



ARQUITECTURA Y SISTEMAS OPERATIVOS

CLASE 1

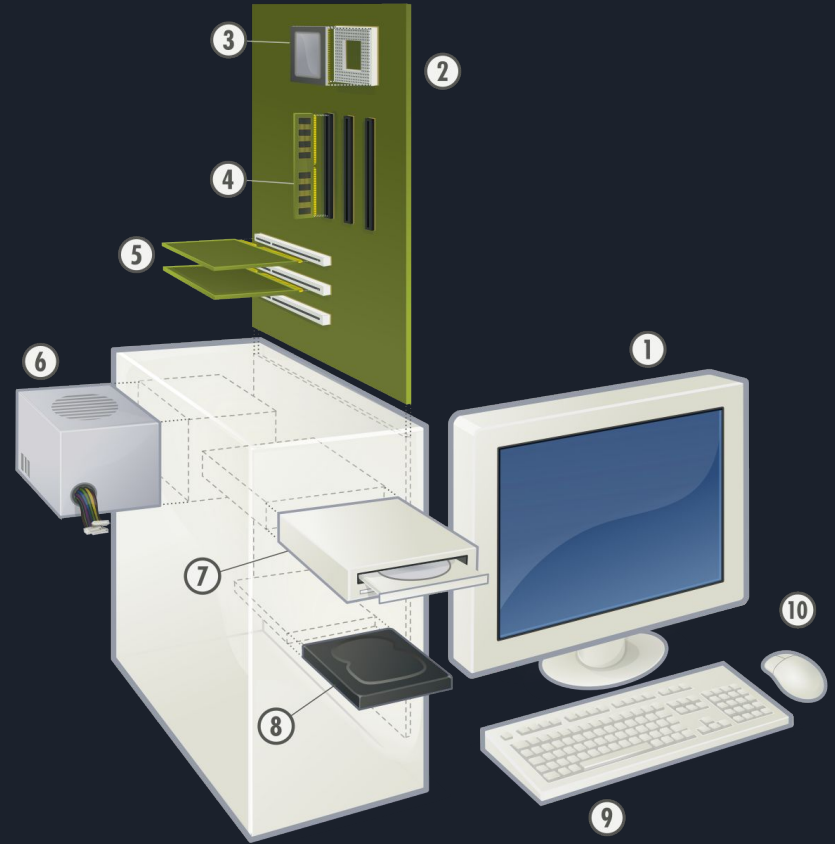
ARQUITECTURA DE COMPUTADORAS

HARDWARE

Hardware, se refiere a todos los componentes físicos que conforman una computadora.

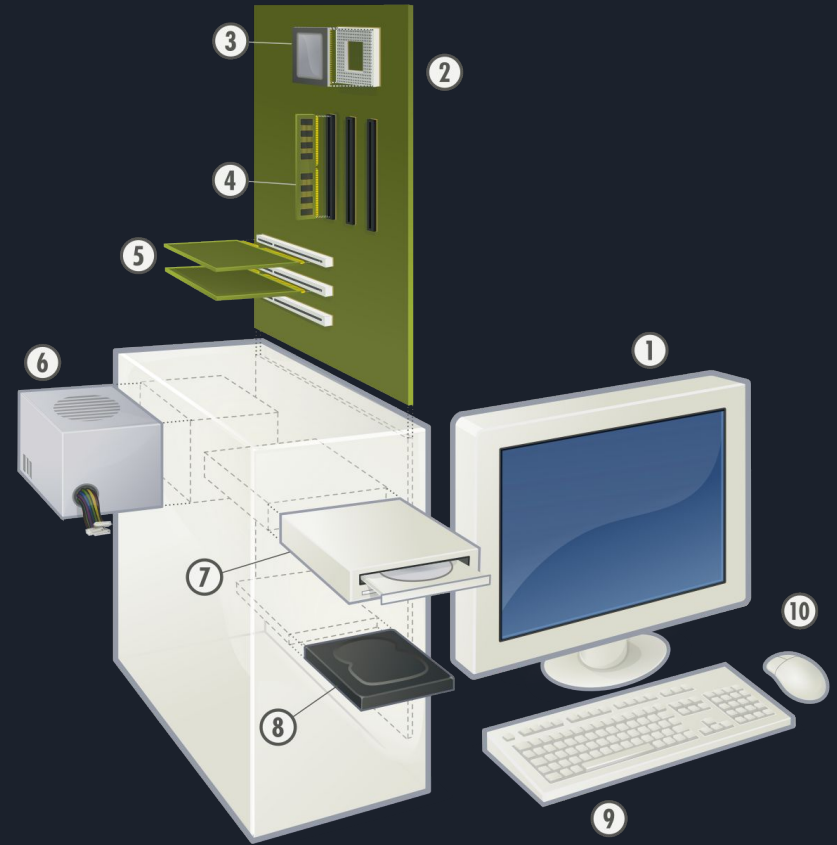
O

Todo lo que se puede tocar de una computadora, es llamado **Hardware**.

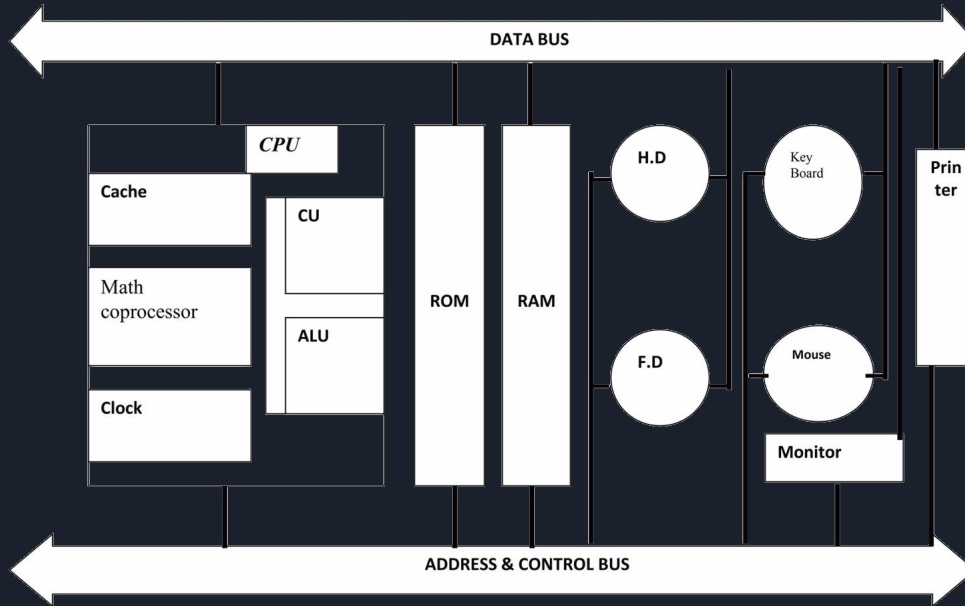


HARDWARE

- Monitor / Pantalla
- Teclado
- Mouse
- Mother
- CPU
- Memoria
- Dispositivos PCI (Ej: Placa de video)
- Fuente
- Discos



ARQUITECTURA DE COMPUTADORAS



Básicamente, Arquitectura de Computadoras define cómo debe funcionar el **Hardware**, es decir, cómo estos componentes deben conectarse y comunicarse entre sí.

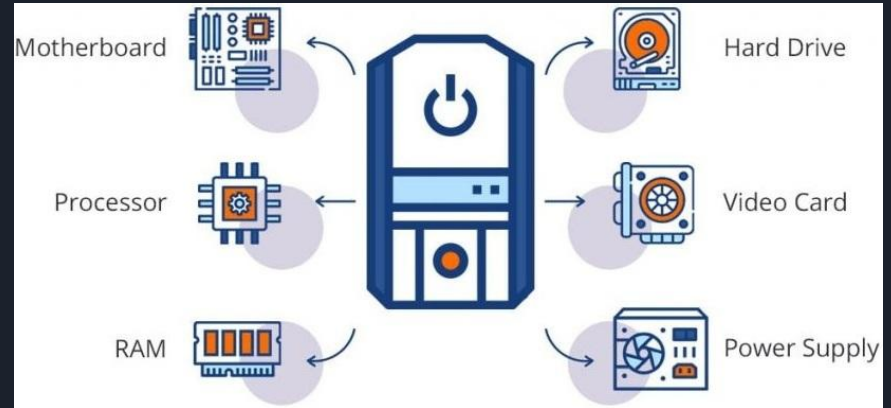


EQUIPO

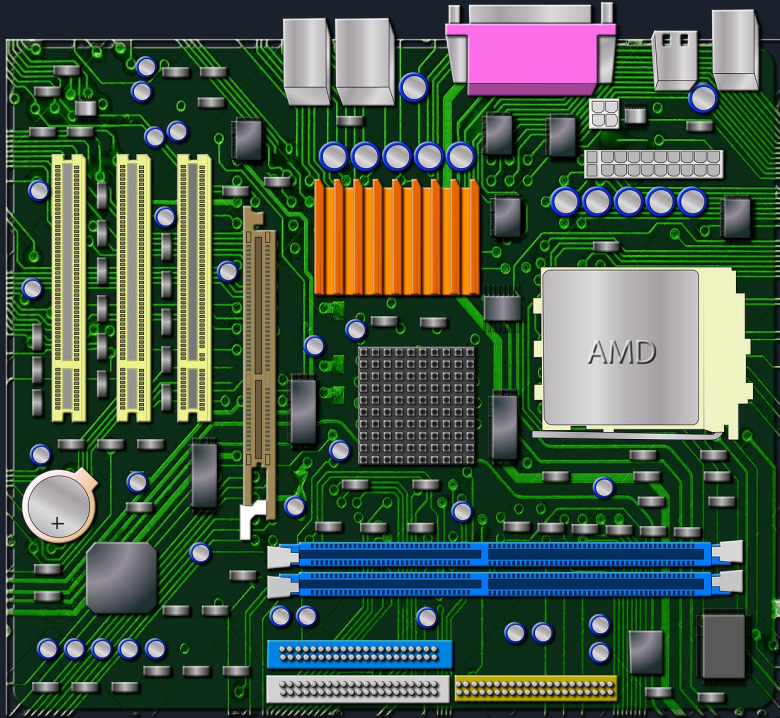
SYSTEM UNIT

Un **Sistema** se puede definir como un conjunto de componentes relacionados.

Así, un Equipo o *System Unit* es la parte de la computadora que contiene los principales dispositivos que realizan operaciones lógicas.



MOTHERBOARD



La *Motherboard* o **Placa Madre** es el principal componente en un sistema de computadora.

Porque básicamente, todos los dispositivos que conforman una computadora están conectados a la *mother* a través de diferentes puertos o interfaces.

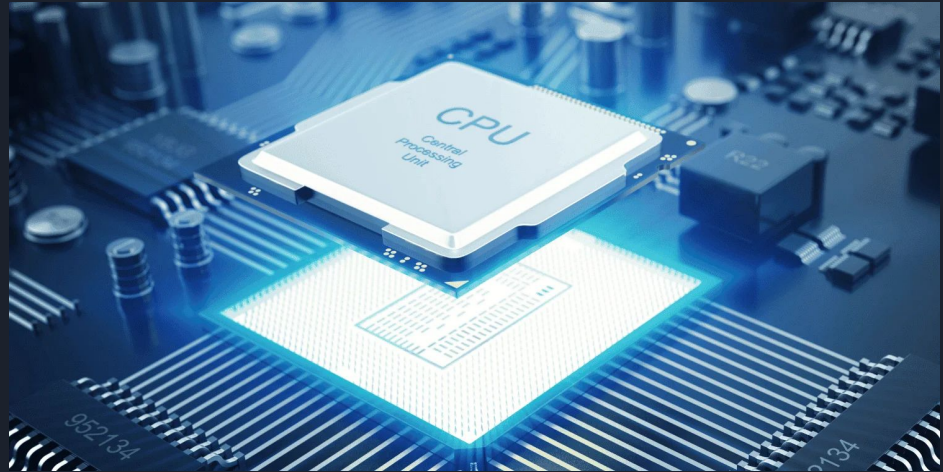


CPU

CENTRAL PROCESSING UNIT

Es el componente encargado de **ejecutar** las instrucciones y realizar cálculos que le permiten al software producir resultados para el usuario.

Cada tarea que realiza una computadora, desde su inicio hasta abrir una aplicación, pasa por del procesador.

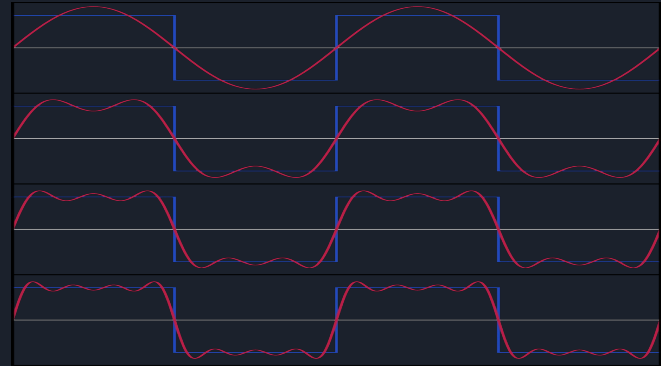
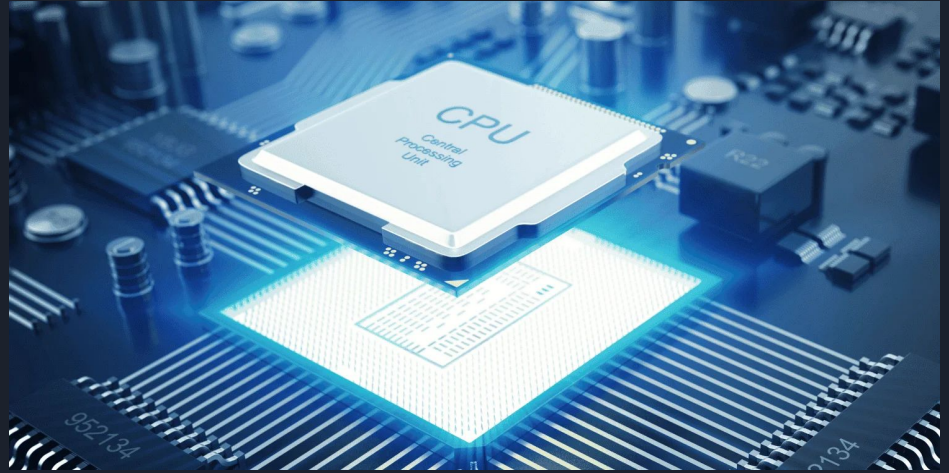


CPU CLOCK

Clock: La mayoría de las CPU son circuitos síncronos, esto es, utilizan una señal de reloj (clock) para regular sus operaciones secuenciales.

El clock es un circuito oscilador externo, que genera un número constante de pulsos por segundo, en forma de onda cuadrada periódica.

La frecuencia de los pulsos de reloj determina la velocidad a la que una CPU ejecuta las instrucciones.

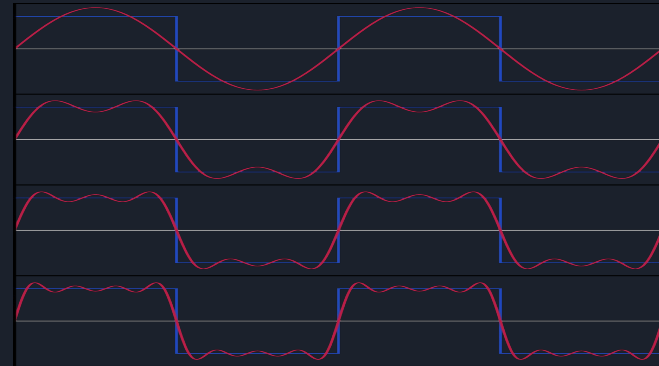
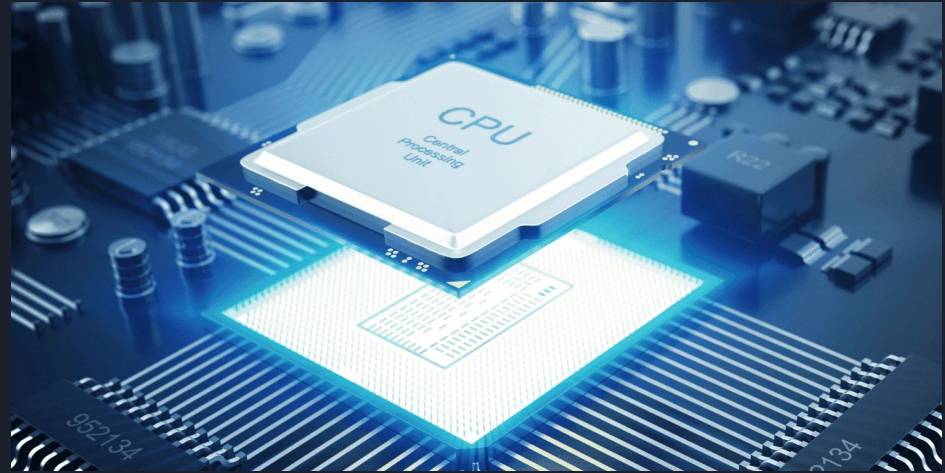


CPU CLOCK

1 Hertz = Un determinado evento tiene una frecuencia de 1 vez por segundo.

1 GHz = 1×10^9 (1 000 000 000) Hz.

Período de oscilación de 1 nanosegundo.

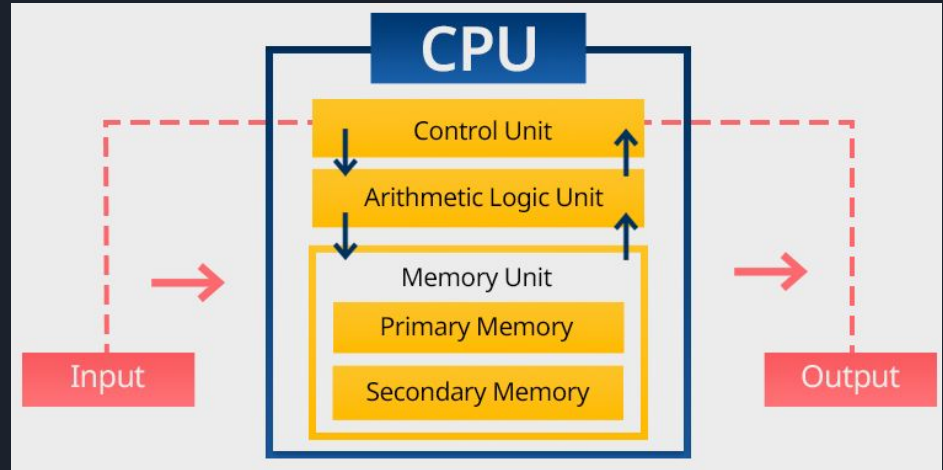


CPU

CU Y ALU

Control Unit (CU): controla el conjunto de operaciones de la computadora. Busca (*Fetch*) instrucciones de la memoria principal (RAM), las interpreta, busca los datos necesarios y le indica a la **ALU** qué debe hacer.

Arithmetic Logic Unit (ALU): se encarga de manejar las funciones lógicas y aritméticas dentro del CPU. Toma las instrucciones decodificadas por la CU y realiza las operaciones correspondientes.



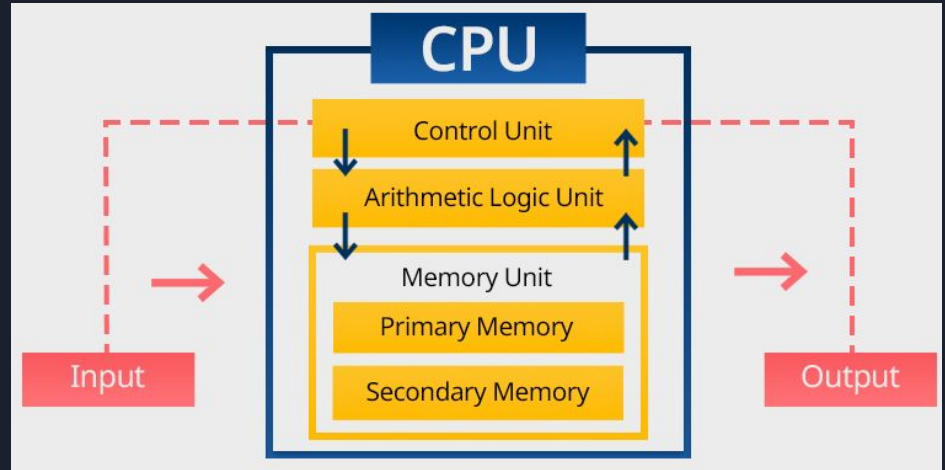
CPU

REGISTROS

Registros: Son locaciones de almacenamiento de *propósito específico* de alta velocidad. Almacena temporalmente los datos en uso por la CPU.

El **Program Counter (PC)** es un registro que almacena la dirección de memoria de la siguiente instrucción a ejecutar.

El **Current Instruction Registry (CIR)** almacena temporalmente la instrucción actual para que la CPU la decodifique y produzca las señales de control correctas para la etapa en ejecución.

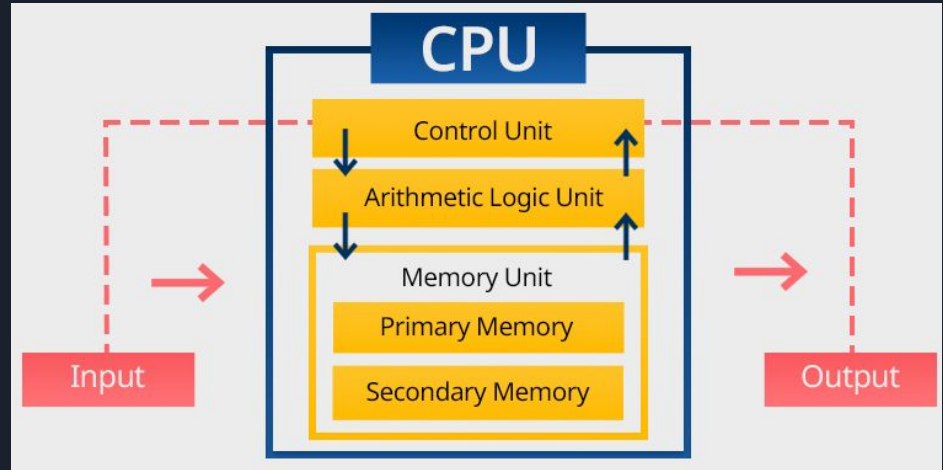


CPU

REGISTROS

Memory address register (MAR): se encarga de almacenar la dirección donde se ubica la instrucción en memoria. Cuando inicia una señal de lectura, la instrucción en la dirección del MAR se lee y se almacena en el *Memory data register (MDR)*.

Memory data register (MDR): se encarga de almacenar temporalmente las instrucciones leídas desde la dirección del MAR. Funciona como un registro bidireccional, porque puede enviar la salida de la memoria a la CPU o de la CPU a la memoria.

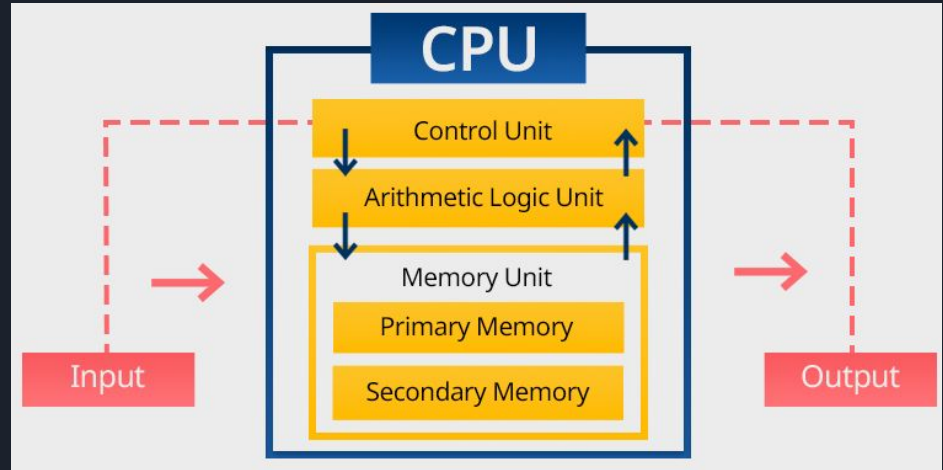


CPU

CACHE Y BUSES

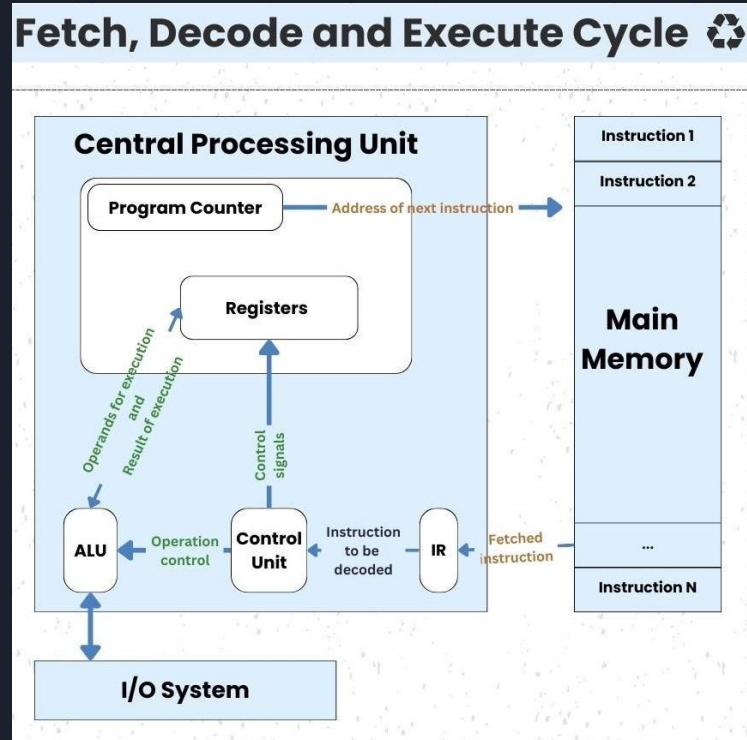
Cache: Memoria interna de la CPU que le permite almacenar temporalmente datos e instrucciones a los que accede con frecuencia. De esta manera los accede mucho más rápido que si tuviera que ir a buscarlos a la memoria principal.

Buses: es un grupo de conductores eléctricos que transportan señales de la computadora de un lado a otro.



CPU

INSTRUCTIONS CYCLE

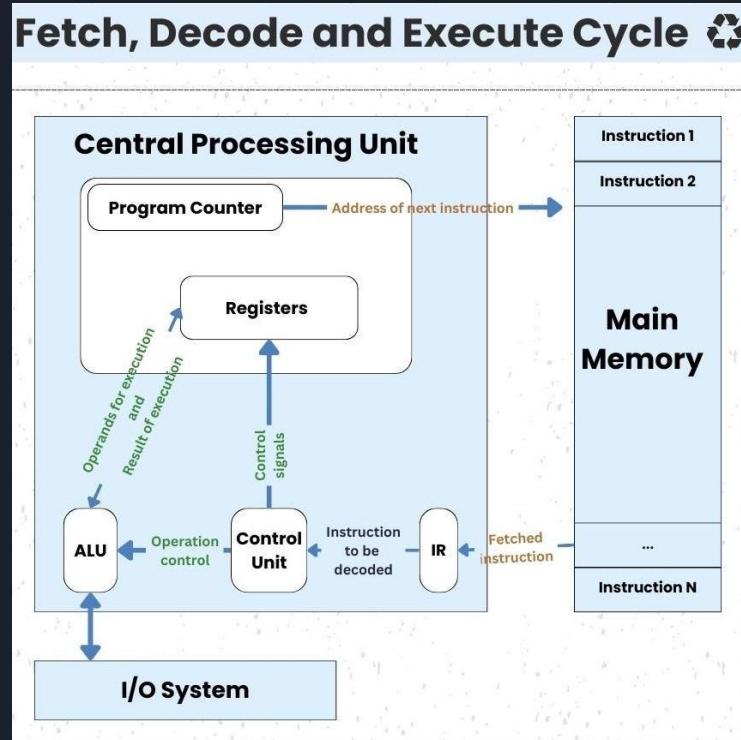


Los CPU pueden tener diferentes ciclos dependiendo de los distintos **conjuntos de instrucciones**, pero en general:

- FETCH
- DECODE
- EXECUTE
- MEMORY ACCESS (if needed)
- REGISTRY WRITE-BACK (if needed)

CPU

INSTRUCTIONS CYCLE



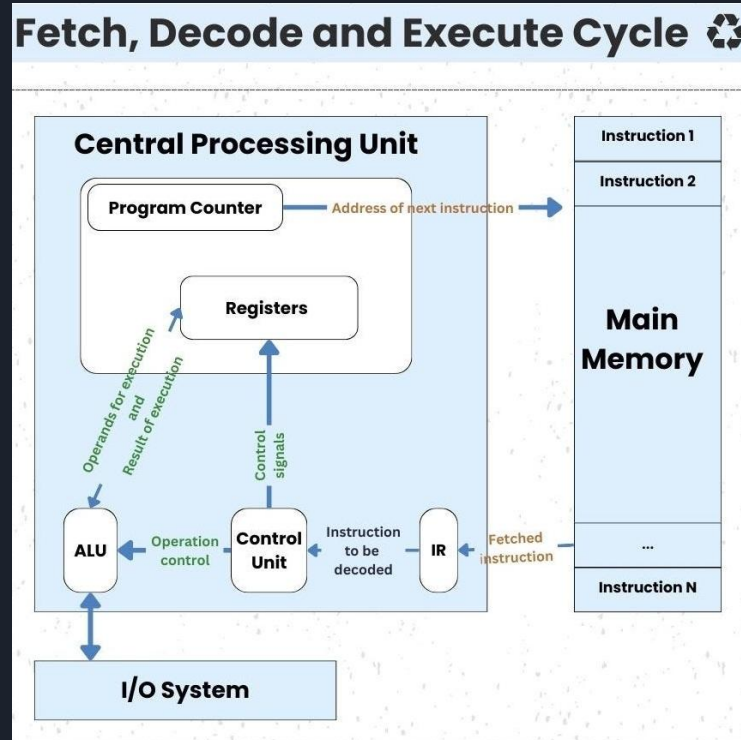
Fetch: el ciclo inicia recuperando la siguiente instrucción a ejecutar.

Se consulta al PC la dirección de la instrucción en memoria (utilizando la MAR), y luego, la instrucción pasa desde el MDR al CIR.

Al final de esta etapa, el PC apunta a la siguiente instrucción que se leerá en el siguiente ciclo.

CPU

INSTRUCTIONS CYCLE

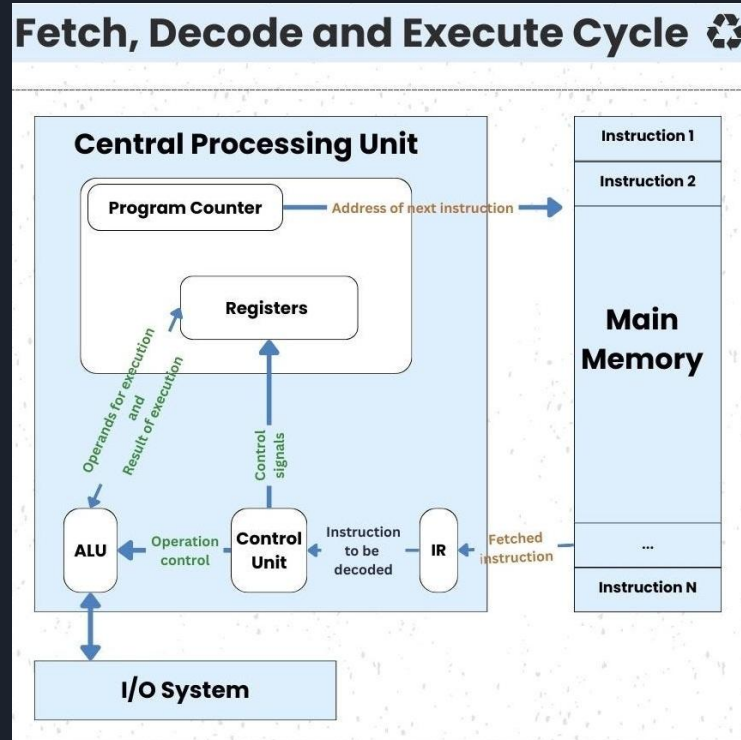


Decode: en esta etapa, la instrucción codificada en el CIR es interpretada por la CU.

Determina qué operaciones y operandos adicionales se requieren para la ejecución y envía las señales correspondientes a los componentes dentro de la CPU para preparar la ejecución de la instrucción.

CPU

INSTRUCTIONS CYCLE



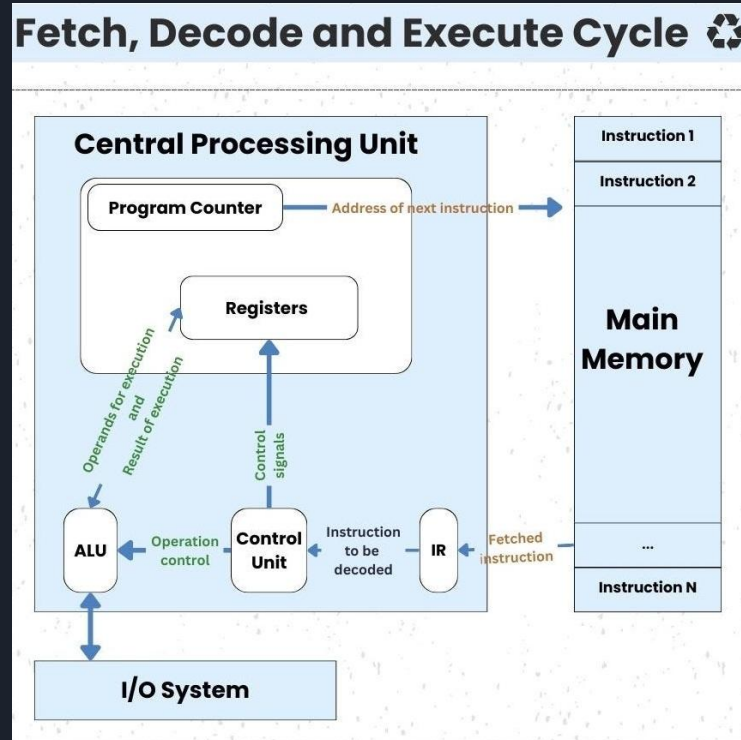
Execute: las unidades funcionales relevantes de la CPU realizan la operación especificada por la instrucción.

La ALU puede ejecutar operaciones lógicas o aritméticas, leer o escribir datos en la memoria, y los resultados se almacenan en registros o memoria según lo requiera la instrucción.

Repetición del ciclo.

CPU

INSTRUCTIONS CYCLE



Memory Access: La etapa de acceso a memoria se utiliza para recuperar los datos necesarios para ejecutar una instrucción.

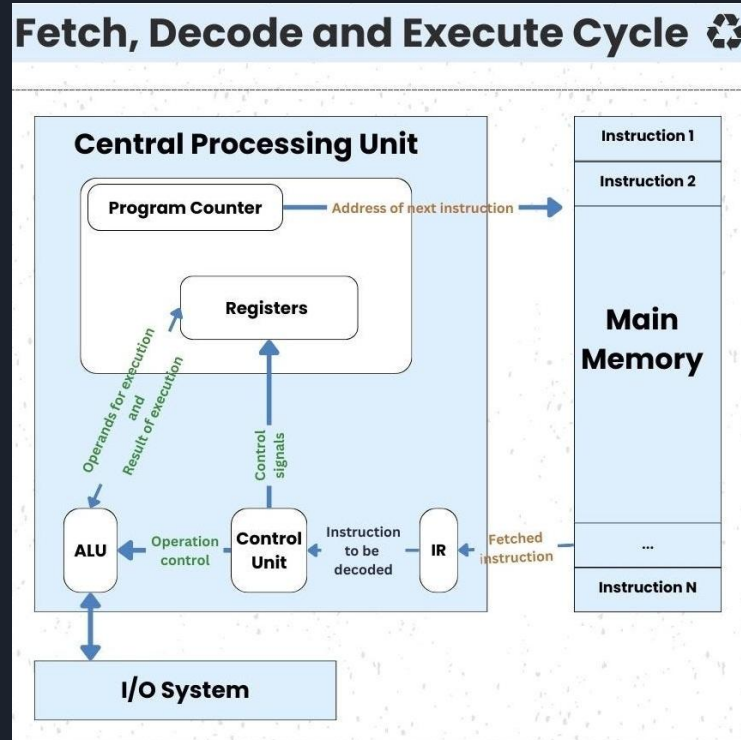
Esta etapa solo ocurre si la instrucción requiere datos de la memoria.

Por ejemplo:

- $x = 5$
- $y = x + 3$

CPU

INSTRUCTIONS CYCLE



Registry write-back: La etapa de reescritura del registro se utiliza si la ejecución de la instrucción afecta a los datos.

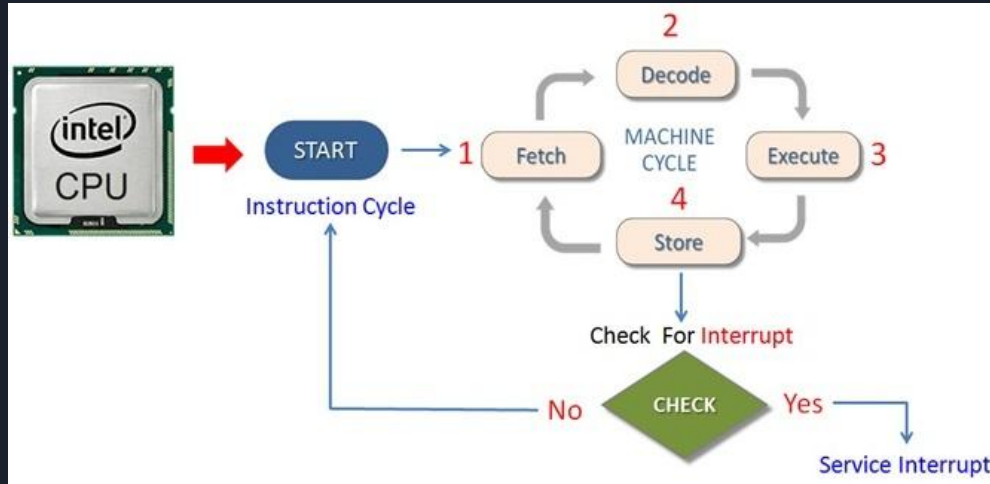
Esta es otra etapa que no siempre forma parte del ciclo.

Por ejemplo:

- $x = 5$
- $y = x + 3$

CPU

INSTRUCTIONS CYCLE - INTERRUPTIONS

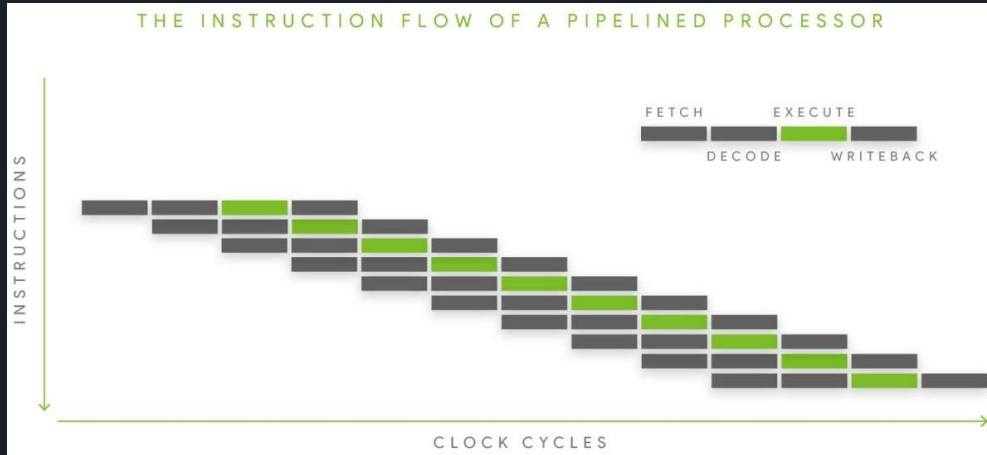


Además, en la mayoría de los procesadores, pueden producirse **interrupciones**.

Esto hace que la CPU salte a una rutina de servicio de interrupciones, la ejecute y luego regrese a la instrucción que iba a ejecutar.

En algunos casos, la instrucción puede interrumpirse a mitad de proceso, pero no tendrá ningún efecto y se volverá a ejecutar tras el retorno de la interrupción.

CPU PIPELINE



Pipeline, es una técnica para implementar paralelismo a nivel de instrucciones dentro del procesador.

Intenta mantener cada parte del procesador ocupada con alguna tarea, dividiendo las instrucciones entrantes en una serie de pasos secuenciales, cada tarea es ejecutada por diferentes unidades de procesamiento, logrando que diferentes partes de la instrucciones sean procesadas en paralelo.

RAM

RANDOM MEMORY ACCESS

Es un dispositivo de memoria de acceso aleatorio.

Permite **leer** o **escribir** datos en casi la misma cantidad de tiempo, independientemente de la ubicación física de los datos dentro de la memoria.

Hay otros medios de almacenamiento de acceso directo, donde el tiempo requerido para leer y escribir varía significativamente dependiendo de sus ubicaciones físicas en el medio, debido a limitaciones mecánicas. Por ejemplo: las velocidades de rotación.



RAM

RANDOM MEMORY ACCESS

Es la memoria principal del sistema dado que el CPU puede comunicarse directamente.

Los programas de aplicación, para poder ejecutarse deben estar cargados en memoria.

Todos los datos o instrucciones que forman parte de un programa se almacenan en la RAM para que el CPU pueda procesarlas.

Es almacenamiento volátil. Es decir, pierde energía, y también sus datos.



RAM

DYNAMIC RANDOM MEMORY ACCESS

Dynamic RAM: es un tipo de memoria RAM conformada por un transistor y un capacitor que pierde carga con el tiempo, por lo que requiere de un mecanismo externo de actualización que re-escriba los datos.

Tiene la ventaja de ser económica, pero al mismo tiempo más lenta por el mecanismo de actualización.



RAM

STATIC RANDOM MEMORY ACCESS

Static RAM: es un tipo de RAM que utiliza circuitos *latch* (flip-flop) para almacenar de manera permanente cada bit.

La SRAM es una memoria volátil; los datos se pierden al desconectar la alimentación.

Es más costosa y ocupa mayor espacio físico, pero es más rápida que la DRAM.

La memoria Caché es un ejemplo de SRAM.



ROM

READ ONLY MEMORY

Read Only Memory (ROM): es un tipo de memoria donde los datos, una vez escritos, son permanentes. No pueden ser modificados y no se pierden al desconectar la energía.

Electrically Erasable Programmable Read Only Memory (EEPROM). Es una memoria ROM que puede ser modificada.



ROM

READ ONLY MEMORY

Ejemplo: **BIOS** (Basic Input Output System). Es un programa integrado en un chip del *mother* que reconoce y controla los diversos dispositivos que lo componen. Su propósito es garantizar el correcto funcionamiento de todos los componentes conectados a la computadora.



SECONDARY STORAGE

EXTERNAL STORAGE

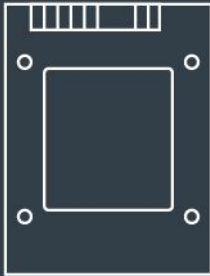
Se refiere al almacenamiento de datos no volátil fuera del hardware interno de una computadora y, por lo tanto, se puede desconectar fácilmente.

Estos dispositivos de almacenamiento pueden ser medios extraíbles, unidades flash (USB y tarjetas de memoria), dispositivos de almacenamiento portátiles o almacenamiento conectado a red.

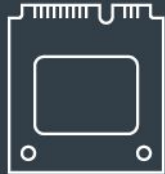
El almacenamiento en la nube basado en la web es la tecnología más reciente para el almacenamiento externo.



HDD



SATA SSD



mSATA SSD



M.2 SSD



eMMC

Muchas Gracias

Jeremías Fassi

Javier E. Kinter

