

# ORGANIZACION DE COMPUTADORAS

## RESUMEN PARA FINAL 2023

### Modelo de Von Neumann

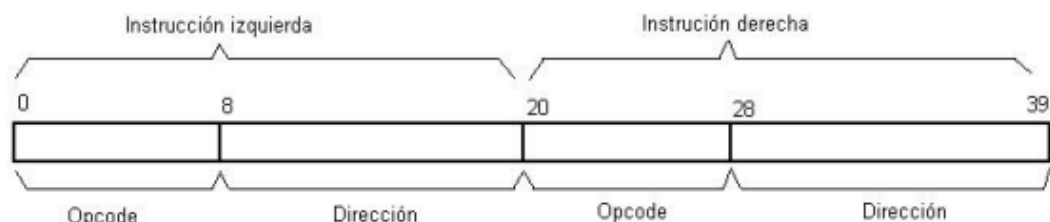
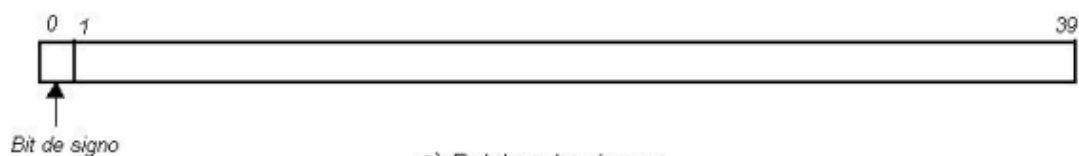
Concepto del programa-almacenado: el programa se representa de forma adecuada para que pueda ser guardado en memoria junto con los datos. Entonces, un computador puede conseguir sus instrucciones leyendolas en memoria y se puede hacer o modificar programas colocando valores en una zona de memoria.

- El principio relevante de la arquitectura von Neumann es que en la memoria se almacenan tanto los datos como las instrucciones y se tratan de igual manera, lo que significa que las instrucciones y los datos son direccionales.
- Las instrucciones son obtenidas por la CPU desde la memoria. La CPU luego decodifica y ejecuta estas instrucciones. El resultado es almacenado de nuevo en la memoria luego que se complete el ciclo de ejecución de las instrucciones.

#### Componentes Principales:

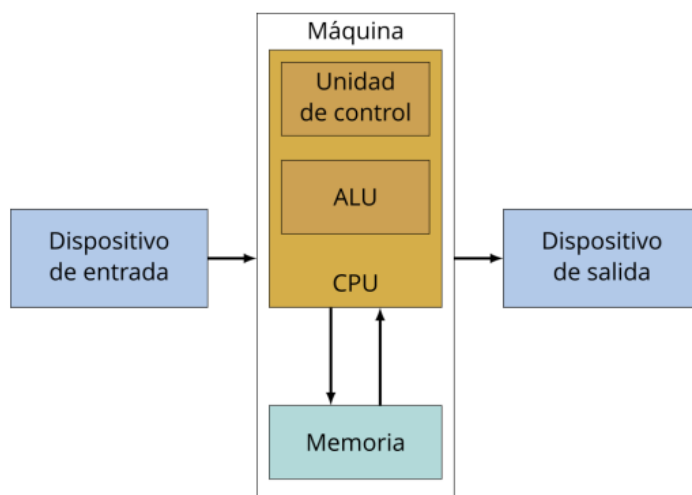
- Unidad de entrada-salida: provee las instrucciones y los datos. Se envían los datos.
- Unidad de memoria: donde se almacenan datos e instrucciones.
  - Memoria principal que almacena tanto datos como instrucciones, con 4096 palabras (posiciones de almacenamiento) de 40 bits (dígitos binarios). Se almacenan los datos y las instrucciones, ambas de forma binaria.
  - Cada número se representa con 1 bit de signo y 39 bits de valor.
  - Una palabra también puede contener 2 instrucciones (cada instrucción compuesta por 8 bits con el código de operación, más 12 bits con la dirección) de 20 bits.

Formato de memoria de IAS

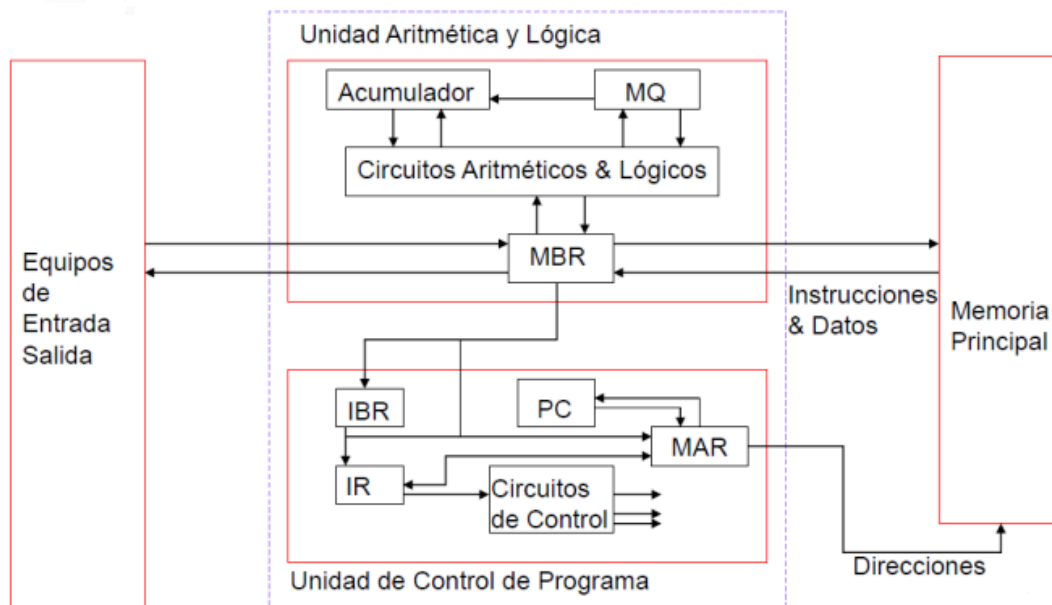


- La computadora tendrá una memoria que puede contener datos, así como el programa que procesa esos datos. En las computadoras modernas esta memoria es la RAM o memoria principal. Esta memoria es rápida y accesible directamente por la CPU.

- La RAM se divide en celdas. Cada celda consta de una dirección y su contenido. La dirección identificará de forma única cada ubicación en la memoria.
- Unidad central de procesamiento (CPU):
  - Unidad aritmetico-logica (ALU): procesa los datos. Capaz de hacer operaciones con datos binarios.
    - Esta parte de la arquitectura está involucrada únicamente en la realización de operaciones aritméticas y lógicas sobre los datos.
    - Están disponibles los cálculos habituales de sumar, multiplicar, dividir y restar, pero también estarán disponibles las comparaciones de datos como 'mayor que', 'menor que', 'igual a'.
  - Unidad de control: dirige la operación. Interpreta las instrucciones en memoria y provoca su ejecución.
    - Controla el funcionamiento de la ALU, la memoria y los dispositivos de entrada/salida de la computadora, indicándoles cómo actuar ante las instrucciones del programa que acaba de leer desde la memoria.
    - La unidad de control gestionará el proceso de mover los datos y programas desde y hacia la memoria. También se ocupará de ejecutar las instrucciones del programa, una a la vez o secuencialmente. Esto incluye la idea de un registro para contener los valores intermedios.
  - Registros:
    - Son áreas de almacenamiento de alta velocidad en la CPU. Todos los datos deben almacenarse en un registro antes de poder procesarse.
    - El registro de direcciones de memoria contiene la ubicación de memoria de los datos a los que se debe acceder. El registro de datos de memoria contiene los datos que se transfieren a la memoria.
- Bus:
  - La información debe fluir entre las diferentes partes de la computadora. En una computadora con la arquitectura von Neumann, la información se transmite de un dispositivo a otro a lo largo de un bus, conectando todas las unidades de la CPU a la memoria principal.
  - El bus de direcciones transporta las direcciones de los datos, pero no los datos, entre el procesador y la memoria.
  - El bus de datos transporta los datos entre el procesador, la memoria y los dispositivos de entrada-salida.



- Tanto la unidad de control como la ALU contiene posiciones de almacenamiento llamadas registros. Los registros pueden ser: MBR, MAR, IR, IBR, PC, MQ.



#### Aspectos Importantes:

- Utiliza el sistema binario. Esto simplifica la implementación de funciones y disminuye la probabilidad de fallos.
- Las instrucciones y datos residen en la memoria. Esto permite la ejecución de programas en forma secuencial, lo que aumenta la velocidad.
- La memoria es direccionable por localidad sin importar el dato almacenado.
- La unidad de control recupera los datos y las instrucciones de la misma manera desde la memoria. Por tanto, el diseño y desarrollo de la unidad de control está simplificado, siendo más barato y más rápido.
- El diseño del chip del microcontrolador es mucho más sencillo, ya que se accederá a una sola memoria. Lo más importante del microcontrolador es el acceso a la RAM y en la arquitectura von Neumann esta se podrá usar tanto para almacenar datos como para almacenar instrucciones del programa.

#### Concepto de programa:

- Antes: se trataba de una programación en hardware, cuando se cambiaban las tareas había que cambiar el hardware.
  - Se aplicaban una secuencia de funciones aritmético-lógicas sobre los datos para obtener resultados.
- Ahora: se trata de una programación en software, en cada paso se efectúa alguna operación sobre los datos.
  - El intérprete de instrucción procesa el código de instrucción que le llega y manda señales de control a las funciones aritmético-lógicas, a las cuales les llegan datos para transformarlos en resultados.
  - Para cada paso se necesita un nuevo conjunto de señales de control. Las instrucciones proporcionan esas señales de control.
  - Aparece el nuevo concepto de preprogramación. No hay que cambiar el hardware. ◦
- Programa: es una secuencia de pasos, donde en cada paso se hace una operación aritmético-lógica. Para cada operación se necesitan diferentes señales de control (la unidad de control saca información de cada instrucción).

## Historia de los computadores

La clasificación de los computadores en generaciones basándose en la tecnología hardware empleada fue ampliamente aceptada. Cada nueva generación se caracteriza por la mayor velocidad, mayor capacidad de memoria y menor tamaño que la generación anterior.

---

Generación	Fechas aprox.	Tecnología	Velocidad (operaciones por seg)
1	1946-1957	Válvulas	40.000
2	1958-1964	Transistores	200.000
3	1965-1971	Pequeña y media integración (>100 componentes x chip)	1.000.000
4	1972-1977	Gran integración (100-3.000 componentes x chip)	10.000.000
5	1978-1991	Alta integración (3.000-100.000 componentes x chip)	100.000.000
6	1992-	Ultra alta integración (100.000-100 millones componentes x chip)	1.000.000.000

---

## Teorema Fundamental de la Numeración

Teorema que relaciona una cantidad expresada en cualquier sistema de numeración posicional con la misma cantidad expresada en el sistema decimal. Siendo  $N^\circ$  el valor decimal de una cantidad expresada en base B y con  $(n+1+m)$  dígitos en posiciones i:  $N^\circ = (\text{dígito})_i \times (\text{base})^i$

### Decimal a binario

Cada dígito de un número binario tiene un valor que depende de su posición. Los valores fraccionarios se representan con potencias negativas de la base 2.

$$10 \text{ (binario)} = (1 \times 2^1) + (0 \times 2^0) = 2 + 0 = 2 \text{ (decimal)}$$

$$\begin{aligned} 1001,101 \text{ (binario)} &= \\ (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3}) &= \\ 8 + 0 + 0 + 1 + 0,5 + 0 + 0,125 &= 9,625 \text{ (decimal)} \end{aligned}$$

n	$2^n$
-6	0,015625
-5	0,03125
-4	0,0625
-3	0,125
-2	0,25
-1	0,5
0	1

n	$2^n$
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096

### Hexadecimal a binario

BCH: A cada combinación posible de 4 dígitos binarios se asocia un símbolo (se divide el número binario en subcadenas de 4 bits).

0000 = 0	0100 = 4	1000 = 8	1100 = C
0001 = 1	0101 = 5	1001 = 9	1101 = D
0010 = 2	0110 = 6	1010 = A	1110 = E
0011 = 3	0111 = 7	1011 = B	1111 = F

### Representación sin signo

Si el número tiene n bits, se puede representar  $2^n$  números distintos. El rango va de  $[0, (2^n - 1)]$ . La cantidad de representaciones distintas depende del número de bits.

$$n=3 \rightarrow 2^n = 2^3 = 8 \rightarrow [0, 7]$$

Decimal	Representación sin signo		
	bit 1	bit 2	bit 3
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

BCD: los dígitos decimales se convierten uno a uno en binario (1 dígito decimal = 4 bits). C (1100) representa el signo + y D (1101) representa el signo -

0000 = 0	0101 = 5
0001 = 1	0110 = 6
0010 = 2	0111 = 7
0011 = 3	1000 = 8
0100 = 4	1001 = 9

- Número desempquetado: 1 byte por dígito (8 bits).
  - Sin signo: Por cada dígito se usan 8 bits, 4 para el binario puro y 4 se completan con 1.

$$834 = 11111000 \ 11110011 \ 11110100$$

$$= F8 \ F3 \ F4$$

- Con signo: Igual que sin signo pero los 4 bits que acompañan al último dígito son reemplazados por el signo (en vez de 1111).

$$+ 834 = 11111000 \ 11110011 \ 11000100$$

$$= F8 \ F3 \ C4$$

- Número empaquetado: 4 bits por dígito.
  - Sin signo: Por cada dígito se usan 4 bits.
  - Con signo: Se agregan 4 bits al final que tienen el signo (C o D).

$$+ 834 = 10000011 \ 01001100$$

$$= 83 \ 4C$$

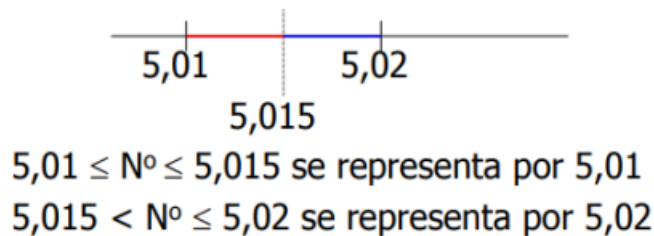
$$- 34 = 00000011 \ 01001101$$

$$= 03 \ 4D$$

Números en punto fijo: se considera que todos los números a representar tienen exactamente la misma cantidad de dígitos y la coma fraccionaria está siempre ubicada en el mismo lugar.

- Rango: diferencia entre el número mayor y el menor.

- Resolución: diferencia entre dos números consecutivos.
- Al convertir un numero decimal a sistema binario tendremos 2 casos:
  - Sin restriccion en la cantidad de bits a usar.
  - Con restriccion.
- Pueden traer error → valor a convertir - valor más cercano a representar en binario. El máximo error cometido en una representación puede considerarse como la mitad de la diferencia (resolución) entre dos números consecutivos.



## Representación con signo

Representación signo-magnitud: con n bits, 1 bit representa al signo y n-1 bits a la magnitud. El bit n-1 (extremo izquierdo) representa sólo al signo.

- Un 0 en el bit de signo indica que el número es positivo y 1 que es negativo.
- Los bits [0, n-2] representan el valor absoluto en binario.
- El rango:  $[-(2^{n-1} - 1), +(2^{n-1} - 1)]$  con 2 ceros (0+ y 0-).
- El intervalo es simetrico. El primer bit solo indica el signo. Hay  $2^n$  cantidad de números distintos.

➤ Ejemplo: n=8 bits

Números negativos	11111111	← $-(2^{n-1} - 1) = -127$
	...	
	10000000	← - 0
Números positivos	01111111	← $+(2^{n-1} - 1) = +127$
	...	
	00000000	← +0

Técnica de complementos:

- Complemento a la base disminuida (Complemento a 1 ó Ca1): Si el número es positivo, los n bits tienen la representación binaria del número (empiezan con 0). Si el número es negativo, los n bits tienen el Ca1 del valor deseado. El Ca1 de un número en base 2 (binario) se obtiene invirtiendo todos los bits (empiezan con 1).
  - El peso que tiene el primer dígito ahora es  $-(2^{n-1} - 1)$  y el resto de los dígitos con pesos positivos.
  - Por definición →  $Ca1 = (b^n - 1) - N^\circ$
  - El rango:  $[-(2^{n-1} - 1), +(2^{n-1} - 1)]$  con 2 ceros (0+ y 0-).
  - El intervalo es simetrico. Los positivos empiezan con 0 y los negativos con 1. Hay  $2^n$  cantidad de números distintos.

➤ Ejemplo:  $n=8$  bits

Números negativos	11111111	← -0
	...	
	10000000	← $-(2^{n-1}-1) = -127$
Números positivos	01111111	← $+(2^{n-1}-1) = +127$
	...	
	00000000	← +0

- Complemento a la base (Complemento a 2 ó Ca2): Si el número es positivo, los  $n$  bits tienen la representación binaria del número (empieza con 0). Si el número es negativo, los  $n$  bits tienen el Ca1 del valor deseado. El Ca1 de un número en base 2 (binario) se obtiene invirtiendo todos los bits y luego sumándole 1 (empiezan con 1).
  - Desde la derecha se escribe el número (base 2) igual hasta el primer "1" uno inclusive y luego se invierten los demás dígitos.

$$+32_{10} = 00100000 \quad \leftarrow \text{"mirando" desde la derecha}$$

$$-32_{10} = 11100000$$

- Por definición  $\rightarrow \text{Ca2} = b^n - N^\circ$
- El rango:  $[-(2^{n-1}), +(2^{n-1} - 1)]$  con 1 cero.
- El intervalo es asimétrico (hay un negativo más).

➤ Ejemplo :  $n=8$  bits

Números negativos	11111111	← -1
	...	
	10000000	← $-(2^{n-1}) = -128$
Números positivos	01111111	← $+(2^{n-1}-1) = +127$
	...	
	00000000	← +0

Técnica del exceso: La representación de un número  $A$  es la que corresponde a la SUMA del mismo y un valor constante  $E$  (o exceso)  $\rightarrow$  Exceso  $E$  de  $A = A + E$

- El signo del número  $A$  resulta de una resta. En binario, NO sigue la regla del bit más significativo.
- Rango de Exceso  $2^{(n-1)}$ :  $[-2^{(n-1)}, +2^{(n-1)}-1]$

si  $n=6$       Exceso 32

$$-2^{(6-1)} = 000000_2$$

$$= 0 - 32$$

$$2^{(6-1)}-1 = 111111_2$$

$$= 63 - 32 = 31$$

$$0_{10} = 100000_2$$

$$= 32 - 32 = 0$$

## Números en coma flotante



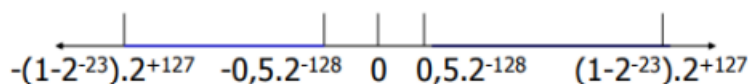
**Mantisa-Exponente:** Un número se puede representar de la forma:  $\pm M \times B^{\pm E}$ . M mantisa (parte más significativa) y E exponente. La base B es implícita (2) y no necesita almacenarse ya que es la misma para todos los números.

- Se asume que hay un bit a la izquierda de la coma de la base.
- Formato típico de coma flotante de 32 bits:
  - El bit más a la izquierda contiene el signo (0 si es positivo, 1 si es negativo).
  - El valor del exponente se almacena en los 8 bits siguientes, de forma sesgada.
    - Un valor fijo (sesgo) se resta de este campo para conseguir el valor del exponente verdadero. Sesgo =  $2^{(k-1)} - 1$ , donde k es la cantidad de bits en el exponente. Para este ejemplo, el sesgo es  $2^{(8-1)} - 1 = 127$ .
  - La parte final de la palabra (23 bits) es la parte significativa o mantisa.
- En punto flotante la resolución no es constante a lo largo del intervalo. Esto hace que muchos cálculos produzcan resultados no exactos y que tengan que redondearse al valor más próximo representable.

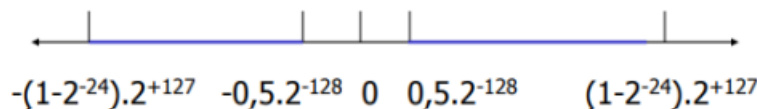
**Normalización:** En un número normalizado, el dígito más significativo de la parte significativa es distinto de 0 (debe ser 1). Las mantisas fraccionarias se definen de la forma: 0,1dddddd.....ddd.

- Como todos los números comienzan con 0,1 el 1 puede no almacenarse. Si no lo almaceno, puedo "adicionar" un bit más a la mantisa. El bit no almacenado se conoce como bit implícito.

### • Sin bit implícito



### • Con bit implícito



- ¿Cómo se escribe un número en punto flotante normalizado?
  - Se escribe el número en el sistema propuesto para la mantisa.
  - Se desplaza la coma y se cambia el exponente hasta obtener la forma normalizada.
  - Se convierte al exponente al sistema propuesto para él.

Ej. - 13,5 . Formato anterior

- 1) 1 1101,100..0 =  $1 \ 1101,100..0 \times 2^0$
- 2) 1 0,110110..0  $\times 2^4$
- 3) 4 en Ca2 = 00000100  
4 en Exceso = 10000100

### ▪ Sin bit implícito

1	10000100	1101100000.....00
---	----------	-------------------

### ▪ Con bit implícito

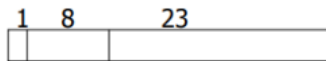
1	10000100	101100000.....00
---	----------	------------------

- Resolución: es la diferencia entre dos representaciones sucesivas, y varía a lo largo del rango, no es constante como en el caso de punto fijo.
- Error Absoluto: es la diferencia entre el valor representado y el valor a representar → Error Relativo = EA/Número a representar
- Error Absoluto máximo  $\leq$  Resolución/2

**Estándar IEEE 754:** El estándar IEEE define tanto el formato simple de 32 bits como el doble de 64 bits, con exponentes de 8 y 11 bits respectivamente. La base implícita es 2.

- Mantisa: fraccionaria normalizada, con la coma después del primer bit que es siempre uno (1,) en M y S.
- Exponente: representado en exceso  $2^{(n-1)} - 1$

➤ Simple precisión



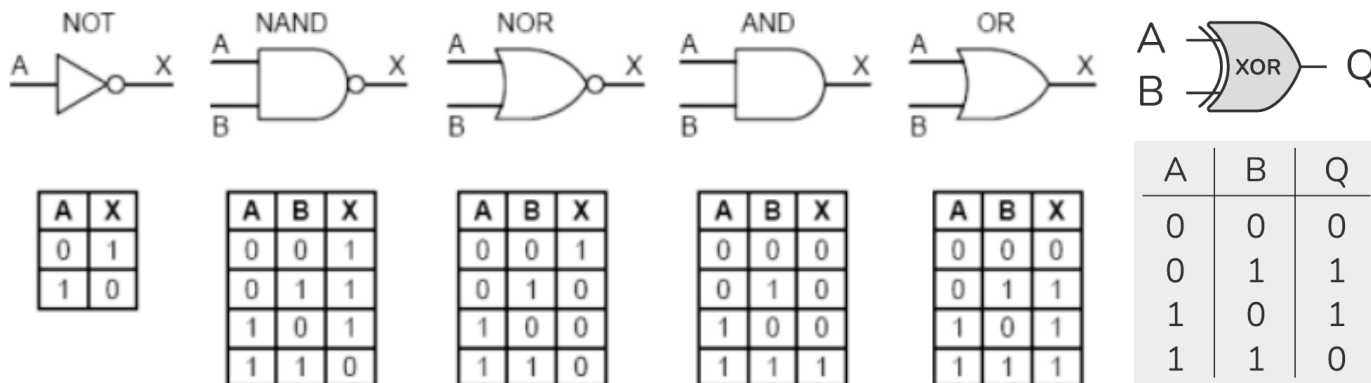
➤ Doble precisión



	Simple	Doble precisión
Bits en signo	1	1
Bits en exponente	8	11
Bits en fracción	23	52
Total de bits	32	64
Exponente en exceso	127	1023
Rango de exponente	$-126 \text{ a } +127$	$-1022 \text{ a } +1023$
Rango de números	$2^{-126} \text{ a } \sim 2^{128}$	$2^{-1022} \text{ a } \sim 2^{1024}$

## Circuitos Lógicos

Un circuito digital es en el que están presentes dos valores lógicos. Las compuertas son dispositivos electronicos que pueden realizar distintas funciones con estos dos valores lógicos. Compuertas básicas: AND, OR, NOT, NAND, NOR, XOR.



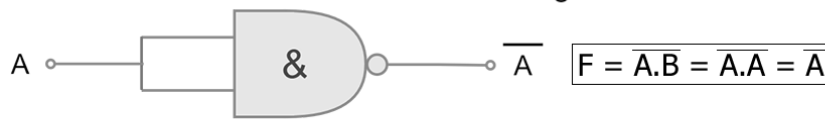
**Álgebra booleana:** para describir los circuitos que pueden construirse combinando compuertas, se requiere un nuevo tipo de álgebra, donde las variables y funciones sólo puedan adoptar los valores lógicos 0 (falso) o 1 (verdadero).

- Las operaciones lógicas básicas son AND, OR y NOT:
  - AND: es verdadera (1) si y sólo si los dos operandos son verdaderos  $\rightarrow A \text{ AND } B = A \cdot B$
  - OR: es verdadera si y sólo si uno o ambos operandos son verdad  $\rightarrow A \text{ OR } B = A + B$
  - NOT: invierte el valor del operando  $\rightarrow \text{NOT } A = \bar{A}$
  - En ausencia de paréntesis, la operacion AND es preferente a la operacion OR. Cuando no hay ambigüedad, la operacion AND se representa con una simple concatenacion en lugar del operador punto.
- Otros tres operadores utiles son: XOR, NAND y NOR:
  - XOR (exclusive-or): es verdadero si y sólo si, uno de los operadores es verdadero (si los dos son verdaderos da falso).
  - NAND: complemento NOT de AND.
  - NOR: complemento NOT de OR.
- En un AND, basta que una de sus entradas sea 0 para que la función valga 0.
- En un OR, basta que una de sus entradas sea 1 para que la función valga 1.
- Hacer el XOR con 1 invierte el valor de la variable. Con un 0 deja el valor de la variable como estaba.
- Identidades del algebra booleana:

Identidad	$1.A=A$	$0+A=A$
Nula	$0.A=0$	$1+A=1$
Idempotencia	$A.A=A$	$A+A=A$
Inversa	$A.\bar{A}=0$	$A+\bar{A}=1$
Conmutativa	$A.B=B.A$	$A+B=B+A$
Asociativa	$(AB).C=A(BC)$	$(A+B)+C=A+(B+C)$
Distributiva	$A+B.C=(A+B).(A+C)$	$A.(B+C)=AB+AC$
Absorción	$A.(A+B)=A$	$A+A.B=A$
De Morgan	$\overline{A.B}=\bar{A}+\bar{B}$	$\overline{A+B}=\bar{A}.\bar{B}$

- Leyes de Morgan: enunciados que describen las interacciones entre varias operaciones de la teoría de conjuntos. Permite obtener la función de una compuerta lógica con la combinación de otras.
  - Construir las puertas básicas usando sólo puertas NAND o NOR:
    - NAND:

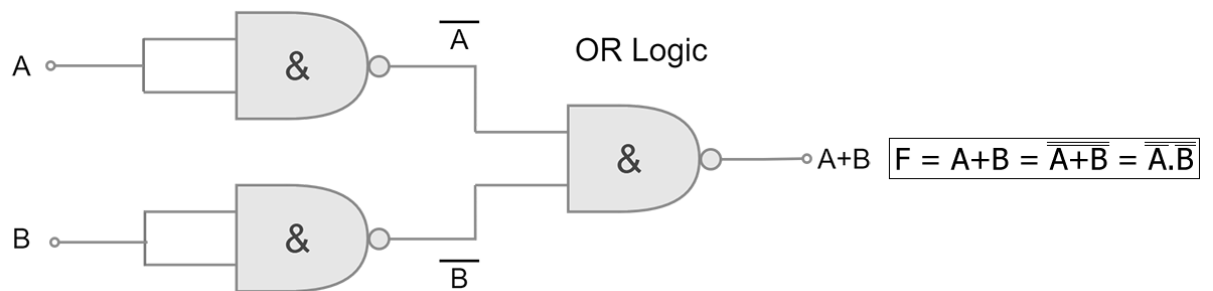
#### NOT Logic



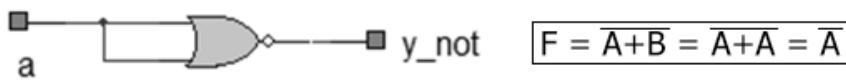
#### AND Logic



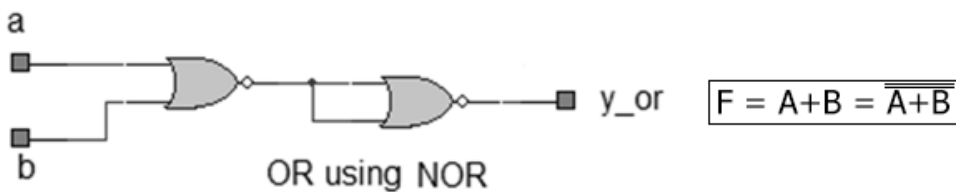
#### OR Logic



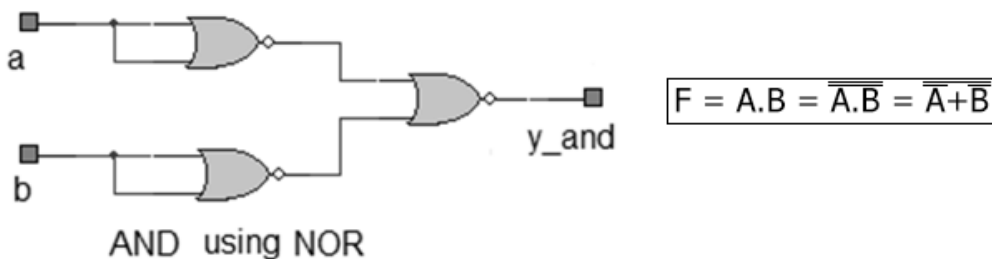
- NOR:



#### NOT using NOR



#### OR using NOR



#### AND using NOR

- Consecuencias de la ley de Moore:
  - El precio de un chip ha permanecido prácticamente invariable a través de este periodo de rápido crecimiento en densidad. El coste de la lógica del computador y de la circuitería de la memoria ha caído a una velocidad drástica.
  - La longitud de las interconexiones eléctricas ha disminuido (en chips más densamente encapsulados, los elementos de la lógica y memoria están más

---

proximos), incrementándose así la velocidad operativa.

- El computador es ahora más pequeño, lo que lo hace más adecuado para más entornos.
- Hay una reducción de las necesidades de potencia y refrigeración.
- Las interconexiones de los circuitos integrados son mucho más fiables que las conexiones soldadas. Con más circuitos en cada chip hay menos conexiones entre chips.

Circuitos a dos niveles: La forma de garantizar que un circuito no tenga más de dos niveles de puertas es partir de la expresión de la función en suma de minterminos. El circuito correspondiente a la suma de minterminos tiene, además de las puertas NOT, un primer nivel de puertas AND y un segundo nivel en el que habrá una única puerta OR, con tantas entradas como minterminos tenga la expresión.

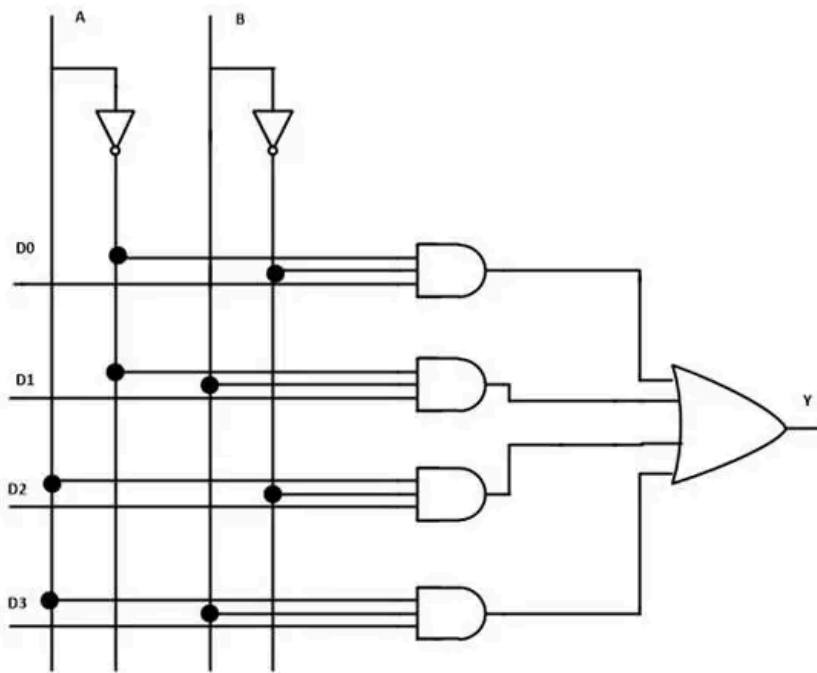
- Suma de Productos: se suman (OR) las combinaciones que hacen salida = 1, para cada combinación se hace el producto (AND) de todas las variables (se niegan las variables que estén en 0).
- Las expresiones de las funciones en suma de minterminos nos permiten construir los circuitos en dos niveles más rápidos posibles que podemos obtener de manera sistemática.
- Simplificación algebraica: las simplificaciones útiles corresponden a los casos en que dos o más minterminos se pueden reducir a un único término producto, en el que aparecerán menos variables. Se puede obtener un solo término producto en los siguientes casos:
  - Si en 2 minterminos todas las variables se mantienen constantes menos 1.
  - Si en 4 minterminos todas las variables se mantienen constantes menos 2.
  - Si en  $2^m$  minterminos todas las variables se mantienen constantes menos  $m$ .
- Minimizar la función: obtener la expresión en suma de productos más simplificada de una función, sacando factor común en los casos especiales.

## Circuitos Combinacionales

Conjunto de puertas interconectadas cuya salida, en un momento dado, es función solamente de la entrada en ese instante. Responden a los valores lógicos en las entradas, la salida está determinada exclusivamente por los valores de las entradas en ese instante. La aparición de la entrada viene seguida casi inmediatamente por la aparición de la salida, con solo retardos de puerta. Si cambia la entrada, cambia la salida. Los valores pasados de las entradas no influyen en los valores de las salidas.

Multiplexor: tiene un determinado número de señales de entrada que compiten para conectarse a una señal única de salida, y unas señales de control que sirven para determinar qué señal de entrada se conecta en cada momento con la salida. En las entradas y salida de un multiplexor conectaremos las señales lógicas que convengan para el funcionamiento de los circuitos.

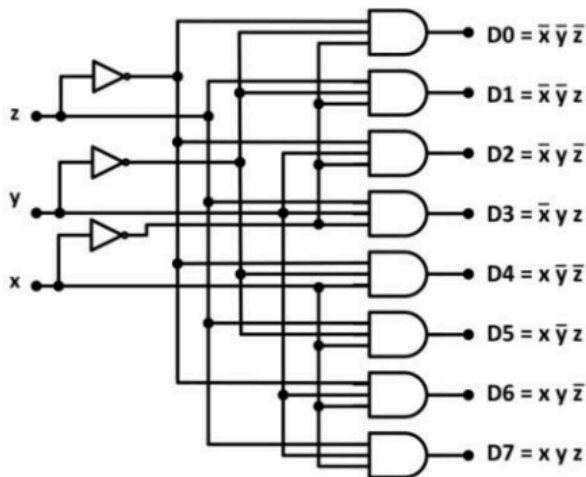
- Las entradas y salida de un multiplexor son:  $2^m$  entradas de datos, identificadas por la letra  $e$  y numeradas desde 0 hasta  $2^m - 1$ . Una salida de datos  $s$ .  $m$  entradas de control (selección), identificadas por la letra  $c$  y numeradas desde 0 hasta  $m-1$ . Una entrada de validación (VAL).
- Multiplexor de 4 a 1: Hay 4 líneas de entrada ( $D_0, D_1, D_2, D_3$ ). Se selecciona una de estas líneas para dar la señal de salida  $F$ . Para seleccionar una de las 4 entradas posibles, se necesita un código de selección de 2 bits, que se implementa con 2 líneas de selección ( $S_1$  y  $S_2$ ).
  - $S_1$  y  $S_2$  se conectan a las puertas AND de forma que, para cualquier combinación de  $S_1$  y  $S_2$ , 3 de las puertas AND tengan la salida 0. La 4ta puerta AND sacará el valor de la línea seleccionada que será 0 o 1.
  - 3 de las entradas de la puerta OR son siempre 0, y la salida de la puerta OR será igual al valor de la puerta de entrada seleccionada.



VAL	S1	S2	F
0	x	x	0
1	0	0	D0
1	0	1	D1
1	1	0	D2
1	1	1	D3

**Decodificador:** dada una combinacion binaria presente en la entrada, indican a qué número decimal corresponde.

- Tienen las señales: Una entrada de validacion (VAL). m entradas de datos, identificadas por la letra e y numeradas de 0 a m-1 que se interpretan como si formaran un número codificado en binario.  $2^m$  salidas, identificadas por la letra s y numeradas de 0 a  $2^m - 1$ , de las cuales solo una vale 1 en cada momento.
- Decodificador de 3 a 8: 3 entradas y 8 salidas.

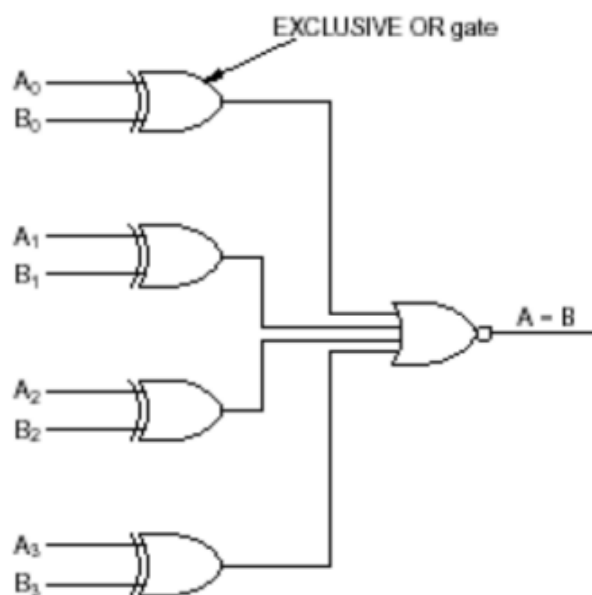


X Y Z	D0	D1	D2	D3	D4	D5	D6	D7
0 0 0	1	0	0	0	0	0	0	0
0 0 1	0	1	0	0	0	0	0	0
0 1 0	0	0	1	0	0	0	0	0
0 1 1	0	0	0	1	0	0	0	0
1 0 0	0	0	0	0	1	0	0	0
1 0 1	0	0	0	0	0	1	0	0
1 1 0	0	0	0	0	0	0	1	0
1 1 1	0	0	0	0	0	0	0	1

**Comparador:** circuitos combinacionales capaces de comparar dos combinaciones presentes en sus entradas indicando si son iguales o diferentes; en caso de ser diferentes, indican cuál de las dos es mayor. Tienen tres salidas que indican el resultado de la comparación:  $A=B$ ,  $A<B$  y  $A>B$ .

- El procedimiento para comparar dos datos binarios consiste primero en comparar el bit más significativo de cada uno de ellos, si éstos son iguales, se compara el siguiente bit más significativo y así sucesivamente hasta encontrar una desigualdad que indica cuál de los datos es mayor o menor. Si se comparan todos los bits de ambos datos y no hay desigualdad entre ellos, entonces evidentemente son iguales.
- Comparador de 4 bits:

Si todos los bits  $A_i$  son iguales a los  $B_i$  la salida es '1'

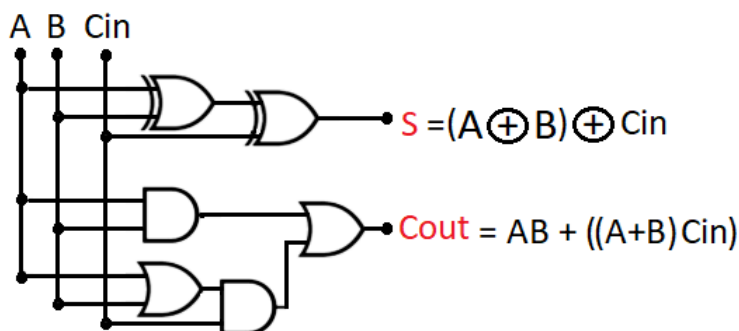


**Sumador Completo:** bloque combinacional que realiza la suma de dos números codificados en binario o bien en complemento a 2.

- Tiene la señales:
  - 2 entradas de datos de n bits (A y B) por donde llegan los números que se tienen que sumar.
  - 1 salida de n bits (S) que tomará el valor de la suma de los números A y B.
  - 1 salida de 1 bits (Cout) que valdrá 1 si cuando se haga la suma se produce acarreo en el bit de más peso.
  - 1 entrada de 1 bit (Cin) por donde llega un acarreo de entrada.

Sumador completo

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



## Circuitos Secuenciales

Las salidas dependen tanto de las entradas como del estado interno del circuito (historia pasada de las entradas). La denominación "secuencial" deriva de la capacidad de recordar la secuencia de valores que toman las señales. Tienen la característica de "almacenar" valores lógicos internamente. Estos valores se almacenan aunque las entradas no estén.

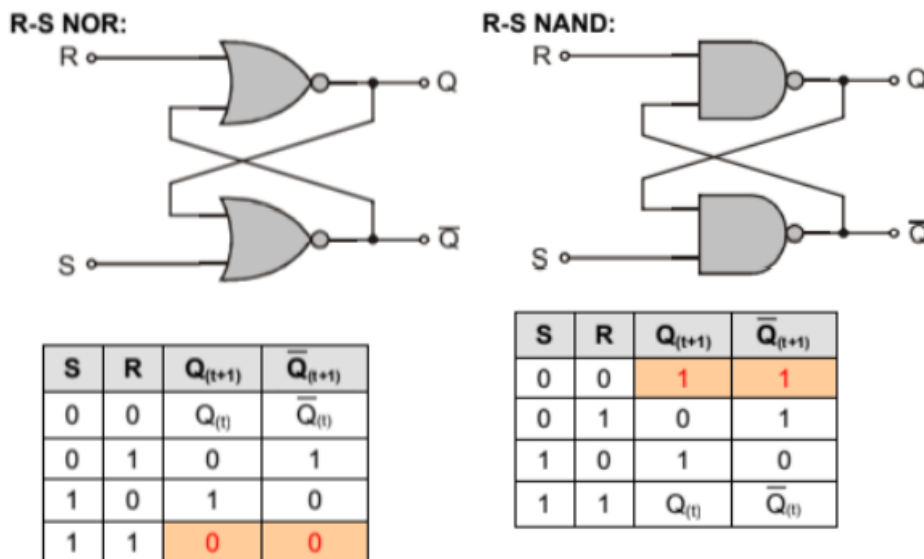
- Es preciso que este circuito sea capaz de recordar o retener los valores anteriores de algunas señales, es decir, debe tener memoria. Ésta es la funcionalidad que distingue los circuitos lógicos secuenciales de los combinacionales. En los circuitos combinacionales, la única noción temporal que interviene es el presente. En cambio, en los circuitos secuenciales se tiene en cuenta la evolución temporal de las señales.
- Para poder determinar cuándo cambia el valor viejo al nuevo, el circuito debe tener un mecanismo de sincronización. En los circuitos secuenciales se utiliza una señal de reloj como forma de sincronización.

**Biestable:** Los biestables son los dispositivos de memoria más elementales, permiten guardar un bit de información. Un biestable tiene dos salidas, Q y Q'. Se dice que Q es "el valor que guarda el biestable" en cada momento, o "el valor almacenado en el biestable", y Q' es su negación. Proviene del hecho de que el biestable puede estar "en dos estados":  $Q = 0$  o  $Q = 1$ .

- Clasificación según la forma de las señales de control:
  - Latch: los que producen el cambio lógico de sus salidas (disparo) mediante niveles de tensión en las señales de control.
  - Flip Flop: los que producen el cambio de sus salidas (disparo) mediante el cambio de los niveles de tensión de las entradas de control. El disparo puede producirse con el flanco de subida o con el flanco de bajada de las señales de control.
- Clasificación según el sincronismo o no de una señal patrón (reloj):
  - Asincrónicos: cuando en la entrada se establece una combinación, las salidas cambiarán.
  - Sincrónicos: la presencia de una entrada especial, determina "cuando" cambian las salidas acorde a las entradas.
- Clasificación por la lógica de control: Esta clasificación surge de la forma lógica de cambio de las salidas del biestable (Q y Q') por el cambio en los valores lógicos de las entradas de control.
  - Se clasifican en: SR, J-K, D y T.

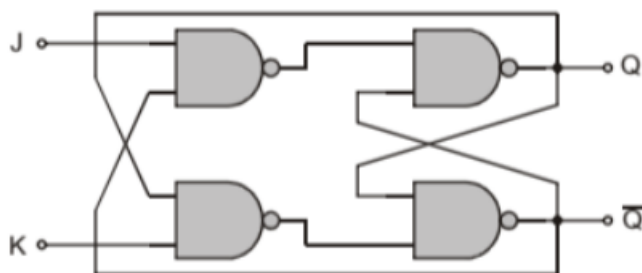
**Latch (básculas asincrónicas):** Son circuitos biestables, asincrónicos (o sea que no tienen entrada de pulsos de reloj) y son activados por niveles de tensión en sus entradas.

- Latch R-S (Set – Reset): estas básculas se pueden realizar con compuertas NOR o con compuertas NAND. Tanto en los latch NOR y NAND existen combinaciones de las señales de entrada que deben evitarse ( $R = S = 1$  para el latch NOR, y  $R = S = 0$  para el latch NAND), ya que con estas combinaciones las salidas Q y Q' toman valores iguales, lo que resulta en una indeterminación.



- Latch JK: Esta báscula elimina la condición de indeterminación de las anteriores. Se realiza partiendo de una báscula R-S NAND a la que se le agregan en sus entradas dos compuertas NAND.



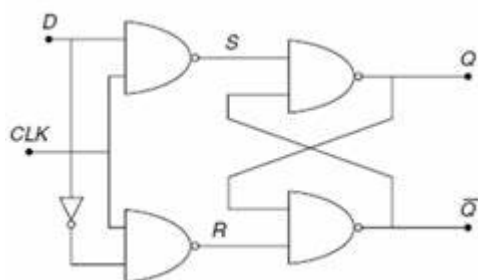


J	K	$Q_{(t+1)}$	$\bar{Q}_{(t+1)}$
0	0	$Q_{(t)}$	$\bar{Q}_{(t)}$
0	1	1	0
1	0	0	1
1	1	$\bar{Q}_{(t)}$	$Q_{(t)}$

- Diferencias entre JK y R-S: cuando ambas entradas del SR son 1, el circuito produce estados no válidos como salida. En JK no hay estados no válidos de salida.

**Flip-flop (basculas sincronicas):** estos también son circuitos biestables, pero en este las salidas se activan según el cambio de las entradas de control y un pulso de reloj. Según la báscula, la activación puede ser por el flanco ascendente o por el flanco descendente del pulso de reloj.

- Flip-Flop J-K: este flip flop posee una entrada "reloj" para sincronizar las entradas de control J y K. Las salidas del flip-flop responden a la lógica de estas entradas cuando ingresa el flanco de la señal de reloj; algunos flip-flop se diseñan para responder al flanco ascendente del reloj, y otros al flanco descendente.
  - Este flip flop también posee dos entradas asincrónicas preset (puesta a 1) y clear (puesta a 0), con las que se puede modificar la salida independientemente del valor que tomen las entradas J-K y del sincronismo del reloj. Para que el flip flop trabaje con sus entradas J-K y reloj, las entradas clear y preset deben dejarse desactivadas.
- Flip-Flop T: este flip flop se obtiene puentando las entradas de un *flip flop J-K*. Se denomina flip flop T por Toggle (conmutar), ya que de la tabla de verdad resultante se puede observar que cuando la entrada T es 0, la salida Q permanece sin cambio; pero cuando T es 1 la salida Q es conmutada en cada ciclo de reloj.
- Flip-Flop D: Este tipo de flip flop también se obtiene a partir de un *flip-flop J-K*, pero en este caso se coloca una compuerta NOT entre las entradas J y K. Se denomina flip flop D por Data (sigue el dato en la entrada). Es decir, en cada transición de la señal de reloj, el dato en la entrada D es copiado a la salida Q. El FF D permite aplicar una sola entrada para cambiar la salida.



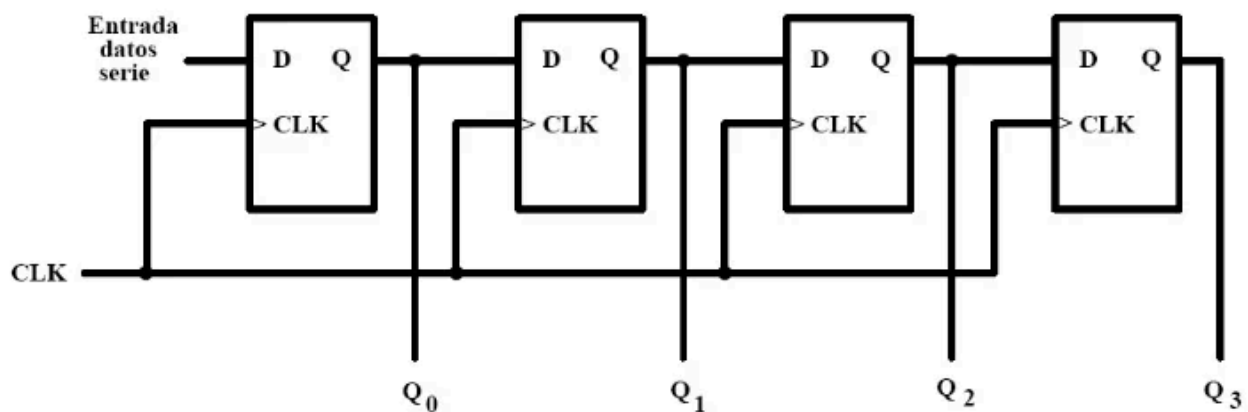
$Q_n$	D	$Q_{n+1}$
0	0	0
0	1	1
1	0	0
1	1	1

- Flip-Flop SR: S y R son las entradas que tendrán efecto cuando CK tome el valor 1. En el FF SR hay que aplicar 2 entradas diferentes para cambiar de estado.

Registro paralelo: bloque secuencial formado por  $n$  *biestables*  $D$ , que permite guardar el valor de una palabra de  $n$  bits. Cuando el clear está en 0, los  $n$  biestables que forman el registro se comportan como  $n$  biestables  $D$  con señal de carga.

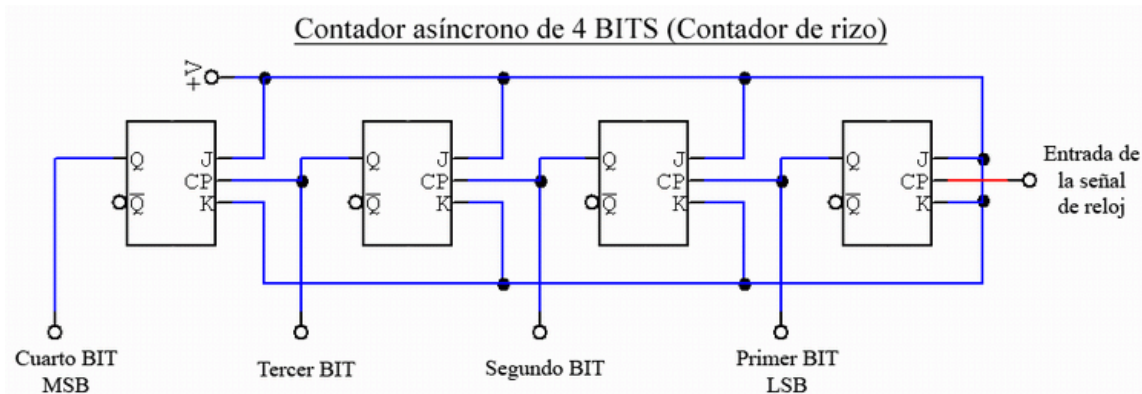
- Tiene las señales: 1 entrada de datos de  $n$  bits ( $E$ ). Cada uno de los bits de este bus está conectado con la entrada  $D$  de uno de los  $n$  biestables que forman el registro. 1 salida de datos de  $n$  bits ( $S$ ), que es un bus formado por las salidas  $Q$  de los  $n$  biestables que forman el registro. 2 entradas de control de 1 bit (load y clear). Están conectadas respectivamente a la señal load y a la entrada asincrónica  $R$  de cada uno de los biestables. 1 entrada de reloj, conectada a las entradas de reloj de todos los biestables.
- Registro con desplazamiento: un registro de desplazamiento acepta y/o transfiere información vía serie. Los datos se introducen únicamente a través del biestable que está más a la izquierda. Con cada pulso de reloj, los datos se desplazan a la derecha una posición, y el bit más a la derecha se transfiere fuera.

## Entrada serie-salida paralelo

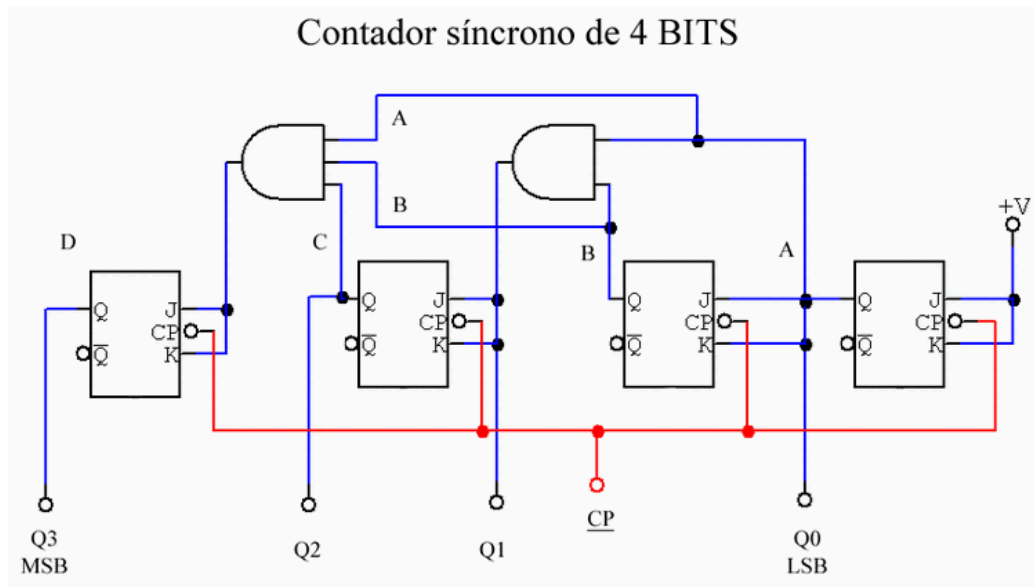


Contador: Son circuitos realizados con *flip flops*  $J$ - $K$  conectados en cascada, y cuya finalidad es contar impulsos o eventos, medir tiempos, medir frecuencias, etc. Se los clasifica en dos formas: asincrónicos y sincrónicos.

- Contador asincrónico (ripple): son relativamente lentos, ya que la salida de un biestable produce un cambio en el estado del siguiente biestable. Se llama también contador de onda, ya que el cambio que se produce para incrementar el contador empieza en un extremo y se transfiere como una onda hasta el otro extremo.
  - El contador se incrementa con cada pulso de reloj. Las entradas  $J$  y  $K$  de cada biestable se mantienen a 1 constante (cuando hay un pulso de reloj, la salida  $Q$  se invierte).



- Contador síncrono: todos los biestables cambian de estado a la vez. Como es más rápido que el asincrónico, es el tipo que se usa en CPU. Para mantener una salida a 0, debemos tener las entradas  $J=0$  e indiferencia para  $K$ . Para que haya una transición de 0 a 1, las entradas deben ser  $J=1$  y  $K=d$ .



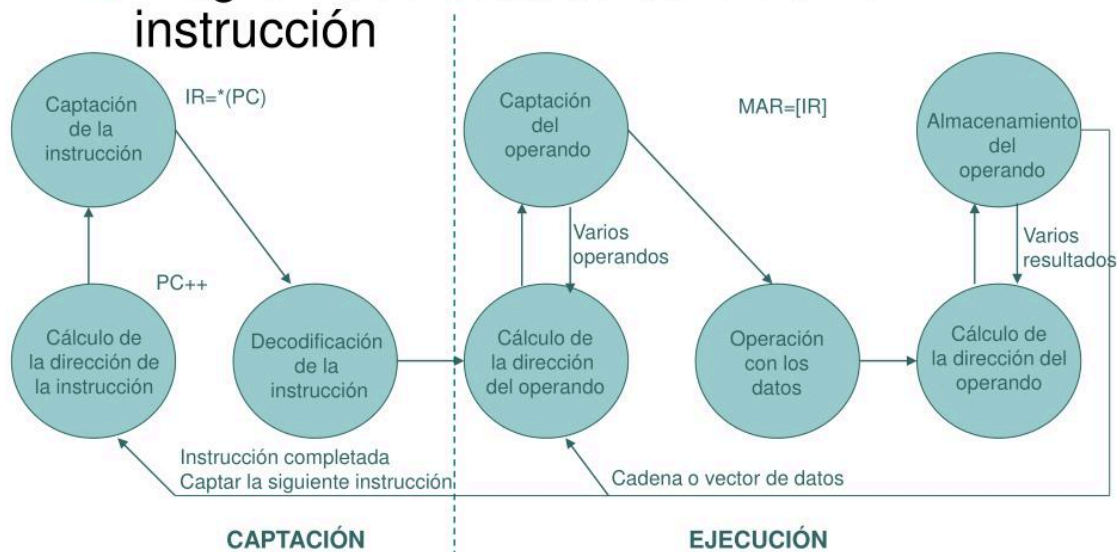
## Ciclo de Instrucción

El procesamiento que requiere una instrucción se denomina ciclo de instrucción. Este tiene dos etapas conocidas como ciclo de captación y ciclo de ejecución. La ejecución del programa se para solo si la máquina se desconecta, se produce algún tipo de error irreparable o ejecuta una instrucción del programa que detiene al computador.

Diagrama de estados del ciclo de instrucción: para cualquier ciclo de instrucción dado, algunos estados pueden no estar y otros pueden repetirse. Los estados en la parte superior implican un intercambio entre la CPU y la memoria ó e/s. Los estados en la parte inferior implican sólo operaciones internas en la CPU.

- Cálculo de la dirección de la instrucción: determina la dirección de la siguiente instrucción a ejecutarse.
- Captación de instrucción: lee la instrucción de su posición de memoria a la CPU.
- Decodificación de la operación indicada en la instrucción: analiza la instrucción para determinar el tipo de operación a realizar y los operandos que se usarán.
- Cálculo de la dirección del operando: si la operación implica la referencia a un operando en la memoria ó e/s, entonces se determina la dirección.
- Captación de operando: busca el operando en la memoria ó e/s.
- Operación con los datos: ejecuta la instrucción.
- Cálculo dirección resultado
- Almacenamiento de operandos: almacenamiento resultado.

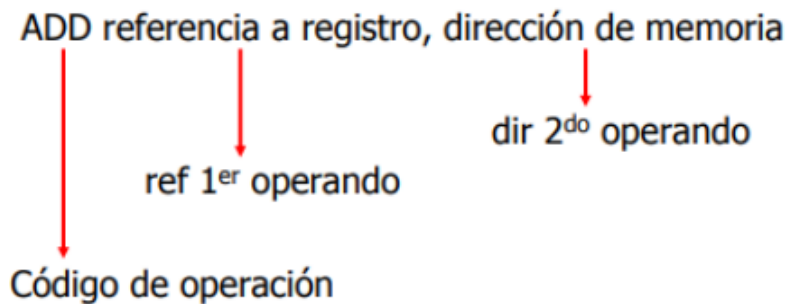
### Diagrama de estados del ciclo de instrucción



#### Ejemplo - Instrucción ADD:

- Buscar la instrucción en memoria: la CPU copia el valor del PC al MAR y de ahí al bus de direcciones. La UC envía las señales necesarias para una operación de lectura. Se pueden leer uno ó más bytes. A través del bus de datos al MBR y luego al IR.
- Incrementar el PC: después de buscar la instrucción, la CPU debe incrementar el PC para apuntar a "lo que sigue". Puede ser un dato, dirección ó la siguiente instrucción.
- Decodificar la instrucción: el paso siguiente es decodificar la instrucción para saber qué operación hacer (suma, resta, etc.). En este momento la CPU no sólo se entera de la operación, sino también dónde se encuentran los datos sobre los cuales operar. La CPU determina si tiene que ir a buscar un operando a memoria, que ocupa una celda o más (byte/s) y lo hace en este momento.
  - Si es necesario, buscar una constante en una dirección de memoria.
  - Si es necesario, incrementar el PC para que apunte más allá de la constante.
  - Si es necesario calcular la dirección del operando.
- Buscar uno de los operandos, desde memoria ó registro.

- Buscar el otro operando desde registro.
- Realizar la Suma.
- Almacenar el resultado.



## Formatos de Instrucción

El funcionamiento del procesador está determinado por las instrucciones que ejecuta. Estas instrucciones se denominan instrucciones máquina o instrucciones del computador. Al conjunto de instrucciones distintas que puede ejecutar el procesador se denomina repertorio de instrucciones del procesador. Cada instrucción debe contener la información que necesita el procesador para su ejecución.

- Los elementos de una instrucción son:
  - Código de operación: especifica la operación a realizar. La operación se indica mediante un código binario denominado código de operación (codop).
  - Referencia a operandos fuente u origen: la operación puede implicar a uno o más operandos origen, es decir, operandos que son entradas para la instrucción.
  - Referencia al operando de destino o resultado: la operación puede producir un resultado.
  - Referencia a la siguiente instrucción: dice al procesador de donde captar la siguiente instrucción tras completarse la ejecución de la instrucción actual. En la mayoría de los casos se ubica a continuación de la instrucción actual.
- Los operandos fuente y resultado pueden estar en tres lugares:
  - Memoria: como en las referencias a instrucciones siguientes, debe indicarse la dirección de memoria principal o de memoria virtual.
  - Registro de la CPU: salvo raras excepciones, un procesador contiene uno o más registros que pueden ser referenciados por instrucciones máquina.
    - Si solo existe un registro, la referencia a él puede ser implícita.
    - Si existe más de uno, cada registro tendrá asignado un número único y la instrucción debe contener el número del registro deseado.
  - Dispositivo de E/S: las instrucciones deben especificar el módulo y dispositivo de E/S para la operación. En el caso de E/S asignadas en memoria, se dará otra dirección de memoria principal o virtual.

**Número de direcciones:** En teoría, se necesitan 4 direcciones para ejecutar una instrucción: 2 direcciones para hacer referencia a los operandos, 1 donde almacenar el resultado y la dirección de la próxima instrucción. En la práctica, la dirección de la próxima instrucción queda implícita, por lo que existen instrucciones de 3, 2, 1 y 0 direcciones. En todos los casos se supone que la dirección de la siguiente instrucción está implícita, y que se va a realizar una operación con 2 operandos origen y 1 resultado.

- Máquina para 3 direcciones: La dirección de la próxima instrucción está almacenada en un registro de la CPU, llamado Contador de Programa PC. Referencias = 72 bits.
  - Operan con datos en registros exclusivamente.
- Máquina para 2 direcciones: Hay que mover el Op1 a un registro temporal. Menos elección donde guardar el resultado. Referencias = 48 bits.
  - Operan con datos en registros y con datos en memoria.
  - En cada operación, se permite un acceso máximo a memoria.

- Máquina para 1 dirección: Registros especiales en la CPU (acumulador). Instrucciones para cargar y descargar el acumulador. Un operando y resultado en lugar predefinido. Referencia = 24 bits.
- Las instrucciones con 0 direcciones son aplicables a una organización especial de memoria, llamada pila (stack).

Número de direcciones	Representación simbólica de instrucción	Interpretación
3	OP A, B, C	B OP C → se guarda en A
2	OP A, B	A OP B → se guarda en A
1	OP A	Acumulador OP A → se guarda en el Acumulador
0	OP	(T-1) OP T → se guarda en T (tope de la pila)

○ Ejemplo - evaluar  $a=(b+c)*d - e$ :

<u>3 direcciones</u>	<u>2 direcciones</u>	<u>1 dirección</u>
add a, b, c	mov a, b	load b
mul a, a, d	add a, c	add c
sub a, a, e	mul a, d	mul d
	sub a, e	sub e
		store a
3 instruc./3 acc. MI 9 acc. MD	4 instruc./4 acc. MI 11 acc. MD	5 instruc./5 acc. MI 5 acc. MD

Diseño del repertorio de instrucciones: el conjunto de instrucciones es el medio que tiene el programador para controlar la CPU. Los aspectos más importantes de diseño son:

- Repertorio de operaciones (tipos de operaciones): cuántas y qué operaciones considerar, y cuán complejas deben ser.
  - Transferencia de datos: Mov (load/store)
  - Aritméticas: Add, Sub, Inc, Dec, Mul
  - Lógicas: And, Or, Xor, Not
  - Conversión E/S: In, Out
  - Transferencia de control: salto, bifurcación
  - Control del sistema: usadas por S.O.
- Tipos de datos: los distintos tipos de datos con los que se efectúan operaciones.
  - Direcciones: pueden considerarse como números enteros sin signo.
  - Números: enteros, punto fijo, punto flotante.
  - Caracteres: ASCII, BCD.
  - Datos lógicos: Verdadero (1) o Falso (0).
- Formatos de instrucciones: longitud de la instrucción en bits, número de direcciones, tamaño de los distintos campos, etc.
- Registros: número de registros del procesador que pueden ser referenciados por las instrucciones y su uso.
- Direccionamiento: el modo o modos de direccionamiento mediante los cuales puede especificarse la dirección de un operando.
  - El modo de direccionamiento de una instrucción influye en el número de ciclos que tardará en ejecutarse.
  - Los modos de direccionamiento permitidos influyen en el ciclo de reloj de la máquina.

- El conjunto de los modos de direccionamiento que se permiten en una determinada máquina influirá en la complejidad del hardware.

#### Criterios de diseño:

- Longitud de Instrucción: el aspecto de diseño más básico a considerar en el formato es la longitud o tamaño de la instrucción. Esta decisión afecta, y se ve afectada por, el tamaño de la memoria, su organización, la estructura de buses, la complejidad del procesador y la velocidad del procesador.
  - El deseo de disponer de un repertorio de instrucciones máquina potente y la necesidad de ahorrar espacio. El programador desea más codops y operandos (permite redactar programas más cortos para resolver las mismas tareas), más modos de direccionamiento (dan más flexibilidad para implementar ciertas funciones) y mayor rango de direcciones. Todo esto requiere de bits y empuja hacia longitudes de instrucción mayores.
  - Debiera cumplirse o bien que el tamaño de la instrucción fuera igual al tamaño de las transferencias a memoria, o bien que uno fuera un múltiplo del otro. En caso contrario, no conseguiremos un número íntegro de instrucciones durante un ciclo de captación.
  - Velocidad de transferencia de memoria. Esta velocidad no ha seguido el mismo aumento que la velocidad de los procesadores. Por esto, la memoria puede convertirse en un cuello de botella si el procesador puede ejecutar las instrucciones más rápido que lo que tarda en captarlas. Las soluciones pueden ser utilizar la memoria caché o utilizar instrucciones más cortas.
  - La longitud de la instrucción debiera ser un múltiplo de la longitud de un carácter, que normalmente es 8 bits, y de la longitud de los números en coma fija.
- Asignación de los bits: para una longitud de instrucción dada, existe claramente un compromiso entre el número de codops y la capacidad de direccionamiento. Un mayor número de codops reduce, para un formato de instrucción de una longitud dada, el número de bits disponibles para direccionamiento. Los siguientes factores, relacionados entre sí, afectan a la definición del uso dado a los bits de direccionamiento:
  - Número de modos de direccionamiento: El modo de direccionamiento puede a veces indicarse de manera implícita. Para los casos que deben ser explícitos, requieren 1 o más bits de modo.
  - Número de operandos: cada dirección de operandos (2 operandos) podría requerir su propio indicador de modo dentro de la instrucción, o el uso del indicador de modo podría estar limitado a sólo uno de los campos de direcciones.
  - Registros frente a memoria: En el caso de 1 solo registro visible para el usuario (acumulador) la dirección del operando está implícita y no consume bits de la instrucción. Incluso con varios registros, solo se necesitan unos pocos bits para especificar el registro.
  - Número de conjuntos de registros: hay una tendencia de pasar de un solo banco de registros de uso general a un grupo de 2 o más conjuntos especializados. Ventaja: para un número dado de registros, una partición funcional de estos requiere menos bits de la instrucción.
  - Rango de direcciones: para referencias a memoria, el rango de direcciones que puede utilizarse está relacionado con el número de bits de direccionamiento. Raramente se emplea direccionamiento directo. Es conveniente permitir desplazamientos bastante más largos que los del registro de direcciones, y esto requiere de un número relativamente grande de bits de direcciones en la instrucción.
  - Granularidad de las direcciones: el direccionamiento por bytes es conveniente para manipular caracteres pero requiere, para el tamaño de memoria dado, de más bits de direcciones.

## **Modos de Direccionamiento**

Para reducir el tamaño de las especificaciones necesarias de una instrucción (código de operación y direcciones) se aplican dos métodos generales: Si un operando va a usarse varias veces puede colocarse en un registro: usar registro para una variable tiene 2 ventajas, el acceso es más rápido y se necesitan menos bits. Especificar uno ó más operandos en forma implícita.

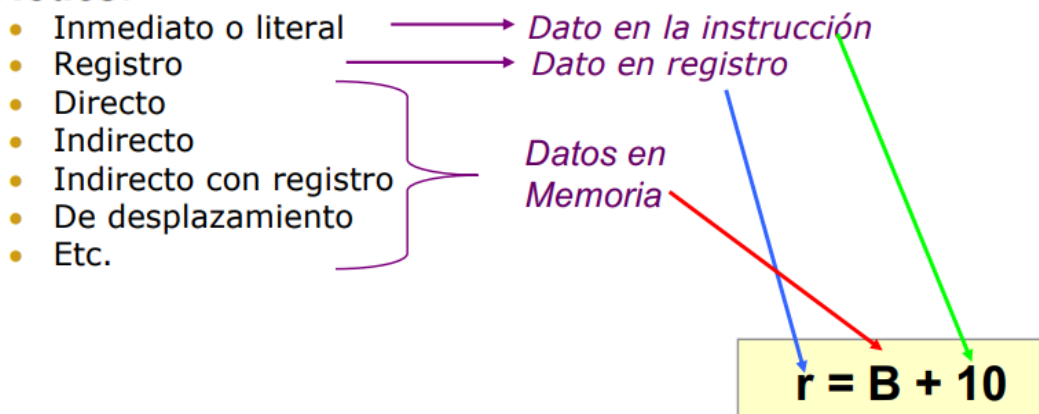


Un modo de direccionamiento especifica la forma de calcular la dirección de memoria efectiva de un operando mediante el uso de la información contenida en registros. Modo en el que se indica la dirección efectiva de un operando, es decir, su ubicación. Los modos de direccionamiento tienen como objetivos:

- Disminuir la cantidad de bits en la instrucción.
- La dirección puede que no se conozca hasta el momento de ejecutar el programa.
- Manejo más eficiente de datos (arreglos).

Tipos de modos de direccionamiento:

### Modos:



- Inmediato: El dato aparece en la propia instrucción y, por tanto, es constante. El operando se obtiene automáticamente de la memoria al mismo tiempo que la instrucción. No requiere una referencia extra a la memoria de datos. Se utiliza para definir constantes y para inicializar variables.
  - MOV AX, 12
  - ADD AX, 5
- Directo: El campo de dirección tiene la dirección efectiva del operando (la instrucción contiene la dirección de la posición de memoria donde se encuentra el dato). Es simple, pero tiene un espacio limitado de direcciones por cantidad de bits del campo. Sólo requiere un acceso a memoria. Se usa para acceder a variables globales, cuya dirección se conoce en el momento de compilación.
  - MOV A, 17H
  - ADD AX, [100]
- Por registro: Conceptualmente igual al directo, pero se especifica un registro en lugar de una posición de memoria (el dato se encuentra almacenado en un registro). La referencia a registro usa menos bits que la especificación de la dirección y no requiere acceso a memoria de datos.
  - ADD AX, BX
- Indirecto por memoria: En la instrucción está la dirección de la dirección del operando (la instrucción contiene la dirección de la posición de memoria donde se encuentra la dirección del dato). Trata de solucionar el problema del directo. Requiere dos accesos a memoria.
- Indirecto por registro: En la instrucción se especifica el registro que tiene almacenada la dirección (la instrucción contiene el número de registro donde se encuentra la dirección del dato). Requiere un acceso a registro y otro a memoria.
  - ADD AX, [BX]
- Por desplazamiento: Combina capacidades de indirecto y directo. Requiere que la instrucción tenga dos campos de dirección. Estos dos campos se suman para producir la dirección efectiva. Los tres usos más comunes son:
  - Relativo (a PC): La instrucción contiene un campo con una constante llamada desplazamiento. La dirección de memoria del dato se calcula como la suma de dicha constante más el contenido del PC (contador de programa). Requiere un acceso a un registro específico (PC). El resultado de la suma corresponde a la dirección destino si un salto condicional se va a realizar.
    - Saltos condicionales: JZ, JNZ, JS, JNS, JC, JNC, JO, JNO.



- De registro base: La instrucción contiene dos campos: un número de registro y una constante llamada desplazamiento. La dirección de memoria del dato se calcula como la suma de dicha constante más el contenido del registro. Requiere un acceso a registro y otro a memoria.
  - ADD AX, [Bp+100]
- Indexado: La instrucción contiene dos campos registro. La dirección de memoria del dato se calcula como la suma del contenido de ambos registros. Requiere dos accesos a registros.
  - ADD R3, (R1+R2)
- Del stack (pila): El stack ó pila es un arreglo lineal de localidades de memoria. Es una lista ó cola donde el último en entrar es el primero en salir. Es una zona de memoria reservada. Asociado con la pila o stack hay un registro apuntador (o registro puntero de pila), cuyo valor es la dirección tope de pila o stack.

Instrucciones Intel 8086/MSX88: Tienen la forma → Instrucción Destino, Fuente. Destino y fuente son 2 operandos, donde c/u de ellos está especificado por alguno de los modos de direccionamiento vistos, el otro operando es un registro de la CPU. En una suma hay 2 operandos y el resultado se almacena en el lugar del operando izquierdo (destino).

INSTRUCCIONES DE TRANSFERENCIA DE DATOS		
INSTRUCCIÓN	COMENTARIO	OBS
MOV <i>dest,fuente</i>	Copia <i>fuentes</i> en <i>dest</i>	Puede ser una etiqueta (dir.directo) o [BX], siendo (BX) una dirección de memoria (dir.indirecto).
PUSH <i>fuentes</i>	Carga <i>fuentes</i> en el tope de la pila	<i>dest</i> y <i>fuentes</i> solo pueden ser registros de 16 bits.
POP <i>dest</i>	Desapila el tope de la pila y lo carga en <i>dest</i>	
IN <i>dest,fuentes</i>	Carga el valor en el puerto <i>fuentes</i> en <i>dest</i>	Puede ser un operando inmediato o una etiqueta.
OUT <i>dest,fuentes</i>	Carga en el puerto <i>dest</i> el valor en <i>fuentes</i>	

INSTRUCCIONES ARITMETICAS		
INSTRUCCIÓN	COMENTARIO	OBS
ADD <i>dest,fuentes</i>	Suma <i>fuentes</i> y <i>dest</i>	Puede ser una etiqueta (dir.directo) o [BX], siendo (BX) una dirección de memoria (dir.indirecto).
ADC <i>dest,fuentes</i>	Suma <i>fuentes</i> , <i>dest</i> y flag C	
SUB <i>dest,fuentes</i>	Resta <i>fuentes</i> a <i>dest</i>	
SBB <i>dest,fuentes</i>	Resta <i>fuentes</i> y flag C a <i>dest</i>	
CMP <i>dest,fuentes</i>	Compara <i>fuentes</i> con <i>dest</i>	
NEG <i>dest</i>	Negativo de <i>dest</i>	<i>dest</i> solo puede ser mem o reg. mem puede ser una etiqueta (dir.directo) o [BX], siendo (BX) una dirección de memoria (dir.indirecto).
INC <i>dest</i>	Incrementa <i>dest</i>	
DEC <i>dest</i>	Decrementa <i>dest</i>	

INSTRUCCIONES LOGICAS		
INSTRUCCIÓN	COMENTARIO	OBS
AND <i>dest,fuente</i>	Operación <i>fuente</i> AND <i>dest</i> bit a bit	Puede ser una etiqueta (dir.directo) o [BX], siendo (BX) una dirección de memoria (dir.indirecto).
OR <i>dest,fuente</i>	Operación <i>fuente</i> OR <i>dest</i> bit a bit	
XOR <i>dest,fuente</i>	Operación <i>fuente</i> XOR <i>dest</i> bit a bit	
NOT <i>dest</i>	Complemento a 1 de <i>dest</i>	<i>dest</i> solo puede ser mem o reg. mem puede ser una etiqueta (dir.directo) o [BX], siendo (BX) una dirección de memoria (dir.indirecto).

INSTRUCCIONES DE TRANSFERENCIA DE CONTROL		
INSTRUCCIÓN	COMENTARIO	OBS
RET	Retorna de la subrutina	-          (IP)← mem, mem es la dirección de memoria llamada etiqueta
CALL <i>etiqueta</i>	Llama a subrutina cuyo inicio es <i>etiqueta</i>	
JZ <i>etiqueta</i>	Salta si el último valor calculado es cero	
JNZ <i>etiqueta</i>	Salta si el último valor calculado no es cero	
JS <i>etiqueta</i>	Salta si el último valor calculado es negativo	
JNS <i>etiqueta</i>	Salta si el último valor calculado no es negativo	
JC <i>etiqueta</i>	Salta si el último valor calculado produjo carry	
JNC <i>etiqueta</i>	Salta si el último valor calculado no produjo carry	
JO <i>etiqueta</i>	Salta si el último valor calculado produjo overflow	
JNO <i>etiqueta</i>	Salta si el último valor calculado no produjo overflow	
JMP <i>etiqueta</i>	Salto incondicional a <i>etiqueta</i>	

- La arquitectura de MSX88 pretende ser una simplificación de la del Intel 8088, siendo sus rasgos arquitecturales más destacados los siguientes:
  - Arquitectura interna de 16 bits y externa de 8 bits.
  - Bus de direcciones de 16 bits.
  - Registros de uso general AX; BX; CX y DX, de 16 bits, pudiéndose tratar también como registros de 8 bits(AH, AL, BH, BL, ...).
  - ALU de 16 bits capaz de ejecutar las operaciones: ADC, SUB, AND, OR, XOR, NOT, INC y DEC.
  - Registro de indicadores con los flags de: Cero, Paridad, Paridad Auxiliar, Signo, Overflow, e indicador de interrupciones.
  - Registros contador de programa (IP) y puntero de pila (SP) de 16 bits.
  - Soporta los siguientes modos de direccionamiento: Dato Inmediato, Registro, Relativo a contador de programa , Relativo a pila, Directo e Indirecto basado en el registro BX.
- Las instrucciones del SX88 están codificadas con cero, uno o dos operandos (registro, memoria, dato inmediato). Las operaciones se hacen entre registros, registros y memoria, datos inmediatos y registros, o entre datos inmediatos y memoria, pero nunca entre memoria y memoria.

#### Ejemplos de instrucciones Intel 8086 por modo de direccionamiento:

- Direccionamiento por Registro:

ADD AX,BX ➡ AX=AX+BX  
ADD AL,AH ➡ AL=AL+AH  
MOV AL,CH ➡ AL=CH  
SUB AX,BX ➡ AX=AX - BX

- Direccionamiento Inmediato:

ADD AX,35AFh ➡ AX=AX+35AFh

ADD AL,15 ➡ AL=AL+15

MOV AL,3Eh ➡ AL=3Eh

SUB AX,1234h ➡ AX=AX - 1234h

- Direccionamiento Directo:

- ADD AX, [35AFh]

- ➡ AX = AX + contenido direcc. 35AFh y 35B0h

- ADD AL, DATO

- ➡ AL = AL + contenido variable DATO (8 bits)

- MOV CH, NUM1

- ➡ CH = contenido variable NUM1 (8 bits)

- Direccionamiento Indirecto por Registro:

- ADD AX, [BX]

- ➡ AX = AX + dato almacenado en dirección contenida en BX y la que sigue

- MOV [BX], AL

- ➡ dato en la dirección contenida en BX = AL

- Direccionamiento base+índice:

- MOV CX, [BX+SI]

- ➡ CX = dato almacenado en la direcc. BX+SI y la siguiente

- MOV [BX+DI], AL

- ➡ dato almacenado en la direcc. BX+DI = AL

- Direccionamiento Relativo por Registro:

- MOV AL, [BX+2]

- ➡ AL=dato almacenado en dir BX+2

- MOV [BX+2Ah], AX

- ➡ dato almacenado en dir BX+2Ah y la que sigue = AX (16 bits)

- Direccionamiento relativo base+índice:

---

- MOV AL, [BX+SI+2]

➡ AL = dato almacenado en la dir BX+SI+2

- MOV [BX+DI+2Ah], AX

➡ dato almacenado en la dir BX+DI+2Ah y la que sigue = AX (16 bits)

- Saltos:

· **INSTRUCCIÓN DE SALTO INCONDICIONAL.**

**JMP dirección**

11101001	LSB dir	MSB dir
----------	---------	---------

· **INSTRUCCIONES DE SALTO CONDICIONAL.**

**JZ desplazamiento**

01110100	desplazam.
----------	------------

# Memoria

Principio de localidad: asegura que los programas acceden únicamente a una porción relativamente pequeña de su espacio de direccionamiento durante un corto espacio de tiempo. Existen dos tipos diferentes de localidad:

- Temporal: si se hace referencia a un objeto, existe una cierta tendencia a volver a referenciar en un corto espacio de tiempo (bucles en un programa o llamadas a subrutinas).
- Espacial: si se hace referencia a un objeto, también tenderán a ser referenciados los demás objetos que están ubicados en direcciones próximas a éste (acceso a vectores de datos o a la memoria de instrucciones).

## Jerarquía de Memoria

Para aprovechar la localidad, se implementa la memoria como una memoria jerárquica de manera que existan diferentes niveles de memoria con distintos tamaños y velocidades. La memoria más rápida (SRAM) constituye el nivel superior y se coloca próxima al procesador. El nivel inferior está constituido por la memoria más lenta (discos), que tiene un precio mucho menor, colocado detrás del nivel superior.

- Los programas se dividen en bloques de tamaño fijo que se cargan en la memoria más rápida para aprovechar las ventajas del principio de localidad.
- Debido al principio de localidad temporal, la mayoría de las veces se encuentra el dato en la memoria más rápida, ya que es bastante probable que el dato haya sido accedido anteriormente.
- La interacción entre los diferentes tipos de memoria se aprovecha de forma tal que se logra un comportamiento, por parte de la computadora, equivalente al que tendría con una memoria única, grande y rápida, cuando en realidad tiene distintos tipos de memoria trabajando en forma coordinada.

Tiempo de acceso		Capacidad
1 nsec	Registros	< 1kb
2 nsec	Cache	1-8Mb
10 nsec	Memoria principal	1-64 GB
10 mseg	Discos magnéticos	1-500 GB
100 sec	Cintas magnéticas	100-1000 GB

- A medida que ascendemos tenemos mayor rendimiento y más costo por bit. Entre la memoria principal y la secundaria hay otro tipo de memoria para salvar la brecha. También aumenta la frecuencia de accesos a ese tipo de memoria.

Memoria de acceso aleatorio: aleatorio significa que se puede acceder a cualquier celda de memoria en el mismo tiempo, independientemente de la posición en la estructura de la memoria.

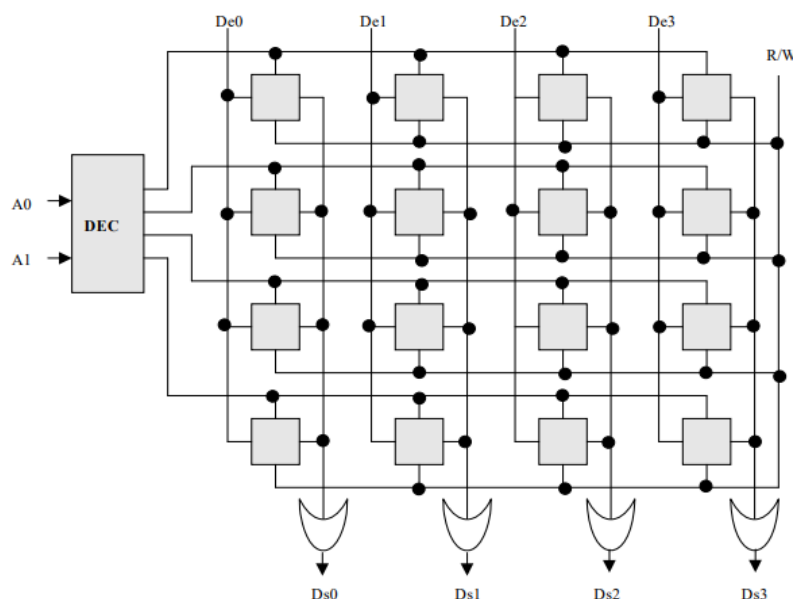
- RAM (Random Access Memory): una característica distintiva de las RAM es que es posible tanto leer datos como escribir rápidamente nuevos datos en ellas. Tanto la lectura como la escritura se ejecutan mediante señales eléctricas. Es una memoria volátil, si se interrumpe la alimentación se pierden los datos. Se pueden utilizar solo como almacenamiento temporal.
  - SRAM (RAM Estática) - DRAM (RAM Dinámica):
    - Ambas son volátiles, es decir, debe aplicarse continuamente tensión de alimentación a la memoria para mantener los valores de los bits.

- Una celda de memoria DRAM es más simple que una SRAM y en consecuencia más pequeña. Por esto, las DRAM son más densas (celdas más pequeñas = más celdas por unidad de superficie) y más económicas que SRAM. DRAM almacena más información que SRAM en la misma superficie. Los 'capacitores' son más chicos que los flip-flop.
- DRAM requiere de circuitería para realizar el refresco. Los 'capacitores' se descargan.
- Las DRAM tienden a ser las preferidas para memorias grandes. Usada como memoria principal.
- Las SRAM son más rápidas. Usada como memoria caché.
- ROM (Read-Only Memory).

Organización de memoria principal: cada chip contiene un arreglo de celdas de memoria. Se organiza internamente como una matriz de celdas de memoria de  $n \times m$ , donde  $n$  es el número de palabras que puede almacenar el chip de memoria y  $m$  es el número de bits por palabra. En las memorias de semiconductor se han empleado dos enfoques organizacionales: 2D y 2½D.

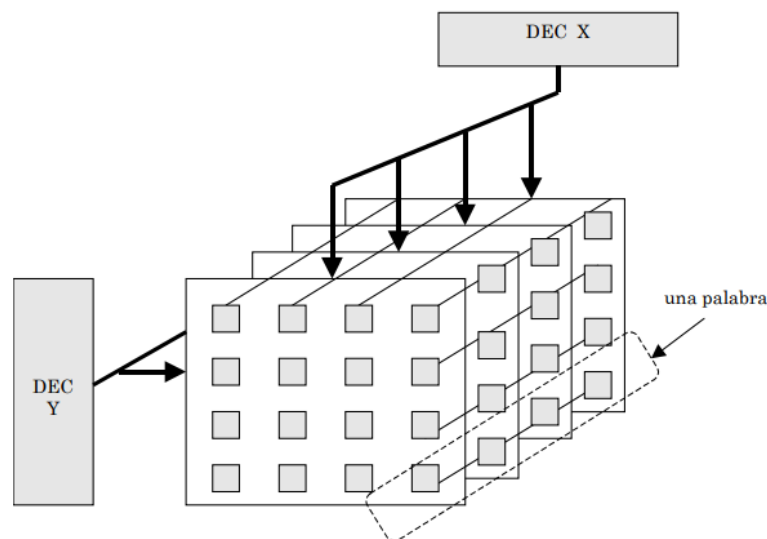
- Comparación 2D - 2½D:
  - En 2D todos los bits están en el mismo chip. En 2½D los bits de una misma palabra estarán en distintos chips.
  - 2D es muy larga y estrecha, con un gran número de palabras de pocos bits. Cada línea de selección de palabra tiene que tener un manejador y conectarse al decodificador. Ocupan mucha superficie.
  - En 2½D al usar decodificación separada de filas y columnas, reduce la complejidad de los decodificadores.
  - 2D dificulta el uso eficaz de los circuitos correctores de error. En 2½D al estar los bits dispersos en distintos chips hay menor probabilidad de error.
- Organización 2D: El arreglo está organizado en  $2^m$  palabras de  $n$  bits cada una. Cada línea horizontal (una de  $2^m$ ) se conecta a cada posición de memoria, seleccionando un renglón. Las líneas verticales conectan cada bit a la salida. El decodificador que está en el chip, tiene  $2^m$  salidas para  $m$  entradas (bits del bus de direcciones).
  - Organización por palabras (lineal).
  - Se utiliza en memorias de capacidad reducida. Gran rapidez de acceso.
  - Esta organización tiene el inconveniente de que el selector (decodificador) de palabras crece exponencialmente con el tamaño de la memoria. Igual le ocurre al número de entradas (fan-in) de las puertas OR que generan la salida de datos.

En la siguiente figura se muestra la organización 2D de un chip de memoria con 4 palabras de 4 bits:



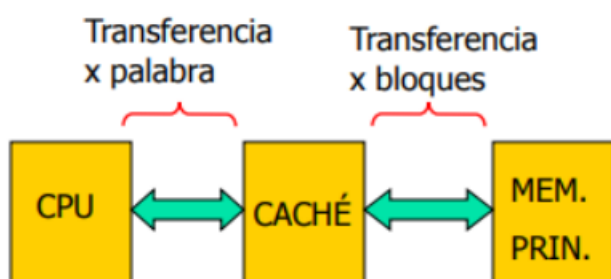
- Organización 2½ D: utiliza dos decodificadores con  $m/2$  entradas y  $2^{m/2}$  salidas. El arreglo es 'cuadrado' y funciona igual que 2D. Los bits de una misma palabra están dispersos en distintos chips.
  - Requiere menos puertas lógicas.
  - Organización por bits.
  - En lugar de una única selección (decodificador) de  $2^n$  salidas en esta organización se utilizan dos decodificadores de  $2^{(n/2)}$  operando en coincidencia. Las líneas de dirección se reparten entre los dos decodificadores (una selección de renglón y una selección de columna).
  - Para una configuración dada de las líneas de dirección se selecciona un único bit de la matriz. Por ello se la denomina también organización por bits.

En esta organización se necesitan varias matrices de celdas básicas, tantas como bits deba tener la palabra de memoria, actuando sobre ellas en paralelo los circuitos de decodificación:



- Cada módulo de memoria cubre el espacio de direccionamiento requerido, pero sólo cubre una parte de la palabra. Una solución puede ser usar varios módulos 'en paralelo'.
- La longitud de la palabra es la deseada, pero los módulos no tienen la capacidad deseada. Una solución puede ser cubrir un cierto rango de direcciones con módulos de memoria 'en serie'. Cada módulo 'estará' en direcciones distintas.

Memoria Caché: su objetivo es lograr que la velocidad de la memoria sea lo mas rapida posible, consiguiendo al mismo tiempo un tamaño grande al precio de memorias semiconductoras menos costosas.



- Hay una memoria principal relativamente grande y más lenta, junto con una memoria caché más pequeña y rápida. La caché contiene una copia de partes de la memoria principal. Cuando el procesador intenta leer una palabra de memoria, se hace una comprobacion para determinar si la palabra está en la caché.
  - Si es así, se entrega dicha palabra al procesador.

- 
- Si no, un bloque de memoria principal, consistente en un cierto número de palabras, se transfiere a la caché y después la palabra es entregada al procesador.
  - En todo momento, un subconjunto de los bloques de memoria reside en líneas de la caché. Si se lee una palabra de un bloque de memoria, dicho bloque es transferido a una de las líneas de caché.
  - La efectividad de la caché se expresa a través de la frecuencia de aciertos: es decir el número de veces que la caché acierta direcciones.
    - Cuando el procesador busca un dato en la memoria se pueden producir dos situaciones: Acierto (hit): los datos están en el nivel superior. Fallo (miss o fault): los datos no están en el nivel superior.

Tipos de memoria externa: en periféricos de almacenamiento de datos

- Discos Magnéticos.
- Discos Ópticos:
  - CD: disco compacto. Un disco no borrrable que almacena informacion de audio digitalizada.
  - CD-ROM: disco compacto de memoria de solo-lectura. Un disco no borrrable usado como una memoria de datos de un ordenador.
  - DVD: disco versatil digital. Una tecnologia para producir representacion de informacion de video digitalizada y comprimida, así como grandes cantidades de otros datos digitales. El DVD basico es de solo-lectura (DVD-ROM).
  - DVD-R: DVD grabable. Similar al DVD-ROM, el usuario puede escribir en el disco una sola vez. Solo se utilizan discos de una cara.
  - DVD-RW: DVD grabable. Similar al DVD-ROM, el usuario puede borrar y reescribir el disco varias veces. Solo se utilizan discos de una cara.
- Cintas Magnéticas.



---

## Periféricos

Los periféricos se pueden dividir en las siguientes categorías:

- Unidades de entrada: teclado, mouse, etc.
- Unidades de salida: monitor, impresora, etc.
- Unidades de almacenamiento masivo: cinta magnética, disco magnético, discos ópticos, etc.
- Unidades de entrada-salida mixtas: terminal interactivo, modem.

### Unidades de Entrada

Los periféricos de entrada, también conocidos como dispositivos de entrada, son todos los componentes de hardware que hacen posible introducir datos o cualquier tipo de información a una computadora. Se les conoce como dispositivos de entrada, justamente por ser el medio que permite la “entrada” de datos provenientes del exterior. El desafío del diseño de dispositivos de entrada de datos manual es reducir el número de partes móviles.

Teclado: los teclados son similares a los de una máquina de escribir, correspondiéndole cada tecla a uno o varios caracteres, funciones u órdenes. Para seleccionar uno de los caracteres de una tecla puede ser necesario pulsar simultáneamente dos o más teclas, una de ellas la correspondiente al carácter.

- Funcionamiento: Al pulsar una tecla se cierra un conmutador que hay en el interior del teclado, esto hace que unos circuitos codificadores generen el código de entrada-salida correspondiente al carácter seleccionado, apareciendo éste en la pantalla si no es un carácter de control.
- Los teclados cuentan con una serie de botones de atajo que pueden ser reconocidos por el sistema operativo. Estas teclas cumplen con diferentes funciones y se representan como F1, F2, F3...
- Muchos teclados funcionan como dispositivos compuestos, ya que pueden traer botones extras que permiten controlar otras tareas. Desde apagar el ordenador, hasta pausar una reproducción multimedia. Igualmente, existen muchos tipos de teclado, siendo el teclado QWERTY el más popular de todos.
- Tasas de entrada muy lentas → 10 caracteres de 8 bits por segundo en teclado.

Mouse: el ratón o mouse es un pequeño periférico que está constituido por una bola que puede girar libremente, y se acciona haciéndola rodar sobre una superficie plana. Los ratones más modernos utilizan un sensor láser que reconoce este movimiento y lo transforma de manera inmediata en datos espaciales de manera más precisa y eficaz.

- Un ratón es un dispositivo pensado para apuntar y luego realizar dos operaciones: click y arrastrar.
- El mouse es más rápido que el teclado: 1 cambio en los bits de la posición X e Y por milisegundo.
  - Click de mouse → bit por 1/10 segundo.

### Unidades de Salida

Se considera un dispositivo de salida a cualquier periférico o unidades que reciben información de una computadora, habitualmente para la visualización, proyección o reproducción física de los datos sugeridos. Entre las funciones de estos dispositivos encontramos las siguientes:

- Tiene como misión obtener todos los datos que el sistema operativo está procesando para su reproducción.
- Adquiere el resultado de los datos procesados del computador relacionado a determinada información o archivos con los que se encuentra trabajando.
- No tienen la capacidad de enviar información, sólo pueden recibirla (unidireccional).
- Emiten los datos procesados de manera que puede ser recibida por el usuario y que permita su interpretación correcta por medio de la emisión de caracteres que caractericen en distintas maneras (pueden ser audio, texto, números, imágenes, entre otros).

---

Monitor: Estos periféricos de salida permiten visualizar y reproducir la información que emite el ordenador por medio de los distintos elementos luminosos. Al monitor se conoce también como VDT (terminal de visualización de vídeo) y VDU (unidad de visualización de vídeo), este dispositivo está compuesto por: circuitos, una pantalla, una fuente de alimentación, botones para ajustar la configuración de la pantalla y carcasa que incluye todos estos componentes.

- Los primeros monitores de computadora estaban compuestos por un tubo de rayos catódicos y una pantalla fluorescente. En la actualidad este instrumento está diseñado para utilizar la tecnología de pantalla plana, habitualmente retro iluminada por un diodo emisor de luz (LED).
- Las principales características de los monitores son las siguientes:
  - Permite visualizar los procesos de los computadores personales.
  - Solo es utilizado para mostrar información, no proporciona una fuente de entrada.
  - Los monitores actuales poseen una alta resolución, se manifiestan por los pixeles que representan los puntos que conforman la imagen digital pero va a estar determinada la calidad de imagen por la tarjeta de vídeo del ordenador.
  - Por medio de un programa se puede visualizar el gráfico de rendimiento del procesador.
  - Tolera el uso de programas específicos para gráficos.
  - Permite el uso de distintos tipos de conexiones (DVI, HDMI, VGA, Displayport, Rayo, USB).
- Una imagen de pantalla no es continua sino que se forma por multitud de puntos de imagen (pixel). La pantalla está dividida en celdas, en cada una de las cuales puede ir un carácter. La celda está construida por una matriz regular de puntos de imagen.
- Clasificación según capacidad de mostrar o no colores:
  - Monitor monocromo: los colores usuales en un monitor monocromático son el blanco y negro, ámbar o verde.
    - bits de colores → 1 color = 1 bit.
  - Monitor color (True Color): el color de cada punto se obtiene con una mezcla de los colores rojo, verde y azul, pudiendo programar la intensidad de cada color básico.
    - bits de colores → RGB = 8R 8G 8B = 24 bits = 3 Bytes.
- Clasificación según su capacidad de representación:
  - Pantallas de caracteres (alfanumérico): sólo admiten caracteres (24x80). Se debe poseer una memoria de imagen (RAM) que almacena la información correspondiente a cada celda y una memoria de solo lectura (ROM) donde se almacenan los patrones de los caracteres representados como una matriz de puntos.
  - Pantallas gráficas: permiten trazados de líneas y curvas continuas. El usuario tiene acceso al punto de imagen, pudiendo representar en ellas imágenes configuradas no sólo con las formas de caracteres incluidos en la ROM. Se requiere una memoria de imagen que pueda contener la información de cada punto de imagen (intensidad, color, etc.).
    - La calidad depende de la densidad de puntos de imagen.
- Los principales parámetros que caracterizan a una pantalla son:
  - Tamaño
  - Número de celdas o caracteres
  - Resolución: número de puntos de imagen en pantalla. Este número no depende del tamaño de la pantalla.
    - CGA: 640x200 píxeles.
    - VGA: 640x480 píxeles.
    - SVGA (Super VGA): 800 x 600 píxeles (resolución 800 x 600 en monitores de 14 pulgadas y hasta 1200 x 1600 en 20 pulgadas.).
    - XGA: 1024x768 píxeles.
    - SXGA: 1280x1024 píxeles.
    - FHD (No Entrelazada): 1920x1080 píxeles.
- Memoria de video de una pantalla:
  - Pantalla de caracteres (alfanumérico) → filas x columnas x (8 + 1 bit por atributo + bits de colores).
  - Pantalla gráfica → resolución x bits de colores.

- Velocidad de transmisión (bps) = (altura de imagen x base de imagen x bits de colores) / tiempo
- Bits de colores:
  - = [ memoria de video / (filas x columnas) ] - (8 + bits de atributo)
  - = (velocidad x tiempo) / (altura de imagen x base de imagen)
- Cantidad de colores =  $2^{(\text{bits de colores})}$

Impresora: Se encarga de tomar los datos electrónicos almacenados en una computadora u otro dispositivo y genera una copia impresa. Las impresoras tienen dos partes diferenciadas: la parte mecánica y la parte electrónica. La parte mecánica, además de encargarse de seleccionar el carácter a partir del código de entrada-salida correspondiente, debe dedicarse a la alimentación y arrastre del papel.

- Las características de las impresoras son las siguientes:
  - Permite obtener el documento físico que se encuentra guardado en el computador personal.
  - Admite la impresión a color o blanco y negro.
  - Las atribuciones pueden cambiar según el tipo de impresora que se vaya a utilizar debido a que posee funciones específicas de impresión.
  - Algunos modelos permiten la impresión desde dispositivos móviles.
  - Así mismo algunos modelos admiten el almacenamiento en la nube.
  - Permite la conexión del periférico por medio de un cable o de forma inalámbrica.
  - Algunos modelos permiten la impresión de forma directa desde un dispositivo Pendrive.
  - Emite un informe de consumo y gasto de la tinta.
  - Tiene la capacidad de realizar ampliaciones y reducciones.
  - Según el modelo de impresión puede presentarse un número de impresiones por minuto o por segundo.
  - El mismo caso se presenta para la calidad de la impresión, la cual va a estar determinada por la cantidad de puntos por pulgada que puede imprimir un determinado modelo.
- La impresora se compone de los siguientes elementos:
  - Cartucho de tinta.
  - Abrazaderas de cartuchos de tinta.
  - Soporte del papel.
  - Alimentador de hojas.
  - Cubierta.
  - Bandeja de salida.
  - Guías laterales.
  - Cabezal de impresión.
  - Panel de control.
  - Botones.
  - Indicadores luminosos.
- Clasificación según la calidad de impresión: tiene en cuenta la calidad de presentación y de contraste de los caracteres impresos.
  - Impresoras normales: impresoras de línea, de rueda y térmicas.
  - Impresoras de semicalidad: impresoras matriciales.
  - Impresoras de calidad: impresoras margaritas y láser.
- Clasificación según el fundamento del sistema de impresión:
  - Impresoras por impacto: son similares a las máquinas de escribir. Son muy ruidosas y tradicionalmente han sido las más utilizadas. Impresoras de rueda, bola, margarita, matriciales, cilindro, cadena, etc.
  - Impresoras sin impacto: forman los caracteres sin necesidad de golpes mecánicos y utilizan otros principios físicos para transferir las imágenes al papel. Impresoras térmicas, de inyección de tinta y láser.
- Clasificación según la forma de imprimir los caracteres (Métodos de Impresión):
  - Impresoras de caracteres: realizan la impresión por medio de un cabezal que va escribiendo la línea carácter a carácter. El cabezal se desplaza a lo largo de la línea que se está imprimiendo.

- Las impresoras de caracteres, como las matriciales, imprimen en un rango de velocidad entre 200 y 400 caracteres por segundo (cps), que supone de 90 a 180 líneas por minuto (lpm).
  - Impresoras matriciales, de inyección de tinta, térmicas y de margarita.
- Impresoras de líneas: se imprimen simultáneamente todos o varios de los caracteres correspondientes a una línea de impresión.
  - Las impresoras de línea presentan un amplio rango de velocidades, desde 400 a 2000 líneas por minuto.
  - Impresoras de cinta, de cadena y de tambor.
- Impresoras de páginas: se caracterizan por contener un tambor rotativo donde se forma con tinta o tóner la imagen de la página a imprimir.
  - La velocidad de las impresoras de página oscila entre 4 y 800 páginas por minuto (ppm) para impresiones en blanco y negro, y la décima parte para la impresión en color.
  - Impresoras láser.
- Monocromático → 1 pixel = 1 bit.
- Cantidad de información transferida = número de caracteres x filas x columnas.
- Parámetros de una impresora:
  - Velocidad de escritura.
  - Caracteres por línea.
  - Ancho del papel o longitud del carro.
  - Densidad de líneas.
  - Tipos de letras.
  - Color.
  - Resolución.

## Unidades de Almacenamiento

Los dispositivos de almacenamiento guardan la información en un soporte que no permite el acceso inmediato desde el programa y se requiere un paso previo de lectura que recupera dicha información desde el almacenamiento y lo coloca en la memoria.

- Principales características:
  - Reutilizables.
  - Elevada capacidad de almacenamiento.
  - No volátiles.
  - Más económicos que la memoria principal.
- Se distinguen los siguientes tipos de dispositivos de almacenamiento:
  - Discos magnéticos.
  - Cintas magnéticas.
  - Discos ópticos.

**Discos magnéticos:** son sistemas de almacenamiento de información que constituyen el principal soporte utilizado como memoria masiva auxiliar. A pesar de que son más costosos que las cintas magnéticas, son sistemas de acceso directo, y con ellos se consiguen tiempos medios de acceso menores que con las cintas magnéticas.

- Tiempos:
  - Tiempo de búsqueda de la pista (tseek): Tiempo que tarda la cabeza en posicionarse en la pista correcta.
  - Tiempo de espera al sector (tlatencia): Tiempo que tarda el sector en alcanzar la cabeza.
    - El retardo rotacional medio de los discos es de unos 2 ms (velocidad de 3600 rpm a 15000 rpm).
    - El retardo rotacional medio de los disquetes está entre los 100 y 50 ms (velocidad de 300 a 600 rpm).
  - Tiempo de acceso = Tiempo de búsqueda de la pista + Tiempo de espera al sector.
    - En las unidades de cabezas fijas → Tiempo de acceso = Tiempo de espera al sector.

- Tiempo de Transferencia de datos: tiempo que tarda la operacion de lectura o escritura (desplazando el sector bajo la cabeza) →  $T. \text{ de transferencia} = \text{número de bytes a transferir} / (\text{número de bytes de una pista} \times \text{velocidad de rotación})$
- Tiempo Total:  $T. \text{ de Acceso} + T. \text{ de Transferencia de datos}$
- Capacidad del disco = capacidad del sector x sectores/pista x pistas/cara x número de caras
  - Grabación en varias zonas: para aumentar la capacidad, los discos duros modernos utilizan la grabacion en varias zonas, en la que la superficie se divide en varias zonas concentricas (16).
    - Dentro de una zona, el número de bits por pista es constante.
    - Las zonas más alejadas del centro contienen más bits (más sectores) que las zonas proximas al centro.
    - Permite capacidades de almacenamiento mayores a expensas de una circuiteria de alguna forma más compleja.
    - Como la cabeza del disco se mueve de una zona a otra, la longitud, a lo largo de la pista, de los bits individuales cambia, provocando un cambio en el tiempo de lectura y escritura.
- Básicamente existen cuatro tipos de unidades de discos:
  - Discos de cabezas fijas.
  - Paquetes de discos.
  - Discos Winchester (disco rígido).
  - Disquetes.
- Parámetros que caracterizan un disco:
  - Tipo de disco.
  - Capacidad.
  - Tamaño.
  - Tiempo medio de acceso.
  - Velocidad de transferencia, de rotación.
  - Número de superficies, de cabezas, de pistas, de sectores por pista, de palabras por sector.
  - Bits por palabra.
  - Densidad máxima.
  - Código de grabacion.

Cintas magnéticas: es el medio más barato para almacenar grandes cantidades de datos. Las cintas magneticas son un soporte de informacion barato y de gran capacidad, pero son muy lentas (acceso secuencial).

- La capacidad de una cinta depende fundamentalmente de su longitud, densidad de grabación, longitud de bloque y formato de grabación.
- Tipos de cinta magnética:
  - Cintas tradicionales:
    - De columnas de vacío.
    - De brazos tensores.
  - Unidades de cassette:
    - De audio.
    - Digitales.
    - Cartuchos.

Discos ópticos: La informacion se almacena en el disco compacto en forma digital (logica binaria) de modo semejante a los de audio.

- Desventajas:
  - En la mayoría de estos dispositivos, una vez grabados no pueden ser reutilizados para escribir. Lo que nos obliga a construir dispositivos de memoria de sólo lectura.
  - Son más lentos que los soportes magnéticos.
- Ventajas:
  - Gran compactación: pueden almacenar entre 60 y 100 veces más datos que un disco magnetico de igual diametro.
  - Acceso directo.

- Alta velocidad: el tiempo de acceso es similar a los discos magnéticos y la velocidad de transferencia es mayor debido a la mayor densidad de grabación.
- Bajo costo.
- Capacidad de almacenamiento.
- La gran ventaja de los discos ópticos es la permanencia de la información durante tiempos muy superiores a la grabada en soportes magnéticos.
- Diferencias entre CD y DVD:
  - Un disco DVD tiene la misma apariencia que un CD, pero pueden contener toda la información de 25 CDs y ofrece imagen y sonido digital de calidad superior a la del tradicional disco compacto.
  - Entre las características que diferencian al DVD del CD, figuran su mayor resistencia a cambios de temperatura y la forma de lectura de la información.
  - Mientras que en el CD se encarga de la lectura un rayo láser de infrarrojos, en el DVD esta labor la realiza con un rayo láser dual con diferente longitud de onda, capaz de leer las distintas capas del disco. El láser rojo tiene una longitud de onda más corta, lo que ofrece un mayor espacio de almacenaje y una mayor capacidad para evitar errores por tiempo.
  - La razón fundamental por la que el DVD puede contener mucha más información que un CD es que, además de albergar distintas capas, las hendiduras o pits en las que se registra la información son más pequeñas en el DVD que en el CD, así como es más pequeña la separación entre las distintas pistas.
  - A diferencia del disco compacto convencional, el DVD puede almacenar información por ambas caras y en distintas capas. Según el número de capas de que disponga, ofrecerá mayor o menor capacidad de almacenamiento de información.
  -

## Unidades de E/S o Mixtas

Un periférico mixto es aquel periférico que puede cumplir funciones tanto de entrada como de salida. También son llamados dispositivos de entrada-salida. En los dispositivos o periféricos mixtos (o de entrada-salida) el flujo principal de datos es en ambos sentidos.

**Terminales interactivos:** la combinación de un monitor de video con su correspondiente teclado se llama frecuentemente terminal y es normal acoplar varios terminales a un computador que se encarga de procesar las distintas tareas que cada usuario le ordena. Una terminal es un teclado y un monitor que permite al usuario interactuar con la computadora mediante una línea de comando.

- Terminales no inteligentes (tontas): Sólo son capaces de ejecutar operaciones de entrada-salida simples. Dispositivo que permite el acceso a un ordenador remoto y es capaz de transmitirle comandos. Cuando recibe una respuesta por parte del computador encargado de procesar las órdenes, la imprime en un papel o la muestra en una pantalla.
- Terminales inteligentes: Capaces de ejecutar ciertos procesos tales como manipulación de texto, posibilidades gráficas o programas simples dirigidos por menú para ayudar a la entrada de datos. Esto es posible al incluir microprocesadores en los terminales. Cuenta con algunos elementos similares a los del PC, como una CPU. Aunque es capaz de resolver algunos comandos por sí sola y funcionar de manera independiente, lo cierto es que continúa muy ligada al ordenador central.

**Modem:** dispositivo que permite conectar dos computadores remotos utilizando la línea telefónica de forma que puedan intercambiar información entre sí. Un módem es un dispositivo que posee conversores A/D y D/A especialmente adecuados para conectar líneas telefónicas al computador. Las señales provenientes de una línea telefónica son interpretadas y atendidas por el modem, permitiendo que otra computadora transmite información directamente a la nuestra.

- Modulación-demodulación: proceso de transformación de la información para que pueda ser transportada por el canal de comunicación.
- Baudios: número de veces de cambio de la señal por segundo en la línea de transmisión. Los modems modernos pueden enviar 4 o más bits por baudio.

- 
- La máxima tasa baudio para el sistema telefónico es 2400.
  - Bits por segundo: número efectivo de bits por segundo que se transmiten en una línea por segundo.
    - Un módem de 600 baudios puede transmitir a 1200, 2400 y 19200 bps.
  - Parámetros de un modem: A las terminales y computadoras se les llama DTE y a los circuitos (módem) de conexión con la red se les llama DCE. El DCE es el proveedor del servicio, mientras que el DTE es el dispositivo conectado. Los DCE se encargan de transmitir y recibir bits uno a uno. Los DTE y DCE están comunicados y se pasan tanto datos de información como de control. Para que se puedan comunicar dos DTE hace falta que ambos cooperen y se entiendan con sus respectivos DCE. También es necesario que los dos DCE se entiendan y usen los mismos protocolos.
    - Equipo terminal de datos (DTE): Terminal, Impresora, Ordenador Personal, Router o Puente.
      - El término DTE se utiliza principalmente para aquellos dispositivos que visualizan información del usuario. También incluye los dispositivos que almacenan o generan datos para el usuario. Las unidades del sistema, los terminales y las impresoras todos se encuentran en la categoría DTE.
    - Equipo de terminación de circuito de datos (DCE): Módem o CSU/DSU.
      - DCE incluye los dispositivos que pueden utilizarse para ganar acceso a un sistema a través de las líneas de telecomunicaciones. Las formas más comunes de DCE son los módems y los multiplexores.
    - Línea de transmisión que une los DCE.