

Resumen para Final

Organización de Computadoras

UNIDAD 1: Computadoras digitales

Conceptos Introductorios

Cuando hablamos de la **Arquitectura** de computadoras hacemos referencia a los atributos de un sistema que son visibles para un programador, o, en otras palabras, son aquellos atributos que tienen un impacto directo en la ejecución lógica de un programa. En cambio, la **Organización** de computadoras refiere a las unidades funcionales y sus interconexiones, que dan lugar a especificaciones arquitectónicas. Entre los atributos de organización se incluyen detalles de hardware invisibles al programador tales como señales de control, interfaces entre el computador y los periféricos y la tecnología de memoria usada. Se puede decir que una arquitectura puede perdurar muchos años, pero la organización cambia con el avance de la tecnología.

Funcionamiento Básico

Las funciones básicas que una computadora puede llevar a cabo son 4:

- **Procesamiento de datos:** el procesador tiene que ser capaz de procesar datos. Estos datos pueden adoptar una gran variedad de formas, y el rango de los requisitos de procesamiento es amplio, pero, hay sólo unos pocos métodos o tipos fundamentales de procesamiento de datos.
- **Almacenamiento de datos:** es esencial que una computadora almacene datos, incluso si la computadora está procesando datos al instante los tiene que guardar temporalmente, al menos aquellos datos con los que está trabajando en un momento dado. También, la computadora lleva a cabo una función de almacenamiento de datos a largo plazo.
- **Transferencia de datos:** la computadora tiene que ser capaz de transferir datos entre ella y el mundo exterior. Cuando se reciben o se llevan datos a un dispositivo que está directamente conectado con la computadora, el proceso se conoce como E/S y el dispositivo recibe el nombre de periféricos. El proceso de transferencia de datos a largas distancias se lo conoce como comunicación de datos.
- **Control:** debe haber un control de las últimas tres funciones. Este control es ejercido por los entes que proporcionan a la computadora instrucciones. Una unidad de control gestiona los recursos de la computadora y dirige las prestaciones de sus partes funcionales en respuesta a las instrucciones.

Estructura

Una computadora es una máquina digital y síncrona con cierta capacidad de cálculo y comunicación con el mundo exterior. Es **digital** porque se maneja con señales eléctricas y la información se representa por medio de dos valores (0 y 1). Es **síncrona** porque todas las operaciones internas se realizan en instantes de tiempos definidos y controladas por un reloj central. Posee **capacidad de cálculo** numérico y lógico mediante la Unidad Aritmético-Logical (ALU) que es la encargada de realizar las operaciones y tiene **comunicación con el mundo exterior** mediante módulos de E/S que permite conectar dispositivos periféricos, con los que puede realizar operaciones de entrada y salida de datos.

Una computadora es un sistema complejo, por lo tanto, se organiza como un sistema jerárquico. Un sistema jerárquico es un conjunto de subsistemas interrelacionados, en el cual,

cada uno se organiza en una estructura jerárquica, hasta que se alcanza el nivel más bajo de subsistemas esenciales. Esta organización es esencial, tanto para el diseño de la computadora, como también, para su descripción. El diseñador necesita tratar con un nivel a la vez. El comportamiento de cada nivel depende solo de una caracterización abstracta y simplificada del sistema que hay en el siguiente nivel más bajo.

Una computadora posee 4 componentes estructurales básicos:

- **Unidad Central de Procesamiento (CPU):** Controla el funcionamiento de la computadora y lleva a cabo sus funciones de procesamiento de datos.
- **Memoria Principal:** Almacena los datos.
- **E/S:** Transfiere datos entre la computadora y el mundo exterior.
- **Sistemas de Interconexión:** Es un mecanismo que proporciona la comunicación entre la CPU, la memoria y la E/S.

A su vez, la CPU se puede dividir en 4 elementos:

- **Unidad de Control:** Controla el funcionamiento de la CPU y de toda la computadora.
- **Unidad Aritmético-Logica (ALU):** Lleva a cabo las funciones de procesamiento de datos de la computadora.
- **Registros:** Proporciona almacenamiento interno a la CPU.
- **Interconexiones CPU:** Son mecanismos que proporcionan la comunicación entre la unidad de control, la ALU y los registros.

Clasificación de las computadoras

Se distinguen 3 tipos de clasificaciones principales: **según su finalidad, según su funcionamiento y según su clasificación comercial.**

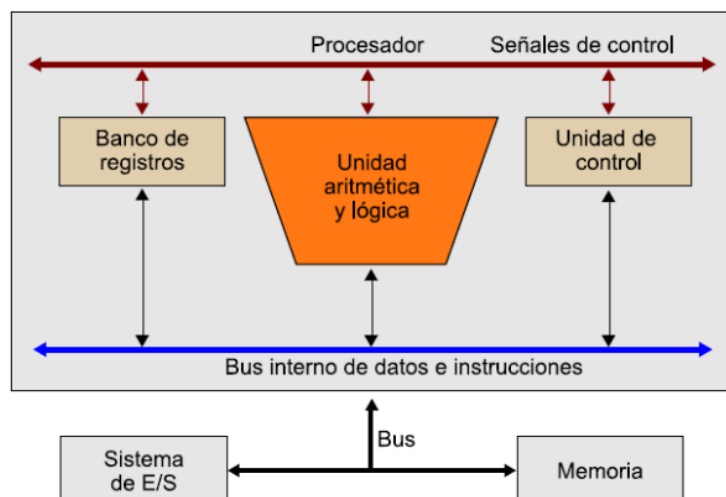
1. *Según su finalidad:*
 - **Propósito específico:** son construidas para realizar funciones concretas.
 - **Propósito general:** son construidas pensando en un rango de aplicación mucho más amplio, por ejemplo, la gestión de un almacén, el procesamiento de textos.
2. *Según su funcionamiento:*
 - **Computadora digital:** procesa datos cuya representación responde a valores como 0 y 1, operando con ellos en distintas y diversas etapas sucesivas.
 - **Computadora analógica:** tienen semejanza con instrumentos de medida como amperímetros o voltímetros.
 - **Computadora híbrida:** posee características de los dos tipos anteriores. Generalmente, los cálculos se realizan de forma analógica y la entrada y salida de datos se hace de forma digital.
3. *Según su clasificación comercial:*
 - **Supercomputadoras:** son las computadoras más potentes, complejas, grandes y costosas, utilizadas por científicos y técnicos para resolver cuestiones de alta complejidad.
 - **Mainframes:** son computadoras con alta velocidad de proceso y capacidad de memoria, utilizadas a diario por grandes empresas para realizar diferentes operaciones.
 - **Minicomputadoras:** son computadoras de tamaño medio, con costo y tamaño reducido, pero de igual forma cuentan con una capacidad de proceso elevada.

- **Microcomputadoras:** son las computadoras de menor tamaño, capacidad de proceso y memoria, utilizadas mayoritariamente a nivel doméstico por su relativo bajo costo.

Arquitectura Von Neumann

La mayoría de las computadoras actuales de propósito general, presentan una estructura interna basada en la arquitectura definida por **John von Neumann** en los años 40, mientras colaboraba con el proyecto ENIAC. Esta arquitectura define que una computadora está compuesta por cuatro elementos principales: Una *Unidad Central de Proceso (CPU)*, una *Memoria Principal (MP)*, un *Módulo de Entrada/Salida (E/S)* y un *Sistema de Interconexión* para estos componentes. La característica principal de esta arquitectura, es que la Unidad Central de Proceso está conectada a una única memoria donde se guardan conjuntamente, datos e instrucciones. La memoria es accesible por posición, independientemente si se trata de datos o instrucciones, y la ejecución de las instrucciones se realiza de forma secuencial. En la arquitectura von Neumann, el tamaño de la unidad de datos o instrucciones está fijado por el ancho del bus que comunica la memoria con la CPU, por lo que, si tenemos un procesador con un bus de 8 bits, este tendrá que manejar datos e instrucciones de una o más unidades de 8 bits de longitud. Si se tiene que acceder a una instrucción de mayor tamaño, tendrá que realizar más de un acceso a memoria, lo que implica las siguientes limitaciones:

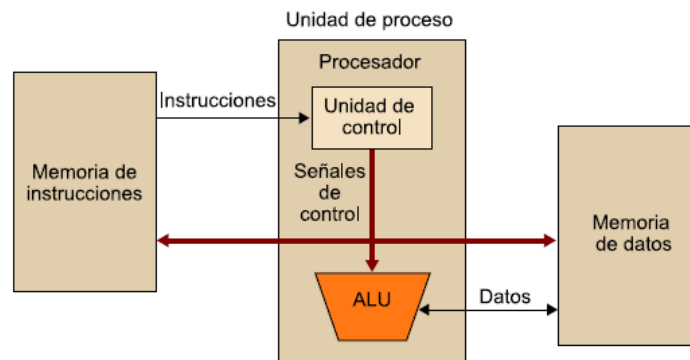
- Se requieren múltiples accesos para manejar instrucciones complejas.
- Al tener un único bus, no se puede buscar una nueva instrucción hasta que no finalice la que está en curso.
- Esto supone una limitación en la velocidad de operación del procesador.



Arquitectura Harvard

La arquitectura Harvard fue diseñada para atacar las limitaciones de la arquitectura von Neumann. Se propuso una solución conceptualmente sencilla: construir un procesador unido a dos tipos diferentes de memoria (una memoria de instrucción y otra memoria de datos) mediante dos buses independientes entre sí (bus de instrucciones y bus de datos). Esto le permite a la computadora leer y ejecutar instrucciones de forma simultánea. Posee las siguientes ventajas respecto a la arquitectura von Neumann:

- El tamaño de las instrucciones no está relacionado con el tamaño de los datos, por lo tanto, puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria.
- El tiempo de acceso a las instrucciones se puede superponer con el tiempo de acceso a los datos, lo que nos proporciona una mayor velocidad de procesamiento.



Evolución histórica de las computadoras

◊ GENERACIÓN CERO: COMPUTADORAS MECANICAS (1642 – 1944).

- **Calculadora de Pascal (1642):** construida por Blaise Pascal, podía sumar y restar. La construyo para ayudar a su padre a calcular impuestos.
- **Máquina de Leibniz (1694):** construida por von Leibniz basada en la máquina de Pascal pero que además podía multiplicar y dividir.
- **Z1 de Konrad Zuse (1938):** fue una calculadora mecánica basada en el sistema binario que operaba con electricidad y tenía capacidad de ser programada.
- **Harvard Mark I (1944):** recibía instrucciones y datos desde tarjetas perforadas y era capaz de realizar cinco operaciones (suma, resta, multiplicación, división y referencia a resultados anteriores).

◊ PRIMERA GENERACIÓN: TUBOS DE VACÍO (1946 – 1958).

- Computadoras construidas por **válvulas de vacío**.
- Disipaban gran cantidad de calor y ocupaban una superficie muy amplia.
- Las tareas se ejecutaban en forma secuencial y las operaciones de E/S estaban encadenadas en el tiempo.
- **ENIAC (1946):** construida en la Universidad de Pennsylvania, fue la primera computadora de propósito general, creada para dar respuesta a las necesidades militares de USA.
- **EDVAC (1949):** construida con la intención de resolver muchos de los problemas de la ENIAC. Recibió varias actualizaciones, incluyendo un dispositivo de E/S para tarjetas perforadas, memoria adicional y una ALU.
- **IAS (1952):** creada por von Neumann, era una computadora de programa almacenado. La estructura general de esta computadora contaba con: Una MEMORIA PRINCIPAL, una ALU, una UC y un dispositivo de E/S.

- **UNIVAC I (1951):** fue la primera computadora fabricada para un propósito no militar, además de ser la primera computadora comercial de USA. Era usada tanto para administración como para negocios.

◊ SEGUNDA GENERACIÓN: TRANSISTORES (1955 – 1965).

- Caracterizada por el descubrimiento del **transistor** que permitió, al ser más pequeño, barato y de menor consumo, la creación de computadoras más accesibles en tamaño y precio.

- Aparecen los **primeros lenguajes de programación** de alto nivel (FORTRAN, ALGOL, COBOL). Los programas eran hechos a medida por un grupo de expertos.

◊ TERCERA GENERACIÓN: CIRCUITOS INTEGRADOS (1963 – 1971).

- Comienza con la invención del **Circuito Integrado de Silicio** (varios componentes en un solo bloque) que posibilitó la inserción de varios transistores en un solo chip.

- Circuitos del tipo **SSI (Small Scale Integration)** y **MSI (Medium Scale Integration)**, que permitieron un gran incremento en la velocidad interna de la computadora y la reducción del consumo energético.

◊ CUARTA GENERACIÓN: INTEGRACIÓN A GRAN ESCALA (1971 – 1990).

- Con la tecnología **LSI (Large Scale Integration)** se logró incluir una CPU completa en una sola pastilla.

- Nacen computadoras extremadamente baratas y pequeñas, logrando insertarlas en el mercado industrial.

- El software obtuvo un considerable avance, volviéndose más interactivo para el usuario. Aparecen los procesadores de texto y las hojas de cálculo.

◊ QUINTA GENERACIÓN: ACTUALIDAD (1990 – ACT).

- Surge la competencia por el dominio internacional del mercado de las computadoras.

- El objetivo principal de esta generación, es el desarrollo de un lenguaje que brinde la posibilidad de comunicarse con las computadoras en un lenguaje cotidiano y no a través de códigos.

- Se desarrollan las **microcomputadoras** y las **supercomputadoras**.

- En los microprocesadores aparecen técnicas para aumentar su velocidad, tales como la **segmentación de instrucciones** y la **paralelización**.

Diseño para conseguir mejores prestaciones

Velocidad del microprocesador

- **Predicción de ramificación:** el procesador se anticipa al software y predice qué ramas, o grupos de instrucciones, se van a procesar después con mayor probabilidad. Si el procesador acierta la mayoría de las veces, puede pre captar las instrucciones correctas y almacenarlas para mantener al procesador ocupado.

- **Análisis del flujo de datos:** el procesador analiza qué instrucciones dependen de los resultados de otras instrucciones o datos, para crear una organización optimizada de

instrucciones. De hecho, las instrucciones se regulan para ser ejecutadas cuando estén listas, independientemente del orden original del programa. Esto evita retrasos innecesarios.

- **Ejecución especulativa:** utilizando los dos anteriores, algunos procesadores ejecutan especulativamente instrucciones antes de que aparezcan en la ejecución del programa, manteniendo los resultados en posiciones temporales. Esto permite al procesador mantener sus máquinas de ejecución tan ocupadas como sea posible, ejecutando instrucciones que es probable que se necesiten.

Equilibrio de prestaciones

Mientras que la velocidad del procesador y la capacidad de la memoria han crecido rápidamente, la velocidad con la que los datos pueden ser transferidos entre la memoria principal y el procesador se ha quedado dramáticamente retrasada. La interfaz entre el procesador y la memoria principal es el camino más importante de todo el computador, ya que es el responsable de llevar el constante flujo de instrucciones y datos entre los chips de la memoria y el procesador. Si la memoria o la interfaz no logran mantener el ritmo de las insistentes demandas del procesador, éste se estanca en una posición de espera y se pierde así tiempo de procesamiento valioso.

Conforme los computadores se hacen más rápidos y potentes, se desarrollan aplicaciones más sofisticadas, que se apoyan en el uso de periféricos con demandas intensivas de E/S. La generación actual de procesadores puede manejar los datos producidos por estos dispositivos, pero aún queda el problema de mover esos datos entre el procesador y los periféricos. Las estrategias en relación con esto incluyen esquemas de caches y almacenamiento, más el uso de buses de interconexión de más alta velocidad y con estructuras más elaboradas.

Rendimiento en las computadoras

Se define rendimiento de un sistema como la capacidad que tiene dicho sistema para realizar un trabajo en un determinado tiempo. Es inversamente proporcional al tiempo, es decir, cuanto mayor sea el tiempo que necesite, menor será el rendimiento.

Los computadores ejecutan las instrucciones que componen los programas, por lo tanto, el rendimiento de un computador está relacionado con el tiempo que tarda en ejecutar los programas. De esto se deduce que el tiempo es la medida del rendimiento de un computador. El rendimiento de un procesador queda en función de tres factores:

- **Frecuencia de la CPU**, la cual depende fundamentalmente de la tecnología de fabricación del procesador. Cuanto mayor sea la frecuencia de la CPU, mejor será el rendimiento.

- **Número de instrucciones del programa** el cual depende del programador, del lenguaje de programación y del compilador. Cuanto mayor sea el número de instrucciones del programa peor rendimiento tendrá.

- **CPI** que depende de diseño interno o arquitectura del computador y del software o instrucciones que se hayan elegido. Es importante optimizar el programa con instrucciones que tengan pocos ciclos. Cuanto mayor sea el CPI, peor será el rendimiento.

Unidades de medida de rendimiento global

4. MIPS

Es el acrónimo de 'Millones de Instrucciones Por Segundo'. Es una forma de medir la potencia de los procesadores. Sin embargo, esta medida sólo es útil para comparar procesadores con el mismo juego de instrucciones y usando benchmarks que fueron compilados por el mismo compilador y con el mismo nivel de optimización. Esto es debido a que la misma tarea puede necesitar un número de instrucciones diferentes si los juegos de instrucciones también lo son; y por motivos similares en las otras dos situaciones descritas.

$$MIPS = \frac{N^{\circ} \text{ Inst. Programa}}{T_{\text{Programa}} * 10^6} = \frac{N^{\circ} \text{ Inst. Programa}}{N^{\circ} \text{ Inst. Programa} * CPI * T_{CPU} * 10^6} = \frac{F_{CPU}}{CPI * 10^6}$$

La ventaja de esta unidad de medida es su fácil comprensión ya que un mayor número de MIPS indicará una mayor velocidad de la máquina. En el mundo de Linux se suelen referir a los MIPS como 'BogoMips'. El equivalente en la aritmética de punto flotante de los MIPS son los flops.

5. FLOPS

Es el acrónimo de Floating point Operations Per Second (operaciones de punto flotante por segundo). Se usa como una medida del rendimiento de una computadora, especialmente en cálculos científicos que requieren un gran uso de operaciones de coma flotante.

Surgen ya que los MIPS no hacen distinción entre operaciones normales y operaciones en coma flotante. Las computadoras exhiben un amplio rango de rendimientos en punto flotante, por lo que a menudo se usan unidades mayores que el FLOPS.

Los prefijos estándar del Sistema Internacional de Medidas pueden ser usados para este propósito, dando como resultado:

- o MegaFLOPS (MFLOPS, 10^6 FLOPS).
- o GigaFLOPS (GFLOPS, 10^9 FLOPS).
- o TeraFLOPS (TFLOPS, 10^{12} FLOPS).
- o PetaFLOPS (PFLOPS, 10^{15} FLOPS).
- o ExaFLOPS (EFLOPS, 10^{18} FLOPS).

6. Benchmark

Es una técnica utilizada para medir el rendimiento de un sistema o componente del mismo, frecuentemente en comparación con el que se refiere específicamente a la acción de ejecutar un benchmark. La palabra benchmark se traduce al castellano como comparativa.

El benchmark es el resultado de la ejecución de un programa informático o un conjunto de programas en una máquina, con el objetivo de estimar el rendimiento de un elemento concreto, y poder comparar los resultados con máquinas similares.

También, puede realizarse un “benchmark de software”, es decir, comparar el rendimiento de un software contra otro o de parte del mismo, por ejemplo, comparar distintas consultas a una base de datos para saber cuál es la más rápida o directamente parte de código.

Niveles de Abstracción

Para describir un sistema jerárquico se puede utilizar dos aproximaciones: empezar por la parte más baja y se van construyendo los niveles de jerarquía basándose en el nivel inferior (modelo *bottom-up*) o se realiza una descripción desde el nivel más alto y se descompone el sistema en módulos de jerarquía inferior (modelo *top-down*). Una posible clasificación de los niveles estructurales, de menor a mayor, sería:

1. **Nivel de componentes.**
2. **Nivel electrónico**
3. **Nivel digital**
4. **Nivel RLT (nivel de transferencia de registros)**
5. **Nivel PMS**

UNIDAD 2: Aritmética de Computadoras

Definición de Bit, nibble, byte, palabra, relación con lenguajes de alto nivel.

- **Bit:** Es el acrónimo de Binary Digit (dígito binario) el cual utiliza dos dígitos (el 0 y el 1). Es la unidad mínima de información empleada en informática o en cualquier dispositivo digital. Podemos representar dos valores cuales quiera, basta con asignar uno de esos valores al estado de “apagado” (0) y el otro al estado de “encendido” (1).
- **Nibble o Cuado:** Es el conjunto de 4 bits. Este puede representar una cifra en hexadecimal.
- **Byte:** Es un conjunto de 8 bits contiguos. No es lo mismo que un *octeto*, el cual es un conjunto de 8 bits (dos nibbles). Un byte comprende el *sub-campo direccionable más pequeño* del tamaño de palabra natural de la computadora. Esto es, la unidad de datos binarios más pequeña en que la computación es significativa, o se pueden aplicar las cotas de datos naturales.
- **Palabra:** Es una cadena finita de bits que son manejados como un conjunto por la máquina. Su tamaño hace referencia al número de bits contenidos en ella. El tamaño de una palabra se refleja en muchos aspectos de la estructura y las operaciones de las computadoras. Los ordenadores modernos normalmente tienen un tamaño de palabra de 16, 32 o 64 bits.

La **relación de estas definiciones con los lenguajes de alto nivel** está dada por la elección de los tipos de datos de cada lenguaje, lo que determinara el rango en que puede manejarse cada tipo de dato. Por ejemplo, en C+ se elige un tipo de entero sin signo de 8 bits, el cual está restringido en el rango de 0 a 255.

Teorema Fundamental de la Numeración

El **Teorema Fundamental de la Numeración** relaciona una cantidad expresada en cualquier sistema de numeración posicional, con la misma cantidad expresada en el sistema decimal. La representación generalizada de un número en una base b es la siguiente:

$$N \equiv \dots n_2 * b^2 + n_1 * b^1 + n_0 * b^0 + n_{-1} * b^{-1} \dots$$

El Teorema Fundamental de la Numeración dice que el valor decimal de una cantidad expresada en otro sistema de numeración, está dado por la fórmula:

$$N^d = \sum_{i=-m}^n (digito)_i * (base)^i$$

- n : número de dígitos de la parte entera.
- m : número de dígitos de la parte fraccionaria.
- i : indica la posición del dígito respecto de la coma.
- base: es la base del sistema de numeración (2 para binario, 16 para hexadecimal, etc.)

Sistemas de numeración posicionales

- **SISTEMA DECIMAL**: es un sistema de notación posicional formado por 10 dígitos ($b=10$) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. Por ejemplo, el número $842,97_{10} = (8*10^2 + 4*10^1 + 2*10^0 + 9*10^{-1} + 7*10^{-2}) = 800 + 40 + 2 + 0,9 + 0,07$
- **SISTEMA BINARIO**: es un sistema de notación posicional formado por 2 dígitos ($b=2$) {0, 1} a los que se denominan bits. El número $1011_2 = (1*2^3 + 0*2^2 + 1*2^1 + 1*2^0) = 8 + 0 + 2 + 1 = 11_{10}$
- **SISTEMA HEXADECIMAL**: está formado por 16 dígitos ($b=16$) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F} que representan los decimales del 0 al 15. El número $1A2_{16} = (1*16^2 + 10*16^1 + 2*16^0) = 256 + 160 + 2 = 418_{10}$
- **SISTEMA OCTAL**: es un sistema de notación posicional formado por 8 dígitos ($b=8$) {0, 1, 2, 3, 4, 5, 6, 7}. El número $275_8 = (2*8^2 + 7*8^1 + 5*8^0) = 128 + 56 + 5 = 189_{10}$

Estos cuatro sistemas mencionados son utilizados comúnmente en la electrónica digital. Para identificar el sistema en el que se trabaja se puede utilizar la notación matemática de la base o se puede añadir al final de número una letra para identificar el sistema (D=10, B=2, H=16, O=8)

Representaciones Numéricas: números enteros con y sin signo

- **Números enteros sin signo (BSS)**: Si el número tiene n bits se pueden representar 2^n números distintos. Por ejemplo: $2_{10} = 010_2$. El Rango va desde 0 a (2^{n-1})
- **Números enteros con signo (BCS)**: Con n bits, 1 bit representa al signo (el más significativo) y $n-1$ bits a la magnitud. Un 0 en el bit de signo indica que el número es positivo y un 1 que es negativo. Los otros bits representan al número. Por ejemplo: $+32_{10} = 00100000_2$. El rango va de $-(2^{n-1}-1)$ a $+(2^{n-1}-1)$, con dos representaciones del cero.

- **Numero en Punto Fijo:** Se considera que todos los numero a representar tienen exactamente la misma cantidad de dígitos y la coma fraccionaria está siempre ubicada en el mismo lugar, como en el sistema decimal. En el sistema binario, por ejemplo, $11,10_2 = 3,5_{10}$. La diferencia principal entre la representación en papel y su almacenamiento en la PC es que no se guarda coma alguna, se supone que está en un lugar determinado.

El rango es la diferencia entre el número mayor y menor, por ejemplo, si $n = 3$, el rango es $111 - 000 = 111$. La resolución es la diferencia entre dos números consecutivos.

- **Decimal Codificado en Binario (BCD):** Es un estándar para representar número decimales en sistema binario, en donde cada dígito decimal es codificado con una secuencia de 4 bits. Con esta codificación especial de los dígitos decimales en el sistema binario, se pueden realizar operaciones aritméticas como suma, resta, multiplicación y división de números en representación decimal sin perder en los cálculos la precisión ni tener las inexactitudes en que normalmente se incurre con las conversiones de decimal a binario puro y de binario puro a decimal.

En BCD cada cifra que representa un dígito decimal se representa con su equivalente binario en cuatro bits. Esto es así porque es el número de bits necesario para representar el número 9, el número más alto que se puede representar en BCD.

Con BCD solo se utilizan 10 de las 16 posibles combinaciones que se pueden formar con números de 4 bits, por lo que el sistema pierde capacidad de representación, aunque se facilita la compresión de los números. Para números de más de una cifra hacen falta dos números BCD. Desde que los sistemas informáticos empezaron a almacenar los datos en conjuntos de ocho bits, hay dos maneras comunes de almacenar los datos:

- **Desempaquetado:** Omisión de los cuatro bits más significativos.
- **Empaquetado:** Almacenamiento de dos datos BCD, en el que también se incluye en primer lugar el signo, por lo general 1100 para el positivo y 1101 para el negativo.

Ejemplos:

127 -- Desempaquetado (sin signo): 11110001, 11110010, 11110111 = F1F2F7 (F indica sin signo).

+127 -- Empaquetado (con signo): 00010010, 01111100 = 127C.

+834 -- Desempaquetado (con signo): 11111000, 11110011, 11000100 = F8F3C4.

-834 -- Empaquetado (con signo): 10000011, 01001101 = 834D.

+8435 -- Desempaquetado (con signo): 11111000, 11110100, 11110011, 11000101 = F8F4F3C5.

+8435 -- Empaquetado (con signo): 00001000, 01000011, 01011100 = 08435C.

La ventaja del código BCD frente a la representación binaria clásica es que no hay límite para el tamaño de un número.

Técnica de Complementos

Complemento a 1

Los n bits representan al número. Si el número es positivo, la representación es la misma. En cambio, si es negativo (comienza con 1), los n bits tienen el Ca1 del valor deseado. El Ca1 de un número en base 2 se obtiene invirtiendo todos los bits. Por ejemplo: $+32_{10} = 00100000$; $-32_{10} = 11011111$. El rango es de $-(2^{n-1}-1)$ a $+(2^{n-1}-1)$. Con dos 0.

Complemento a 2

Los n bits representan al número. Si el número es positivo, los n bits tienen la representación binaria del número (como siempre). Si es negativo, los n bits tienen el Ca2 del valor deseado. El Ca2 de un número se obtiene con Ca1 y luego sumándole 1. Por ejemplo: $+32_{10} = 00100000$; $-32_{10} = 11100000_2$. El rango va desde $-(2^{n-1})$ a $+(2^{n-1}-1)$, con un 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). Dado un valor, el número representado se obtiene restando el valor del exceso. $A = (\text{exceso } E \text{ de } A) - E$. El signo del número A resulta de esta resta. Por ejemplo: si $n = 6$, exceso $32 = -2^{6-1} = 00000_2 = 0 - 32$. El rango va desde (-2^{n-1}) a $(2^{n-1}-1)$.

Flags

Son bits que el procesador establece de modo automático acorde al resultado de cada operación realizada. Algunos tipos de banderas son:

- **Z (cero)**: vale 1 sólo si el resultado de la operación son todos los bits 0.
- **C (carry)**: vale 1 si en la suma hay acarreo. En la resta vale 1 si hay borrow. Cuando la operación involucra BSS, C en 1 indica que está fuera de rango.
- **N (negativo)**: vale 1 si el resultado es negativo.
- **V (overflow)**: si vale 1 indica una condición de fuera de rango en Ca2.

Aritmética con enteros

- Negación en Ca2: Se realiza siguiendo las siguientes reglas:
 1. Obtener el complemento booleano de cada bit del entero
 2. Tratando el resultado como un entero binario sin signo, sumarle 1.
- Suma en Ca2: Se suman los bits directamente. Si se suman dos números positivos y el resultado es negativo, o se suman dos negativos y el resultado es positivo, existe *overflow*.
- Resta en Ca2: Se restan los bits directamente y sucede lo mismo que en la suma, si a un número positivo se le resta uno negativo y el resultado es negativo o, si a un número negativo se le resta uno positivo y el resultado es positivo, existe *overflow*.

Fundamentos de la representación en punto flotante, normalización, error de la representación

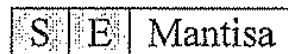
Números en punto flotante

Un número muy grande se puede representar, tanto en la aritmética decimal, como en la binaria de la forma $\pm M \times B^{\pm E}$, donde M es la mantisa, B es la base y E es el exponente.

Como la base en binario siempre es 2, es implícita y no necesita almacenarse. Entonces almacenamos M y E en vez del número completo, ahorramos bits.

Mantisa y Exponente se almacenan en sistemas de punto fijo (BSS, BCS, Ca1, Ca2, Exceso) y de esto dependerá el rango y la resolución (el rango en punto flotante es más grande que en punto fijo).

El formato para la representación en punto flotante es:



Normalización

La **normalización** sirve para hacer operaciones aritméticas teniendo un único par de valores de mantisa y exponentes para un número. Todas las mantisas comienzan con 0,1 o 1,1; por lo tanto, el 1 no lo almacenamos (para ahorrar bits), entonces tenemos un bit más para representar. El 1 que no almacenamos se llama Bit Implícito y no hay que olvidarse que existe.

Ejemplo: Mantisa BCS de 23 bits, fraccionaria normalizada y exponente Exceso 8 bits entero.

- $\text{Max} + = 0,11111111111111111111111 \times 2^{1111111} = +(1-2^{-23}) \times 2^{+127}$
- $\text{Min} + (\neq 0) = 0,10000000000000000000000 \times 2^{0000000} = +(0,5) \times 2^{-128}$
- $\text{Max} - (\neq 0) = 1,00000000000000000000000 \times 2^{0000000} = -(0,5) \times 2^{-128}$
- $\text{Min} - = 1,01111111111111111111111 \times 2^{1111111} = -(1-2^{-23}) \times 2^{+127}$

Ahora, ¿Cómo se escribe un número en Punto Flotante normalizado?

1. Se escribe el número en el sistema propuesto para la mantisa.
2. Se desplaza la coma y se cambia el exponente hasta obtener la forma normalizada (Se le suma tanto como lugares se corrió la coma hacia la izq.).
3. Se convierte el exponente al sistema propuesto para él

Ejemplo: $-13,5_{10}$ en el formato del ejemplo anterior.

1. $1\ 1101,10000 \dots 00000 = 1\ 1101,100 \dots 000 \times 2^0$
2. $1\ 0,1101100 \dots 00000 \times 2^4$ (Se corre la coma 4 lugares, se suma 4 al exponente)
3. 4 en Ca2 = 00000100; 4 en exceso = 10000100.

Y finalmente, lo representamos en la recta:

- Sin bit implícito: 1 10000100 110110000000000000000000

- Con bit implícito: 1 10001000 101100000000000000000000

Resolución

Es la diferencia entre dos representaciones sucesivas y varia 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 absoluto máximo \leq Resolución/2.

Error Relativo

Es el error absoluto dividido por el numero a representar (expresado en porcentaje)

Representación Estándar del IEEE 754

Se desarrollo para facilitar la portabilidad de los programas de un procesador a otro y para alentar el desarrollo de programas numéricos sofisticados. Existen dos representaciones: simple precisión y doble precisión. La mantisa es fraccionaria normalizada, con la coma después del primer bit (que es siempre 1). El exponente está representado en exceso ($2^{n-1} - 1$)

• PRECISIÓN SIMPLE (32 bits).		
SIGNO (1 bit)	EXPONENTE (8 bits)	MANTISA (23 bits)
• PRECISIÓN DOBLE (64 bits).		
SIGNO (1 bit)	EXPONENTE (11 bits)	MANTISA (52 bits)
- EXPONENTE sin signo y en exceso $2^{n-1} - 1$ (127 para precisión simple y 1023 para precisión doble).		

IEEE 754	Simple Precisión	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 a +127	-1022 a +1023
Rango de números	2^{-126} a 2^{127}	2^{-1022} a 2^{1023}

• **CASOS ESPECIALES:**

SIGNO	EXPONENTE	MANTISA	SIGNIFICADO
0	Todos 1	Todos 0	Más infinito ($+\infty$).
1	Todos 1	Todos 0	Menos infinito ($-\infty$).
0 ó 1	Todos 1	Distinta de 0	Not a Number (NaN).
0 ó 1	Todos 0	Todos 0	Ceros (Positivo y Negativo).
0 ó 1	Todos 0	Distinta de 0	Número muy pequeño (cerca al cero).

• **RANGOS DE REPRESENTACION Y ERRORES:**

MAXIMO N° POSITIVO	$MANTISA\ MAXIMA * BASE^{MAXIMO\ EXPONENTE\ POSITIVO}$
MINIMO N° POSITIVO	$MANTISA\ MINIMA * BASE^{MAXIMO\ EXPONENTE\ NEGATIVO}$
MAXIMO N° NEGATIVO	$-(MANTISA\ MINIMA) * BASE^{MAXIMO\ EXPONENTE\ NEGATIVO}$
MAXIMO N° POSITIVO	$-(MANTISA\ MAXIMA) * BASE^{MAXIMO\ EXPONENTE\ POSITIVO}$

- **ERROR ABSOLUTO:** $EA(x) = |N^{\circ} \text{ a Representar} - N^{\circ} \text{ Representado}|$
- **ERROR RELATIVO:** $ER(x) = |N^{\circ} \text{ a Representar} - N^{\circ} \text{ Representado}| / N^{\circ} \text{ a Representar}$

- Representar el número -118.625_{10} en precisión simple:

$$118_{10} = 1110110_2 \quad \text{---} \rightarrow 1110110,101 \quad \text{---} \rightarrow \text{NORMALIZADO: } 1,110110101 * 2^6$$

$$0,625_{10} = 101_2 \quad \text{---} \rightarrow \text{EXPONENTE} \quad \text{---} \rightarrow 6+127 = 133 = 10000101_2$$

SIGNO= 1	EXPONENTE= 10000101	MANTISA= 110110101000000000000000
-----------------	----------------------------	--

Operaciones aritméticas en punto flotante

Suma y Resta

Cuando sumamos o restamos dos números en coma flotante se deben comparar los exponentes y hacerlos iguales, para lo cual hay que desplazar o alinear uno de ellos respecto al otro. Dados dos números en representación en coma flotante como:

$$X = m_x 2^{e_x}$$

$$Y = m_y 2^{e_y}$$

las operaciones de suma y resta se definen de la siguiente forma, suponiendo que $X_e < Y_e$:

$$X + X = (m_x 2^{e_x - e_y} + m_y) 2^{e_y} = (m_x + m_y 2^{e_y - e_x}) 2^{e_x}$$

$$X - X = (m_x 2^{e_x - e_y} + m_y) 2^{e_y} = (m_x - m_y 2^{e_y - e_x}) 2^{e_x}$$

Las operaciones de suma y resta como la multiplicación y la división pueden producir desbordes, por producir resultados demasiado grandes (desbordamientos) o demasiado pequeños (subdesbordamientos). Hay 4 tipos de reposos posibles:

1. **Desbordamiento del exponente:** Es cuando un exponente positivo E excede su valor máximo permitido. En algunos ordenadores el número X se representa entonces como $+\infty$.
2. **Subdesbordamiento del exponente:** Es cuando un exponente negativo E excede de su valor máximo permitido. Esto significa que el número X es demasiado pequeño y se puede considerar como igual a 0.
3. **Subdesbordamiento de mantisa:** En el proceso de alineación de las mantisas, si los dígitos se desplazan hacia la derecha más allá de su bit menos significativo, lo que sucede es que se pierden y es como redondear el resultado.
4. **Desbordamiento de mantisa:** En la suma de dos mantisas del mismo signo se puede producir un arrastre del bit más significativo. Esto se soluciona mediante la renormalización, desplazando a la derecha un bit la mantisa y ajustando el exponente.

Algoritmo de la suma- resta

Para realizar la suma o la resta de dos operandos en representación coma flotante debemos realizar previamente la separación de los exponentes y de las mantisas para su tratamiento posterior, y después realizar esta serie de paso:

1. Seleccionar el numero con menor exponente y desplazar su mantisa a la derecha tantas veces como indique la diferencia en módulo de los exponentes.
2. Hacer que el exponente resultado sea igual al mayor de los exponentes.
3. Realización de las operaciones de suma o resta con las mantisas.
4. Normalización del resultado. Una vez realizada la suma se debe normalizar desplazando los bits de la mantisa hacia la izquierda o la derecha con lo cual habrá que cambiar el valor del exponente.
5. Comprobar las condiciones de rebose.

Multiplicación

La multiplicación y la división en punto flotante son más sencillas de realizar. Vemos la fórmula que nos permitirá realizar estas operaciones manualmente.

$$X * Y = (m_x * m_y) 2^{x_e + y_e}$$

$$X/Y = (m_x / m_y) 2^{x_e - y_e}$$

Algoritmo de multiplicación y división

1. Realizar la Suma-Resta de los exponentes.
2. Multiplicar-Dividir las mantisas y determinar el signo del resultado.
3. Normalizar el valor resultado, si es necesario.
4. Comprobar las condiciones de rebose.

Representaciones alfanuméricas, ASCII, EBCDIC

Representación alfanumérica

- Letras (mayúsculas y minúsculas)
- Dígitos decimales (0-9)
- Signos de puntuación (; , . : etc.)
- Caracteres especiales (* + = etc.)
- Caracteres u ordenes de control (TAB Intro etc.)

FIELDATA: 26 letras mayúsculas + 10 dígitos + 28 caracteres especiales (Total= 64 combinaciones -> código de 6 bits).

ASCII: FIELDATA + minúsculas + ctrl. (Total= 128 combinaciones -> código de 7 bits).

ASCII extendido: ASCII + multinacional + serigráficos + matemática (256 combinaciones -> código de 8 bits).

EBCDIC (Extended BCD Interchange Code): similar a ASCII, pero de IBM (código de 8 bits).

UNIDAD 3: Lógica Digital

Compuertas Lógicas

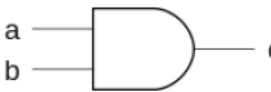

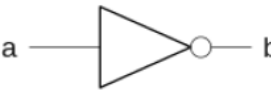
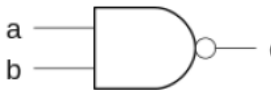


Son diminutos dispositivos electrónicos que pueden calcular diversas funciones con las señales 0 y 1. Las compuertas constituyen la base de hardware sobre la que se construyen todas las computadoras digitales.

Existen 6 tipos de compuertas:

- **NOT o inversora:** sea una variable A con valor 1, diremos que su complemento o estado inverso será $\sim A=0$.
- **AND:** es la multiplicación de dos o más conjuntos. También se denomina intersección de conjuntos y se representa como $A*B$.
- **OR:** es la suma de dos o más conjuntos. Esta operación también se denomina reunión de conjuntos y puede representarse como $A+B$
- **XOR (OR exclusiva)**
- **NAND (NOT AND)**
- **NOR (NOT OR)**

Siendo las dos últimas las más importantes, llamadas “burbujas de inversión” o compuertas universales.

Representación

NOMBRE	SÍMBOLO	FUNCIÓN	TABLA															
AND		$f(a,b) = a * b$	<table><tr><th>A</th><th>B</th><th>f(a,b)</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	f(a,b)	0	0	0	0	1	0	1	0	0	1	1	1
			A	B	f(a,b)													
			0	0	0													
			0	1	0													
			1	0	0													
1	1	1																
OR		$f(a,b) = a + b$	<table><tr><th>A</th><th>B</th><th>f(a,b)</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	f(a,b)	0	0	0	0	1	1	1	0	1	1	1	1
			A	B	f(a,b)													
			0	0	0													
			0	1	1													
			1	0	1													
1	1	1																
NOT		$f(a) = \sim a$	<table><tr><th>A</th><th>f(a)</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	f(a)	0	1	1	0									
			A	f(a)														
			0	1														
1	0																	
NAND		$f(a,b) = \sim(a * b)$	<table><tr><th>A</th><th>B</th><th>f(a,b)</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	f(a,b)	0	0	1	0	1	1	1	0	1	1	1	0
			A	B	f(a,b)													
			0	0	1													
			0	1	1													
			1	0	1													
1	1	0																
NOR		$f(a,b) = \sim(a + b)$	<table><tr><th>A</th><th>B</th><th>f(a,b)</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	f(a,b)	0	0	1	0	1	0	1	0	0	1	1	0
			A	B	f(a,b)													
			0	0	1													
			0	1	0													
			1	0	0													
1	1	0																
XOR		$f(a,b) = a \oplus b$ $= a * \sim b + \sim a * b$	<table><tr><th>A</th><th>B</th><th>f(a,b)</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	f(a,b)	0	0	0	0	1	1	1	0	1	1	1	0
			A	B	f(a,b)													
			0	0	0													
			0	1	1													
			1	0	1													
1	1	0																

Algebra de Boole

Se usa para describir los circuitos que pueden construirse combinando compuertas. Una función booleana tiene una o más variables de entrada y produce un resultado que depende

solo de los valores de dichas variables. Puesto que una función booleana de n variables solo tiene 2^n posibles combinaciones de los valores de entrada, la función puede describirse totalmente con una *Tabla de Verdad* de 2^n renglones, cada uno de los cuales indica el valor de la función para una combinación distinta de valores de entrada.

Distintas combinaciones

A	B	NOT A	A OR B	A AND B
0	0	1	0	0
0	1	1	1	0
1	0	0	1	0
1	1	0	1	1

A	B	A NOR A	A NAND B	A XOR B
0	0	1	1	0
0	1	0	1	1
1	0	0	0	1
1	1	0	0	0

Nota:

- OR siempre es 1 cuando hay un 1 en las variables.
- AND siempre es 0 cuando hay un 0 en las variables.
- XOR es 0 cuando ambas son iguales y 1 cuando son distintas.

Implementación de funciones booleanas

Para implementar una función (F) booleana se usa el “método general para determinar una formula lógica dada una tabla de verdad: el método de la suma y el producto (SOP)”. Con ella, dada una tabla de verdad, se continua con los siguientes dos sencillos pasos:

1. Verificar las salidas con valor 1
2. Sumar todos los términos

Así, se obtiene F y se puede hacer el dibujo de los circuitos, usando compuertas AND y OR, o cambiándolas por compuertas universales o a las otras.

Por ejemplo:

A	B	F	RESULTADO
0	0	0	A NAND B
0	1	1	NOT A AND B
1	0	0	NOT A NAND B
1	1	1	A AND B

Por lo tanto, F es igual a NOT A AND B **OR** A AND B

Lógica combinatoria, codificadores, decodificadores, multiplexores

Circuitos combinatoriales

Un **circuito combinatorial** es un conjunto de puertas interconectadas entre sí, cuya salida en un momento dado, es función solamente de la entrada en ese instante. La aparición de la entrada viene seguida casi inmediatamente por la aparición de la salida, con solo retardos de las puertas. En general, un circuito combinatorial consiste en n entradas y m salidas. Se puede definir de 3 formas:

1. **Tabla de verdad:** para cada una de las 2^n combinaciones posibles de las n señales de entradas, se enumera el valor binario de cada una de las m señales de salida.

2. **Símbolo gráfico:** describe la organización de las interconexiones entre puertas.
3. **Ecuaciones booleanas:** cada señal de salida se expresa como una función booleana de las señales de entrada.

Equivalencias de circuitos

Nombre	Función AND (*)	Función OR (+)
Ley de identidad	$1 * A = A$	$0 + A = A$
Ley nula	$0 * A = 0$	$1 + A = 1$
Ley de idempotencia	$A * A = A$	$A + A = A$
Ley inversa	$A * \text{NOT } A = 0$	$A + \text{NOT } A = 1$
Ley conmutativa	$A * B = B * A$	$A + B = B + A$
Ley asociativa	$(A * B) * C = A * (B * C)$	$(A + B) + C = A + (B + C)$
Ley distributiva	$A + B * C = (A + B) * (A + C)$	$A * (B + C) = A * B + A * C$
Ley de absorción	$A * (A + B) = A$	$A + A * B = A$
Ley de De Morgan	$\text{NOT}(A * B) = \text{NOT } A + \text{NOT } B$	$\text{NOT}(A + B) = \text{NOT } A * \text{NOT } B$

Multiplexores

El **multiplexor** conecta varias entradas con una única salida. En un momento dado, se selecciona una de las entradas para que pase a la salida. Los multiplexores se usan en circuitos digitales para controlar el enrutamiento de señales y datos. También se usan como convertidor de datos paralelos en serie.

Un multiplexor es un circuito con 2^n entradas de datos, una salida de datos y n entradas de control que seleccionan una de las entradas de datos. La entrada de datos seleccionada se “pasa por compuertas” hacia la salida.

Codificadores

Un **codificador** es un circuito combinacional con 2^n entradas y n salidas, cuya misión es presentar en la salida el código binario correspondiente a la entrada activada.}

Existen dos tipos fundamentales: codificadores sin prioridad y codificadores con prioridad. En el caso de los codificadores sin prioridad, puede darse el caso de salidas cuya entrada no pueda ser conocida: por ejemplo, la salida 0 podría indicar que no hay ninguna entrada activada o que se ha activado la entrada número 0.

En cambio, los codificadores con prioridad, cuando existe más de una señal activa, la salida codifica la de mayor prioridad (generalmente correspondiente al valor decimal más alto). Adicionalmente, se codifican dos salidas más: una indica que ninguna entrada está activa, y la otra, que alguna entrada está activa. Esta medida permite discernir entre los supuestos de que el circuito estuviera deshabilitado por la no activación de la señal de capacitación, que el circuito no tuviera ninguna entrada activa, o que la entrada número 0 estuviera activada.

Decodificadores

Un decodificador es un circuito combinacional con varias líneas de salida, con una sola de ellas seleccionada en un instante dado, dependiendo del patrón de líneas de entrada. En general, un decodificador tiene n entradas y 2^n salidas.

Su función es la de direccionar espacios de memoria. Un decodificador de n entradas puede direccionar 2^n espacios de memoria.

Comparadores

El comparador compara dos palabras de entrada. El comparador de la siguiente figura acepta dos entradas, A y B, cada una de 4 bits y produce 1 si son iguales y 0 en caso contrario. El circuito se basa en la compuerta XOR.

Circuitos secuenciales

Se usan para proporcionar memoria o información de estado. La salida actual de un circuito secuencial depende no sólo de la entrada actual, sino también de la historia pasada de las entradas. La salida actual depende de la entrada actual y del estado actual del circuito.

Biastable

Un **Biastable**, también llamado Flip-Flop, es la forma más sencilla de un Circuito Secuencial, con la capacidad de permanecer en uno de dos estados posibles (0 y 1) durante un tiempo indefinido en ausencia de entrada, utilizando el principio de retroalimentación. Posee dos salidas que siempre son complementarias entre sí (Q y \bar{Q}) y puede funcionar como una memoria de 1 bit. Dependiendo del tipo de entradas, se puede clasificar en:

- **Sincronismo activado por nivel:** el sistema lee las entradas cuando el reloj está en estado ALTO (1 o H) o BAJO (0 o L). También llamado activación por pulso.

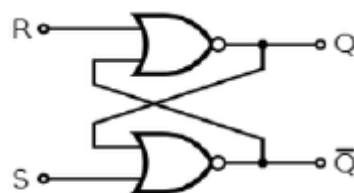


- **Sincronismo activado por flanco:** el sistema lee las entradas cuando se produce la transición de sus pulsos, puede ser: ascendente (cuando cambia de 0 a 1) o descendente (cuando cambia de 1 a 0).



Tipos de biestables

- **Biastable S-R (latch S-R):** El circuito tiene dos entradas, S (set) y R (reset), y dos salidas, Q y \bar{Q} , y consiste en dos puertas NOR conectadas por realimentación. Este tipo de circuitos puede funcionar como una memoria de 1 bit, las entradas S y R sirven para escribir los valores, 1 y 0 en la memoria. El cerrojo S-R es asíncrono porque, si las entradas no cambian, tampoco cambian las salidas.

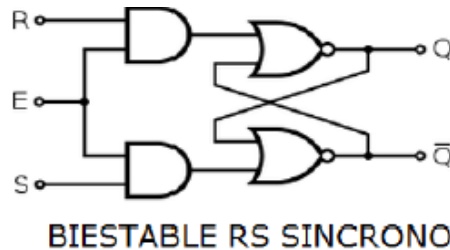


BIESTABLE RS ASINCRONO

S	R	Q	\bar{Q}
0	0	Q	\bar{Q}
0	1	0	1
1	0	1	0
1	1	x	x

Las combinaciones $S=R=1$ no son válidas, porque producirán una salida inconsistente ($Q=NQ=0$).

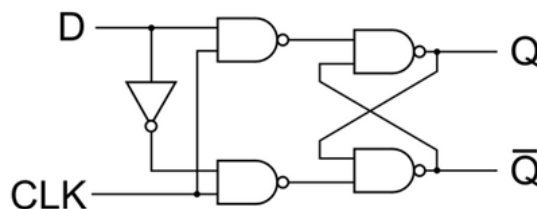
- **Biestable S-R Síncrono:** Además de las dos entradas R y S, posee una entrada más, denominada CLK, que consiste en un reloj. La diferencia con el S-R es que los cambios sólo ocurren en un pulso de reloj (entrada de 1 bit que va cambiando de 0 a 1 en tiempos regulares, constantemente).



CK	S	R	Q	NQ
1	0	0	Q	NQ
1	0	1	0	1
1	1	0	1	0
1	1	1	X	X
0	X	X	Q	NQ

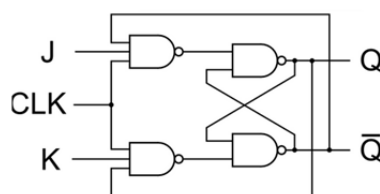
- **Biestable D:** Es un circuito de una sola entrada (D), la cual copia su valor a Q. Tiene la ventaja de que no hay error cuando las entradas son 1.

Como la salida del biestable D es siempre igual al valor más reciente aplicado a la entrada, recuerda y produce la última entrada. Por lo tanto, sirve para guardar un bit. También se le llama “biestable de retardo” porque retrasa un 0 o un 1 aplicado a la entrada durante un pulso de reloj.



CK	D	Q	NQ
1	0	0	1
1	1	1	0
0	x	Q	NQ

- **Biestable J-K:** En base al biestable S-R, tiene dos entradas, pero en este caso todas las combinaciones posibles de los valores de entradas son válidos. Cuando $J=K=1$, la salida se invierte (función de conmutación).



CK	J	K	Q	NQ
1	0	0	Q	NQ
1	0	1	0	1
1	1	0	1	0
1	1	1	NQ	Q
0	X	X	Q	NQ

- **Biestable T:** Se construye a partir de un J-K y siempre se niegan las salidas. Las dos entradas siempre valen 1 y lo que genera cambios en la salida es el reloj, por lo tanto, se considera un temporizador.



T	Q	NQ
1	NQ	Q
x	Q	NQ

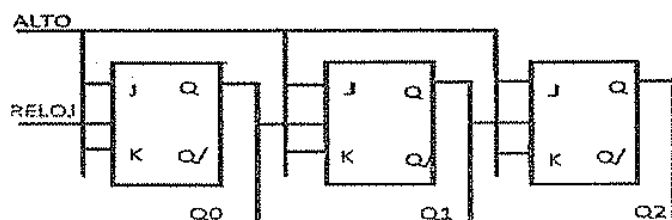
Registros

1. **Registros paralelos:** consisten en un conjunto de memorias de 1 bit que se pueden leer o escribir simultáneamente. Para su funcionamiento, se usan biestables S-R. Una señal de control, llamada *validación de dato de entrada*, controla la escritura en los registros de los valores provenientes de las líneas de señales. Estas líneas pueden ser salidas de multiplexores, ya que los datos que se pueden cargar en un registro pueden provenir de una gran variedad de fuentes. La salida se controla de una forma similar. Como una característica extra, hay disponible una línea de puesta a cero (reset), que permite poner a 0 fácilmente el registro.
2. **Registros de desplazamiento:** un registro de este tipo acepta y/o transfiere información vía serie. Se pueden usar como interfaz de dispositivos serie de E/S. Además, pueden usarse en la ALU para realizar desplazamientos lógicos y funciones de rotación. En este ultimo uso, necesitan equiparse con circuitos de lectura/escritura, tanto paralela como seria.

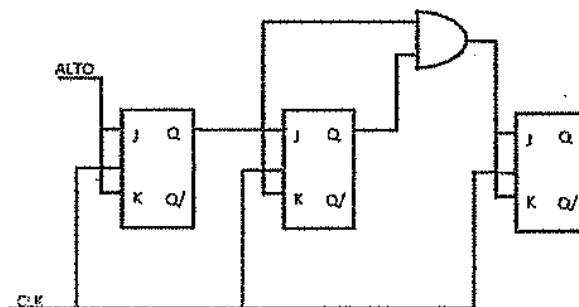
Contadores

Un contador es un registro cuyo valor se puede incrementar en 1 modulo la capacidad de ese registro. Así, un circuito hecho con n biestables puede contar hasta $2^n - 1$. Cuando el contador se incrementa más allá de su valor máximo, se pone a 0. Un ejemplo es el contador de programa.

Los contadores pueden ser síncronos o asíncronos. Los contadores asíncronos son relativamente lentos, ya que la salida de un biestable produce un cambio en el estado del siguiente biestable.



En un contador síncrono, todos los biestables cambian de estado a la vez. Como el último tipo es mucho más rápido, es el tipo que se usa en la CPU.



Aplicaciones de lógica digital en la Unidad Aritmético-Lógica (ALU) y en la Unidad de control.

Lógica en la Unidad de Control (UC)

La lógica interna de la UC produce señales de control de salida a partir de las señales de entrada. La UC debe controlar el estado del ciclo de instrucción. Al final de cada subciclo, la UC emite una señal que hace que el generador de temporización se reinicie y emita el primer tiempo. Además, la UC debe establecer los valores adecuados de las señales de control para definir el siguiente subciclo a ejecutar.

Unidad Aritmético-Lógica (ALU)

La ALU se compone básicamente de: circuito operacional, registros de entradas, un registro acumulador y un registro de estado, conjunto de registros que hacen posible la realización de cada una de las operaciones.

La mayoría de las acciones de la computadora son realizadas por la ALU. La ALU toma datos de los registros del procesador. Estos datos son procesados y los resultados de esta operación se almacenan en los registros de salida de la ALU. Otros mecanismos mueven datos entre estos registros y la memoria.

UNIDAD 4: Unidad Central de Procesamiento (CPU)

Organización de la computadora

La CPU es el “cerebro” de la computadora. Su función es ejecutar programas almacenados en la memoria principal buscando instrucciones y examinándolas para después ejecutarlas unas tras otra. Sus componentes están conectados por un bus, que es una colección de alambres paralelos que transmiten direcciones, datos y señales de control. Los buses pueden ser externos a la CPU, cuando la conectan a la memoria y a los dispositivos de E/S, pero también internos. Los buses internos de la CPU se pueden clasificar 3 tipos con diferentes funciones: el bus de datos, que por este circulan los datos, el bus de control, que por este circulan las señales de control y por el que circulan las direcciones de memoria es el bus de direcciones. Cuando se hable de un microprocesador de 32 bits, se está diciendo que el número de líneas del bus interno es de 32

La CPU se compone de varias partes. La Unidad de Control (UC) se encarga de buscar instrucciones de la memoria principal y determinar su tipo. La Unidad Aritmético-Lógica (ALU) realiza operaciones como suma y AND booleano necesarias para ejecutar instrucciones.

La CPU también contiene una memoria pequeña y de alta velocidad que sirve para almacenar resultados temporales y cierta información de control. Esta memoria se compone de varios registros, cada uno de los cuales tiene cierto tamaño y función.

El registro más importante es el contador de programa (PC), que apunta a la siguiente instrucción que debe buscarse para ejecutarse. Otro registro importante es el registro de instrucciones (IR), que contiene la instrucción que se está ejecutando.

Casi todas las instrucciones pueden dividirse en una de dos categorías: **registro-memoria** o **registro-registro**. Las instrucciones registro-memoria permiten buscar palabras de la memoria a los registros, donde pueden utilizarse como entradas de la ALU en instrucciones subsecuentes. Una instrucción registro-registro típica busca dos operandos de los registros, los coloca en los registros de entrada de la ALU, realiza alguna operación con ellos y coloca el resultado en uno de los registros. El proceso de hacer pasar dos operandos por la ALU y almacenar el resultado se llama *ciclo del camino de datos* y es el corazón de casi todas las CPU.

Características de las instrucciones máquina

El funcionamiento de la CPU está determinado por las instrucciones que ejecuta. Estas instrucciones se denominan instrucciones máquina.

Elementos de una instrucción máquina

- **Código de operación:** especifica la operación a realizar. La operación se indica mediante un código binario, denominado “codop”.
- **Referencia a operandos fuente:** la operación puede implicar uno o más operandos fuente, es decir, operandos que son entradas para la instrucción.
- **Referencia al operando resultado:** la operación puede producir un resultado.
- **Referencia a la siguiente instrucción:** dice a la CPU de donde captar la siguiente instrucción tras completarse la ejecución de la instrucción actual.

La siguiente instrucción a captar está en memoria principal o secundaria. Los operandos fuente y resultado pueden estar en alguna de las siguientes áreas:

- **Memoria principal o virtual:** debe indicarse su dirección.
- **Registro de la CPU:** si es un solo registro, la referencia es implícita, si es más de uno, debe indicarse el número de registro deseado en la instrucción.
- **Dispositivo de E/S:** la instrucción debe especificar el módulo y el dispositivo para la operación.

Tipo de Instrucciones

El repertorio de instrucciones debe ser suficientemente amplio como para expresar cualquiera de las instrucciones de alto nivel. Los tipos de instrucciones se clasifican de la siguiente manera:

- **De Procesamiento de datos:** instrucciones aritméticas y lógicas.
- **De Almacenamiento de datos:** instrucciones de memoria.
- **De transferencia de datos:** instrucciones de E/S.
- **De control:** instrucciones de comprobación y bifurcación.

Las instrucciones aritméticas proporcionan capacidad computacional para procesar datos numéricos. Las instrucciones lógicas operan sobre los bits de una palabra, el lugar de considerarlos como números, proporcionando capacidad para el procesamiento de cualquier otro tipo de datos que el usuario deba emplear. Las instrucciones de E/S se necesitan para transferir programas y datos a la memoria, y devolver los resultados de los cálculos al usuario. Las instrucciones de comprobación o test se emplean para comprobar el valor de una palabra de datos o el estado de un cálculo. Las de bifurcación se usan entonces para bifurcar a diferentes conjuntos de instrucciones dependiendo de la decisión tomada.

Tamaño de las instrucciones

- **Instrucciones de tamaño fijo:** todas las instrucciones ocuparan el mismo numero de bits. Esta alternativa simplifica el diseño del procesador y la ejecución de las instrucciones puede ser más rápida.
- **Instrucciones de tamaño variable:** el tamaño de las instrucciones dependerá del numero de bits necesario para cada una.

Diseño del repertorio de instrucciones

El repertorio es el medio que tiene el programador para controlar la CPU, por lo tanto, deben considerarse las necesidades de éste a la hora de diseñarlos. Los aspectos más importantes de diseño son:

- **Repertorio de operaciones:** cuántas y qué operaciones considerar y de qué complejidad.
- **Tipos de datos:** los distintos tipos de datos con los que se efectúan las operaciones.
- **Formatos de instrucciones:** longitud de la instrucción, número de direcciones, etc.
- **Registros:** numero de registros de la CPU que pueden ser referenciados y uso.
- **Direccionamiento:** el modo mediante el que puede especificarse las direcciones de un operando.

Tipos de operandos

Las instrucciones máquina operan con datos. Las categorías más importantes de datos son:

- **Direcciones:** ver direccionamiento. Son enteros sin signo.
- **Números:** enteros o en coma fija, en coma flotante y en decimal.
- **Caracteres:** ASCII, EBCDIC.
- **Datos lógicos:** representación orientada a bits.

Tipos de operaciones

- Transferencia de datos
- Aritméticas.
- Lógicas.
- Conversión.
- Entrada-Salida.
- Control del sistema.
- Control de flujo.

Lenguaje de máquina y ensamblador

Lenguaje de máquina

Los circuitos electrónicos de una computadora pueden reconocer y ejecutar directamente un conjunto limitado de instrucciones sencillas, y todos los programas tienen que convertirse en una serie de esas instrucciones para que la computadora puede ejecutarlos. Dicho grupo de instrucciones primitivas que una computadora puede ejecutar constituyen su *lenguaje máquina* y permiten a las personas comunicarse con la computadora.

Lenguaje de Ensamble

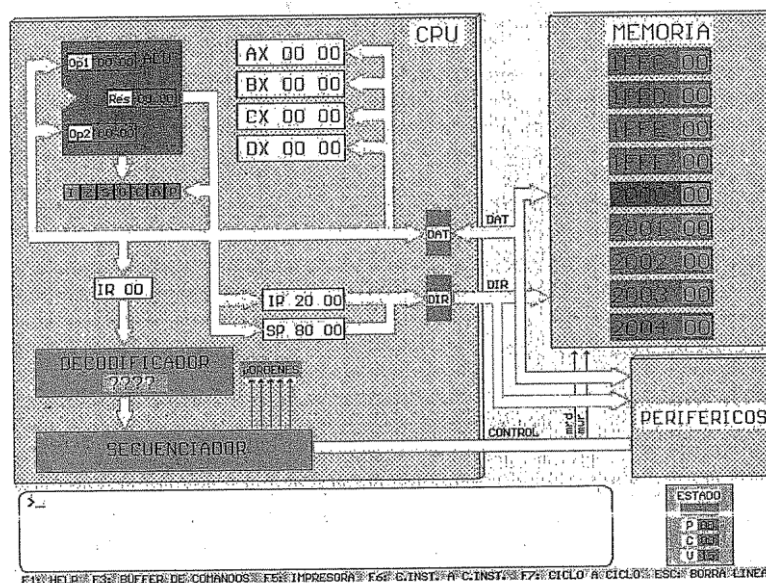
Un lenguaje ensamblador puro es una representación simbólica de un programa en lenguaje de máquina subyacente, en el que cada enunciado produce exactamente una instrucción de máquina, es decir, existe una relación uno a uno entre las instrucciones de máquina y los enunciados del programa en lenguaje assembly.

Ventajas del Leguaje ensamblador

- Puede producir código mucho más pequeño y rápido que el se puede producir en un lenguaje de alto nivel.
- Buen desempeño.
- Acceso total a la maquina/hardware.
- Útil para generar los programas de computadores portátiles de escasos recursos, como las tarjetas inteligentes, procesadores incorporados a aparatos domésticos y asistentes digitales portátiles inalámbricos.

El Simulador

El programa a utilizar se llama MSX88 y simula una computadora basada en una versión simplificada del célebre procesador 8086 de Intel, denominado SX88.



En la figura podemos distinguir los siguientes bloques:

- CPU
 - ALU
 - Registros AX, BX, CX, DX, SP, IP, IR.
 - Registros de Flags.
 - Decodificador.
 - Secuenciador.
- Memoria principal.
- Periféricos
- Bus de datos y de direcciones.
- Toma de entrada de comandos.

La CPU es la encargada de ejecutar un programa contenido en la memoria instrucción por instrucción. La ALU es la encargada de ejecutar las operaciones aritméticas y lógicas entre los registros temporarios Op1 y Op2, dejando el resultado en Res. Dichas operaciones serán las dispuestas por la instrucción en ejecución. Los registros AX, BX, CX, y DX son de uso general, 16 bits de longitud, se pueden dividir en 2 partes de 8 bits cada uno. Ejemplo: AX en AH y AL.

El registro IP contiene la dirección de memoria de la próxima instrucción a ser ejecutada. El registro SP contiene la dirección de memoria del tope de la pila. El registro de flags nos mostrará el estado de las banderas o flags luego de cada operación. Son de 8 bits que indican el estado de las correspondientes 8 banderas. De estas 8 se utilizan:

- **Bandera de cero:** identificada por la letra Z.
- **Bandera de overflow:** identificada por la letra O.
- **Bandera de carry/borrow:** identificada por la letra C.
- **Bandera de signo del número:** identificada por la letra S.

Organización de los registros

Dentro de la CPU hay un conjunto de registros que funcionan como un nivel de memoria, por encima de la memoria principal y de la cache en la jerarquía. Los registros de la CPU son de dos tipos:

1. **Registros visibles para el usuario:** permiten minimizar las referencias a memoria principal cuando se optimiza el uso de los registros.
2. **Registros de control y de estado:** son utilizado por la UC para controlar el funcionamiento de la CPU y por programas privilegiados del sistema operativo para controlar la ejecución de programas.

Registros visibles para el usuario

Se clasifican de la siguiente manera:

- **Uso general:** pueden ser asignados por el programador a diversas funciones.
- **Datos:** se usan únicamente para contener datos y no se pueden emplear en el cálculo de una dirección de operando.
- **Direcciones:** pueden ser de uso más o menos general, o pueden estar dedicados a un modo de direccionamiento particular.
- **Códigos de condición (flags):** son bits fijados por el hardware de la CPU como resultado de una operación.

Registros de control y de estados

La mayoría no son visibles para el usuario. Son esenciales cuatro registros para la ejecución de una instrucción:

- **Contador de Programa (PC):** contiene la dirección de la instrucción a captar.
- **Registro de Instrucción (IR):** contiene la instrucción captada más recientemente.
- **Registro de dirección de memoria (MAR):** contiene la dirección de una posición de memoria.

- **Registro intermedio de memoria (MBR):** contiene la palabra de datos a escribir en memoria, o la palabra leída más recientemente.

Direccionamiento

Como los campos de direcciones en un formato de instrucciones usual están bastante limitados, es deseable poder referenciar un rango elevado de posiciones de memoria principal o virtual. Para esto se emplean las técnicas de direccionamiento. para explicarlas, usamos la siguiente notación:

- **A=** contenido de un campo de dirección en la instrucción.
- **R=** contenido de un campo de dirección en la instrucción que referencia a un registro.
- **EA=** dirección real (efectiva) de la posición que contiene el operando que se referencia.
- **(X)=** contenido de la posición X.

Direccionamiento inmediato

El operando está presenta en la propia instrucción:

$$\text{OPERANDO} = A$$

Este modo puede usarse para definir y utilizar constantes, o para fijar valores iniciales de variables. La ventaja de este modo es que, una vez captada la instrucción, no se requiere una referencia a memoria para obtener el operando, ahorrándose así un ciclo de memoria o de caché. La desventaja es que el tamaño del número está restringido a la longitud del campo de direcciones que, en su mayoría, es pequeño comparado con la longitud de la palabra. Ejemplo: MOV AX, 6H.

Direccionamiento directo

El campo de direcciones contiene la dirección efectiva del operando:

$$EA = A$$

Solo requiere una referencia a memoria y no necesita ningún calculo especial. La limitación obvia es que proporciona un espacio de direcciones restringido. Ejemplo:

```
ORG 1000h
NUM1 db 10h
NUM2 db 20h
ORG 2000h
MOV cl, NUM1
MOV al, NUM2
ADD AL, CL
HLT
END.
```

Direccionamiento indirecto

El campo de direcciones referencia a la dirección de una palabra de memoria que contiene la dirección completa del operando:

$$EA = (A)$$

La ventaja es que para una longitud de la palabra de N bits, se dispone de un espacio de direcciones de 2^N . La desventaja es que la ejecución de la instrucción requiere dos referencias a memoria para captar el operando: una para captar su dirección y otra para obtener su valor. Ejemplo:

```
DIR1 db 1001h
...
MOV AX, [DIR1]
HLT
END.
```

Direccionamiento de registros

Es similar al directo, aunque ahora el campo de direcciones referencia a un registro:

$$EA = R$$

Puede referenciarse un total de 8 o 16 registros de uso general. Las ventajas de este modo son que sólo es necesario un campo pequeño de direcciones en la instrucción y no se requieren referencias a memoria. La desventaja es que el espacio de direcciones está muy limitado.

Direccionamiento indirecto de registros

Es igual al indirecto, aunque el campo de direcciones hace referencia a un registro:

$$EA = (R)$$

Las ventajas y limitaciones de este modo son básicamente las mismas que se tienen en el indirecto. La limitación del espacio se supera haciendo que el campo de direcciones referencie a una posición de una palabra completa (un registro completo) que contenga la dirección. Además, este modo emplea una referencia menos a memoria que el direccionamiento indirecto. Ejemplo:

```
MOV AX, [BX]
```

Direccionamiento con desplazamiento

Combina las posibilidades de los direccionamientos directo e indirecto con registro

$$EA = A + (R)$$

Requiere que las instrucciones tengan dos campos de direcciones, al menos uno de ellos explícito. El valor contenido en uno de los campos de direcciones (valor = A) se utiliza

directamente, el otro campo de direcciones, o una referencia implícita definida por el codop, se refiere a un registro cuyo contenido se suma a cl. para generar la dirección efectiva.

Tres de los usos más comunes del direccionamiento con desplazamiento son:

- **Direccionamiento relativo:** el registro direccionado implícitamente es el PC, es decir, la dirección de instrucción actual se suma al campo de direcciones para producir el valor EA. La dirección efectiva es un desplazamiento relativo a la dirección de la instrucción. Este tipo de desplazamiento aprovecha el concepto de localidad.
- **Direccionamiento con registro base:** el registro referenciado contiene una dirección de memoria, y el campo de dirección contiene un desplazamiento desde dicha dirección. La referencia a registro puede ser implícita o explícita. Este direccionamiento también aprovecha la localidad de las referencias a memoria.
- **Indexado:** el campo de dirección referencia una dirección de memoria principal, y el registro referenciado contiene un desplazamiento positivo desde esa dirección. Un uso importante del indexado es como mecanismo eficiente para ejecutar operaciones iterativas.

Direccionamiento de pila

El valor del puntero tiene la dirección del tope de pila. Alternativamente, los dos elementos del tope pueden residir en registros de la CPU, en cuyo caso el puntero de pila hace referencia al tercer elemento de la pila. El puntero de pila se mantiene en un registro. Así, las referencias a posiciones de la pila en memoria son, de hecho, direcciones de acceso indirecto con registro. El modo de direccionamiento de pila es una forma de direccionamiento implícito. Las instrucciones máquina no necesitan incluir una referencia a memoria, sino que operan implícitamente con la cabecera de la pila. Ejemplo:

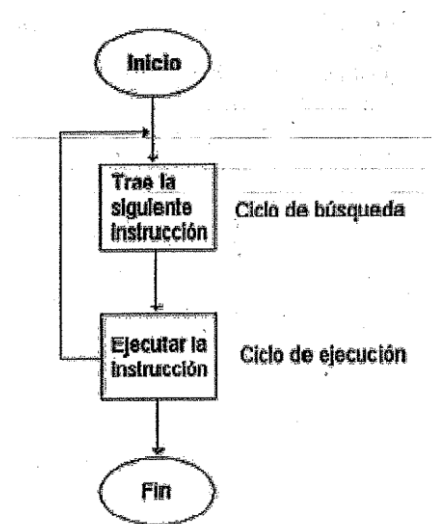
```
POP AX
PUSH AX
POPF AX
PUSHF AX
```

Ciclo de instrucción

Los ciclos de captación y ejecución

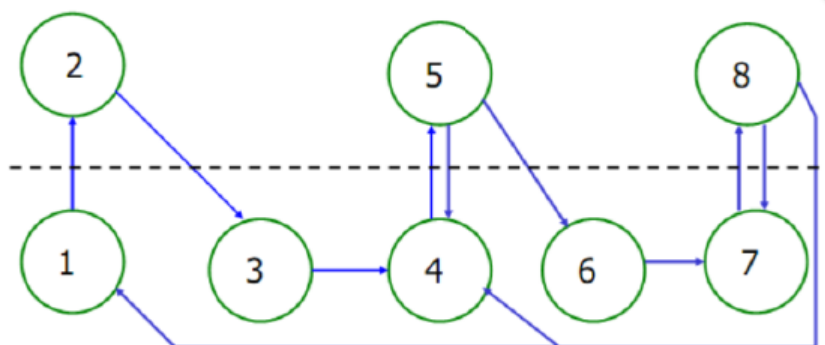
Al comienzo de cada instrucción, la CPU capta una instrucción de memoria. La instrucción captada se almacena en IR. La instrucción se escribe utilizando un código binario que especifica la acción que debe realizar la CPU. La CPU interpreta la instrucción y lleva a cabo la acción requerida. En general, puede ser de 4 tipos:

1. **Procesador-memoria:** deben transferirse datos desde la CPU a la memoria, o desde la memoria a la CPU.
2. **Procesador-E/S:** deben transferirse datos a o desde el exterior mediante transferencias entre la CPU y un módulo de E/S.
3. **Procesamiento de datos:** la CPU ha de realizar alguna operación aritmética o lógica con los datos.
4. **Control:** una instrucción puede especificar que la secuencia de ejecución se altere (por ejemplo, un salto).



Estados de ciclo de instrucción

1. **Cálculo de la dirección de la instrucción (IAC):** determina la dirección de la siguiente instrucción a ejecutar.
2. **Captación de la instrucción (IF):** la CPU lee la instrucción desde su posición en memoria.
3. **Decodificación de la operación indicada en la instrucción (IOD):** analiza la instrucción para determinar el tipo de operación a realizar y el/los operandos a utilizar.
4. **Cálculo de la dirección del operando (OAC):** si la instrucción implica una referencia a un operando en memoria o disponible mediante E/S, determina la dirección del operando.
5. **Captación del operando (OF):** capta el operando desde memoria o se lee desde el dispositivo de E/S.
6. **Operación con los datos (DO):** realiza la operación indicada en la instrucción.
7. **Calculo dirección resultado (OAC)**
8. **Almacenamiento del operando (OS):** escribe el resultado en memoria o lo saca a través de un dispositivo de E/S.



Interrupciones

Las interrupciones proporcionan una forma de mejorar la eficiencia del procesador. Existen diferentes clases de interrupciones:

- **Programa:** generadas por alguna condición que se produce como resultado de la ejecución de una instrucción, como la división por cero.
- **Temporización:** generadas por un temporizador interno al procesador. Permite al sistema operativo realizar ciertas funciones de manera regular.
- **E/S:** generadas por un controlador de E/S para indicar la finalización sin problemas de una operación o para avisar de ciertas condiciones de error.
- **Fallo de hardware:** generadas por un fallo tal como un error de paridad en la memoria.

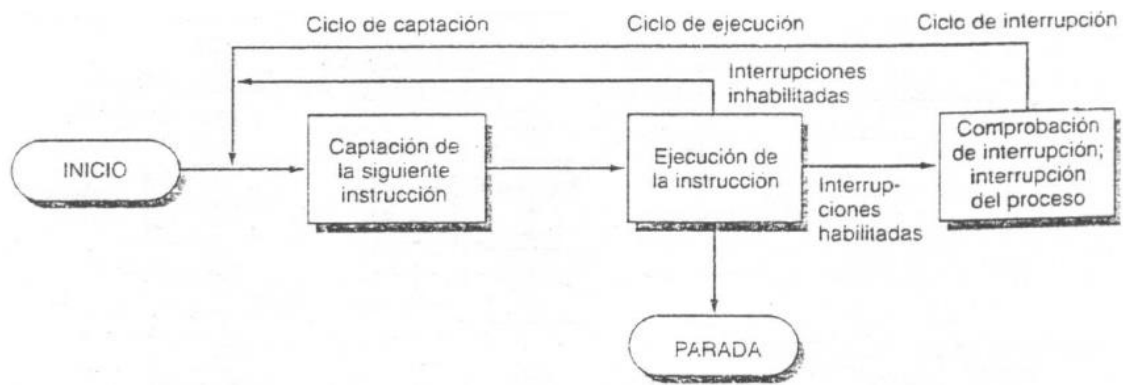
Las interrupciones y el ciclo de instrucción

Para permitir el uso de interrupciones, se añade un ciclo de interrupción al ciclo de instrucción. En el ciclo de interrupción, el procesador comprueba si ha generado una interrupción, indicada por la presencia de una señal de interrupción. Si no hay señales pendientes, el procesador continúa con el ciclo de captación y accede a la siguiente instrucción del programa en curso. Si hay una interrupción pendiente, el procesador hace lo siguiente:

1. **Suspende la ejecución** del programa en curso y **guarda su contexto**.
2. **Carga el contador de programa** con la dirección de **comienzo de una rutina de gestión de interrupción**.

Ciclo de instrucción (con interrupciones)

1. **CICLO DE CAPTACION:** las operaciones realizadas en esta fase son las siguientes:
 - A. *Leer la instrucción.*
 - B. *Decodificar la instrucción.*
 - C. *Actualizar el Contador de Programas.*
2. **LECTURA DE OPERANDOS FUENTE:** se debe repetir para todos los operandos que tenga la instrucción.
3. **CICLO DE EJECUCION:**
 - A. *Ejecución de la instrucción.*
 - B. *Almacenamiento del operando de destino (Resultado)*
4. **CICLO DE INTERRUPCIONES:** en esta fase se verifica si se ha activado alguna línea de petición de interrupción del procesador en el transcurso de la ejecución de la instrucción.



Estructuras de Interconexión

Son el conjunto de líneas que conectan los módulos de un computador. Los tipos de intercambio son:

- **Memoria:** un modulo de memoria está constituido por N palabras de la misma longitud. A cada palabra se le asigna una única dirección numérica (de 0 a N-1). Una palabra de datos puede leerse o escribirse en la memoria. La posición de memoria para la operación se especifica mediante una dirección.
- **Modulo de E/S:** la E/S es funcionalmente similar a la memoria. Hay dos tipos de operaciones: leer y escribir. Existen líneas de datos externas para la entrada y la salida de datos por un dispositivo externo. Por último, un módulo de E/S puede enviar señales de interrupción al procesador.
- **Procesador:** lee instrucciones y datos, escribe datos una vez que los ha procesado, y usa ciertas señales para controlar el funcionamiento del sistema.

La estructura de interconexión debe ser cobertura a los siguientes tipos de transferencia:

1. Memoria a procesador.
2. Procesador a memoria.
3. E/S a procesador.
4. Memoria a E/S y viceversa.

Interconexión con buses

Estructura del bus

El bus del sistema está constituido por entre 50 y 100 líneas, que se pueden clasificar en:

- **Líneas de datos:** transmiten datos entre los módulos del sistema. El conjunto se denomina bus de datos.
- **Líneas de dirección:** se usan para designar la fuente o el destino del dato situado en el bus de datos. La anchura del bus de direcciones determina la máxima capacidad de memoria posible.
- **Líneas de control:** se usan para controlar el acceso y el uso de las líneas de datos y de direcciones, ya que son compartidas.

Jerarquía de buses

Si se conecta un gran número de dispositivos al bus, las prestaciones pueden disminuir por dos causas principales:

1. A más dispositivos conectados al bus, mayor es el retardo de propagación, el cual determina el tiempo que necesitan los dispositivos para coordinarse en el uso del bus.
2. El bus puede convertirse en un cuello de botella a medida que las peticiones de transferencia acumuladas se aproximan a la capacidad del bus. Esto se resuelve aumentando el ancho del bus, aunque un único bus ya no se usa.

Elementos de diseño de un bus

Los parámetros que clasifican los buses son:

- **Tipo de bus:** se divide en dos tipos genéricos.
 - **Dedicadas:** está permanentemente asignada a una función o a un subconjunto físico de componentes de la computadora.
 - **Multiplexadas:** mismas líneas para diferentes usos.
- **Método de arbitraje:** se clasifican en
 - **Centralizado:** un árbitro es el responsable de asignar tiempos en el bus.
 - **Distribuido:** cada módulo dispone de lógica para controlar el acceso y los módulos actúan conjuntamente para compartir el bus.
- **Temporización:** hace referencia a la forma en la que se coordinan los eventos en el bus. Se clasifican en:
 - **Síncrona:** la presencia de un evento en el bus está determinada por un reloj.
 - **Asíncrona:** la presencia de un evento en el bus es consecuencia y depende de que se produzca un evento previo.
- **Anchura del bus:** cuanto más ancho es el bus de datos, mayor es el número de bits que se transmiten a la vez. Cuanto más ancho es el bus de direcciones, mayor es el rango de posiciones a las que se puede hacer referencia.
- **Tipos de transferencia de datos:** todos los buses permiten tanto transferencias de lecturas como de escrituras. En ciertos buses también son posibles algunas operaciones combinadas, como lectura-modificación-escritura o lectura-después de-escritura.

Bus PCI

El bus PCI es un bus de ancho de banda elevado, independiente del procesado, que se puede usar como bus de periféricos o bus para una arquitectura de entreplanta. Está diseñado para permitir una cierta variedad de configuraciones basadas en microprocesadores, incluyendo sistemas tanto de uno como de varios procesadores. Proporciona un conjunto de funciones de uso general. Utiliza temporización síncrona y un esquema de arbitraje centralizado.

Descripción de procesadores actuales

Las técnicas incorporadas a los procesadores de hoy en día están:

- **Predicción de ramificación:** el procesador se anticipa al software y predice qué grupo de instrucciones se van a procesar después con mayor probabilidad. Si el procesador acierta la mayoría de las veces, puede precaptar las instrucciones correctas y almacenarlas para mantener al procesador ocupado.
- **Análisis de flujo de datos:** el procesador analiza qué instrucciones dependen de los resultados de otras instrucciones o datos, para crear una organización optimizada de instrucciones. Las instrucciones se regulan para ser ejecutadas cuando estén listas, independientemente del orden original del programa. Esto evita retrasos innecesarios.
- **Ejecución especulativa:** utilizando la predicción de ramificación y el análisis de flujo de datos, algunos procesadores ejecutan especulativamente instrucciones antes de que aparezcan en la ejecución del programa, manteniendo los resultados en posiciones temporales. Esto permite al procesador mantener sus máquinas de ejecución lo más ocupadas posible, ejecutando instrucciones que es probable que se necesiten.

Principios de diseño para computadoras modernas

1) Todas las instrucciones se ejecutan en hardware, éstas no se interpretan con microinstrucciones. La eliminación de un nivel de interpretación hace que la mayor parte de las instrucciones sean más rápidas.

2) Maximizar el ritmo con que se emiten instrucciones. Este principio sugiere que el paralelismo puede desempeñar un papel importante en el mejoramiento del desempeño.

3) Las instrucciones deben ser fáciles de decodificar. Todo lo que pueda agilizar el proceso de decodificación es útil, como por ejemplo hacer que las instrucciones tengan una longitud fija, con un número pequeño de campos. Cuando menor sea el número de formatos de instrucciones distintos, mejor.

4) Sólo las operaciones de carga y almacenamiento deben hacer referencias a memoria. Ya que el acceso a memoria puede tardar mucho, la mejor manera de evitar retrasos es que solo las instrucciones LOAD y STORE hagan referencia a memoria.

5) Incluir abundantes registros. Puesto que el acceso a memoria es bastante lento, es necesario contar con muchos registros (32 al menos) para que, una vez que se ha obtenido una palabra, se pueda mantener en un registro hasta que ya no se necesite. Quedarse sin registros y tener que escribirlos en la memoria solo para volver a cargarlos después es poco recomendable y debe evitarse preferentemente. La mejor manera de lograr esto es tener suficientes registros.

Procesador Pentium

Organización de los registros

Incluye los siguientes registros:

- **Generales:** ocho registros de 32 bits, algunos de uso especial.
- **De Segmento:** seis de 16 bits. Indexan tablas de segmento. El registro de segmento de código (CS) referencia el segmento que contiene la instrucción que se está ejecutando. El registro de segmento de pila (SS) referencia el segmento que contiene una pila visible para el usuario. Los demás registros (DI, SI, BP y SP) permiten al usuario referenciar, al mismo tiempo, hasta cuatro segmentos de datos distintos.
- **Indicadores:** el registro EFLAGS contiene los códigos de condición y diversos bits de modo.
- **Puntero de instrucción:** contiene la dirección de la instrucción en curso.

Hay también registros dedicados específicamente a la unidad de coma flotante:

- **Númérico:** cada registro contiene un número en coma flotante de 80 bits de precisión ampliada y 8 que funcionan como una pila.
- **De control:** tiene 16 bits que controlan el funcionamiento de la unidad de coma flotante, precisión simple, doble o ampliada, y bits para habilitar o inhabilitar diversas condiciones de excepción.
- **De estado:** de 16 bits contiene bits que reflejan el estado presente de la unidad de coma flotante, códigos de condición que informan sobre el resultado de la última operación, e indicadores de excepción.
- **Palabra de etiquetas:** de 16 bits contiene una etiqueta de 2 bits para cada registro numérico de coma flotante, que indica la naturaleza de los contenidos del registro correspondiente. Los cuatro valores posibles son válidos, cero, especial y vacío.

Ejemplos de procesadores modernos

- **Pentium**
- **Pentium II**
- **Intel Celeron**
- **Intel Xeon**
- **Pentium D**
- **Intel Core 2 Duo**

Procesador PowerPC

Organización de los registros

La unidad de coma fija incluye los siguientes:

- **Generales:** hay 32 registros de 64 bits de uso general. Se pueden utilizar para cargar, almacenar y manipular operandos de datos, y también se pueden usar para direccionamiento indirecto a través de registros.
- **Registro de excepción (XER):** incluye 3 bits que informan sobre excepciones en operaciones aritméticas con enteros.

La unidad de coma flotante contiene otros registros visibles para el usuario:

- **Generales:** hay 32 registros de uso general de 64 bits, usados para todas las operaciones de coma flotante.
- **Registro de estado y control de coma flotante (FPSCR):** de 32 bits contiene bits que controlan el funcionamiento de la unidad de coma flotante, y bits que guardan el estado resultante de operaciones de coma flotante.

La unidad de procesamiento de saltos contiene estos registros visibles para el usuario:

- **Registro de condición:** consta de 8 campos de código de condición de 4 bits.
- **Registro de enlace:** puede usarse en una instrucción de bifurcación condicional para el direccionamiento indirecto de la dirección destino.
- **Cuenta:** puede usarse para controlar iteraciones de bucles. Se decrementa cada vez que es examinado por una instrucción de bifurcación condicional.

UNIDAD 5: Memoria

Conceptos básicos sobre sistemas de memoria de computadoras.

Características de los sistemas de memoria.

- **Ubicación:** Puede ser interna o externa. La memoria interna suele identificarse con la memoria principal. Sin embargo, el procesador necesita su propia memoria local en forma de registros y la CU del procesador también usa su propia memoria interna. La memoria externa consta de dispositivos periféricos de almacenamiento tales como discos y cintas que son accesibles por la CPU a través de los controladores de E/S.
- **Capacidad:** Hace referencia a la cantidad de información que se puede almacenar. La unidad utilizada para especificar la capacidad de almacenamiento de información es el byte (1 byte = 8 bits) o palabra (8, 16 y 32 bits), y a la hora de indicar la capacidad, se utilizan diferentes prefijos que representan múltiplos del byte. En informática, la capacidad de almacenamiento habitualmente se indica en múltiplos que sean potencias de dos; en este caso se utilizan los prefijos definidos por la International Electrotechnical Commission (IEC).

- 2^{10} byte = 1024 bytes = 1 KIB (kibibyte)

- 2^{20} byte = 1024 KIB = 1 MIB (mebibyte)
- 2^{30} byte = 1024 MiB = 1 GIB (gibibyte)
- 2^{40} byte = 1024 GIB = 1 TIB (tebibyte)

- **Método de acceso**

1. *Secuencial*: Se accede desde la última posición a la que se ha accedido, leyendo en orden todas las posiciones de memoria hasta llegar a la posición deseada. El tiempo de acceso depende de la posición a la que se quiere acceder y de la posición a la que se ha accedido anteriormente.
2. *Directo*: La memoria se organiza en bloques y cada bloque de memoria tiene una dirección única, se accede directamente al principio de un bloque y dentro de este se hace un acceso secuencial hasta llegar a la posición de memoria deseada. El tiempo de acceso depende de la posición a la que se quiere acceder y de la última posición a la que se ha accedido.
3. *Aleatorio*: Cada posición direccionable de memoria tiene un único mecanismo de acceso, cableado físicamente. El tiempo para acceder a una posición dada es constante e independiente de la secuencia de accesos previos. Por tanto, cualquier posición puede seleccionarse aleatoriamente y puede ser direccionada y accedida directamente.
4. *Asociativa*: Se trata de un tipo de memoria de acceso aleatorio donde el acceso se hace basándose en el contenido y no en la dirección. Se especifica el valor que se quiere localizar y se compara este valor con una parte del contenido de cada posición de memoria; la comparación se lleva a cabo simultáneamente con todas las posiciones de la memoria.

- **Unidad de transferencia**: Es el número de bits que se leen o escriben en memoria a la vez. Para memorias internas es igual al número de líneas de E/S de datos del módulo de memoria. Para memorias externas, los datos se transfieren en bloques.

1) **Palabra de memoria**: Es la unidad de organización de la memoria desde el punto de vista del procesador; el tamaño de la palabra de memoria se especifica en bytes o bits y suele coincidir con el número de bits utilizados para representar números y con la longitud de las instrucciones. La palabra de memoria es el número de bytes máximo que se pueden leer o escribir en un solo ciclo de acceso a la memoria.

2) **Unidad de direccionamiento**: La memoria interna se puede ver como un vector de elementos, una colección de datos contiguos, en la que cada dato es accesible indicando su posición o dirección dentro del vector. La unidad de direccionamiento especifica cuál es el tamaño de cada elemento; habitualmente a la memoria se accede como un vector de bytes – cada byte tendrá su dirección–, aunque puede haber sistemas que accedan a la memoria como un vector de palabras, en los que cada dirección corresponda a una palabra. El número de bits utilizados para especificar una dirección de memoria fija el límite máximo de elementos dirigibles, el tamaño del mapa de memoria; si tenemos n bits para las direcciones de memoria, el número máximo de elementos dirigibles será de 2^n .

3) **Unidad de transferencia:** En un acceso a memoria se puede acceder a un byte o a varios, con un máximo que vendrá determinado por el número de bytes de una palabra de memoria; es decir, en un solo acceso se leen o escriben uno o varios bytes. Cuando se especifica la dirección de memoria a la que se quiere acceder, se accede a partir de esta dirección a tantos bytes como indique la operación de lectura o escritura. En memoria externa, se accede habitualmente a un bloque de datos de tamaño muy superior a una palabra. En discos es habitual transferir bloques del orden de los Kbyte.

- **Prestaciones**

Se utilizan tres parámetros de medida de prestaciones:

- **Tiempo de acceso:** para memorias de acceso aleatorio es el tiempo que tarda en realizarse una operación de lectura o escritura. Para otras memorias, es el tiempo de acceso en que se tarda en situar el mecanismo de lectura/escritura en la posición deseada.
- **Tiempo de ciclo de memoria:** se aplica a las memorias de acceso aleatorio y consiste en el tiempo de acceso y algún tiempo más que se requiere, antes de que pueda iniciarse un segundo acceso a memoria.
- **Velocidad de transferencia:** es la velocidad a la que pueden transferirse los datos a, o desde, una unidad de memoria

- **Características físicas**

La memoria se puede clasificar según **características físicas** diferentes; básicamente podemos distinguir dos clasificaciones. La primera distingue entre:

- **Memoria Volátil:** memoria que necesita una corriente eléctrica para mantener su estado; estas memorias incluyen registros, memoria cache y memoria principal.
- **Memoria no Volátil:** mantiene el estado sin necesidad de corriente eléctrica, incluye memorias de solo lectura, memorias programables, memoria flash, dispositivos de almacenamiento magnético y óptico.

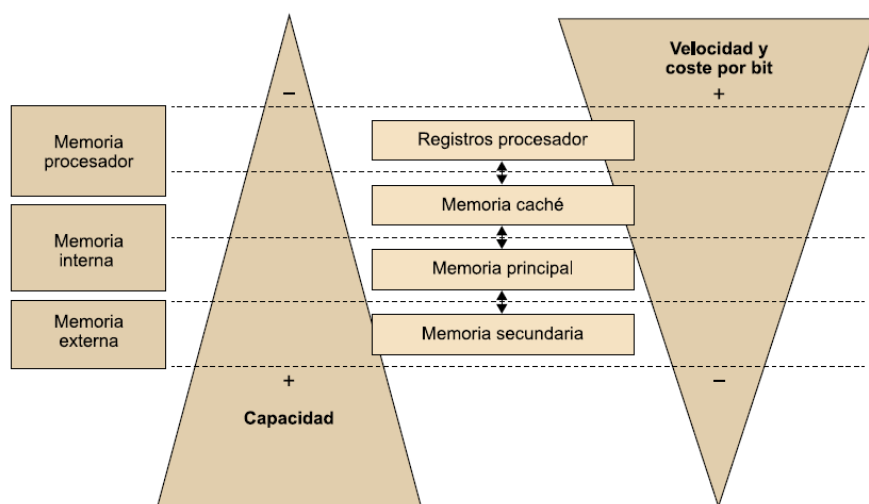
La segunda clasificación distingue entre:

- **Memoria de semiconductores:** es una memoria que utiliza elementos semiconductores, transistores, en su construcción; incluye: registros, memoria cache, memoria principal, memorias de solo lectura, memoria flash.
- **Memoria magnética:** utiliza superficies imantadas para guardar la información; dentro de esta categoría se incluyen básicamente discos y cintas magnéticas.
- **Memoria óptica:** utiliza elementos de almacenamiento que pueden ser leídos y escritos mediante luz láser; se incluyen dispositivos de CD, DVD, Blu-ray.

Jerarquía de memoria

El objetivo en el diseño del sistema de memoria, es obtener gran capacidad y un tiempo de acceso reducido con el precio más bajo posible. Como no existe ninguna tecnología que cumple simultáneamente estos requisitos, la memoria de una computadora se estructura en varios niveles con el objetivo de conseguir mejores prestaciones, y forma lo que se denomina Jerarquía de Memoria.

En una **Jerarquía de Memoria** se utilizan varios tipos de memorias, con distintas características de capacidad, velocidad y coste, divididas en diferentes niveles (Memoria del procesador, memoria interna y memoria externa). Cada nivel de la Jerarquía se caracteriza también por la distancia a la que se encuentra del procesador. Los primeros niveles son los más próximos y esto es así porque también son los niveles con velocidad más elevada.



Existe un compromiso entre las 3 características claves (costo, capacidad y tiempo de acceso):

1. A menor tiempo de acceso, mayor coste por bit.
2. A mayor capacidad, menor coste por bit.
3. A mayor capacidad, mayor tiempo de acceso.

Cuando se desciende en la jerarquía, ocurre que:

1. Disminuye el coste por bit.
2. Aumenta la capacidad.
3. Aumenta el tiempo de acceso.
4. Disminuye la frecuencia de acceso a la memoria por parte del procesador.

El objetivo final de la jerarquía de memorias es conseguir que, cuando el procesador acceda a un dato, este se encuentre en el nivel más rápido de la jerarquía. Obtenemos así, una memoria a un coste moderado, con una velocidad próxima a la del nivel más rápido y la capacidad del nivel más alto.

Localidades de referencia

La localidad de las referencias, también conocida como el principio de localidad, es un fenómeno según el cual, basándonos en el pasado reciente de un programa podemos predecir con una precisión razonable qué instrucciones y datos utilizará en un futuro próximo.

Los casos más importantes de localidad son la localidad espacial, la localidad secuencial y la localidad temporal.

Localidad temporal

Si en un momento una posición de memoria particular es referenciada, entonces es muy probable que la misma ubicación vuelva a ser referenciada en un futuro cercano. En este caso, es común almacenar una copia de los datos referenciados en caché para lograr un acceso más rápido a ellos.

Localidad espacial

Si una localización de memoria es referenciada en un momento concreto, es probable que las localizaciones cercanas a ella sean también referenciadas pronto. En este caso, es común estimar las posiciones cercanas para que estas tengan un acceso más rápido.

Localidad secuencial

Las direcciones de memoria que se están utilizando suelen ser contiguas. Esto ocurre porque las instrucciones se ejecutan secuencialmente. Para obtener beneficios de la gran frecuencia con la que ocurren casos de localidad espacial o temporal, muchos sistemas de memoria utilizan una jerarquía de niveles de memoria.

Organización de la memoria en una computadora

La memoria en una computadora moderna está formada típicamente por 4 niveles fundamentales:

- **Registros**

El **registro** es el espacio de memoria que se encuentra dentro del procesador, integrado dentro del mismo chip de este. Se utilizan celdas de memoria de tipo estático, SRAM, para su implementación. Es el espacio de memoria en el cual el procesador puede acceder más rápidamente a los datos. Este espacio de memoria es accesible al programador de lenguaje de ensamblador y, si se gestiona bien, permite minimizar el número de accesos a la memoria interna, que son bastante más lentos.

- **Memoria principal (Memoria interna)**

En la **memoria principal** se almacenan los programas que se deben ejecutar y sus datos, es la memoria visible para el programador mediante su espacio de direcciones. La memoria principal se implementa utilizando diferentes chips conectados a la placa principal del computador y tiene una capacidad mucho más elevada que la memoria caché (del orden de Gbytes o de Tbytes en supercomputadores). Utiliza tecnología DRAM (Dynamic RAM), que es

más lenta que la SRAM, pero con una capacidad de integración mucho más elevada, hecho que permite obtener más capacidad en menos espacio.

- **Memoria Secundaria (Memoria Externa)**

La **memoria secundaria** corresponde a dispositivos de almacenamiento secundario: discos magnéticos, cintas magnéticas, discos ópticos, dispositivos de memoria flash, etc., y también se pueden considerar sistemas de almacenamiento en red. Estos dispositivos son gestionados por el sistema de ficheros del sistema operativo mediante el sistema de entrada/salida. Los dispositivos que forman la memoria externa se conectan al computador con algún tipo de bus (serie o paralelo). Estos dispositivos se pueden encontrar físicamente dentro del computador conectados por buses internos del computador (IDE, SATA, SCSI, etc.) o pueden estar fuera del computador conectados por buses externos (USB, Firewire, eSATA, Infiniband, etc.).

- **Memoria Virtual**

Decimos que un computador utiliza **memoria virtual** cuando las direcciones de memoria de los programas se refieren a un espacio de memoria superior al espacio de memoria físico, espacio de memoria principal. La memoria virtual libera al programador de las restricciones de la memoria principal. En estos computadores diferenciamos entre el mapa de direcciones lógicas o virtuales (las direcciones que utilizan los programas) y el mapa de direcciones físicas o reales (las direcciones de la memoria principal). El espacio de memoria virtual utiliza como soporte un dispositivo de almacenamiento externo (habitualmente un disco magnético), mientras que el espacio de memoria físico se corresponde con la memoria principal del computador.

Organización Interna de la Memoria

Una memoria principal se compone de un conjunto de celdas básicas dotadas de una determinada organización. Cada celda soporta un bit de información. Los bits se agrupan en unidades direccionables denominadas palabras. La longitud de palabra la determina el número de bits que la componen y constituye la resolución de la memoria (mínima cantidad de información direccionable). La longitud de palabra suele oscilar desde 8 bits (byte) hasta 64 bits. Cada celda básica es un dispositivo físico con dos estados estables (o semi-estables) con capacidad para cambiar el estado (escritura) y determinar su valor (lectura). Aunque en los primeros computadores se utilizaron los materiales magnéticos como soporte de las celdas de memoria principal (memorias de ferritas, de película delgada, etc.) en la actualidad sólo se utilizan los materiales semiconductores.

Desde un punto de vista conceptual y con independencia de la tecnología, consideraremos la celda básica de memoria como un bloque con tres líneas de entrada (entrada dato, selección y lectura/escritura) y una de salida (salida dato). La celda sólo opera (lectura o escritura) cuando la selección está activa.

Organización 2D (Memorias Estáticas)

Las celdas se organizan en una matriz de dos dimensiones, en la que las filas vienen dadas por el número de palabras (n) y las columnas por la longitud (cantidad de bits) de cada palabra. Para seleccionar la palabra deseada se decodifican los m bits de dirección en un decodificador $m \times 2^m$, que tiene una señal de salida individualizada para cada palabra de memoria. Se usa la misma conexión para la lectura que para la escritura, bastando activar al transductor correspondiente para definir la operación.

Organización 3D (Memorias Dinámicas)

Se establecen n planos de memoria (uno para cada bit de la palabra). Dentro de cada plano se selecciona el punto de memoria haciendo coincidir las líneas de selección X e Y. 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. Para una configuración dada de las líneas de dirección se selecciona un único bit de la matriz.

Memoria Principal

Todos los tipos de memoria principal se implementan utilizando tecnología de semiconductores y tienen el transistor como elemento básico de su construcción. El elemento básico en toda memoria es la celda. Una celda permite almacenar un bit, un valor 0 o 1 definido por una diferencia de potencial eléctrico. La manera de construir una celda de memoria varía según la tecnología utilizada. La memoria interna es una memoria de acceso aleatorio; se puede acceder a cualquier palabra de memoria especificando una dirección de memoria. Una manera de clasificar la memoria interna según la perdurabilidad es la siguiente:

- **Memoria Volátil**

La **memoria volátil** es la memoria que necesita una corriente eléctrica para mantener su estado, de manera genérica denominada RAM. Las memorias volátiles pueden ser de dos tipos:

1. **SRAM: La memoria estática de acceso aleatorio (SRAM)** implementa cada celda de utilizando un flip-flop básico para almacenar un bit de información, y mantiene la información mientras el circuito de memoria recibe alimentación eléctrica. Para implementar cada celda de memoria son necesarios varios transistores, por lo que la memoria tiene una capacidad de integración limitada y su coste es elevado en relación con otros tipos de memoria RAM, como la DRAM; sin embargo, es el tipo de memoria RAM más rápido.
2. **DRAM: La memoria dinámica (DRAM)** implementa cada celda de memoria utilizando la carga de un condensador. A diferencia de los flip-flop, los condensadores con el tiempo pierden la carga almacenada y necesitan un circuito de refresco para mantener la carga y mantener, por lo tanto, el valor de cada bit almacenado. Eso provoca que tenga un tiempo de acceso mayor que la SRAM. Cada celda de memoria está formada por solo un transistor y un condensador; por lo tanto, las celdas de memoria son mucho más pequeñas que las celdas de memoria SRAM, lo que garantiza una gran escala de integración y al mismo tiempo permite hacer memorias más grandes en menos espacio.

- **Memoria no Volátil**

La **memoria no volátil** mantiene el estado sin necesidad de corriente eléctrica. Las memorias no volátiles pueden ser de diferentes tipos:

1. **Memoria de Solo Lectura o ROM (read only memory):** Se trata de memorias de solo lectura que no permiten operaciones de escritura y, por lo tanto, la información que contienen no se puede borrar ni modificar. Este tipo de memorias se pueden utilizar para almacenar los microprogramas en una unidad de control microprogramada. La

grabación de la información en este tipo de memorias forma parte del proceso de fabricación del chip de memoria.

2. **Memoria programable de solo lectura o PROM (programmable read only memory):**
Cuando hay que fabricar un número reducido de memorias ROM con la misma información grabada, se recurre a otro tipo de memorias ROM: las memorias ROM programables (PROM). A diferencia de las anteriores, la grabación no forma parte del proceso de fabricación de los chips de memoria, sino que se efectúa posteriormente con un proceso eléctrico utilizando un hardware especializado para la grabación de memorias de este tipo. Como el proceso de programación no forma parte del proceso de fabricación, el usuario final de este tipo de memorias puede grabar el contenido según sus necesidades. El proceso de grabación o programación solo se puede realizar una vez.
3. **Memoria reprogramable mayoritariamente de lectura:** Esta puede ser de 3 tipos
 - 1- **EPROM (Erasable Programmable Read Only Memory):** Se trata de memorias en las que habitualmente se hacen operaciones de lectura, pero cuyo contenido puede ser borrado y grabado de nuevo. Hay que destacar que el proceso de borrar es un proceso que borra completamente todo el contenido de la memoria; no se puede borrar solo una parte. Para borrar, se aplica luz ultravioleta sobre el chip de memoria EPROM; para permitir este proceso, el chip dispone de una pequeña ventana sobre la cual se aplica la luz ultravioleta. La grabación de la memoria se hace mediante un proceso eléctrico utilizando un hardware específico.
 - 2- **EEPROM (Electrically Erasable Programmable Read Only Memory):**
Permiten borrar el contenido y grabar información nueva; sin embargo, a diferencia de las memorias EPROM, todas las operaciones son realizadas eléctricamente. Para grabar datos no hay que borrarlos previamente; se permite modificar directamente solo uno o varios bytes sin modificar el resto de la información. Son memorias mayoritariamente de lectura, ya que el proceso de escritura es considerablemente más lento que el proceso de lectura.
 - 3- **Memoria Flash:** El borrado es eléctrico de la información, con la ventaja de que el proceso de borrar y grabar es muy rápido. La velocidad de lectura es superior a la velocidad de escritura, pero las dos son del mismo orden de magnitud. Este tipo de memoria no permite borrar la información byte a byte, sino que se deben borrar bloques de datos enteros.

Memoria Caché

La **Memoria Caché** se sitúa entre la memoria principal y el procesador, puede estar formada por uno o varios niveles. Tiene un tiempo de acceso inferior al de la memoria principal con el objetivo de reducir el tiempo de acceso medio a los datos, pero también tiene un tamaño mucho más reducido que la memoria principal. Si un dato está en la memoria caché, es posible proporcionarlo al procesador sin acceder a la memoria principal, si no, primero se lleva el dato de la memoria principal a la memoria caché y después se proporciona el dato al procesador.

Para trabajar con memoria caché, la memoria principal se organiza en bloques de palabras, de manera que cuando hay que trasladar datos de la memoria principal a la memoria caché se lleva un bloque entero de palabras de memoria. La memoria caché también se organiza en bloques que se denominan líneas. Cada línea está formada por un conjunto de palabras (el mismo número de palabras que tenga un bloque de memoria principal), más una etiqueta

compuesta por unos cuantos bits. El contenido de la etiqueta permitirá saber qué bloque de la memoria principal se encuentra en cada línea de la memoria caché en un momento dado.

Cada vez que el procesador quiere acceder a una palabra de memoria, primero se accede a la memoria caché; si la palabra de memoria se encuentra almacenada en la memoria caché, se proporciona al procesador y diremos que se ha producido un **acierto**. En caso contrario, se lleva el bloque de datos de la memoria principal que contiene la palabra de memoria hacia la memoria caché y, cuando la palabra ya está en la memoria caché, se proporciona al procesador; en este caso diremos que se ha producido un **fallo**.

Memoria Externa (Memoria Secundaria)

La **memoria externa** está formada por dispositivos de almacenamiento secundario (discos magnéticos, CD, DVD, BluRay, etc.). Estos dispositivos se pueden encontrar físicamente dentro o fuera de la computadora. La memoria externa es de tipo no volátil; por lo tanto, los datos que se quieran mantener durante un tiempo indefinido o de manera permanente se pueden almacenar en dispositivos de memoria externa. El método de acceso varía según el dispositivo: generalmente los dispositivos basados en disco utilizan un método de acceso directo, mientras que otros dispositivos, como las cintas magnéticas, pueden utilizar acceso secuencial.

Los datos almacenados en la memoria externa son visibles para el programador en forma de bloques de datos, no como datos individuales (bytes), normalmente en forma de registros o ficheros. El acceso a estos dispositivos se lleva a cabo mediante el sistema de E/S del computador y es gestionado por el sistema operativo.

Discos Magnéticos

Un disco es un plato circular construido con metal o plástico, cubierto por un material magnetizable. Los datos se graban en él y después se recuperan del disco a través de una bobina, llamada cabeza. Durante una operación de lectura o escritura, la cabeza permanece quieta mientras el plato rota bajo ella.

El mecanismo de escritura se basa en el campo magnético producido por el flujo eléctrico que atraviesa la bobina. Se envían pulsos a la cabeza, y se graban patrones magnéticos en la superficie bajo ella, con patrones diferentes para corrientes positivas y negativas. El mecanismo de lectura se basa en la corriente eléctrica que atraviesa la bobina, producida por un campo magnético que se mueve respecto a la bobina. Cuando la superficie del disco pasa bajo la cabeza, se genera una corriente de la misma polaridad que la que produjo la grabación magnética.

Organización y formato de los datos

Los datos se organizan en un conjunto de anillos concéntricos, llamados pistas. Cada pista es del mismo ancho que la cabeza. Las pistas están separadas por bandas vacías, lo que proviene errores debidos a deslineamientos de la cabeza o simplemente interferencias del campo magnético.

Los datos se transfieren hacia y desde el disco en bloques. El bloque es menor que la capacidad de una pista. De acuerdo con esto, los datos se almacenan en regiones del tamaño de un bloque, denominadas sectores.

Características físicas

Las principales características son:

- **Desplazamiento de cabezas:** las cabezas pueden ser:
 - *Fijas:* hay una cabeza de lectura/escritura por pista. Todas las cabezas se montan en un brazo rígido que se extiende a través de todas las pistas.
 - *Móvil:* hay sólo una cabeza de lectura/escritura que se monta en un brazo. Como la cabeza debe poder posicionarse encima de cualquier pista, el brazo debe extenderse o retraerse para este propósito.
- **Transportabilidad del disco:** se divide en dos categorías:
 - *No Extraíble:* está permanentemente montado en la unidad de disco.
 - *Extraíble:* puede ser quitado y sustituido por otro disco.
- **Superficie:** puede ser de una sola superficie o doble.
- **Platos:** puede tener un plato único o varios platos apilados verticalmente. En conjunto, se llama paquete de disco y disponen de varios brazos.
- **Mecanismo de la cabeza:** se clasifican en 3 tipos:
 - *Separación fija:* la cabeza se posiciona a una distancia fija sobre el plato, dejando entre ambos una capa de aire.
 - *Contacto:* la cabeza efectúa un contacto físico con el medio durante la operación de lectura o escritura. Se usa con los disquetes.
 - *Separación aerodinámica:* las cabezas están montadas en unidades herméticamente cerradas, casi libre de contaminación. Estos discos operan más cerca de la superficie, por tanto, permiten una densidad de datos mayor. La cabeza está en el contorno de una hoja de metal aerodinámica, que reposa suavemente sobre la superficie del plato cuando el disco no se mueve. La presión de aire generada por el giro del disco es suficiente para hacer subir la hoja encima de la superficie.

Parámetros para medir las prestaciones de un disco

Cuando una unidad de disco funciona, el disco rota a una velocidad constante. Para leer/escribir, la cabeza debe posicionarse en la pista deseada y al principio del sector deseado. En un sistema de cabeza móvil, el tiempo que tarda en posicionarse en la pista es el '*tiempo de búsqueda*'. Una vez seleccionada la pista, se debe esperar hasta que el sector apropiado rote hasta alinearse con la cabeza; este tiempo que pasa se conoce como '*tiempo de latencia*'. El tiempo que se tarda en llegar a la posición de lectura/escritura, '*tiempo de acceso*', se conoce como la suma del '*tiempo de búsqueda*' más '*tiempo de latencia*'. Una vez posicionada la cabeza, se lleva a cabo la operación de lectura/escritura, lo que conlleva un tiempo de transferencia de datos.

o *Tiempo de búsqueda* = constante del disco * pistas atravesadas + tiempo de comienzo.

o *Tiempo de latencia* = medio: 8,3 ms. en discos, entre 100 y 200 ms. en disqueteras.

o *Tiempo de transferencia* = bytes a transferir / (bytes de una pista * velocidad de rotación).

o *Tiempo de acceso medio total* = tiempo e/ pistas * cantidad de pistas + tiempo de medio giro.

o *Capacidad del disco* = bytes/sector*sectores/pista*pistas/superficie* nº de superficies.

RAID

Son esquemas estandarizados para el diseño de base de datos para discos múltiples. El esquema RAID consta de 7 niveles independientes, desde 0 hasta 6. Estos niveles no implican una relación jerárquica, sino que designan métodos diferentes que poseen tres características comunes:

1. RAID es un conjunto de unidades físicas de disco vistas por el sistema operativo como una única unidad lógica.
2. Los datos se distribuyen a través de las unidades físicas del conjunto de unidades.
3. La capacidad de los discos redundantes se usa para almacenar información de paridad que garantice la recuperación de los datos en caso de fallo de disco.

La estrategia RAID reemplaza una unidad de disco de gran capacidad por unidades múltiples de menor capacidad y distribuye los datos de forma que se puedan habilitar accesos simultáneos a los datos de varias unidades, mejorando, por tanto, las prestaciones de E/S, y permitiendo más fácilmente aumentos en la capacidad.

Nivel 0 de RAID

Para RAID 0, los datos del usuario y del sistema están distribuidos a lo largo de todos los discos del conjunto. Esto tiene una ventaja frente al uso de un único y gran disco: Si hay pendientes dos peticiones diferentes de E/S para dos bloques de datos diferentes, entonces es muy probable que los bloques pedidos estén en diferentes discos. Entonces, las peticiones se pueden emitir en paralelo, reduciendo el tiempo de cola de E/S.

En Raid 0 los datos están organizados en forma de tiras de datos a través de los discos disponibles. Todos los datos del usuario y del sistema se ven como almacenados en un disco lógico. El disco se divide en tiras, que pueden ser bloques, sectores o alguna otra unidad. Un conjunto de tiras consecutivas se llama “franja”.

Nivel 1 RAID

RAID 1 se diferencia de los niveles 2 a 6 en como se consigue la redundancia. En este nivel, la redundancia se logra duplicando todos los datos. Como en RAID 0, se hace una distribución de datos, pero en este caso, cada franja lógica se proyecta en dos discos físicos separados, de forma que cada disco del conjunto tiene un disco espejo que contiene los mismos datos.

La organización RAID 1 tiene 3 ventajas:

1. Una petición de lectura puede ser servida por cualquiera de los discos que contienen los datos pedidos, lo que hace que disminuya el tiempo de búsqueda.
2. Una petición de escritura requiere que las dos tiras se actualicen y esto puede hacerse en paralelo. Entonces, el resultado de la escritura viene determinada por la menos rápida de las dos escrituras.
3. La recuperación tras un fallo es sencilla.

La principal desventaja es el coste. Requiere el doble de espacio del disco lógico que puede soportar.

Nivel 2 RAID

Los niveles 2 y 3 de RAID usan una técnica de acceso paralelo. En un conjunto de acceso paralelo, todos los discos miembros participan en la ejecución de cada petición de E/S.

Como en los esquemas de RAID, se usa la descomposición de datos en tiras. En el caso de RAID 2 y 3, las tiras son muy pequeñas. Con RAID 2, el código de corrección de errores se calcula a partir de los bits de cada disco, y los bits del código se almacenan en las correspondientes posiciones de bit en varios discos de paridad.

Aunque RAID 2 requiere menos discos que el nivel 1, es bastante caro. El número de discos redundantes es proporcional al logaritmo del número de discos de datos. En una sola lectura se accede a todos los discos simultáneamente. En una escritura sencilla, la operación de escritura debe acceder a todos los discos de datos y de paridad.

Nivel 3 RAID

Se organiza de manera similar a RAID 2, la diferencia es que requiere solo un disco redundante. RAID 3 usa un acceso paralelo, con datos distribuidos en pequeñas tiras. En vez de un código de corrección de errores, se calcula un bit de paridad para el conjunto de bits individuales que están en la misma posición en todos los discos de datos.

RAID 3 puede conseguir velocidades de transferencia de datos muy altas y las mejoras de prestaciones en cuanto a E/S es notable. Por otra parte, sólo se puede ejecutar a la vez una petición de E/S, por lo tanto, el rendimiento sufre.

Nivel 4 RAID

Los niveles 4 y 5 de RAID usan una técnica de acceso independiente. En un conjunto de acceso independiente, cada disco opera independientemente, de forma que peticiones de E/S separadas se atienden en paralelo. Debido a esto, son más adecuados los conjuntos de acceso independiente para aplicaciones que requieren velocidades de petición de E/S altas y son menos adecuados para aplicaciones que requieren velocidades altas de transferencia de datos.

Con RAID 4 se calcula una tira de paridad, bit a bit, a partir de las correspondientes tiras de cada disco de datos, y los bits de paridad se almacenan en la correspondiente tira del disco de paridad.

RAID 4 lleva consigo una penalización en la escritura cuando se realiza una petición de escritura de E/S pequeña. Cada vez que se realiza una escritura, el software de gestión del conjunto debe actualizar, no sólo los datos del usuario, sino también, los bits de paridad correspondientes.

Nivel 5 RAID

RAID 5 está organizado de manera similar al 4. La diferencia es que distribuye las tiras de paridad a lo largo de todos los discos. Una distribución típica es un esquema cíclico. Para un

conjunto de n discos, la tira de paridad está en diferentes discos para las primeras n tiras, y este patrón se repite.

La distribución de las tiras de paridad a lo largo de todas las unidades evita el potencial cuello de botella de E/S encontrado en RAID 4.

Nivel 6 RAID

En este esquema se hacen dos cálculos de paridad distintos, que se almacenan en bloques separados en distintos discos. Por tanto, un conjunto RAID 6 cuyos datos requieran N discos, consta de $N+2$ discos.

La ventaja de RAID 6 es que proporciona una disponibilidad de los datos extremadamente alta, porque además de la paridad calculada como RAID 4 y 5, tiene un algoritmo de comprobación independiente que hace posible la regeneración de los datos, incluso si dos de los discos fallan. Por otra parte, RAID 6 incurre en una penalización de escritura, ya que cada escritura afecta a dos bloques de paridad.

Memoria Óptica

CD y CD-ROM

Tanto el **CD** de audio como el **CD-ROM** comparten una tecnología similar. La principal diferencia es que los lectores de CD-ROM son más robustos y tienen dispositivos de corrección de errores. El disco se forma a través de una resina, como el policarbonato, y se cubre con una superficie altamente reflectante (normalmente aluminio). La información digital se graba como una serie de hoyos microscópicos en la superficie reflectante. La superficie con los hoyos se protege con una capa final de laca transparente. La información se recupera con un láser de baja potencia, el cual pasa a través de la capa protectora transparente mientras un motor hace girar el disco sobre el láser. La intensidad de la luz reflejada cambia si se encuentra en un hoyo. Un foto sensor detecta este cambio que se convierte en una señal digital. La información se empaqueta con densidad uniforme a lo largo del disco en segmentos del mismo tamaño y se explora a la misma velocidad, rotando el disco a una velocidad variable. Se dice que el láser lee los hoyos a una velocidad lineal constante (CLV). Los datos de un CD-ROM se organizan en una secuencia de bloques, que constan de:

- *Sincronización*: Identifica el principio del bloque.
- *Cabecera*: Contiene la dirección del bloque y el byte de modo:
 - *Modo 0*: Campo de datos en blanco.
 - *Modo 1*: Uso de código de corrección de errores y 2.048 bytes de datos.
 - *Modo 2*: 2.336 bytes de datos de usuario sin código de corrección de errores.
- *Datos*: datos del usuario.
- *Auxiliar*:
 - *Modo 2*: Datos del usuario adicionales.
 - *Modo 1*: Código de corrección de errores de 288 bytes.

Los CD-ROM son apropiados para la distribución de grandes cantidades de datos a un gran número de usuarios.

Ventajas: (respecto con los discos magnéticos).

- La capacidad de almacenamiento es mucho mayor.

- Se puede replicar en grandes cantidades de forma barata.
- Es extraíble, puede ser usado como memoria de archivo.

Desventajas: (respecto con los discos magnéticos).

- Es de sólo lectura y no se puede actualizar.
- El tiempo de acceso es mayor que el de una unidad de disco magnético, tanto como medio segundo.

Disco Óptico borrable

Se puede escribir y reescribir rápidamente, como cualquier disco magnético. Se usa la tecnología magnético-óptica: Se utiliza la energía de un haz de láser, junto con un campo magnético, para grabar y borrar información, invirtiendo los polos magnéticos en una pequeña área del disco cubierta con un material magnético.

Las principales ventajas sobre los discos magnéticos son:

- Alta capacidad: más de la mitad.
- Intercambiabilidad: son extraíbles.
- Seguridad: más vida.

DVD

Sustituye a las cintas VHS de vídeo analógicas con su gran capacidad de almacenamiento, y sustituye al CD-ROM en los PC y servidores: el DVD lleva al video a la edad digital. Proporciona películas con una calidad de imagen impresionante; se pueden acceder aleatoriamente como en los CDs. Se puede grabar un gran volumen de datos (7 veces más que un CD-ROM). Claves que lo diferencia de los CD-ROM:

- Un DVD estándar almacena 4,7 GB por capa; los de doble capa y una cara almacenan 8,5 GB.
- Utiliza un formato de compresión (MPEG) par imágenes de pantalla completa de alta calidad.
- Un DVD de una capa puede almacenar una película de dos horas y media.

Discos Magnéticos Ópticos (WORM)

Es un CD de una-escritura-varias-lecturas, para las aplicaciones en las que sólo se necesitan unas pocas copias de un conjunto de datos. Para proporcionar un acceso más rápido, el WORM usa velocidad angular constante*, sacrificando parte de su capacidad.

Técnica típica para fabricar el disco: usar un láser de alta potencia para producir una serie de ampollas en el disco; así un láser de baja potencia puede producir calor suficiente para reventar las ampollas pregrabadas. Este disco óptico es atractivo para almacenar archivos de documentos y ficheros; proporciona una grabación permanente de grandes cantidades de datos del usuario.

* Velocidad angular constante: incrementa el espacio lineal entre bits de información grabados en los segmentos más externos del disco; explorando la información a la misma velocidad, girando el disco a una velocidad fija.

Cinta Magnética

Los sistemas de cinta usan la misma técnica de lectura y grabación que los discos. El medio es una cinta de plástico flexible, cubierta por un óxido magnético, análoga a una cinta de grabación doméstica. Se estructura en un pequeño número de pistas paralelas. Las primeras usaban 9 pistas (almacén de datos de un byte, con un bit de paridad); las nuevas usan 18 o 36 pistas (una palabra o doble palabra digital); los datos se leen y escriben en bloques contiguos (registros físicos) que están separados por bandas vacías (bandas inter-registros).

Es un dispositivo de acceso secuencial (tiene que leer los registros uno a uno para llegar al requerido; tiene que rebobinar un cierto trecho y empezar a leer si el registro deseado está en alguna posición anterior) que está en movimiento sólo durante las operaciones de lectura/escritura. Los extremos contienen unas marcas metálicas pegadas denominadas BOT y EOT para la detección automática del inicio y fin de la cinta, respectivamente. La capacidad depende de su longitud, densidad de grabación, longitud de bloque y formato de grabación. Las cintas magnéticas fueron el primer tipo de memorias secundarias.

Tipos de unidades de cinta magnética:

- **Cintas tradicionales de columnas de vacío:** las columnas de vacío tienen como objetivo mantener constante la tensión de la cinta bajo la estación de lectura/grabación.
- **Cintas tradicionales de brazos tensores:** Son más sencillas ya que no necesitan columnas de vacío, pero con ellas obtiene menor velocidad.
- **Unidades de cassette de audio:** se utilizan en microcomputadoras domésticas y pequeños sistemas informáticos; el movimiento de la cinta se realiza con motores que actúan directamente sobre los carretes, no siendo tan rápidos ni precisos como las dos anteriores.
- **Unidades de cassette digitales:** disponen de cabestrantes para control de la velocidad de lectura/grabación.

UNIDAD 6: Periféricos

Comunicación hombre-maquina

Permite la comunicación con el usuario de la computadora. Esta comunicación se realiza a través de periféricos (ej. un monitor)

Terminales

Las terminales de las computadoras constan de dos partes: un teclado y un monitor.

Teclado

Cuando utilizamos un teclado, 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 E/S correspondiente al carácter seleccionado, apareciendo éste en la pantalla si no es un carácter de control. Los teclados contienen los siguientes tipos de teclas:

- **Teclado principal:** Contiene los caracteres alfabéticos, numéricos y especiales.
- **Teclas de desplazamiento del cursor:** Permiten desplazar el cursor a izquierda, derecha, arriba y abajo.

- **Teclado numérico:** Es habitual en los teclados de computadora que las teclas correspondientes a los caracteres numéricos (cifras decimales), signos de operaciones básicas y punto decimal estén repetidas para facilitar al usuario la introducción de datos numéricos.
- **Teclas de funciones:** Son teclas cuyas funciones son definibles por el usuario o están.
- **Teclas de funciones locales:** Controlan funciones propias del terminal, como impresión del contenido de imagen cuando el computador está conectada a una impresora.

Monitores de CRT

Un monitor es una caja que contiene un tubo de rayos catódicos (CRT) y sus fuentes de potencia. El CRT contiene un cañón que puede disparar un haz de electrones contra una pantalla fosforescente cerca del frente del tubo (los monitores a color tienen 3 cañones: rojo, verde y azul). Una imagen de pantalla completa normalmente se redibuja entre 30 y 60 veces cada segundo. Para producir un patrón de puntos en la pantalla, hay una rejilla dentro del CRT que, cuando se aplica en ella un voltaje positivo el haz choca la pantalla y brilla y, cuando el voltaje es negativo la pantalla no brilla. Así, el voltaje aplicado a la rejilla hace que el patrón de bits aparezca en pantalla. Este mecanismo permite convertir una señal eléctrica binaria en una imagen formada por puntos brillantes y oscuros.

Monitor LCD (Pantalla de cristal líquido)

una pantalla LCD consiste en dos placas de vidrio paralelas entre las que hay un volumen sellado que contiene cristal líquido. Cada placa tiene conectados electrodos transparentes que crean campos eléctricos en el cristal. Una luz ilumina la pantalla desde atrás. Diferentes partes de la pantalla reciben diferentes voltajes, y con esto se controla la imagen que se exhibe. En los LCD color se utilizan filtros ópticos para separar la luz blanca en componentes rojo, verde y azul.

Clasificación de pantallas:

o Según la capacidad o no de mostrar colores se clasifican en:

- **Monitor monocromo:** Los colores usuales en un monitor monocromático son el blanco y negro.
- **Monitor color:** El color de cada punto se obtiene con mezcla de los colores rojo, verde y azul.

o Según su capacidad de representación se pueden clasificar en:

- **Pantallas de caracteres:** Sólo admiten caracteres.
- **Pantallas gráficas:** Permiten trazados de líneas y curvas continuas.

Principales parámetros que caracterizan a una pantalla:

- **Tamaño:** Se describen en función del tamaño de la diagonal principal, y se da en pulgadas.
- **Número de celdas o caracteres:** lo usual es una representación de 24 filas * 80 col. de caracteres.
- **Resolución:** número de puntos de imagen en pantalla. Este número no depende del tamaño de la pantalla.

Terminales de mapa de caracteres

Se usan comúnmente tres tipos de terminales: de mapa de caracteres, de mapa de bit RS-232-C. Todas ellas pueden usar cualquier tipo de teclado, pero difieren en la forma en que se comunica con ellas y en el manejo de las salidas.

En una computadora personal hay dos formas de organizar las salidas que se envían a la pantalla: un mapa de caracteres y un mapa de bits. En el mapa de caracteres, en la tarjeta de comunicación en serie hay una porción de memoria, llamada *memoria de video*, además de algunos circuitos para acceder al bus y generar señales de video.

Para exhibir caracteres, la CPU los copia en la memoria de video en bytes alternados. Cada carácter tiene asociado un byte de atributo que describe la forma en que debe exhibirse el carácter.

La tarea de la tarjeta de video consiste en traer una y otra vez caracteres de la RAM de video y generar la señal necesaria que se alimenta al monitor.

Terminales de mapa de bits

Usan pixeles que representan un bit de información. La RAM de video se ve como un gran arreglo de bits. El software puede crear ahí cualquier patrón que desee, y se exhibirá instantáneamente.

Las terminales de mapa de bits suelen usarse para presentar imágenes que contienen varias ventanas. Usar mapa de bits tiene dos desventajas: la primera, requieren una cantidad considerable de RAM de video. La segunda desventaja es su desempeño, ya que es muy costoso desplazar la imagen en la pantalla.

Terminales RS-232-C

Se crearon con el fin de que cualquier terminal pueda usarse con cualquier computadora. Tienen un conector estandarizado de 25 terminales. Para comunicarse, la computadora y la terminal contienen un chip llamado *transmisor-receptor universal asincrónico (UART)*, además de lógica para acceder al bus. Las entradas del teclado de la Terminal se someten a una convención paralelo- serie en la Terminal y el UART las reensambla en la computadora.

Mouse

Es una pequeña caja de plástico que descansa sobre la mesa junto al teclado. Cuando el ratón se mueve sobre la mesa, también se mueve un puntero en la pantalla, que apunta a los elementos. El ratón tiene uno, dos o tres botones en la parte superior, que permiten seleccionar opciones de menú.

Existen tres tipos de ratones: mecánicos, ópticos y optomecánicos.

Mecánicos: tenían dos ruedas de caucho que sobresalían por debajo y tenían sus ejes perpendiculares entre sí. Cuando el ratón se movía paralelo a su eje principal, una rueda giraba. Cuando se movía perpendicularmente, giraba la otra rueda. Cada rueda impulsaba un resistor variable. Si se medían los cambios en la resistencia, era posible ver qué tanto había

girado cada rueda y así calcular qué tanto se había movido el ratón en cada dirección. Este tipo se reemplazó por uno con una esfera que sale ligeramente de su base.

Ópticos: no tiene rueda ni esfera, tiene un LED y un fotodetector en la base. El ratón se usa sobre una base que contiene una cuadrícula de líneas. A medida que el ratón se mueve sobre ella, el fotodetector percibe el paso de las líneas por los cambios en la cantidad de luz que se refleja. Unos circuitos internos cuentan el número de líneas que se cruzan en cada dirección.

Optomecánicos: tiene una esfera que hace girar dos ejes perpendiculares entre sí. Los ejes se conectan a codificadores provistos de ranuras a través de las cuales puede pasar la luz. A medida que el ratón se mueve, los ejes giran y pulsos de luz inciden sobre los detectores cada vez que una ranura pasa entre un LED y su detector. El número de pulsos detectados es proporcional a la cantidad de movimiento.

Tabla digitalizadora

Está formada por un dispositivo apuntador movable más un panel sensible de hasta 1m² de tamaño. La resolución puede llegar hasta milésimas de milímetro. El dispositivo apuntador suele ser una mirilla o lápiz electrónico similar al lápiz óptico que tiene 4 botones que sirven, entre otras funciones, para recoger las coordenadas de la posición actual.

Para determinar la posición del puntero se pueden diferenciar 3 métodos

- **Electromagnéticos**
- **Por ultrasonidos**
- **Por presión**

Comunicación máquina-mundo físico

Permite la comunicación con dispositivos remotos. Para realizar esta conexión, se emplean una serie de periféricos que se denominan de forma genérica periféricos de control. Existen 4 tipos de periféricos de control: Cadena de lectura analógica, Cadena de lectura digital, Cadena de acción analógica, Cadena de acción digital.

La conexión con el modulo de E/S se realiza a través de señales de control, estado y datos. Los datos se intercambian en forma de un conjunto de bits que son enviados/recibidos a/desde el modulo de E/S. Las señales de control determinan la función que debe realizar el dispositivo. Las de estado indican el estado del chip.

Puertos normalizados

Permiten la conexión desde el exterior hacia la placa base. Funcionan con las características estándar y su fabricante tiene que conocer las normas estándar para que no surjan problemas de conexión con los periféricos. Las conexiones o puertos son:

- **Puertos Serie (COM):** Transmiten la información BIT a BIT. No son adecuados para transferir grandes cantidades de información. En ellos se conectan el modem, cuando es externo.

- **Puertos Paralelos (LPT):** Permiten una mayor velocidad en la transmisión de datos. La información se transmite en bytes o múltiplos. Es la conexión típica de impresoras o escáneres.
- **Puerto USB (Universal Serial Bus):** En ellos se conectan, en general, dispositivos que necesitan una alta velocidad de transferencia de datos como, por ejemplo, un escáner o una cámara digital, aunque también existen impresoras con este tipo de conexión.

Impresoras

Son periféricos que escriben la información de salida sobre papel. Su comportamiento inicialmente era muy similar al de las máquinas de escribir, pero hoy día son mucho más sofisticadas, pareciéndose algunas en su funcionamiento a máquinas fotocopadoras conectadas en línea con la computadora.

Clasificación y tipos de impresoras:

A) Calidad de impresión: Tiene en cuenta la calidad de presentación y de contraste de los caracteres impresos.

- **Impresoras normales:** Como las impresoras de línea, de rueda y térmicas.
- **Impresoras de semicalidad:** Como algunas impresoras matriciales.
- **Impresoras de calidad:** Como las impresoras margarita e impresoras láser.

B) Sistema de impresión: Según la forma en que realizan la impresión.

- **Por impacto de martillos:** El fundamento de las impresoras por impacto es similar al de las máquinas de escribir. Las impresoras de impacto son muy ruidosas y han sido las más utilizadas.
- **Sin impacto:** forman los caracteres sin necesidad de golpes mecánicos y utilizan otros principios físicos para transferir las imágenes al papel.

C) Forma de imprimir los caracteres:

- **Impresoras de caracteres:** Realizan la impresión por medio de un cabezal que va escribiendo la línea carácter a carácter.
- **Impresoras de líneas:** En estas impresoras se imprimen simultáneamente todos o varios de los caracteres correspondientes a una línea de impresión.
- **Impresoras de páginas:** Aquí se incluyen un grupo de impresoras que actúan de forma muy similar a las máquinas fotocopadoras.

Tipos de impresoras:

- **Impresora de Matriz:** El tipo de impresora más económico es la impresora de matriz, en la que una cabeza de impresión que contiene entre 7 y 24 agujas activables electromagnéticamente se mueve a lo largo de cada línea de impresión. Las impresoras de matriz son económicas y muy confiables, pero son lentas, ruidosas y malas para imprimir gráficos. Se usan en tickets, o para imprimir en formatos grandes, entre otros.
- **Impresora de Inyección de Tinta:** La cabeza de impresión móvil, que lleva un cartucho de tinta, se mueve horizontalmente a lo ancho del papel mientras rocía tinta con sus

boquillas. Dentro de cada boquilla, una pequeña gota de tinta se calienta eléctricamente más allá de su punto de ebullición, hasta que hace explosión, sale y choca con el papel. Luego la boquilla se enfría y el vacío que se produce succiona otra gota de tinta. Son económicas, silenciosas y con buena calidad, pero también son lentas, sus cartuchos son caros y producen impresiones saturadas de tinta.

- **Impresora Láser:** La impresora láser combina una alta calidad, excelente flexibilidad, buena velocidad y costo moderado en un mismo periférico. Permite usar todos los colores y puede imprimir páginas completas de texto o gráfico a gran velocidad. Su desventaja consiste en que son costosas.
- **Impresoras térmicas:** Son similares a las impresoras de agujas. Se utiliza un papel especial termosensible que se ennegrece al aplicar calor. El calor se transfiere desde el cabezal por una matriz de pequeñas resistencias en las que al pasar una corriente eléctrica por ellas se calientan, formándose los puntos en papel.

Parámetros que caracterizan a una impresora:

A) Velocidad de escritura: normalmente se expresa en las siguientes unidades:

- *Impresoras de caracteres:* Caracteres por segundo (cps).
- *Impresoras de líneas:* Líneas por minuto (lpm).
- *Impresoras de páginas:* Páginas por minuto (ppm).

B) Caracteres por línea: es el número máximo de caracteres que se pueden escribir en una línea.

C) Ancho del papel: se suele expresar en pulgadas.

D) Densidad de líneas: se expresa normalmente en líneas por pulgada e indica el espaciado entre líneas.

E) Color: es la posibilidad de imprimir en colores o no.

F) Resolución: la resolución se suele expresar como número de puntos por unidad de superficie.

Scanner

Es un dispositivo que recuerda a una fotocopidora y que se emplea para introducir imágenes en un computador. Las imágenes que se desee capturar deben estar correctamente iluminadas para evitar brillo y tonos no deseados. Son dispositivos de entrada de datos de propósito especial que se emplean conjuntamente con paquetes software para gráficos y pantallas de alta resolución. La mayor parte de los scanner capturan imágenes en color generando una determinada cantidad de bits por cada punto. La cantidad de espacio de almacenamiento que se necesita para una imagen depende de las dimensiones máximas de la imagen y de la resolución de captura del equipo; la resolución se describe en 'cantidad de puntos por pulgada' ('dot per inch' en inglés o con el acrónimo 'dpi') en el sentido horizontal (eje x) y en el vertical (eje y).

Comunicación máquina-máquina

Permite la comunicación con elementos de equipo (discos magnéticos, cintas, etc.). La comunicación es digital. Si los computadores están cerca se conectan directamente en Banda Base, mediante un bus paralelo que transmite bytes o palabras. Si existe una distancia de más de 100 mts. se utilizan Módems.

Modem

Es un dispositivo que permite conectar dos computadores remotos utilizando la línea telefónica de forma que puedan intercambiar información entre sí. La información que maneja el computador es digital, sin embargo, por las limitaciones físicas de las líneas de transmisión, no es posible enviar información digital a través de un circuito telefónico, solo pueden transmitirse señales analógicas. Entonces, para poder utilizar las líneas telefónicas (y en general cualquier línea de transmisión) para el envío de información entre computadoras digitales, es necesario un proceso de transformación de la información. Durante este proceso la información se adecua para ser transportada por el canal de comunicación. Este proceso se conoce como modulación-demodulación.

Un **Módem** es un dispositivo que posee conversores A/D y D/A especialmente adecuados para conectar líneas telefónicas al computador. De este modo las señales provenientes de una línea telefónica (por ejemplo, por una llamada) son interpretadas y "atendidas" por el módem, permitiendo que otra computadora transmita información directamente a la nuestra. Recíprocamente, nuestra voz, una imagen o datos guardados en nuestra computadora pueden ser manejados por el módem para ponerlos (previa llamada) sobre la línea telefónica que del otro lado tendrá una computadora receptora (o eventualmente un ser humano que descuelga un teléfono).

- **Baudios:** los módems envían datos como una serie de tonos a través de la línea telefónica. Los tonos se "encienden" (ON) o apagan (OFF) para indicar un 1 o un 0 digital. El baudio es el número de veces que esos tonos se ponen a ON o a OFF. Los módems modernos pueden enviar 4 o más bits por baudio.

- **Bits por segundo (BPS):** es el número efectivo de bits/seg que se transmiten en una línea por segundo. Un módem de 600 baudios puede transmitir a 1200, 2400 o, incluso a 9600 BPS.