




# Unidad III. ARQUITECTURA DE COMPUTADORAS

Organización y diseño de computadoras




# 1) Arquitectura Von Neumann vs Arquitectura Harvard

## VON NEUMANN

Tradicionalmente los sistemas con microprocesadores se basan en esta arquitectura, en la cual la unidad central de proceso (CPU), está conectada a una memoria principal única (casi siempre sólo RAM) donde se guardan las instrucciones del programa y los datos. A dicha memoria se accede a través de un sistema de buses único (control, direcciones y datos).

En un sistema con **arquitectura Von Neumann** el tamaño de la unidad de datos o instrucciones está fijado por el ancho del bus que comunica la memoria con la CPU. Así un microprocesador de 8 bits con un bus de 8 bits, tendrá que manejar datos e instrucciones de una o más unidades de 8 bits (bytes) de longitud. Si tiene que acceder a una instrucción o dato de más de un byte de longitud, tendrá que realizar más de un acceso a la memoria.

El tener un único bus hace que el microprocesador sea más lento en su respuesta, ya que no puede buscar en memoria una nueva instrucción mientras no finalicen las transferencias de datos de la instrucción anterior.



# 1) Arquitectura Von Neumann vs Arquitectura Harvard

Las principales limitaciones que nos encontramos con la **arquitectura Von Neumann** son:

- La limitación de la longitud de las instrucciones por el bus de datos, que hace que el microprocesador tenga que realizar varios accesos a memoria para buscar instrucciones complejas.
- La limitación de la velocidad de operación a causa del bus único para datos e instrucciones que no deja acceder simultáneamente a unos y otras, lo cual impide superponer ambos tiempos de acceso

La **arquitectura Von Neumann** realiza o emula los siguientes pasos secuencialmente:

**1)** Obtiene la siguiente instrucción desde la memoria en la dirección indicada por el contador de programa y la guarda en el registro de instrucción.

**2)** Aumenta el contador de programa en la longitud de la instrucción para apuntar a la siguiente.

**3)** Decodifica la instrucción mediante la unidad de control. Ésta se encarga de coordinar el resto de componentes del ordenador para realizar una función determinada.

**4)** Se ejecuta la instrucción. Ésta puede cambiar el valor del contador del programa, permitiendo así operaciones repetitivas.

**5)** Regresa al paso N° 1.

# 1) Arquitectura Von Neumann vs Arquitectura Harvard

Aquí expone el meco de Rocha

## HARVARD

Este modelo, que utilizan los Microcontroladores PIC, tiene la unidad central de proceso (CPU) conectada a dos memorias (una con las instrucciones y otra con los datos) por medio de dos buses diferentes.

Una de las memorias contiene solamente las instrucciones del programa (Memoria de Programa), y la otra sólo almacena datos (Memoria de Datos).

Para un procesador **RISC (Reduced Instrucción Set Computer)**, el set de instrucciones y el bus de memoria de programa pueden diseñarse de tal manera que todas las instrucciones tengan una sola posición de memoria de programa de longitud.





# 1) Arquitectura Von Neumann vs Arquitectura Harvard

## Ventajas de esta arquitectura:

- \* El tamaño de las instrucciones no está relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa.

- \* El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad en cada operación.

desventaja es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que pueda ser necesario incluir en los programas, ya que estas tablas se encontraran físicamente en la memoria de programa.

## 2) Acumulador

Cuando se agrega un número A con un número B, la operación producirá una suma S y un carry C.

- **Medio Sumador:** Si hay dos números, A y B, la operación de sumar dos números se puede mostrar como:

$$A + B = S \ C$$

$$0 + 0 = \text{Suma } 0 \text{ Llevar } 0$$

$$0 + 1 = \text{Suma } 1 \text{ Llevar } 0$$

$$1 + 0 = \text{Suma } 1 \text{ Llevar } 0$$

$$1 + 1 = \text{Suma } 0 \text{ Llevar } 1$$

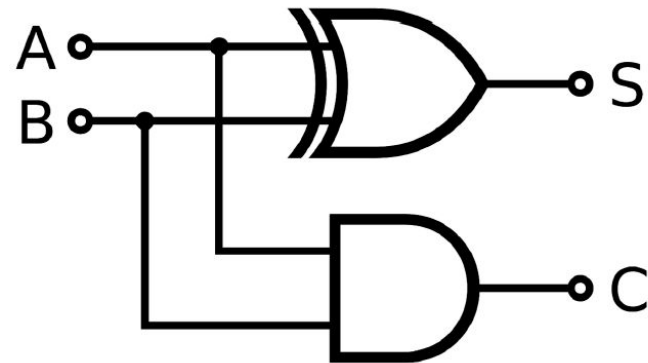
- En tabla de verdad se verá de la siguiente forma.

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- El medio sumador puede construirse a partir de las puerta AND y la puerta XOR.

$$S = A \oplus B$$

$$C = A \cdot B$$



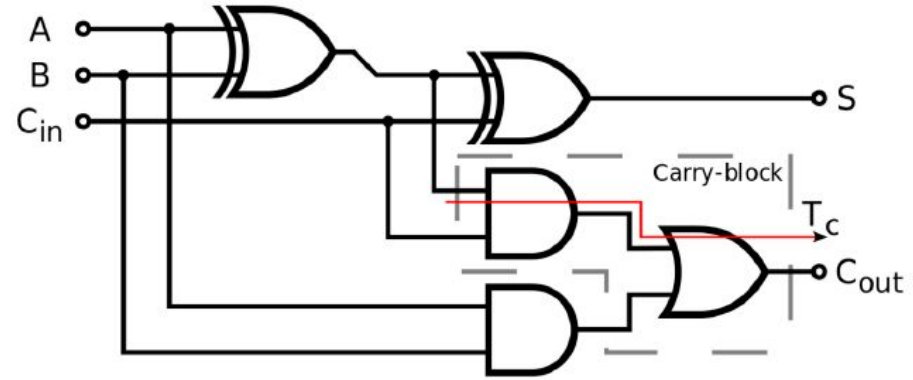
## 2) Acumulador

- **Sumador Completo:** Un sumador completo es un circuito lógico que realiza una operación de suma en tres números binarios de un bit.
- El sumador completo produce una suma de las dos entradas y el valor de acarreo.

Input			Output	
A	B	C <sub>i</sub>	C <sub>o</sub>	S
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1

$$S = (A \oplus B) \oplus C_{in}$$

$$C_{out} = (A \cdot B) + (C_{in} \cdot (A \oplus B))$$

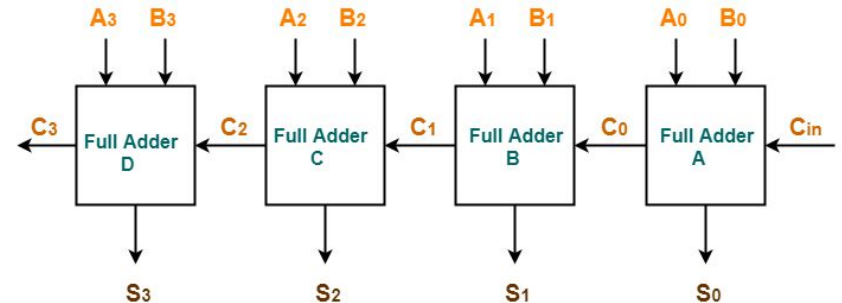


- Se puede construir un sumador completo a partir de dos medios sumadores, conectando la suma de esa una entrada al segundo sumador, conectando C<sub>i</sub> a la otra entrada y OR la dos salidas de transporte.

## 2) Acumulador.

- **Ripple Carry Adder:** Es posible crear un circuito lógico utilizando múltiples sumadores completos para agregar números de N bits. Cada sumador completo ingresa un  $C_{in}$  que es el  $C_{out}$  del sumador anterior. Este tipo de sumadores es un sumador de arrastre, ya que cada bit de arrastre “ondula” al siguiente sumador completo.

- Se debe de tomar en cuenta que **SOLO** el primer sumador completo puede reemplazarse por un medio sumador.
- El diseño del sumador de transporte de ondulación es simple, lo que permite un tiempo de diseo rápido; sin embargo, el sumador de arrastre de ondulaciones es relativamente lento, ya que cada sumador completo debe esperar a que se calcule el bit de arrastre del sumador completo anterior.



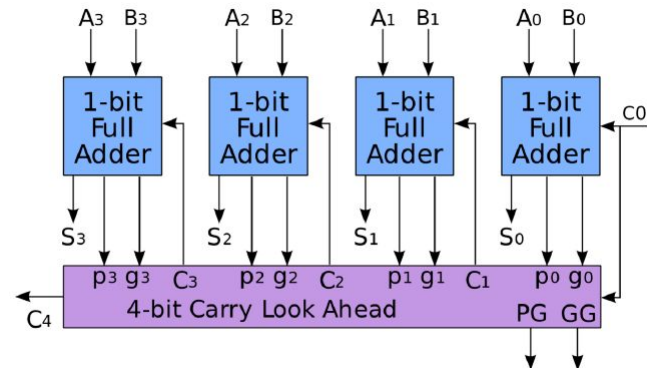
4-bit Ripple Carry Adder



## 2) Acumulador.

- **Llevar sumadores de anticipación:**  
Para reducir el tiempo de cálculo, los ingenieros idearon formas más rápidas de agregar dos números binarios mediante el uso de sumadores de anticipación de acarreo.
- Funcionan creando dos señales (P y G) para cada posición de bit, en función de si un acarreo se propaga desde una posición de bit menos significativa (al menos y la entrada es un '1'),

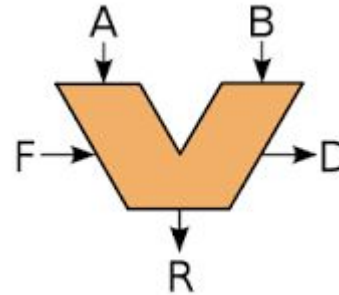
- Se genera un acarreo en esa posición de bit (ambas entradas son '1'), o si se elimina un carry en esa posición de bit (ambas entradas son '0').
- En la mayoría de los casos, P es simplemente la salida de suma de un medio sumador y G es la salida de arrastre del mismo sumador.
- Después de que se generan P y G, se crean los acarreos para cada posición de bit.



### 3) Unidad Aritmético Lógica

¿Qué es?

Se le conoce como ALU (Arithmetic Logic Unit), que en su interior tiene circuitos digitales que realizan las operaciones aritméticas y lógicas, entre dos operandos.





### **3) Unidad Aritmético Lógica**

## Operaciones lógicas

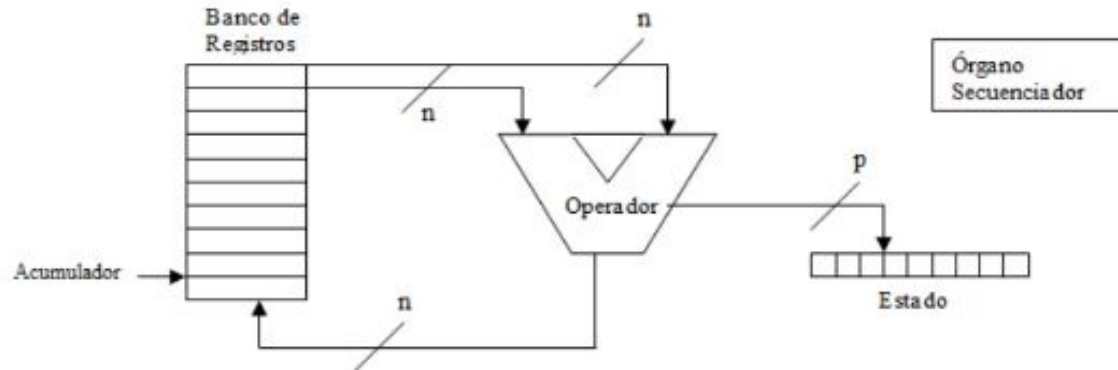
Las operaciones que se encuentran generalmente en las computadoras son:

- negación (NOT)
- suma (OR)
- producto (AND)
- suma exclusiva (XOR)

### 3) Unidad Aritmético Lógica

## Estructura

La ALU generalmente se compone de uno o varios operandos, de un conjunto de registros, de unos biestables de estado y, en algunos casos, de un órgano secuenciador.





## 4) Program Counter

También conocido como registro de instrucciones PC, o puntero de dirección

Un registro en el que lleva mucha memoria pero poca capacidad en el que permite guardar y acceder a los valores.

Algunos de tipos de registros que existen:

1. Registros de datos: para guardar números.
2. Registros de memoria: para direcciones
3. Registros de propósito general: pueden guardar datos y direcciones y es fundamental para la arquitectura de von Neumann
4. Registro de índice: para direcciones indexados y sumas y restas

El contador de programa es un registro que se encuentra en la unidad de procesamiento central (CPU)

# Program Counter

## Funciones:

1. Proporcionar alojamiento para la instrucción que va ser ejecutado.
2. Recibir instrucciones en secuencia.
3. Proporcionar una ejecución lógica de la tarea que se va a ejecutar.

Existen tipos de instrucciones como:

procesamiento de datos: Operaciones aritméticas y lógicas

Almacenamiento de datos: Transferencia dentro del sistema

Instrucciones E/S: Transferencia de datos entre la computadora y los dispositivos externos.



# Program counter



Fases para la ejecución de las instrucciones:

1. **Leer la instrucción:** El registro de contador nos dice dónde está la instrucción que debemos leer, verificar sus partes para saber que debemos realizar.
2. **Lectura de los operando de la fuente:** Las operaciones se realizan mediante su direccionamiento, si es directo no hay que hacer ninguna operación pero si es indirecto tenemos que hacer cálculos y accesos a la memoria.
3. **Ejecución de la instrucción y su almacenamiento:** Para cada código son diferentes operaciones, pero una vez ejecutado guardamos el resultado obtenido y guardarlo.
4. **Comprobación:** Se verifica si se ha activado alguna línea de petición de interrupción. y para ello primero lo almacena el contador de programa en donde empieza la rutina después ejecutamos el servicio de interrupción y recuperamos los datos.

## 5) Unidad de Control

Es uno de los tres bloques funcionales principales en los que se divide una unidad central de procesamiento(CPU).

Componentes:

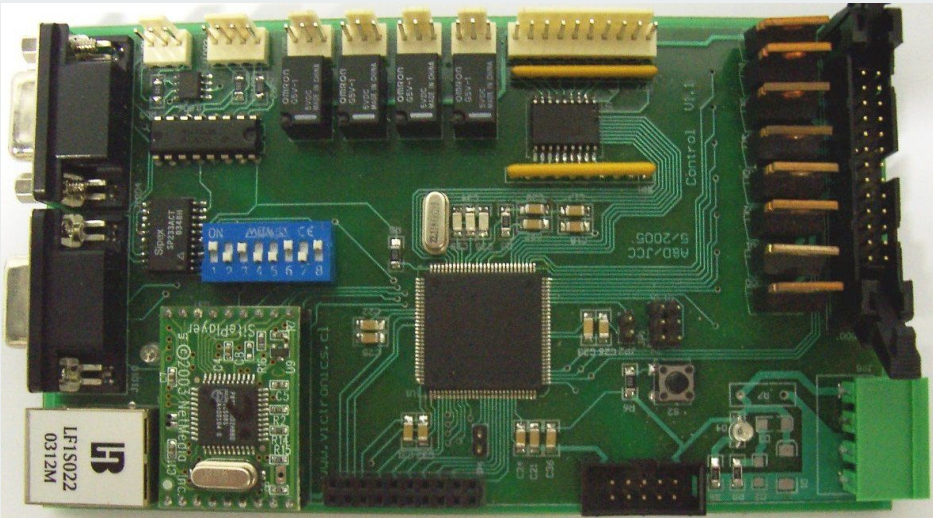
**Registro de instrucción:** Almacena la instrucción que se ejecuta.

**Registro de contador de programas:** Dirección de memoria de la sig instrucción a ejecutar.

**Controlador y decodificador:** Interpreta la instrucción para el sig proceso. Extraer el código de operación de la instrucción en curso.

**Secuenciador:** Genera microórdenes necesarias para ejecutar la instrucción.

**Reloj:** Proporciona una sucesión de impulsos eléctricos a intervalos constantes.








## Función

Su función es buscar las instrucciones en la memoria principal, decodificarlas (interpretación) y ejecutarlas, empleando para ello la unidad de proceso.

Existen dos tipos de unidades de control: las **cableadas**, usadas generalmente en máquinas sencillas, y las **microprogramadas**, propias de máquinas más complejas.

**Cableadas:** Es construida de puertas lógicas, circuitos biestables, circuitos codificadores, circuitos decodificadores, contadores digitales y otros circuitos digitales. Su control está basado en una arquitectura fija, es decir, que requiere cambios en el cableado si el conjunto de instrucciones es modificado o cambiado.



**Microprogramada:** Fue introducida la idea de microprogramación como un nivel intermediario para ejecutar instrucciones de programa de computadora.

Fueron organizados como una secuencia de microinstrucciones y almacenados en una memoria del control especial.


La ventaja principal de la unidad de control microprogramada es la simplicidad de su estructura

## Funciones:

En un procesador regular que ejecuta nativamente las instrucciones x86, la unidad de control realiza las tareas de leer (*fetch*), decodificar, manejo de la ejecución y almacenamiento de los resultados.

En uno de estos procesadores la unidad de control está dividida en otras unidades debido a la complejidad del trabajo que debe realizar

Almacena los datos más utilizados de modo que se buscan primero en la computadora y luego en la RAM.



## **6) Operaciones aritméticas (Suma, Resta, Multiplicación y División)**