

# INSTITUTO POLITÉCNICO NACIONAL

## ESCOM



*Unidad de Aprendizaje*

## *Arquitectura de Computadoras*

M. en C. María Elena Aguilar Jáuregui

*Febrero 2025*

*¿Qué es una computadora?*



# Computadora

Conocida como PC (*personal computer*) es un dispositivo electrónico que puede recibir, almacenar, procesar y enviar datos de acuerdo a las instrucciones de un programa.

Consta de varios componentes clave:



Hardware

# Software

- Sistema Operativo:  
Windows, Linux, MacOS



- Lenguajes de Programación:  
C, JavaScript, Python, Java, C++, CSS, perl, php, swift.



- Aplicaciones específicas:  
Procesadores de texto, navegadores WEB, programas de edición, programas específicos.



# *¿Qué es la arquitectura de computadoras?*



# Arquitectura de computadoras

Es el estudio y diseño de los componentes y la estructura interna de un sistema de cómputo



# Arquitectura de Computadoras

Define qué hace la computadora (vista lógica)

Es fundamental para comprender cómo funcionan los sistemas informáticos a nivel hardware y cómo interactúan con el software.

Enfocada a los componentes esenciales de una computadora y su organización, así como las técnicas de diseño y optimización de sus componentes para lograr un rendimiento eficiente.



La **Arquitectura de Computadoras** permite mejorar el diseño de sistemas informáticos, tanto para el desarrollo de software (afecta directamente cómo se escribirán y ejecutarán los programas) como para la ingeniería de hardware y la interacción entre ambos, optimizando el rendimiento de trabajo.



- 1.1 Fundamentos de arquitectura de computadoras
  - 1.1.1 Arquitectura y organización de computadoras
  - 1.1.2 Memoria de programa
  - 1.1.3 Memoria de datos
  - 1.1.4 Unidad aritmética lógica
  - 1.1.5 Registros
  - 1.1.6 Unidad de control
- 1.2 Arquitecturas clásicas
  - 1.2.1 Arquitectura Von Neuman
  - 1.2.2 Arquitectura RISC y CISC
  - 1.2.3 Arquitectura Harvard
- 1.3 Arquitecturas especializadas
  - 1.3.1 Arquitectura Superscalar
  - 1.3.2 Arquitectura vectorial

## 2.1 Formato de las instrucciones

### 2.1.1 Formato tipo R

### 2.1.2 Formato tipo I

### 2.1.3 Formato tipo J

### 2.1.4 Otros formatos

## 2.2 Tipos de instrucción

### 2.2.1 Instrucciones de carga y almacenamiento

### 2.2.2 Instrucciones de aritméticas y lógicas

### 2.2.3 Instrucciones de corrimiento

### 2.2.4 Instrucciones de brinco condicional e incondicional

### 2.2.5 Instrucciones de manejo de subrutinas

### 2.2.6 Otras instrucciones

## 2.3 Programación

### 2.3.1 Instrucciones de carga y almacenamiento

### 2.3.2 Instrucciones aritméticas y lógicas

### 2.3.3 Instrucciones de corrimiento

### 2.3.4 Instrucciones de brinco condicional e incondicional

### 2.3.5 Instrucciones de manejo de subrutinas

### 2.3.6 Implementación usando simuladores

- 3.1 Unidad aritmética y lógica
  - 3.1.1 Análisis usando sumador con acarreo en cascada
  - 3.1.2 Análisis usando sumador con acarreo anticipado
  - 3.1.3 Análisis usando otras arquitecturas
  - 3.1.4 Implementación usando HDL en lógica reconfigurable
- 3.2 Memoria de programa, datos y pila.
  - 3.2.1 Análisis con ROM y RAM multipuerto
  - 3.2.2 Implementación de pila en software y hardware
  - 3.2.3 Implementación usando HDL en lógica reconfigurable
- 3.3 Archivos de registros
  - 3.3.1 Análisis con RAM multipuerto
  - 3.3.2 Análisis usando otras arquitecturas
  - 3.3.3 Implementación usando HDL en lógica reconfigurable
- 3.4 Unidad de control
  - 3.4.1 Ruta de datos del procesador
  - 3.4.2 Microinstrucciones
  - 3.4.3 Memorias de microcódigo
  - 3.4.4 Decodificador de instrucción
  - 3.4.5 Registro de estado
  - 3.4.6 Análisis de banderas para determinar condiciones
  - 3.4.7 Autómata de control
  - 3.4.8 Implementación usando HDL en lógica reconfigurable
- 3.5 Implementación de procesador RISC monociclo
  - 3.5.1 Implementación usando HDL en lógica reconfigurable
  - 3.5.2 Ejecución de programas en los procesadores
  - 3.5.3 Implementación de analizadores lógicos en FPGA para la depuración de programas en ejecución

## 4.1 Segmentación

### 4.1.1 Etapas de segmentación

### 4.1.2 Registros inter etapa

### 4.1.3 Implementación usando HDL en lógica reconfigurable

## 4.2 Riesgos (Hazards)

### 4.2.1 Riesgo estructural

### 4.2.2 Riesgos de datos

### 4.2.3 Método de bypassing

### 4.2.4 Método de forwarding

### 4.2.5 Implementación usando HDL en lógica reconfigurable

## 4.3 Riesgo de Control

### 4.3.1 Predictores de salto

### 4.3.2 Predicción estática

### 4.3.3 Predicción dinámica

### 4.3.4 Implementación usando HDL en lógica reconfigurable

## 5.1 Jerarquía de la memoria

### 5.1.1. Memoria principal

### 5.1.2 Memoria Caché

### 5.1.3 Memoria Virtual

### 5.1.4 Tecnologías de memorias

## 5.2 Memoria caché

### 5.2.1 Cache hit y miss.

### 5.2.2. Estrategia Write Through y Write Back

### 5.2.3 Caché con mapeo directo

### 5.2.4 Caché asociativa por conjuntos

### 5.2.5 Caché completamente asociativo

### 5.2.6 Algoritmos de sustitución

### 5.2.7 Implementación usando HDL en lógica reconfigurable

## 5.3 Memoria virtual

### 5.3.1 Paginación.

### 5.3.2. Unidad de Administración de Memoria (MMU).

### 5.3.3. Tablas de páginas

### 5.3.4. Arquitecturas de TLB.

### 5.3.5. Algoritmos de reemplazo de páginas

# 1 Componentes fundamentales de la computadora

**Unidad Central de Procesamiento (CPU):** Estructura y funcionamiento de la CPU, partes principales (ALU, registros, unidades de control, memoria) y el ciclo de instrucción.

**Memoria:** Diferentes tipos de memoria primaria y secundaria (RAM, caché, discos duros, SSDs, virtual y la jerarquía de memoria).

**Dispositivos de Entrada/Salida (I/O):** Dispositivos que interactúan con la CPU y la memoria para proporcionar y obtener información.

## 2. Organización y estructura del Sistema

**Bus de datos:** Conexiones de datos entre los diferentes componentes de la computadora.

**Arquitectura de buses:** Tipos de buses y cómo se gestionan las transferencias de datos.

## 3 Conjuntos de instrucciones

**ISA:** Estudia el conjunto de instrucciones que una CPU puede ejecutar y cómo se representan en lenguaje ensamblador y máquina.



## 4 Pipelines y paralelismo

**Pipelining:** Técnica para aumentar el rendimiento mediante la superposición de la ejecución de varias instrucciones.

**Paralelismo:** Ejecución de múltiples instrucciones en paralelo, ya sea en múltiples núcleos o mediante técnicas como SIMD (Single Instruction, Multiple Data).

## 5. Arquitecturas avanzadas

**Procesadores multi-core:** Diseño de procesadores con múltiples núcleos para mejorar el rendimiento.

**Arquitectura de memoria distribuida:** Gestión de la memoria en sistemas multiprocesador.



Máquina que resuelve problemas ejecutando las instrucciones que recibe del usuario.



Secuencia de Instrucciones que describen cómo realizar cierta tarea. Las instrucciones primitivas permiten la comunicación entre usuario-máquina, este es el Lenguaje Máquina.

Instrucciones originales de la computadora  
Lenguaje 0 (L0)



Instrucciones del usuario  
Lenguaje 1 (L1)

**Traducción:** Todo el programa escrito en L1 se convierte primero en un programa escrito en L0, el programa en L1 se desecha, se carga en la memoria el programa L0 y se ejecuta por la computadora.

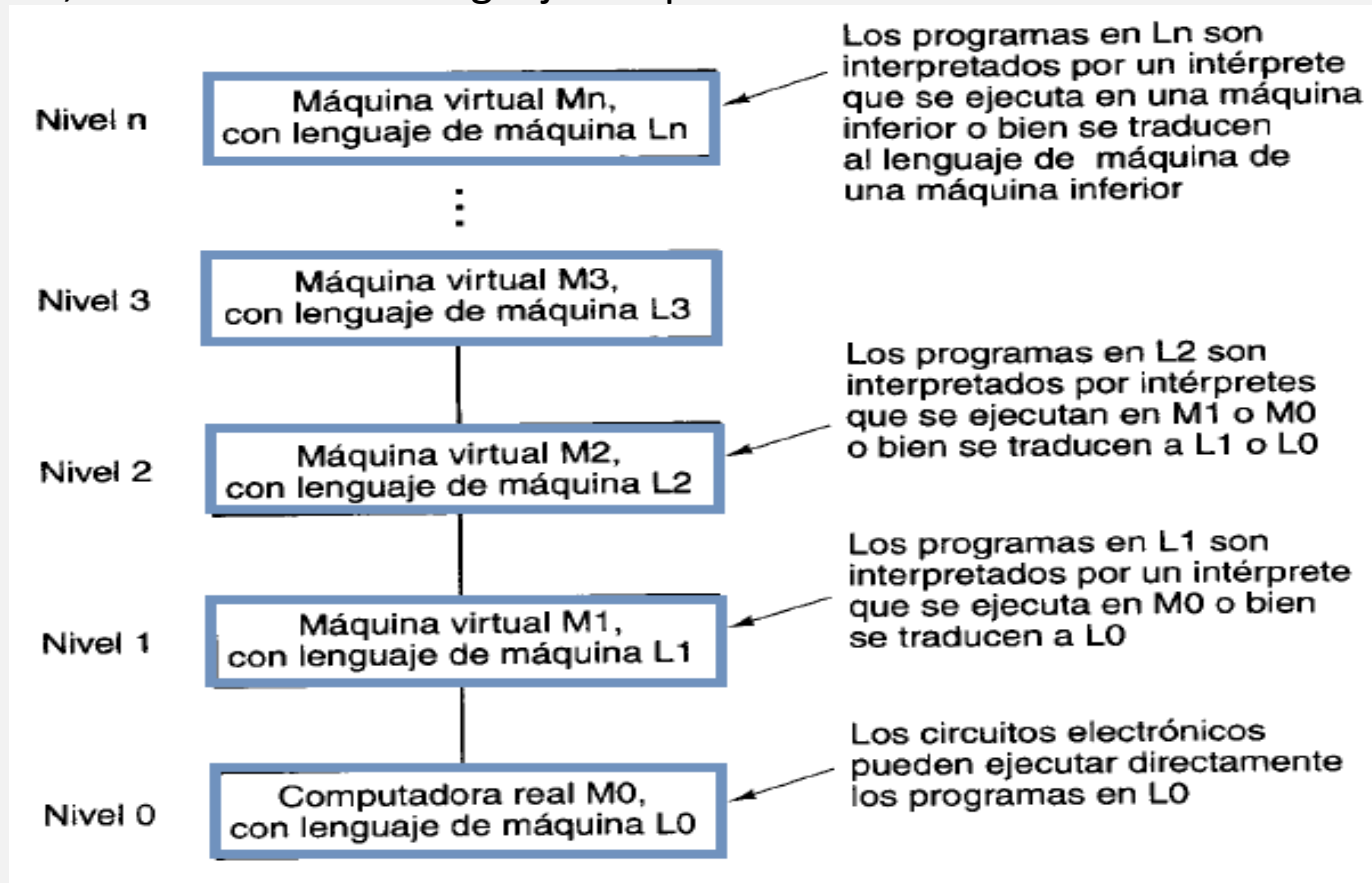
**Interpretación:** A partir de un programa escrito en L1, cada instrucción se examina y se decodifica y se ejecuta inmediatamente. No se genera ningún programa traducido. El programa que implementa esta técnica se llama **intérprete** y es el que controla la computadora.

# Máquina Virtual

Se puede entender como una capa adicional situada sobre el hardware y el sistema operativo. Esta capa permite simular un entorno de hardware completo, con la ejecución de múltiples sistemas operativos y aplicaciones en un mismo hardware físico.

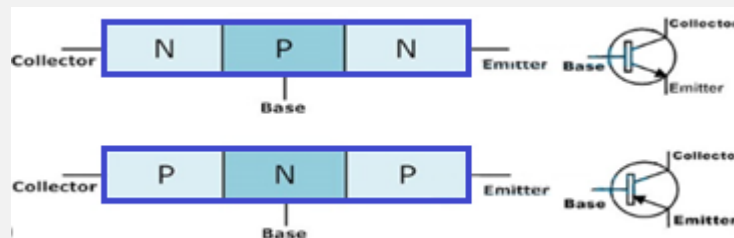
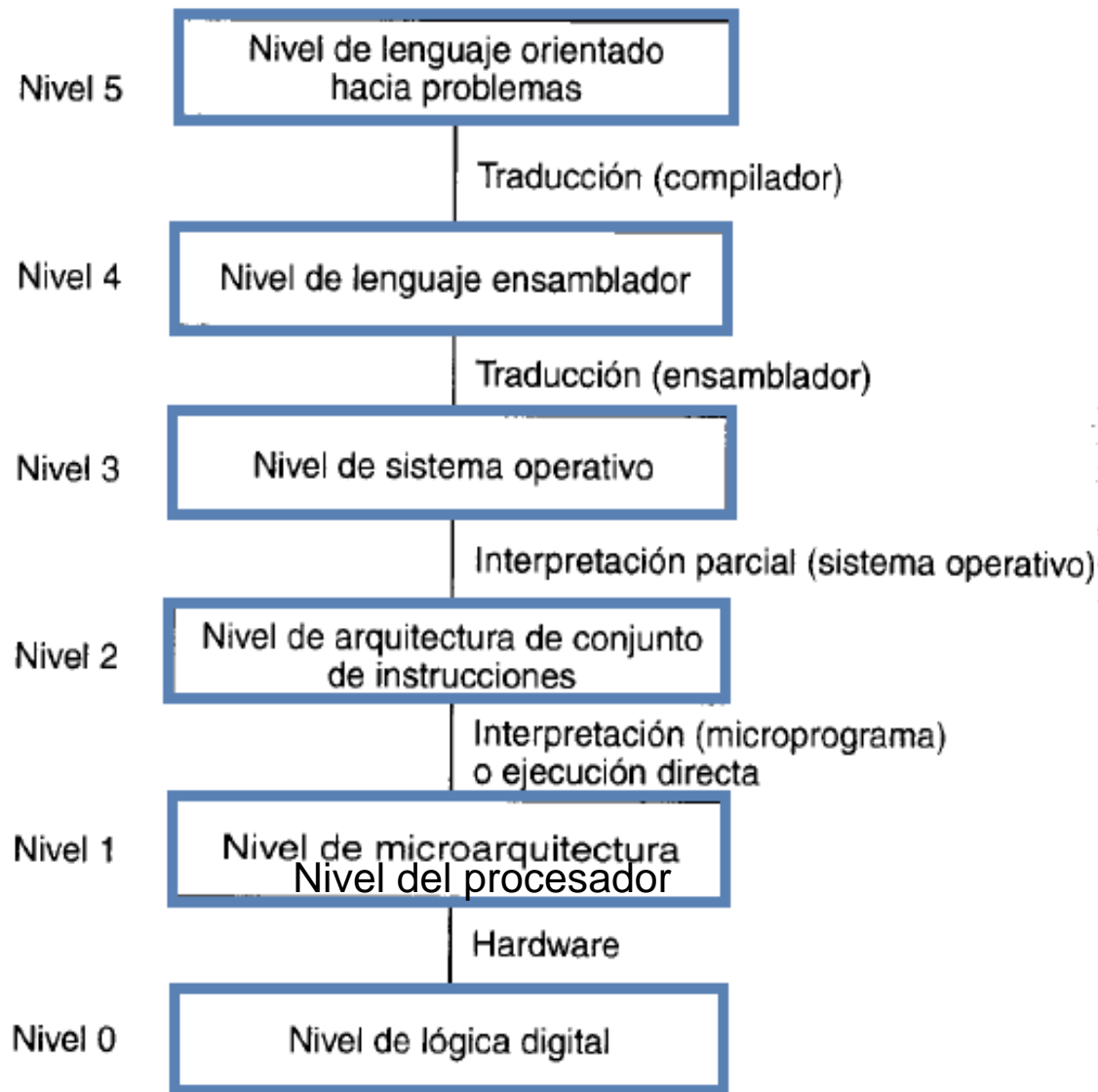
## Máquina multinivel

Una computadora con n niveles puede verse como n máquinas virtuales distintas, cada una con su lenguaje máquina.



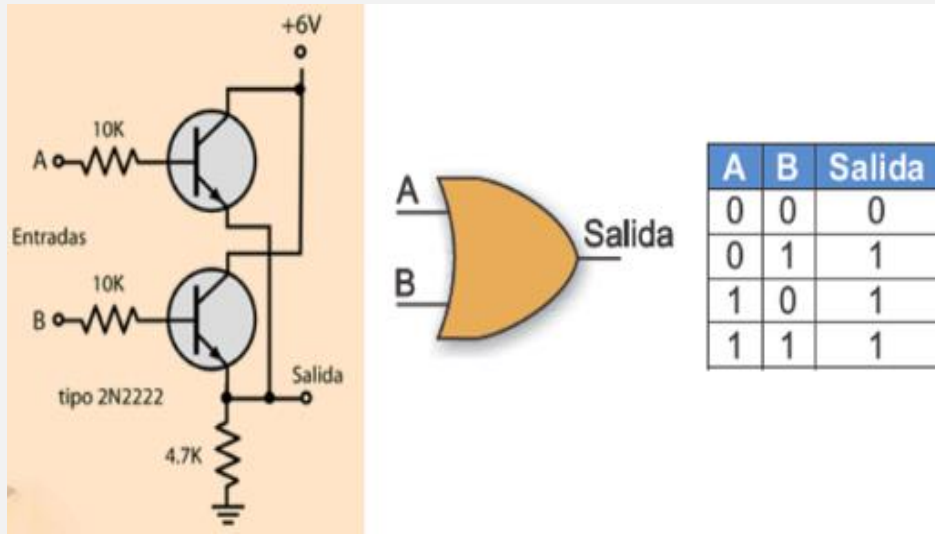
# Máquina multinivel contemporánea

Una computadora está formada por elementos electrónicos y para el análisis y diseño es necesario la división jerárquica en subsistemas relacionados. Cada nivel evoluciona en forma estructural hasta alcanzar el nivel elemental o fundamental.

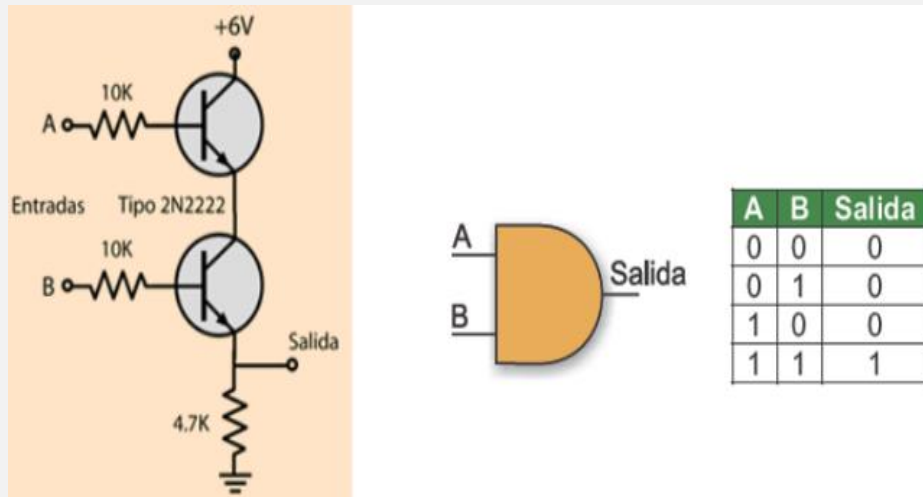


# Nivel de lógica digital

## Compuerta OR



Cualquiera de los diodos base-emisor entran en conducción suficiente para llevarlo a la saturación, el voltaje del emisor respecto a tierra está próximo a 5 voltios y se puede usar como lógica alta (1 o alto)

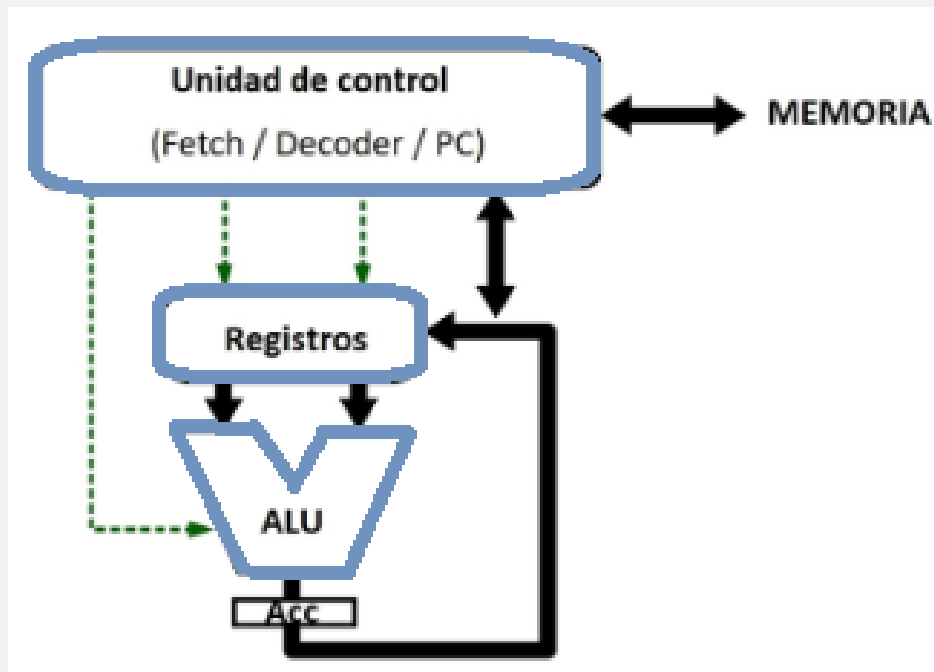


Cuando ambos diodos base-emisor entran en conducción suficiente para llevarlos a la saturación, el voltaje de salida respecto a tierra está próximo a 5 voltios y se puede usar como lógica alta (1 ó high)

## Nivel de microarquitectura

En este nivel se tiene una colección de 8 a 32 registros para formar una memoria local y la ALU que efectúa operaciones aritméticas sencillas. Los registros se conectan a la ALU y forman una Trayectoria de datos. Se selecciona uno o dos registros para las operaciones.

Aquí se describe cómo están organizados internamente los procesadores, los cuales contienen unidades de ejecución (ALU, FPU), unidades de control, registros, cachés, buses y otros elementos.





## Nivel de arquitectura del conjunto de instrucciones (Instruction Set Architecture) o nivel ISA

Cada fabricante de computadoras publica su manual de referencia de Lenguaje Máquina

Se describe el conjunto de instrucciones de la máquina que el microprograma o los circuitos de ejecución del hardware ejecutan en forma interpretativa.

## Nivel de máquina del sistema operativo

Es la estructura de los primeros sistemas operativos. Constaba de un sólo programa compuesto de un conjunto de rutinas entrelazadas de tal forma que cada una puede llamar a cualquier otra

La **arquitectura de computadoras** se enfoca en cómo los diferentes elementos de una computadora se organizan e interrelacionan para ejecutar instrucciones y realizar tareas de manera eficiente.

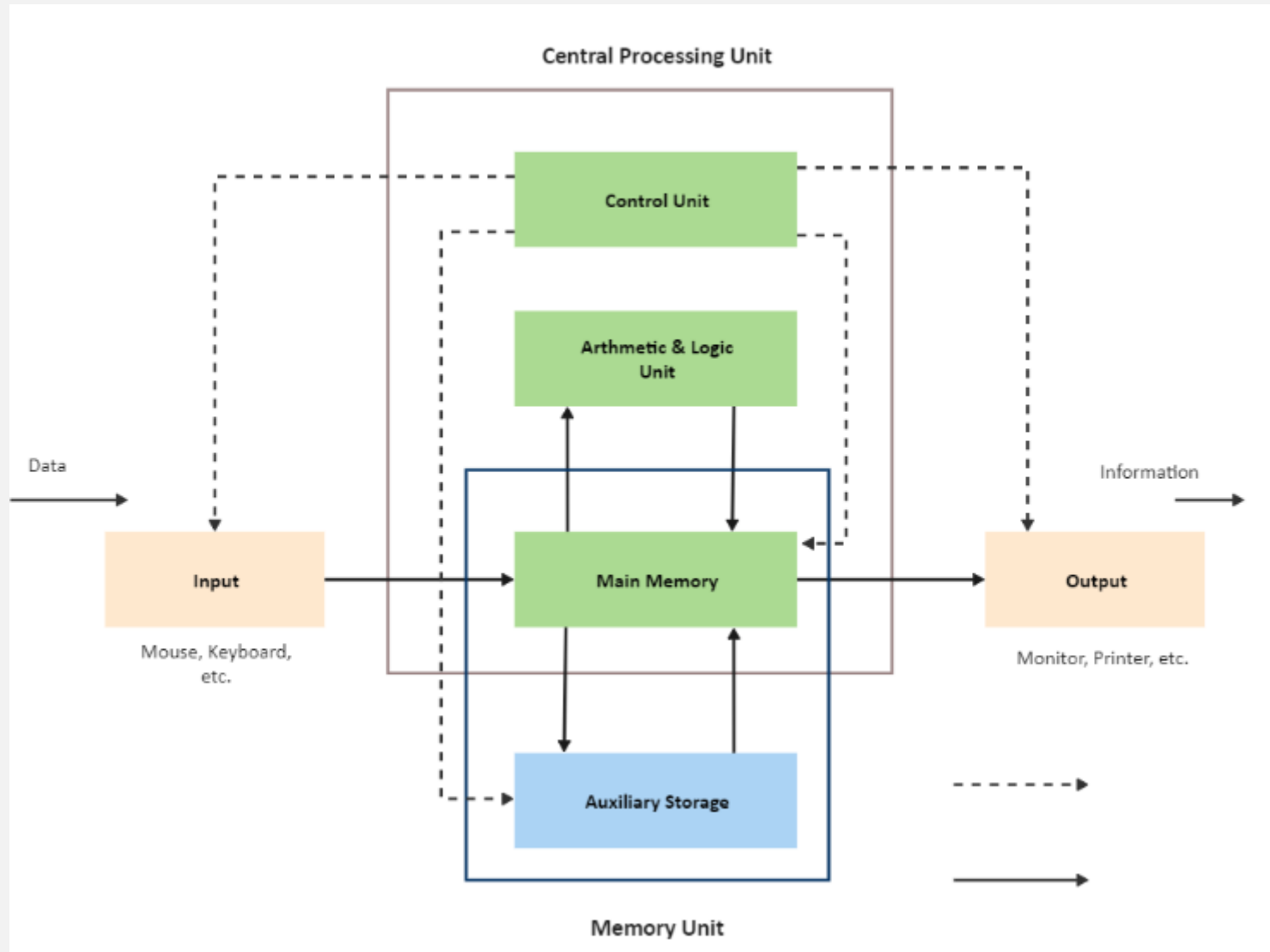


Diagrama a bloques de una computadora

La **organización de computadoras** se enfoca en cómo los componentes y subsistemas interactúan entre sí para realizar las operaciones especificadas en la arquitectura.

- **Unidad Central de Procesamiento (CPU):** Cómo están organizados y conectados los diferentes componentes de la CPU (ALU, registros, unidad de control, entre otros).

- **Memoria:** Jerarquía de memoria (caché, RAM, almacenamiento secundario) y cómo se accede a ella.

- **Interconexiones:** Buses y otros medios que permiten la comunicación entre los diferentes componentes del sistema.

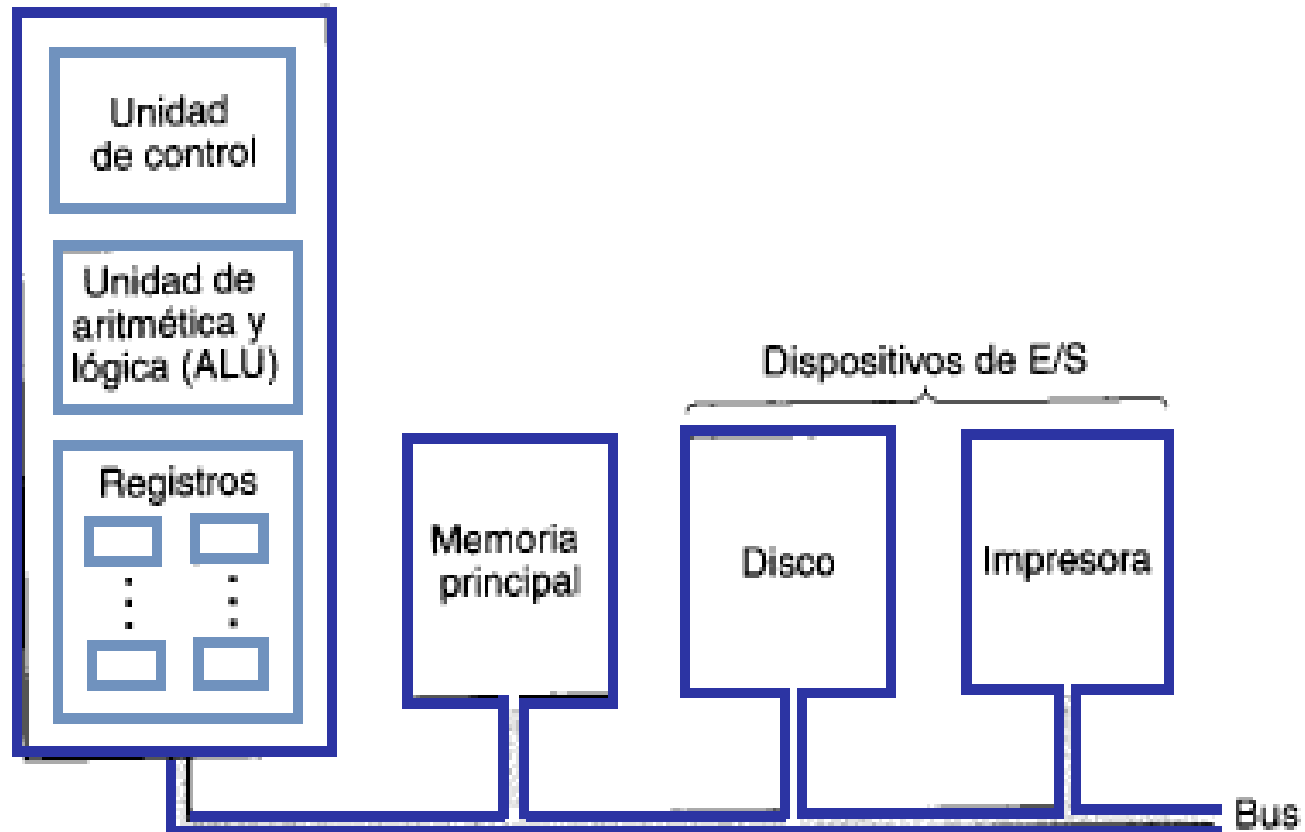
- **Sistemas de E/S:** Implementación física y lógica de los subsistemas que permiten la entrada y salida de datos.

- **Unidades de almacenamiento:** Cómo se gestionan y organizan los dispositivos de almacenamiento como discos duros y SSDs.



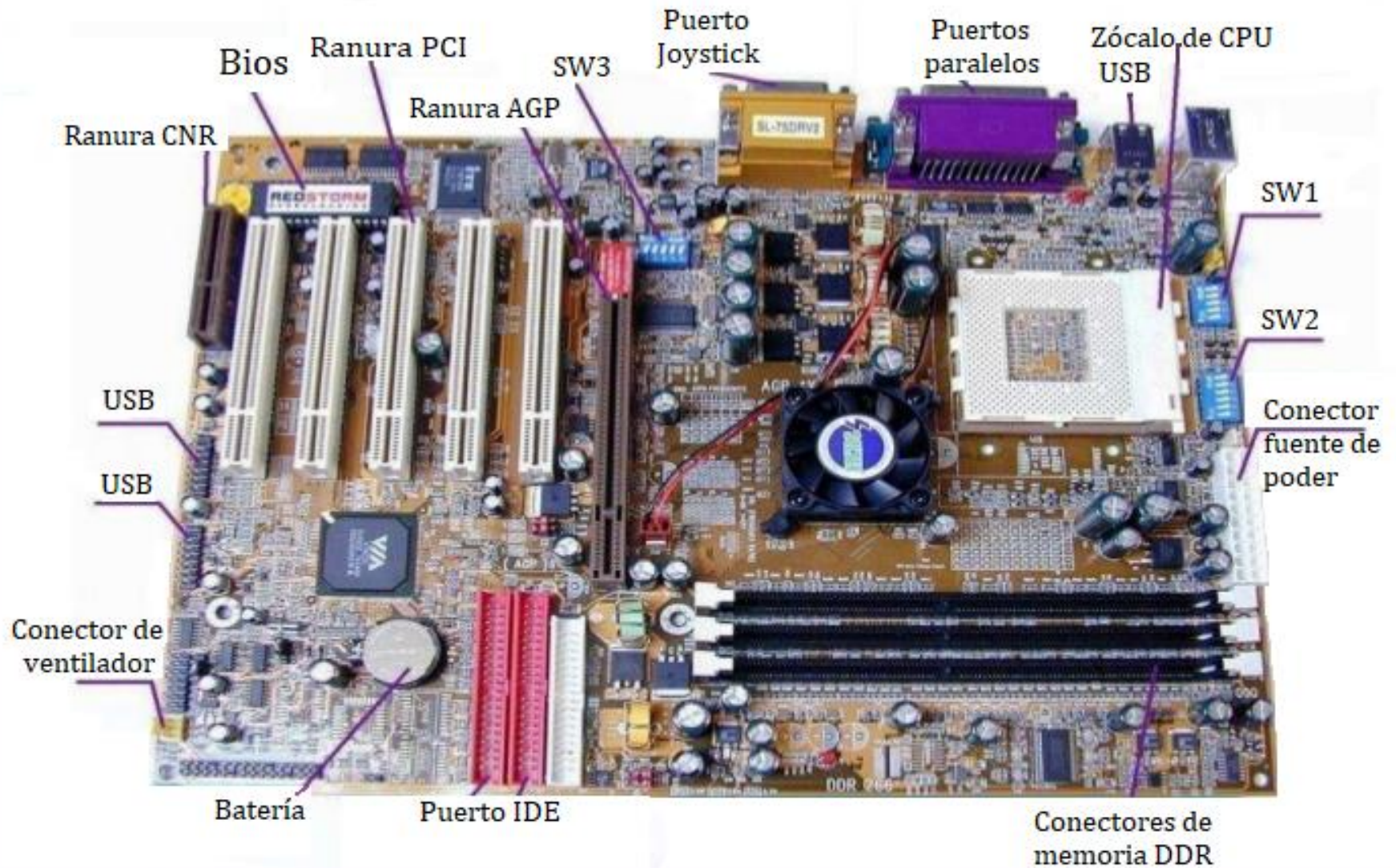
# Organización de una computadora sencilla

Unidad central de procesamiento (CPU)



En una computadora se ejecutan 3 operaciones básicas:  
Entrada de datos, proceso de datos y salida de datos.

## PLACA MADRE



Tarjeta Madre o Placa base

# Principales componentes de una computadora o sistema informático

## Unidad de Entrada

La unidad de entrada es la plataforma desde donde los datos primarios pasan al sistema informático. La entrada puede tener cualquier forma.

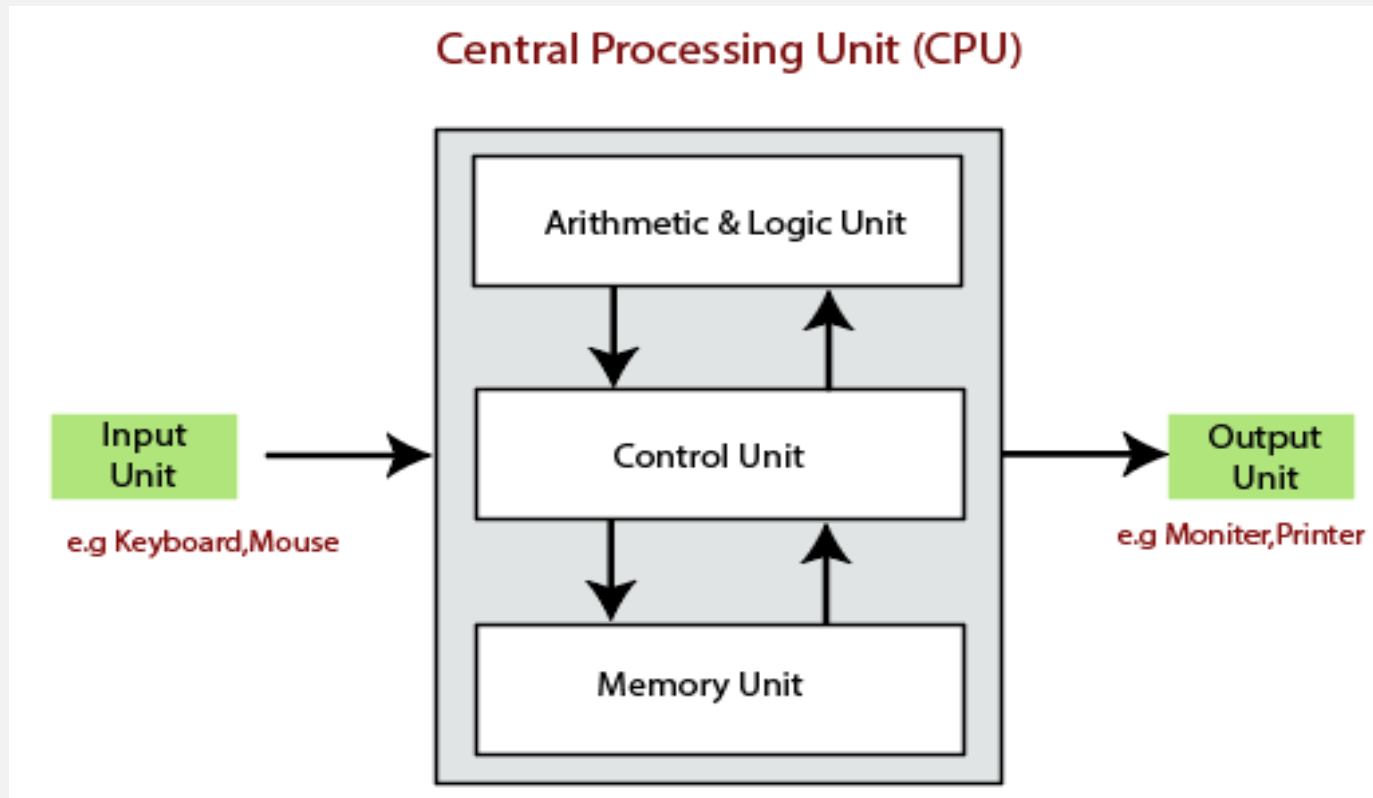
Todos los datos de entrada pasan de la unidad de entrada a la unidad de almacenamiento de la computadora.



# Principales componentes de una computadora o sistema informático

## Unidad Central de Procesamiento (CPU o Central Processing Unit).

La **Unidad Central de Procesamiento** consta de: la Unidad aritmética y lógica (ALU), la Unidad de control (CU) para procesar los datos y presentarlos como salida y la Unidad de memoria



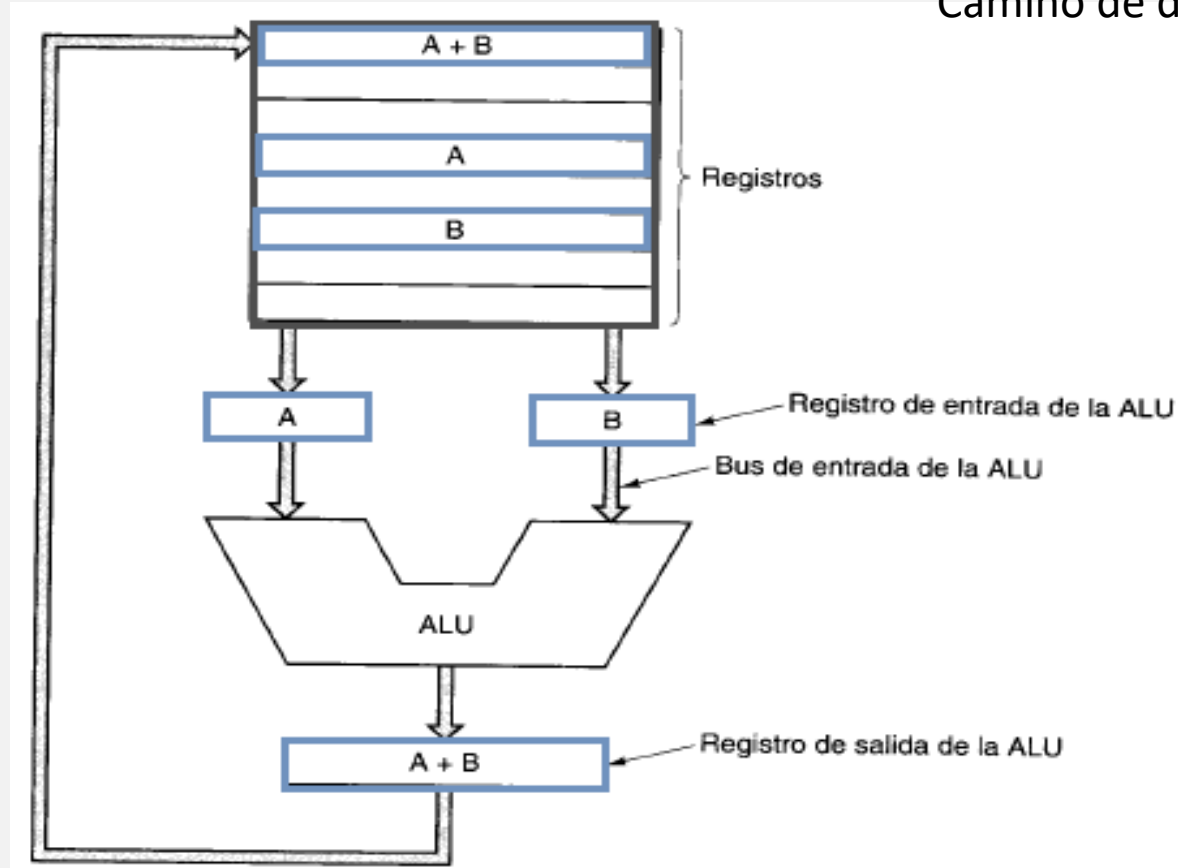


# Unidad de Aritmética y Lógica (ALU)

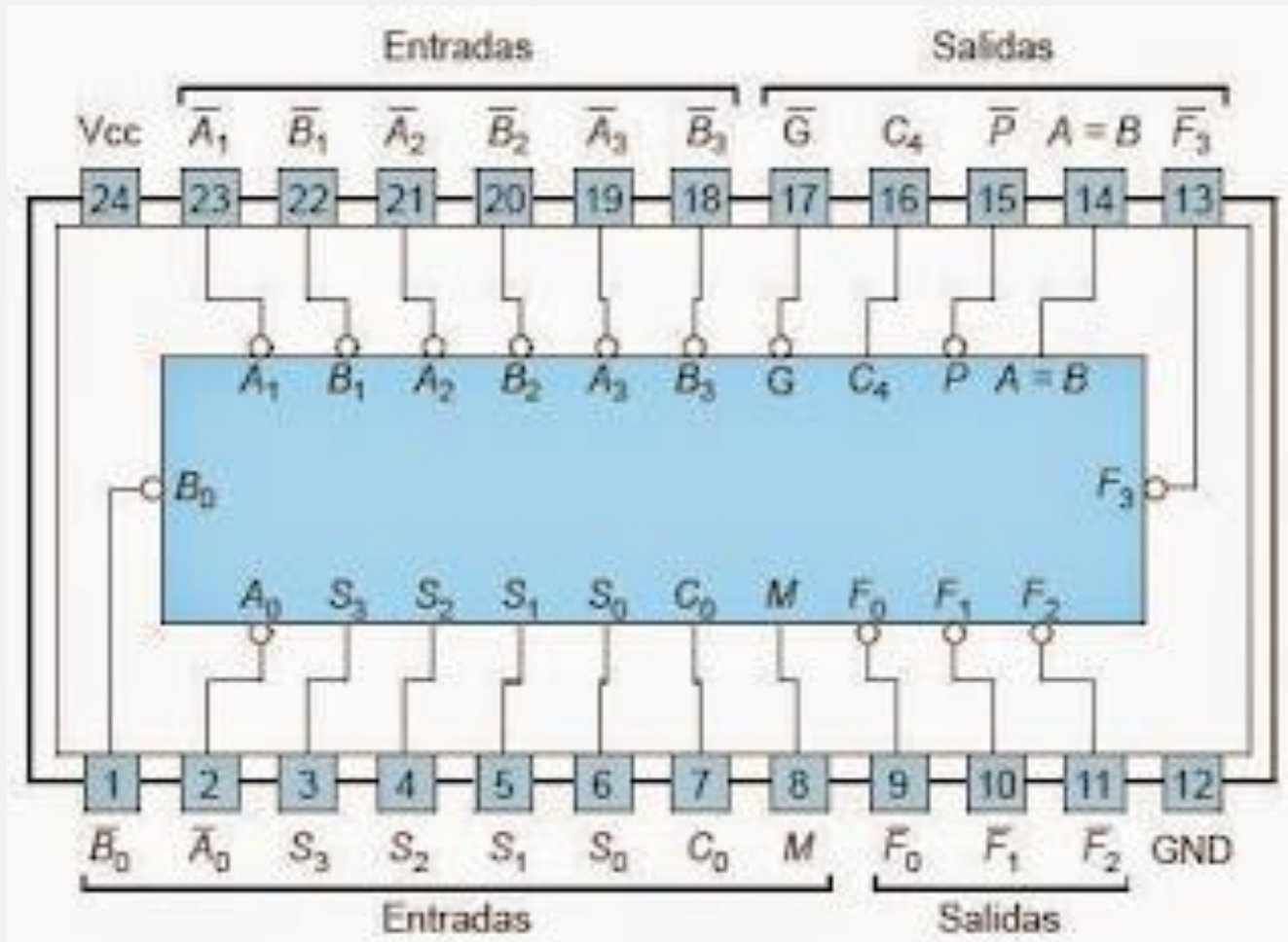
Conocida como ALU (*Arithmetic Logic Unit*) es un circuito digital encargado de procesar estos dígitos binarios y calcular operaciones aritméticas (suma, resta, multiplicación, etc) y lógicas (y, o, si, no) entre dos números entregando los resultados como salida asíncrona.

Camino de datos de una máquina típica

Instrucciones del tipo registro-registro



## Circuito integrado ALU



ALU 74LS181

La función de cada uno de los pines de este IC, son:

Pines 2, 23, 21 y 19, respectivamente /A0 ... /A3, son las entradas de uno de los operandos, activos en BAJO.

Pines 1, 22, 20 18, respectivamente / B0 ... /B3, entradas de los operandos restantes, activos en BAJO.

Pines 9, 10, 11 y 13, respectivamente /F0 ... /F3, son las salidas de la ALU, donde se tendrán los resultados de las entradas.

Pin 8 M , es una de las líneas de control; por medio de esta línea se le indica al circuito la operación a realizar, Si M=1 realiza operaciones lógicas y realiza operaciones aritméticas si M=0.

Pin 7 Cn, es la entrada de acarreo esta entrada deberá ser 0 en operaciones aritméticas; en caso de ser un 1 habrá que sumarlo a la función aritmética que se seleccione.

Pin 14 A=B, es una salida de colector abierto e indica cuándo las cuatro salidas están a nivel ALTO. Si se selecciona la operación aritmética de la resta es salida se activará cuando ambos operandos sean iguales.

Pin 17 /G, salida de generación de acarreo. En operación aritmética de la suma, esta salida indica que la salida F es mayor o igual a 16, y en la resta F es menor que cero.

Pin 15 /P, salida de propagación de acarreo. En la operación aritmética de la suma, esta salida indica que F es mayor o igual a 15 en la resta que F es menor que cero..

Las salidas /G y /P se utilizan para acoplar varios circuitos integrados del tipo 74181 en cascada empleando el método de propagación en paralelo.

Pin 16 Cn+4 es la salida de acarreo.

Pines 6, 5, 4 y 3, respectivamente So ... S3, son las líneas de control del circuito; mediante éstas se selecciona la función que ha de realizar el circuito.

Pin 24 = VCC

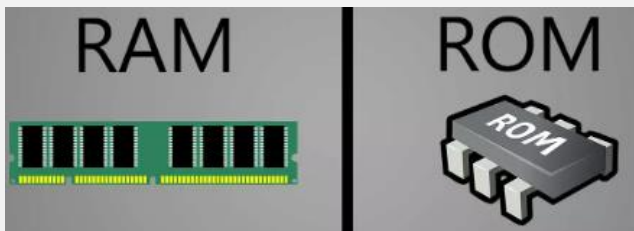
Pin 12 = GND

## Unidad de Almacenamiento

Los datos brutos o primarios que proviene de la unidad de entrada se guardan en la Unidad de Almacenamiento. Este es el lugar donde se almacenan los datos a procesar y los datos procesados.

La Unidad de Almacenamiento se clasifica en dos partes: Unidad Primaria y Secundaria

### •Almacenamiento Primario



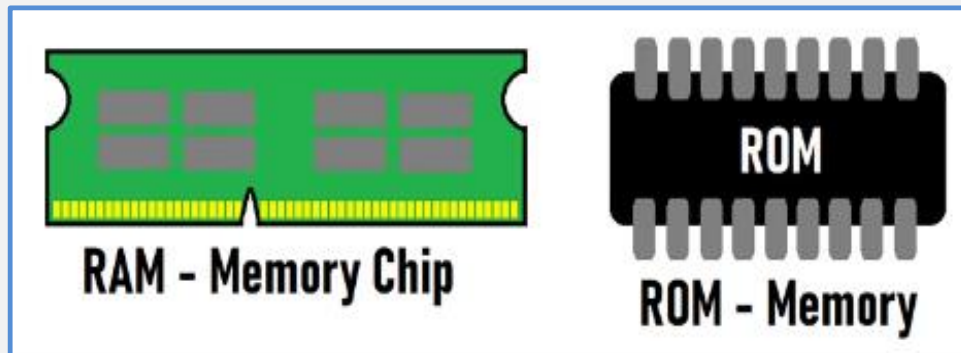
Se conoce como la memoria principal y contiene instrucciones, datos y programas que se están utilizando en un momento determinado.

Reside en la tarjeta madre y contiene la ROM y la RAM de la computadora.

# Unidad de Almacenamiento

## Memoria

- **RAM (Memoria de Acceso Aleatorio):** Almacena temporalmente los datos y las instrucciones que la CPU necesita en tiempo real. Es volátil, lo que significa que pierde su contenido al apagarse la computadora.
- **ROM (Memoria de Solo Lectura):** Almacena el firmware, que es el software fundamental para iniciar la computadora. No es volátil.



# Unidad de Almacenamiento

## •Almacenamiento Secundario



unidad de disco duro (HDD)



unidad de estado sólido (SSD)

Es un dispositivo de almacenamiento de datos no volátil y permanente.

Es el lugar donde se almacenan los datos durante un tiempo corto o largo.

El almacenamiento secundario apoya al almacenamiento primario y son dispositivos conocidos como disco duro de la computadora.

Se utiliza principalmente como dispositivo de copia de seguridad.

# Memoria de programa

Es tipo de memoria almacena las instrucciones que el procesador debe ejecutar.

Estas instrucciones conforman el programa que controla el funcionamiento del sistema.

Existen diferentes tipos de memorias de programa, como la ROM donde se almacena código que no se puede modificar, o la memoria flash, que permite reprogramar y actualizar el contenido.

La memoria de programa es fundamental en sistemas operativos, microcontroladores y otros dispositivos, ya que define el comportamiento y las funciones que ejecutará el hardware.



# Memoria de datos

La memoria de datos se utiliza para almacenar temporalmente la información con la que el procesador está trabajando.

La memoria de datos guarda los valores que manipulan las instrucciones, como variables, resultados intermedios y otros datos necesarios durante la ejecución de un programa.

La memoria de datos es generalmente volátil, lo que significa que su contenido se pierde cuando se apaga el sistema.

Un ejemplo típico es la RAM que es el espacio donde los programas cargan y almacenan datos mientras están en funcionamiento.

Este tipo de memoria es esencial para el procesamiento dinámico y eficiente de la información en un sistema.

# Registros

Los registros son pequeñas unidades de almacenamiento ubicadas dentro del procesador (CPU) que permiten guardar datos temporalmente para su procesamiento inmediato.

Son mucho más rápidos que la memoria RAM y son utilizados por el procesador para realizar operaciones de manera eficiente.

Los registros almacenan datos, direcciones de memoria o resultados intermedios durante la ejecución de un programa.

# Registros

Existen varios tipos de registros, entre ellos:

**Registros de datos:** Guardan valores numéricos o datos que se están procesando.

**Registros de direcciones:** Almacenan direcciones de memoria, es decir, dónde se encuentran los datos o instrucciones en la memoria principal.

# Registros

**Registros de control:** Contienen información de control o del estado del procesador para supervisar y coordinar el flujo de ejecución de instrucciones.

**Registros de propósito general:** Son usados por el procesador para diversas operaciones aritméticas o lógicas según las necesidades del programa.

La cantidad y el tamaño de los registros varían según la arquitectura del procesador, éstos permiten un acceso rápido a la información necesaria para ejecutar instrucciones.

## Unidad de Control (UC-CU)

Es un circuito digital dentro del procesador que dirige las operaciones y maneja las señales de control que entran y salen del CPU.

Esta unidad detecta cuándo el procesador central del CPU necesita datos y cuándo no. Si los datos son necesarios, los recupera de la Unidad de Almacenamiento y los transfiere al CPU. La CU convierte los datos en señales y los pasa al procesador central.

y determinar que tipo de instrucción. Así también, distribuye cada uno de los procesos al área correspondiente para ejecución.

# Instrucciones

El campo de operación de una instrucción especifica la acción que debe realizarse. Esta acción se ejecuta sobre los operandos, es decir, sobre datos almacenados en los registros de la computadora o en palabras de memoria.

Ejemplo:

**ADD** R3, R2, R1

El **campo de operación** es **ADD** y especifica la operación a realizar.

Los **operandos** **R3**, **R2** y **R1** son los datos sobre los cuales se realiza la operación suma. Los valores almacenados en los registros **R2** y **R1** se suman y el resultado se guarda en **R3**.

## ***Buses de interconexión***

Los **buses** son componentes clave en la arquitectura de las computadoras, permiten la transferencia de datos, instrucciones y señales de control entre los distintos componentes del sistema, como la CPU, la memoria, los dispositivos de almacenamiento y periféricos.

Las configuraciones de cómo los componentes de hardware se comunican entre sí es por medio del:

- Bus único

Todos los dispositivos y componentes están conectados a un solo bus compartido, para transmitir datos, direcciones, y señales de control. El costo de su uso es reducido y simple su implementación. Sin embargo, provoca cuellos de botella y reduce el rendimiento en su acceso simultáneo, así también produce mayor latencia.

Ejemplo:

En sistemas antiguos como en las primeras computadoras

## ***Buses de interconexión***

- Bus dedicado

Es una conexión directa entre dos componentes específicos o entre un componente y un grupo determinado de componentes. Esto significa que cada par de comunicación tiene su propio canal de datos. Esto produce mayor rendimiento, menor latencia y mayor ancho de banda para cada conexión. Pero conlleva mayor complejidad en el diseño de hardware, mayor costo y requiere de más espacio físico en la placa base lo que limita la escalabilidad.

Ejemplo:

**PCI Express (PCIe):** Aunque técnicamente es una red de buses punto a punto, cada conexión entre la CPU y una tarjeta PCIe (tarjeta gráfica) puede verse como un bus dedicado, ofreciendo un camino de datos directo y de alta velocidad.



## Modos de direccionamiento

Los modos de direccionamiento son técnicas utilizadas para especificar la ubicación de los operandos en las instrucciones de una computadora.

El modo de direccionamiento especifica como interpretar la información que contienen cada campo de operando y localizarlo de acuerdo a su tipo.

Estos modos de direccionamiento pueden clasificarse en **proprios** e **improprios**.

En los propios el operando se localiza en una dirección de memoria específica. En los improprios el operando puede localizarse en los registros del procesador.

# Modos de direccionamiento

Estos modos determinan cómo se accede a los datos que deben ser procesados por la CPU.

## Ejemplo:

### Direccionamiento Propios

- **Absoluto**
- **Indirecto**
- **Relativo**
- **Por base y desplazamiento**

# Modos de direccionamiento

Estos modos determinan cómo se accede a los datos que deben ser procesados por la CPU.

## Ejemplo:

### Direccionamiento impropios

- Inmediato
- Directo por registro
- Implícito
- De Pila

# Procesamiento de la información

- Por medio de la unidad de entrada, se introduce la información (datos o instrucciones) que es convertida a lenguaje máquina (para ser entendida por el procesador), la información se almacena en la unidad de memoria de la computadora.
- Desde la memoria, los datos pasan al CPU. Es aquí donde la unidad de control identifica y dirige qué y cómo se realizarán las operaciones en la ALU (se realizan todas las operaciones aritméticas-lógicas necesarias).
- Una vez procesada la información se almacenan en la memoria principal de la computadora y luego la memoria decidirá cuándo emitir estos datos hacia la unidad de salida.
- Todo este viaje de datos por todo el sistema informático está controlado por la Unidad de Control del CPU.

# Principales componentes de una computadora o sistema informático

## Unidad de Salida

Es a través de la cual la computadora proporciona la salida a los datos.  
La unidad de salida es siempre a través de hardware.



# Evolución de la Arquitectura de Computadoras

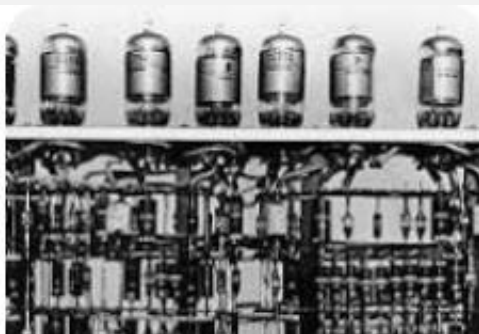
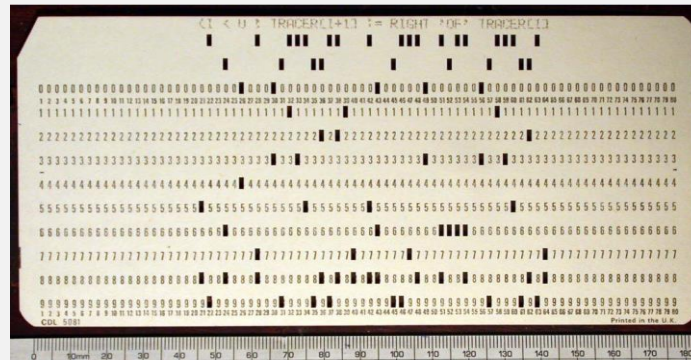
- **Primera Generación (1940-1956) Bulbos-Tubos al vacío**

Las primeras computadoras empleaban bulbos como circuitos y tambores magnéticos que permitían almacenar información (memoria de almacenamiento primario y posteriormente almacenamiento auxiliar). Eran grandes equipos electrónicos que ocupaban cuatros enteros.

Manejar estos equipos era de muy alto costo y el consumo de una gran cantidad de electricidad, generaban mucho calor la cual era la causa de mal funcionamiento.



Relé electromecánico usado en la MARK 1



# Evolución de la Arquitectura de Computadoras

- **Primera Generación (1940-1956) Bulbos-Tubos al vacío**

La primera generación de computadoras se basaba en lenguaje máquina para realizar operaciones y solo podían resolverse en un tiempo a la vez.

Ejemplos:



ENIAC



UNIVAC



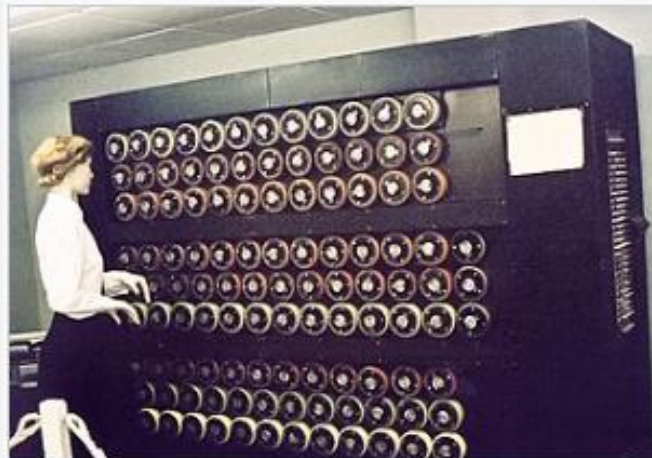
# Conceptos de algoritmo y computación

## Máquina de Turing

**Alan Mathison Turing (1912-1954)** es considerado uno de los padres de la ciencia de la computación y un precursor de la informática moderna.

Formalizó los conceptos de algoritmo y computación con la creación de la **máquina de Turing**, su construcción se denominó computadora de “propósito general”, lo que hoy en día se entiende como **computadora programable**.

Durante la Segunda Guerra Mundial, contribuyó significativamente al esfuerzo de guerra con el trabajo de descifrar los códigos nazis, especialmente los generados por la máquina **Enigma**.



El **Bombe** replicaba la acción de varias máquinas **Enigma**.



Después de la guerra, diseñó uno de los primeros computadores electrónicos programables digitales en el Laboratorio Nacional de Física del Reino Unido, conocido como el Pilot ACE (Automatic Computing Engine-Motor de computación automática) y la computadora de la Universidad de Mánchester llamada Manchester Mark 1. Ambas compartían algunas características de diseño de esa época, como:

Características	Pilot ACE	Manchester Mark 1 (más tarde conocida como Ferranti Mark 1)
Componentes electrónicos	Válvula de vacío (bulbos) para realizar operaciones lógicas	Válvula de vacío (bulbos) para realizar operaciones lógicas
Memoria	Memoria de línea de retardo de mercurio, un sistema en el que los bits de información se almacenaban como ondas acústicas en columnas de mercurio.	Además de la memoria de tubo de Williams-Kilburn, tenía memorias de línea de retardo para el almacenamiento secundario.
Arquitectura	La arquitectura era serial, procesando un bit a la vez, lo que lo hacía más lento pero también más simple en comparación con los diseños paralelos.	De von Neumann, aunque con algunas particularidades como la implementación de una unidad de control más sofisticada que permitía operaciones en paralelo dentro de un ciclo de instrucción.
Programación	Se programaba mediante instrucciones almacenadas en la memoria, una innovación clave del concepto de programas almacenados introducido por John von Neumann.	Utilizaba un código binario de instrucciones que se cargaba en la memoria antes de la ejecución.

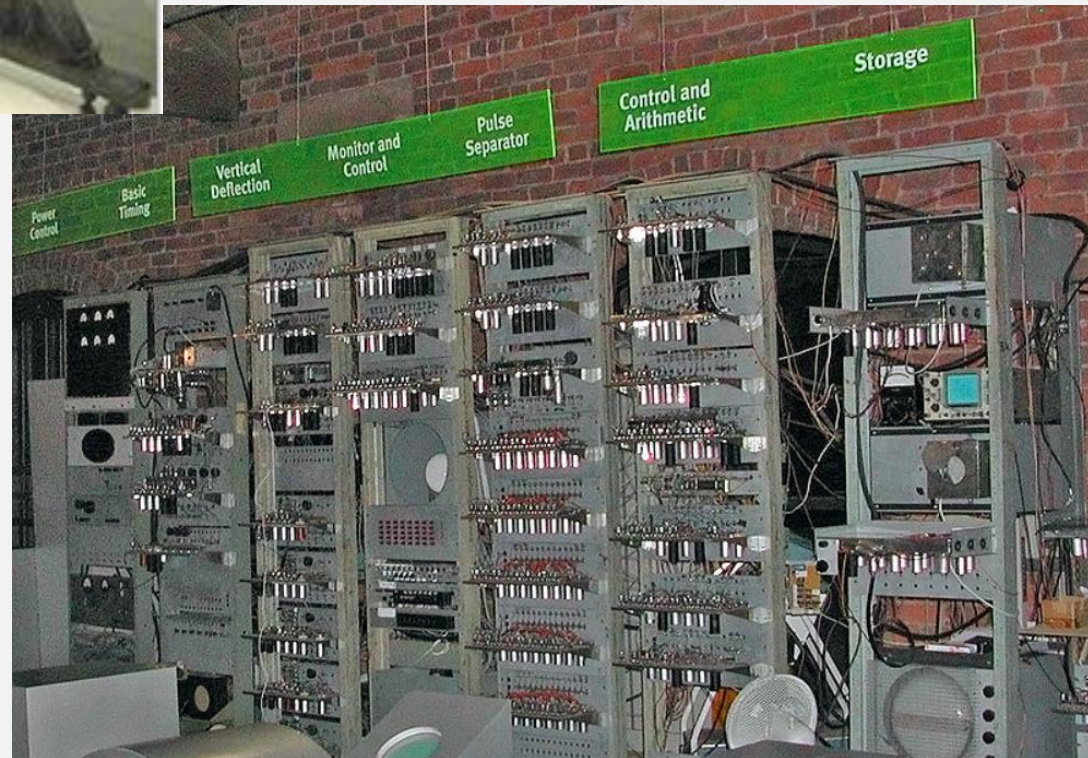
## Pilot ACE (1947)



Ambas computadoras fueron pioneros en el uso de programas almacenados, lo que marcó una revolución en la computación al permitir que las computadoras no solo realizaran cálculos sino que también fueran reprogramables sin cambiar su hardware físico.

## Manchester Mark I (1948)

La tecnología de válvulas de vacío hacía a estas máquinas grandes, caras y con consumo de energía elevado comparado con los estándares actuales.



En el ámbito de la inteligencia artificial, Turing es reconocido por haber ideado la prueba de Turing en 1950. Un criterio mediante el cual se podía evaluar la inteligencia de una máquina si sus respuestas durante la prueba son indistinguibles de las de un humano. También se creó el Abbreviated Code Instruction que dio origen a los lenguajes de programación.

Paralelamente a la construcción de ACE existía un proyecto similar en E.U. llamado EDVAC de von Neumann.

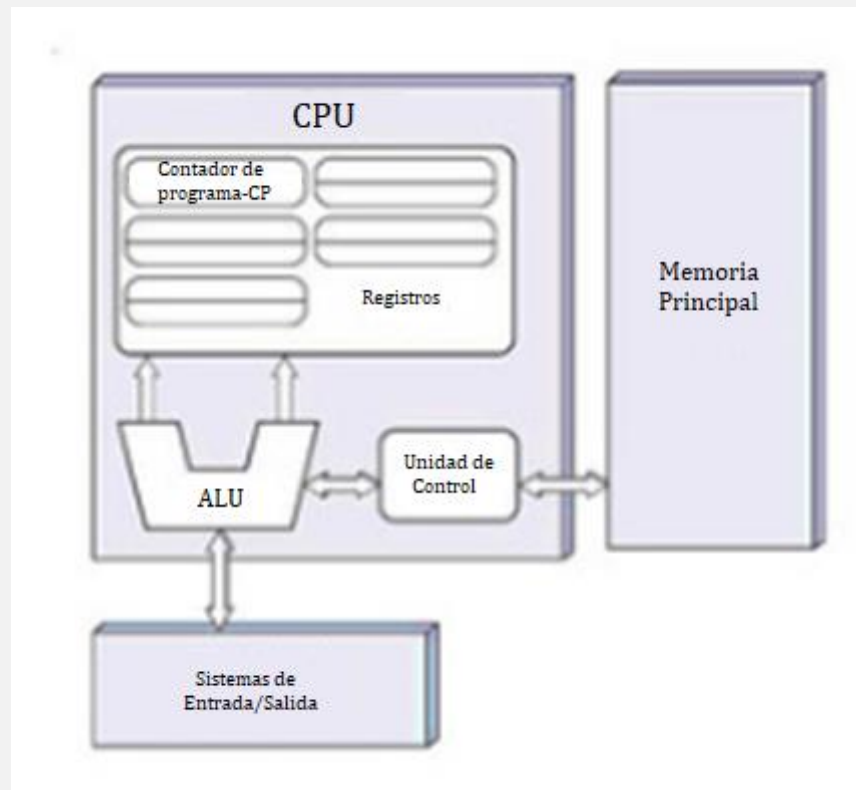


# Evolución de la Arquitectura de Computadoras

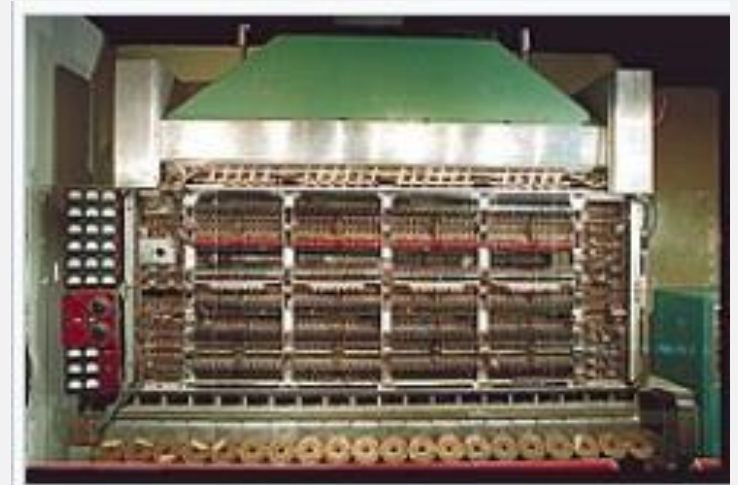
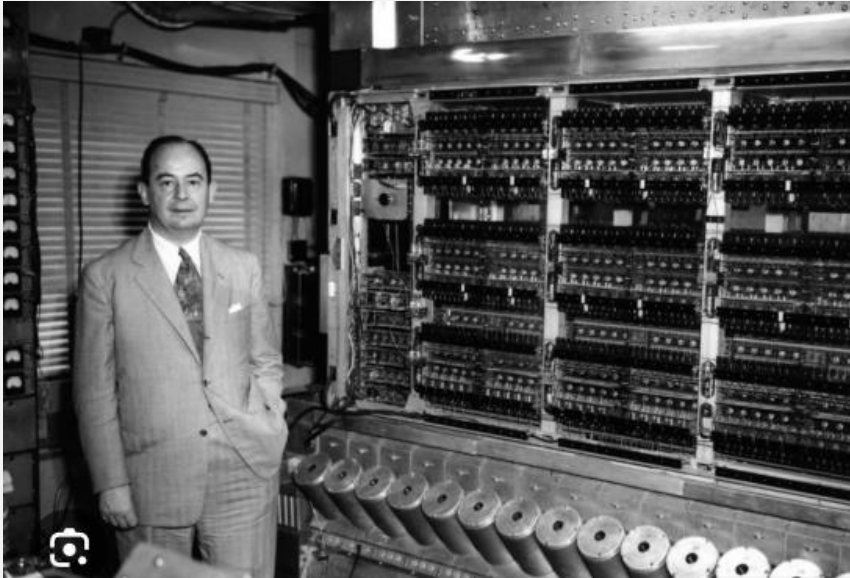
- **La Revolución digital: Computadoras de Von Neumann**

La propuesta de John von Neumann en 1945 sentó las bases de la arquitectura de computadoras modernas. Su diseño incluía una unidad de procesamiento, memoria y control, un esquema que todavía se utiliza en la mayoría de las computadoras hoy en día.

## Arquitectura J. von Neumann







La **IAS machine** fue la primera computadora digital construida (1942-1951) por el Instituto para el Estudio Avanzado (IAS, *Institute for Advanced Study*), en Princeton, NJ. EU.

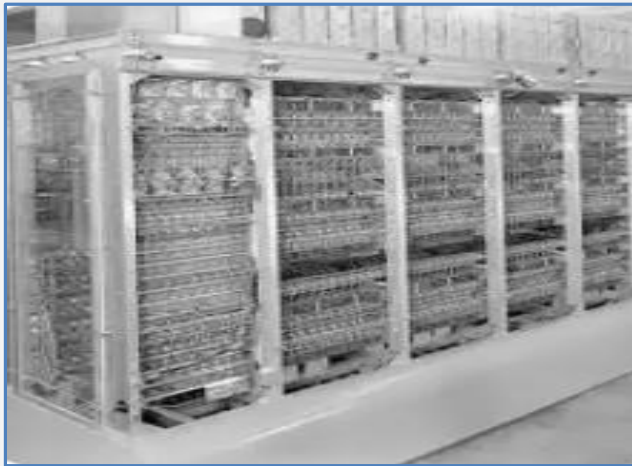
El artículo que describe el diseño del IAS machine fue editado por John von Neumann.

# Evolución de la Arquitectura de Computadoras

- **Segunda generación 1956-1963: Transistores**

El PDP-1 (Programmed Data Processor-procesador de datos programable) fue la primera computadora diseñada y construida por Digital Equipment, permitía el uso de un joystick y un lápiz óptico para juegos como Spacewar y Ratón en el laberinto.

En 1957 el matemático Von Neumann colaboró para la construcción de una computadora avanzada llamada MANIAC (Mathematical Analyser).



MANIAC I estaba basada en la arquitectura de von Neumann del IAS, desarrollada por John von Neumann.



La primera computadora totalmente de transistores fue la TRADIC, de los laboratorios Bell.

La IBM TX-0 (1958) tenía un monitor de vídeo relativamente pequeño y con dispositivo de salida audio.

# Evolución de la Arquitectura de Computadoras

- **Segunda generación 1956-1963: Transistores**

Los transistores (dispositivos de material semiconductor que amplifica la señal, abre/cierra un circuito) reemplazaron a los tubos de vacío.

En la actualidad, los procesadores contienen millones de transistores nanoscópicos.

Ejemplos:



IBM 1401



Burroughs B7800

- Tercera generación (1964-1971): Circuitos Integrados

Circuitos integrados

- minicomputadoras y mainframes
- herramientas esenciales para empresas y gobiernos
- Incremento en la velocidad y eficiencia de las computadoras
- Aún eran grandes y costosas
- Proporcionaban potencia de procesamiento

- Se creó el concepto de memoria virtual, multiprogramación y sistemas operacionales complejos.
- En la década de 1960 se acuñó el termino de **Software**



IBM 360



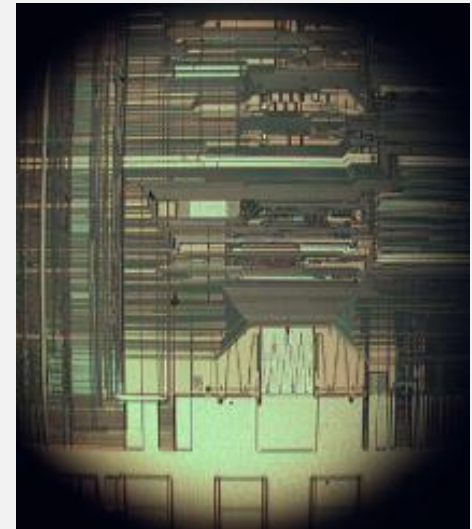
- **Cuarta generación (1971-1988)**

## **La Revolución Personal: Computadoras Personales**

La década de 1970 marcó el surgimiento de las minicomputadoras o computadoras personales, como la icónica IBM PC.

Estas máquinas permitieron que las personas tuvieran acceso a la tecnología de la información en sus hogares y cambió la forma en que vivimos y trabajamos.

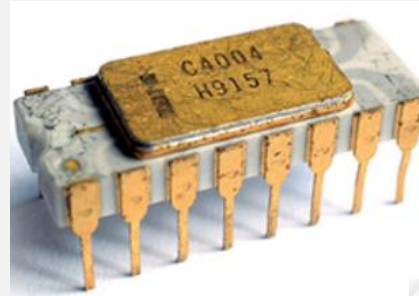
Esta generación surge con los circuitos integrados VLSI (Very large Scale Integrated) a muy gran escala.



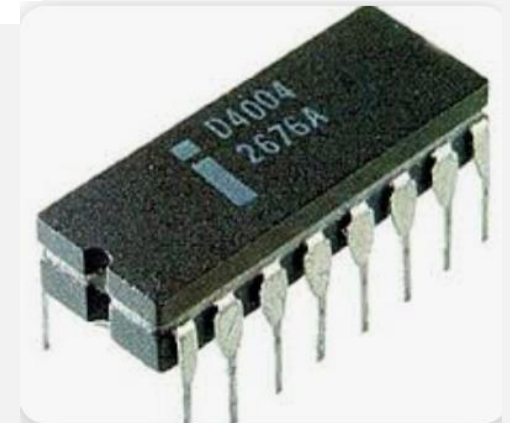
- Cuarta generación (1971-1988)

## La Revolución Personal: Computadoras Personales

Intel introduce en el mercado el circuito integrado: *Microprocesador (el 4004, de cuatro bits).*



El primer software creado por Paul Allen y Bill Gates es Microsoft, una adaptación de Basic para el intérprete Altair Basic que se ejecutaba en un microprocesador Altair 8800 de MITS y después de computadoras Bus S-100.



- **Cuarta generación (1971-1988)**

## **La Revolución Personal: Computadoras Personales**

Surgieron muchos fabricantes de computadoras personales que utilizaron diferentes arquitecturas, mismas que compitieron en el mercado.

Esta generación se caracterizó por grandes avances tecnológicos realizados en un tiempo muy corto.

Los sistemas operativos alcanzaron un notable desarrollo, sobre todo por la posibilidad de generar gráficos (interfaz gráfica) a grandes velocidades para el uso de ventanas, iconos, menús desplegables para generar una comunicación amigable entre usuario y el sistema.



# Tecnología Mexicana

## *Computadora Heterárquica de Procesamiento Paralelo- AHR*

Proyecto en la UNAM (1979-1982) dirigido por Adolfo Guzmán Arenas. Esta computadora era capaz de alojar desde 5 hasta 64 procesadores Z-80 trabajando simultáneamente (paralelismo) en la ejecución de un programa escrito en el lenguaje LISP.

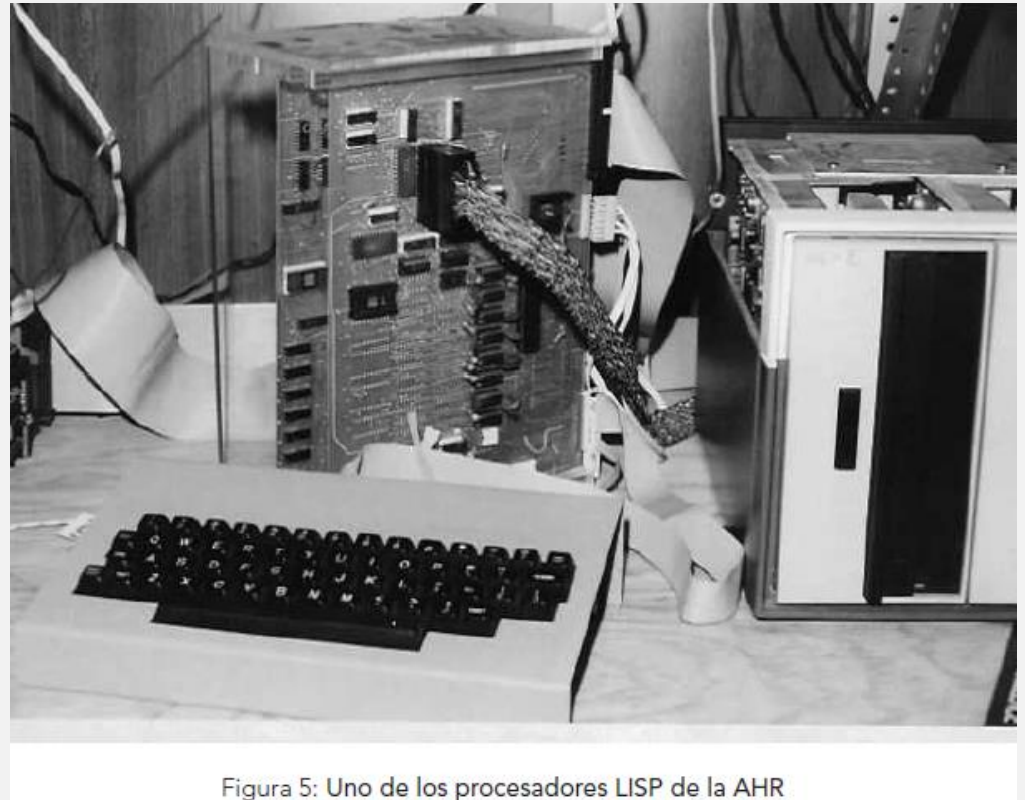
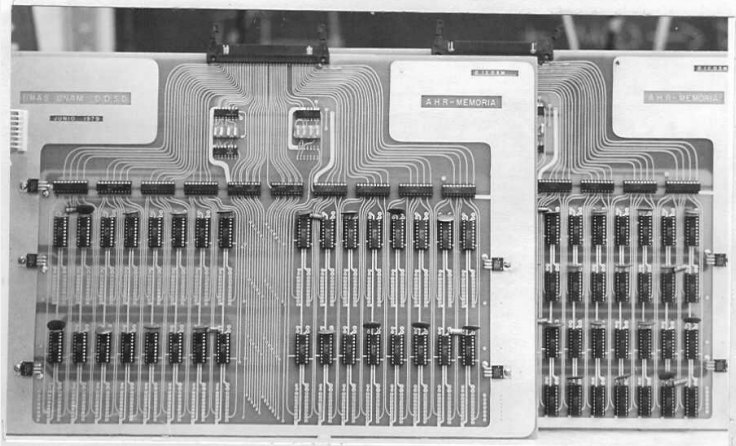
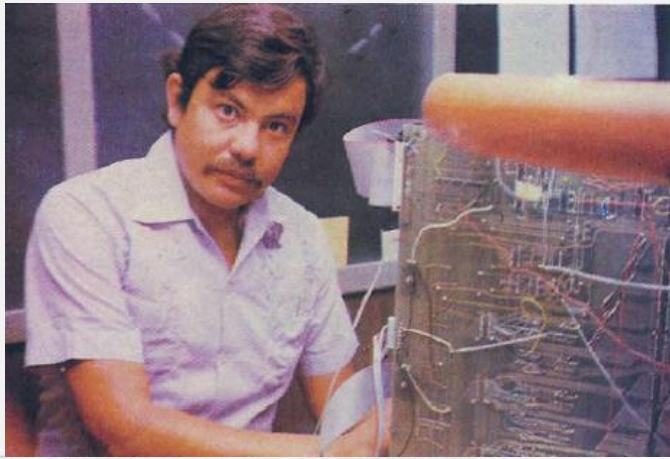


Figura 5: Uno de los procesadores LISP de la AHR



# Tecnología Mexicana

## *La Computadora IPN E-16*

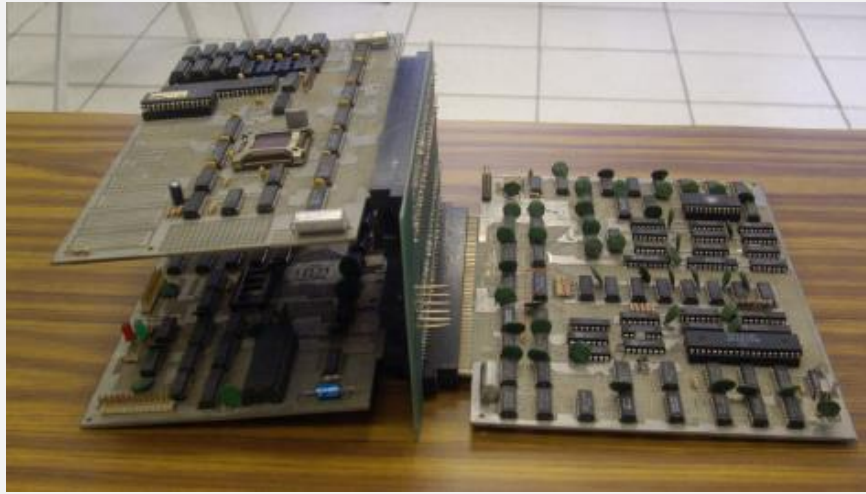
En Agosto de 1984, se completó en el Centro de Investigación Tecnológica en Computación del Instituto Politécnico Nacional (CINTEC-IPN), el primer prototipo de la computadora Almita II, siendo su principal diseñador el doctor Miguel Lindig Bos.

El procesador central de esta computadora era un Intel 80186 de 16 bits el cual operaba a una velocidad de 8 MHz. Ese procesador era representativo del estado de la tecnología en aquella época. El diseño de Almita II fue un notable logro de ingeniería mexicana pues se llegó a comprobar experimentalmente que su velocidad de procesamiento era hasta 3.4 veces más alta que la de la primera computadora personal que IBM lanzara al mercado



# Tecnología Mexicana

## *La Computadora IPN E-16*



Vista lateral de las tarjetas de la IPN-E16



Tabla 1. Resumen de las características de las computadoras mexicanas discutidas en este artículo

	<b>AHR</b>	<b>SMU-BUAP</b>	<b>Turing-850</b>	<b>IPN E-16</b>
<b>Sistema Operativo</b>	<i>Ninguno</i>	<i>CP/M</i>	<i>CP/M</i>	<i>MS-DOS</i>
<b>Software escrito</b>	<i>Intérprete de LISP</i>	<i>Modificaciones a CP/M y compilador Fortran</i>	<i>BIOS escrito en ensamblador Z80</i>	<i>Genérico</i>
<b>Procesador central</b>	<i>Z80@6MHz</i>	<i>NEC-V20 @8MHz</i>	<i>Z-80@6MHz</i>	<i>80188@8MHz</i>
<b># de uprocesadores</b>	<i>Hasta 64</i>	<i>1</i>	<i>2</i>	<i>1</i>
<b>Memoria RAM</b>	<i>256KB</i>	<i>256KB</i>	<i>256KB</i>	<i>512KB</i>
<b># de usuarios</b>	<i>1</i>	<i>10</i>	<i>1</i>	<i>1</i>
<b>Gabinete</b>	<i>Ninguno</i>	<i>Metálico</i>	<i>Metálico</i>	<i>Metálico</i>
<b>Computadoras construidas</b>	<i>1</i>	<i>&lt;5</i>	<i>1</i>	<i>1189</i>
<b>Desarrollado durante los años</b>	<i>1979-1982</i>	<i>1979-1983</i>	<i>1982-1984</i>	<i>1984-1987</i>

## **El Auge de las Computadoras portátiles**

Con el tiempo, las computadoras personales se volvieron más pequeñas y portátiles. Las laptops se convirtieron en un elemento común en nuestras vidas, brindando la capacidad de trabajar y comunicarse en cualquier lugar.

## **La Era de la Computación Móvil**

La llegada de los smartphones y las tabletas a principios del siglo XXI cambió la forma en que se interactúa con la tecnología. Estos dispositivos compactos, impulsados por potentes microprocesadores, se han convertido en extensiones del ser humano.

## **Hacia lo Desconocido: Computadoras Cuánticas**

En la actualidad, se está en la cúspide de una nueva revolución: la computación cuántica. Estas computadoras aprovechan los principios de la mecánica cuántica para realizar cálculos exponencialmente más rápido que las computadoras clásicas. Un emocionante salto hacia el futuro.



Los microprocesadores han transformado el diseño electrónico moderno, permitiendo que los dispositivos sean cada vez más pequeños e invadiendo prácticamente todo lo imaginable

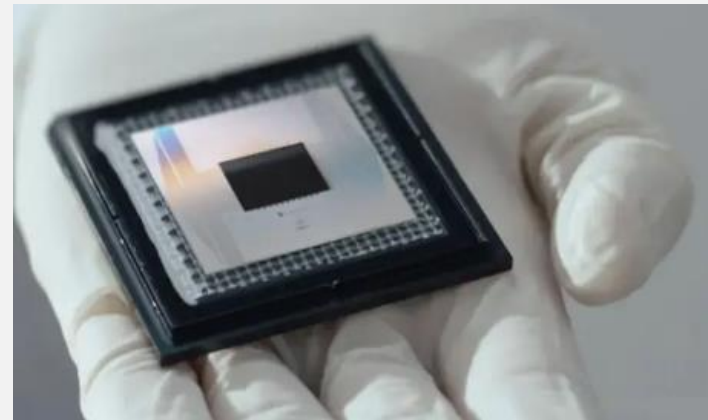
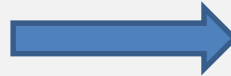
- Teléfonos inteligentes
  - Electrodomésticos
  - Cepillos de dientes eléctricos
  - Ascensores
  - Sistemas de seguridad
  - Automóviles
  - Bolsas de aire
  - Equipos agrícolas
  - Televisores
  - Procesadores de tarjetas de crédito
  - Surtidores de combustible
- 
- 10.000 nanómetros de ancho del circuito del microprocesador Intel 4004 original
  - 100.000 nanómetros de ancho promedio de un cabello humano
  - 10 nanómetros de ancho del circuito de los microprocesadores Intel más pequeños de la actualidad

# Hacia lo Desconocido: Computadoras Cuánticas

JiuZhang3 → Computadora cuántica experimental (calcula un millón de veces más rápido que su predecesor).

Sycamore → Computadora cuántica de Google (resuelve en 200 segundos una tarea que una supercomputadora convencional lo haría en 10 mil años en realizar).

Chip cuántico experimental  
llamado "Willow" de Google



# Arquitectura Von Neumann

La arquitectura de Von Neumann es un diseño conceptual de computadoras que se basa en la idea de un programa almacenado y que se utiliza como base para las computadoras modernas.

La base de la arquitectura Von Neumann es el concepto de **memoria compartida** donde el programa, instrucciones y los datos son almacenados y la CPU ejecuta secuencialmente las instrucciones y realiza el acceso a los mismos a través de un único bus de comunicación.

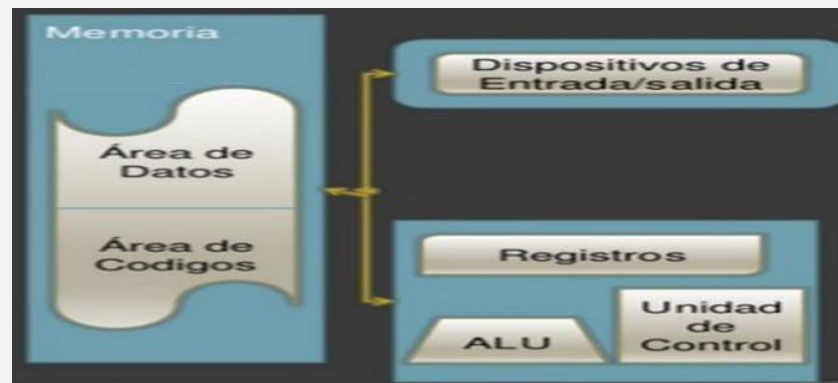
Este modelo usa una **única unidad de procesamiento** para manejar tanto las instrucciones como los datos.

## Características

- Memoria unificada para datos e instrucciones.
- Un solo bus de datos entre la CPU y la memoria.

### Ejemplo

Computadoras personales y diversos sistemas en general.



## *Características principales de la arquitectura*

**Unidad Central de Procesamiento (CPU):** Es el "cerebro" de la computadora, donde se realizan las operaciones lógicas y aritméticas.

**Memoria:** Almacena tanto los datos como las instrucciones del programa. En la arquitectura de Von Neumann, la memoria no distingue entre datos e instrucciones; ambos se almacenan en la misma memoria y se acceden de manera similar.

**Unidad de Control:** Interpreta las instrucciones almacenadas en la memoria y coordina las operaciones entre la CPU y la memoria.

**Unidades de Entrada/Salida (I/O):** Permiten la interacción entre la computadora y el exterior, como teclados, monitores, discos duros, etc.

**Buses:** Son las vías por las cuales se comunican los diferentes componentes del sistema. Hay buses para datos, direcciones y control.

## Arquitectura Von Neumann

Simplicidad y capacidad de ser programada (las instrucciones del programa puedan modificarse mientras la computadora está en funcionamiento).

Presenta un inconveniente conocido como el "cuello de botella de Von Neumann" (el acceso a la memoria por parte de la CPU para buscar tanto datos como instrucciones puede limitar el rendimiento de la computadora)

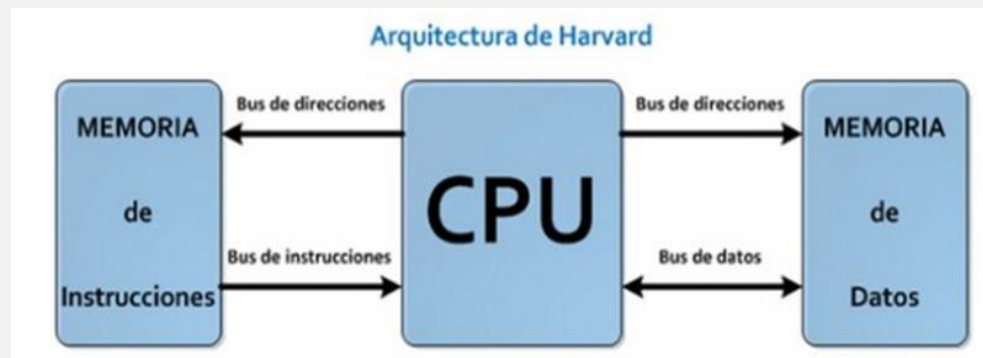
## Arquitectura de Harvard

Es un diseño de computadora donde las instrucciones y los datos se almacenan en memorias físicamente separadas.

La unidad central de procesamiento de este modelo emplea microcontroladores PIC, conectada a través de dos buses distintos hacia dos memorias una que contiene las instrucciones y la otra los datos.

Las instrucciones del programa se almacenan en una memoria de solo lectura (ROM), mientras que los datos se almacenan en una memoria de lectura/escritura (RAM). Esto permite el acceso simultáneo a instrucciones y datos, lo que aumenta la velocidad de procesamiento.

Los buses para las instrucciones y los datos, permite que las operaciones de fetch (obtener) de instrucciones y de datos puedan ocurrir en paralelo, mejorando el rendimiento.



## ***Arquitectura RISC (Reduced Instruction Set Computer)***

### ***Computadora de conjunto de instrucciones reducido***

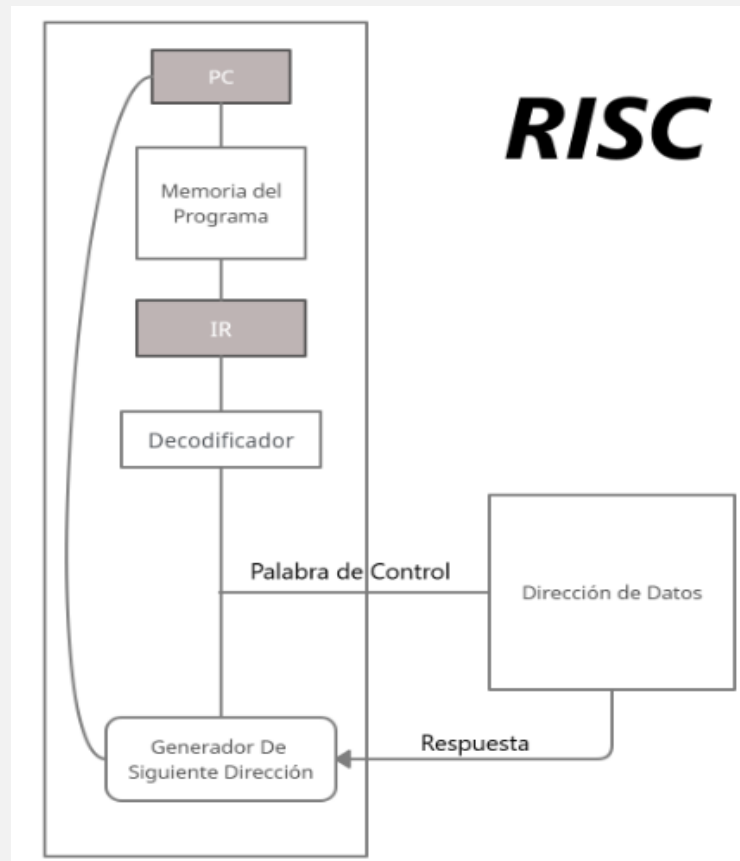
La arquitectura **RISC** utiliza un número reducido de instrucciones básicas, que son más simples y se pueden ejecutar en un solo ciclo de reloj.

La mayoría de las arquitecturas RISC usan instrucciones de longitud fija, lo que simplifica el decodificador de instrucciones y acelera el proceso de decodificación. El diseño es registro-registro y pocos modos de direccionamiento.

La RISC tiende a tener una mayor cantidad de registros internos para almacenar datos temporales, lo que reduce el número de accesos a la memoria principal, que es más lento que el acceso a los registros.

En RISC, las operaciones de carga y almacenamiento de datos desde/hacia la memoria son instrucciones separadas y explícitas, en lugar de estar integradas en otras operaciones como en CISC.

La simplicidad de las instrucciones facilita el pipelining, donde múltiples instrucciones pueden estar en diferentes etapas de ejecución simultáneamente, mejorando la eficiencia.





## Ejemplos de Arquitecturas RISC

ARM (usado en muchos dispositivos móviles y microcontroladores)

MIPS (utilizado en ciertos sistemas embebidos y en educación)

RISC-V (una arquitectura de código abierto que está ganando popularidad)

SPARC (utilizada por Sun Microsystems en sus estaciones de trabajo y servidores)

## ***Arquitectura CISC (Complex Instruction Set Computer)***

Se caracteriza por un conjunto de instrucciones más complejo y versátil comparado con RISC.

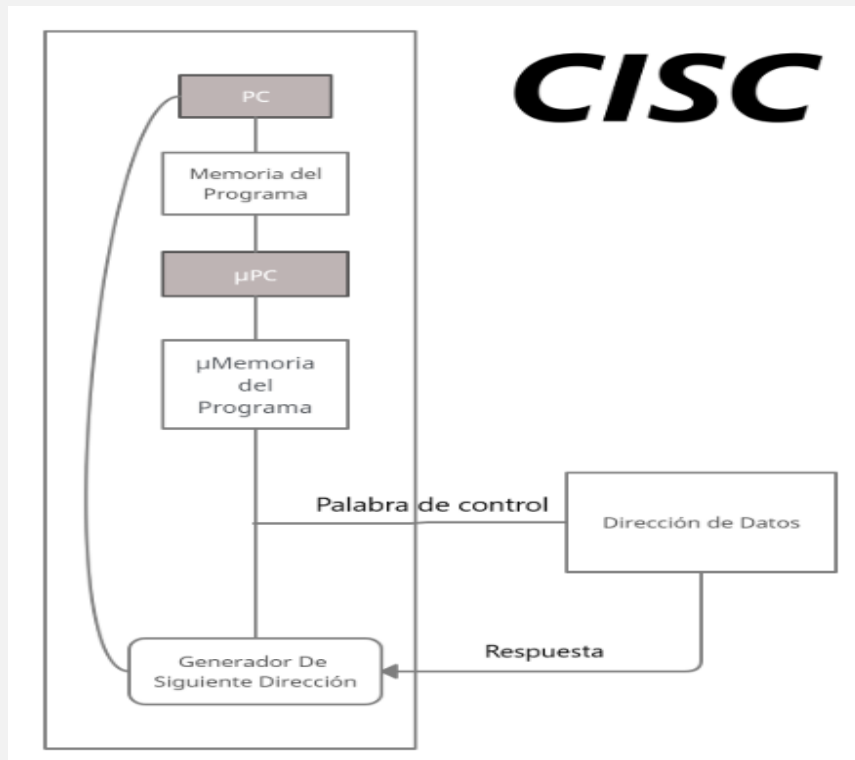
### **Conjunto de Instrucciones Complejo**

La CISC tiene un conjunto de instrucciones amplio y complejo, algunas de las cuales pueden requerir varios ciclos de reloj para completarse. Puede realizar múltiples operaciones de bajo nivel en una sola instrucción. Esto incluye operaciones que pueden combinar carga, cálculo y almacenamiento en una sola instrucción.

Las instrucciones en CISC pueden tener longitudes variables, lo que complica el decodificador y puede hacer que el pipeline sea menos eficiente, pero permite una mayor flexibilidad en la programación.

Las arquitecturas CISC tienen menos registros disponibles en comparación con RISC, lo que puede llevar a más accesos a la memoria para operaciones que requieren datos temporales.

Muchas instrucciones CISC son implementadas mediante microcódigo, que es una capa intermedia de instrucciones más simples que el procesador ejecuta internamente. Esto permite que las instrucciones complejas se descompongan en operaciones más simples.



El número de instrucciones por programa se puede reducir incorporando el número de operaciones en una sola instrucción, haciendo así las instrucciones más complejas

## **Ejemplos de Arquitecturas CISC**

x86, que es ampliamente utilizado en computadoras personales y servidores.

VAX de Digital Equipment Corporation.

PDP-11, también de DEC, que influyó en el diseño de sistemas operativos como Unix.

# Arquitectura Pipeline

El **modelo Pipeline** se denomina en el mundo de la informática a una serie de elementos de procesamiento de datos ordenados de tal modo que la salida de cada uno es la entrada del siguiente.

Se refiere a un diseño en el que varias instrucciones pueden estar en diferentes etapas de ejecución al mismo tiempo. Esto permite una ejecución más eficiente de las instrucciones.

## Características

- **Segmentación de las instrucciones**

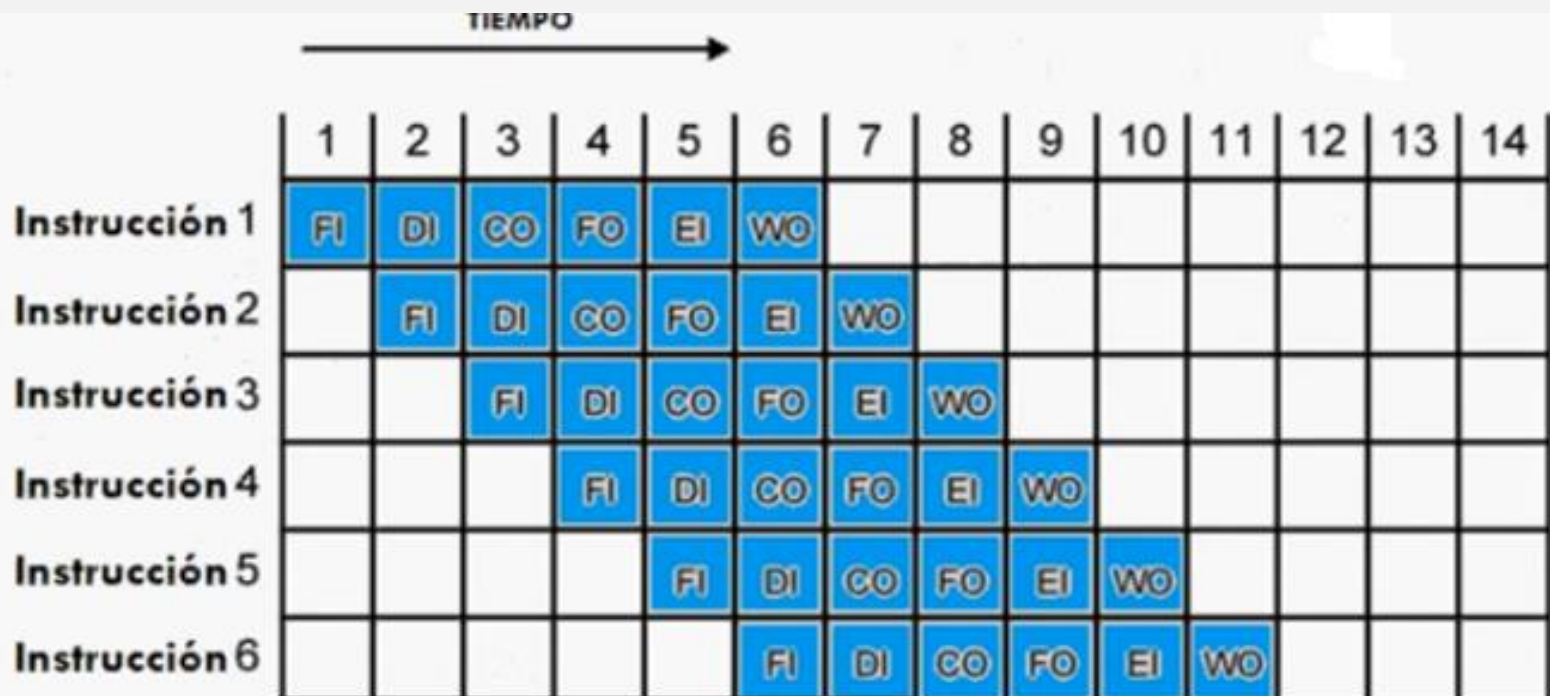
Las instrucciones se dividen en varias fases (búsqueda, decodificación, ejecución) que se pueden superponer.

- Cuando se ejecuta la etapa de decodificación DI de la primera instrucción, se puede ejecutar al mismo tiempo la etapa de búsqueda FI de la segunda instrucción, por lo que se ejecutan hasta 6 instrucciones diferentes a la vez.

# Arquitectura Pipeline

## ETAPAS de una Instrucción:

- Buscar instrucción (FI)
- Decodificar instrucción (DI)
- Calcular operandos (CO)
- buscar operandos (FO)
- Ejecutar instrucciones (EI)
- Escribir resultado (WO)
- El método esencialmente realiza la superposición de la ejecución en estas etapas con varias instrucciones



# Arquitectura Pipeline

- **Paralelismo**

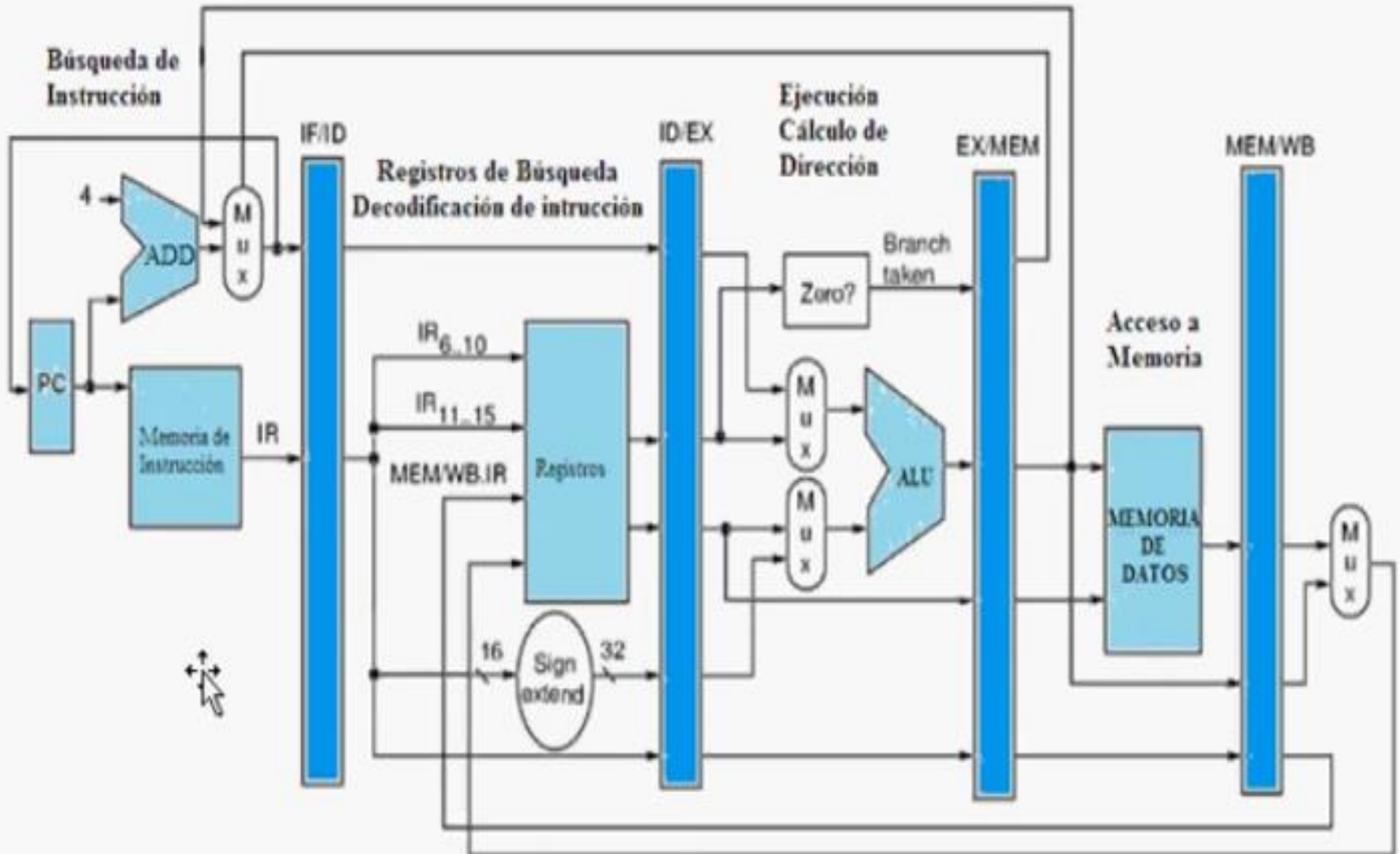
Cuando varias instrucciones se ejecutan simultáneamente en diferentes etapas, por lo tanto aumenta el paralelismo.

La **arquitectura Pipeline** (técnica para implementar simultaneidad a nivel de instrucciones dentro de un solo procesador). Un flujo de datos se convierte en un proceso comprendido por varias fases secuenciales, **siendo la entrada de cada fase la salida de la anterior**, con un almacenamiento temporal de datos entre procesos.

Beneficios: Aumenta significativamente el rendimiento sin necesidad de aumentar la frecuencia del reloj.

Inconveniente: Más complejidad en la gestión de conflictos entre instrucciones (dependencias de datos).

# Pipeline





# Arquitecturas especializadas

Las **arquitecturas especializadas** son diseños de computadoras adaptados para realizar tareas específicas de manera más eficiente que una arquitectura general.

Estas arquitecturas están optimizadas para resolver problemas concretos o manejar cargas de trabajo específicas, y han ganado importancia con el auge de aplicaciones como inteligencia artificial, procesamiento de gráficos y criptografía.

## Arquitectura de Procesadores de Gráficos (GPU)

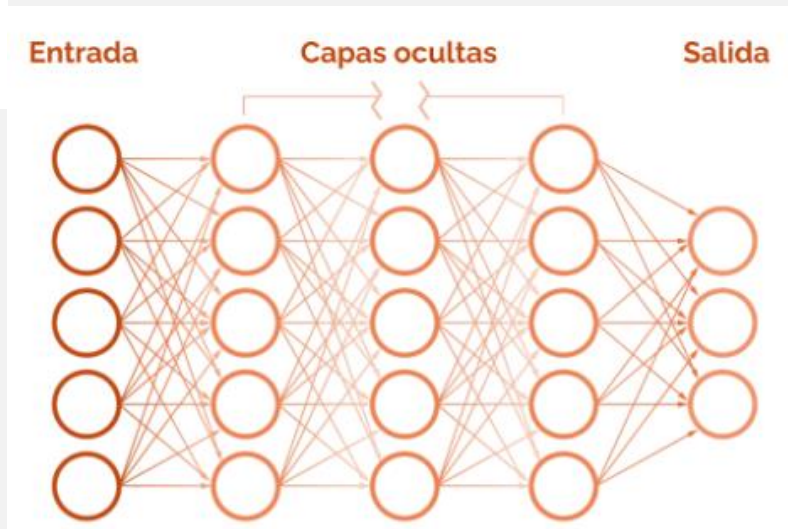
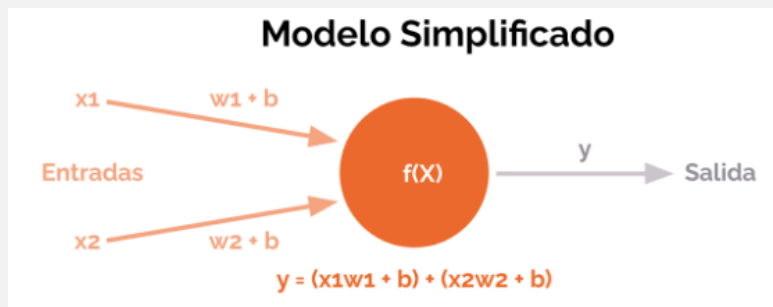
Están diseñados para acelerar el procesamiento gráfico, las **GPUs** han evolucionado para manejar tareas altamente paralelizables, como el entrenamiento de redes neuronales y el procesamiento de grandes cantidades de datos.

### Características

- **Arquitectura masivamente paralela:** Las GPUs tienen cientos o miles de núcleos que pueden realizar cálculos simultáneos.
- **Procesamiento vectorial:** Optimizado para realizar las mismas operaciones sobre múltiples datos al mismo tiempo (SIMD - Single Instruction, Multiple Data).

## Arquitectura de Procesadores de Gráficos (GPU)

•**Uso en IA y aprendizaje profundo:** Las GPUs han sido adoptadas ampliamente en el campo de la inteligencia artificial para entrenar redes neuronales profundas, gracias a su capacidad de procesamiento paralelo.



# Arquitecturas de Procesadores de Gráficos (GPU)

## Beneficios

Capacidad para manejar grandes volúmenes de cálculos paralelos. Alto rendimiento en aplicaciones gráficas y de aprendizaje profundo.

## Inconvenientes

No es adecuada para tareas que requieren procesamiento secuencial o que no son fácilmente paralelizables.



# Arquitectura de Procesadores de Señales Digitales (DSP)

Están diseñados para procesar señales en tiempo real, como el audio, video y señales de telecomunicaciones.

## Características

- **Procesamiento de datos en tiempo real:** Los DSPs están optimizados para realizar operaciones matemáticas, como multiplicaciones y sumas, de manera muy rápida.
- **Eficiencia energética:** Están diseñados para consumir poca energía, lo que los hace ideales para dispositivos portátiles y sistemas embebidos.
- **Aplicaciones:** Procesamiento de audio, video, telecomunicaciones, procesamiento de imágenes (patrones) y radar.

# Arquitecturas especializadas

## Arquitectura de Procesadores de Señales Digitales (DSP)

### **beneficio**

- Excelentes para tareas de procesamiento continuo de datos.
- Bajo consumo energético.

### **Inconveniente**

- No son tan versátiles como las CPUs o GPUs para tareas generales.

## Arquitectura de Redes Neuronales (NPU's o TPU's)

Las NPUs (Neural Processing Units) y las TPUs (Tensor Processing Units, de Google) son procesadores especializados en acelerar las operaciones de inteligencia artificial, específicamente para tareas de redes neuronales.

### Características

- **Optimización para operaciones matriciales:** Las NPUs y TPUs están diseñadas para acelerar cálculos como multiplicaciones de matrices y convoluciones, comunes en el entrenamiento y la inferencia de redes neuronales.
- **Paralelismo extremo:** Utilizan un gran número de unidades de procesamiento en paralelo para manejar operaciones sobre grandes conjuntos de datos.
- **Bajo consumo energético:** Diseñadas para ser más eficientes que las GPUs en tareas de IA.



# Arquitecturas especializadas

## Arquitectura de Redes Neuronales (NPUs o TPUs)

### **Beneficio**

- Rendimiento superior en aplicaciones de IA.
- Reducción en el tiempo de entrenamiento de modelos

### **Inconveniente**

- No son eficientes en tareas que no están relacionadas con redes neuronales o matrices.

# Arquitecturas especializadas

## Field-Programmable Gate Array –FPGA

Los **FPGAs** son dispositivos semiconductores configurables que permiten la reprogramación para realizar una amplia variedad de tareas.

### Características

- Reconfigurabilidad:** A diferencia de los procesadores tradicionales, los FPGAs pueden ser reprogramados para cambiar su funcionalidad según sea necesario.
- Altamente paralelos:** Los FPGAs permiten ejecutar múltiples operaciones simultáneamente, haciendo que su rendimiento sea similar al de un hardware dedicado.
- Aplicaciones:** Se utilizan en sistemas embebidos, telecomunicaciones, procesamiento de imágenes y seguridad criptográfica.

# Arquitecturas especializadas

## Field-Programmable Gate Array- FPGA

### **Beneficio**

- Flexibilidad extrema para realizar cualquier tipo de tarea.
- Alta eficiencia energética cuando se configura para tareas específicas.

### **Inconveniente**

- La programación de FPGAs es compleja y requiere un diseño especializado.
- El rendimiento puede no ser tan alto como el de un circuito integrado específico.

# Procesadores ASIC (Application-Specific Integrated Circuit)

Los **ASICs** son chips diseñados específicamente para realizar una tarea particular, con un rendimiento y eficiencia inigualables en esa tarea.

## Características

- **Diseño fijo:** Una vez fabricado, el ASIC solo puede realizar la tarea para la que fue diseñado, lo que lo hace extremadamente eficiente para esa aplicación.
- **Aplicaciones:** Se usan en criptografía (minería de criptomonedas), telecomunicaciones y otras áreas donde se requiere un alto rendimiento en una tarea muy específica.

## Beneficios

- Máxima eficiencia en términos de rendimiento y consumo energético.
- Ideal para aplicaciones que necesitan procesamiento de datos extremadamente rápido.

## Inconvenientes

- No pueden ser reprogramados ni reutilizados para otras tareas.
- Costosos de desarrollar y fabricar.

## Arquitectura SIMD y MIMD

- **SIMD (Single Instruction, Multiple Data)**

Ejecuta una misma instrucción sobre múltiples datos al mismo tiempo. Utilizado en procesamiento de gráficos, cálculos científicos y redes neuronales.

- **MIMD (Multiple Instruction, Multiple Data)**

Permite ejecutar diferentes instrucciones en diferentes datos simultáneamente, utilizado en sistemas multiprocesador y supercomputadoras.

### Beneficios

- **SIMD:** Alta eficiencia en tareas paralelizables.
- **MIMD:** Más flexibilidad y rendimiento en tareas más diversas.

### Inconvenientes

- **SIMD:** No es ideal para tareas que no son paralelizables.
- **MIMD:** Mayor complejidad y consumo de recursos.

# Arquitectura de Computación Cuántica

La **computación cuántica** explota fenómenos de la mecánica cuántica, como la superposición y el entrelazamiento, para realizar cálculos exponencialmente más rápidos en problemas específicos.

## Características

- **Qubits:** A diferencia de los bits tradicionales, los qubits pueden representar 0 y 1 al mismo tiempo.
- **Aplicaciones:** Criptografía, simulación de moléculas y materiales, optimización y aprendizaje automático.

## Beneficios

- Potencial para resolver problemas intratables para las computadoras clásicas.

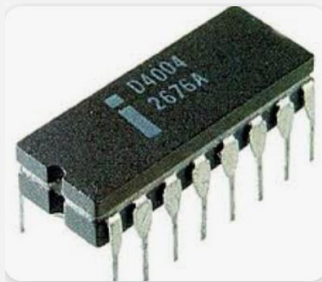
## Inconveniente

- Tecnología aún en desarrollo y difícil de implementar a gran escala.

# El microprocesador

- Un microprocesador es un componente central en una computadora que realiza la mayoría de las operaciones de procesamiento de datos.
- Se le puede considerar el "cerebro" del sistema, ya que ejecuta instrucciones de programas y coordina las actividades de otros componentes.

## Arquitectura x86



4004 -Intel



8086 -Intel



8088-Intel



80186-Intel



...



# Arquitectura ARM-Advanced RISC Machine

## Máquina avanzada de RISC

El microprocesador de arquitectura ARM está basado en la arquitectura RISC y sus ventajas son la eficiencia energética, y son aptos para dispositivos móviles que funcionan con batería.

microprocesador de  
Arquitectura ARM

Teléfonos inteligentes (Samsung y Apple)

Relojes inteligentes (Apple Watch)

Tabletas y computadoras portátiles  
(MacBooks)

Tarjetas de desarrollo Raspberry Pi





# Microcontrolador

- Un microcontrolador es un tipo específico de microprocesador diseñado para realizar tareas de control en sistemas embebidos.
- Un microcontrolador está optimizado para tareas específicas y suele integrarse en dispositivos electrónicos para realizar funciones de control y automatización.
- Tamaño compacto, periféricos integrados, eficientes en consumo de energía.

Ejemplos:

- **AVR** (como el ATmega328), **PIC** (como el PIC16F84), **ARM Cortex-M** (como el STM32)