



Sistemas Informáticos

Unidad 1

Introducción

ÍNDICE

1. **Introducción a sistemas informáticos**
2. **Sistema binario**
3. **Hexadecimal**
4. **Historia de ordenadores**
5. **Electricidad**
6. **Sistemas digitales**
 - a. **Boole**
 - b. **Puertas Lógicas**
 - c. **Tablas de Verdad**
7. **Arquitectura de computadoras**

Sistema informático

Definición

Un sistema informático es un conjunto interconectado de componentes que trabajan con datos binario para:

- Procesar
- Almacenar
- Transmitir

Estos componentes pueden incluir **hardware** (como computadoras, dispositivos de almacenamiento, periféricos) y **software** (sistemas operativos, aplicaciones, programas), así como también redes de comunicación que permiten la transferencia de datos entre diferentes componentes.



Niveles en un sistema informático estándar.

Nota: Existen sistemas que carecen de sistema operativo.

Sistema Binario

Sistema numérico que solo puede representar dos valores distintos (también llamados estados) **¡NO a la vez!**

bit: Unidad mínima de información binaria

Proviene de **B**inary **D**igit

Byte: 8 bits*

bit se escribe
en minúscula
Byte con
mayúscula

También SÍ-NO, VERDADERO-FALSO,
ENCENDIDO-APAGADO según aplicación.



En sistemas informáticos se asocian con el 1 o el 0. Se trata de una abstracción para representar fenómenos físicos. Ejemplos:

- Alto o bajo voltaje (0-5 voltios) en un circuito integrado
- Norte o Sur en un disco duro (siendo el sentido del campo magnético)
- Posición de un interruptor físico
- Conmutación de un transistor según el voltaje de entrada
- Agujeros en un papel (tarjeta perforada)
- El código Morse (puntos y rayas)

No es binario en sentido estricto porque hace falta el espacio (por lo que es *ternario*)

¿Sabías que?

Con una luz, como un láser o LED se pueden emitir pulsos, apagado (0) o encendido (1), para transmitir información.

Conversión

Para convertir un número **decimal a binario** existen diferentes técnicas:

- Dividir por 2 consecutivamente el número
- Sumar el valor de los pesos hasta obtener el número

Para conocer el número bits necesarios:

Calcular el \log_2 (logaritmo en base 2) del *número en decimal*:

$$\text{nº bits} = \log_{10}(\text{número}) / \log_{10} 2$$

Como: $\log_{10} 2 = 0.3$

$$\text{bits} = \log_{10}(\text{número}) * 3.33$$

Binario a Decimal

- Sumar los pesos (potencia de 2 según posición) multiplicando por 1.

¡Ojo que la posición empieza en 0!

Ejemplo: 10101₂

$$1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 1*16 + 0*8 + 1*4 + 0*2 + 1*1 = 21_{10}$$

CONVERSIÓN

8 bits = 1 byte = 256 decimal

4 bits = 1 hex

2 Hex digits = 1 byte

Hex digits = nº Bytes/2

¡OJO! Esto es para los estados de un sistema. En almacenamiento-memoria NO aplica igual

Relación Posición- Peso bits

Posición	7	6	5	4	3	2	1	0
Potencia	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Peso	128	64	32	16	8	4	2	1

Sistema Hexadecimal

El sistema hexadecimal como su nombre indica es en base 16.

Permite representar la información de información más **comprimida** y **legible**.

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

La conversión a binario es sencilla tanto para los humanos como para las máquinas por lo que es muy útil en computación.

Al representar 16 valores, cada dígito se corresponde con 4 bits. Ejemplos:

$$AE_{16} = 10 \cdot 16^1 + 14 \cdot 16^0 = 10 \cdot 16 + 14 \cdot 1 = 174_{10}$$

1 0 1 0

1 1 1 0

$$B_{16} = 11_{10} = 1011_2$$

$$FF_{16} = 255_{10} = 1111 \ 1111_2$$

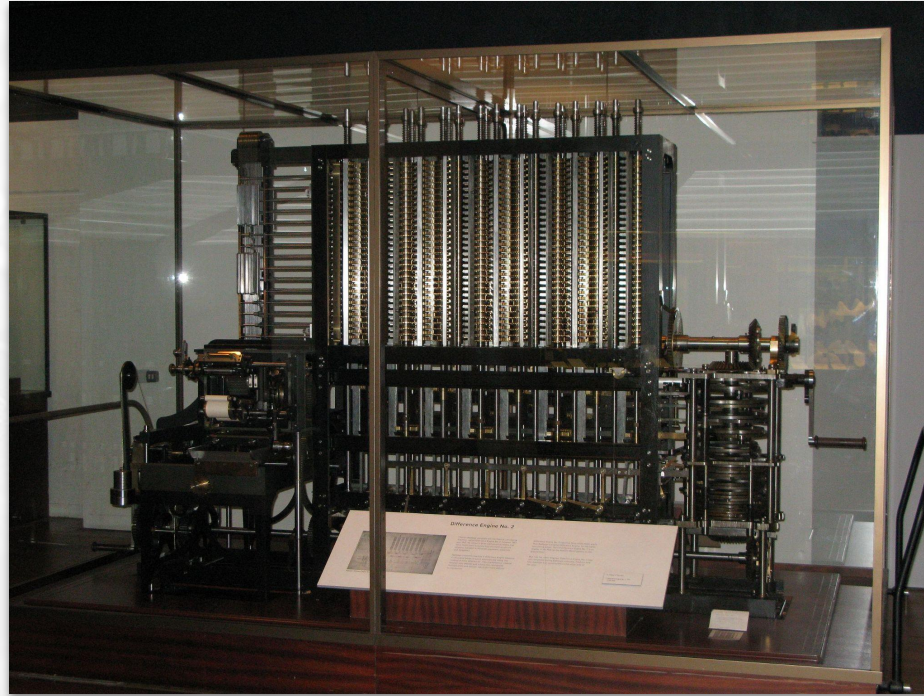
Nota: El Hexadecimal se lee/escribe con **0x** delante.

Ejemplo: 0xFFAA25

De esta manera el ordenador sabe que es hexadecimal.

decimal	hexa
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

Historia de Ordenadores



<https://padlet.com/pedro275/cronolog-a-ordenadores-nnew4y8ejl273djj>

Electricidad

Conjunto de fenómenos físicos relacionados con la **presencia y flujo** de **cargas eléctricas**

Para que exista corriente eléctrica, los **electrones** de la **última capa de valencia** del átomo deben estar **libres** (que cambia según el elemento) y debe de existir un **circuito**. La corriente aparece cuando se aplica un **voltaje**.

Según conducen o no, los materiales se clasifican en:

Aislantes: aire, madera, vidrio, caucho, plástico, etc.

Conductores: hierro, oro, cobre, etc.

Nota: Existen gases y conductores electrolíticos que conducen electricidad.

Semiconductores: silicio, germanio, galio, fósforo, azufre, etc.

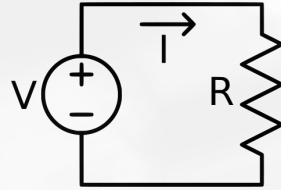
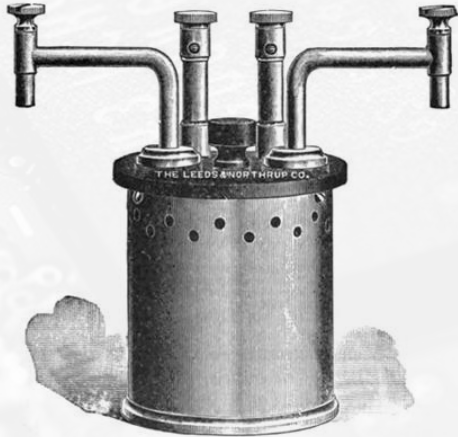
Nota: No todos son metales

DC (Corriente Continua): Los electrones (la corriente) se mueven en un sentido

AC (Corriente Alterna): Los electrones (la corriente) se mueven en ambos sentido

Ley de Ohm

$$V = I * R$$



El **sentido convencional de la corriente eléctrica** va desde el **polo positivo** al **polo negativo** de una fuente de energía eléctrica, pero en la realidad, los electrones, que son las partículas que transportan la carga eléctrica, se mueven en **sentido contrario**, desde el **polo negativo al polo positivo**.

Esta convención se estableció antes de que se conociera la naturaleza de los electrones y se ha mantenido por razones históricas y de conveniencia en la mayoría de las aplicaciones eléctricas y electrónicas.

Tensión, Voltaje, Diferencia de potencial: Se mide en Voltios (V)
Cantidad de energía eléctrica por unidad de carga disponible en un circuito

Intensidad de corriente: Se mide en Amperios (A)
Es el flujo de electrones a través de un conductor.

- Continua (DC)
- Alterna (AC)

Resistencia: Se mide en ohmios (Ω)



Como despejar ley de Ohm

Analógico Vs Digital

DIGITAL

- **Ventajas:** robusto al ruido* y estable, exacto-repetible (corrección de errores), reprogramable (mayor flexibilidad de uso), compresión de los datos, encriptación

Implican un mejor almacenamiento y transmisión de la señal

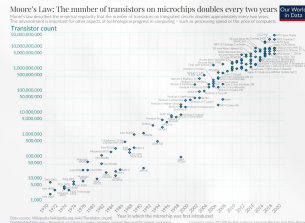
- **Desventajas:** el muestreo de la señal implica reducción de la “calidad” al ser **discreto**, mayor complejidad de circuitería, mayor latencia (tiempo de procesamiento)

ANALÓGICO

- **Ventajas:** menor latencia, señal **continua** que representa más fielmente las señales físicas
- **Desventajas:** las contrarias a las ventajas del digital

Transistores

Ley de Moore: se duplica el número cada dos años (está llegando a su fin->átomo tamaño)



Circuitos integrados aka Chips

Lógica Booleana

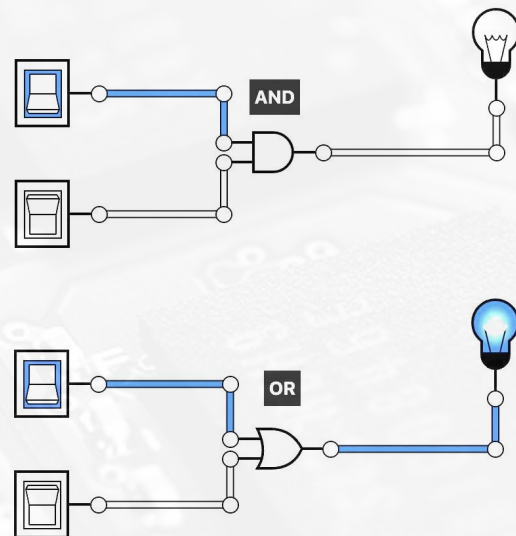
Tablas de verdad

Puertas Lógicas

OR		
A	B	Salida
0	0	0
0	1	1
1	0	1
1	1	1

AND		
A	B	Salida
0	0	0
0	1	0
1	0	0
1	1	1

NOT	
Entrada	Salida
0	1
1	0



Niveles de abstracción

A más alto, más cercano al lenguaje humano

software

hardware

Lenguajes de alto nivel

python, java,
javascript, c#

Lenguajes de medio/alto nivel

c, c++, go, fortran

Lenguaje simbólico. Mnemónicos de
instrucciones: Ej. *mov*

ensamblador

Lenguaje Máquina

Hexadecimal
0xFF00AF

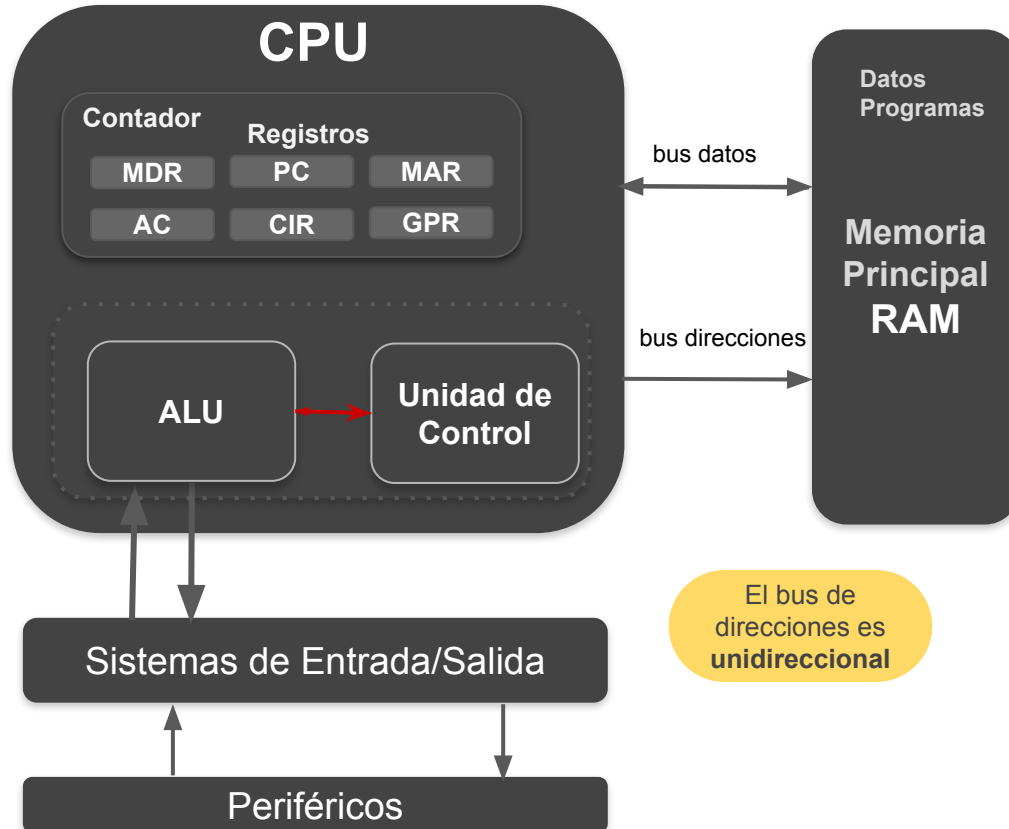
Binario

1010101

Los lenguajes de “**medio**” nivel tienen
“**acceso**” al hardware

Arquitectura Von Neumann (básica)

Actualmente sigue en uso pero con cambios



CPU

- [Registros](#)
- Unidad de control
- ALU

Memoria

Bus de Datos y Direcciones

CPU (Unidad Central de Procesamiento)

Interpreta las **instrucciones** de un programa informático.

En la actualidad, los **microprocesadores** están constituidos generalmente por un único circuito integrado (chip), aunque existen procesadores multinúcleo que integran múltiples núcleos de procesamiento en un solo circuito integrado

NOTA: La CPU es esencialmente un microprocesador pero NO todos los microprocesadores son CPU.

Componentes de la CPU

- **ALU (unidad Aritmético-Lógica)**
- **Unidad de Control**
- **Registros**



ALU

Unidad Aritmético Lógica

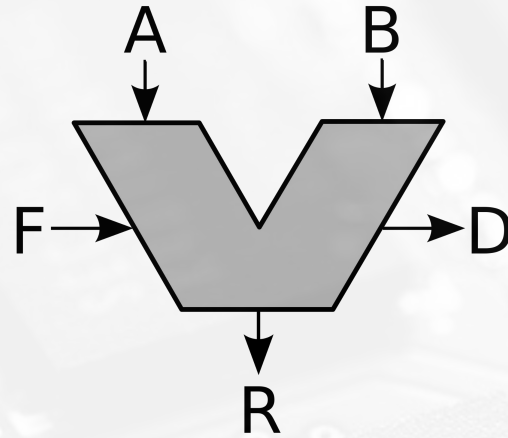
Operaciones Aritméticas: realiza operaciones aritméticas, como sumar, restar, multiplicar y dividir números; así como desplazamiento de bits . Cuando se le da una instrucción para realizar una operación aritmética, la ALU toma los datos de los registros de la CPU o de la memoria, realiza la operación deseada y coloca el resultado nuevamente en un registro o en la memoria.

Operaciones Lógicas: Además de las operaciones matemáticas, la ALU también puede realizar operaciones lógicas, como AND, OR, NOT y XOR. Estas operaciones se utilizan para realizar comparaciones y tomar decisiones en el procesamiento de datos. Por ejemplo, se pueden usar para verificar si dos números son iguales o para determinar si se cumple una condición lógica.

Los **Registros** de la ALU almacenan temporalmente los datos y los resultados de las operaciones. Esto le permite realizar cálculos de manera eficiente y mantener un seguimiento de los valores mientras se procesan.

La ALU no opera por sí sola; está controlada por la **Unidad de Control** de la CPU. La Unidad de Control se encarga de decirle a la ALU qué operación realizar y en qué datos trabajar. También coordina el flujo de datos entre la ALU, la memoria y otros componentes de la CPU.

nota: Una CPU puede tener más de una ALU



A, B: Operandos (entrada)
F: Opcode de Unidad de Control
D: Flags de Estado de Salida
R: Salida

Tamaño de la ALU: puede variar según la arquitectura de la CPU. Generalmente está diseñada para manejar un cierto número de bits a la vez. Por ejemplo, una ALU de 32 bits es capaz de realizar operaciones en números de 32 bits a la vez. La de 64 bits con 64.

Operaciones aritméticas en binario

Suma

Resta

Multiplicación

División

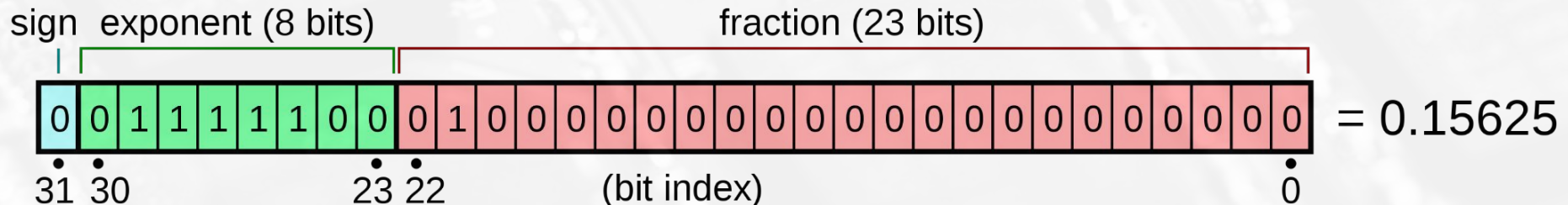
Representación de números negativos

Complemento a 1

Complemento a 2

*“La ALU no resta números binarios, **suma binarios negativos**, por eso esta conversión al negativo”*

Coma flotante



Unidad de Control

Una **instrucción** en una CPU es una orden específica que guía la ejecución de operaciones y tareas en el procesador de una computadora. Estas instrucciones son la base de la programación de bajo nivel (lenguaje máquina→ unos y ceros) y son fundamentales para el funcionamiento de cualquier computadora. Diferentes CPU tienen sus propias instrucciones, que se especifican detallan en **ISA** (*instruction set architecture*)

Requiere de un **clock** (reloj) que marca la velocidad de ejecución de las instrucciones. En cada ciclo, se recupera una instrucción de memoria (una instrucción puede tardar más de un ciclo), se decodifica para comprender qué hacer, se ejecuta la operación y, finalmente, se actualizan los registros o la memoria

Ciclo de instrucción

1. **Fetch** (Búsqueda): La CPU obtiene la instrucción desde la memoria principal en la dirección especificada por el contador de programa.
2. **Decode** (Decodificación): La CPU interpreta la instrucción para comprender qué operación debe realizar y qué datos debe usar.
3. **Execute** (Ejecución): La CPU realiza la operación especificada por la instrucción en sus registros internos y/o en la memoria.
4. **Writeback** (Escritura): Si es necesario, los resultados de la operación se almacenan en registros internos o en la memoria.

Conjunto de Instrucciones

- Transferencia de información
- Aritmético-Lógica y desplazamiento
- Transferencia de control (condicionales, bifurcaciones, etc)

ARQUITECTURAS

CISC

Compuestas por instrucciones complejas compuestas de otras más simples. Esta arquitectura requiere la Unidad de Control estar reprogramada con memoria de control.

El tamaño de instrucción difiere entre ellas. La multitarea es “aparente”.

Ej. x86

RISC

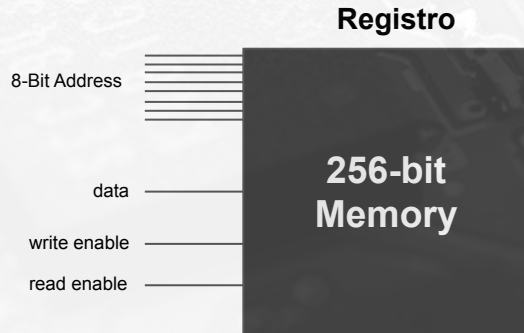
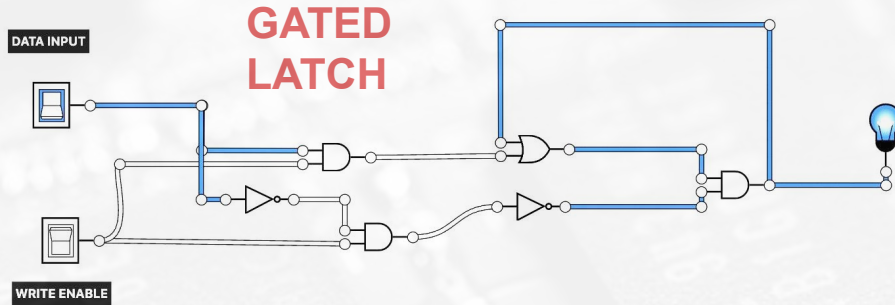
Compuestas por instrucciones reducidas. Menos instrucciones y más simples. Son implementadas en el circuito, es decir, las Unidades de Control son cableadas.

Solo las instrucciones de carga y almacenamiento acceden a la memoria de datos.

En este caso el tamaño de instrucción es el siempre el mismo. Permite multitarea “real”.

Ej. ARM

Registros y Memoria

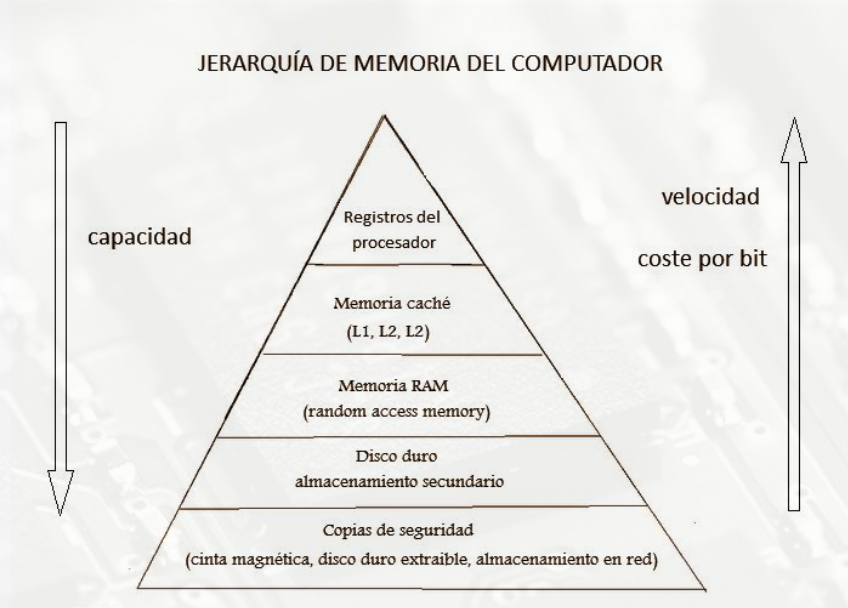


Esto es para arquitectura de 32bits $\rightarrow 32 \times 8\text{-bit address} = 256 \text{ bits}$

RAM

8 Bit Data	Address		Data
	0	1	
	2	3	
	4	5	
8 Bit Address Input	6	7	
	8	8	
	9	10	
	11	12	
Read enable	13	14	
Write enable	15		
	255		

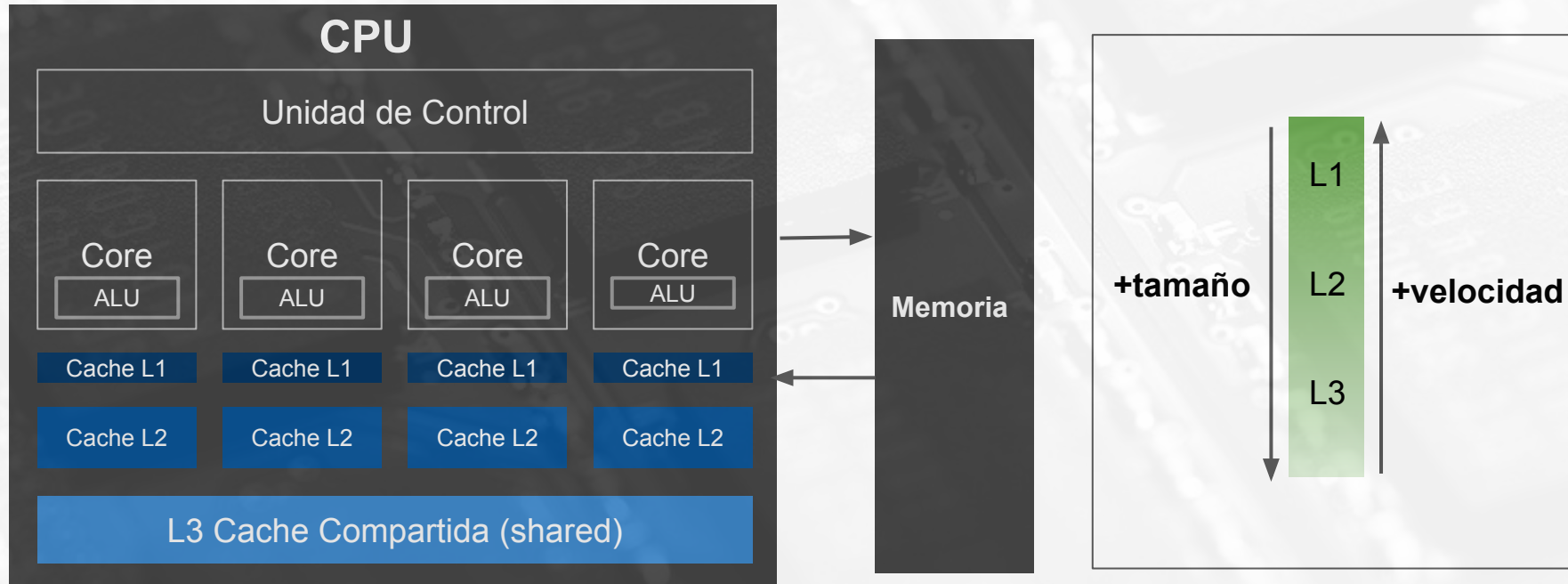
Jerarquía de memoria



https://es.wikipedia.org/wiki/Jerarqu%C3%ADa_de_memoria

Memoria Caché

La memoria caché se utiliza para mejorar el rendimiento de la CPU. Al almacenar los datos y las instrucciones que la CPU va a necesitar en un futuro inmediato, la memoria caché evita que la CPU tenga que acceder a la memoria principal, que es más lenta (100x aprox comparada con la caché más rápida). Diferentes niveles: **L1**, **L2**, **L3**. La caché L3 es compartida por los diferentes núcleos (core). Existe un L4 en la que Intel está trabajando



Microprocesador

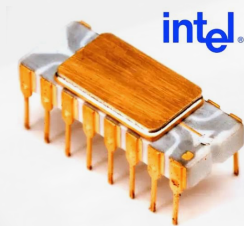
Circuito integrado central de un sistema informático

Un microprocesador es un circuito integrado (chip) que contiene la CPU de un ordenador (o más de una). Es el componente principal de un ordenador y se encarga de realizar todas las operaciones de cálculo y control necesarias para que el ordenador funcione.

Se clasifican en función de su arquitectura, su velocidad de reloj y el número de núcleos que tienen.

- Intel 4004 (en la imagen) fue el primer microprocesador dentro de un chip de uso general.

- 4 bits
- Frecuencia de reloj de 750 KHz.
- 16 pines
- formado por 2.300 transistores



Se diseñó primero para uso en calculadoras

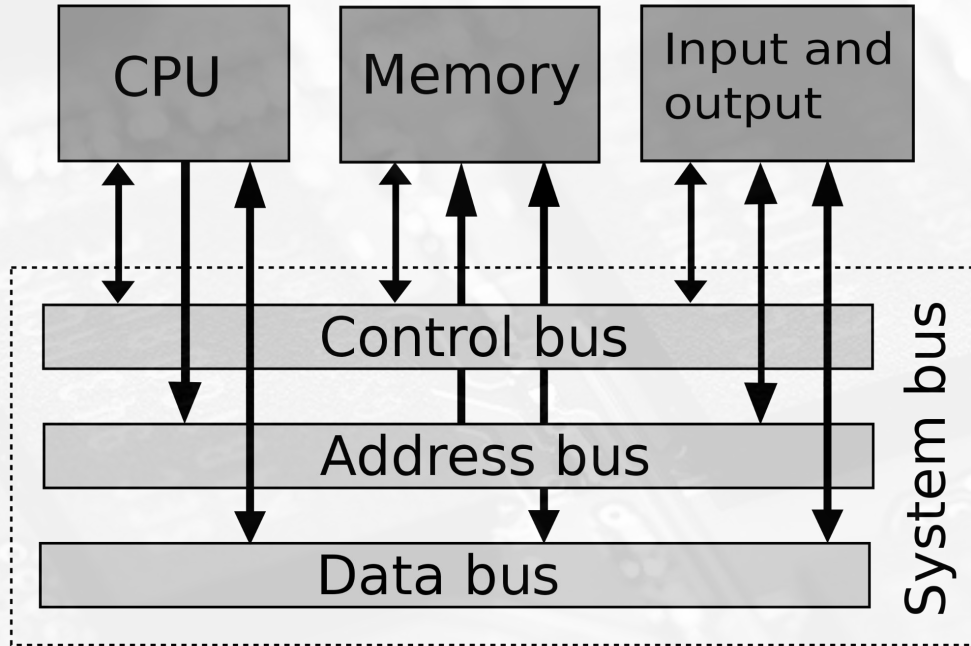


El ALTAIR 8800 fue el primer ordenador en usarlo



Busicom 141-PF. Primer dispositivo en usar el chip.

BUSES DE DATOS

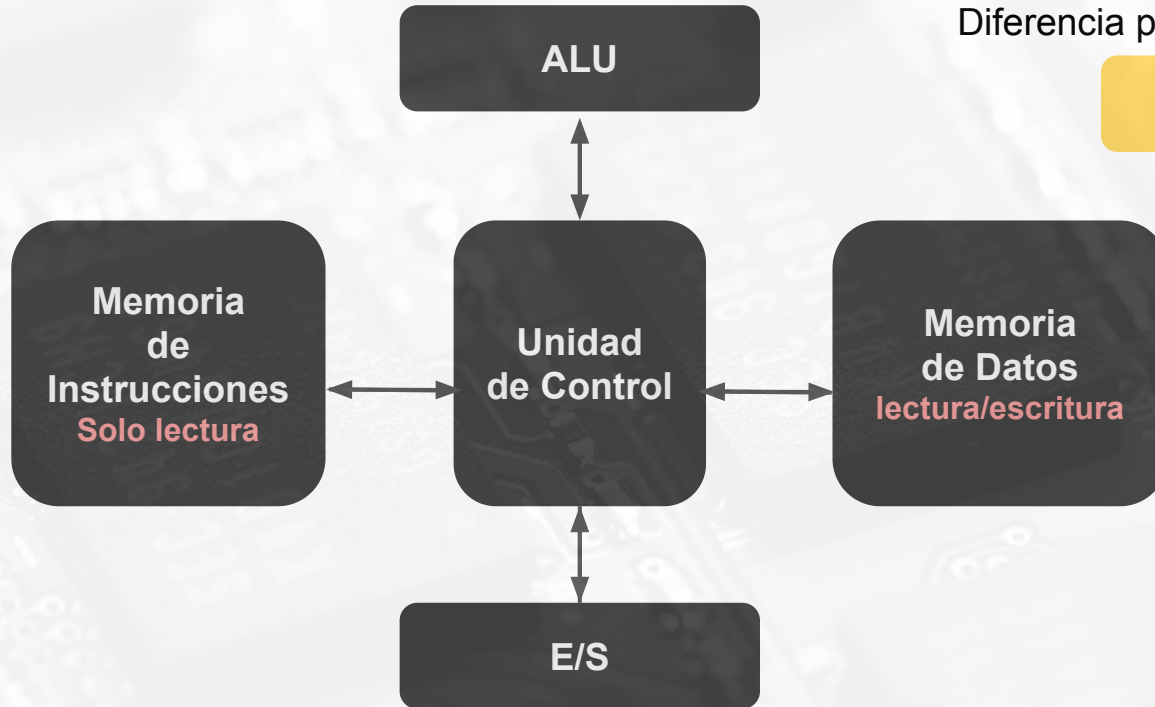


Control Bus (Bus de control): envía señales de control que indican a los diferentes componentes de la CPU cuándo deben realizar ciertas operaciones, como leer o escribir en la memoria, ejecutar una instrucción, o realizar operaciones aritméticas. El Control Bus es esencial para coordinar las operaciones internas de la CPU.

Address Bus (Bus de Direcciones): es un conjunto de líneas que se utiliza para transmitir direcciones de memoria. Permite que la CPU indique la ubicación de la memoria que desea leer o escribir. El tamaño del Address Bus determina la capacidad máxima de dirección de memoria que la CPU puede acceder, lo que a su vez afecta la cantidad máxima de RAM que una computadora puede utilizar. **SOLO UNA DIRECCIÓN A LA CPU!**

Data Bus (Bus de Datos): transfiere datos entre la CPU y la memoria u otros dispositivos. Este bus transporta la información que se está leyendo desde la memoria o la que se está escribiendo en ella, así como los datos que se utilizan en cálculos y operaciones dentro de la CPU. El tamaño del Data Bus determina la cantidad de datos que se pueden transferir en **paralelo** entre la CPU y la memoria.

Arquitectura Harvard



Diferencia principal con Arquitectura Von Neumann

Guarda instrucciones y datos en **diferentes** lugares

Harvard Modificada:

Permite que los contenidos de la memoria de instrucciones sean accedidas como si fuesen datos.

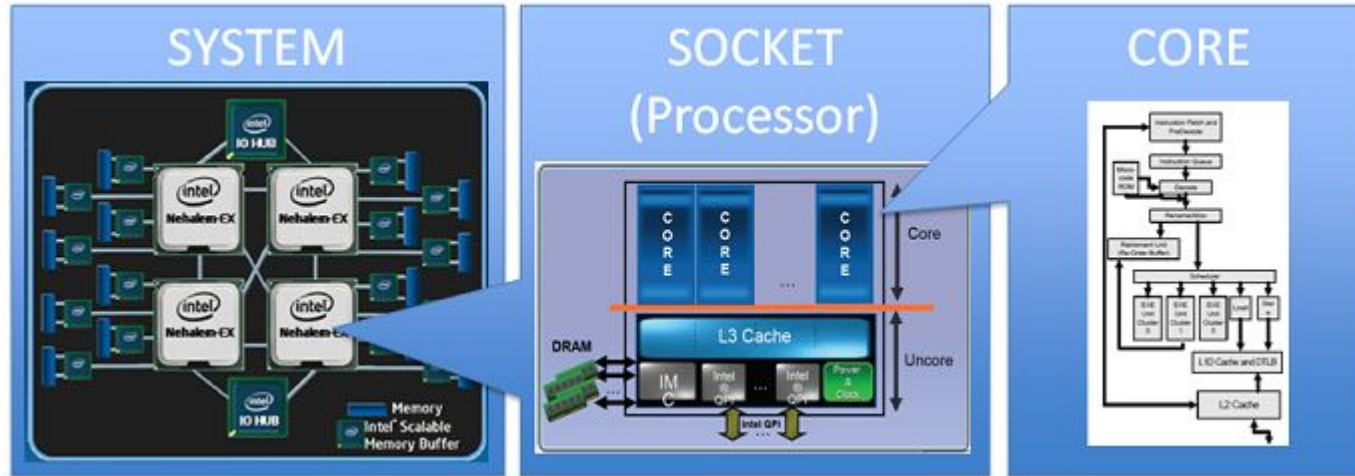
La memoria de instrucciones y datos ocupan diferente espacio de direcciones.

La memoria de instrucciones y datos tienen caminos hardware separados de la unidad central de proceso (CPU)

La memoria de instrucciones y datos pueden ser accedidas de diferente manera

Núcleo (Core) Vs CPU

Núcleo es una unidad de procesamiento independiente dentro de una CPU. Puede haber más de uno, lo que permite ejecutar más de una tarea al mismo tiempo en paralelo*



<https://www.intel.com/content/www/us/en/developer/articles/tool/performance-counter-monitor.html>

*Existen otras técnicas para ejecutar en paralelo

Multihilo (hyperthreading)

Un hilo es una unidad de ejecución dentro de un programa. Un programa puede contener varios hilos, cada uno de los cuales puede ejecutarse de manera independiente. El multihilo es una técnica de programación que permite que un programa ejecute varios hilos de manera simultánea.

Los núcleos y el multihilo están relacionados entre sí porque el multihilo permite que un programa ejecute varios hilos, y los núcleos proporcionan la capacidad de ejecutar estos hilos de manera simultánea.

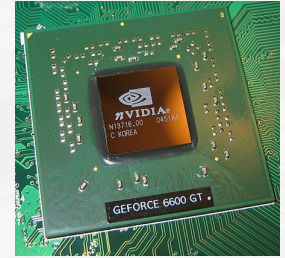
Paralelismo

[Race Condition](#)

GPU

Procesador diseñado específicamente para el procesamiento de gráficos. Las GPU son mucho más eficientes que las CPU, o unidades centrales de procesamiento, para realizar tareas que requieren cálculos de coma flotante en paralelo, como el renderizado de gráficos 3D.

Las GPU se componen de muchos núcleos de procesamiento que trabajan en paralelo. Esto les permite realizar tareas de renderizado de gráficos mucho más rápido que las CPU. Las GPU también tienen una memoria especial, llamada memoria de vídeo (VRAM), que se utiliza para almacenar los datos de los gráficos.



No todos los procesos son fácilmente *parallelizables*. Un proceso no es paralelizable si no se puede dividir en partes que puedan ejecutarse de forma independiente. En general, **los procesos que requieren acceso a los mismos datos o recursos NO son paralelizables.**

Por ejemplo, el audio en tiempo real, que es eminente serial/secuencial (aunque existen compañías que lo están llevando a cabo)

FPU

Al igual que CPU es un tipos diferentes de procesadores. Las CPU están diseñadas para realizar tareas generales, mientras que las FPU están diseñadas para realizar cálculos de coma flotante.

Las principales diferencias entre las FPU y las CPU son:

Eficiencia: Las FPU son mucho más eficientes que las CPU para realizar cálculos de coma flotante.

Estructura: Las FPU tienen una estructura diferente a las CPU. Las FPU están diseñadas específicamente para realizar cálculos de coma flotante.

Uso: Las FPU se utilizan principalmente para realizar cálculos de coma flotante. Las CPU se utilizan para una amplia gama de tareas, incluyendo el procesamiento de texto, el cálculo y la ejecución de programas.

Las FPU se utilizan en una amplia gama de aplicaciones:

Cálculo científico: Las FPU se utilizan para realizar cálculos científicos complejos, como los cálculos de ingeniería y los cálculos de física.

Procesamiento de imágenes: Las FPU se utilizan para procesar imágenes digitales, como el retoque fotográfico y la creación de efectos especiales.

Procesamiento de audio: Las FPU se utilizan para procesar audio digital, como la compresión de audio y la creación de efectos de sonido.

Juegos: Las FPU se utilizan para renderizar gráficos 3D en los videojuegos.

SoC (System on a Chip)

Es un **circuito integrado** que integra todos o la mayoría de los componentes de un sistema informático o electrónico en un solo chip (procesador, memoria, los controladores de entrada/salida (E/S), y otros componentes como el procesador gráfico GPU)

Los SoC son cada vez más comunes en una amplia gama de dispositivos, desde teléfonos inteligentes y tabletas hasta televisores y automóviles. Ofrecen una serie de ventajas sobre los sistemas tradicionales basados en múltiples chips, como un menor costo, un menor consumo de energía y un mayor rendimiento.

Normalmente la arquitectura tiene juego de instrucciones RISC basados en arquitectura Von Neumann. No obstante, existen también basados en arquitectura Harvard (que es común su uso en [sistemas embebidos](#)).

