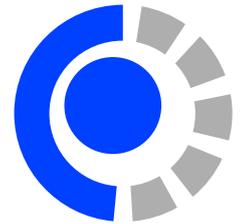


# Trabajo práctico N° 8

## Traza de código máquina en la MCBE



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



**Objetivo:** Comprender el ciclo de instrucción de una computadora simple.

### 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)

## Ejecución de un programa en la MCBE

Suponga la máquina *MCBE* en su estado inicial con contenido de memoria indicado en cada inciso (asuma por simplicidad que el resto de la memoria se inicializa en cero). Describir el efecto de la ejecución de cada una de las instrucciones del programa haciendo la traza de su ejecución. Luego explique en lenguaje natural cual es el resultado de la ejecución del programa.

1.

| Dirección | Contenido binario |
|-----------|-------------------|
| 0         | 01000111          |
| 1         | 01111111          |
| 2         | 10001000          |
| 3         | 01100111          |
| 4         | 11100010          |
| 5         | 11011011          |
| 6         | 00100000          |
| 7         | 00000010          |
| 8         | 11111111          |

Ejemplo de Resolución:

(Se utiliza el símbolo "-" para indicar que no se produjo ningún cambio)

| Búsqueda de la instrucción |          | Decodificación de la instrucción |          | Ejecución de la instrucción |                      |          |          |
|----------------------------|----------|----------------------------------|----------|-----------------------------|----------------------|----------|----------|
| PC                         | IR       | Cod. Op.                         | Operando | Acumulador                  | Memoria              | Salida   | PC       |
| 00000000                   | 01000111 | 010                              | 00111    | 00000010                    | -                    | -        | 00000001 |
| 00000001                   | 01111111 | 011                              | 11111    | -                           | -                    | 00000010 | 00000010 |
| 00000010                   | 10001000 | 100                              | 01000    | 00000001                    | -                    | -        | 00000011 |
| 00000011                   | 01100111 | 011                              | 00111    | -                           | (00000111)←-00000001 | -        | 00000100 |
| 00000100                   | 11100010 | 111                              | 00010    | -                           | -                    | -        | 00000101 |
| 00000101                   | 11011011 | 110                              | 11011    | -                           | -                    | -        | 00000000 |
| 00000000                   | 01000111 | 010                              | 00111    | 00000001                    | -                    | -        | 00000001 |
| 00000001                   | 01111111 | 011                              | 11111    | -                           | -                    | 00000001 | 00000010 |
| 00000010                   | 10001000 | 100                              | 01000    | 00000000                    | -                    | -        | 00000011 |
| 00000011                   | 01100111 | 011                              | 00111    | -                           | (00000111)←-00000000 | -        | 00000100 |
| 00000100                   | 11100010 | 111                              | 00010    | -                           | -                    | -        | 00000110 |
| 00000110                   | 00100000 | 001                              | 00000    | -                           | -                    | -        | -        |

**Descripción en lenguaje natural:** El programa imprime 2 y 1, y luego termina su ejecución.

|    | Dirección | Contenido binario |
|----|-----------|-------------------|
| 2. | 0         | 0100 0110         |
|    | 1         | 1010 1000         |
|    | 2         | 0110 0110         |
|    | 3         | 1010 0111         |
|    | 4         | 1110 0000         |
|    | 5         | 0010 0000         |
|    | 6         | 0000 1101         |
|    | 7         | 0000 1100         |
|    | 8         | 0000 0010         |

|    | Dirección | Contenido binario |
|----|-----------|-------------------|
| 3. | 0         | 01011110          |
|    | 1         | 10000101          |
|    | 2         | 10100110          |
|    | 3         | 01111111          |
|    | 4         | 00100000          |
|    | 5         | 00010100          |
|    | 6         | 00000101          |

|    | Dirección | Contenido binario |
|----|-----------|-------------------|
| 4. | 0         | 01011110          |
|    | 1         | 01100110          |
|    | 2         | 10000110          |
|    | 3         | 10100111          |
|    | 4         | 01111111          |
|    | 5         | 00100000          |
|    | 6         | 00000000          |
|    | 7         | 00000110          |

|    | Dirección | Contenido binario |
|----|-----------|-------------------|
| 5. | 0         | 01011110          |
|    | 1         | 10001011          |
|    | 2         | 01101011          |
|    | 3         | 01001001          |
|    | 4         | 10101010          |
|    | 5         | 01101001          |
|    | 6         | 11100010          |
|    | 7         | 11011001          |
|    | 8         | 00100000          |
|    | 9         | 00000100          |
|    | 10        | 00000001          |
|    | 11        | 00000000          |

6.

| Dirección | Contenido binario |
|-----------|-------------------|
| 0         | 01011110          |
| 1         | 01100111          |
| 2         | 01000110          |
| 3         | 10000111          |
| 4         | 01101000          |
| 5         | 00100000          |
| 6         | 00000101          |
| 7         | 00000000          |
| 8         | 00000000          |

# Anexo

## Descripción del Modelo Computacional Binario Elemental (MCBE)

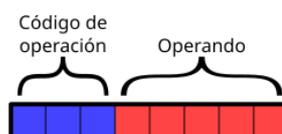
**Memoria:** consta de 32 posiciones de 8 bits. Las direcciones 0 a 29 corresponden a direcciones que pueden ser escritas y leídas. La dirección 30 es de **sólo lectura**, permite leer datos del dispositivo de entrada, por ejemplo un teclado. La dirección 31 es de **sólo escritura**, permite escribir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

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



| Código de operación<br><i>3 bits</i> | Operando<br><i>5 bits</i> | Descripción                                                                                                                                                            |
|--------------------------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 010                                  | <i>dirección</i>          | <b>Memoria → Acumulador.</b> Copia un byte desde la dirección de memoria al acumulador.                                                                                |
| 011                                  | <i>dirección</i>          | <b>Acumulador → Memoria.</b> Copia el contenido del acumulador en esa dirección de memoria.                                                                            |
| 100                                  | <i>dirección</i>          | <b>Suma.</b> El contenido de la dirección se suma al acumulador, y el resultado se almacena en el acumulador.                                                          |
| 101                                  | <i>dirección</i>          | <b>Resta.</b> El contenido de la dirección se resta al acumulador, y el resultado se almacena en el acumulador.                                                        |
| 110                                  | <i>desplazamiento</i>     | <b>Salto incondicional.</b> Se suma (en complemento a 2) el desplazamiento al <b>PC</b> .                                                                              |
| 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. |
| 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.                                      |
| 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.                                                         |