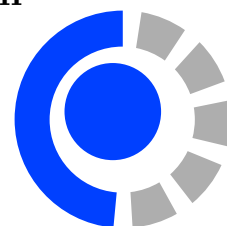


# Trabajo práctico N° 8 bis

## Programación de la MCBE, traducción, y ejecución



Introducción a la computación  
Departamento de Ingeniería de Computadoras  
Facultad de Informática - Universidad Nacional del Comahue



**Objetivo:** Comprender cómo realizar una traza de un programa en lenguaje ensamblador, y el proceso de ensamblado.

### Recursos bibliográfico:

- *Andrew S. Tanenbaum*. Organización de computadoras: un enfoque estructurado. Cuarta edición, editorial Pearson Educación, 2000. ISBN 970-170-399-5.

### Lectura propuesta:

- Apunte de la materia. *Capítulo 7: El software. Sección 7.1 y 7.2*. Disponible en: [https://ic.fi.uncoma.edu.ar/uploads/misc/apunte\\_ic.pdf](https://ic.fi.uncoma.edu.ar/uploads/misc/apunte_ic.pdf)

## Modelo Computacional Binario Elemental (MCBE)

1. Escriba un programa en ensamblador del *MCBE* que lea un número desde la entrada, y muestre por pantalla el doble de ese número.
  - a) Traduzca el programa desarrollado en el punto anterior a código máquina de *MCBE*.
  - b) Realice la traza del ejecutable creado en el paso anterior.
  - c) Modifique el programa para que se imprima el cuádruple del número ingresado ¿Cuántas líneas se modificaron en lenguaje ensamblador? ¿Cuántas líneas se vieron modificadas en el lenguaje máquina?
2. Escriba un programa en ensamblador del *MCBE* que lea dos números desde la entrada, e imprima el primero la cantidad de veces indicada por el segundo.
  - a) Traduzca el programa desarrollado en el punto anterior a código máquina de *MCBE*.
  - b) Realice la traza del ejecutable creado en el paso anterior.
3. Escriba un programa en ensamblador del *MCBE* que lea dos números desde la entrada y muestre por pantalla el resultado de su multiplicación.
  - a) Traduzca el programa desarrollado en el punto anterior a código máquina de *MCBE*.
  - b) Realice la traza del ejecutable creado en el paso anterior.

# Anexo

| Op. | Código de operación<br><i>3 bits</i> | Operando<br><i>5 bits</i> | Descripción                                                                                                                                                            |
|-----|--------------------------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LD  | 010                                  | <i>dirección</i>          | <b>Memoria → Acumulador.</b> Copia un byte desde la dirección de memoria al acumulador.                                                                                |
| ST  | 011                                  | <i>dirección</i>          | <b>Acumulador → Memoria.</b> Copia el contenido del acumulador en esa dirección de memoria.                                                                            |
| ADD | 100                                  | <i>dirección</i>          | <b>Suma.</b> El contenido de la dirección se suma al acumulador, y el resultado se almacena en el acumulador.                                                          |
| SUB | 101                                  | <i>dirección</i>          | <b>Resta.</b> El contenido de la dirección se resta al acumulador, y el resultado se almacena en el acumulador.                                                        |
| JMP | 110                                  | <i>desplazamiento</i>     | <b>Salto incondicional.</b> Se suma (en complemento a 2) el desplazamiento al <b>PC</b> .                                                                              |
| JZ  | 111                                  | <i>desplazamiento</i>     | <b>Salto condicional.</b> Si el acumulador es cero, se suma (en complemento a 2) el desplazamiento al <b>PC</b> , en caso contrario el <b>PC</b> se incrementa en uno. |
| HLT | 001                                  | <i>(sin uso)</i>          | <b>Detiene la maquina.</b> No se ejecutan nuevas instrucciones. Los registros y la memoria quedan con el último valor que tenían.                                      |
| NOP | 000                                  | <i>(sin uso)</i>          | <b>No operación.</b> No tiene ningún efecto sobre el acumulador ni memoria. El <b>PC</b> se incrementa en uno.                                                         |

**Memoria:** consta de 32 posiciones de 8 bits. Las direcciones 0 a 29 corresponden a direcciones que pueden ser escritas y leídas. La dirección 30 es de **sólo lectura**, permite leer datos del dispositivo de entrada, por ejemplo un teclado. La dirección 31 es de **sólo escritura**, permite escribir datos en el dispositivo de salida, por ejemplo en una pantalla o una impresora.

**Registro PC:** registro de 8 bits, contiene la dirección de la próxima instrucción a ejecutar. Se inicializa en cero.

**Registro IR:** registro 8 bits donde se guarda la instrucción que se esta decodificando o ejecutando.

**Registro acumulador:** registro de 8 bits donde se almacena un número entero representado en *complemento a 2*.

**Etiquetas predefinidas:**

**IN:** dirección 30, entrada, dirección de solo lectura.

**OUT:** dirección 31, salida, dirección de solo escritura.

**Instrucciones:** de 8 bits, los 3 bits más significativos almacenan el código de operación, y los 5 menos significativos almacenan el operando.

