ARQUITECTURA DE COMPUTADORAS

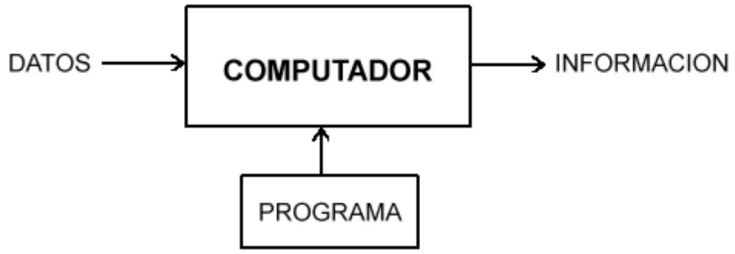
- Estructura, organización y funciones de una computadora. Modelo de Turing y Von Neumann.
- Elementos de un Procesador: Unidad de aritmética y lógica, Unidad de control y registros.Canales de Direcciones, Control y Datos.

Computadora. Es una maquina digital, basada en circuitos electrónicos digitales, para el procesamiento de datos. Tiene las siguientes capacidades:

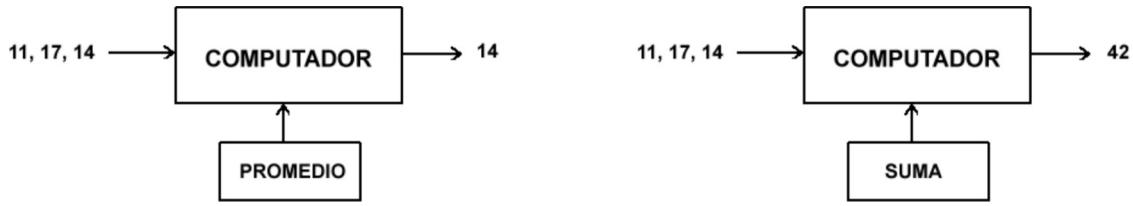
- Entrada. Permite el ingreso de datos en un formato digital que puede ser simple o, a través de una interface, elaborada.
- Procesamiento. Permite procesar los datos ingresados y a través de un proceso de análisis y transformación, conocido como **programa**, permite obtener datos resultantes a los que llamaremos **información**.
- Salida. Permite representar o transferir el contenido de información obtenido hacia un dispositivo de salida.

La computadora moderna es una **máquina de propósito general**, es decir que el proceso a aplicar a los datos puede ser cambiado según la naturaleza del problema a resolver.

Esto se consigue a través de la implementación del concepto de **programa almacenado**. Es decir no solo se ingresan los datos a procesar sino también las instrucciones del proceso a aplicar, el cual es conocido como **PROGRAMA**.



Por ejemplo en el siguiente grafico vemos que los datos y el computador son los mismos, pero el programa aplicado es diferente, y por tanto la información resultante es también diferente.

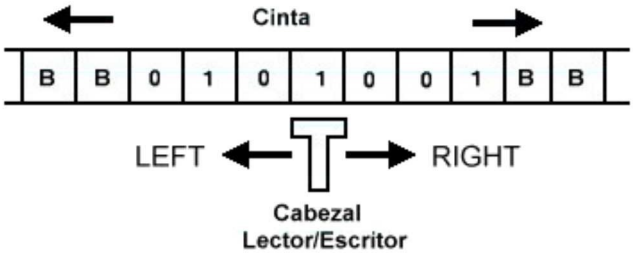


Modelo de Turing: Maquina de Turing

Planteado por Alan Turing en 1936. Es un modelo matemático de un dispositivo que implementa un autómata finito.

Este dispositivo es importante porque plantea los conceptos de Almacenamiento y Unidad de Control, los cuales son componentes fundamentales de un computador.

La Maquina de Turing consta de una cinta infinita conteniendo elementos de datos. Ademas cuenta con un Cabezal Lector/Escritor, que le permite posicionarse en un elemento especifico, y sobre dicho elemento puede Leer o Escribir. Igualmente el Cabezal tiene la propiedad de poder moverse hacia la Izuierda (Left>) o hacia le Derecha (Right).



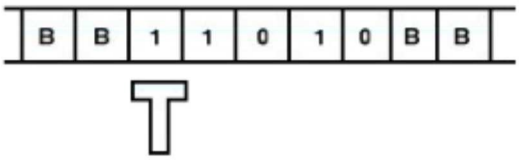
En esta maquina empezamos en un estado inicial q_0 y a partir de ahí realizamos transiciones hasta llegar a un estado final q_F.

Cada transición tiene el formato:

(Estado Actual, Simbolo Actual) --> (Estado Siguiente, Simbolo a grabar, Desplazamiento)

Simbolo Actual se refiere al valor leído por el cabezal en su posición actual. Si el valor es B entonces significa que es un valor en blanco o vacio.
Simbolo a grabar, se refiere al símbolo que graba el cabezal en la posición actual, antes de realizar el desplazamiento.
Desplazamiento, es el movimiento del Cabezal, que puede ser a Izquierda (L), a Derecha (R), o sin movimiento (-).

Por ejemplo dada una cinta con los siguientes elementos y el cabezal en la posición especificada.

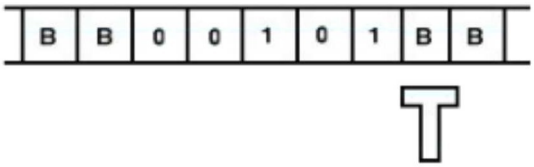


Las siguientes transiciones

δ (q0, 0) = (q0, 1, D)
δ (q0, 1) = (q0, 0, D)
δ (q0, B) = (qF, B, -)

Se empieza en el estado inicial q0 y realizamos la transición que corresponda, de acuerdo al valor leído. Pasamos luego al siguiente estado (que puede ser el mismo estado), realizamos una grabación en la posición actual y después realizamos un desplazamiento.

El resultado de la ejecución de dicho conjunto de transiciones (que podemos ver como un "programa") dado ese valor inicial de la cinta será:



Apreciamos que el resultado es la negación binaria del numero original.

Ejem. Dado un numero binario obtener el siguiente numero binario, es decir realizar la operación incremento.

Analizando vemos que si el digito menos significativo es 0 entonces solo habrá que cambiarlo a 1. Pero si dicho digito es 1 entonces habrá que cambiarlo a 0 y los dígitos siguientes, de derecha a izquierda, que sean 1 seran cambiados a 0 y asi hasta encontrar el primer 0, que se cambia por 1 y termina. En caso de que no haya ningún 0 en ese recorrido entonces hay que añadir un 1 a la izquierda del digito mas significativo.

Por ejem. el incremento de
10110 --- > 10111
10111 --- > 11000
11111 --- > 100000 (se ha añadido un 1 bit)

Las reglas de transición serian:

δ (q0, 0) = (q0, 0, D) // Se avanza al extremo derecho del numero
δ (q0, 1) = (q0, 1, D)
δ (q0, B) = (q1, B, I) // Al llegar al blanco retrocedemos una posición
δ (q1, 0) = (qF, 1, -) // Si es 0 cambiamos por 1 y terminamos.
δ (q1, 1) = (q1, 0, I) // Si es 1 cambiamos por 0 y desplazamos a Izq.
δ (q1, B) = (qF, 1, -)

Para los números binarios del ejemplo y considerando que empezamos en el estado q0, y con el cabezal en el digito mas significativo, las secuencias de aplicación de las reglas de transición serian:

10110 → 2da regla, 1ra, 2da, 2da, 1ra, 3ra, 4ta
10111 → 2da regla, 1ra, 2da, 2da, 2da, 3ra, 5ta, 5ta, 5ta, 4ta
11111 → 2da regla, 2da, 2da, 2da, 2da, 3ra, 5ta, 5ta, 5ta, 5ta, 5ta, 6ta

Si bien la maquina de Turing proporciono la idea de unidad de control, almacenamiento y rutina de procesamiento, su implementación practica en una computadora no es eficiente debido a su acceso secuencial.

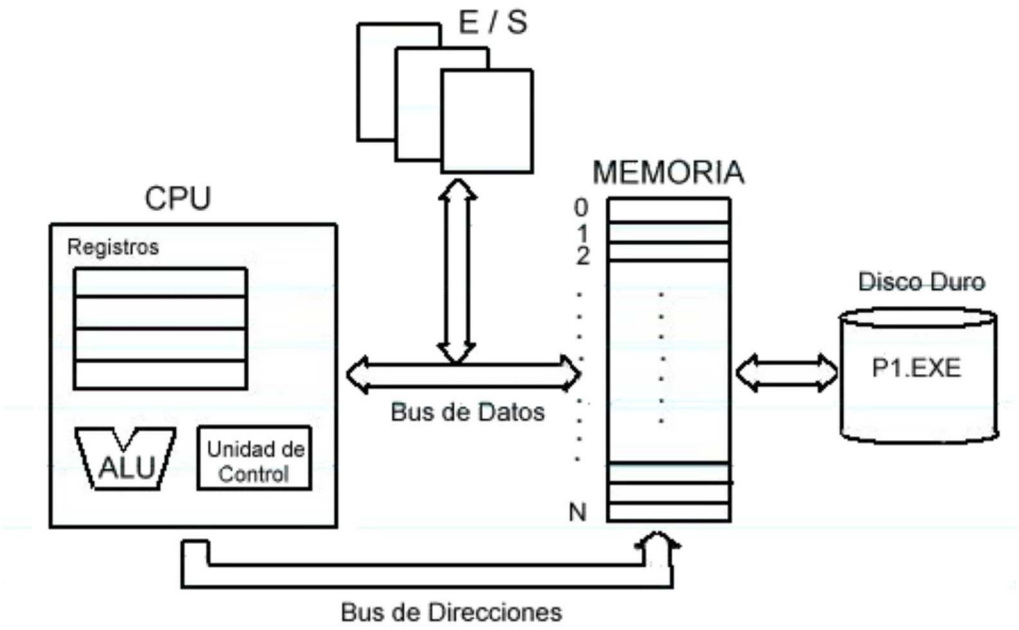
Arquitectura de Von Neumann
Es un modelo conceptual, es decir una arquitectura, que muestra los componentes de una computadora y su funcionamiento correspondiente. Fue planteado en 1945 por John Von Neumann.

La Arquitectura de Von Neumann consta de 3 componentes fundamentales:

1. Unidad de Memoria. Para el almacenamiento tanto de datos como de instrucciones.

2. Unidad Central de Procesamiento (CPU). Encargada de la decodificación, interpretación y ejecución de las instrucciones.
   - Unidad de Control. Sincroniza y controla las operaciones de transferencia y ejecución.
   - Unidad Aritmético Lógica (ALU). Se encarga de realizar las operaciones lógicas, aritméticas y binarias.
   - Registros. Dispositivos de almacenamiento al interior de la CPU y que están en conexión directa con la Unidad de Control y con la ALU,

3. Unidad de Entrada Salida. Proveen los medios para la comunicación de datos desde y hacia el exterior del computador.

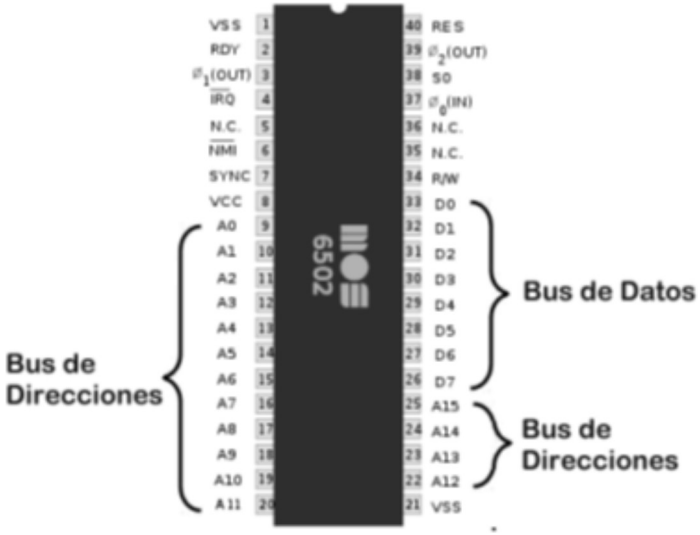


Actualmente la CPU es implementado a través de un **microprocesador**, es decir un circuito integrado, igualmente la memoria se implementa a través de modulos, conocidos como bancos, de memoria.

Estos modulos se comunican por líneas circuitales, agrupadas en conjuntos conocidos como **buses**. Los buses principales son el **Bus de Datos**, que permite transferir bytes entre

la CPU y la Memoria, y el **Bus de Direcciones**, que permite seleccionar la dirección especifica de la memoria que debe accesarse.

Ejemplo: En un microprocesador de 8 bits clásico como es el MOS 6502 podemos apreciar las líneas del Bus de Datos (D0 – D7) y las líneas del Bus de Direcciones (A0 – A15). Las otras líneas del procesador tienen funciones de control, por lo que algunos los conocen como Bus de Control, pero al ser líneas con funcionalidades independientes ese término no es tan consistente como en el caso de los otros buses.



Este procesador es de 8 bits, es decir su Bus de Datos es de 8 lineas (D0-D7), lo cual se corresponde con el tamaño de sus registros internos. En cada lectura o escritura desde o hacia la memoria puede transferir 1 byte.

Su bus de direcciones es de 16 bits (A0-A15) por lo que tiene una capacidad de direccionamiento de 64k.

Registros

Los registros permiten contener a las instrucciones a ejecutar, junto con sus operandos. Estos registros tienen acceso directo a la Unidad de Control y al ALU, lo cual permite ejecutar las instrucciones.

Existen 2 registros especiales que son importantes en la ejecución de instrucciones:

Registro IP o Puntero de Instrucciones (Instruction Pointer), almacena la dirección de la próxima instrucción a ejecutarse.

Registro Flag o Registro de Banderas, contiene un conjunto de bits indicadores de condicion, los cuales reflejan las características de la ultima operacion realizada. Esto permite evaluar condicionales y por tanto es la base para las instrucciones de control.

Entre los principales indicadores de condición tenemos:

- Zero: Se pone a 1 cuando el resultado es 0, en caso contrario se pone a 0
- Signo: Se pone a 1 cuando el resultado es positivo, 0 en caso contrario
- Carry: Se pone a 1 cuando hay un acarreo en la suma, 0 en caso contrario
- Overflow: Se pone a 1 cuando hay un desborde (por ejem. en una multiplicación)
- Paridad: En paridad par se pone a 1 para completar el numero de bits 1 a una cantidad par. Ejem. Si R=1011 entonces P=1, dando un total par de 4 bits.

Ejemplo: En un procesador 8086, de 16 bits, se tienen los siguientes registros: AX, BX, CX, DX. Todos ellos de 16 bits.
Un ejemplo de programa en código maquina es:

```
mov ax, 2       // El registro AX recibe el valor 2
mov bx, 3       // El registro BX recibe el valor 3
add ax, bx      // Se suma AX con BX y el resultado se guarda en AX
```

Puesto en el computador tenemos:

Vemos que la instrucción ADD AX, BX corresponde a un código maquina que es 01D8, es decir 2 bytes (representado en hexadecimal).

Entonces si en la memoria coloco 2 bytes con contenido:

01D8 (hexadecimal) = 00000001 11011000 (en binario)

estamos colocando entonces una instrucción ADD AX, BX.

Tambien observamos que el programa completo, compuesto por las 3 instrucciones, ocupa un total de 8 bytes: B8 02 00 BB 03 00 01 D8

Ejemplo: El siguiente programa obtiene la suma de los impares menores que 10

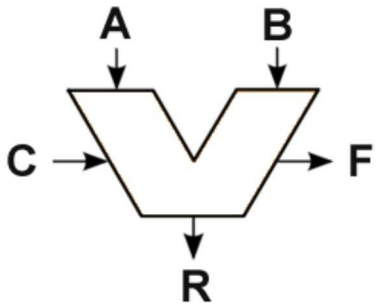
```
      mov ax, 0       // aquí guardamos la suma
      mov bx, 1       // bx contiene a los impares, empezando por 1
E1:   add  ax, bx     // ax = ax + bx
      add  bx, 2      // obtenemos el siguiente impar
      cmp bx, 0A      // comparamos el impar con 10
      jl    E1        // Si es menor saltamos a la dirección asociada a E1
      nop             // No Operación
```

```
0859:0100 B80000          MOV     AX,0000
0859:0103 BB0100          MOV     BX,0001
0859:0106 01D8            ADD     AX,BX
0859:0108 83C302          ADD     BX,+02
0859:010B 83FB0A          CMP     BX,+0A
0859:010E 7CF6            JL      0106
0859:0110 90              NOP
```

Vemos que la etiqueta E1 ha tomado el valor de dirección que le corresponde, 0106.

Unidad Aritmetico Logica (ALU). Es un circuito digital que realiza operaciones aritméticas, lógicas o binarias entre uno o dos operandos.

A y B son los operandos
R es el resultado

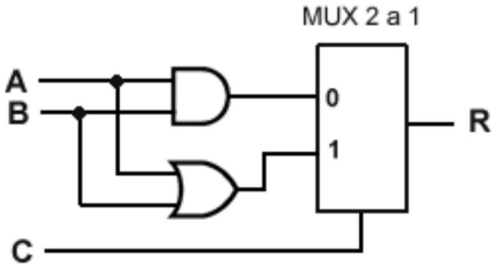


C es el control que determina que operación se va a realizar entre A y B
F es un indicador de condición resultante, proporciona información general sobre el resultado.

Ejemplo. Diseñar una ALU de 1 bit que implemente las operación AND y OR

C=0	A and B
C=1	A or B

El circuito lógico es:



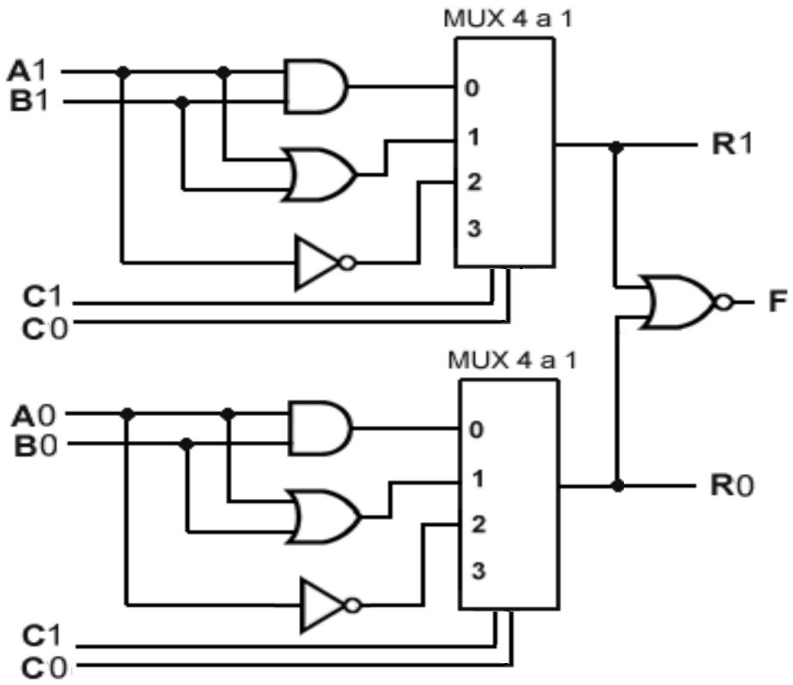
La entrada de control C determina que operación que entra al multiplexor se conecta con la salida R. Si C=0 entonces R = A and B, pero si C=1 entonces R = A or B

Ejemplo. Diseñar una ALU de 2 bits y con 3 operaciones (AND, OR, NOT A), además debe haber un bit de condición Zero que arroja 1 cuando el resultado es 0.

Como son 3 operaciones entonces necesitamos 2 bits de control, los cuales nos dan 4 combinaciones, de los cuales solo usaremos 3.

C=00	A and B
C=01	A or B
C=10	not A

El circuito lógico es:



Ejercicios:

1. Implementar una Maquina de Turing que obtenga el complemento A2 de un numero binario. Algoritmo: Se recorre de derecha a izquierda dejando los dígitos sin cambiar hasta encontrar el primer 1, el cual se no se cambia, pero los siguientes digitos se cambian 0 por 1 y 1 por 0.

2. Implementar una Maquina de Turing que intercambia el valor del digito menos significativo con el valor del digito mas significativo.

3. Cuantas líneas de dirección necesita un procesador para direccionar una memoria de 16k?

4. Averigue cual fue el primer procesador Intel en contar con arquitectura de 32bits. Igualmente averigue cual fue el primero con arquitectura 64bits.

5. Para que se utiliza el complemento A2 en el computador?

6. Diseñar una ALU de 2 bits que implemente las operaciones:

C=00	A and B
C=01	A or B
C=10	A + B

   Asimismo debe tener un bit de condición C = Carry que se pone a 1 cuando hay acarreo en la suma. Usar Half Adder para la suma.

7. Para una ALU de 2 bits implementar un bit de Paridad Impar.

8. Revisar la información del circuito integrado 74LS181 el cual es una ALU de 4 bits que tiene implementado 16 operaciones aritméticas y 16 operaciones lógicas.

Videos Recomendados:

1. Funcionamiento maquina von Neumann
   https://www.youtube.com/watch?v=eDKzR06bpd8

2. Arquitectura de Von Neuman
   https://www.youtube.com/watch?v=G_S7hQJdp3g

3. Funcionamiento del Modelo de Von Neumann
   https://www.youtube.com/watch?v=apM1_35fdRA

4. The von Neumann Model
   https://www.youtube.com/watch?v=H0xGKKpKaRE

5. Circuito de la ALU de 4 bits 74LS181
   https://www.youtube.com/watch?v=kper5h1BDDM