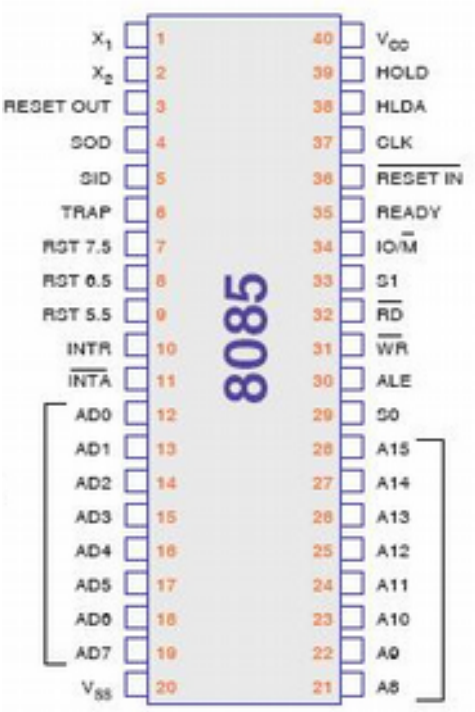
# Práctica 3

*Fundamentos de Arquitectura de Computadores*


# Práctica 3

*Ejercicios de introducción al lenguaje ensamblador*

## Introducción

El objetivo de esta tercera práctica evaluable es consolidar 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 tercera práctica, y última en la serie de prácticas con el lenguaje ensamblador 8085, se propone la resolución de una serie de problemas con un grado de complejidad superior.

## 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 3*" ***antes*** de la fecha indicada para esta actividad.

**IMPORTANTE:** 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 **3.25 sobre 10 puntos.** Es imprescindible implementar todos los ejercicios, incluidos extras, para alcanzar la máxima puntuación.

# Ejercicios a resolver

1) **Realizar un programa para el cálculo de la división de dos números.**

   a) Usar dos números de 8 bits almacenados en memoria (posiciones 0x0100 y 0x0101).
   b) El cociente debe quedar almacenado en la posición 0x0102 y el resto de la división en 0x0103
   c) Manejad todas las posibles casuísticas especiales de valores en el dividendo y el divisor, por ejemplo, cuando el dividendo o el divisor sean 0.
   d) La asignación de memoria para el programa será la siguiente:

| **Posición** | **Contenido** |
|---|---|
| 0x0000 | Programa |
| 0x0100 | Dividendo |
| 0x0101 | Divisor |
| 0x0102 | Cociente |
| 0x0103 | Resto |

2) **Realizar un programa para la búsqueda de un elemento dentro de una lista.**

   a) Tenemos en memoria una lista de números (desordenada) que contiene N datos. El valor N se especifica en el puerto de E/S 00h.
   b) El primer dato de la lista se encuentra en la posición de memoria 0x00FE.
   c) El número a buscar en la lista proporcionada en memoria se encuentra dado en el puerto 01h. Si tras la búsqueda se encuentra dicho número, se muestra en el puerto de E/S 02h su posición RELATIVA en la lista. Por ejemplo, si el número coincide con el tercer número de la lista, el puerto de E/S 02h mostrará el valor 03h.
   d) En caso de que el número esté repetido en la lista, se mostrará la posición relativa de la última coincidencia encontrada.
   e) De lo contrario, si no existe en la lista el número buscado, se muestra el valor FFh.
   f) En resumen, la asignación de memoria para el programa será la siguiente:

| **Posición** | **Contenido** |
|---|---|
| 0x0000 | Programa |
| 0x00FE | Dirección del primer elemento de la lista |
| 00h | Puerto de E/S con el número de elementos N que contiene la lista |
| 01h | Puerto de E/S con el número a buscar |
| 02h | Puerto de E/S para mostrar la posición relativa. |

**Extra:**

i) En el caso de valores repetidos para el número buscado, también es necesario mostrar por el puerto 03h el número de veces que se repite el número en la lista.

ii) En caso de que el número no estuviera en la lista, hay que insertarlo al final de la lista, es decir, en la siguiente posición de memoria siguiente al último elemento de la lista. Además, este cambio implica realizar varios ajustes: **a)** mostrar en el puerto de E/S 02h la posición relativa tras ser insertado en la lista; **b)** mostrar en el puerto de E/S 03h el número de veces que se repite el número insertado (en este caso, una vez); y **c)** mostrar por el puerto de E/S 00h el nuevo valor de N.

## 3) Búsqueda de coincidencias en una lista

a. Se trata de encontrar un número dentro de una lista de números almacenados en memoria. El número a encontrar se facilita en el puerto de E/S 00h.
b. Las direcciones de memoria de inicio y de fin de la lista se proporcionan, respectivamente, en las direcciones de memoria 0x0100+0x0101 y 0x0102+0x0103. Véase por ejemplo que la combinación del par de inicio 0x0100+0x0101 comprende 16 bits, almacenándose en 0x0100 los 8 bits LSB (menos significativos) y en 0x0101 los 8 bits MSB (más significativos) de la dirección de memoria del primer elemento de la lista en memoria.
c. Las direcciones de memoria de inicio y de fin de la lista proporcionada deben de ser superiores a la dirección 0x0103. En caso contrario, el programa mostrará el valor FFh por el puerto de E/S 01h y terminará. De la misma forma, el programa también mostrará por dicho puerto el valor FFh en caso de que la dirección de inicio de la lista sea mayor que la dirección de fin.
d. Por último, el programa deberá mostrar por el puerto de E/S 01h el número de coincidencias (veces que se repite) del número buscado.
e. La asignación de memoria y puertos de E/S para el programa será la siguiente:

| **Posición** | **Contenido** |
|---|---|
| 0x0000, … | Programa |
| 0x0100 | Parte menos significativa (LSB) de dirección de inicio |
| 0x0101 | Parte más significativa (MSB) de dirección de inicio |
| 0x0102 | Parte menos (LSB) significativa de dirección de fin |
| 0x0103 | Parte más significativa (MSB) de dirección de fin |
| 00h | Puerto de E/S con valor de dato a buscar en la lista de memoria |
| 01h | Puerto de E/S con el número de coincidencias. También para mostrar el valor FFh en caso de error. |