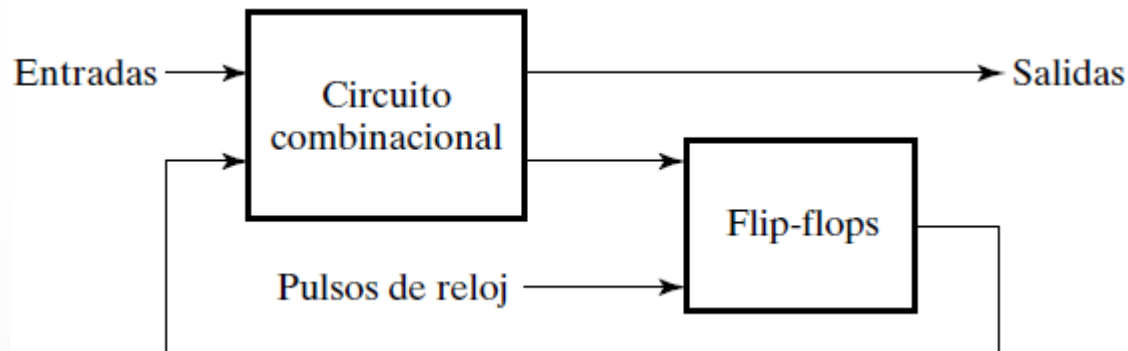


Unidad 4

Arquitectura general y organización funcional de computadoras. Descripción de los distintos bloques (Memoria, ALU, Unidad de control y Unidad de E/S). Proceso de búsqueda y ejecución de las instrucciones. Interrupciones. Estructuras de interconexión. Buses.

Máquinas de estados finitos

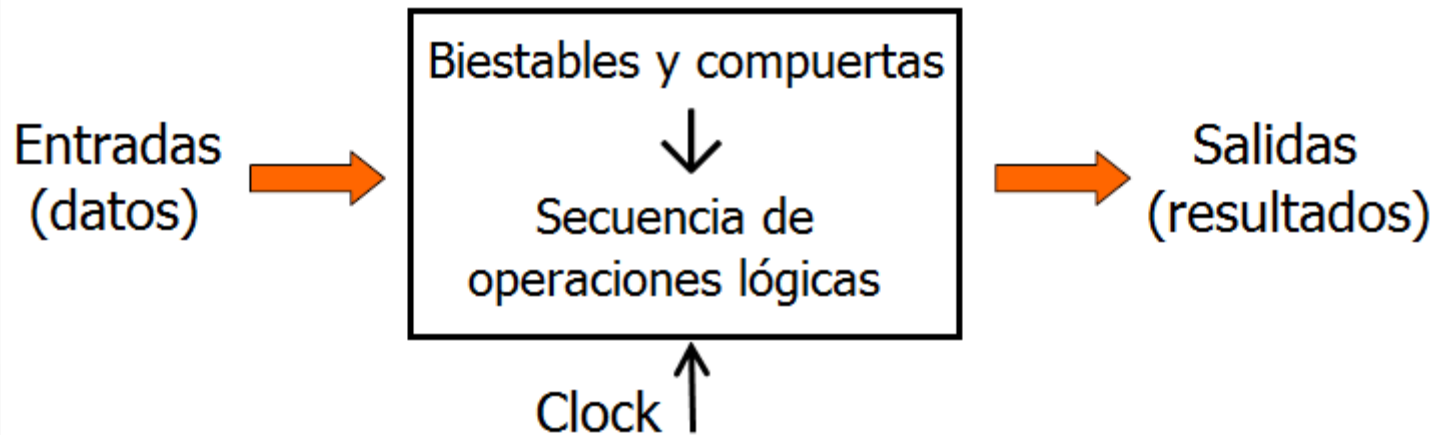
Los circuitos secuenciales síncronos vistos poseen un número determinado de elementos de memoria. Esto significa que existe un número finito de estados posibles del circuito, dado por la cantidad máxima de posibles combinaciones de valores de sus flip flops (para n FF, será 2^n).



Un circuito como el descrito, que sólo puede evolucionar a través de un conjunto finito de estados posibles, constituye una máquina de estados finitos (MEF).

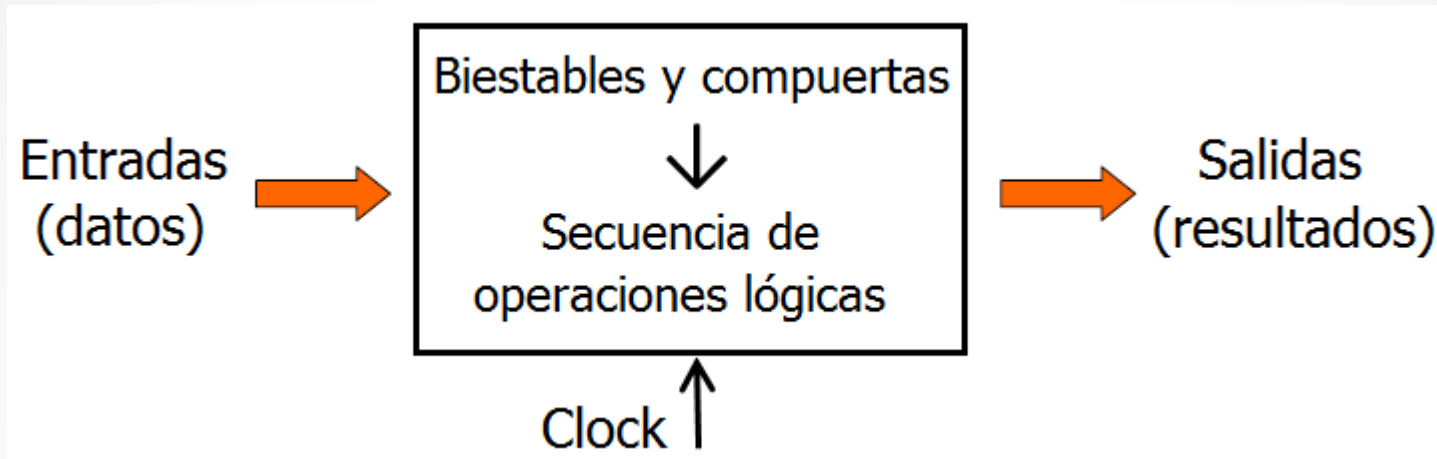
Diseños cableados (1)

Teóricamente, es factible resolver cualquier problema que pueda ser expresado como una *secuencia* mediante una máquina de estados finitos:



Para ello, resulta necesario expresar el problema mediante una diagrama de estados o equivalente, simplificarlo (si es posible), y construir el circuito secuencial.

Diseños cableados (2)



La MEF produce señales digitales que dependen de sus elementos - biestables y compuertas- y de sus conexiones.

Dicho de otro modo: sus tareas están determinadas por el *hardware*.

Por lo tanto, si se modifican las condiciones del problema, o es necesario resolver uno nuevo → es necesario modificar esos biestables y/o compuertas y/o conexiones:

→ cambiar el hardware

Diseños programables (1)

Un problema complejo puede ser resuelto por medio de una secuencia de operaciones más simples, siguiendo un encadenamiento lógico (un *algoritmo*)

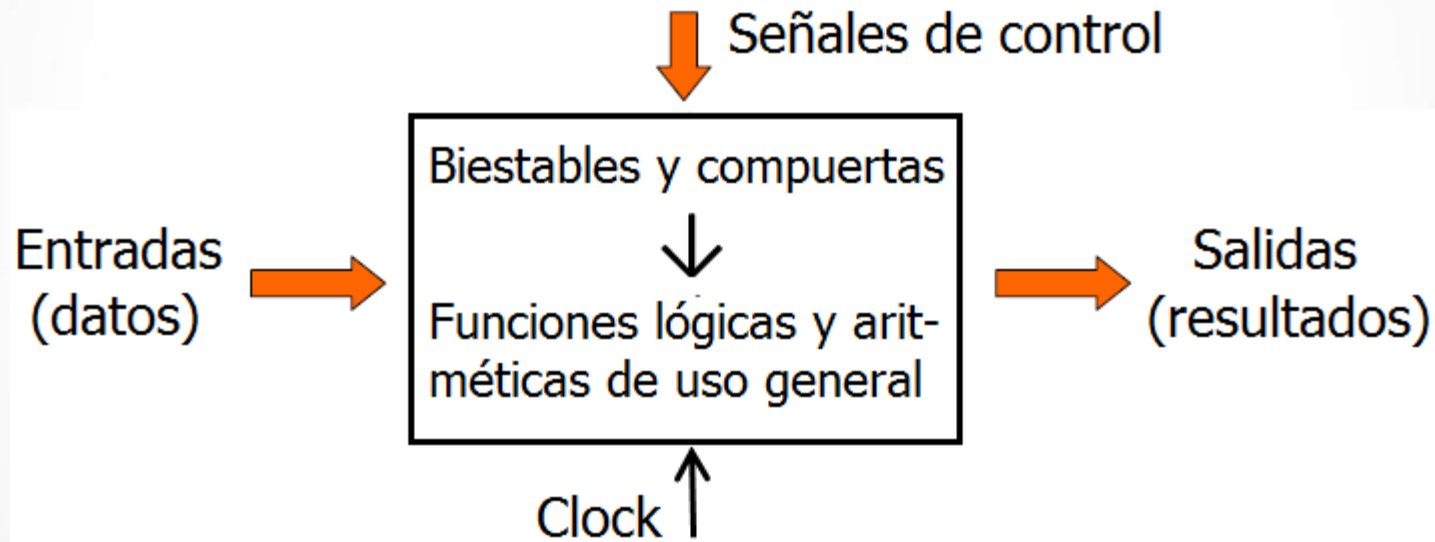
Por ejemplo,...



...una **multiplicación** puede ser planteada como una sucesión de **sumas** y **desplazamientos** de los dígitos de los operandos

Diseños programables (2)

Una filosofía de diseño alternativa a los diseños cableados consiste en un circuito también compuesto por biestables y compuertas, **pero de estructura fija**, capaz de realizar un **conjunto limitado y predeterminado de operaciones lógicas y aritméticas**.



El *movimiento de datos* dentro de la estructura y la *operación* específica que el circuito realizará en cada ciclo del reloj está determinado por un conjunto de **señales de control externas**

Diseños programables (3)

Las *operaciones* lógicas o aritméticas (complementos, sumas, desplazamientos, etc.) o los *movimientos* de datos dentro de la estructura, que se realizan en un ciclo de reloj, se denominan **microoperaciones**.

→ Puede realizarse más de una microoperación por ciclo de reloj

El *conjunto de señales de control* que determinan *qué microoperación o microoperaciones* se realizan en cada ciclo de reloj se denomina **palabra de control** o **microinstrucción**.

Un *conjunto de microoperaciones* puede agruparse en una secuencia fija para desarrollar una operación más compleja, llamada **instrucción**.

Finalmente, un *conjunto de instrucciones* pueden combinarse *libremente* para, mediante algoritmos, crear un **programa**.

Diseños programables (4)

Las salidas del circuito dependen, en cada estado, de las funciones lógicas y estados de sus biestables y compuertas y de sus interconexiones –como en un diseño cableado– , pero además, **de las señales de control provenientes de la instrucción.**

Dicho de otro modo: sus salidas se encuentran programadas por el *software*.

Ahora, si cambian las condiciones del problema, –o si hay que resolver uno nuevo– es posible **utilizar el mismo hardware, modificando sólo el software.**

→ Surge el concepto de *programación*.

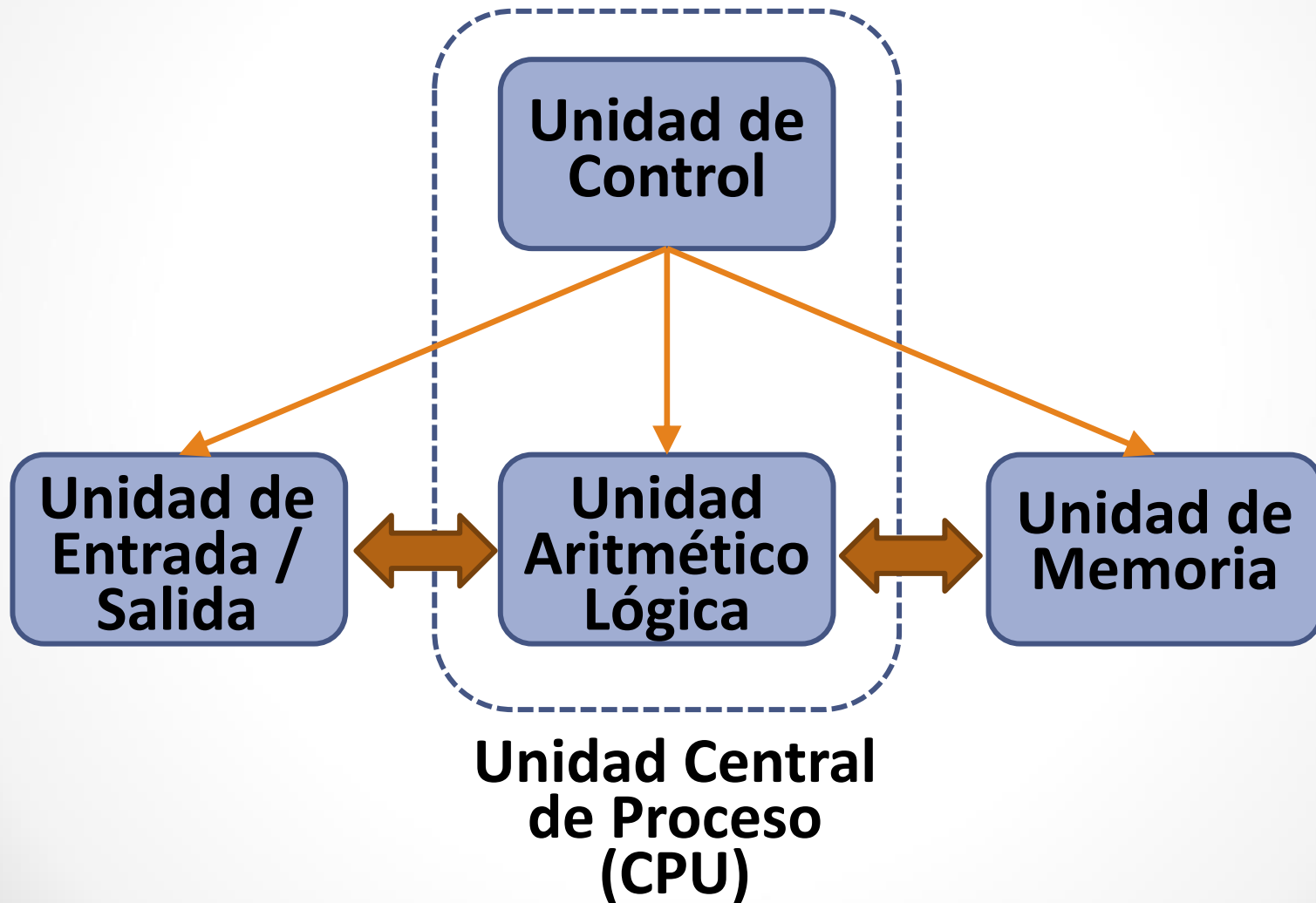
Modelo Von Neumann

- Su funcionamiento se basa en el concepto de **programa almacenado en memoria**. La memoria principal almacena:
 - ✓ **Instrucciones**: cadena de bits que controla el funcionamiento del computador.
 - ✓ **Datos**: que procesa y genera cada instrucción.
- Las palabras en memoria siguen una **organización lineal**.
 - ✓ Todas las palabras de memoria tienen el mismo tamaño.
 - ✓ No hay distinción explícita entre direcciones y datos.

Modelo Von Neumann

- La ejecución de las instrucciones es **secuencial**:
 - ✓ El secuenciamiento de las instrucciones es implícito, y viene determinado por el orden en que han sido almacenadas en la memoria. Este secuenciamiento sólo puede ser modificado por instrucciones específicas de salto.
 - ✓ El contador de programa (PC) indica en cada instante cual es la siguiente instrucción a ejecutar.

Modelo Von Neumann

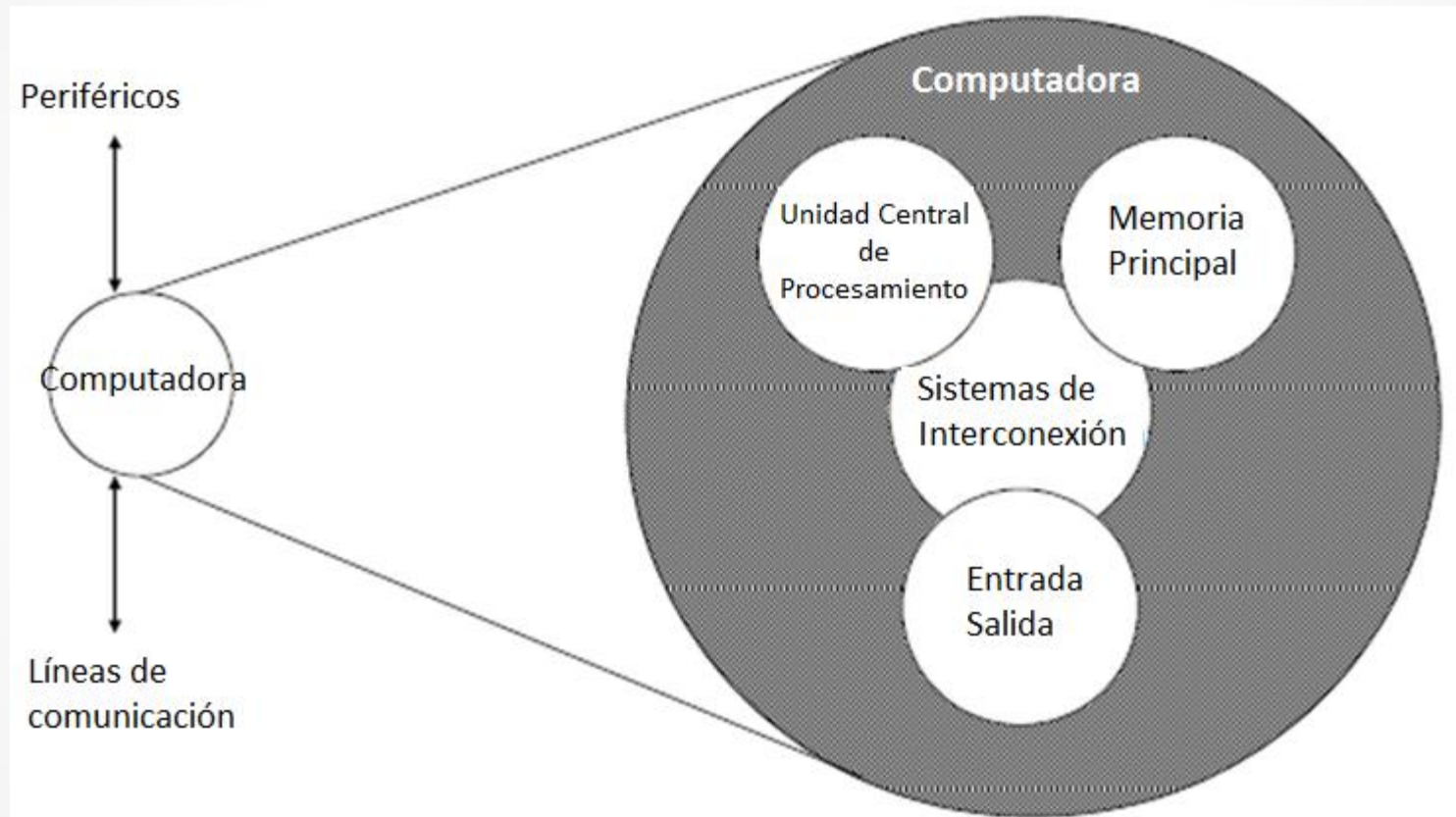


Componentes del computador

Para implementar una máquina programable con las especificaciones funcionales anteriores, una computadora utiliza componentes específicos:

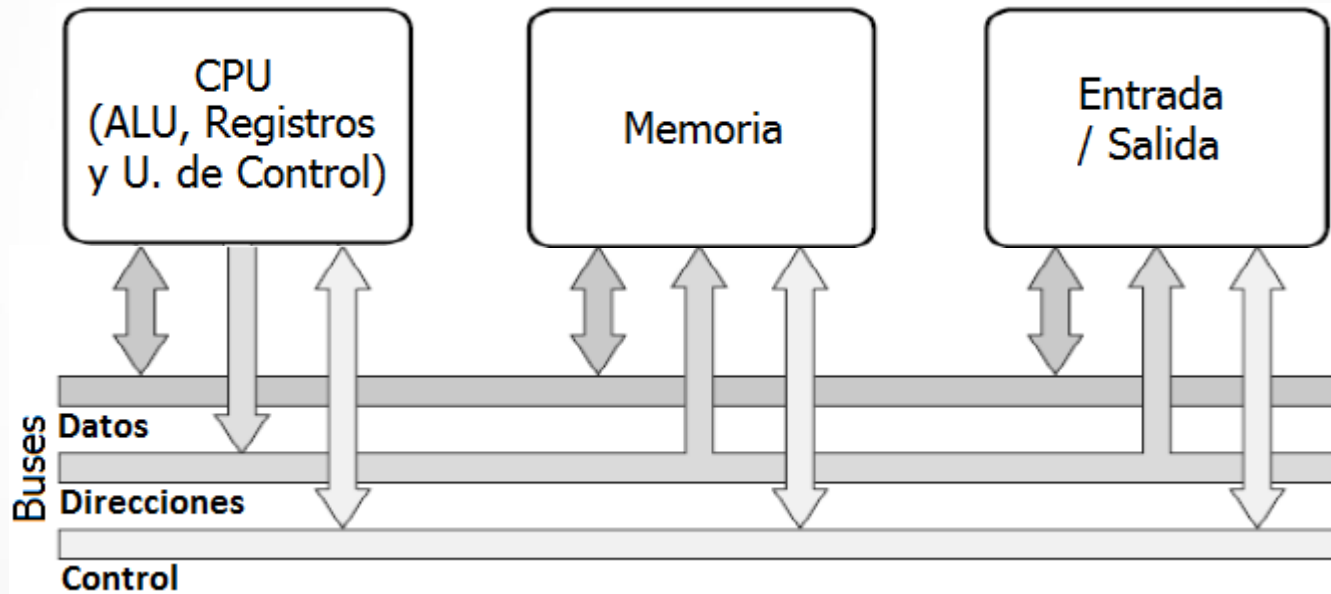
- Un módulo de ***Entrada / Salida***, por el que ingresan y egresan los datos;
- Un módulo de ***Memoria***, en el que se almacenan datos de entrada y resultados procesados, así como las instrucciones del programa;
- Una ***Unidad Central de Proceso (CPU)***, en la que se interpretan las instrucciones del programa, se realizan las operaciones lógicas y matemáticas sobre los datos provenientes de los módulos anteriores y se envían a estos los resultados de ese procesamiento.

Componentes del computador



Componentes del computador

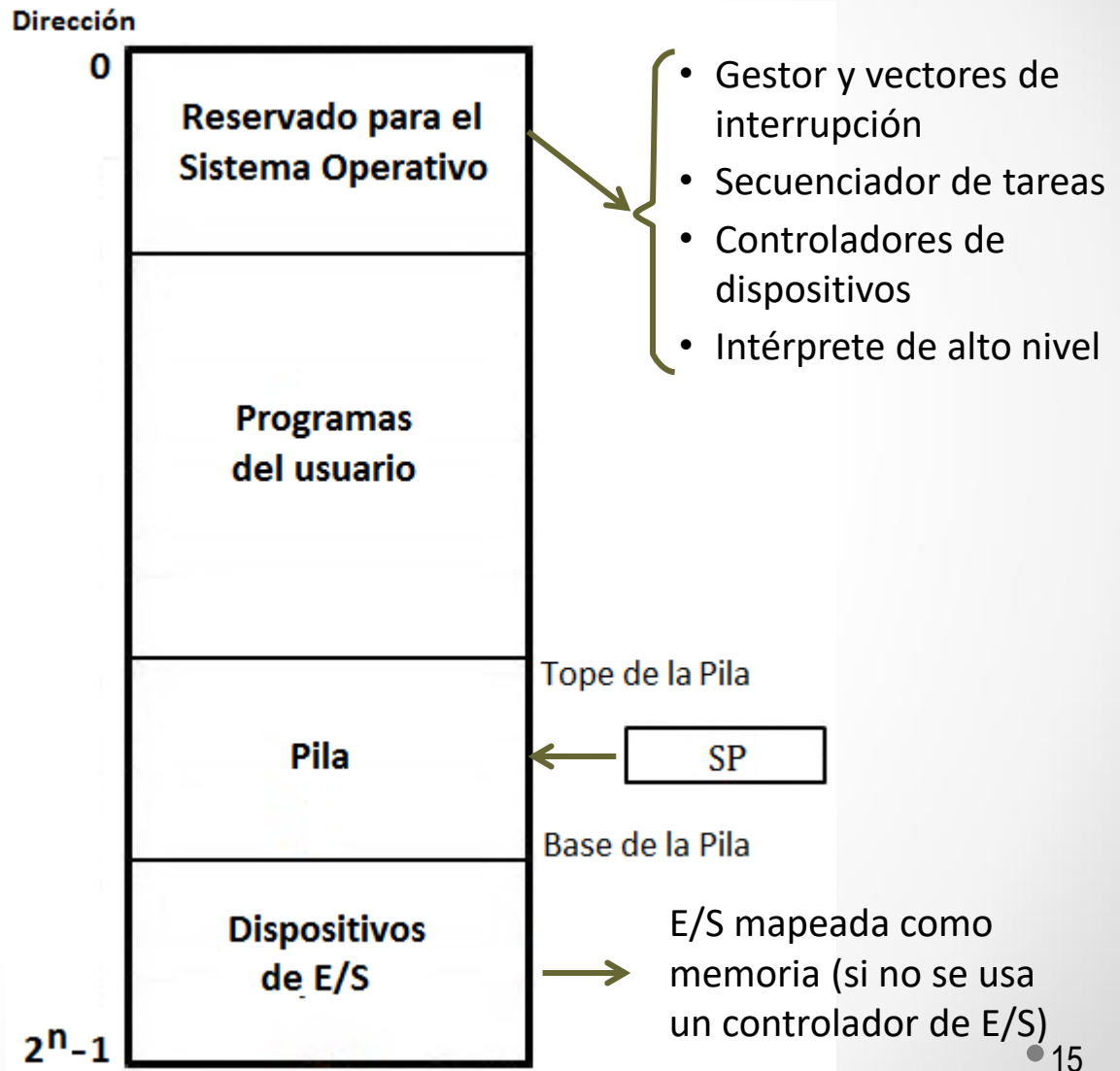
La Unidad Aritmético-Lógica, los Registros, y la Unidad de Control forman la Unidad Central de Procesamiento (CPU)



Finalmente, los tres módulos (CPU, Memoria y E/S), se interconectan a través de tres conjuntos de líneas: los buses de Datos, de Direcciones y de Control

Organización de la memoria principal

En general, el uso de la memoria principal se organiza con zonas reservadas, para el Sistema Operativo (o una parte de éste); la pila, y los dispositivos de E/S (si se mapean como memoria). Esto significa que la cantidad de memoria disponible para la ejecución de los programas del usuario es menor que el tamaño de la memoria. (n = número de líneas de dirección)

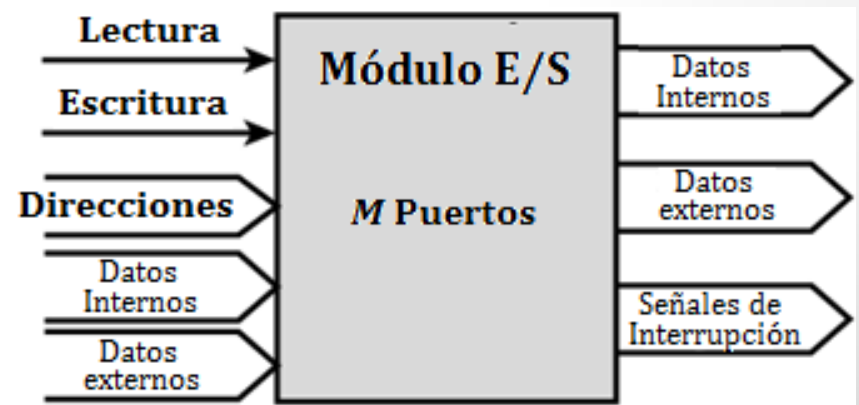
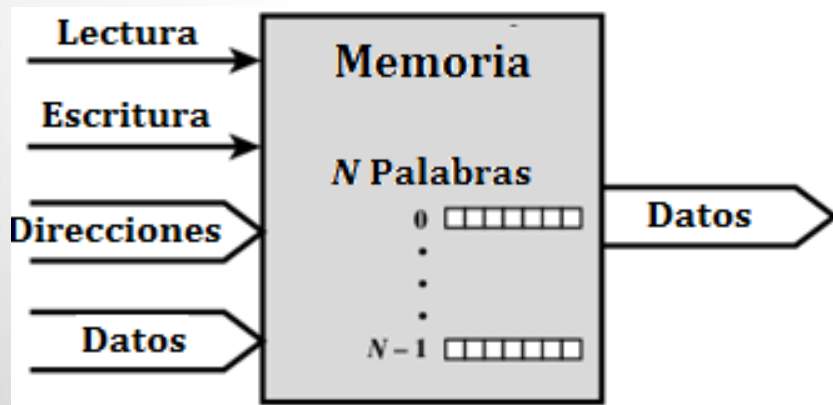


Módulo de Entrada/Salida (E/S)

Es el módulo que maneja los dispositivos con los que el sistema intercambia información con el exterior. Funcionalmente, es similar a la memoria.

Recibe y envía datos de y al sistema. En general, puede manejar M dispositivos externos a través de *Puertos*, seleccionados por líneas de dirección del sistema. Posee un conjunto de líneas de datos entrantes y salientes, uno por cada puerto.

Genera además las señales de interrupción a la CPU para cada dispositivo.



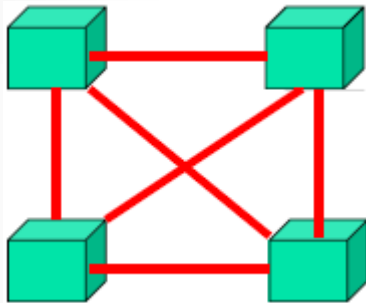
Estructuras de Interconexión

La naturaleza de los subsistemas del computador dan lugar a los siguientes tipos de transferencia:

- **Memoria a procesador:** el procesador lee una instrucción o un dato desde la memoria.
- **Procesador a memoria:** el procesador escribe un dato en memoria
- **E/S a procesador:** Este lee datos de un dispositivo de E/S a través de un modulo de E/S.
- **Procesador a E/S:** el procesador envía datos a un módulo de E/S.
- **Memoria a E/S y viceversa:** Un módulo de E/S puede intercambiar datos directamente con la memoria, sin tener que pasar a través del procesador, utilizando acceso directo a memoria(DMA, *Direct Memory Access*) .

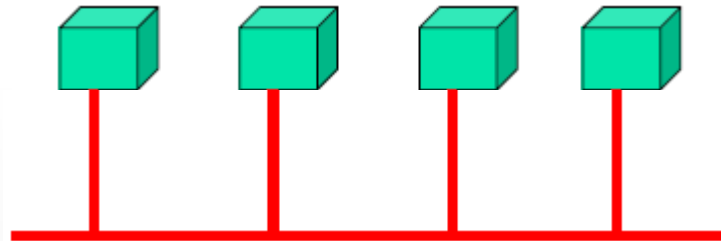
Estructuras de Interconexión

Por lo anterior, es necesario interconectar los componentes. Una opción “natural” es:



Un esquema de conexiones independientes entre los dispositivos hace que la comunicación entre ellos sea rápida. Pero tiene el problema de que si se agrega uno nuevo, hay que generar las conexiones a todos los existentes.

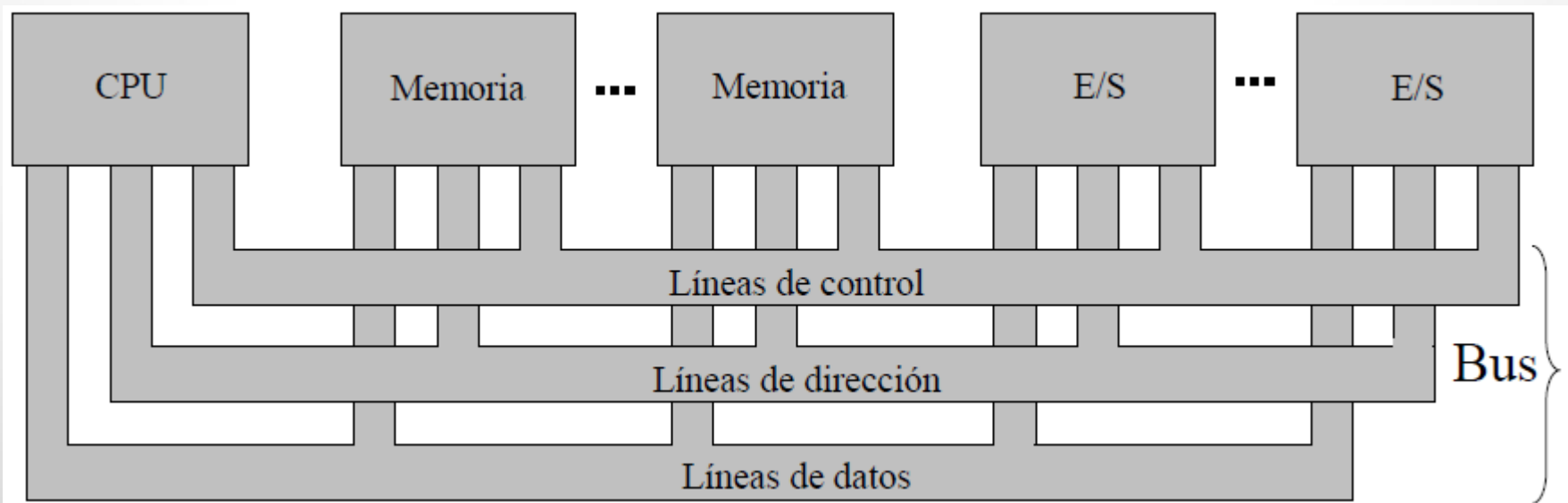
Una solución que simplifica este problema es la utilización de un medio compartido:



Buses

Un bus es un camino de comunicación que interconecta dos o más dispositivos.

- Usualmente, son “broadcast” (todos pueden verlo)
- Generalmente están agrupados por función, si bien son líneas de comunicación independientes: por ejemplo, un bus de datos de 32 bits son 32 canales de datos separados de un bit cada uno.
- No se incluyen o representan las líneas de energía



Tipos de buses

Direcciones: Identifica la fuente o destino de un dato en la memoria o en los puertos de E/S.

- El ancho del bus de direcciones determina la máxima capacidad de memoria posible en el sistema. Por ejemplo, un bus de 16 bits define un espacio para direcciones de $2^{16} = 64K$ posiciones.

Datos: Transmite datos propiamente dichos o instrucciones.

- El ancho del bus de datos determina las prestaciones del sistema. Por ejemplo, si se utiliza un bus de 8 bits y se debe operar un dato de 16 bits, serán necesarios dos accesos a la memoria.

Control: Transmite información de control y temporización.

- Señal de escritura/lectura en memoria y en dispositivos de E/S
- Petición y reconocimiento de interrupción; petición y cesión de bus.
- Señales de reloj.

Opciones de diseño

Dedicados: uso de líneas separadas para direcciones y para datos.

- 16 líneas de direcciones
- 16 líneas de datos
- 1 línea de control de lectura o escritura (R/W)
 - ✓ Ventaja: menos disputas por acceso al bus.
 - ✓ Desventaja: se incrementa tamaño y precio.

Multiplexados: uso de las mismas líneas en tiempos separados.

- 16 líneas de direcciones o datos
- 1 línea de control de lectura o escritura (R/W)
- 1 línea de control para definir direcciones o datos (A/D)
 - ✓ Ventaja: menos líneas → se reduce tamaño y precio.
 - ✓ Desventajas: se complica la circuitería y se reduce velocidad del computador.

Cuestiones de diseño

Arbitraje: en un momento dado, sólo un módulo puede estar enviando información al bus. El mecanismo necesario es:

- 1) Obtener el uso del bus
- 2) Enviar el dato o la petición de un dato en el caso de una escritura o lectura respectivamente

→ El acceso es secuencial y debe proveerse un mecanismo de ordenamiento sobre que dispositivo lo utiliza.

- **Centralizado:** utiliza un controlador de bus o árbitro (se usa un módulo separado o puede ser parte de la CPU).
- **Distribuido:** cada módulo incluye sistema de control de acceso y entre todos controlan el bus.

Jerarquía de buses

En principio, un bus permite la conexión de un gran número de dispositivos. Sin embargo, con el aumento de conexiones:

1. El bus se hace más lento.

Por razones físicas, buses más cargados tienen que hacerse más largos, y eso provoca más demora en la propagación de las señales a través de él.

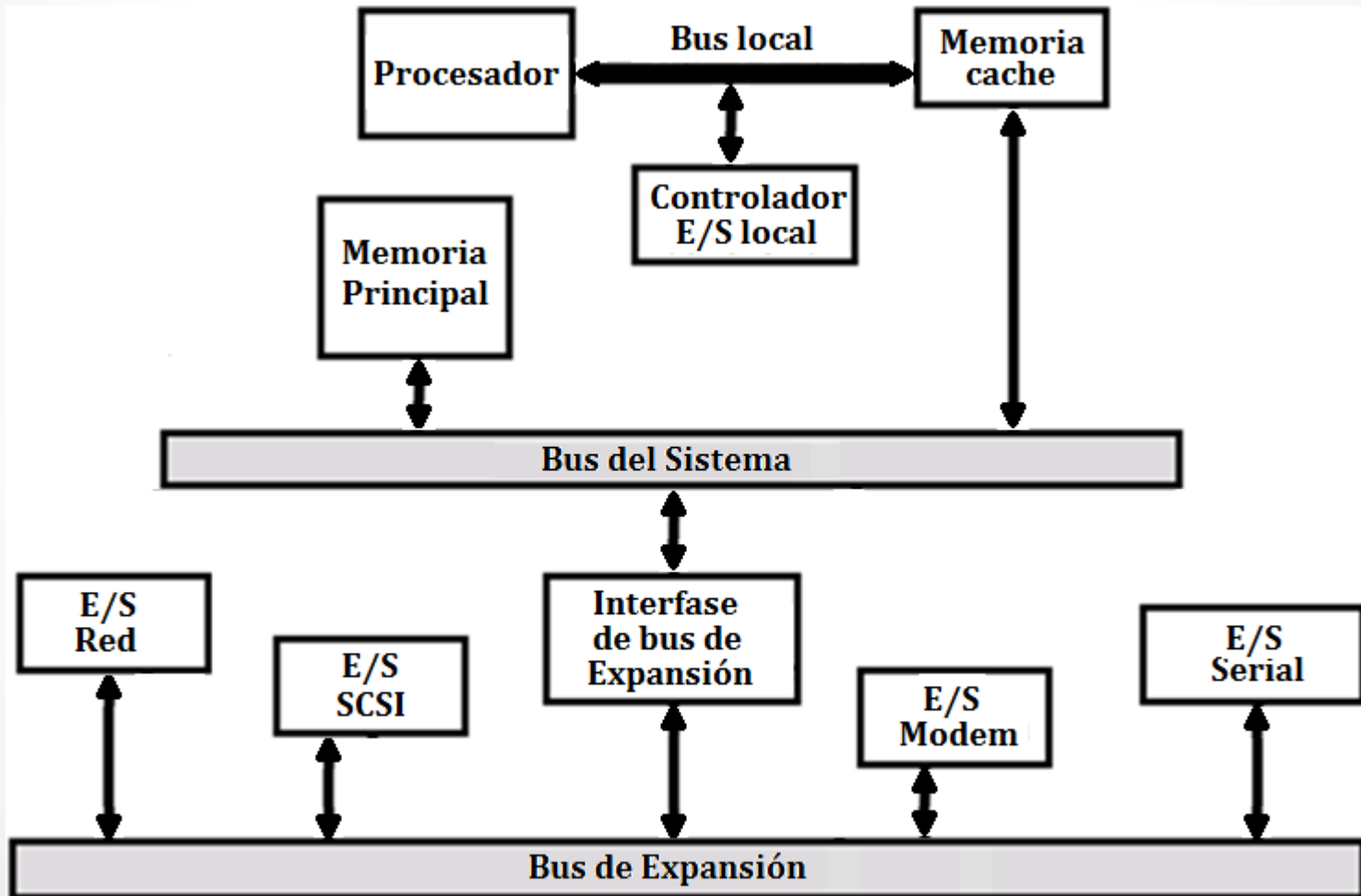
2. Crece la disparidad de velocidades de cada dispositivo.

Los dispositivos son de diferentes velocidades. Si los dispositivos lentos utilizan mucho el bus, la velocidad general será aproximadamente la de ellos.

Por lo tanto, la mayoría de los computadores utilizan varios buses, organizados jerárquicamente de acuerdo a la velocidad de los elementos que se conectan a ellos.

Jerarquía de buses

Los buses “superiores” en la jerarquía interconectan los elementos más rápidos:



Referencias

- Stallings, Williams - Organización y Arquitectura de Computadoras - 5º Ed. - Prentice Hall. Año 2000.

→ *Capítulo 3*