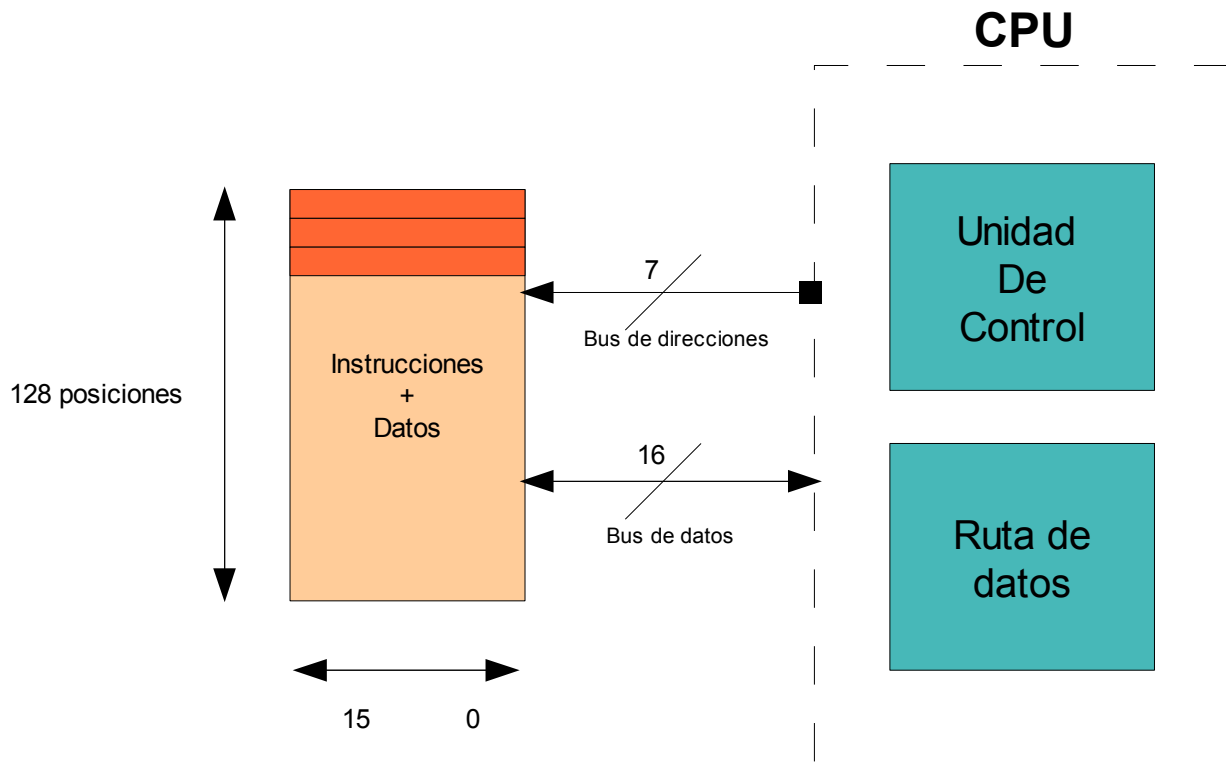


# 1) INTRODUCCIÓN A LA MÁQUINA SENCILLA

La máquina sencilla (MS) es un computador concebido para fines exclusivamente didácticos. MS sólo dispone de dos bloques: **Unidad Central de Proceso y memoria principal**. La CPU contiene unidad de control y ruta de datos. La memoria principal consta de 128 posiciones de 16 bits cada una, que sirve para almacenar instrucciones y datos.



## 2) REPERTORIO DE INSTRUCCIONES

La simplicidad de la MS llega al extremo de que su Unidad de Control sólo es capaz de interpretar y ejecutar cuatro instrucciones máquina.

1. **SUMAR** el contenido de dos posiciones de memoria.
2. **MOVER** el contenido de una posición de memoria a otra.
3. **COMPARAR** los contenidos de dos posiciones de memoria.
4. **SALTAR** a la instrucción cuya dirección se indica, si el resultado de la operación anterior ha sido cero.

Además, la MS sólo dispone de un modo de direccionamiento, el DIRECTO. Los operandos y los resultados que manipula la MS se almacenan en memoria y la única forma de referenciarlos es proporcionando su ubicación.

## 2.1) SUMAR

ADD Fuente, Destino -----> (D) = (D)+(F)

La MS posee un señalizador o “flag”, llamado FZ, que es un bit contenido en un registro, que carga con un 1 cuando el resultado de la operación que se ha ejecutado es cero. En caso contrario FZ=0. En el caso de la suma:

$$\begin{aligned} \text{FZ} &= 1 \text{ si } (D)+(F) = 0 \\ \text{FZ} &= 0 \text{ si } (D)+(F) \neq 0 \end{aligned}$$

## 2.2) MOVER

MOV Fuente, Destino -----> (D) = (F)

Al ejecutar esta instrucción también queda afectado el FZ, de forma que, cuando se transfiere es un 0 entonces FZ = 1.

$$\begin{aligned} \text{FZ} &= 1 \text{ si } (F) = 0 \\ \text{FZ} &= 0 \text{ si } (F) \neq 0 \end{aligned}$$

## 2.3) COMPARACIÓN

CMP Fuente, Destino

Compara el contenido de 2 posiciones de memoria, produciendo:

$$\begin{aligned} \text{FZ} &= 1 \text{ si } (D) = (F) \\ \text{FZ} &= 0 \text{ si } (D) \neq (F) \end{aligned}$$

## 2.4) SALTO

BEQ Destino

En esta instrucción se empieza comprobando el valor de FZ. Si vale 1, se pasa a ejecutar la instrucción apuntada por el operando D.

$$\begin{aligned} \text{Si } \text{FZ} &= 1, \text{ entonces } \text{PC} = \text{D} \\ \text{Si } \text{FZ} &= 0, \text{ entonces } \text{PC} = \text{PC}+1 \end{aligned}$$

# EJERCICIOS

1.- Después de ejecutar el siguiente programa en la MS:

```
MOV 12,13
MOV 13,14
ADD 15,13
```

12	00000000 11111111
13	11111111 00000000
14	10101010 10101010
15	01010101 01010101

a) ¿Cuál es el contenido de las posiciones de la figura?

**Solución:**

```
12= 00FF
13=5654
14=00FF
15= 5555
```

b) ¿Cuál es el valor de FZ?

**Solución:**

FZ=0

2.- Sabiendo que los contenidos en la figura se obtienen después de haber ejecutado el programa en la MS:

36	00000000 11111111
37	00000000 11111111
38	11111111 11111111
39	11111111 11111111
3A	00000000 11111111

```
MOV 36, 37
ADD 37,38
MOV 38,39
CMP 39, 3A
```

a) Averiguar el valor inicial que tenía la posición 38 antes de ejecutar el programa.

**Solución:**

38=FF00

b) ¿Cuál es el valor de FZ tras la ejecución del programa?

**Solución:**

FZ=0

3.- Se pide un programa en lenguaje ensamblador de la MS para realizar una multiplicación, sabiendo que el multiplicando (a) está depositado en la posición 100 de memoria y el multiplicador (b) está en la 101. El resultado (c) debe guardarse en la posición 102 de la memoria.

El programa va a requerir un contador (i) que lleve la cuenta del número de veces que se va sumando el multiplicando. Para este fin se destina la posición de memoria 103, que inicialmente tendrá un 0.

El programa necesitará la participación de dos constantes, que también se depositan en dos direcciones de memoria. Así, en la dirección 104 hay almacenado un 1 y en la dirección 105 un 0.

**Solución:**

- Se hará sumando el multiplicando consigo mismo tantas veces como diga el multiplicador.
- Se inicializan a 0 tanto el contador como el resultado acumulado.
- El programa consistirá en un bucle que se repite mientras no se acabe el proceso.
- El bucle suma el multiplicando al resultado parcial y lo almacena en el mismo lugar.
- Utiliza instrucciones de comparación y correspondiente salto para tomar las decisiones.
- Como no tiene salto incondicional, realiza en su lugar un salto si se cumple una condición obvia.

El programa detallado es el siguiente:

Dirección	Etiqueta	Nemónico	Fuente	Destino	Comentarios
0	begin	MOV	105	102	c=0
1		MOV	105	103	i=0
2	while	CMP	103	101	mientras i<b
3		BEQ	8	8	si FZ=1, PC=8
4		ADD	100	102	c=c+a
5		ADD	104	103	i=i+1
6		CMP	50	50	siempre FZ=1
7		BEQ	2	2	salto a PC=2
8	end				

4.- Obtener los formatos binarios que corresponden a las instrucciones del siguiente programa de la MS.

```
MOV 105,106
CMP 106,107
BEQ 100
```

**Solución:**

COP	Fuente	Destino
"01"	1101001	1101010
"10"	1101010	1101011
"11"	XXXXXXXX	1100100

5.- El formato de instrucciones de un computador tiene 3 campos: COP, Fuente, destino. El computador dispone de un banco de registros con 8 registros (R0...R7) de 8 bits cada uno. Si inicialmente los registros están cargados con el valor 0000 0010.

a) Calcular el contenido de los mismos después de ejecutar el siguiente programa, en el cual el código de operación se expresa con nemónicos y operandos por el nombre de los registros, pues sólo se dispone del modo de direccionamiento por registro.

```
ADD R2, R3
ADD R0, R1
SUB R4, R5
NOT R6
XOR R5, R7
```

**Solución:**

R0 = R2 = R4 = R7 = 0000 0010

R1 = 0000 0100; R3 = 0000 0100; R5 = 0000 0000; R6 = 1111 1101;

b) Confeccionar un programa para que el computador del ejemplo anterior, partiendo del mismo estado inicial de sus 8 registros, es decir, que contengan el valor 0000 0010, pasen a contener el valor 0000 0000, después de ejecutar el programa pedido.

**Solución (una posible):**

Nemónico	Operandos (F,D)	Comentarios
SUB	R1, R0	R0 = 0000 0000
AND	R0, R1	R1 = 0000 0000
AND	R0, R2	R2 = 0000 0000
AND	R0, R3	R3 = 0000 0000
AND	R0, R4	R4 = 0000 0000
AND	R0, R5	R5 = 0000 0000
AND	R0, R6	R6 = 0000 0000
AND	R0, R7	R7 = 0000 0000