

Práctico 2

Programas en lenguaje ensamblador

Ejercicio 1

Escribir un programa que busque en un bloque de memoria la primer aparición de un dato de 8 bits, utilizando y sin utilizar las instrucciones de transferencia y comparación de bloques. Suponer que la dirección de comienzo del bloque de memoria está en HL, el tamaño del bloque está en BC y el dato a buscar está en A. En caso de encontrar el dato el programa debe dejar el flag Z en 1 y la dirección siguiente a donde lo encontró en HL. En caso contrario debe dejar el flag Z en cero.

Ejercicio 2

Escribir un programa que implemente la operación de multiplicación de números codificados en complemento a dos. Suponer que los números son de 8 bits y están en los registros A y B. El resultado debe guardarse en el registro A.

Ejercicio 3

Escribir una rutina que compruebe un bloque de RAM, escribiéndolo primero con una disposición de prueba consistente en bytes alternados de AAh y 55h. La rutina debe leer a continuación los datos escritos en el bloque para ver si han sido almacenados adecuadamente. Si es así el programa escribe una disposición de prueba invertida y lo lee de nuevo para verificar que el almacenamiento es correcto. Si ambas disposiciones de prueba han sido correctamente almacenadas el bit 0 del puerto de salida 01h se pone a 1; si no, se pone a 1 el bit 1 del mismo puerto. Suponer que el bloque de memoria a comprobar empieza en la posición 400h y tiene 256 bytes de longitud.

Ejercicio 4

Realizar una rutina que, a partir de dos dígitos hexadecimales en ASCII que se encuentran en el byte apuntado por HL y el siguiente (dígito menos significativo en la posición más baja de memoria), devuelva en el acumulador el número binario correspondiente.

Ejercicio 5.

Escribir un algoritmo que multiplique un número natural por 10 (buscar un algoritmo sencillo para este caso particular, teniendo en cuenta que: $10*a = 8*a + 2*a$). Implementarlo en assembler de Z80.

Ejercicio 6.

Sean A y B dos parejas de dígitos decimales en BCD empaquetado (es decir los dos dígitos en 8 bits). Escribir el código assembler Z80 necesario para efectuar su suma:

- a) Sin utilizar la instrucción DAA.
- b) Utilizando la instrucción DAA.

Ejercicio 7

Escribir un programa para el Z80 que invierta el orden de los bits del byte en la dirección de memoria apuntada por IX ($b_0 \leftrightarrow b_7, b_1 \leftrightarrow b_6, b_2 \leftrightarrow b_5, b_3 \leftrightarrow b_4$).