

# Arquitectura de Computadores

## Resumen

Pablo Josue Rojas Yepes  
2023-2

# Definición de Arquitectura de Computadores

Es estructura y diseño organizativo de los componentes físicos y lógicos de un sistema computacional.

Establece las reglas y principios que guían la manera en que los componentes interactúan, comunican y coordinan sus acciones para lograr un funcionamiento eficiente y coherente del sistema.



# Componentes Básicos de un Sistema Computacional



Unidad Central de Procesamiento

Memoria Principal (Memoria Caché)

Dispositivos de I/O

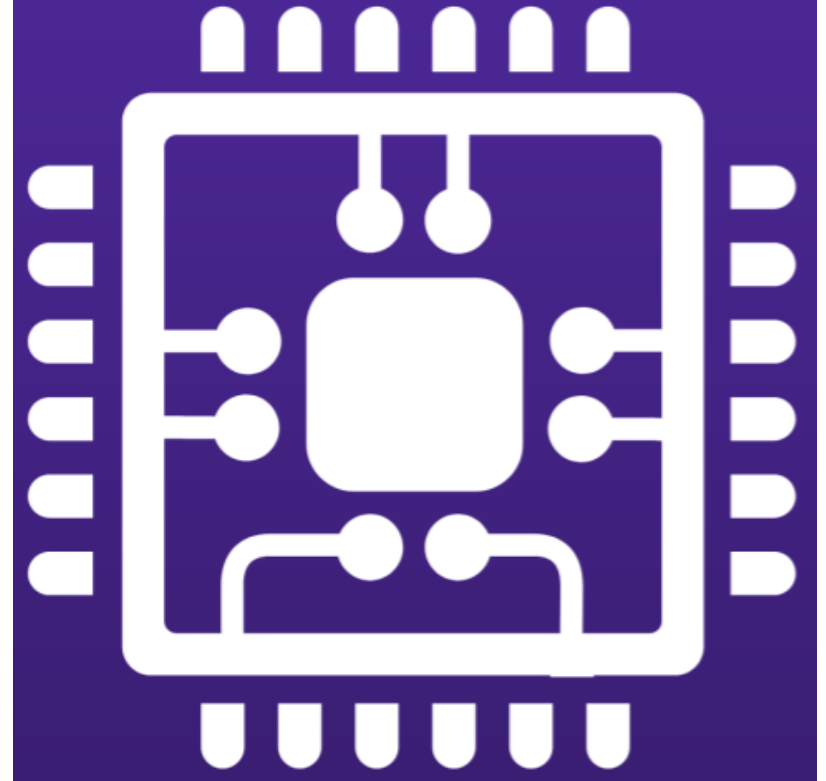
Chipset (Unidad de Control, MMU, Buses)

Almacenamiento

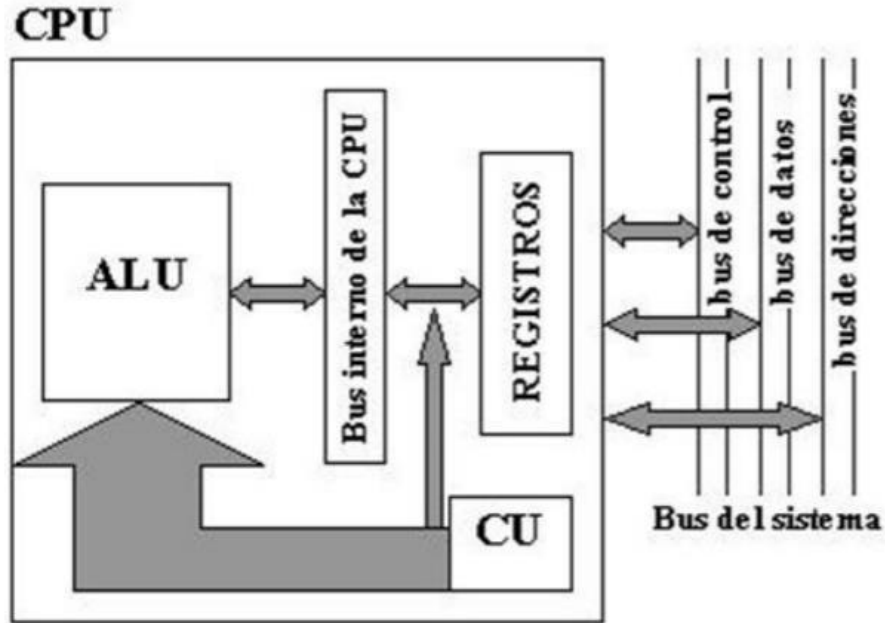
# CPU

La CPU es el cerebro de la computadora y es responsable de ejecutar instrucciones y realizar operaciones matemáticas y lógicas.

Es el componente principal y esencial de una computadora que realiza la mayoría de las operaciones de procesamiento y control.



# Estructura de la CPU



Unidad de Control (UC)

Unidad Lógica Aritmética (ALU)

Registros

Contador de Programa (PC)

Memoria Caché, Memoria Principal, MMU

Buses

# Ciclo Básico de la CPU

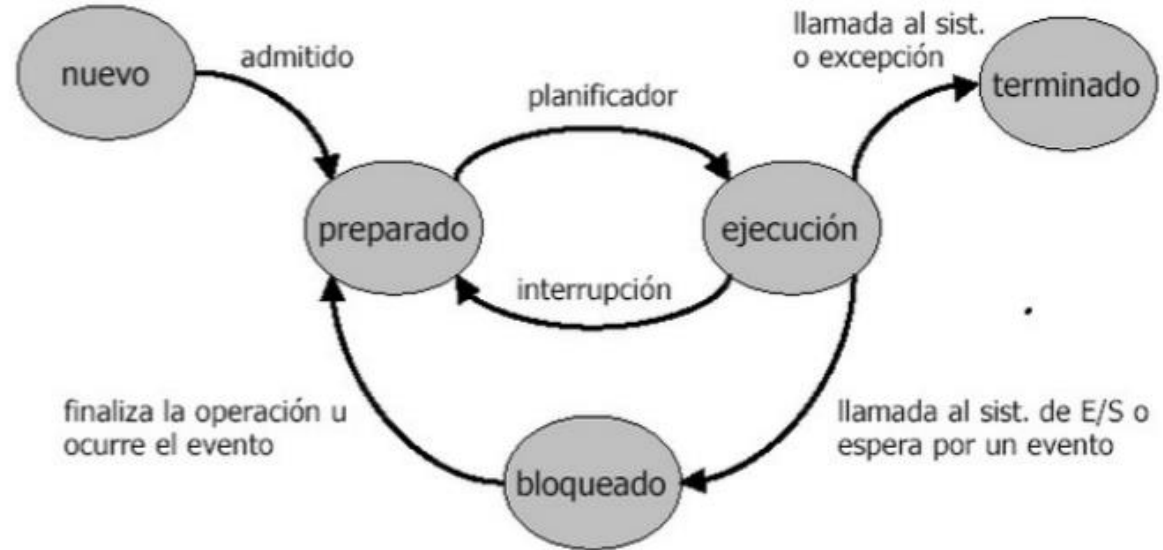
Búsqueda

Decodificación

Ejecución

Pipeline stage:

- Buscar Instrucción
- Decodificar Instrucción
- Ejecutar Instrucción
- Acceder a Memoria
- Escribir Resultados



# Jerarquía de Memoria

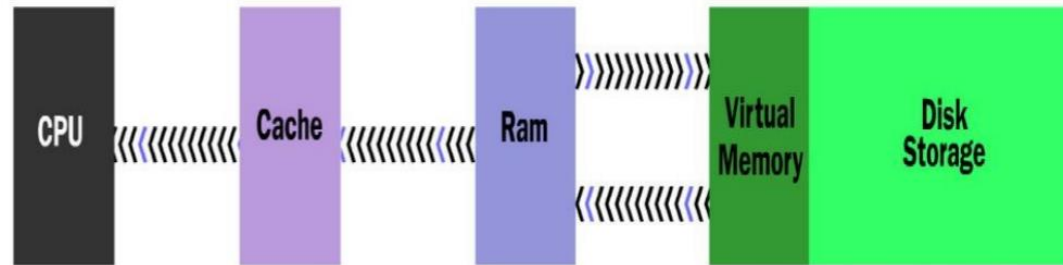
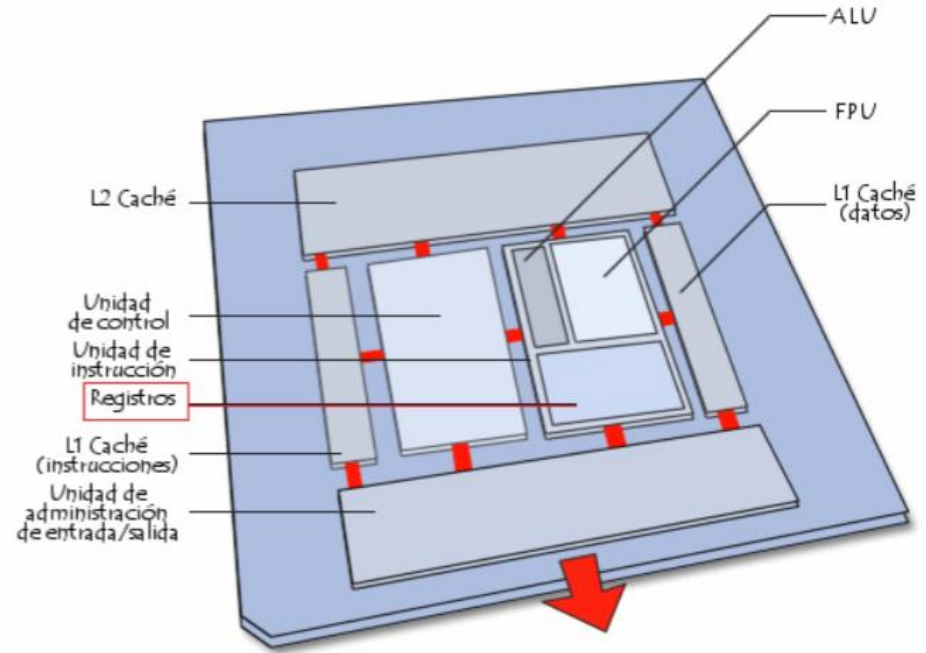


Se refiere a la organización y estructura de diferentes niveles de memoria en un sistema informático.

Diseñada para optimizar el rendimiento y la eficiencia en el acceso a los datos y las instrucciones utilizadas por el procesador.

# Tipos de memoria

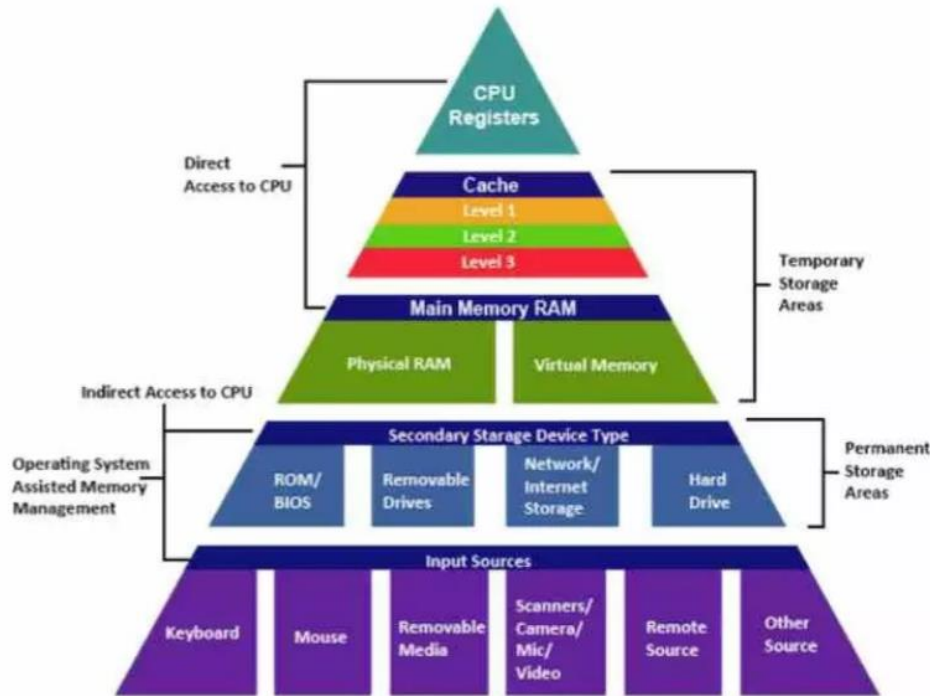
- Registros
- Caché
- RAM
- Memoria Virtual
- Almacenamiento



© 2000 How Stuff Works, Inc



# Organización de la Caché



Cada nivel de caché está diseñado para equilibrar la velocidad, la capacidad y el costo de almacenamiento, de manera que se puedan aprovechar las ventajas de la memoria rápida y de mayor capacidad al mismo tiempo.

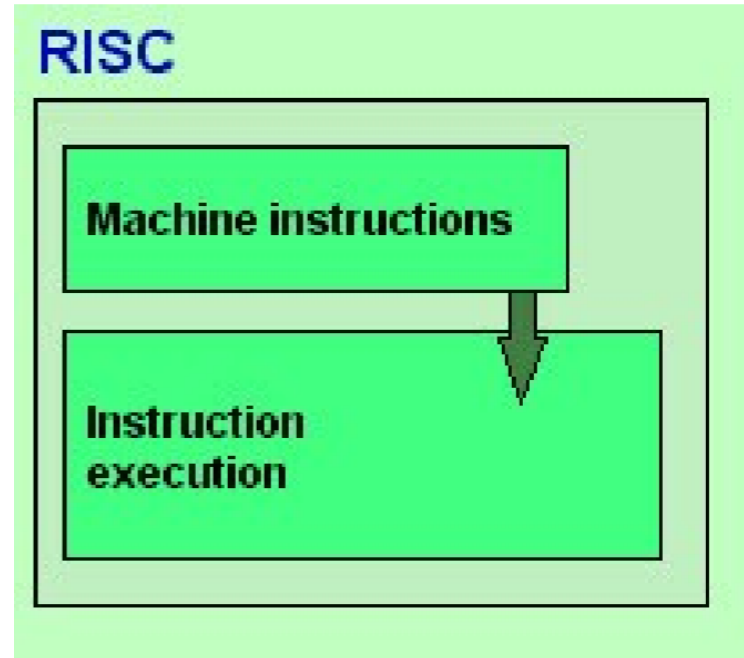
# Arquitecturas RISC

Busca simplificar la ejecución de instrucciones y optimizar el rendimiento.

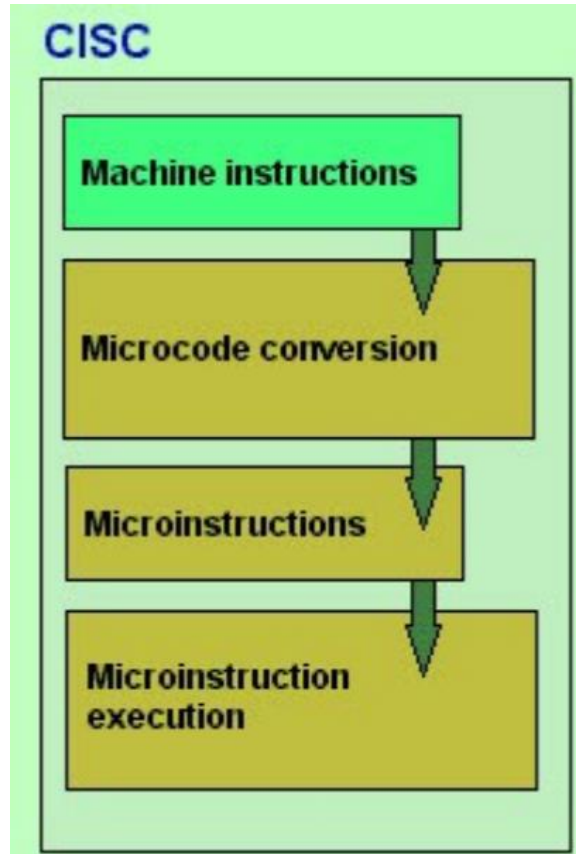
Se centra en un conjunto reducido de instrucciones simples y eficientes.

Se caracteriza por tener un conjunto de instrucciones limitado y simple.

Cada instrucción RISC realiza una tarea específica y se ejecuta en un ciclo de reloj.



# Arquitecturas CISC



Puede requerir más instrucciones para lograr ciertas tareas en comparación con la arquitectura CISC.

Se caracteriza por incluir un conjunto diverso y complejo de instrucciones en la arquitectura de la CPU.

Busca proporcionar instrucciones más completas y variadas para facilitar la programación de tareas complejas.

Las instrucciones son más ricas en funcionalidades y pueden realizar múltiples operaciones en una sola instrucción.

# Formatos de Instrucciones - Aspectos clave

Define el conjunto de operaciones que la CPU puede realizar y cómo estas operaciones son representadas y manipuladas internamente.

Conjunto de Instrucciones (ISA)

Operandos y Modos de Direcccionamiento

Cantidad de Registros y su Organización

Unidades Funcionales

Control de Flujo y Predicción de Saltos

Manejo de Excepciones y Trampas

The Hack ALU operation

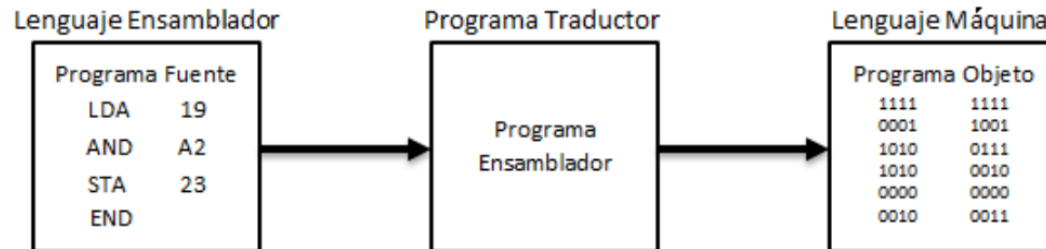
pre-setting the x input		pre-setting the y input		selecting between computing + or &	post-setting the output	Resulting ALU output
zx	nx	zy	ny	f	no	out
if zx then x=0	if nx then x=!x	if zy then y=0	if ny then y=!y	if f then out=x+y else out=x&y	if no then out=!out	out(x,y)=
1	0	1	0	1	0	0
1	1	1	1	1	1	1
1	1	1	0	1	0	-1
0	0	1	1	0	0	x
1	1	0	0	0	0	y
0	0	1	1	0	1	!x
1	1	0	0	0	1	!y
0	0	1	1	1	1	-x
1	1	0	0	1	1	-y
0	1	1	1	1	1	x+1
1	1	0	1	1	1	y+1
0	0	1	1	1	0	x-1
1	1	0	0	1	0	y-1
0	0	0	0	1	0	x+y
0	0	0	1	1	1	y-x
0	0	0	0	0	0	x&y
0	1	0	1	0	1	x y

# Codificación de operaciones y operandos

- Código de Operación (Opcode)
- Operandos
- Modo de Dirección
- Otros Campos (Opcional)

*Opcode (6 bits) |*  
*Registro de Destino (5 bits) |*  
*Registro Fuente 1 (5 bits) |*  
*Registro Fuente 2 (5 bits)*

000000 | 000000 | 000000 | 000000



# Modos de direccionamiento

Registro a Registro (Register to Register)

Inmediato (Immediate)

Directo (Direct)

Indirecto (Indirect)

Registro Indirecto (Register Indirect)

Desplazamiento (Displacement)

Indexado (Indexed)

Base-Desplazamiento (Base-Displacement)

Pila (Stack)

*ADD R1, R2, R3*

*MOV R1, #10*

*MOV R1, [1000]*

*MOV R1, [R2]*

*MOV R1, [R2]*

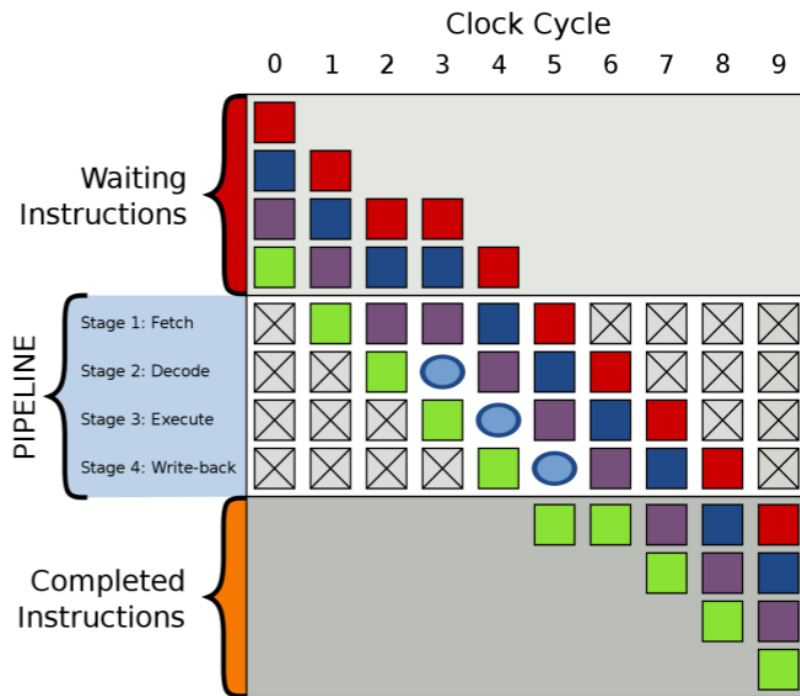
*MOV R1, [R2+4]*

*MOV R1, [R2+R3]*

*MOV R1, [R2+4]*

*PUSH R1*

# Pipeline en el Diseño de CPU



Permite superponer la ejecución de múltiples etapas de instrucciones.

Mientras una instrucción está siendo ejecutada en una etapa, la siguiente instrucción puede comenzar a procesarse en la etapa anterior.

Aumentar la eficiencia y el rendimiento de la CPU.

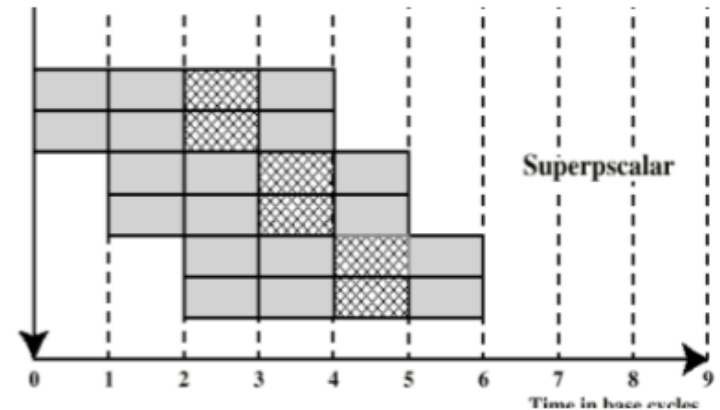
Una nueva instrucción se inicia en el pipeline en cada ciclo de reloj, incluso antes de que la anterior haya completado todas las etapas.

# Ejecución superescalar

Puede ejecutar varias instrucciones al mismo tiempo.

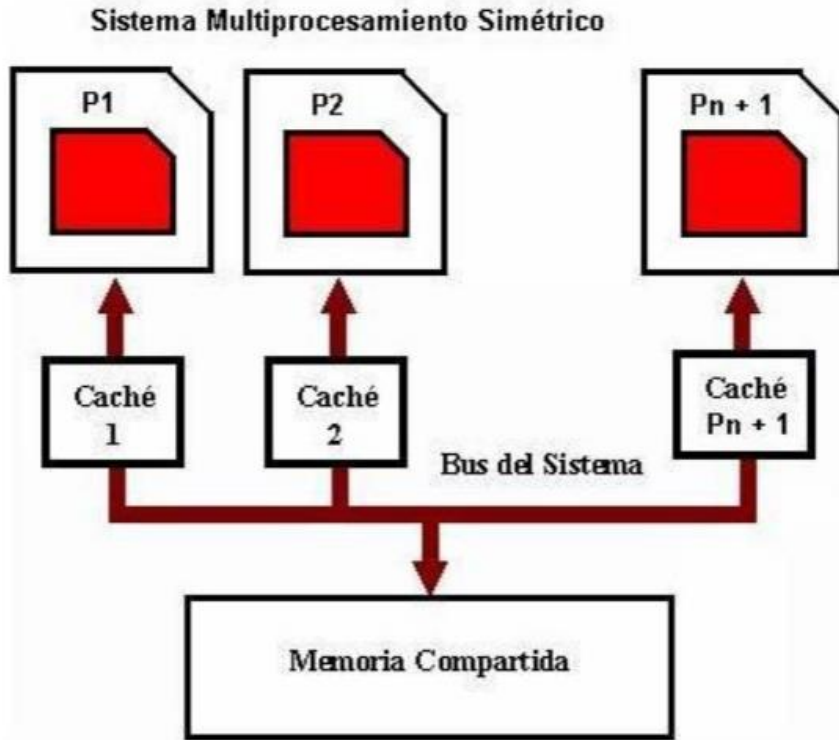
Pueden realizar múltiples tareas de manera simultánea al tener múltiples unidades de ejecución independientes.

Están diseñadas para manejar diferentes tipos de instrucciones y pueden ejecutarlas en paralelo.





# Ejecución múltiple



Permite la ejecución de múltiples instrucciones de manera concurrente.

Permite que múltiples hilos de ejecución se ejecuten simultáneamente en el mismo núcleo de CPU.

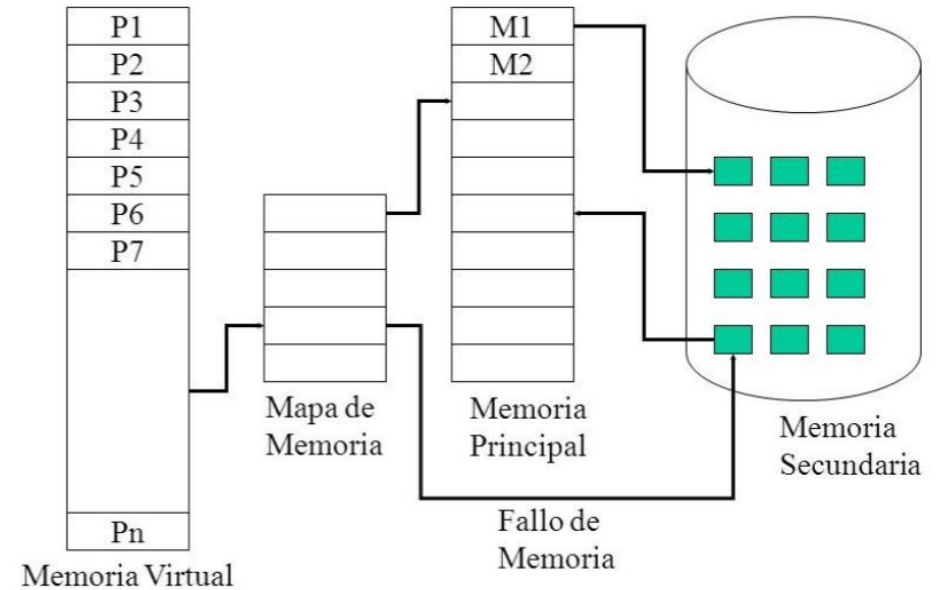
Involucra múltiples núcleos o unidades de procesamiento independientes en la misma CPU, lo que permite la ejecución simultánea de múltiples instrucciones.

# Gestión de Memoria Virtual

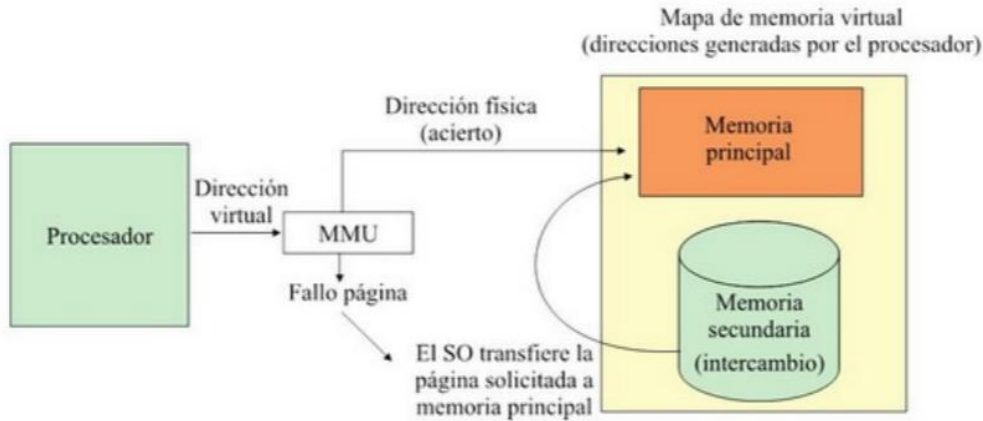
Técnica para administrar el almacenamiento de datos en un sistema.

Permite trabajar con más datos de los que pueden caber físicamente en la memoria RAM disponible.

Los datos necesarios se mantienen en la RAM, aquellos que no están siendo utilizados se almacenan en el disco y se traen de vuelta a la RAM cuando son necesarios.



# Paginación



Es un método de administración de la memoria que divide la memoria en bloques de tamaño fijo llamados "páginas".

La MMU se encarga de traducir las direcciones virtuales generadas por el procesador en direcciones físicas correspondientes.

# Segmentación

La segmentación divide la memoria en segmentos de diferentes tamaños y cada segmento puede representar una sección lógica independiente del programa, como el código, los datos, la pila, etc.

Permite una administración más flexible de la memoria, ya que los segmentos pueden tener tamaños diferentes y representar partes distintas del programa.



# Estructura del Sistema I/O



Se refiere a cómo están organizados los componentes que permiten que una computadora interactúe con dispositivos periféricos (teclados, ratones, discos, impresoras).

Se organiza en una serie de componentes y protocolos que facilitan la comunicación entre el procesador y los dispositivos periféricos.

# Organización Básica de I/O

Controladores I/O

Puertos I/O

Registros de Estado y Control

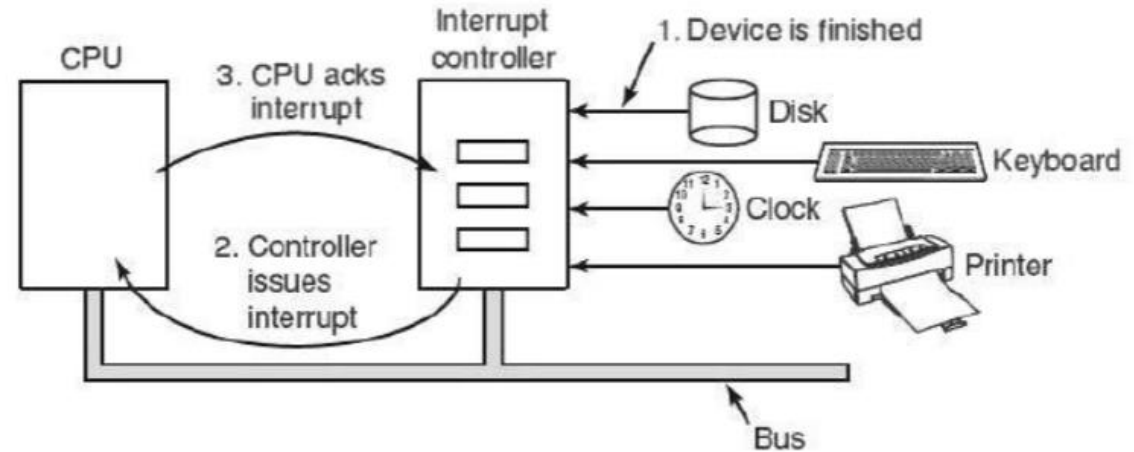
Buffers I/O

Interrupciones y Controladores de Interrupciones

Buses

Módulos de DMA

Drivers de Control



# Funciones y Características de Drivers



Gestión de Interrupciones

Traducción de Comandos

Gestión de la Memoria Intermedia (Buffering)

Control de Flujo de Datos

Configuración y Diagnóstico

Gestión de Energía

Actualizaciones y Mantenimiento

# Ensamblador

Lenguaje de programación de bajo nivel para un dispositivo específico.

Cada familia de procesadores tiene su propio conjunto de instrucciones.

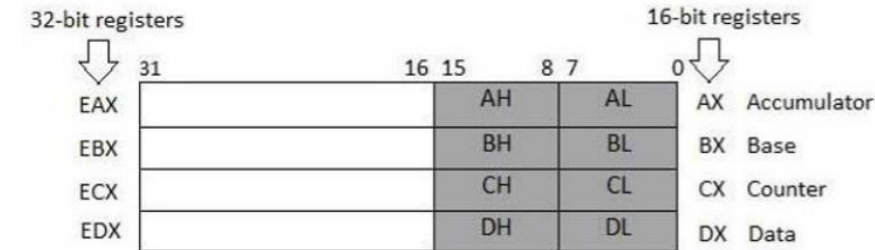
Está diseñado para una familia específica de procesadores.

Un programa en ensamblador puede dividirse en tres secciones: data, text, bss.

```
[0x00000000]> pd
0x00000000  90      nop
0x00000001  90      nop
0x00000002  6800009c00  push 0x9c0000 ; 0x009c0000
0x00000007  e8c7ace37b  call 0x7be3acd3
               0x7be3acd3(unk)
0x0000000c  bb04009c00  mov ebx, 0x9c0004
0x00000011  8903      mov [ebx], eax
0x00000013  e81903f47b  call 0x7bf40331
               0x7bf40331()
0x00000018  bb08009c00  mov ebx, 0x9c0008
0x0000001d  8903      mov [ebx], eax
0x0000001f  bb00009c00  mov ebx, 0x9c0000
0x00000024  c60300    mov byte [ebx], 0x0
-> 0x00000027  68e8030000  push 0x3e8 ; 0x000003e8
0x0000002c  e81124e37b  call 0x7be32442
               0x7be32442(unk)
=< 0x00000031  ebf4      jmp 0x100000027
0x00000033  90      nop
0x00000034  ff      invalid
0x00000035  ff      invalid
0x00000036  ff      invalid
0x00000037  ff      invalid
```



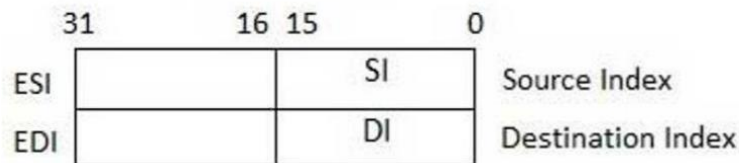
# Sintaxis



Pointer registers



Index registers



Instrucciones, Pseudo-ops y Macros.

[etiqueta] mnemonic operandos  
[;comentario]

Dirección absoluta / Dirección de segmento

Segmento de datos / Segmento de código / Pila (stack).

Registro (datos, punteros índices)

Llamadas al sistema, Variables y Constantes

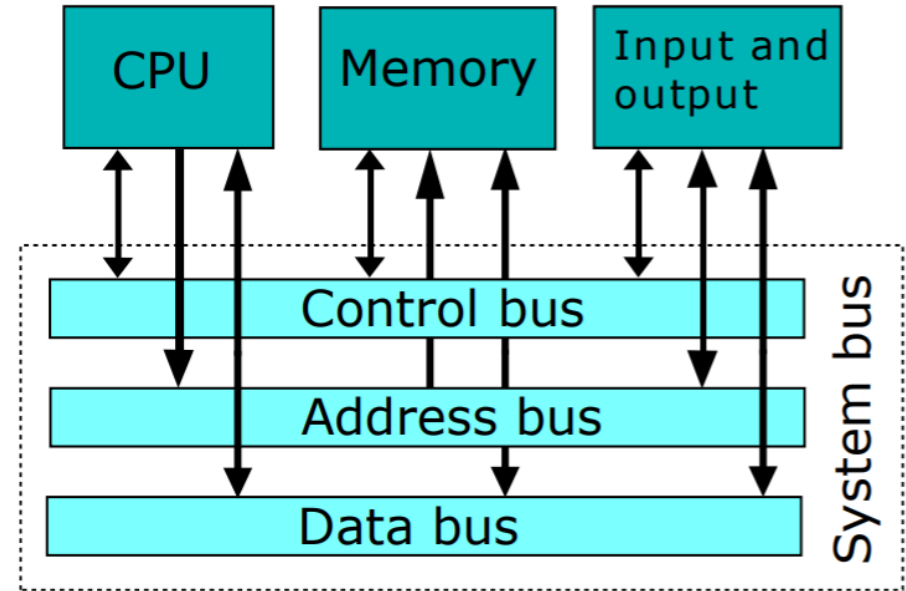
# Tipos de Buses

Conjunto de líneas de comunicación que permiten la transferencia de datos entre los diferentes componentes de un sistema.

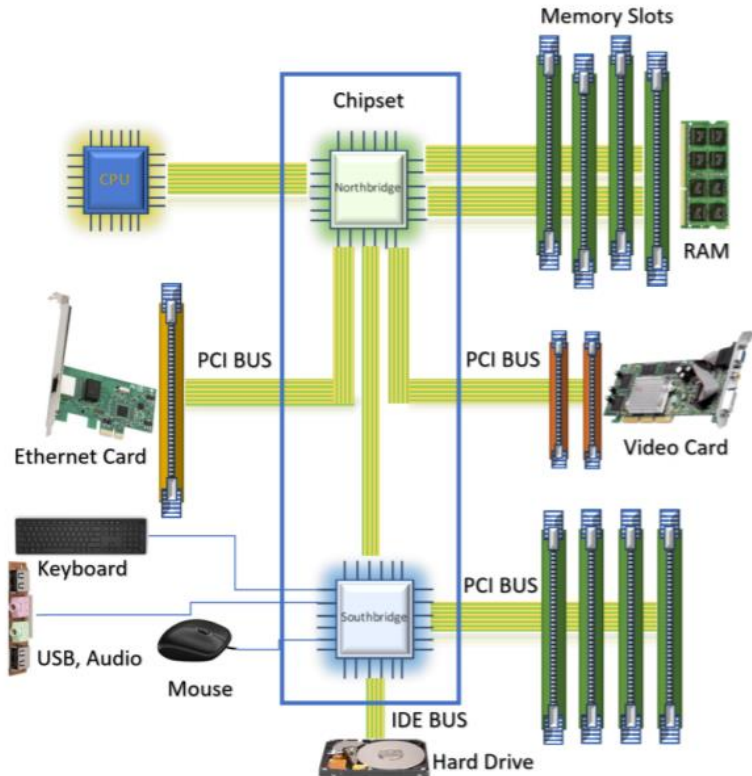
Bus de datos (Bidireccional, lectura escritura de datos)

Bus de direcciones (Unidireccional, Indica la ubicación exacta de un dato en la memoria)

Bus de control (Unidireccional, Lleva señales que coordinan y sincronizan las operaciones en el sistema)



# Chipset



Permitir la comunicación y la transferencia de datos entre estos componentes de manera eficiente y coordinada.

**Northbridge:** Se encargaba de la comunicación entre la CPU, la memoria RAM, la tarjeta gráfica y otros componentes de alta velocidad.

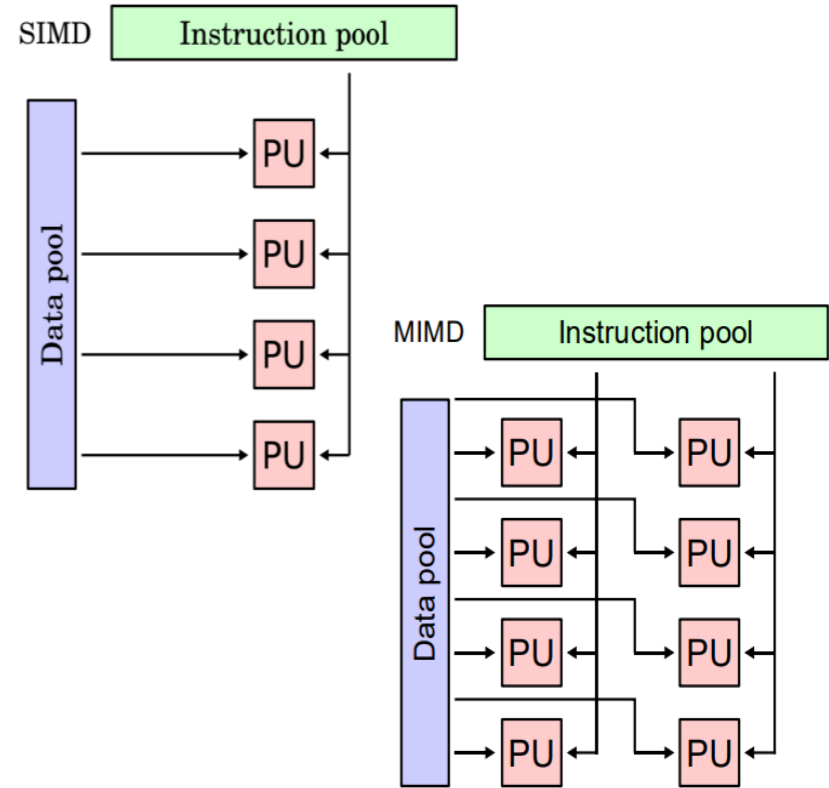
**Southbridge (Puente Sur):** Se encarga de gestionar y controlar una variedad de componentes de menor velocidad, como los puertos USB, los puertos SATA, las interfaces de audio, los puertos PCI y PCI Express, así como otras funciones de I/O.

# Procesamiento en Paralelo

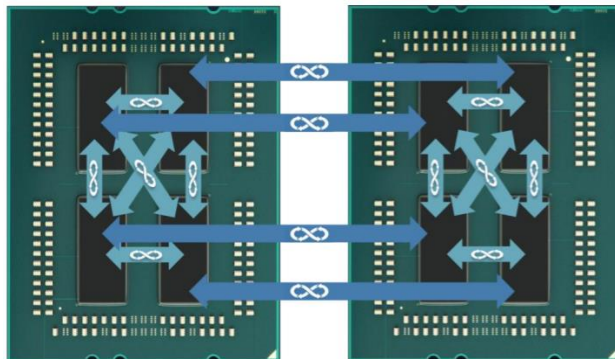
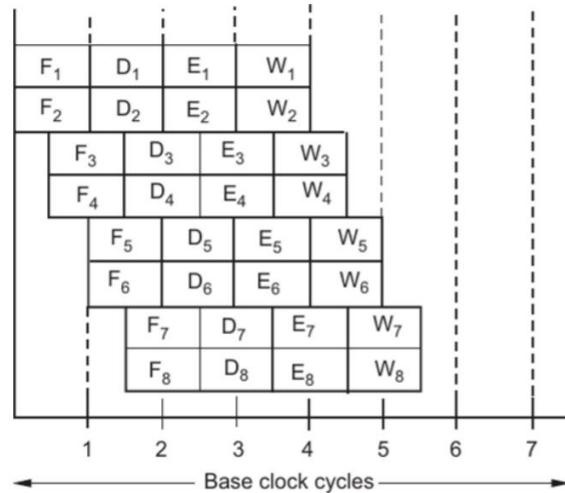
Capacidad de realizar múltiples operaciones o tareas de manera simultánea, puede llevar a cabo varias tareas al mismo tiempo.

Conceptos:

Tareas e Instrucciones Concurrentes, Procesamiento Multihilos (Multithreading), Arquitecturas SIMD y MIMD, Pipeline, Sistemas Multiprocesador, Sistemas Multinúcleo, Memoria Compartida vs. Memoria Distribuida, Sincronización y Coordinación, Cuellos de Botella, Escalabilidad, Latencia y Ancho de Banda.



# Modelos de Procesamiento en Paralelo



Source: AMD

Procesamiento en Paralelo de Instrucciones Múltiples (MIMD)

Procesamiento en Paralelo de Datos Múltiples (SIMD)

Procesamiento en Masa (Array Processing)

Procesamiento Vectorial

Procesamiento Pipelining

Procesamiento Superescalar

Procesamiento Asimétrico

Procesamiento en Red (Network Processing)

# Sistemas Multiprocesador Comunes.

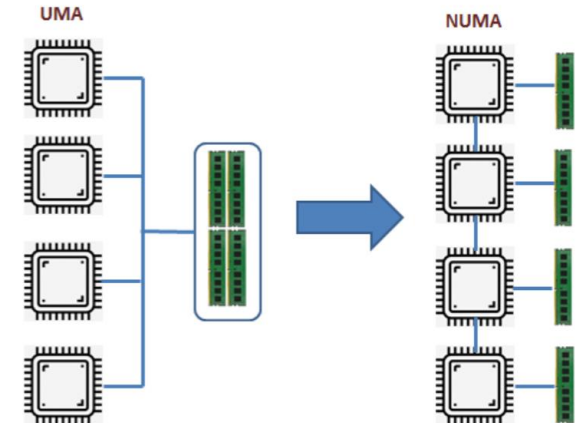
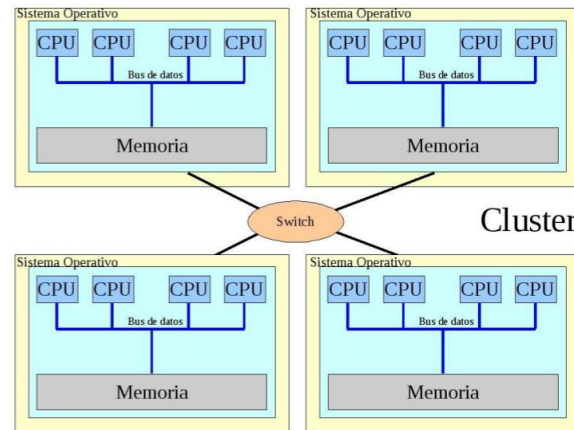
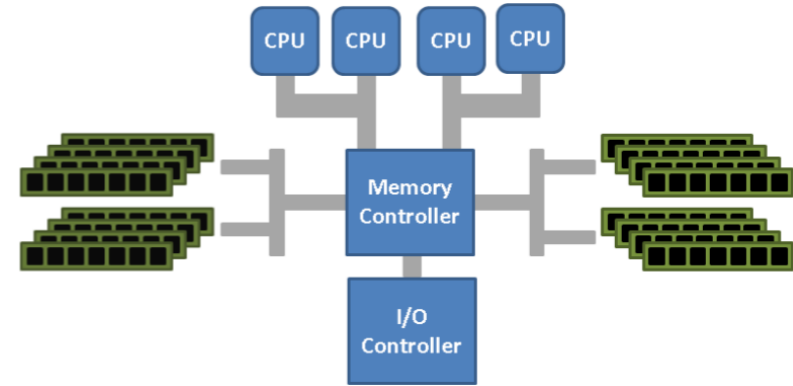
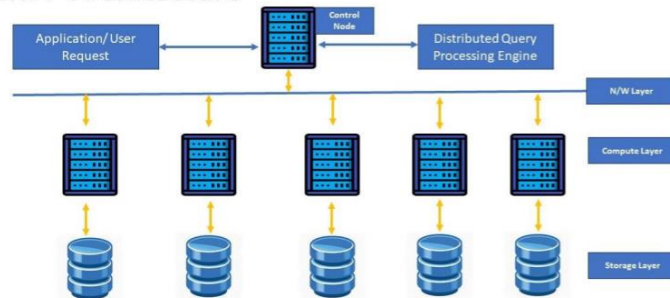
SMP (Symmetric Multiprocessing)

NUMA (Non-Uniform Memory Access)

MPP (Massively Parallel Processing)

Clusters de Procesadores

MPP Architecture



**"Las respuestas son importantes, pero las preguntas son aún más importantes". Nancy Willard.**

**Preguntas?**