

# la unidad de control (I)

## ANEXO

---

### 8.1 Introducción

**Definición de Instrucción Máquina:** Conjunto de dígitos binarios que indican a la CPU del ordenador la operación a realizar y la ubicación de los operandos.

El conjunto de dígitos que forma una instrucción se puede dividir en dos partes: **Código de Operación** y **Campo/s de Dirección**:

#### Instrucción Máquina

Código de Operación	Campo de Dirección 1	Campo de Dirección 2	...
---------------------	----------------------	----------------------	-----

**Definición de Código de Operación:** Parte de la instrucción máquina (subconjunto de dígitos binarios) que codifican la operación.

**Definición de Campo de Dirección:** Subconjunto de dígitos de la instrucción que indican la ubicación de los operandos o dónde debe almacenarse el resultado.

---

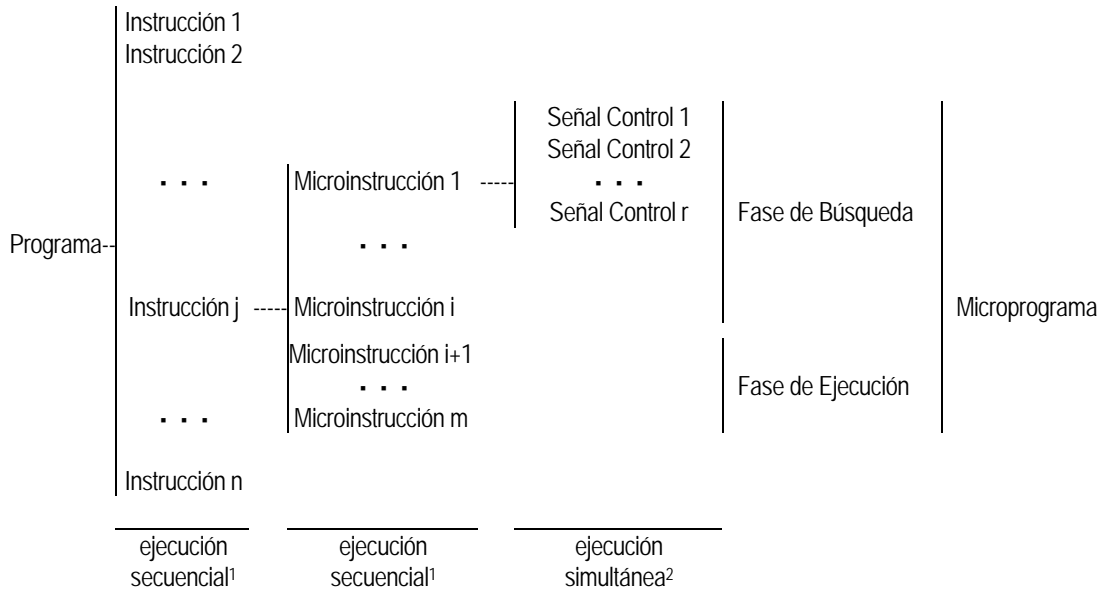
### 8.2.3 El secuenciador central

La unidad de control cableada es la que incorporan los denominados microprocesadores RISC (Conjunto Reducido de Instrucciones), mientras que los denominados microprocesadores CISC (Conjunto Completo de Instrucciones) usan unidad de control microprogramada.

En esta asignatura sólo vamos a ver la unidad de control cableada.

### 8.3 Resumen de algunos conceptos importantes

La relación entre los conceptos más importantes vistos hasta aquí puede resumirse en el siguiente esquema:



<sup>1</sup>Ejecución secuencial significa que las instrucciones o las microinstrucciones se ejecutan de manera consecutiva, es decir, hasta que no acaba una no es iniciada la siguiente.

<sup>2</sup>Ejecución simultánea significa que las señales de control se lanzan a la vez, por lo que una microinstrucción durará, como mínimo, lo que dure su señal de control o microorden más lenta

Este esquema muestra la relación entre los principales conceptos estudiados hasta aquí en este tema. Como se indica, un programa se compone de instrucciones, estas de microinstrucciones que se dividen en dos fases la de búsqueda y la de ejecución. Al conjunto de microinstrucciones que ejecutan una instrucción se le llama microprograma. Por último, una microinstrucción se compone de órdenes más elementales denominadas señales de control o microórdenes.

### 8.5 Ejemplo 2 de U.C. cableada

#### 8.5.1.- Características Generales

- Tamaño instrucción máquina: 8 dígitos binarios
- Tamaño de palabra de memoria: 8 dígitos binarios
- Ciclo máquina: 1 ciclo de reloj para todas las microinstrucciones

#### 8.5.2.- Descripción de las señales de control o microórdenes

**IPC.-** Incrementa en uno el valor del registro PC ( $PC \leftarrow PC+1$ ).

**TPC.-** El contenido del registro PC se transfiere al registro MAR.

**TIR.-** La parte del campo de dirección (6 dígitos menos significativos) de la instrucción almacenada en el registro de instrucción, se transfiere al registro MAR.

**TB.-** El contenido del Bus de datos es transferido al registro de instrucción.

**E.-** Se habilita (selecciona) la memoria.

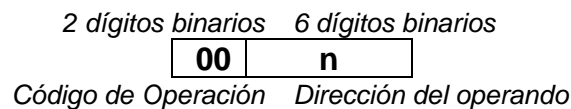
- R/W.-** Señal de control que le indica a la memoria la operación a realizar. La simbología usada va a ser: si se escribe  $R/W$  significa que la operación es lectura, si se escribe  $\overline{R/W}$  significa que la operación es escritura.
- W.-** El contenido del Bus de datos es transferido al Registro de Complementación e Incrementación (RIC).
- I.-** Se incrementa en una unidad el contenido del Registro de Complementación e Incrementación ( $RIC \leftarrow RIC+1$ ).
- C.-** Se complementa el contenido del Registro de Complementación e Incrementación.
- W<sub>A</sub>.-** La salida del Sumador Paralelo es transferida al registro Acumulador (ACC).
- R<sub>A</sub>.-** El contenido del registro ACC es transferido al Bus de datos.

### 8.5.3.- Juego de Instrucciones de la Máquina

Vamos a crear un computador capaz de sumar, restar y almacenar el resultado en memoria, es decir, una calculadora sencilla. Para esto, definimos el conjunto de las 4 instrucciones que va a ser capaz de ejecutar:

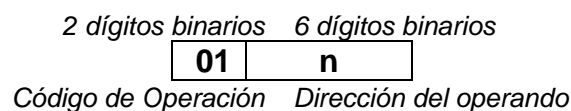
- **Instrucción 1**

- Descripción: Sumar el contenido del ACC con el complemento a 2 del contenido de una dirección de memoria n. El resultado se almacena en el ACC. En definitiva esta instrucción resta el contenido del ACC del contenido de la posición de memoria n, dejando el resultado en el ACC.
- Simbólicamente:  $ACC \leftarrow ACC + C_2 (n)$
- Formato de instrucción:



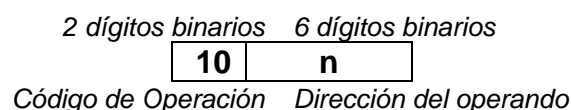
- **Instrucción 2**

- Descripción: Sumar el contenido del ACC con el de una dirección de memoria n. El resultado va al ACC.
- Simbólicamente:  $ACC \leftarrow ACC + (n)$
- Formato de instrucción:



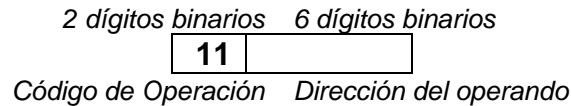
- **Instrucción 3**

- Descripción: Escribir el contenido del ACC en una dirección de memoria n.
- Simbólicamente:  $(n) \leftarrow ACC$
- Formato de instrucción:



- **Instrucción 4**

- Descripción: Parar la máquina.
- Simbólicamente: FIN
- Formato de instrucción:



### 8.5.4.- Microprogramas

Los microprogramas de cada una de las instrucciones anteriores son:

- **Instrucción 1:**  $ACC \leftarrow ACC + C_2 (n)$

Fase	Paso	Microinstrucción
Fase de búsqueda	1.-	TPC ( <i>pasamos la dirección de memoria de la instrucción a ejecutar al MAR</i> )
	2.-	IPC, ( <i>se actualiza el valor del PC para que apunte a la siguiente instrucción a ejecutar</i> ) E, R/W, ( <i>el contenido de la posición de memoria seleccionada pasa al Bus</i> ) TB ( <i>la información que circula por el Bus, que es la instrucción, es almacenada en el registro de instrucción</i> )
Fase de ejecución	3.-	TIR ( <i>pasamos la dirección del operando al MAR</i> )
	4.-	E, R/W, W ( <i>se lee el operando de la memoria central y se almacena en el RIC</i> )
	5.-	C
	6.-	I ( <i>complementar e incrementar implica realizar el complemento a 2</i> )
	7.-	$W_A$

- **Instrucción 2:**  $ACC \leftarrow ACC + (n)$

Fase	Paso	Microinstrucción
Fase de búsqueda	1.-	TPC
	2.-	IPC, E, R/W, TB
Fase de ejecución	3.-	TIR
	4.-	E, R/W, W
	5.-	$W_A$

- **Instrucción 3:**  $(n) \leftarrow ACC$

Fase	Paso	Microinstrucción
Fase de búsqueda	1.-	TPC
	2.-	IPC, E, R/W, TB
Fase de ejecución	3.-	TIR
	4.-	$R_A, E, \overline{R/W}$ (el contenido del ACC pasa al Bus y de ahí a la memoria)

- **Instrucción 4:** Fin

Fase	Paso	Microinstrucción
Fase de búsqueda	1.-	TPC
	2.-	IPC, E, R/W, TB
Fase de ejecución	3.-	Fin

### 8.5.5.- Ejemplo de Programa

Descripción del problema a resolver: realizar un programa que reste el contenido de la posición de memoria 43 del contenido de la posición de memoria 32, almacenando el resultado en el ACC.

Simbólicamente:  $ACC \leftarrow (32) - (43)$ .

- **Algoritmo**

1. Poner el ACC a 0.
2. Sumar el ACC con el contenido de la posición de memoria 32
3. Restar el ACC del contenido de la posición de memoria 43

- **Programa**

10	n	\	
00	n	/	→ Ponemos el ACC a 0
01	100000		ACC ← (32)
00	101011		ACC ← (32) - (43)
11			Fin