Organización y Arquitectura de las Computadoras (OAC)

Maestro
M.I. Jose Isabel García Rocha

Unidad 2 - Componentes de una computadora (2 Clases)

Del hardware cableado al software programable

1. Hardware específico (programa cableado) *(Figura 3.1a)*

- Los cálculos podían resolverse diseñando un **circuito lógico dedicado** para cada operación.
- Ventajas: alta eficiencia para una tarea concreta.
- Limitaciones: no se puede reutilizar → habría que **reconfigurar el hardware para cada nuevo cálculo**.

2. Hardware de propósito general *(Figura 3.1b)*

- Alternativa: construir una **unidad de funciones aritméticas y lógicas de uso general**.
- Este hardware puede realizar distintas operaciones según las **señales de control** recibidas.
- Requiere un **conjunto de señales de control diferentes para cada paso del cálculo**.

3. Aparición del software

- En lugar de reconfigurar hardware:
  - Se asigna un **código a cada conjunto de señales de control**.
  - Una secuencia de códigos constituye un **programa**.
- El sistema ahora incluye:
  - **Intérprete de instrucciones** (decodifica códigos y genera señales de control).
  - **Unidad aritmético-lógica (ALU)** de propósito general.
- Conjunto = **CPU (Unidad Central de Procesamiento)**.

👉 Este salto permitió pasar del **programa cableado** al **programa en software**.

4. Otros componentes necesarios

1. **Entrada (Input):**
   - Introduce datos e instrucciones en el sistema.
   - Traduce la información a señales comprensibles por la CPU.
2. **Salida (Output):**
   - Entrega los resultados procesados en un formato accesible.
   - Ejemplo: pantallas, impresoras, almacenamiento externo.
3. **Memoria principal:**
   - Almacena **instrucciones y datos** en la misma unidad (concepto de von Neumann).
   - Permite ejecución secuencial o alterada (saltos, bifurcaciones).
   - Diferencia con memoria externa: esta es de acceso directo del procesador.

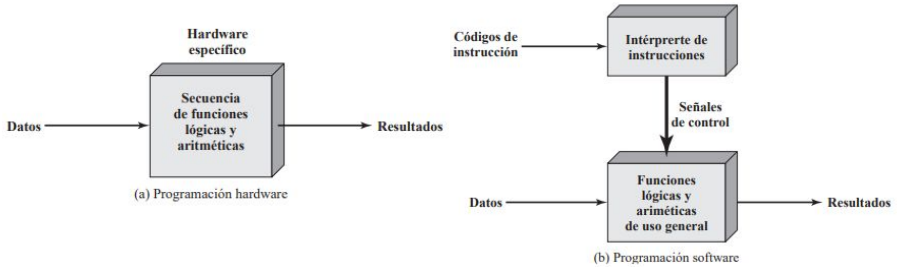


Figura 3.1. Alternativas hardware y software.

Unidad 2 - Componentes de una computadora

Interacción CPU – Memoria – E/S

1. Rol de la CPU en el control

- La **CPU coordina** el funcionamiento general del sistema.
- Se comunica con **memoria** y con **dispositivos de entrada/salida (E/S)** mediante registros internos y buses.

2. Registros internos de la CPU

- **MAR (Memory Address Register):**
  - Guarda la dirección de memoria donde se realizará la **siguiente lectura o escritura**.
- **MBR (Memory Buffer Register):**
  - Contiene el dato que será **escrito en memoria** o el que se **lee desde memoria**.
- **E/S AR (I/O Address Register):**
  - Especifica qué **dispositivo de entrada/salida** se utilizará.
- **E/S BR (I/O Buffer Register):**
  - Almacena temporalmente los datos intercambiados entre CPU y un dispositivo de E/S.

3. Módulo de memoria

- Conjunto de **posiciones numeradas secuencialmente** (direcciones).
- Cada posición almacena un número binario que puede representar:
  - Una **instrucción**.
  - Un **dato**.

4. Módulo de E/S

- Permite la transferencia de datos **entre dispositivos externos ↔ CPU/memoria**.
- Funciones principales:
  - **Entrada:** captar datos desde dispositivos externos.
  - **Salida:** enviar resultados hacia dispositivos externos.
- Incluye **buffers internos** para almacenar datos temporalmente antes de transferirlos.

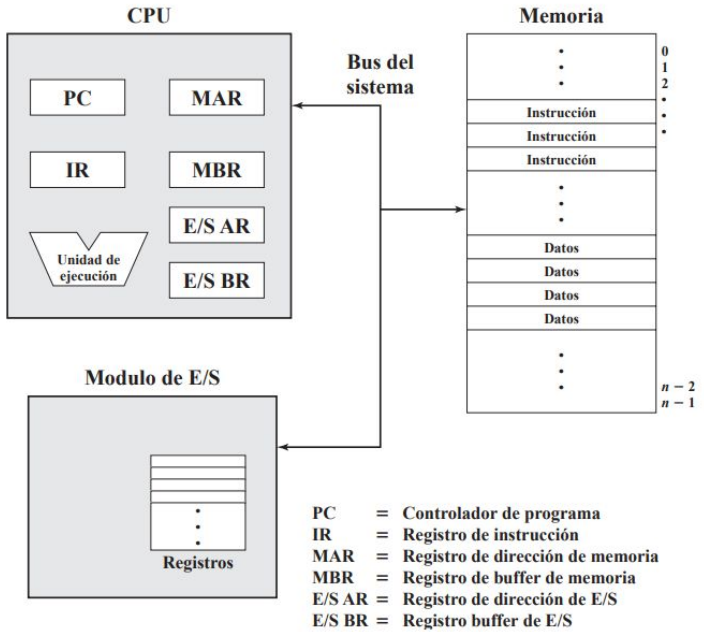


Figura 3.2. Componentes del computador: esquema de dos niveles.

Unidad 2 - Componentes de una computadora

Ciclos de Captación y Ejecución

1. Inicio del ciclo de instrucción

- El ciclo comienza con la **captación (fetch)** de la instrucción desde memoria.
- La CPU usa el **Contador de Programa (PC, Program Counter)** para localizar la instrucción.
- Después de captar la instrucción, el **PC se incrementa** para apuntar a la siguiente instrucción en memoria (ejemplo: posiciones 300, 301, 302...).
- Excepción: si ocurre un salto, el PC se actualiza con la nueva dirección.

2. Registros involucrados

- **IR (Instruction Register):** almacena la instrucción captada.
- **PC (Program Counter):** dirección de la próxima instrucción.
- **AC (Acumulador):** almacenamiento temporal para operaciones aritméticas/lógicas.
- **MAR (Memory Address Register) y MBR (Memory Buffer Register):** registros auxiliares para la transferencia de datos con memoria.

3. Tipos de operaciones que puede ejecutar la CPU

1. **Procesador-Memoria:** transferir datos entre CPU ↔ Memoria.
2. **Procesador-E/S:** transferir datos entre CPU ↔ dispositivos externos.
3. **Procesamiento de datos:** operaciones aritméticas/lógicas en la ALU.
4. **Control:** alteración del flujo del programa (ej. saltos, interrupciones).

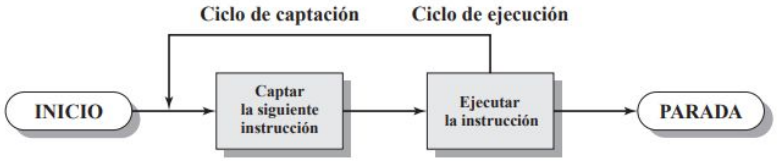


Figura 3.3. Ciclo de instrucción básico.

Unidad 2 - Componentes de una computadora

Ciclos de Captación y Ejecución

4. Ejemplo con máquina hipotética

- Instrucciones y datos de **16 bits**.
- Memoria organizada en **palabras de 16 bits**.
- Conjunto de hasta **16 códigos de operación** (codops) (2^4).
- Espacio de direcciones: **4K palabras (4096)** (2^12).

Fragmento de programa (suma en memoria):

- Sumar el contenido de la dirección **940** con el de la dirección **941**, y guardar el resultado en **941**.

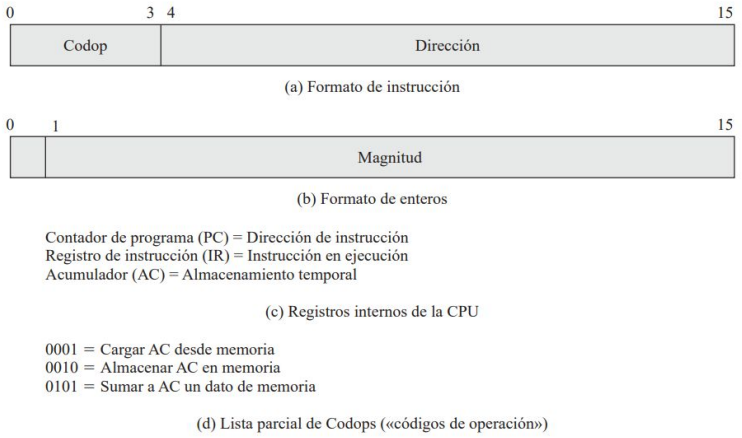


Figura 3.4. Características de una máquina hipotética.

0101 -> 5h

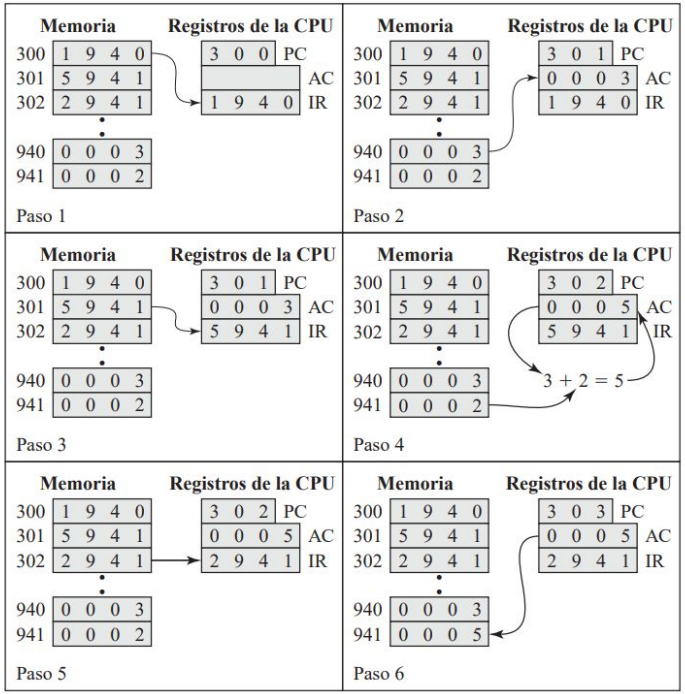


Figura 3.5. Ejemplo de ejecución de un programa (contenidos de la memoria y de los registros en hexadecimal).

Unidad 2 - Componentes de una computadora

Ciclos de Captación y Ejecución

5. Ejemplo más complejo (PDP-11)

- Instrucción simbólica: ADD B,A
- Efecto: suma contenidos de memoria en B y A, y guarda en A.
- En un único ciclo de instrucción:
  1. Se capta ADD.
  2. Se leen operandos A y B desde memoria.
  3. Se ejecuta la suma en la CPU.
  4. Se escribe el resultado en A.
- Requiere más de una referencia a memoria dentro del mismo ciclo.

6. Estados detallados del ciclo de instrucción (Figura 3.6)

1. **IAC (Instruction Address Calculation):** cálculo de la dirección de la instrucción.
   - Normalmente: PC+1.
2. **IF (Instruction Fetch):** captación de la instrucción desde memoria.
3. **IOD (Instruction Operation Decoding):** decodificación de la instrucción → qué operación y qué operandos.
4. **OAC (Operand Address Calculation):** cálculo de dirección del operando.
5. **OF (Operand Fetch):** obtención del operando desde memoria o E/S.
6. **DO (Data Operation):** ejecución de la operación en la ALU.
7. **OS (Operand Store):** almacenamiento del resultado en memoria o salida a un dispositivo.

- Algunos estados pueden repetirse (ej. múltiples operandos).
- No todas las instrucciones pasan por todos los estados.
- Ejemplo: ADD A,B → secuencia: IAC → IF → IOD → OAC → OF → OAC → OF → DO → OAC → OS.

7. Consideraciones adicionales

- Una instrucción puede implicar **múltiples accesos a memoria** (operandos, resultados).
- Algunas arquitecturas permiten operaciones con vectores o cadenas en un solo ciclo de instrucción, lo cual implica repetición de estados de **captación/almacenamiento de operandos**.
- El ciclo de instrucción es un **diagrama de estados**, donde la CPU avanza paso a paso, pero no siempre pasa por todos.

📌 **En conclusión:**
El ciclo de instrucción básico es **captar → decodificar → ejecutar**, pero en la práctica puede involucrar varios sub-estados, múltiples accesos a memoria y operaciones de control. Los registros **PC, IR, AC, MAR, MBR** son fundamentales para llevar el control de cada etapa.

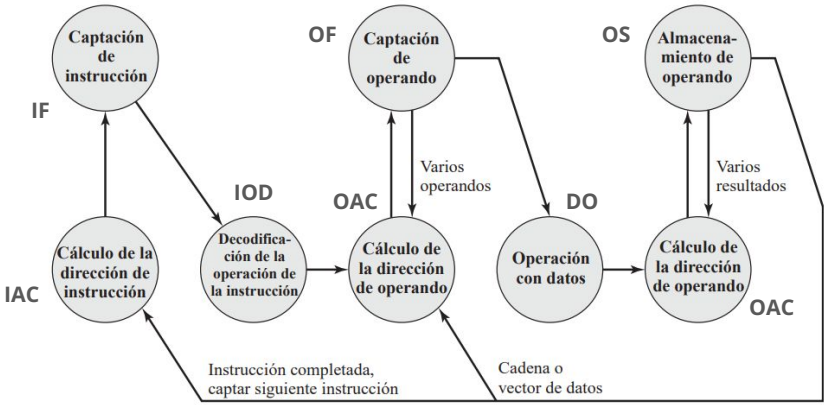
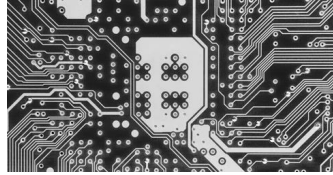


Figura 3.6. Diagrama de estados del ciclo de intrucción.

Unidad 2 - Componentes de una computadora



Ductos de intercomunicación

1. Concepto general

- Un computador está formado por tres módulos básicos:
  - **Procesador (CPU)**
  - **Memoria**
  - **Entrada/Salida (E/S)**
- Estos módulos se comunican entre sí mediante un **conjunto de líneas físicas**, llamadas **estructura de interconexión**.
- La forma en que se diseñe esta estructura depende del tipo de intercambios necesarios entre los módulos.

2. Funcionalidad de los módulos

- **Memoria:**
  - Consta de *N palabras* de igual longitud, cada una con una **dirección única** (0…N−1).
  - Operaciones básicas: **Lectura (Read)** y **Escritura (Write)**.
  - La dirección selecciona qué posición de memoria se usa.
- **Módulos de E/S:**
  - Desde el punto de vista interno, funcionan de manera similar a la memoria.
  - Operaciones: **leer** o **escribir** datos.
  - Pueden manejar varios dispositivos externos → cada dispositivo tiene un **puerto (port)** con dirección propia (0…M−1).
  - Incluyen líneas externas para el flujo de datos y la posibilidad de enviar **interrupciones** al procesador.
- **Procesador:**
  - **Lee** instrucciones y datos desde memoria.
  - **Escribe** datos procesados de vuelta a memoria.
  - Envía señales de control para coordinar el sistema.
  - También recibe señales de **interrupción**.

3. Tipos de transferencias en la interconexión

La estructura debe permitir los siguientes flujos:

1. **Memoria → Procesador:** lectura de instrucciones o datos.
2. **Procesador → Memoria:** escritura de resultados en memoria.
3. **E/S → Procesador:** lectura de datos desde dispositivos externos.
4. **Procesador → E/S:** envío de datos a dispositivos externos.
5. **Memoria ↔ E/S (DMA):** transferencia directa entre memoria y módulos de E/S, sin pasar por el procesador (**Acceso Directo a Memoria**).

4. Estructuras de interconexión comunes

- A lo largo de la evolución de los computadores se han probado diversas estructuras.
- Las más utilizadas son:
  - **Bus único compartido:** un conjunto común de líneas de datos, direcciones y control.
  - **Buses múltiples:** más de un bus para reducir cuellos de botella y mejorar rendimiento.

📌 Conclusión:

La estructura de interconexión es esencial porque define cómo se comunican CPU, memoria y E/S. Su eficiencia impacta directamente en el **rendimiento global**, la **latencia** y la **capacidad de expansión** del sistema.

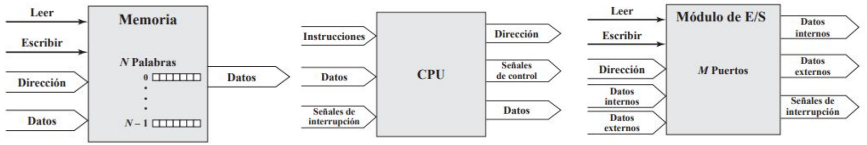


Figura 3.15. Módulos de un computador.

Unidad 2 - Componentes de una computadora

Interconexión con Buses

1. Definición y características

- Un **bus** es un camino de comunicación compartido entre dos o más dispositivos.
- Propiedades clave:
  - **Medio compartido:** todos los dispositivos conectados pueden acceder a las señales.
  - **Conflictos de transmisión:** solo un dispositivo puede transmitir con éxito en un instante dado; si dos lo hacen, las señales se solapan.
  - **Estructura paralela:** un bus consta de varias líneas (conductores eléctricos) que transmiten datos binarios.
  - Ejemplo: un dato de **8 bits** puede transmitirse en paralelo usando 8 líneas.

2. Tipos de buses en un computador

- Los computadores usan diferentes buses en distintos niveles de la jerarquía.
- El más importante: **Bus del sistema (System Bus)** → conecta **CPU, memoria y módulos de E/S**.
- Algunas arquitecturas usan **buses múltiples** para mejorar el rendimiento.

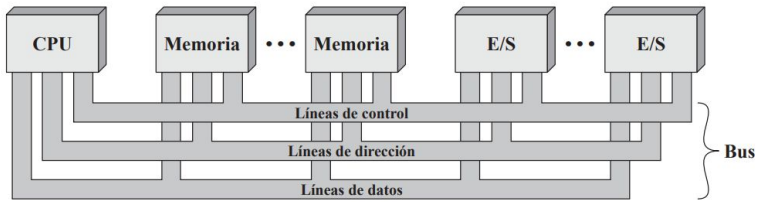


Figura 3.16. Esquema de interconexión mediante un bus.

3. Estructura del bus del sistema

- Generalmente formado por **50 a 100 líneas**.
- Clasificación de las líneas:

1. **Líneas de datos (Data Bus):**
   - Transportan la información (datos o instrucciones).
   - Su número define la **anchura del bus** (ej. 32, 64, 128 bits).
   - Cuanto mayor la anchura, más datos pueden transferirse en paralelo → **mejor rendimiento**.
   - Ejemplo: si el bus tiene 8 bits y la instrucción es de 16 bits, se necesitan 2 accesos a memoria.
2. **Líneas de direcciones (Address Bus):**
   - Indican el **origen o destino** de los datos en el bus.
   - Determinan la **capacidad máxima de memoria** que puede direccionar el sistema.
   - También se usan para direccionar puertos de E/S.
   - Ejemplo: en un bus de 8 bits, las direcciones 0–127 se asignan a memoria y 128–255 a dispositivos de E/S.
3. **Líneas de control (Control Bus):**
   - Coordinan el acceso y el uso de las líneas de datos y direcciones.
   - Incluyen señales de **órdenes** y de **temporización**.
   - Ejemplos típicos:
     - *Memory write* (escribir en memoria).
     - *Memory read* (leer de memoria).
     - *I/O write* (escribir en dispositivo de E/S).
     - *I/O read* (leer desde dispositivo de E/S).
     - *Bus request* / *Bus grant* (petición y cesión de bus).
     - *Interrupt request* / *Interrupt ACK* (gestión de interrupciones).
     - *Transfer ACK* (dato aceptado o colocado en el bus).
     - *Clock* (sincronización).
     - *Reset* (estado inicial de los módulos).

Unidad 2 - Componentes de una computadora

Interconexión con Buses

4. Funcionamiento básico del bus

- **Para enviar un dato:**
  1. El módulo obtiene el control del bus.
  2. Coloca el dato en las líneas de datos.
- **Para solicitar un dato:**
  1. El módulo obtiene el control del bus.
  2. Coloca la petición en las líneas de dirección y control.
  3. Espera a que el otro módulo coloque la respuesta en el bus.

5. Implementación física

- El bus de sistema se implementa con **conductores eléctricos paralelos** en tarjetas de circuito impreso.
- Disposición clásica:
  - Dos columnas verticales de conductores.
  - Ranuras horizontales (**slots**) donde se insertan tarjetas de memoria, procesador, o E/S.
  - Permite **expansión**: añadir memoria o módulos de E/S con nuevas tarjetas.
  - Fácil **mantenimiento**: si una tarjeta falla, se reemplaza.
- En sistemas modernos:
  - Muchos buses se integran **en el chip (on-chip)** → CPU y caché.
  - Otros buses se incluyen en la **tarjeta principal (on-board)**.

📌 **Conclusión:**

El bus es la **columna vertebral de comunicación** en un computador. Su diseño (anchura, número de líneas, control de acceso) impacta directamente en el **rendimiento**, la **capacidad de memoria direccionable** y la **flexibilidad de expansión** del sistema.

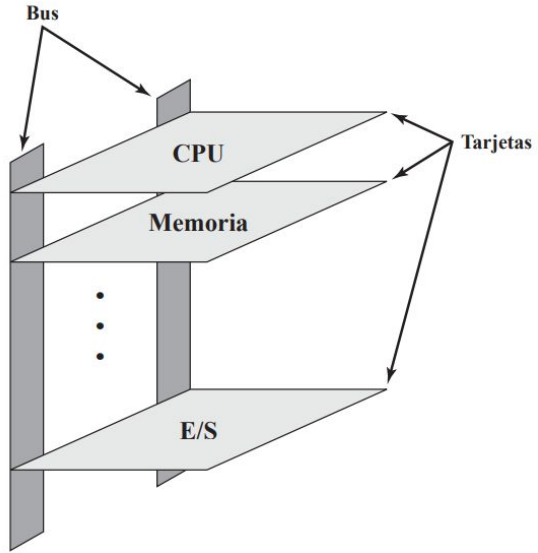


Figura 3.17. Implementación física típica de una arquitectura de bus.

Unidad 2 - Componentes de una computadora

Memoria en los computadores

1. Diversidad de las memorias

- Las memorias son uno de los componentes más variados en:
  - **Tipos**
  - **Tecnología**
  - **Estructura**
  - **Prestaciones**
  - **Costo**
- Ninguna tecnología por sí sola cubre todas las necesidades → se utiliza una **jerarquía de memoria**.
- Clasificación general:
  - **Memoria interna:** accesible directamente por la CPU (registros, caché, memoria principal).
  - **Memoria externa:** accesible mediante E/S (discos, cintas, SSD).

2. Características clave de los sistemas de memoria

- **Ubicación:**
  - Interna → CPU y módulos directamente accesibles.
  - Externa → dispositivos periféricos de almacenamiento.
- **Capacidad:**
  - Expresada en **bytes** o **palabras**.
  - Ejemplos de longitudes de palabra: 8, 16, 32, 64 bits.
- **Unidad de transferencia:**
  - Número de bits leídos/escritos en una operación.
  - Puede ser: palabra, múltiplos de palabra o bloques completos (memoria externa).
- **Número de direcciones:**
  - Byte o palabra.
  - Relación entre **longitud de dirección (A)** y **número de direcciones (N):**

$$2^A = N$$

3. Métodos de acceso

- **Acceso secuencial:**
  - Se recorren los registros en orden hasta llegar al deseado.
  - Ejemplo: cintas magnéticas.
  - Tiempo de acceso: muy variable.
- **Acceso directo:**
  - Cada bloque tiene dirección única basada en su posición física.
  - Ejemplo: discos magnéticos.
  - Tiempo de acceso: depende de la localización física (vecindad + búsqueda).
- **Acceso aleatorio (RAM):**
  - Cada posición tiene su propio mecanismo de direccionamiento.
  - Tiempo de acceso constante, independiente de accesos previos.
  - Ejemplo: memoria principal.
- **Acceso asociativo:**
  - Similar al aleatorio, pero se busca por contenido en vez de por dirección.
  - Recupera datos comparando ciertas posiciones de bits en paralelo.
  - Ejemplo: caché con acceso por contenido.

Unidad 2 - Componentes de una computadora

Memoria en los computadores

4. Parámetros de rendimiento

1. **Tiempo de acceso (latencia):**
   - Tiempo entre la petición y la disponibilidad/almacenamiento del dato.
2. **Tiempo de ciclo de memoria:**
   - Tiempo de acceso + retardo adicional antes de un nuevo acceso.
   - Depende de las líneas de señal y regeneración de datos.
3. **Velocidad de transferencia (R):**
   - Tasa a la que se transfieren datos (bits por segundo).

5. Tipos según tecnología y persistencia

- **Semiconductoras:**
  - Volátiles (RAM: pierden datos sin energía).
  - No volátiles (ROM, Flash).
- **Magnéticas:**
  - No volátiles (discos, cintas).
- **Ópticas y magneto-ópticas:**
  - CDs, DVDs, Blu-ray.
- **Clasificación adicional:**
  - **Volátiles:** requieren alimentación para mantener datos.
  - **No volátiles:** retienen la información sin energía.
  - **No borrables:** su contenido no puede modificarse (ej. ROM)

Tipo de memoria	Clase	Borrado	Mecanismos de escritura	Volatilidad
Memoria de acceso aleatorio (RAM)	Memoria de lectura/escritura	Eléctricamente por bytes	Eléctricamente	Volátil
Memoria de sólo lectura (ROM)	Memoria de sólo lectura	No posible	Mediante máscaras	No volátil
ROM programable (PROM)			Eléctricamente	
PROM borrable (EPROM)	Memoria de sobre-todo-lectura	Luz ultravioleta, chip completo		
Memoria FLASH		Eléctricamente, por bloques		
PROM borrable eléctricamente (EEPROM)		Eléctricamente, por bytes		

6. Organización

- La organización define cómo se disponen los **bits en palabras y bloques**.
- Aunque la forma obvia es organizar la memoria en palabras completas, en la práctica pueden usarse otras disposiciones para optimizar velocidad y eficiencia.

📌 Conclusión:

El diseño de un sistema de memoria implica equilibrar **capacidad, velocidad, costo y persistencia**. Por ello, las computadoras modernas utilizan una **jerarquía de memoria** (registros → caché → RAM → disco → almacenamiento externo).

El elemento básico de una memoria semiconductora es la celda de memoria. Aunque se utilizan diversas tecnologías electrónicas, todas las celdas de memoria semiconductora comparten ciertas propiedades:

- Presentan dos estados estables (o semiestables), que pueden emplearse para representar el 1 y el 0 binarios.
- Puede escribirse en ellas (al menos una vez) para fijar su estado.
- Pueden leerse para detectar su estado.

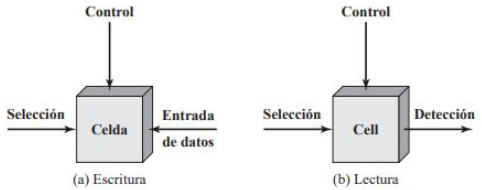
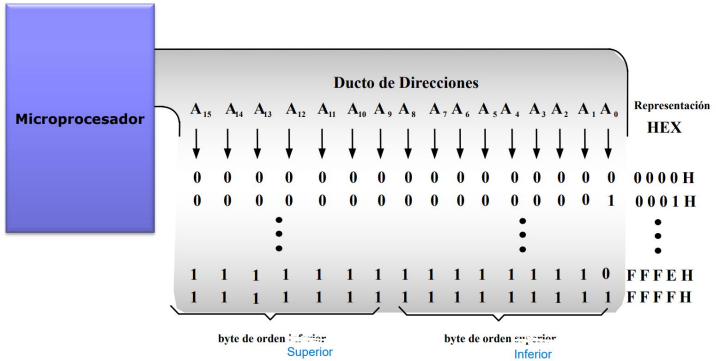


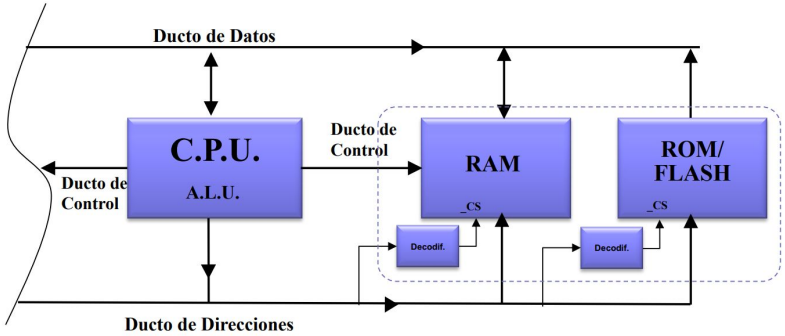
Figura 5.1. Funcionamiento de una celda de memoria.

Unidad 2 - Componentes de una computadora

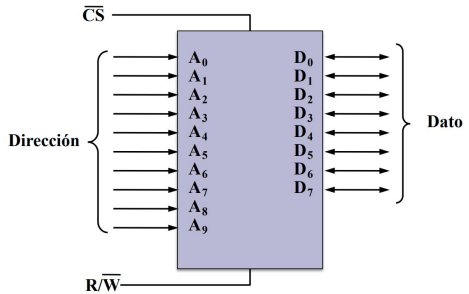
Memoria en los computadores



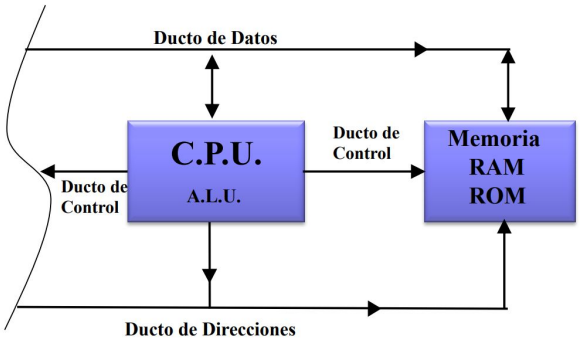
Cantidad de Direcciones = 2^n
donde "n" es el No. de líneas



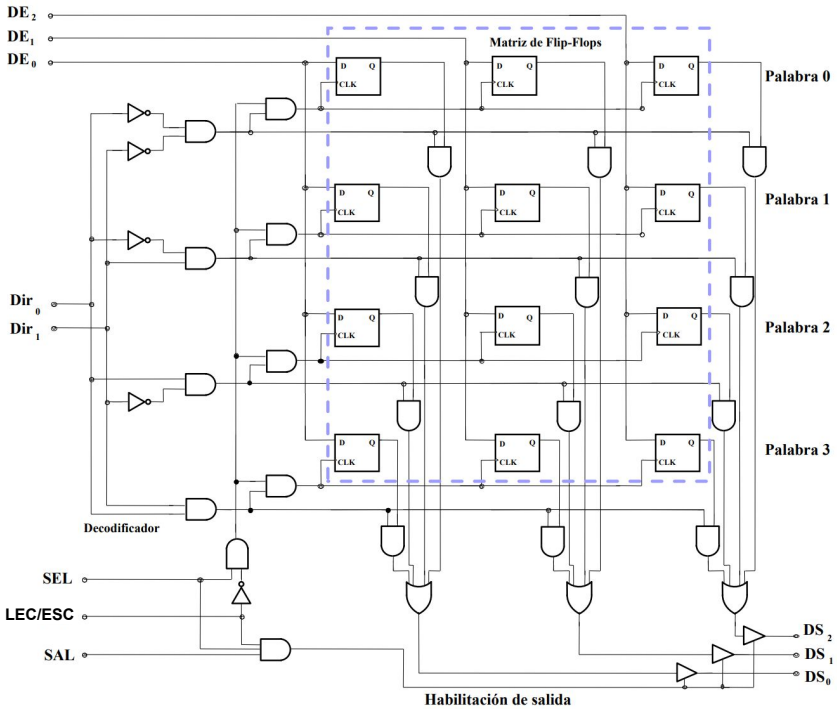
Configuración de una RAM de 1K x 8 bits



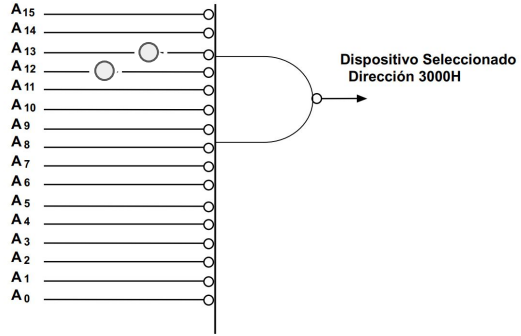
Total de Líneas= 10 (Desde A_0 a A_9)
Total de Direcciones = 2^{10} = 1024
Total de Líneas de Datos = 8 (Desde D_0 a D_7)
Cantidad Total de Memoria = $2^{10} * 8$ = 8192 bits
= 1K Bytes



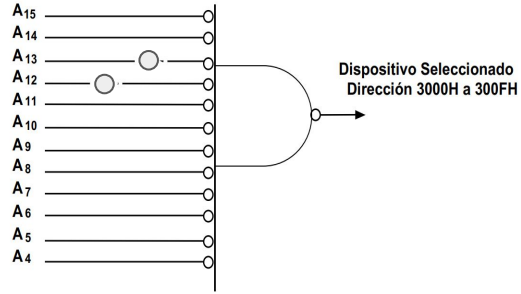
Memoria en los computadores



Decodificadores de direcciones

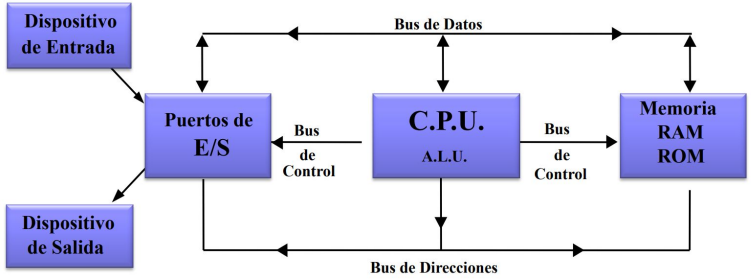


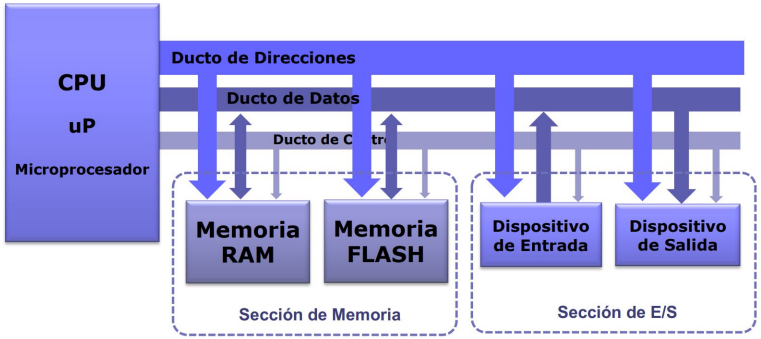
Decodificadores de multiples direcciones

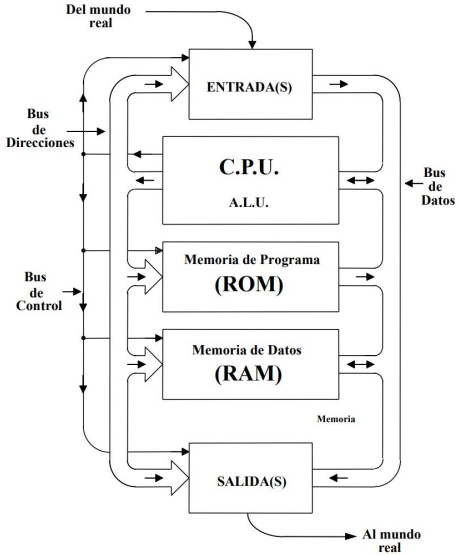


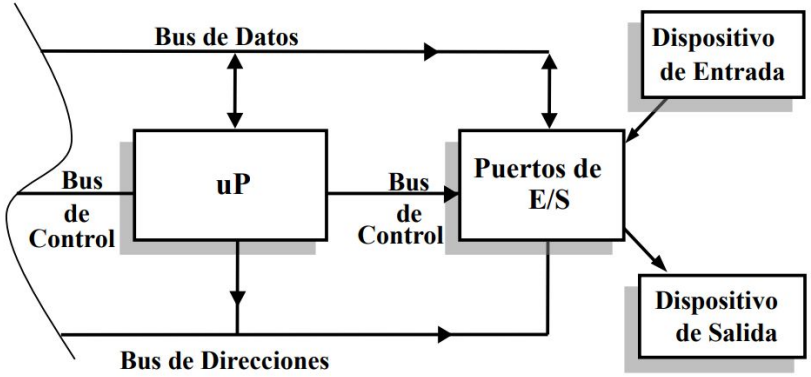
Unidad 2 - Componentes de una computadora

Sección de entrada y salida



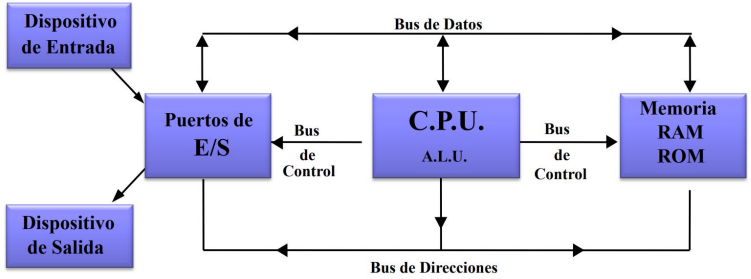


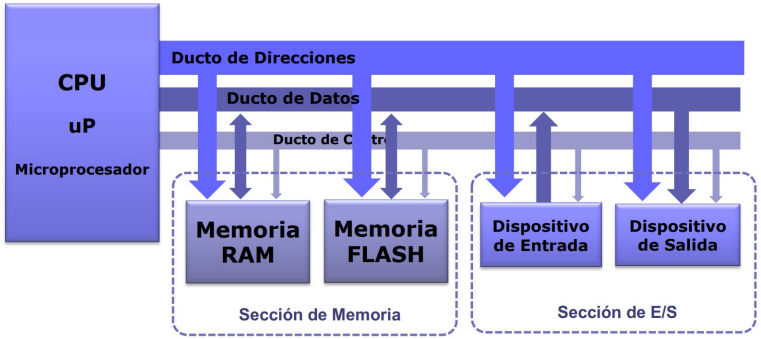


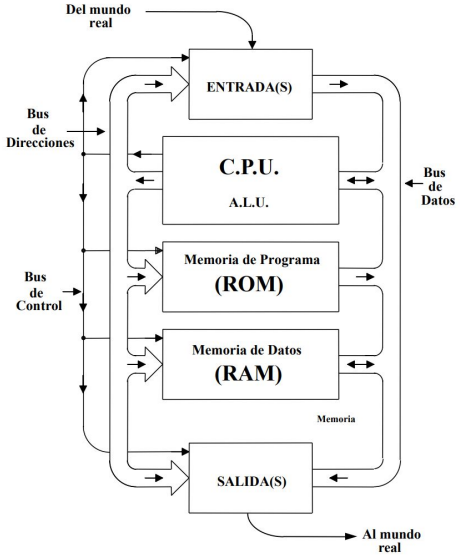


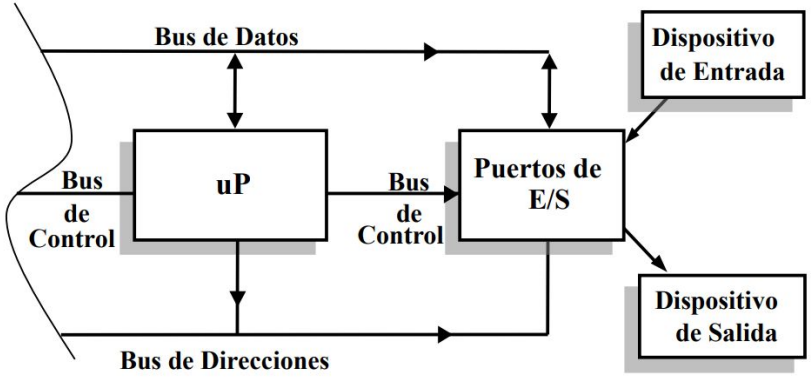
Unidad 2 - Componentes de una computadora

Sección de entrada y salida



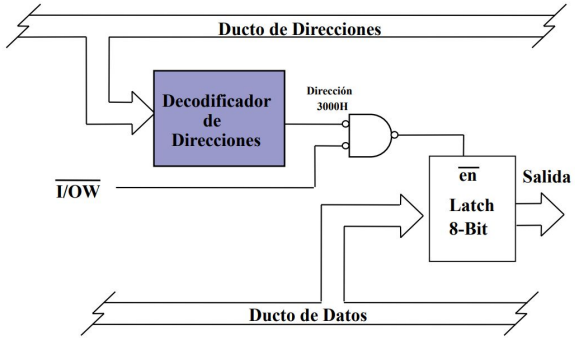






Unidad 2 - Componentes de una computadora

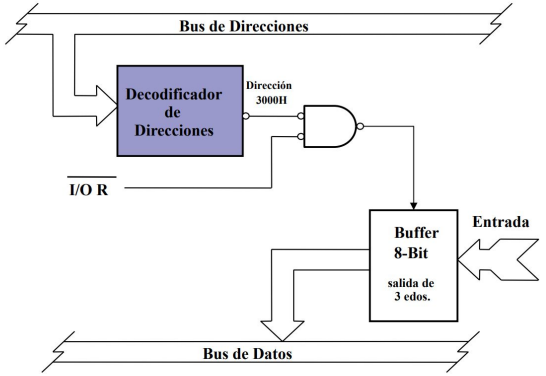
Puertos de Salida



Lenguaje Ensamblador

```
OUT 80h,A    ; sacar el registro acumulador
             ; por el puerto 80h
```

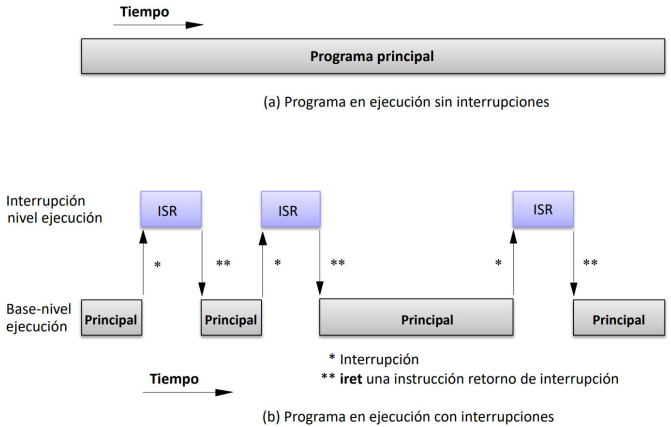
Puertos de Entrada



Lenguaje Ensamblador

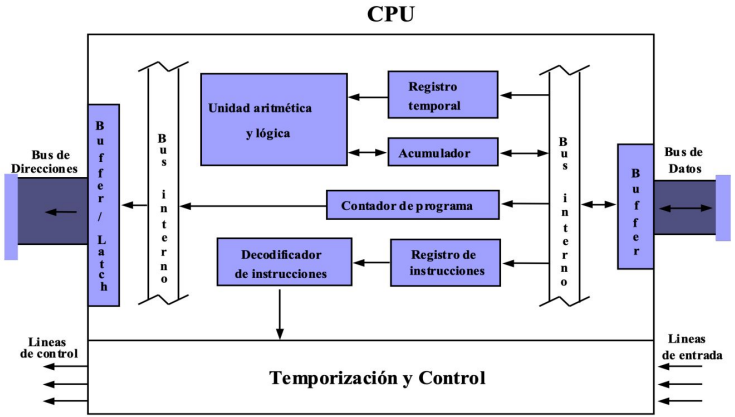
```
IN A,82h     ; leer dato del puerto 82h
             ; y colocarlo en el acumulador
```

Interrupciones



Sección de entrada y salida

Organización simplificada del CPU y la ALU



Funciones principales del CPU de una microcomputadora o Sistema Empotrado

- **Seleccionar, decodificar y ejecutar instrucciones de programa en el orden adecuado.**
- **Transferir datos hacia y desde la memoria, y hacia y desde las secciones de E/S.**
- **Responder a interrupciones externas.**
- **Proporcionar las señales de control y de tiempo necesarias para la totalidad del sistema**

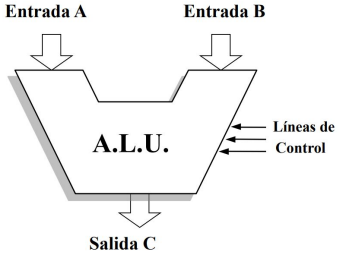
Hardware
Es el nombre que se le da a los dispositivos físicos y circuitos de la computadora.

Software
Se refiere a los programas escritos para la computadora.

Firmware
Es el término que se le da a los programas almacenados permanentemente (programas en ROM o FLASH).

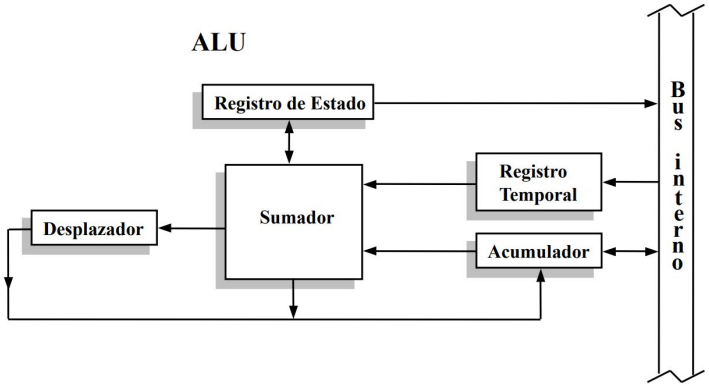
Organización de la ALU

- **Ideal Conceptual**



Operaciones comunes:
Aritméticas: ADD, ADC, SUB, SBB, INC, DEC
Lógicas: AND, OR, NOT, XOR, SHR, SHL

- **Dentro del Procesador**



Unidad 2 - Componentes de una computadora

Sistema operativo

1. Introducción

- Un **Sistema Operativo (SO)** es el software fundamental que administra los recursos del computador.
- Actúa como **intermediario** entre el hardware y los programas de usuario.
- Proporciona una **interfaz amigable** para que los usuarios puedan interactuar con el sistema sin manejar directamente el hardware.

2. Historia breve de los Sistemas Operativos

- **Años 50:** aparecen los primeros SO básicos → procesamiento por **lotes** (batch), con tarjetas perforadas y sin interacción directa.
- **Años 60–70:** surgen la **multiprogramación** y el **tiempo compartido**; nace **UNIX**, base de muchos sistemas modernos.
- **Años 80–90:** popularización de **interfaces gráficas (GUI)**; expansión de **Windows**, **MacOS** y la llegada de **Linux** en 1991.
- **Actualidad:** sistemas operativos orientados a **móviles (Android, iOS)**, **virtualización**, **contenedores** y **computación en la nube**.

3. Funciones principales del Sistema Operativo

- **Gestión de recursos:** administra el uso de la **CPU, memoria, dispositivos de entrada/salida y almacenamiento**.
- **Provisión de abstracciones:** ofrece modelos fáciles de usar, como **procesos**, **archivos** y **memoria virtual**, para simplificar la interacción con el hardware.
- **Seguridad y protección:** controla el acceso de usuarios y programas para evitar conflictos o accesos indebidos.
- **Eficiencia y control:** busca un uso **óptimo, ordenado y confiable** de todos los recursos del sistema.

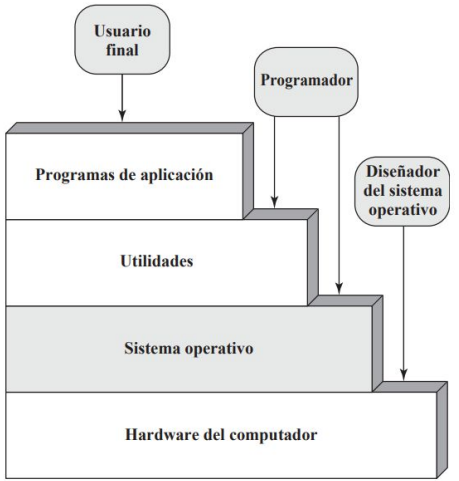


Figura 8.1. Capas y puntos de vista de un computador.

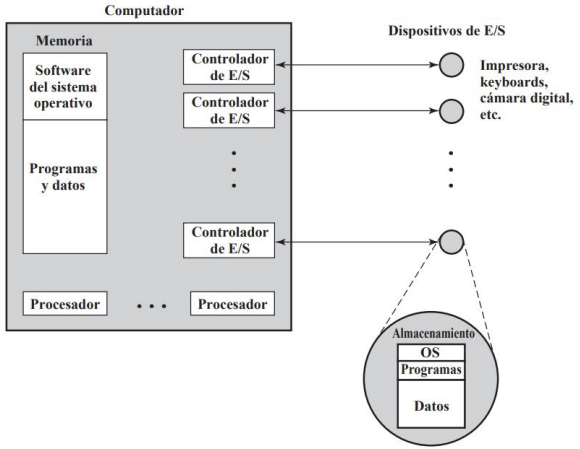


Figura 8.2. El sistema operativo como gestor de recursos.

Unidad 2 - Componentes de una computadora

Sistema operativo

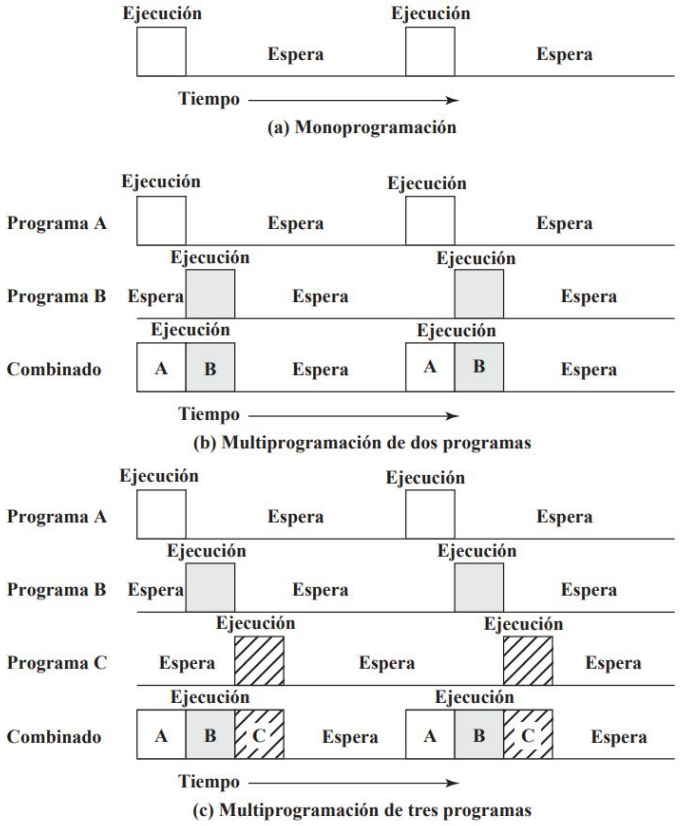


Figura 8.5. Ejemplo de multiprogramación.

4. Tipos de Sistemas Operativos

- **Monotarea vs Multitarea:** un solo programa en ejecución vs varios simultáneamente.
- **Monousuario vs Multiusuario:** orientados a un único usuario o a múltiples usuarios al mismo tiempo.
- **Tiempo compartido:** divide el tiempo de CPU entre procesos para dar la impresión de ejecución simultánea.
- **Tiempo real:** diseñados para responder en plazos estrictos (ej. control industrial, dispositivos médicos).
- **Distribuidos y de red:** permiten que varios computadores trabajen como un sistema integrado o compartan recursos en red.

5. Procesos e Hilos

- **Proceso:** programa en ejecución que incluye código, datos y recursos asignados (CPU, memoria, archivos).
- Cada proceso tiene un **espacio de direcciones propio** y se controla mediante un **PCB (Process Control Block)**.
- **Hilo (thread):** unidad ligera de ejecución dentro de un proceso.
- Los hilos comparten los recursos del proceso, pero cada uno tiene su propia pila y contador de programa.
- Ventaja: permiten **ejecución concurrente** y mejor aprovechamiento de la CPU.

6. Gestión de Memoria

- El SO administra la **memoria principal (RAM)** para que múltiples procesos puedan coexistir de forma ordenada y segura.
- Técnicas principales:
  - **Paginación:** divide la memoria en bloques (páginas y marcos) para una asignación flexible.
  - **Segmentación:** organiza la memoria en segmentos lógicos (código, datos, pila).
- **Memoria virtual:** extiende la RAM usando espacio en disco, permitiendo ejecutar programas más grandes que la memoria física.
- Objetivo: optimizar el uso de memoria y evitar conflictos entre procesos.

Unidad 2 - Componentes de una computadora

Sistema operativo

7. Entrada y Salida (E/S)

- El SO gestiona la comunicación con los **dispositivos externos** mediante **controladores y drivers**.
- Métodos principales de E/S:
  - **Programada (polling):** la CPU consulta continuamente al dispositivo.
  - **Interrupciones:** el dispositivo avisa a la CPU cuando necesita atención.
  - **DMA (Acceso Directo a Memoria):** transfiere datos entre memoria y dispositivo sin sobrecargar la CPU.
- Objetivo: lograr un acceso **rápido y eficiente** a los periféricos.

8. Sistemas de Archivos

- El SO organiza los datos en una estructura jerárquica de **archivos y directorios**.
- Proporciona operaciones básicas: crear, leer, escribir, modificar y eliminar archivos.
- Controla permisos de acceso y seguridad en el manejo de la información.
- Ejemplos de sistemas de archivos:
  - **FAT (File Allocation Table)** → clásico en dispositivos portátiles.
  - **NTFS (New Technology File System)** → usado en Windows.
  - **EXT (Extended File System)** → común en Linux.
  - **APFS (Apple File System)** → usado en macOS e iOS.

9. Seguridad y Protección

- El SO implementa **control de accesos** mediante usuarios, permisos y privilegios para proteger los recursos.
- Garantiza el **aislamiento de procesos** para evitar que un programa afecte a otro.
- Usa técnicas como **sandboxing** para ejecutar aplicaciones en entornos restringidos.
- Objetivo: mantener la **confidencialidad, integridad y disponibilidad** del sistema.