

REPASO Unidad 01

0 → voltaje bajo

1 → voltaje alto

- **Resistor:** Controla el paso de la corriente (El portero, tú pasas, tú no)
- **Capacitor:** Almacena carga eléctrica (El almacen de la electricidad)
- **Diodo:** Deja pasar corriente en una sola dirección (El poli de tráfico, me pasan por aquí, por favor)
- **Transistor:** Funciona como un interruptor, encendiendo o apagando el paso de corriente. Es la base de la lógica y de los microprocesadores modernos. (La puerta básicamente)
- **Bobina:** Almacena energía en forma de campo magnético (El agricultor, to pal campo)
- **Tubo de vacío:** Usado en los primeros equipos electrónicos, antes de los transistores (Si el transistor es la puerta, este es la lona supongo)

Dato clave: El transistor permitió la miniaturización y construcción de procesadores modernos (gracias a él todos son mas chiquitos y chulos hoy en día)

- **Buffer:** Deja pasar la señal sin cambios (Va pasa, anda)
- **NOT:** Convierte 0 en 1 y 1 en 0 (chica a chico y chico a chica)
- **AND:** La salida es 1 solo si ambas entradas son 1 (el niño sale blanco si los padres son blancos)
- **OR:** La salida es 1 si al menos una entrada es 1 (el niño sale negro si un padre es negro)
- **NAND:** Es el inverso de la AND (el niño sale blanco si ambos padres son blancos, sino, sale negro siempre)(sale siempre 1 excepto si son dos 1)
- **NOR:** Es el inverso de la OR (el niño sale blanco si ambos padres son negros, sino, siempre negro) (sale siempre 0 excepto si son dos 0)
- **XOR:** La salida es 1 solo si las entradas son diferentes (sale blanco en un mix, el niño sale negro si son dos negros o dos blancos)
- **XNOR:** Es el inverso del XOR (Pues que el niño es negro si los padres son diferentes y si son iguales, blanco)(sale 1 si son iguales)

Como notita adicional, NAND y NOR son universales, world wide baby, como pitbull

Flip-Flop: Es diferente porque tiene memoria (es distinto y especial)

Puede guardar un valor de 0 o 1 hasta que recibe una señal de reloj (yo te lo guardo hasta que me digas)

Es la base de los registros y de la memoria dentro de un procesador (basicamente el cerebro)

Adder(Sumador): Está construido con compuertas lógicas (and, xor y or)(parece que diga anchoa)
Sirve para sumar números binarios, bit a bit (no tiene pérdida, no me jodas, el sumador suma.)

¡Carry y Overflow!

El **Carry** (o acarreo) es cuando la suma genera un 1 extra que se debe pasar a la siguiente posición, básicamente te han salido gemelos y uno lo abandonas.

El **overflow** ocurre cuando el resultado no cabe en el número de bits disponibles, diferencia? Ni idea, overflow es que no cabe y carry que sobra, uno se lo damos al vecino (carry) y el otro pues lo matamos creo (overflow)

Los sumadores son #TeamALU

Multiplier: Se construye a partir de sumadores y compuertas, más complicadito pero misma lógica, descomponer cálculo en 0 y 1

Se divide en **Multiplexer(MUX)** y **Demultiplexer(DEMUX)**, y no, no es el yosoyplex

El multi pillá una entrada de muchas y la envía a una sola salida, y el demulti pillá una entrada y la envía a muchas salidas, el único chiste que se me ocurre es machista y no voy a formar parte de un entorno misógino así que la dejo a vuestra imaginación.

State machine(Máquina de estados): combinamos flip-flops con lógica combinacional y obtenemos una steit maxin (state machine). En pocas palabras es un sistema que cambia de estado según sus entradas y un reloj, como tú con las notas del instagram, depende de tus md y la hora del día pones una canción triste o de fiesta.

Son la base de los algoritmos en hardware y siguen pasos secuenciales según condiciones

Tienes grinder? → Lo grindas Tienes papel? → Lo lías Tienes lapiz? → Lo prensas

Dynamic RAM y Static RAM

La **DRAM** es la ram de toda la vida, muchos datos en poco espacio y se pierde rápido si no chequeas, como un piso patera

La **SRAM** es más rápida pero más pequeñita (hay que refrescarla aún así pero cada tanto, como un cactus) y cara como sus muertos, un piso en madrid centro

Esta se usa en el procesador y la DRAM en la ram

La **flash memory** no pierde los datos, la construyen puertas NAND y es más lentilla, y dice algo de que los datos se borran en bloques completos y no bit a bit, pero vaya que son los USB

Capas del sistema computacional: El hardware, firmware y software

El **hardware** es lo físico, no voy a desarrollar más, el **firmware** son las instrucciones básicas que controlan directamente al hardware, es decir, el hardware es un objeto y el firmware el que le dice “muévete colega”, y el **software** pues los programitas

Ah y **compilador/intérprete** que es el que traduce los lenguajes de progra a binario para que los entienda la maquina

CISC y RISC

CISC es **Complex Instruction Set Computing**, básicamente muchas instrucciones, facilito y lento

RISC es el juego de mesa, nah es **Reduced Instruction Set Computing**, instrucciones más simples y rápidas, le cuesta así que necesita más instrucciones y optimiza la velocidad

CISC → Fácil pero muchas instrucciones y es lento

RISC → Simple y rápido, le cuesta (se esfuerza) pero optimiza velocidad

BIOS Basic Input Output System

Firmware básico que ejecuta al encender el pc

Inicia el hardware, hace pruebas para que todo esté bien y carga el SO en la memoria

Básicamente la mamá del ordenador

El **bootloader** es un programa pequeñito que va después del BIOS que básicamente copia el so desde el disco a la ram, le da control al so y usa hexadecimal para representar direcciones

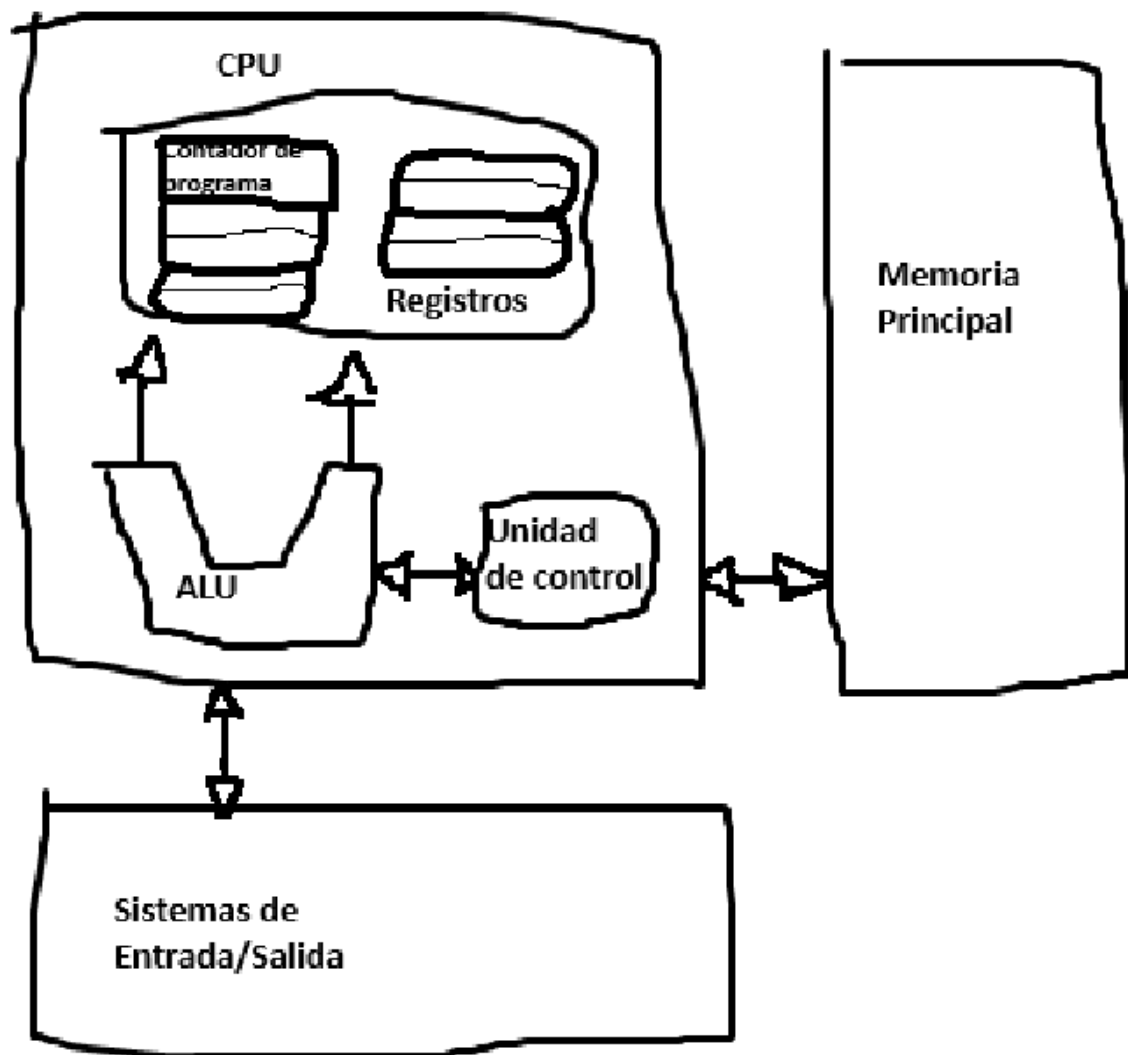
El duendecito que dice tu madre que limpia los platos? Pues es este hijo de puta

Ah y se carga en el 0x7C00 que me la pela pero igual lo pregunta Dani

Arquitectura Von Neumann

Básicamente es un modelo (no con tetas) donde **datos e instrucciones comparten** la misma **memoria**, y la **CPU** funciona de **forma secuencial** siguiendo el **ciclo** de **buscar, decodificar y ejecutar** (cojo la manzana, muerdo la manzana, tiro la manzana). Usa la ALU para operar y los buses como autopistas para mover información entre la cpu, memoria y dispositivos (ALU es el semaforo y los buses las carreteras). Es una arquitectura simple (los cojones) y lineal donde todo pasa por el mismo camino, por lo que puede crear cuellos de botella (no puede, lo hace, ya te lo digo yo)

Dibujito profesional:



Unidad de control: Coordina y dirige el funcionamiento interno de la cpu, es el mandón de la cpu. Lee instrucciones desde la memoria, las interpreta y genera señales de control necesarias para la ALU, registros y buses.

Básicamente la memoria le pasa instrucciones, la unidad de control interpreta lo que le sale de los huecos y se lo manda a sus amigos alu, registros y buses para las lleven a cabo.

ALU – Unidad Aritmético-Lógica: Es la que ejecuta las operaciones de mates y lógicas, usa los registros internos para recibir operandos y producir resultados. Tras cada operación actualiza las banderas que indican condiciones como cero, acarreo u overflow.

En pocas palabras es el friki de las mates, los registros le dicen qué operandos usar y cada vez que opera actualiza la bandera para comentar la situación y ver qué hacemos ahora (como la bandera de la playa)

Memoria Principal: Almacena tanto los programas (instrucciones) como los datos que esos programas utilizan, cada dato o instrucción tiene su posición numerada por una dirección, que la CPU puede leer o escribir. Es volátil, que se pira al apagar.

Básicamente es el que pone los números en la cola de la carnicería y te guarda todas las notitas en la reunión en caso de que quieras leer alguna o cambiarla o lo que te apetezca. Es un secretario pero que cuando sales de la oficina explota y desaparece.

Dispositivos de Entrada/Salida I/O: Son los que permiten que el computador reciba información de fuera y envíe resultados al usuario u otros sistemas. La CPU se comunica con ellos mediante instrucciones específicas o a través de controladores dedicados.

Muy fácil, es el micrófono y altavoz de la cpu, le permite hablarte a ti u otros y escucharte a ti u otros, se comunica con ellos o a muy malas con drivers dedicados. (Es mediante el hardware eh...)

Buses del sistema: Son los canales que conectan todos los elementos del sistema y permiten el transporte de información. El bus de datos mueve valores y resultados entre cpu, memoria y dispositivos. El de direcciones indica donde se realizará la lectura o escritura en memoria, y el de control lleva señales que coordina operaciones, como habilitar lectura/escritura en memoria o manejar instrucciones.

Repito, son básicamente las carreteras. Hay tres, datos, direcciones y control, la de datos lleva valores y resultados a cpu, memoria y dispositivos, la de direcciones no hay que ser un genio para saber qué llevará pero básicamente dice donde se escribe o lee algo en la memoria, y la de control lleva señales para coordinar operaciones para permitir lectura, escritura o el manejo de instrucciones.

Datos: Lleva datos **Direcciones:** Te dice pa donde ir **Control:** Controla (wow)