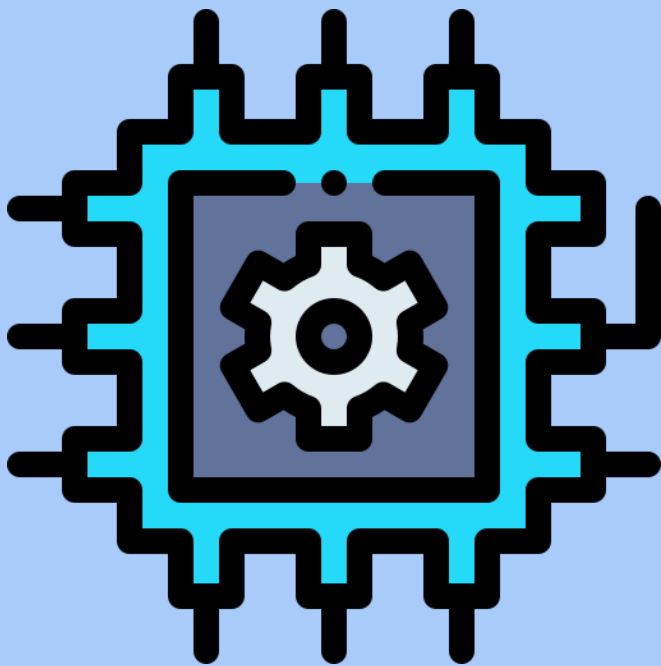


UNIDAD 1: RESUMEN CUADERNO



FUNDAMENTOS DEL HARDWARE
JUAN CARLOS NAVIDAD GARCÍA

Índice

1. Introducción.....	3
2. Comunicación de la información.	4
2.1. Codificación vs Cifrado:.....	4
3. Representación interna de la información.....	5
3.1. Capacidad	5
3.2 Transmisión.....	5
3.2 Unidades de medida para imagen.....	6
4. Sistemas de numeración:.....	6
4.1. Teorema fundamental de la numeración:.....	7
4.2. Sistema Posicional:	8
5. Representación de la información:	10
5.1. Representación alfanumérica:	10
5.2. Representación de instrucciones:.....	10
5.3. Representación de imagen y video:	11
5.4. Representación de sonido:	11
6. Características de un ordenador digital:	12
7. Componentes físicos, Arquitectura de Von Neumann y Harvard:	13
7.1. Componentes, bloques funcionales de la CPU:.....	15
Unidad de Control	15
Unidad Aritmético-Lógica	15
7.2. Función y propósito de la Memoria Interna:.....	16
Jerarquía de memoria.....	16
Funciones por niveles	17
7.3. Buses de Entrada/Salida:	18
Bus de datos:	18
Bus de direcciones:.....	18
Bus de control:	18
Buses multiplexados:.....	18
7.4. Arquitectura interna:.....	19
7.5. Periféricos:.....	19

1. Introducción

En la **teoría de sistemas**, un ordenador, se ve como una caja negra, por la que entran datos y sale información, por lo que debe de existir un proceso que realice esa función.

La diferencia entre **datos** e **información**, es que simplemente, la información es algo más **detallada**, por ejemplo, un **dato sería una fecha de nacimiento**, en cambio, como **información**, tendríamos el cómo **obtener nuestra edad**, en la que habría que hacer la **operación** de la diferencia entre la fecha de nacimiento y la fecha actual.

Los equipos se componen de **componentes lógicos (software)** y de **componentes físicos (hardware)** y entre ellos se encuentra el **firmware**. El **firmware es un software que maneja físicamente el hardware**, como puede ser la **BIOS** de una placa base, estos pueden ser modificables o no, pero eso depende de diferentes situaciones.

La **informática** es la ciencia que estudia la adquisición, representación, tratamiento y transmisión de la información, de manera automática mediante máquinas denominadas ordenadores.

El término **información**, normalmente hace referencia a un conjunto de símbolos que representan hechos, objetos u ideas.

Y como tal, necesitaremos un ordenador, un ordenador es una máquina, capaz de aceptar unos **datos de entrada**, efectuar, con ellos una serie de **operaciones** y **proporcionar la información** resultante a través de un **medio de salida**.

Es importante considerar que este flujo de información se realiza de **forma digital** por lo general. El **término dígito → dedo** nos da una idea aproximada de que la representación no va a tomar cualquier valor del mundo real (**señales analógicas**) sino sólo un número discreto de estos al igual que el **número de dedos es limitado**.

Ligado íntimamente a la **representación interna** de la información está el concepto de **bit**. Por simplicidad y velocidad, los ordenadores internamente sólo van a poder almacenar **dos estados (voltaje o ausencia de este, magnetizado o no, refleja un láser o no, etc)** que matemáticamente vamos a representar con un **cero** o un **uno**. De aquí aparece surge el concepto de **dígito binario, Binary digit o Bit**, ya que en inglés esa palabra tiene connotaciones de algo muy pequeño, esta será la **unidad más pequeña de medida de la información**.

2. Comunicación de la información.

En un sistema de información, es importante que haya un **emisor** y un **receptor**, también es importante que exista un **canal** o **vía de comunicación** y obviamente, un **mensaje**.

Teóricamente, la comunicación tendría el siguiente esquema:

Elementos de un sistema de comunicación. Los elementos básicos que intervienen en el proceso de comunicación son:

- **Emisor:** Ente que transmite la información.
- **Receptor:** Ente que recibe la información.
- **Canal:** Medio por el cual se transmite la información, inevitablemente contendrá ruido.
- **Mensaje:** Conjunto de señales, signos y símbolos que son objeto de la comunicación.
- **Protocolos y estándares:** Por último, es necesario para que la comunicación sea efectiva, que tanto emisor como receptor sigan un determinado conjunto de reglas en la comunicación. A éstas, se les denomina protocolos. En la actualidad existen multitud de protocolos, y muchos de ellos se encuentran estandarizados, como es el caso del conjunto de protocolos TCP/IP.

2.1. Codificación vs Cifrado:

La codificación es el **método que permite representar la información utilizando un conjunto de símbolos que se combinan siguiendo determinadas reglas**. Existen códigos lingüísticos y códigos escritos, como los sistemas de numeración, el código Braille, los jeroglíficos, las partituras, etc.

Las clases de codificación de datos más utilizadas son el **sistema decimal (base 10)**, el **octal (base 8)**, el **binario (base 2)**, el **hexadecimal (base 16)**, el **ASCII** y el **UNICODE**.

El **cifrado** es un método de **protección de datos** que consiste en alterarlos hasta hacerlos ilegibles. Los datos pasan de ser texto sin formato a ser texto cifrado por medio de un método denominado algoritmo. Quien desee acceder a los datos cifrados debe **descodificarlos** primero con la clave de descifrado correcta o conociendo el sistema de cifrado.

Hay diferentes sistemas de cifrado, los referidos a informática serían el sistema de **cifrado simétrico** y **asimétrico** mediante claves y los sistemas de cifrado comunes, como **cifrado por transposición**, **substitución**, **Julio César**, etc.

3. Representación interna de la información.

3.1. Capacidad

La unidad más pequeña de almacenamiento es el **bit**, por lo que, transformando la unidad a **8 bits**, será equivalente a **1 Byte**, y los múltiplos del mismo utilizando prefijos griegos: Kilo, Mega, Giga, Tera, Peta, Exa, Zetta, Yotta.

1 byte = 8bits

1000 bytes = 1 Kilobyte

1000 Kilobytes = 1 Megabyte

1000 Megabytes = 1 Gigabyte

Y así... sucesivamente.

3.2 Transmisión

Al igual, pasa con la velocidad de transmisión, estas medidas por norma general se miden por segundo:

bit/s, **kilobit/s**, **megabit/s**, **gigabit/s**, etc.

Otra unidad, serían los **Baudios**, pero actualmente no se utilizan y actualmente se utilizan el **bit/s**, que es una medida mucho más precisa.

Algo que es equivalente al **Megabit/s** excepto por el hecho de que mide todos los datos que se transfieren incluyendo información de sincronización y control de errores, lo cual permite utilizar unidades redondas a la hora de dar información sobre la tasa de transferencia de un **bus de comunicaciones**. A esta unidad de transmisión se le llama **Transfer** o **Megatransfer**.

3.2 Unidades de medida para imagen

La unidad básica para medir una imagen es el **pixel (picture element)** o **punto de luz**, cuyo tamaño, profundidad de color y distancia a otros puntos va a depender totalmente de la tecnología empleada y la precisión que se le quiera dar. Para sus múltiplos se utilizan los mismos prefijos que para el **bit**.

Puede haber varias tecnologías de paletas de colores podrían ser el **RGB** y **CMYK**.

La **profundidad de color** o **bits por píxel** es un concepto de la computación gráfica que se refiere a la cantidad de bits de información necesarios para representar el color de un píxel. Lo más normal son los **24 bits**, pero puede haber de **32**, **48** o **64**, aunque el ojo humano no podría interpretar más de **32 bits**.

4. Sistemas de numeración:

Se define **sistema de numeración** como el conjunto de símbolos utilizados para la representación de cantidades, así como las reglas que rigen dicha representación.

El sistema de numeración que utilizamos normalmente es el **sistema decimal**,

de base 10 → 1, 2, 3, 4, 5, 6, 7, 8, 9.

4.1. Teorema fundamental de la numeración:

Este teorema relaciona una cantidad expresada en **cualquier sistema de numeración** con la misma cantidad expresada en el **sistema decimal**; es decir, el valor decimal de una cantidad expresada en otro sistema de numeración que utiliza otra base. Viene dado por la fórmula:

$$N_i = \sum_{i=-d}^n (\text{dígito})_i \cdot (\text{base})^i$$

Un ejemplo:

1. Calculamos los valores de la fórmula:

- $d = 0$, no hay coma
- $i = -d = 0$
- $n = 3$

2. Calculamos los pesos asociados a los dígitos según la posición. El peso 0 lo tiene el dígito de la derecha, y el peso n el de la izquierda (véase la Tabla 1.1).

Pesos	3	2	1	0
	10	10^2	10^1	10^0
Dígitos	6	5	7	8

Tabla 1.1. Pesos asociados a la cantidad 6578.

3. Sumamos según la fórmula:

$$(6 \cdot 10^3) + (5 \cdot 10^2) + (7 \cdot 10^1) + (8 \cdot 10^0) \rightarrow \\ \rightarrow 6000 + 500 + 70 + 8 = 6578$$

Un ejemplo de cómo se podría pasar de otro sistema a decimal:

$$100111_2 \rightarrow 975_{10} \\ 1 \cdot 2^{-2} + 1 \cdot 2^{-1} + 1 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 \\ 0.25 + 0.5 + 1 + 0 + 0 + 8$$

4.2. Sistema Posicional:

- **Decimal (base 10):** 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9.
- **Binario (base 2):** 0 y 1.
- **Octal (base 8):** 0, 1, 2, 3, 4, 5, 6 y 7.
- **Hexadecimal (base 16):** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F.

Para transformar un número de **base binaria** a **decimal** se puede aplicar el **teorema fundamental de la numeración**, por ejemplo:

$$10)_2 = 1 \cdot 2^1 + 0 \cdot 2^0 = 2$$

A su vez, para transformar un número entero de **base decimal** a **base binaria**, se obtiene **dividiendo entre 2**, y después se realizan sucesivamente divisiones por los cocientes obtenidos, hasta llegar a un cociente menor a 2. El resultado es el último cociente junto con los restos obtenidos en el orden inverso de aparición.

Para calcular la **parte decimal**, esta se **multiplica por dos** y vamos cogiendo la parte entera resultante en orden de aparición hasta obtener cero o tener la precisión requerida.

$234,765_{10} \rightarrow 11101010,11000011_2$

Igual se hace con **octal** o **hexadecimal**, simplemente cambiamos cada base por la correspondiente.

Aquí enseño una tabla que puede ser de gran ayuda, sobre todo para hexadecimal:

Decimal	Binario	Hexadecimal	octal
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	8	10
9	01001	9	11
10	01010	A	12
11	01011	B	13
12	01100	C	14
13	01101	D	15
14	01110	E	16
15	01111	F	17

5. Representación de la información:

5.1. Representación alfanumérica:

Se le denomina como **codificación de entrada/salida**, que asocia cada carácter a una determinada combinación de bits.

Por ejemplo:

- BCD (Binary Coded Decimal)
- EBCDIC (Extended BCD Interchange Code)
- ASCII (American Standard Code for Information Interchange)
- Unicode (El más utilizado).

5.2. Representación de instrucciones:

Las **instrucciones de un lenguaje en alto nivel** o en **lenguaje ensamblador** se dan en forma de texto, representándose en consecuencia con un **código de E/S**, y el traductor correspondiente se encarga de transformarlas en instrucciones que internamente se representan como **cadena de bits**.

5.3. Representación de imagen y video:

La **codificación de imágenes** se puede realizar de dos formas: mediante **mapas de bits** (indicando un color para cada bit en pantalla) o mediante **representación vectorial** (indicando únicamente los vértices y el color de cada polígono).

A cada **pixel** se le asigna un color que representar, se realiza mediante escalas de color como **RGB (Red, Green and Blue)** o **CMYK (Cian, Magenta, Yellow and Black)**. Los pixeles también se les puede asignar el **canal Alfa** que sería un cuarto valor para representar **transparencia** en la imagen.

La **resolución de la pantalla** viene determinada por el número de bits que tenga, pero esta necesita tener una **profundidad de color**, que sería la cantidad de colores que nos permite la paleta que estemos utilizando, según el número de bits que le dediquemos a cada color. Por ejemplo, con una profundidad de color de **8 bits**, podremos representar **2^8 combinaciones de color**.

Un video está formado por **imágenes estáticas** que pasan muy rápidamente, a esta velocidad se le denomina como **FPS (Frames Per Second)**.

5.4. Representación de sonido:

Para representar **sonido** es necesario digitalizarlo. Este proceso sólo puede tomar un número discreto de valores del mundo real, por lo que se realiza un muestreo que consiste en tomar un cierto número limitado de valores (**frecuencias en el tiempo**) de dicho sonido.

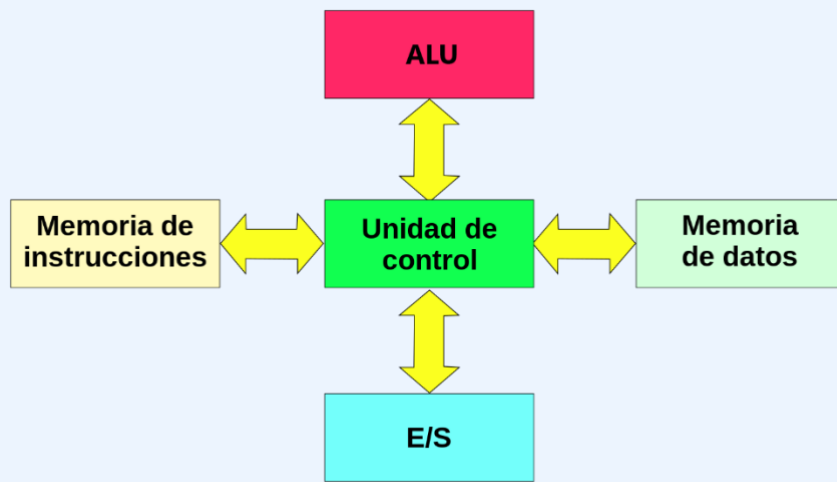
6. Características de un ordenador digital:

Las características más relevantes de un ordenador digital son:

- **Longitud de palabra:** Longitud de la cadena de bits que el computador puede manejar simultáneamente (en paralelo).
- **Memoria principal (RAM):** es donde se almacenan de forma **temporal** los datos de los programas que estás utilizando en este momento. Sus siglas significan *Random Access Memory*, lo que traducido al español sería *Memoria de Acceso Aleatorio*.
- **Memoria secundaria:** Cantidad de memoria de **almacenamiento permanente** de que dispone el ordenador (habitualmente, capacidad del disco duro, en gigabytes o terabytes).
- **Ancho de banda:** El ancho de banda es la cantidad de información por unidad de tiempo que se puede transmitir a través de un **bus o unidad de E/S**. Se mide en **MB/s Mb/s o GT/s**.
- **Frecuencia de reloj:** Frecuencia con la que el **oscilador interno (reloj)** coordina las operaciones del ordenador, habitualmente medidas en **MHZ o GHZ**.
- **MIPS (Millones de instrucciones por segundo):** Medida de rendimiento del computador, que indica el **número de instrucciones máquina** que puede ejecutar un procesador en un segundo, agrupadas en millones.
- **MFLOPS:** Medida análoga a la anterior, pero exclusivamente para **operaciones con números reales en coma flotante**, que presentan mayor carga de cálculo.
- **Test sintéticos (Benchmarks):** Son un conjunto variado de prueba cuyo resultado en tiempo de ejecución es más fiable que la medida en MIPS, MFLOPS o GHZ. El resultado de estos test da lugar a un **ranking de potencia**.

7. Componentes físicos, Arquitectura de Von Neumann y Harvard:

La **Arquitectura Harvard** es una configuración de la computadora en la que los datos y las instrucciones de un programa se encuentran en celdas separadas de la memoria, que se pueden abordar de forma independiente.



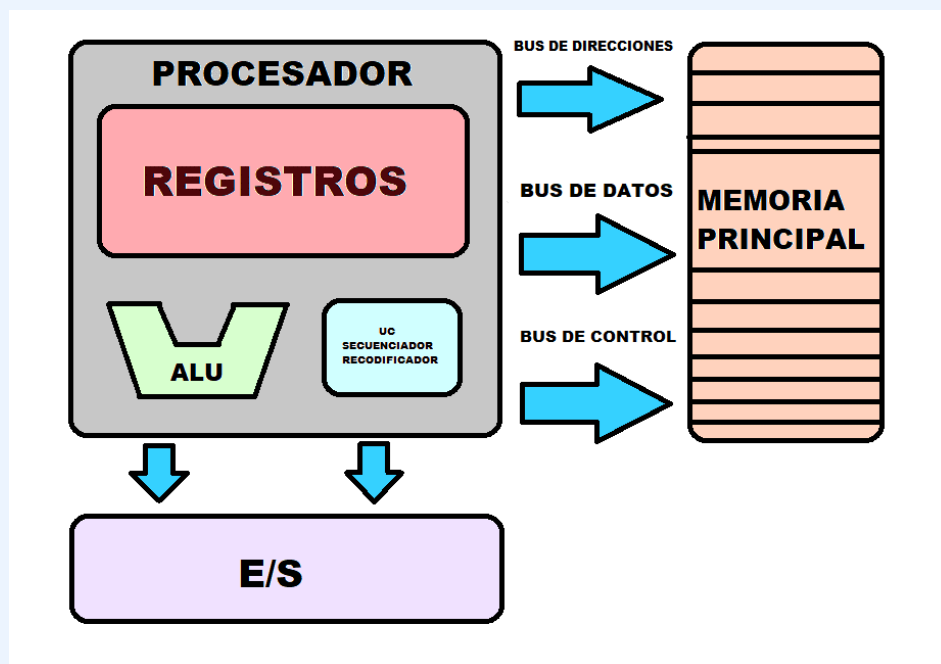
Arquitectura Harvard

La **Arquitectura Von Neumann** tuvo una gran aportación con respecto a la arquitectura preexistente (**arquitectura Harvard**) fue eliminar la lógica de programación basada en relés o conexiones cableadas, unificando memoria de datos y memoria de programa en una única memoria principal y simplificando por consiguiente la doble gestión de control de datos y programa. De esta forma además se garantiza la disponibilidad incremental de espacio en memoria ya que este se puede dividir entre datos y programas a voluntad.

Los elementos funcionales de la **Arquitectura Von Neumann** son, por tanto:

- Unidad de Control
- Unidad Aritmético-lógica
- Memoria Principal
- Unidad de E/S

La unidad de control y la unidad aritmético-lógica se agrupan en la denominada unidad central de proceso.



Arquitectura Von Neumann

7.1. Componentes, bloques funcionales de la CPU:

Unidad de Control

Microprogramada → En el caso de las unidades de control microprogramadas se utiliza un procesador que se encarga de realizar todas las funciones de la **unidad de control**, esto permite optimizar la forma en la que las instrucciones se van a ejecutar e incluso añadir nuevas instrucciones.

Microcableada → Las unidades de control microcableadas se utilizan en procesadores muy simples, ya que en ellas la tabla de registros e instrucciones no se puede cambiar al encontrarse todos los caminos de datos cableados previamente de manera fija, en una unidad de control cableada las instrucciones van a tener siempre el mismo rendimiento y no se van a poder añadir de nuevas ni optimizar su funcionamiento.

Unidad Aritmético-Lógica

Es la unidad funcional encargada de realizar las operaciones **aritméticas** y **lógicas** bajo la supervisión de la unidad de control.

Las operaciones que es capaz de realizar la **ALU** se clasifican en tres grupos:

- **Aritméticas.** Normalmente son la suma (ADD), la resta (SUB), la **multiplicación** (MUL), la **división** (DIV) y el **cambio de signo**.
- **Lógicas.** Normalmente son la **negación** (NOT), la **suma lógica** (OR), el producto lógico (AND) y la **suma exclusiva** (XOR).
- **Desplazamiento.** Consiste en desplazar los bits de una palabra un determinado número de posiciones hacia la derecha o hacia la izquierda.

El **número de operadores** que puede aceptar la **unidad aritmético-lógica** también es variable, clasificándose en operadores **monádicos**, **diádicos** y **triádicos** si es capaz de trabajar con uno, dos o tres datos de forma simultánea respectivamente.

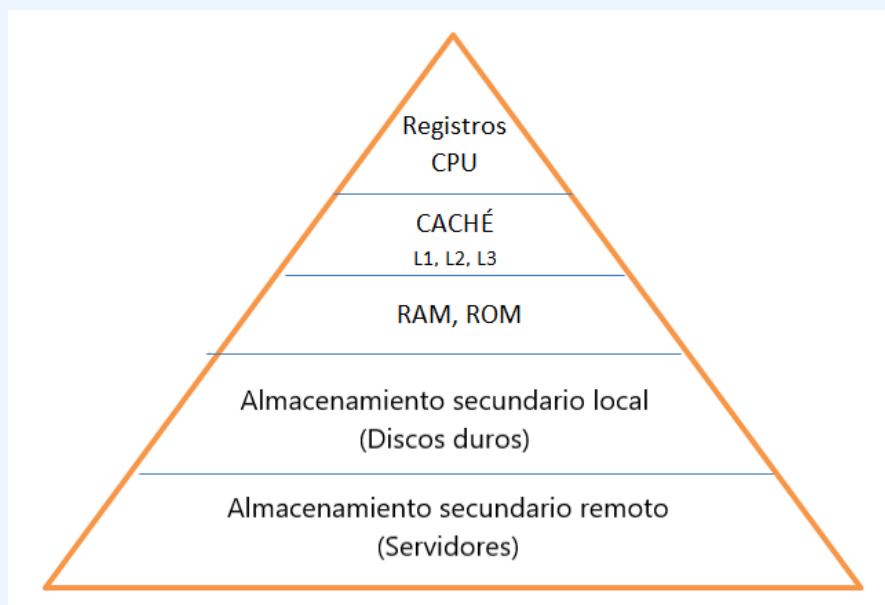
7.2. Función y propósito de la Memoria Interna:

La memoria es considerada un único componente funcional, compuesto de celdas que pueden almacenar bits de forma independiente, con dos operaciones básicas: **lectura** y **escritura**.

Jerarquía de memoria

Las distintas memorias de menor a mayor capacidad y de mayor a menor velocidad y coste se clasifican en:

- Registros (del procesador) • Memoria caché (estructurada en niveles: L1, L2, L3, etc)
- Memoria Principal (RAM Dinámica)
- Memoria secundaria (Discos duros y de estado sólido)
- Memoria masiva auxiliar (Soportes de respaldo)



Pirámide jerárquica de memoria

Funciones por niveles

- **Nivel 0 - Registros:** Almacena datos sobre los que la unidad de control está trabajando directamente.
- **Nivel 1 - Caché:** Memoria formada por circuitos integrados **SRAM**. A su vez se encuentra dividida en varios niveles. Se utiliza para minimizar los accesos a la memoria principal.
- **Nivel 2 - Memoria principal:** Memoria formada por **DRAM**, cuya función es almacenar las instrucciones y datos que se están utilizando en la ejecución de un proceso. Nos encontramos con:

RAM → Random Access Memory

ROM → Read-Only Memory

|

CMOS → BIOS

|

PROM/EEPROM → Electrically Erasable & Programmable Read-Only Memory. - Las EEPROM son las que se utilizan hoy en día.

- **Nivel 3 - Memoria secundaria:** Es el conjunto de dispositivos y soportes de almacenamiento de datos que conforman el subsistema de memoria de la computadora, en comparación con la memoria principal, esta es de lectura y escritura.

7.3. Buses de Entrada/Salida:

El conjunto de componentes de un ordenador se comunica entre sí mediante ciertos caminos, que constituyen la **estructura de interconexión**, o **buses de E/S**. En la arquitectura Von Neumann se distinguen los **buses de datos**, **dirección** y **control**:

Bus de datos:

El **bus de datos** permite el intercambio de datos entre la CPU y el resto de unidades.

Bus de direcciones:

El **bus de direcciones** es un canal del microprocesador totalmente independiente del **bus de datos** donde se establece la dirección de memoria del dato en tránsito.

El **bus de dirección** consiste en el **conjunto de líneas eléctricas** necesarias para establecer una dirección. La capacidad de la memoria que se puede direccionar depende de la cantidad de bits que conforman el bus de direcciones, siendo 2^n el tamaño máximo en bits del **banco de memoria** que se podrá direccionar con n líneas.

Bus de control:

El **bus de control** gobierna el uso y acceso a las líneas de datos y de direcciones, este se compone de **señales de control** que transmiten tanto órdenes como información de sincronización, evitando que haya colisión de información en el sistema, ya que nos encontramos con que este bus controla las líneas que están compartidas por todos los componentes, necesita este mecanismo.

Buses multiplexados:

Algunos diseños utilizan **líneas eléctricas multiplexadas** para el bus de direcciones y el **bus de datos**. Esto significa que un mismo conjunto de **líneas eléctricas** se comportan unas veces como **bus de direcciones** y otras veces como **bus de datos**, pero **nunca al mismo tiempo**. Una **línea de control** permite discernir cuál de las dos funciones está activa.

7.4. Arquitectura interna:

RISC → Reduced Instruction Set Computer - En la arquitectura computacional, **RISC** es un tipo de microprocesador con las siguientes características fundamentales:

- Instrucciones de tamaño fijo y presentadas en un reducido número de formatos.
- Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.

En resumen, son procesadores que están diseñados para ejecutar un **número reducido de tipos de instrucciones** que les permite operar a una velocidad más elevada. Esta arquitectura, principalmente requiere menos cantidad de hardware y una **mayor flexibilidad de construcción**.

CISC → Complex Instruction Set Computer - En la arquitectura computacional, **CISC** es un modelo de arquitectura de computadora. Los **microprocesadores CISC** tienen un conjunto de instrucciones que se caracteriza por ser muy **amplio** y permitir **operaciones complejas** entre operandos situados en la memoria o en los registros internos, en contraposición a la arquitectura **RISC**.

7.5. Periféricos:

Los **periféricos** se interconectan a la **unidad de E/S**.

Cada periférico dispone de un **circuito controlador (controladora)** que gestiona toda la lógica del mismo.

Las peculiaridades de cada dispositivo son gestionadas de forma transparente por el sistema operativo mediante un **software controlador (driver)** que es normalmente provisto por el propio fabricante.

Los periféricos pueden ser clasificados en **dispositivos de Entrada**, de **Salida**, o de **Entrada y Salida**; estos últimos son de tipo **almacenamiento** o **comunicación**.