

Fundamentos de Arquitectura de Computadores



Departamento
de Informática



Práctica 2

Ejercicios de introducción al lenguaje ensamblador

Introducción

El objetivo de esta segunda práctica evaluable es afianzar al estudiante con la programación básica en ensamblador para el caso específico del microprocesador 8085.

En nuestro caso, como ya se comentó en la práctica previa, la herramienta que utilizaremos para llevar a cabo los distintos ejercicios será el simulador en java de 8085 ***j8085sim***.

En esta segunda práctica evaluable, se propone la resolución de una serie de problemas con un grado de complejidad superior con los que aprenderemos a manejar otras operaciones más avanzadas del lenguaje, por ejemplo para el acceso a memoria principal.

Material de referencia

Sobre el software:

- Descarga gratuita del simulador ***j8085sim*** desde Platea.
- Manual de referencia del Lenguaje de Programación 8085

Los documentos citados se encuentran disponibles desde la plataforma Platea.

Fecha y modo de entrega

El código en ensamblador desarrollado para cada uno de los ejercicios se realizará en un fichero independiente **sin incluir ningún comentario dentro del código**. Cada ejercicio resuelto, estará en un fichero diferente, por lo que la práctica se compondrá de varios ficheros. Debéis hacer un fichero .zip (u otro formato de compresión) con todos estos ficheros. A continuación, dentro de la plataforma Platea, debéis adjuntar dicho fichero .zip en la actividad “Entrega de Guión de Prácticas 2” **antes** de la fecha indicada para esta actividad.

IMPORTANTE: Sólo es necesario que uno de los dos compañeros haga la entrega. No obstante, se aconseja asegurarse de que la entrega se hizo correctamente, es decir, que el fichero subido a Platea podrá descargarse sin problema el día de la defensa.

Esta práctica tendrá una calificación máxima de **2.75 sobre 10 puntos**. Es imprescindible implementar todos los ejercicios, incluidos extras, para alcanzar la máxima puntuación.

Ejercicios a resolver

1) Escribir un programa ensamblador para multiplicar dos números cualesquiera

- a) Los números pueden ser cualesquiera y se leerán desde las posiciones de memoria 0x0100 y 0x0101.
- b) Recordad que los números se expresan en hexadecimal.
- c) Toda operación que realice la ALU implica que los operandos deben estar en los registros de uso general (uno de ellos en el acumulador).
- d) Para saber si existe desbordamiento en la operación, mostrad al final el número de veces que se ha activado el flag de acarreo por el puerto 02h.
- e) La salida siempre se da por el puerto 03h.

Extra:

- i) ¿Sabrías explicar a qué corresponde numéricamente del resultado en el puerto 02?
- ii) Se premia una implementación que consiga la máxima eficiencia para la ejecución de cualquier multiplicación

2) Obtener el rango de un conjunto de números.

- a) Se ha realizado un experimento en el que se mide una determinada magnitud física, obteniendo N valores de los que algunos pueden repetirse.
- b) El resultado de dicho experimento, los N valores, se almacena en memoria a partir de la dirección 0x0100, teniéndose en la primera palabra de memoria el número de medidas N y a continuación sus valores.
- c) Realice un programa que determine el rango de las medidas realizadas, es decir, el valor del extremo inferior y superior del conjunto de datos. Mostrar dichos valores por los puertos 00h y 01h.
- d) La asignación de memoria para el programa será la siguiente:

Posición	Contenido
0x0000, ...	Programa
0x0100	Número N de elementos
0x0101	Dirección del primer número de la lista
0x0102, ...	Resto de números de la lista

Extra:

- a) Calcular la moda de la serie de datos y mostrar dicho valor en el puerto 02h.

3) Uso del criterio de paridad

- a. Existe una lista de N elementos en memoria principal a partir de la dirección 0x0000. Todos estos elementos empiezan con un bit a 0, es decir, el bit más significativo o BMS es 0.
- b. El valor N correspondiente al número total de datos en la tabla viene dado en el puerto de E/S 00h.
- c. Pueden existir valores repetidos.
- d. Hacer un programa en código ensamblador que cambie los elementos de la tabla añadiendo como BMS un bit de paridad (criterio par).
- e. La asignación de memoria y puertos de E/S para el programa será la siguiente:

Posición	Contenido
0x0100, ...	Programa
0x0000	Dirección del primer elemento de la lista
0x0001,...	Resto de elementos de la lista
00h	Números de elementos, N