

Actividad 2

Christian Geovany Muñoz Rodríguez

Ingeniería en computación

Código: 221350605

Traductores de lenguajes – D04

Maestro: José Juan Meza Espinosa

Universidad de Guadalajara

**Centro Universitario de Ciencias Exactas e
Ingenierías**

01 de Febrero del 2023



Código:

`org 100h`

`jmp inicio`

`var1 db 45h ; 45h = 01000101`

`var2 db 57h ; 57h = 01010111`

inicio:

`mov al,var1 ; carga el valor de var1 en el registro AX`

`mov bl,var2 ; carga el valor de var2 en el registro BX`

`AND al,bl ; realiza una operacion logica AND entre los valores de AX y BX`

`; y guarda el resultado en AX`

`; 45h AND 57h = 01000101`

`;`
`AND`

`;`
`01010111`

`;`
`=`

`;`
`01000101 = 45h`

`OR al,bl ; realiza una operacion logica OR entre los valores de AX y BX`

`; y guarda el resultado en AX`

`; 45h OR 57h = 01000101`

`;`
`OR`

`;`
`01010111`

`;`
`=`

`;`
`01010111 = 57h`

`mov al,var1`

`XOR al,bl ; realiza una operacion logica XOR entre los valores de AX y BX`

`; y guarda el resultado en AX`

```
    ; 45h XOR 57h = 01000101
;
;          XOR
;          01010111
;          =
;          00010010 = 12h
;
```

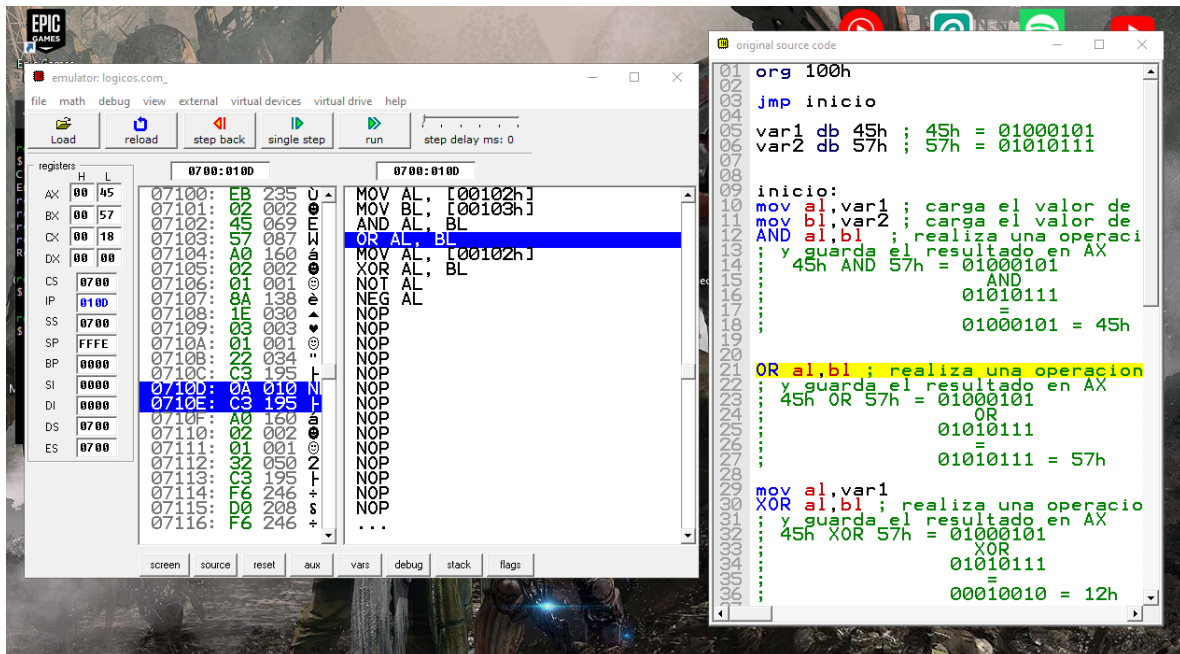
NOT al ; realiza una operacion logica NOT en el valor de AX y guarda el resultado en AX

; NOT 12h = 00010010 -> 11101101 = EDh

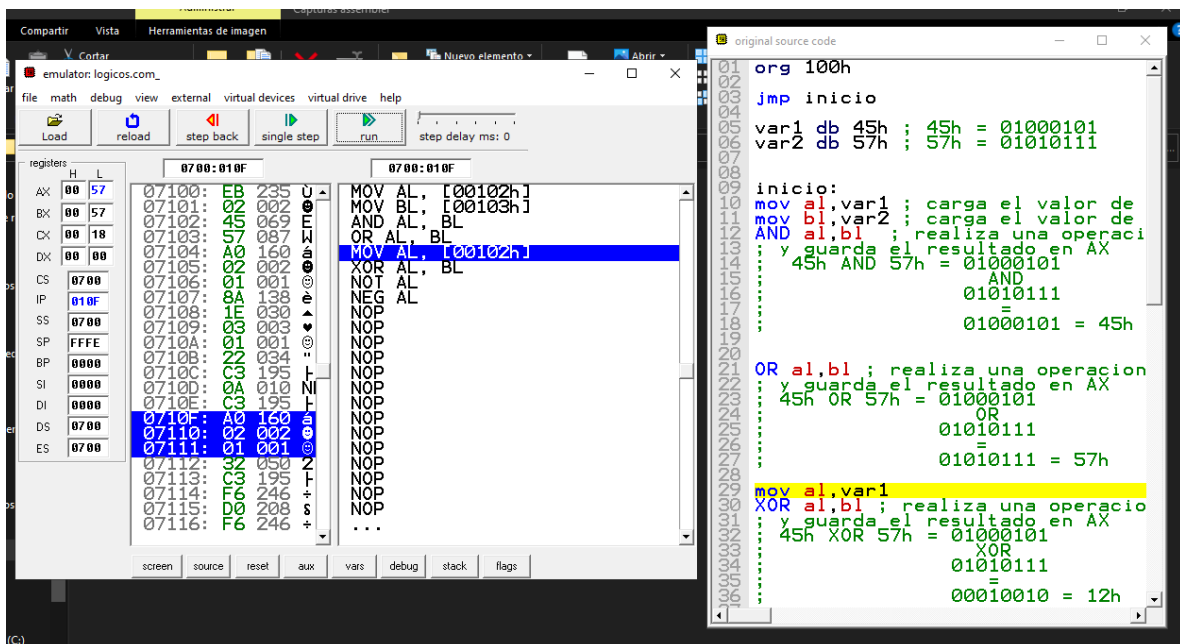
NEG al ; realiza una operacion logica NOT en el valor de AX y guarda el resultado en AX

; NEG EDh = 11101101 -> 00010011 = 13h

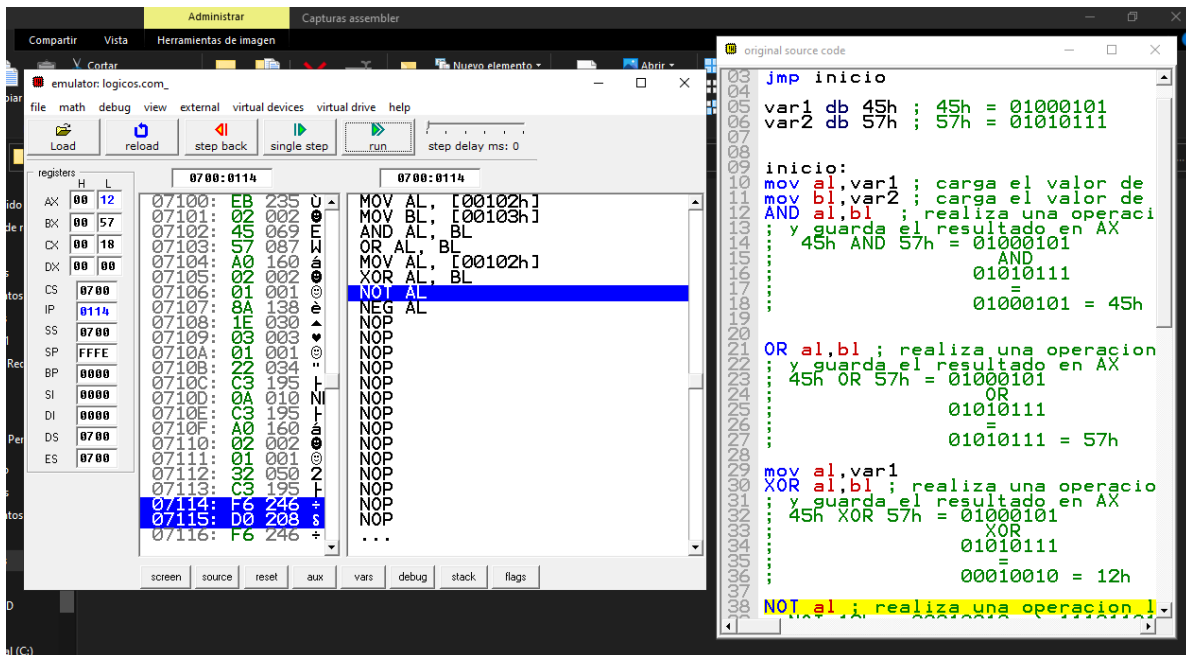
Pantallazos:



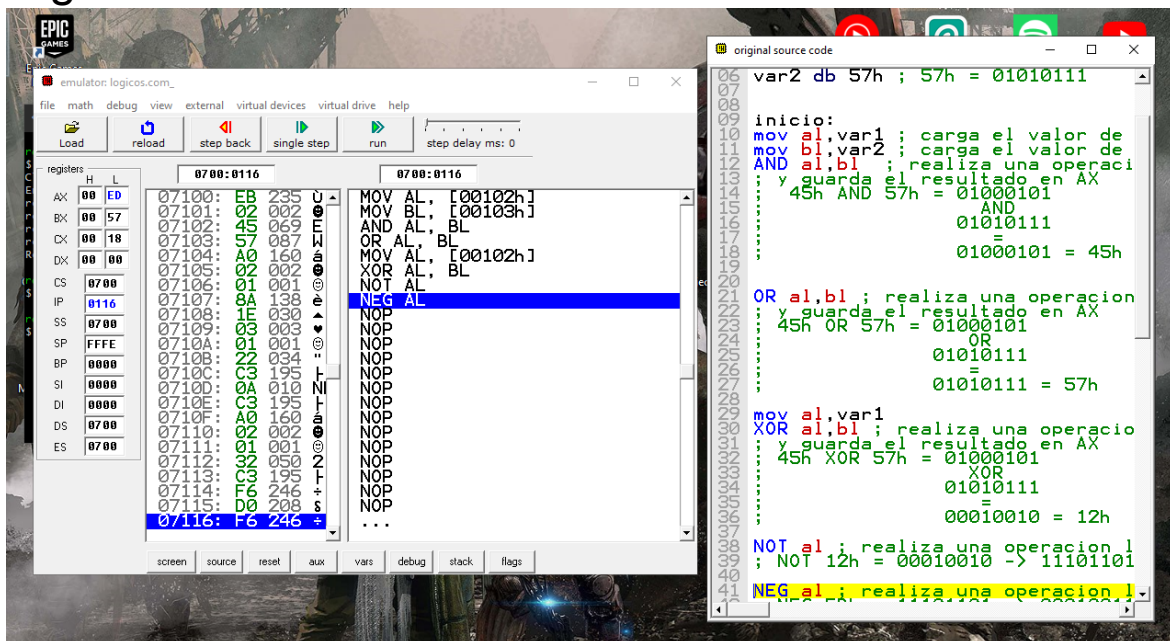
Aquí vemos como al aplicar el operador AND sobre AL y BL el registro AL conserva su valor (45h)



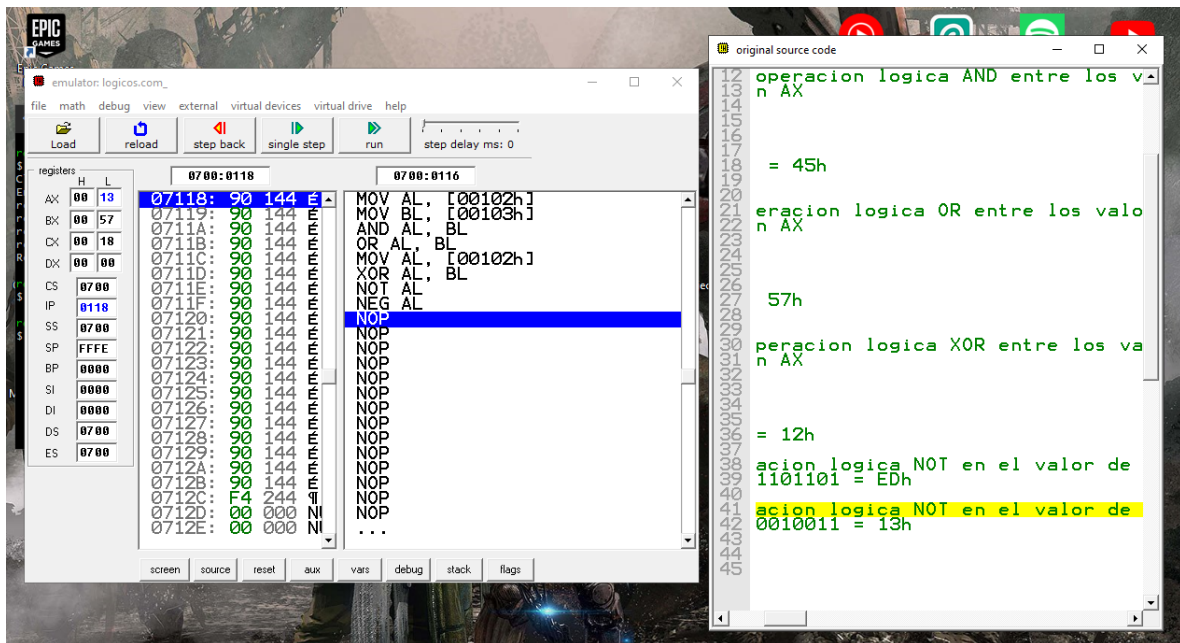
Aquí vemos como al aplicar el operador OR sobre AL y BL el registro AL toma el valor de 57h



Aquí al aplicar el operador XOR entre AL y BL el registro AL toma el valor de 12h



Al negar el registro AL toma el valor de EDh



Al aplicar el operador NEG al registro AL vuelve a tomar el valor de 12h, pero sumado +1 (13h)

Desarrollo

En esta práctica, aprendí a utilizar los operadores lógicos del lenguaje ensamblador 8086, como AND, OR, XOR, NOT y NEG. Estos operadores nos permiten realizar varias operaciones lógicas sobre los registros, permitiendo realizar operaciones lógicas más complejas y entregar un resultado útil para una amplia gama de tareas. Estos operadores se utilizan en el desarrollo de programas para computadores y nos ayudan a simplificar la tarea de realizar cálculos y tareas relacionadas con la lógica. Son útiles para resolver problemas matemáticos, crear programas complejos y realizar cálculos numéricos. Además, con estos operadores podemos controlar el flujo de un programa, realizar ciclos y validar condiciones para ejecutar ciertas partes de nuestro código, lo que nos ayuda a optimizar el rendimiento de nuestros programas y realizar operaciones complejas de manera más eficiente.

Además, estos operadores son útiles para realizar tareas como la búsqueda y la comparación de valores, lo que nos permite crear programas que se ejecutan de forma más rápida y eficiente. Esta práctica me ha permitido comprender los conceptos básicos de la lógica de los operadores lógicos y cómo utilizarlos para mejorar la eficiencia de mis programas. También, me ha ayudado a entender mejor la lógica de programación en lenguaje ensamblador y cómo aplicarla para obtener resultados óptimos.

Al principio, me encontré un poco confundido al tratar de utilizar variables de 16 bits en lugar de 8, ya que estaba acostumbrado a trabajar con variables de 8 bits. Esto

se debía a que las variables de 16 bits tienen una estructura diferente, y por lo tanto se comportan de manera distinta. Por lo tanto, decidí trabajar con variables de 8 bits, ya que me resultó más sencillo de entender y me resultó más fácil de implementar. Esta confusión me ayudó a entender mejor los dos tipos de variables y a comprender su diferencia en cuanto al comportamiento y uso.

Además, esta práctica también me ayudó a recordar los fundamentos de la conversión de números hexadecimales a decimales y viceversa. Esto me ha resultado muy útil ya que con este conocimiento puedo leer los valores de los registros y calcular los resultados de mis operaciones.

Conclusiones

En conclusión, realizar prácticas como esta es importante para familiarizarse con los lenguajes de bajo nivel, adquiriendo conocimientos fundamentales y haciéndonos conscientes de la gran cantidad de aplicaciones que se pueden desarrollar con ellos. Estos lenguajes son la base para un sin fin de programas y lenguajes de programación, y tener un mayor conocimiento de cómo es una comunicación casi directa entre el software y el hardware nos ayudará a crear programas mucho más complejos. Esta práctica también nos ayudará a comprender el funcionamiento de los lenguajes de programación de alto nivel a través de los lenguajes de bajo nivel.

Bibliografía:

Brey, B. B. (2006). *Microprocesadores Intel : 8086/8088, 80186/80188, 80286, 80386 y 80486, Pentium, procesador Pentium Pro, Pentium II, Pentium III y Pentium 4: arquitectura, programación e interfaces*. Pearson Educación.