8 bits
4     result db 0       ; Resultado del OR bit a bit
5
6 section .text
7 global _start
8
9 _start:
10     mov al, [value1]  ; Cargar primer valor en AL
11     mov bl, [value2]  ; Cargar segundo valor en BL
12     or  al, bl        ; Realizar operación OR bit a bit
13     mov [result], al  ; Almacenar resultado en 'result'
```

Programa en Ensamblador para XOR Bit a Bit

HelloWorld.asm 429ee58vq 

```
1 section .data
2     value1 db 0x3F    ; Primer valor de 8 bits
3     value2 db 0xA5    ; Segundo valor de 8 bits
4     result db 0       ; Resultado del XOR bit a bit
5
6 section .text
7 global _start
8
9 _start:
10     mov al, [value1]  ; Cargar primer valor en AL
11     mov bl, [value2]  ; Cargar segundo valor en BL
12     xor al, bl        ; Realizar operación XOR bit a bit
13     mov [result], al  ; Almacenar resultado en 'result'
```