



Instituto **Tecnológico**[®]
de Aguascalientes

ARQUITECTURA DE COMPUTADORAS

Competencia: TEMA 1 Reconocer los conocimientos acerca del funcionamiento del computador

¿Cuál creen que es el modelo de arquitectura de cómputo más relevante en la actualidad y por qué?”

<https://github.com/sserna2025/academicos-Arquitectura/>



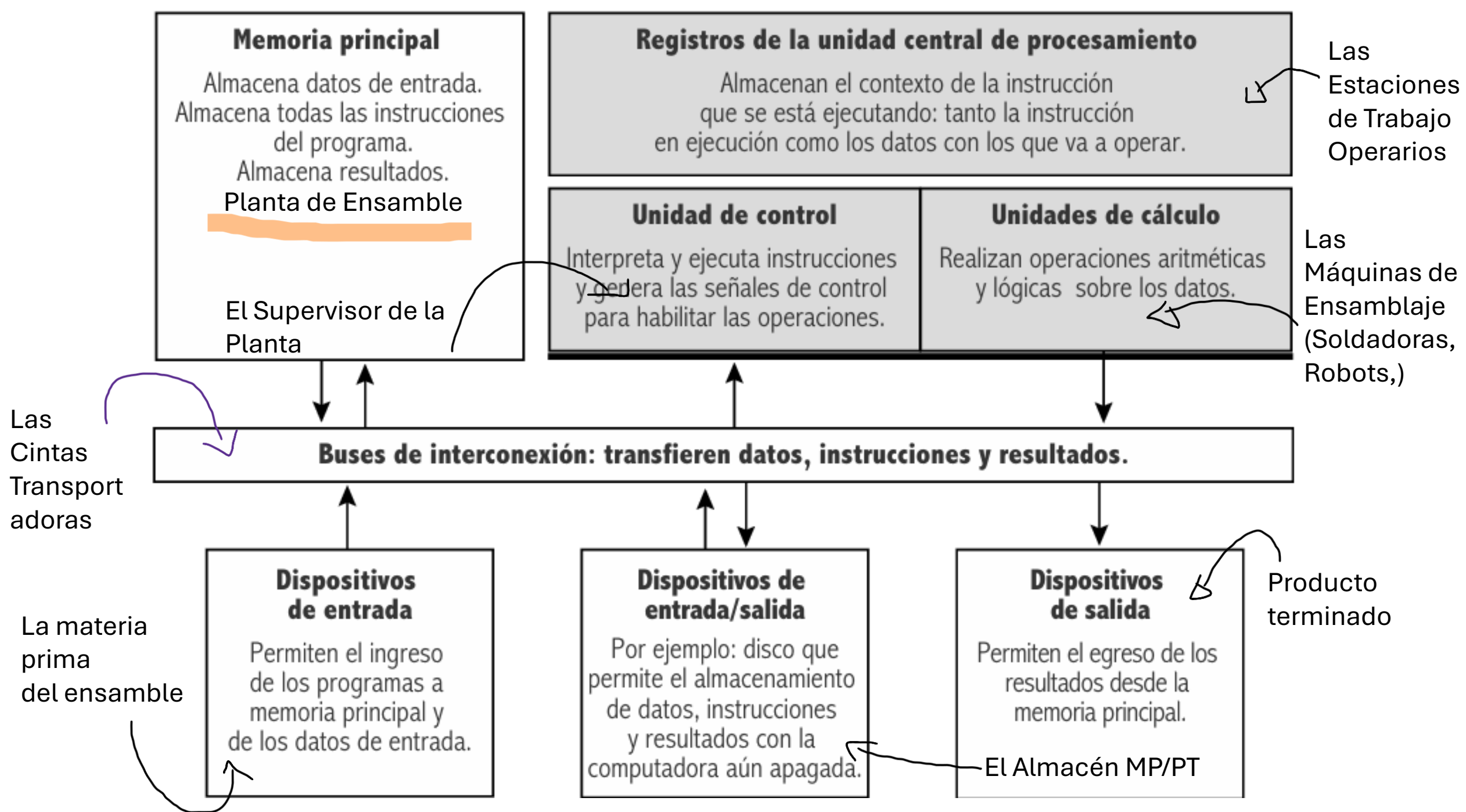
Generalidades

¿Qué es una computadora?

Una computadora es un dispositivo electrónico, diseñado para **aceptar datos** de **entrada** y **realizar operaciones** sobre ellos (organizadas en una secuencia lógica y predeterminada por un algoritmo), para elaborar **resultados** que se puedan obtener como **salidas**.

Un algoritmo computacional se determina por **una secuencia de operaciones finita** que permite resolver un problema computacional. Se representa con instrucciones que la computadora puede interpretar y ejecutar. Al **conjunto de instrucciones** que representa un algoritmo se lo denomina **programa**; expresado de otra manera, un programa es la representación de un algoritmo en un **lenguaje de programación**.

Componentes de una computadora:





Componentes de una computadora

En resumen, la imagen anterior muestra cómo la información (datos e instrucciones) entra en la computadora a través de los dispositivos de entrada. Se almacena en la memoria principal, donde la CPU puede acceder a ella rápidamente. La CPU, a través de la unidad de control y las unidades de cálculo, procesa la información siguiendo las instrucciones del programa. La información procesada (resultados) se envía a través de los buses de interconexión a los dispositivos de salida para ser mostrada al usuario.

Clasificación de las computadoras

Característica	Computadoras Analógicas	Computadoras Digitales	Computadoras Híbridas
Procesamiento	Manipulación directa de magnitudes físicas.	Cálculos aritméticos y lógicos con números binarios.	Conversión de señales analógicas a digitales y procesamiento digital.
Precisión	Limitada por la precisión de los componentes físicos y factores ambientales.	Alta precisión, determinada por el número de bits utilizados.	Depende de la precisión de la conversión analógico-digital; generalmente buena.

Velocidad	Procesamiento en tiempo real para ciertas aplicaciones; rápido en simulaciones de sistemas físicos.	Velocidad de procesamiento muy alta para tareas que se pueden discretizar; puede tener latencia en la conversión A/D.	Combina la velocidad del procesamiento analógico con la precisión y flexibilidad del digital.
Almacenamiento	Difícil almacenamiento de datos a largo plazo; se basa en la configuración física del sistema.	Fácil almacenamiento de grandes cantidades de datos en diversos medios.	Almacenamiento digital de los datos procesados.
Programación	No se programan en el sentido tradicional; se configuran mediante conexiones físicas.	Se programan mediante lenguajes de programación y software.	Se programan las partes digitales; la configuración de la parte analógica es fija.
Flexibilidad	Limitada a la solución de un tipo específico de problema.	Altamente flexible y programable para una amplia gama de aplicaciones.	Mayor flexibilidad que las analógicas puras, pero menor que las digitales puras.

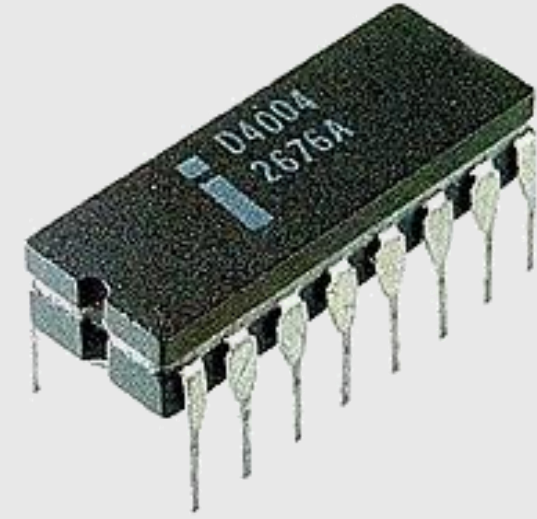
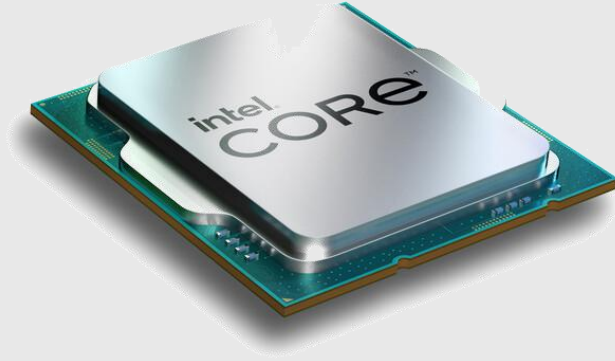
Manejo de Errores	Más tolerante a pequeñas variaciones en la entrada; errores graduales.	Sensible a errores en la entrada digital; errores discretos.	Depende del tipo de error y en qué parte del sistema se produce.
Ejemplos de Uso	<ul style="list-style-type: none">- Termostatos bimetálicos.- Computadoras analógicas para Simuladores de vuelo antiguos.	<ul style="list-style-type: none">- Computadoras personales- Smartphones- Servidores- Supercomputadoras.- Sistemas embebidos.	<ul style="list-style-type: none">- Sistemas de control industrial que requieren procesamiento de señales analógicas y control digital- Sistemas de adquisición de datos .- Simuladores híbridos (ej., simuladores de vuelo modernos con componentes analógicos y digitales).

Modelos de Arquitectura

Modelos de Arquitecturas de Cómputo

La arquitectura de cómputo se refiere a la estructura y organización de un sistema informático. Hay diferentes modelos que describen cómo se pueden estructurar los componentes de un sistema.

A continuación se presentan tres modelos principales: clásicos, segmentados y de multiprocesamiento.



Característica

Microprocesador Moderno (Intel Core)

Microprocesador Antiguo

Tamaño y complejidad

Circuito integrado de gran tamaño, millones de transistores, múltiples núcleos.

Chip pequeño, pocos transistores, un solo núcleo.

Tecnología de fabricación

Procesos de fabricación nanométrica, transistores más pequeños y densos.

Procesos de fabricación más antiguos, transistores más grandes.

Velocidad

Velocidades de reloj mucho más altas (gigahertz), mayor capacidad de procesamiento.

Velocidades de reloj más bajas (megahertz), capacidad de procesamiento limitada.

Consumo de energía

Diseñados para ser energéticamente eficientes.

Mayor consumo de energético

1.1.1 Modelos de arquitectura

El modelo de **von Neumann** es la base de la arquitectura de las computadoras modernas. Este modelo fue propuesto por John von Neumann y presenta varias características fundamentales que permiten a las computadoras funcionar de manera eficiente.

Características Principales

1.Programa almacenado: Una de las ideas más revolucionarias de von Neumann fue almacenar tanto los datos como las instrucciones en la misma memoria. Esto significa que los programas pueden ser almacenados en la memoria de la computadora y ejecutados cuando sea necesario. Por ejemplo, cuando abres un programa como un procesador de texto, el software está almacenado en la memoria y la computadora puede acceder a él para realizar tareas de edición de texto.

1.1.1 Modelos de arquitectura Clásicos

2. Ciclo de instrucción: El ciclo de instrucción es el corazón del funcionamiento de una computadora basada en von Neumann. Este ciclo se repite continuamente y consta de cuatro etapas principales:
- Buscar la instrucción: La CPU obtiene la siguiente instrucción de la memoria.
 - Decodificarla: La CPU interpreta lo que debe hacer con la instrucción.
 - Ejecutarla: La CPU realiza la operación necesaria, como sumar dos números.
 - Almacenar el resultado: La CPU guarda el resultado de la operación en la memoria.

Por ejemplo, al realizar la operación $3+53+5$, la CPU sigue estas etapas: primero busca la instrucción de sumar, luego interpreta que debe sumar 3 y 5, ejecuta la suma para obtener 8 y finalmente almacena el resultado.

1.1.1 Modelos de arquitectura

1. Cuello de botella de von Neumann:

1. Un cuello de botella se refiere a una limitación en el rendimiento. En el modelo de von Neumann, esto ocurre cuando la CPU tiene que esperar a que los datos se transfieran desde la memoria. Esto puede limitar la velocidad de procesamiento, ya que la CPU no puede hacer nada mientras espera. Por ejemplo, si una computadora necesita procesar grandes cantidades de datos, la velocidad de transferencia entre la CPU y la memoria puede ralentizar el procesamiento.

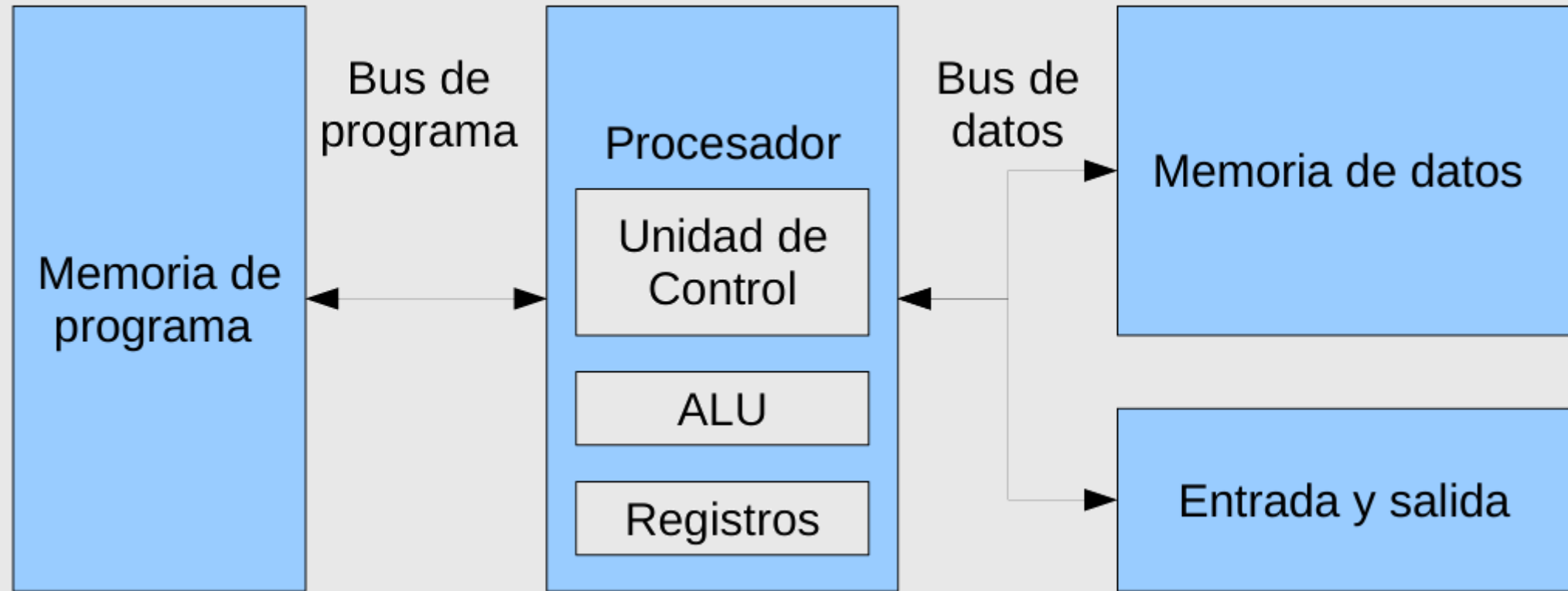
Limitaciones

1. Velocidad de transferencia de datos:

1. La velocidad a la que los datos se mueven entre la CPU y la memoria es un factor limitante en el rendimiento general de una computadora. Si la CPU es muy rápida pero la memoria es lenta, la computadora no funcionará de manera eficiente.



VonNewman





En resumen

1. La unidad de control toma una instrucción de la memoria de programa.
2. Decodifica la instrucción para saber qué operación debe realizar.
3. Obtiene los datos necesarios de la memoria de datos a través del bus de datos.
4. Realiza la operación en la ALU.
5. Almacena el resultado en la memoria de datos o en un registro.
6. Repite el proceso con la siguiente instrucción.

1.1.1 Modelos de arquitectura

Dificultad para manejar tareas paralelas:

1. El modelo de von Neumann está diseñado para ejecutar instrucciones de forma secuencial. Esto significa que solo puede manejar una tarea a la vez, lo que puede dificultar la ejecución de múltiples tareas simultáneamente. Por ejemplo, si intentas descargar un archivo mientras juegas un videojuego, la computadora puede experimentar lentitud porque está tratando de procesar dos tareas al mismo tiempo.

Piensa en una cocina pequeña donde solo puedes hacer una cosa a la vez. Si decides cocinar una receta (**instrucción**), primero debes buscar los ingredientes (**buscar la instrucción**), luego leer la receta (**decodificarla**), cocinar los ingredientes (**ejecutarla**) y finalmente servir el plato (**almacenar el resultado**). Si tienes que esperar a que el agua hierva o que el horno se precaliente (**cuello de botella**), eso ralentiza todo el proceso de cocción. Además, si intentas hacer dos platos a la vez en una cocina pequeña, no podrás hacerlo de manera eficiente.

1.1.2: Segmentadas

Las arquitecturas segmentadas, o arquitecturas con canalización de instrucciones, tienen como objetivo mejorar el rendimiento mediante la ejecución simultánea de varias etapas del ciclo de instrucciones. En este diseño, el procesador se divide en varias unidades funcionales independientes, cada una de las cuales se encarga de procesar diferentes instrucciones simultáneamente.

1.Etapas del ciclo de instrucción : El ciclo de instrucción normalmente consta de varias etapas:

1. Obtener : recuperar la instrucción de la memoria.
2. Decodificar : Interpretar la instrucción para comprender lo que debe hacerse.
3. Ejecutar : Llevar a cabo la instrucción.
4. Escribir de nuevo : guarda el resultado en la memoria.

1.1.2: Segmentadas

En una arquitectura segmentada, mientras se obtiene una instrucción, se puede decodificar otra y se puede ejecutar una tercera.

2.- Unidades funcionales : cada etapa del ciclo de instrucción puede ser gestionada por unidad funcional diferente. Por ejemplo:

1. La unidad A obtiene instrucciones.
2. La unidad B los decodifica.
3. La unidad C los ejecuta.
4. La unidad D escribe los resultados.

Esta división permite un **mayor rendimiento** ya que se procesan múltiples instrucciones al mismo tiempo.

1.1.2: Segmentadas

Consideremos una línea de montaje de una fábrica donde diferentes trabajadores son responsables de diferentes tareas.

- El trabajador 1 (**Buscar**) recoge materias primas.
- El trabajador 2 (**Decodificar**) lee las instrucciones de ensamblaje.
- El trabajador 3 (**Ejecutar**) ensambla el producto.
- El trabajador 4 (**escritura posterior**) empaqueta el producto terminado.

Mientras el Trabajador 1 reúne materiales para el Producto A, el Trabajador 2 puede leer instrucciones para el Producto B, el Trabajador 3 puede ensamblar el Producto C y el Trabajador 4 puede empaquetar el Producto D. Este procesamiento paralelo aumenta la eficiencia general.

1.1.3: Multiprocesamiento

Multiprocesamiento en CPU

Cuando se busca mejorar el desempeño de un programa más allá de lo que permite la técnica de segmentación del cauce, que se refiere al límite teórico de una instrucción por ciclo de reloj, es necesario utilizar múltiples procesadores para la ejecución del programa. Esto se debe a que un solo procesador puede volverse un cuello de botella, limitando la velocidad de procesamiento de tareas complejas.

Existen diferentes tipos de arquitecturas de multiprocesamiento, que se pueden clasificar de la siguiente manera:

1.SISO (Single Instruction, Single Operand): Este tipo de computadoras ejecuta una sola instrucción a la vez con un único operando. Un ejemplo de esto podría ser una calculadora básica, que solo realiza una operación simple a la vez, como sumar dos números.

1.1.2: Segmentadas

1.MISO (Multiple Instruction, Single Operand): Aunque este tipo no se ha implementado, se refiere a un sistema que podría ejecutar múltiples instrucciones en un solo operando. Por ejemplo un chef que realiza varios pasos de preparación en una sola receta con un solo ingrediente base.

2.SIMO (Single Instruction, Multiple Operand): En este caso, un procesador puede ejecutar la misma instrucción sobre múltiples operandos. Por ejemplo, una máquina de hacer helados que puede mezclar varios sabores al mismo tiempo utilizando la misma técnica de mezcla.

3.MIMO (Multiple Instruction, Multiple Operand): Este es el sistema más avanzado y se encuentra en sistemas de multiprocesamiento simétrico (SMP) y clusters. Por ejemplo, un equipo de trabajo donde cada miembro tiene diferentes tareas (instrucciones) pero todos están trabajando con los mismos recursos (operandos) para completar un proyecto.

Bibliografía

Angulo, J. (s.f.). Micro-controladores PIC. Diseño Práctico de Aplicaciones. España: McGraw Hill.

Brey, B. (2002). Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486 Pentium, Pentium Pro Processor, Pentium II, Pentium III, and Pentium IV: Architecture, Programming, and Interfacing (Sexta ed.). USA: Prentice Hall.

Greenfield, W., & Bannatyne, R. (s.f.). Using Microprocessors and Microcomputers: The Motorola Family (Cuarta ed.). USA: Prentice Hall.

Hill, M., Jouppi, N., & Sohi, G. (s.f.). Readings in computer architecture.

Huang, H.-W. (s.f.). MC 68HC12 An Introduction. Software and Hardware Interfacing. USA: Delmar Learning.

Mano, M. (2000). Arquitectura de Computadoras (Tercera ed.). Prentice Hall.