



Arquitectura de computadores I

Perspectiva de alto nivel de los computadores

Septiembre de 2017

Contenido

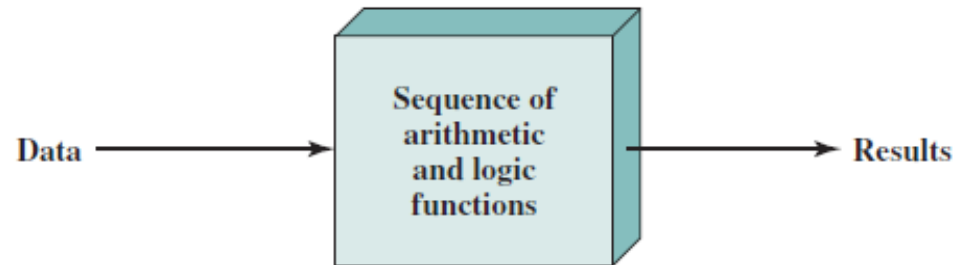
- Componentes del computador
- Funcionamiento del computador
- Estructuras de interconexión
- Interconexión con buses
- Interconexiones punto a punto
- PCI

- 
- Componentes del computador

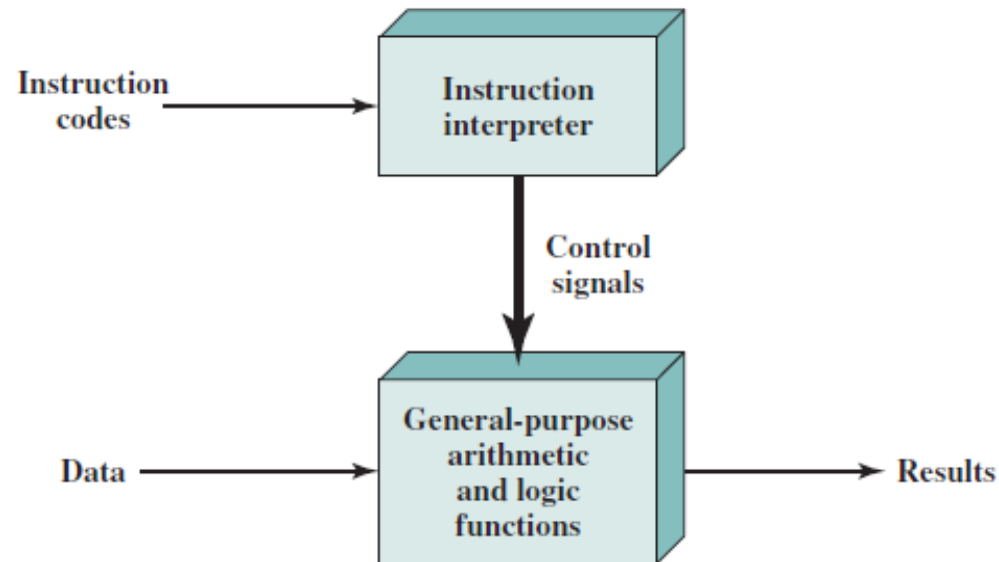
Conceptos generales

- Sistemas de hardware son inflexibles
- Hardware de propósito general puede realizar diferentes tareas, si recibe las señales de control apropiadas
- El cableado interno de control, provee las señales de control

Conceptos generales



(a) Programming in hardware



(b) Programming in software

¿Que es un programa?

- Una secuencia de pasos
- Por cada paso, una operación aritmética o lógica es realizada
- Por cada operación, se requiere un conjunto de señales de control

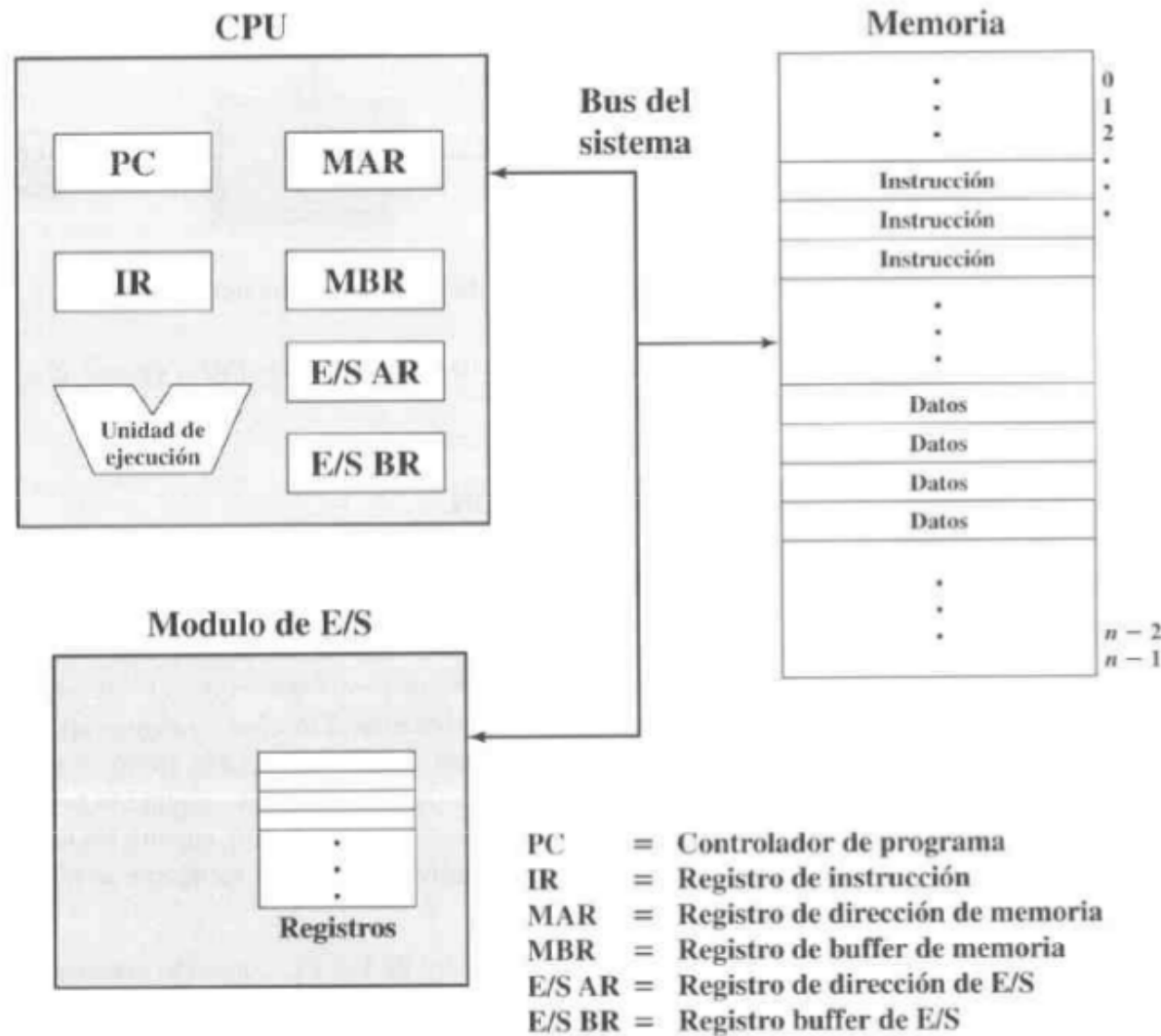
Función de la unidad de control

- Para cada operación un único código es requerido
 - e.g. ADD, MOVE
- Un segmento de hardware acepta el código y se generan las señales de control
- ¡Esto es un computador!

Componentes

- La unidad de control y la unidad lógica constituyen la Unidad Central de Proceso (Central Processing Unit - CPU)
- Datos e instrucciones necesitan ser obtenidos dentro del sistema y sus resultados deben salir del sistema
 - Entradas/Salidas
- Se requiere almacenamiento temporal de código
 - Memoria principal

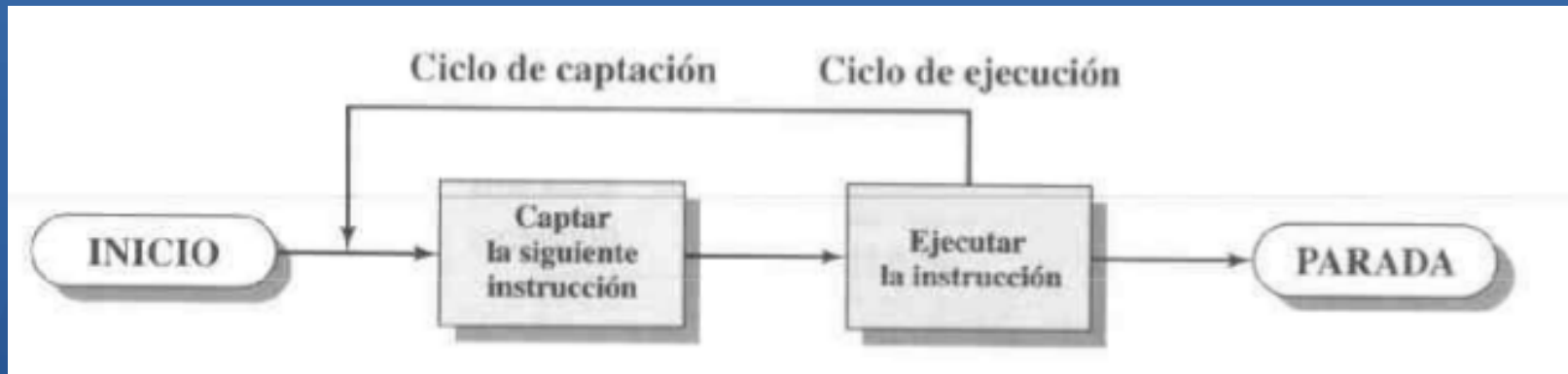
Componentes del computador: Vista alto nivel



- 
- Funcionamiento del Computador

Ciclo de instrucción

- Consta de dos pasos:
 - Captación
 - Ejecución



Ciclo de captación

- El contador de programa (PC) indica la dirección de la siguiente instrucción a captar
- El procesador capta la instrucción desde una localización de memoria indicada por PC
- Incrementa PC
- Instrucción es cargada en el registro de instrucciones (IR)
- El procesador interpreta la instrucción y la ejecuta

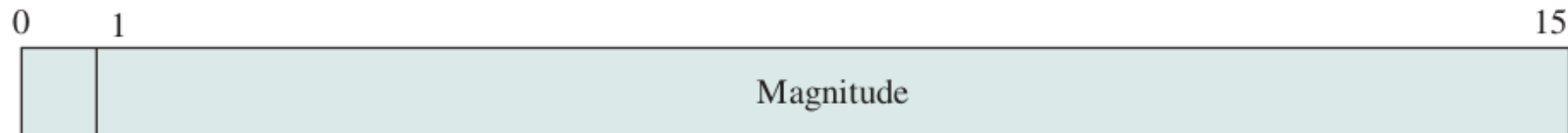
Ciclo de ejecución

- Procesador a memoria
 - Datos transferidos entre la CPU y la memoria principal
- Procesador a E/S
 - Transferencia de datos entre la CPU y el modulo E/S
- Procesamiento de datos
 - Algunas operaciones aritmeticas y lógicas son realizadas
- Control
 - Alteración de la secuencia de operaciones
 - Ejemplo: Salto

Ejemplo de la ejecución de un programa



(a) Instruction format



(b) Integer format

Program counter (PC) = Address of instruction
Instruction register (IR) = Instruction being executed
Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from memory
0010 = Store AC to memory
0101 = Add to AC from memory

(d) Partial list of opcodes

Ejemplo de la ejecución de un programa

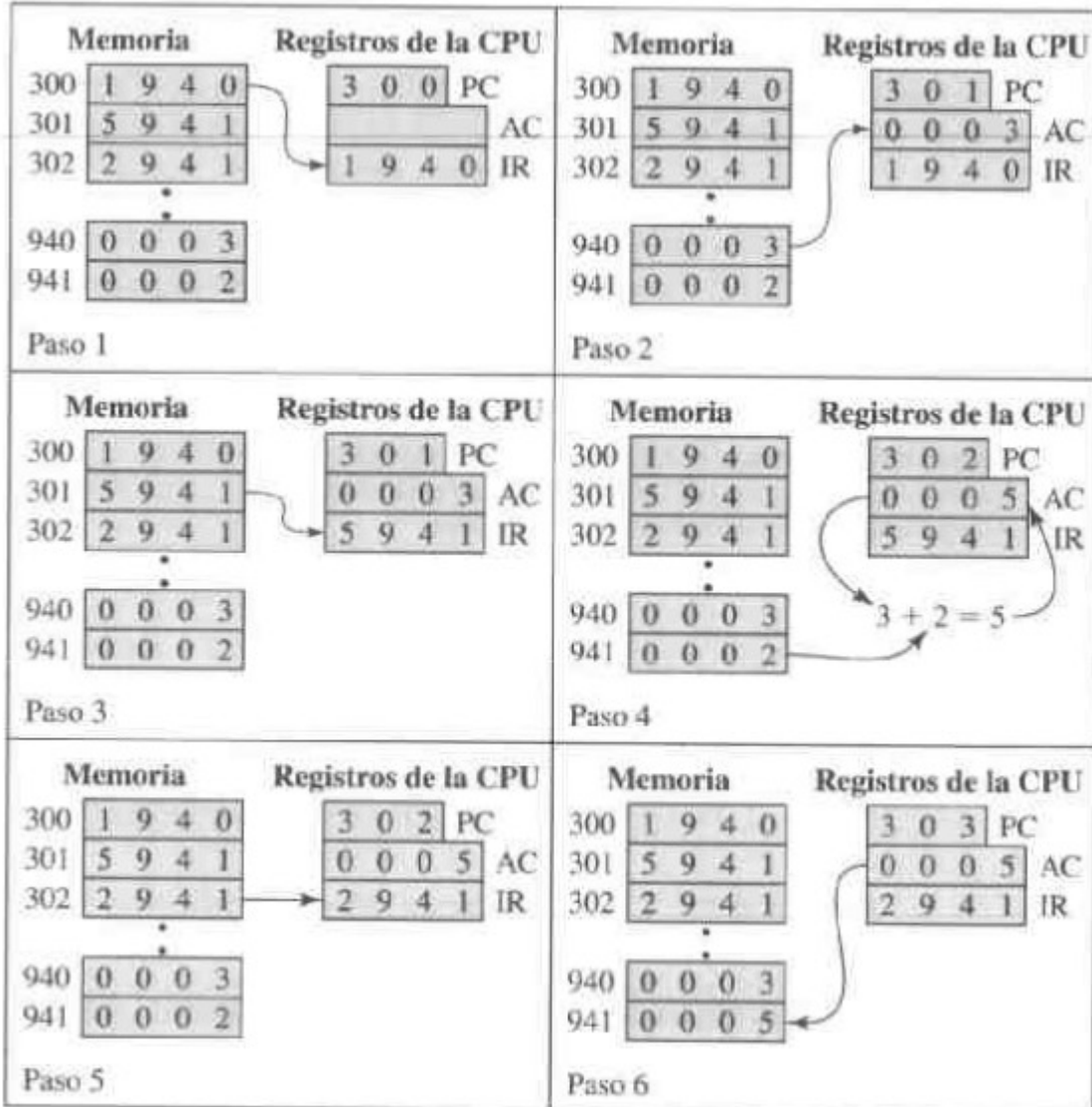
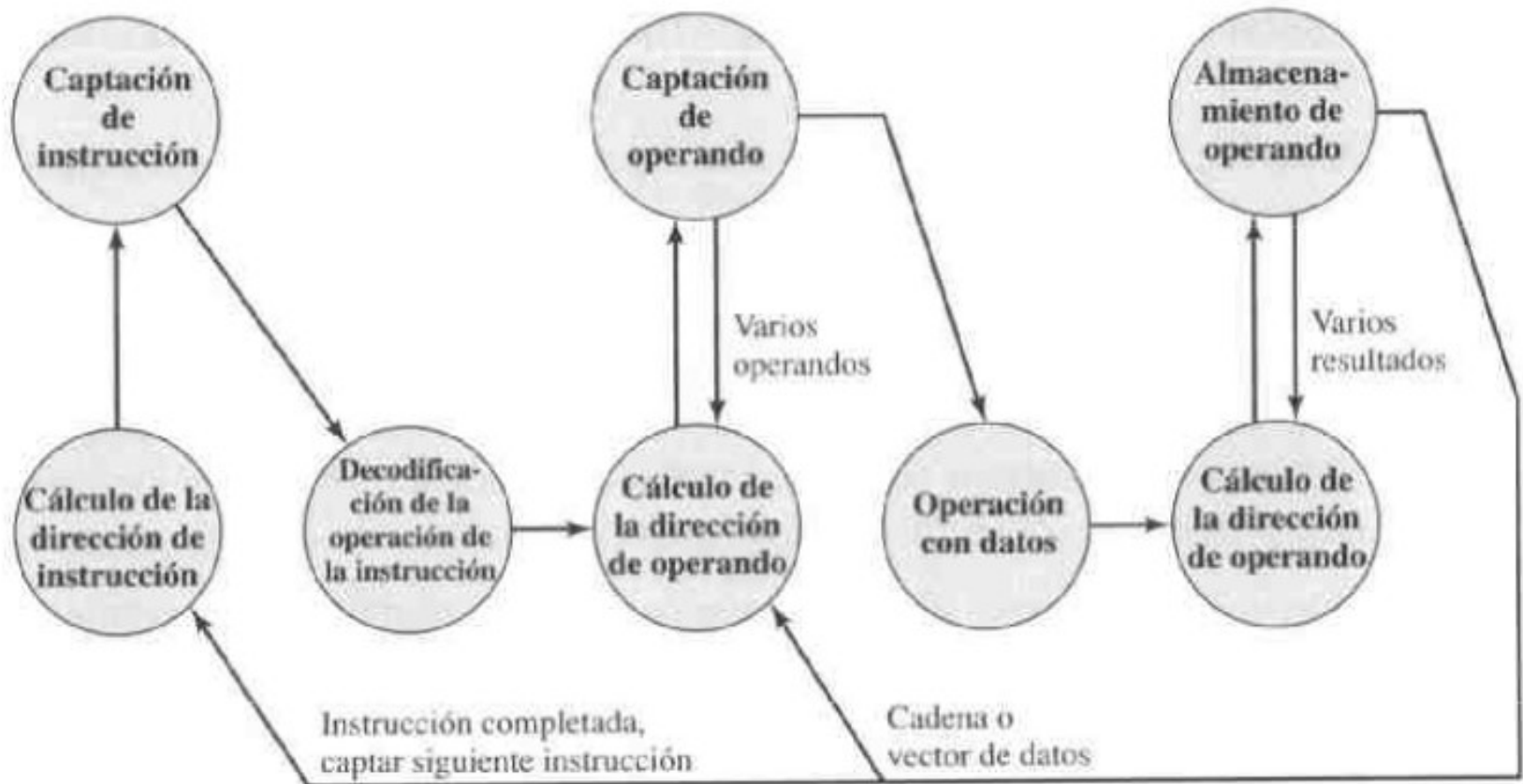


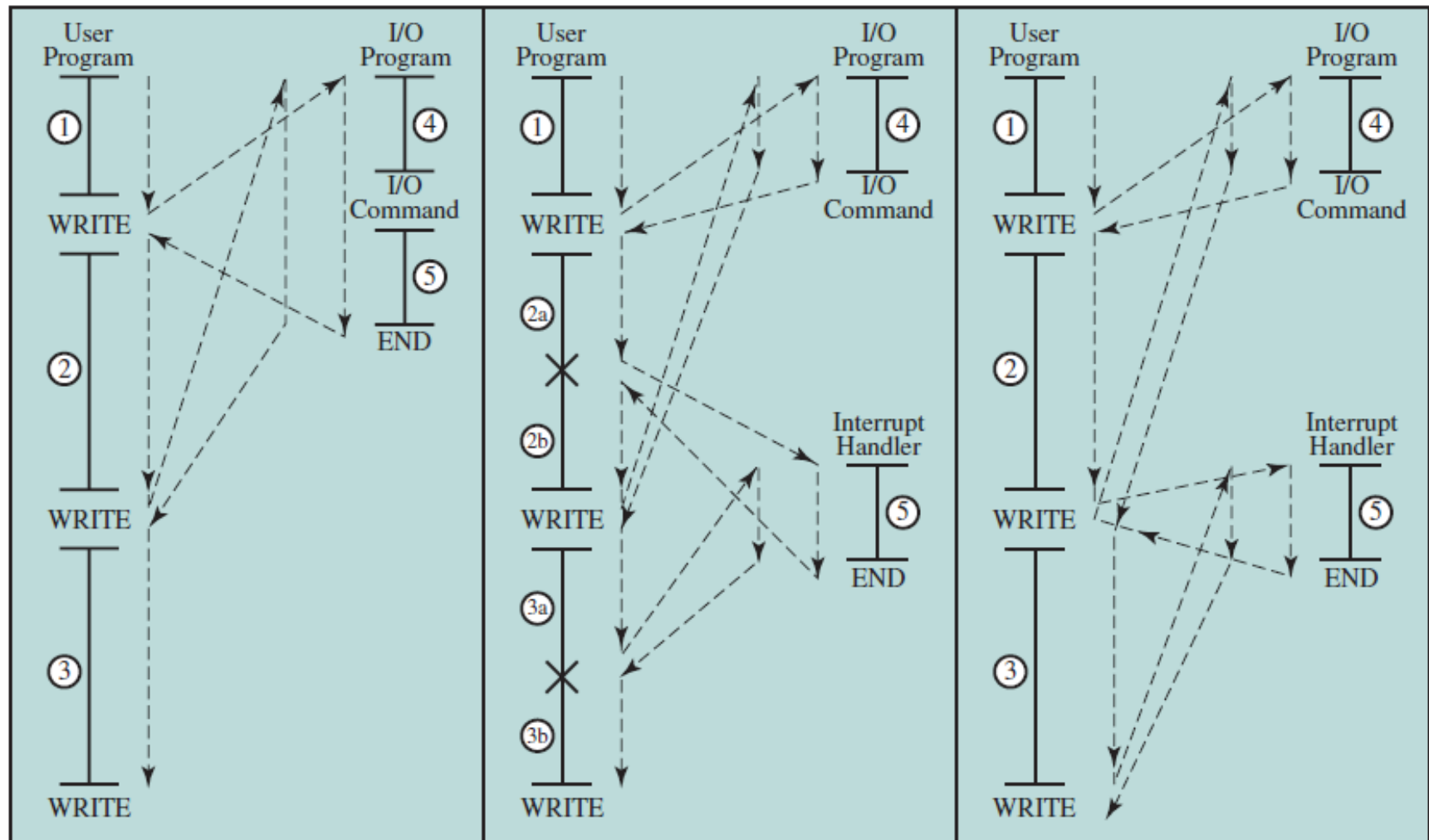
Diagrama de ejecución de una instrucción



Interrupciones

- Mecanismo para que otros módulos (ejemplo E/S) puedan ejecutar una interrupción normal de un proceso
- Programa
 - Desbordamiento, división por cero
- Tiempo
 - Generado por el procesador de tiempo interno
 - Usado en multitarea
- E/S
 - Desde un controlador E/S

Control de flujo en un programa



(a) No interrupts

(b) Interrupts; short I/O wait

(c) Interrupts; long I/O wait

Control de flujo en un programa

USER PROGRAM

$\left. \begin{array}{l} \langle \text{statement} \rangle \\ \langle \text{statement} \rangle \\ \vdots \\ \langle \text{statement} \rangle \end{array} \right\}$ Code segment 1

WRITE

$\left. \begin{array}{l} \langle \text{statement} \rangle \\ \langle \text{statement} \rangle \\ \vdots \\ \langle \text{statement} \rangle \end{array} \right\}$ Code segment 2

WRITE

$\left. \begin{array}{l} \langle \text{statement} \rangle \\ \langle \text{statement} \rangle \\ \vdots \\ \langle \text{statement} \rangle \end{array} \right\}$ Code segment 3

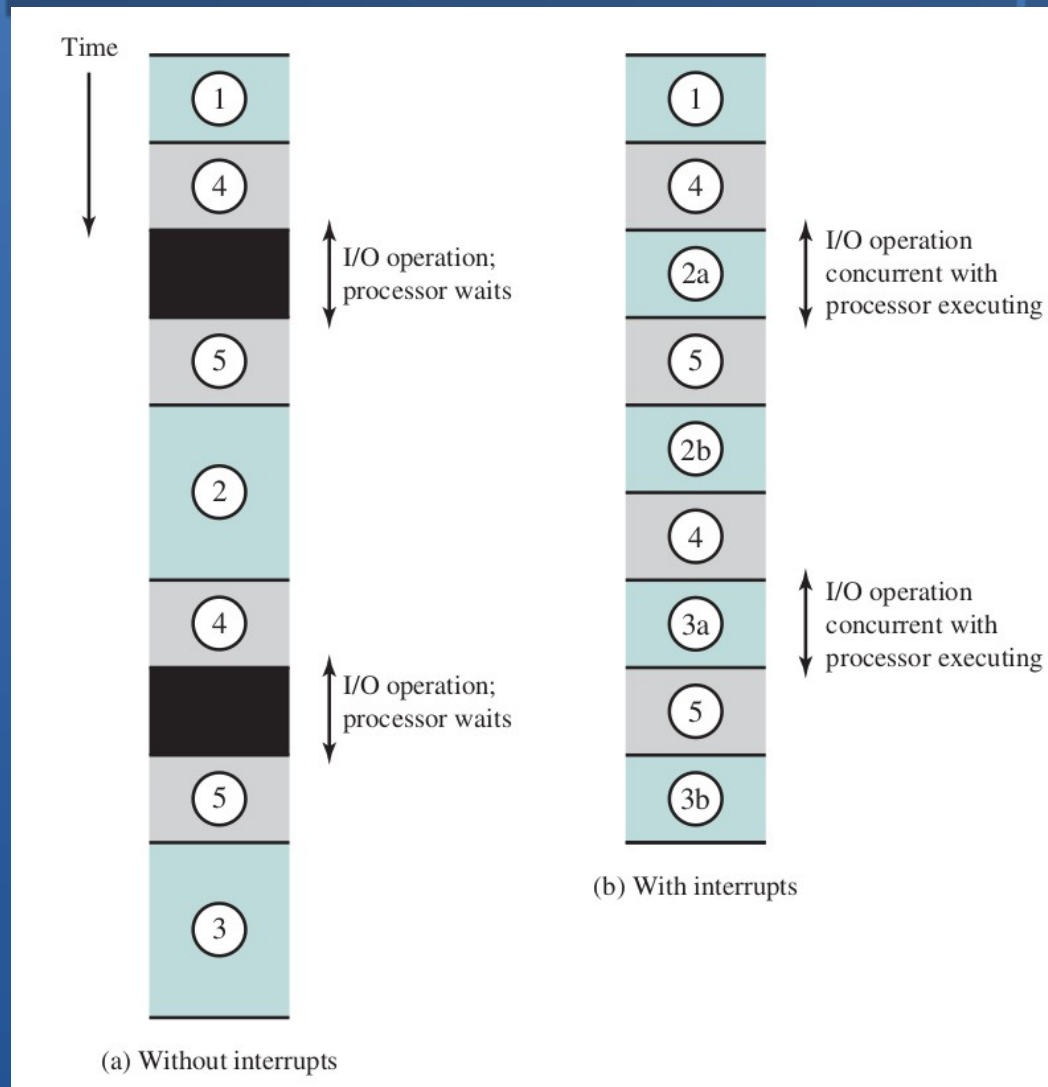
I/O PROGRAM

$\left. \begin{array}{l} \langle \text{statement} \rangle \\ \langle \text{statement} \rangle \\ \vdots \\ \langle \text{statement} \rangle \end{array} \right\}$ Code segment 4

I/O command

$\left. \begin{array}{l} \langle \text{statement} \rangle \\ \langle \text{statement} \rangle \\ \vdots \\ \langle \text{statement} \rangle \end{array} \right\}$ Code segment 5

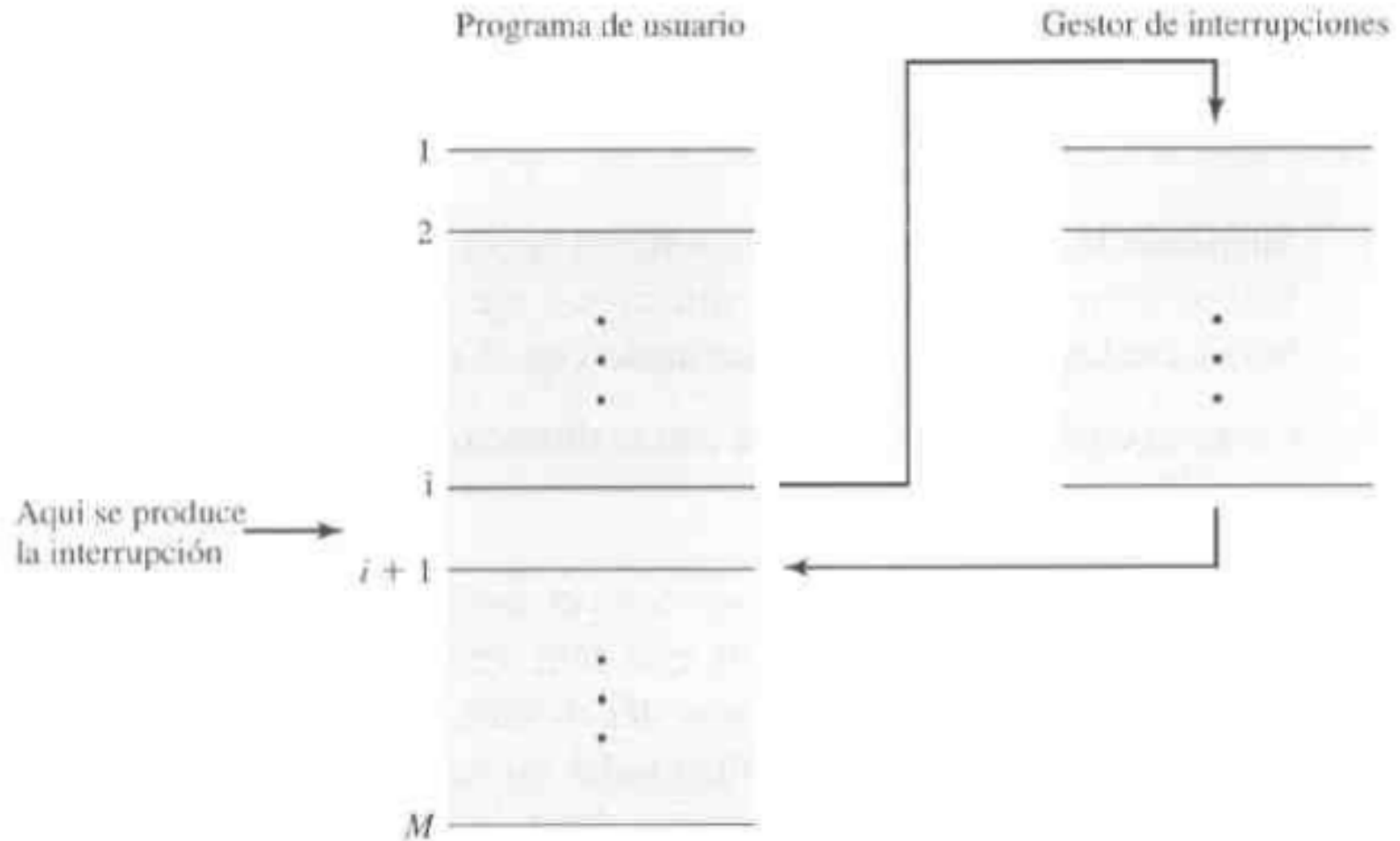
Control de flujo en un programa



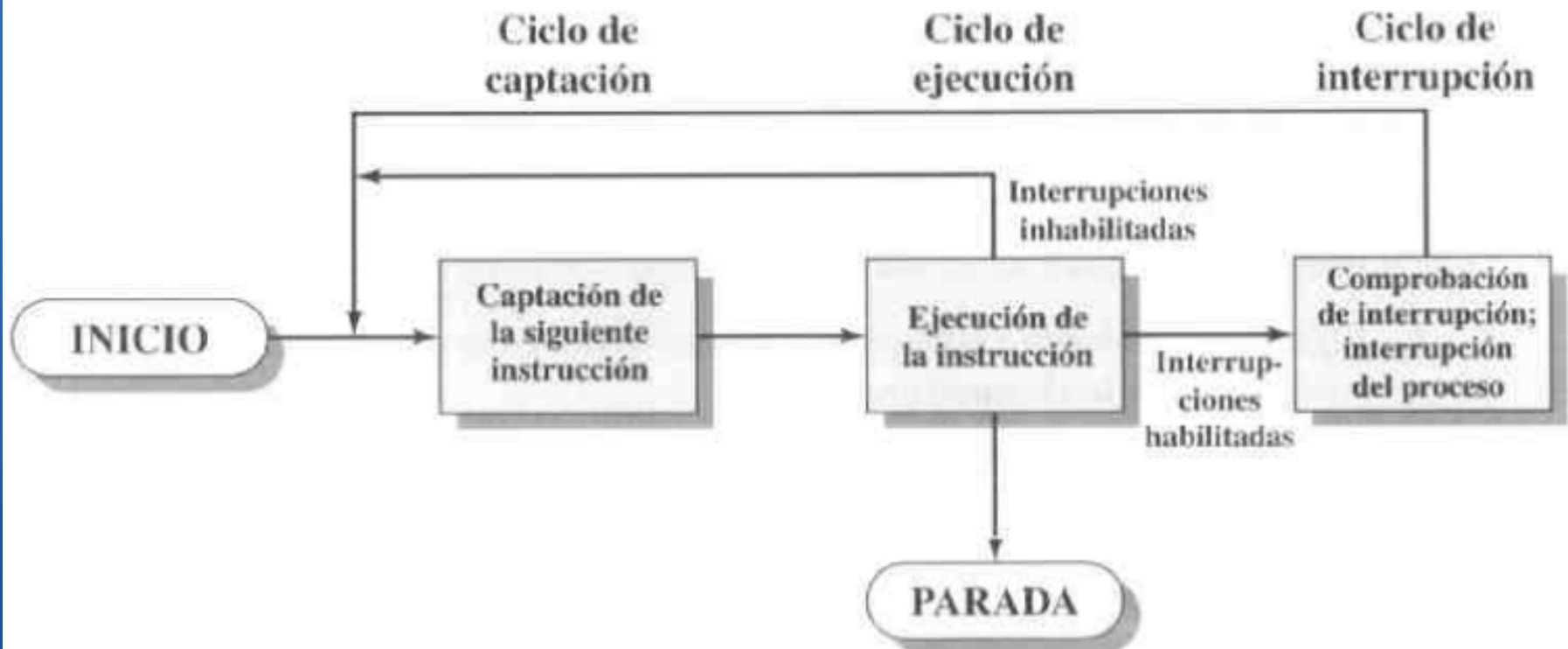
Ciclo de interrupción

- Añadido a la instrucción
- Procesador busca interrupciones
 - Indicada por una señal de interrupción
- Si no hay interrupción, capta la siguiente instrucción
- Si hay una interrupción pendiente
 - Suspende la ejecución del programa actual
 - Guarda contexto
 - Establece PC en la dirección donde inicial la rutina de interrupción
 - Interrumpe el proceso
 - Restaura contexto y continua el programa interrumpido

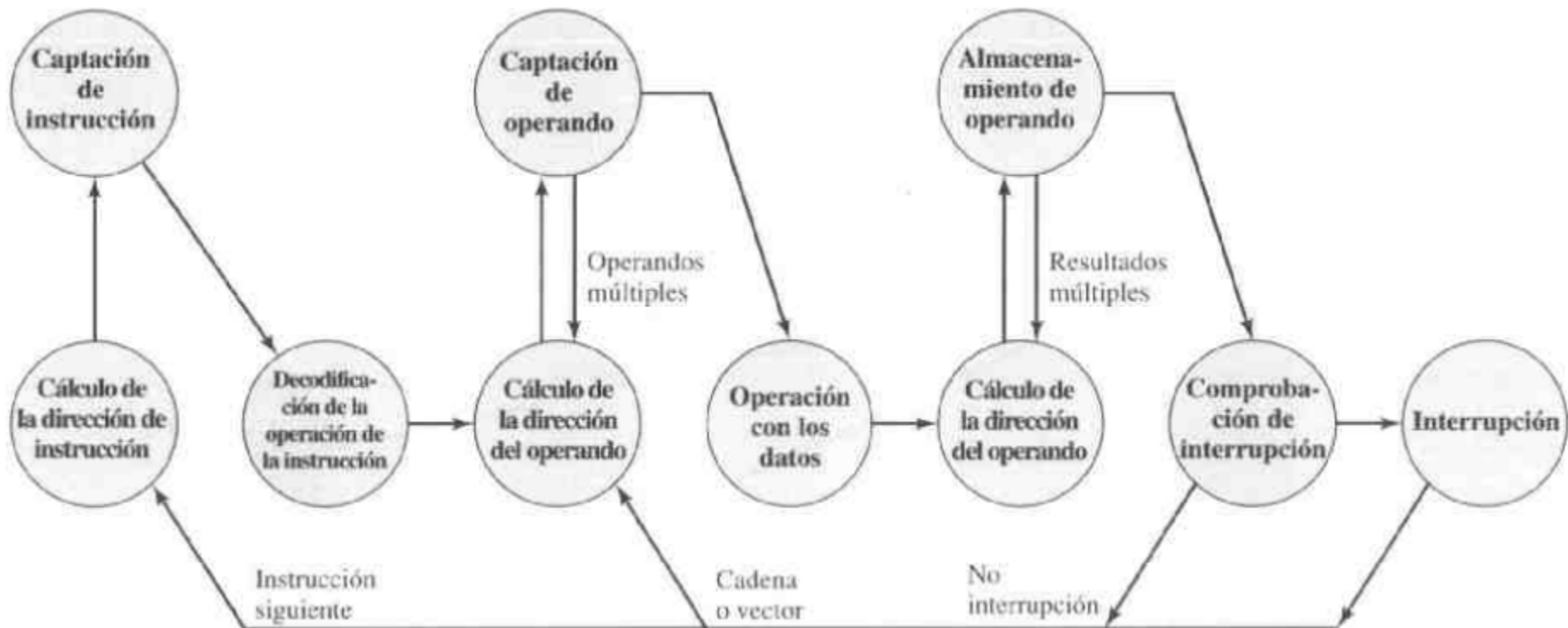
Transferencia de control vía interrupción



Ciclo de instrucción con interrupciones



Ciclo de instrucción con interrupciones



Múltiples interrupciones

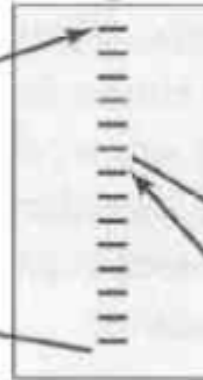
- Apagar interrupciones
 - El procesador ignora otras interrupciones mientras esté procesando una
 - Estas quedan una lista de interrupciones pendientes
- Definir prioridad
 - Interrupciones de baja prioridad pueden ser interrumpidas por interrupciones de alta prioridad
 - Se una interrupción detiene el proceso de una interrupción de menor prioridad, esta continua después de la interrupción de mayor prioridad

Múltiples interrupciones

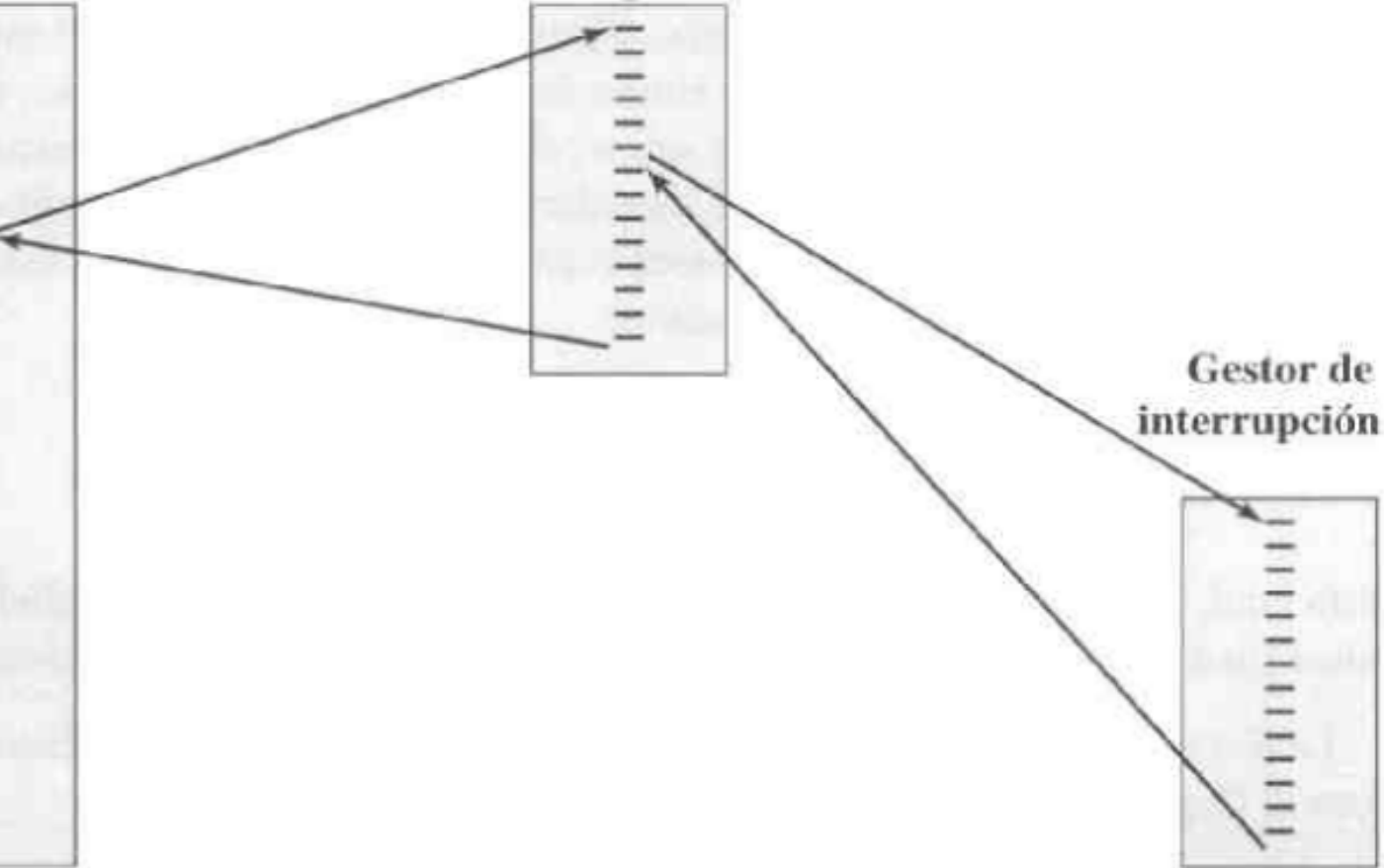
Programa de usuario



Gestor de interrupción X



Gestor de interrupción Y

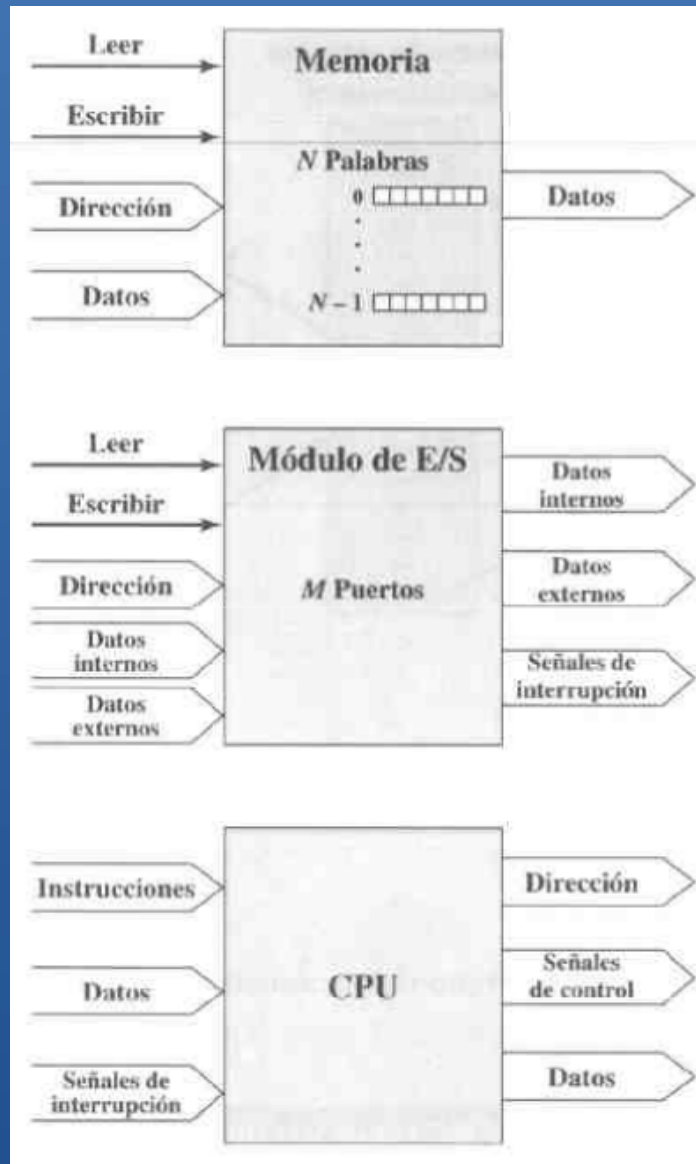


- 
- Estructuras de interconexión

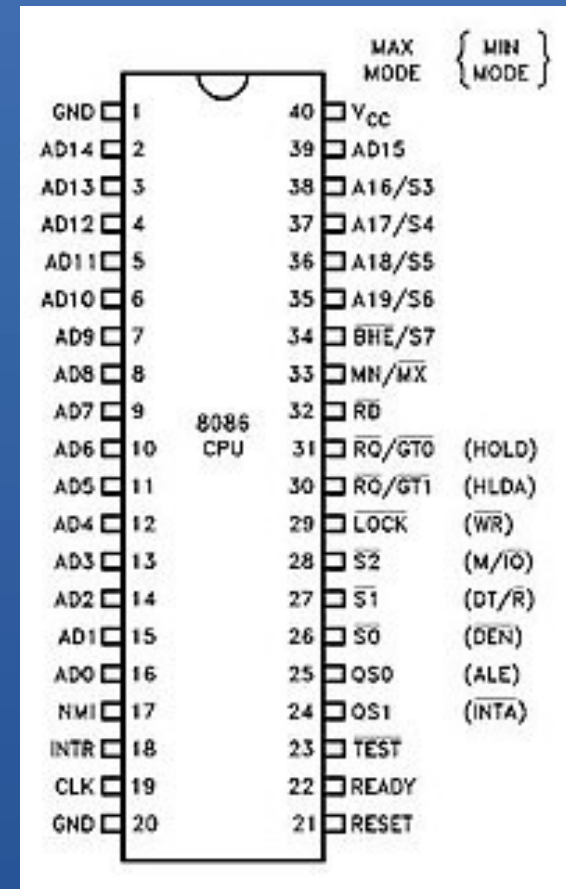
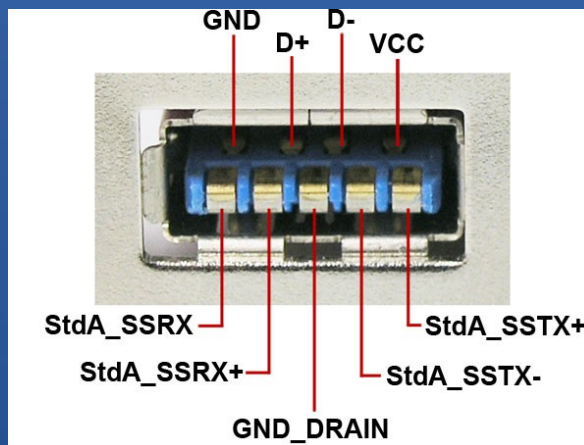
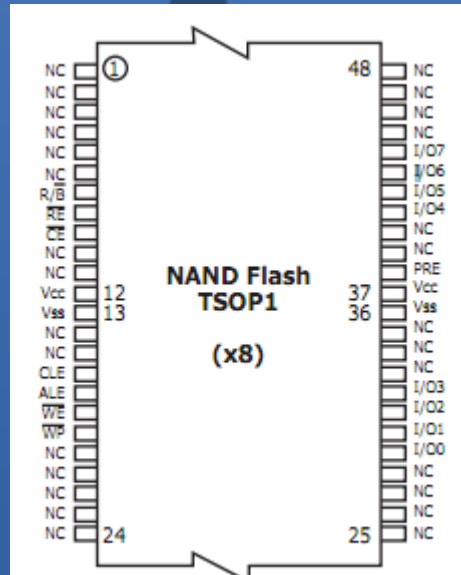
Conexiones

- Todas las unidades del computador deben estar conectadas
- Diferentes tipos de conexiones para diferentes tipos de unidades
 - Memoria
 - Entrada/Salida
 - CPU

Módulos del computador



Modulos del computador



Conexión de la memoria

- Recibe y envía datos
- Recibe direcciones (localizaciones)
- Recibe señales de control
 - Leer
 - Escribir

Conexiones de dispositivos E/S

- Similar a las conexiones de memoria
- Salida
 - Recibe datos desde el computador
 - Envía datos a un periférico
- Entrada
 - Envía datos al computador

Conexiones de la CPU

- Leen instrucciones y datos
- Escriben datos de salida
- Envía señales de control a otras unidades
- Recibe interrupciones

- 
- Interconexión con buses

Buses

- Hay un gran número de sistemas interconectados
- Las estructuras simples y múltiples de buses son muy comunes
- Ejemplo: Buses de control, de direccionamiento, de datos, etc

¿Que es un bus?

- Es una vía de comunicación que conecta dos o más dispositivos
- Es de comunicación de doble vía
- Usualmente están agrupados
 - Un número de canales (caminos) en un bus
 - Ejemplo: un bus de 32 bits, tiene 32 canales de 1 bit

Bus de datos

- Acarrea datos
 - No hay diferencia entre datos e instrucciones en este nivel
- El ancho del bus es clave determinante del rendimiento
 - 8, 16, 32, 64 bit

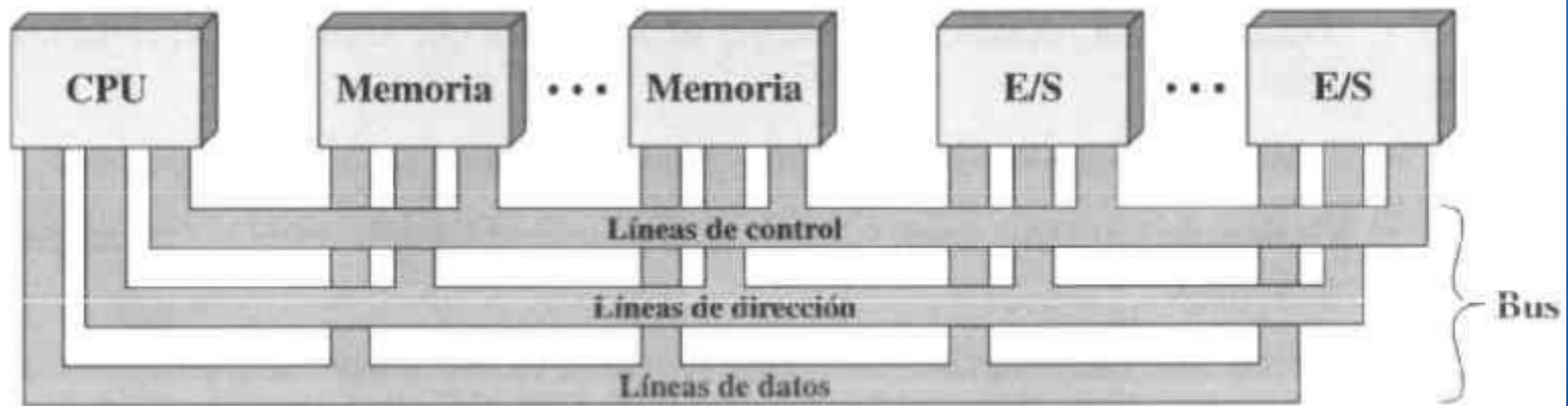
Bus de direcciones

- Identifica la fuente y el destino de los datos
- Ejemplo: La CPU lee una instrucción en memoria que es localizada por una dirección
- El ancho del bus determina la máxima capacidad de memoria del sistema
 - Ejemplo un bus de 16 bit permite manejar un sistema de 64K de memoria

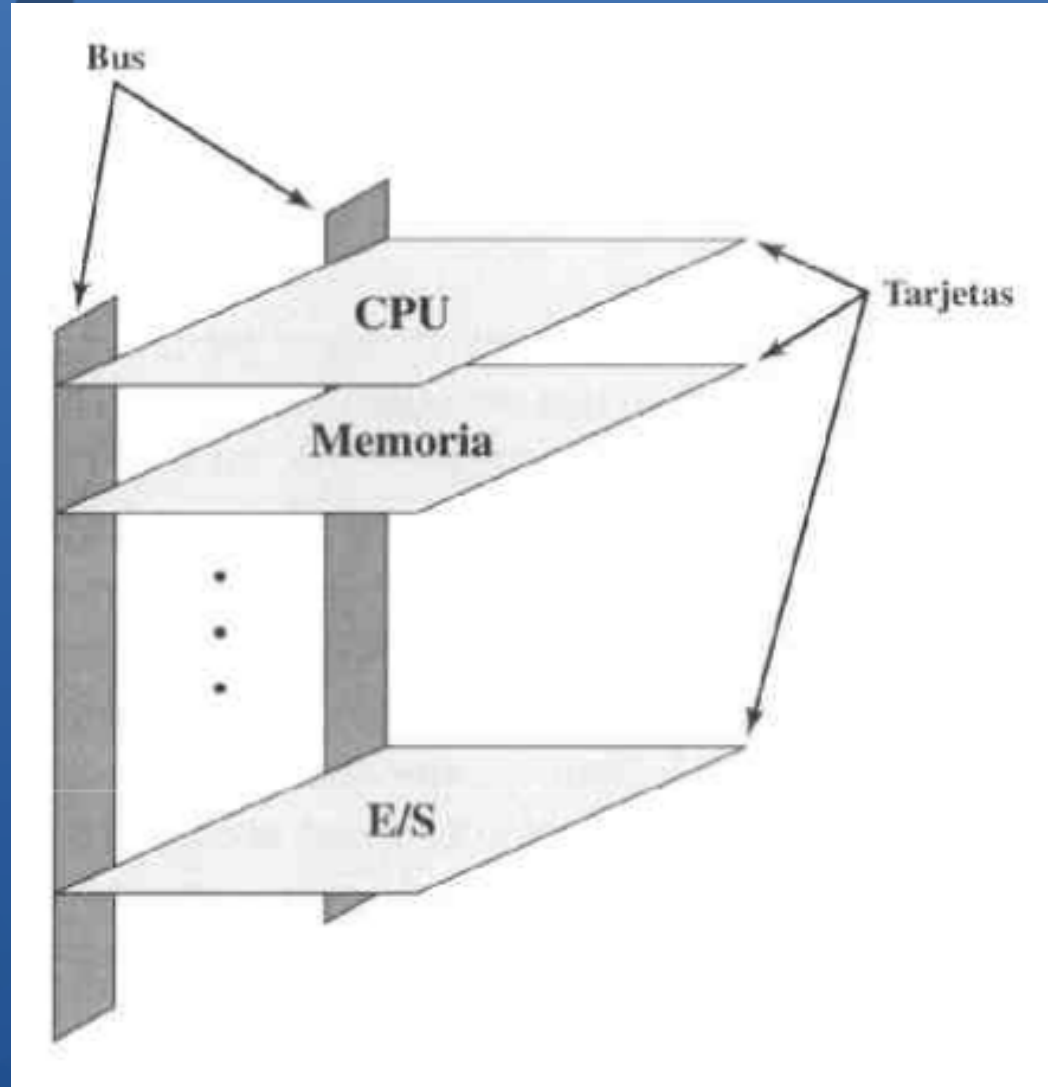
Bus de control

- Controla el flujo de la información
 - Señales de lectura/escritura de memoria
 - Requerimiento de interrupciones
 - Señales de reloj

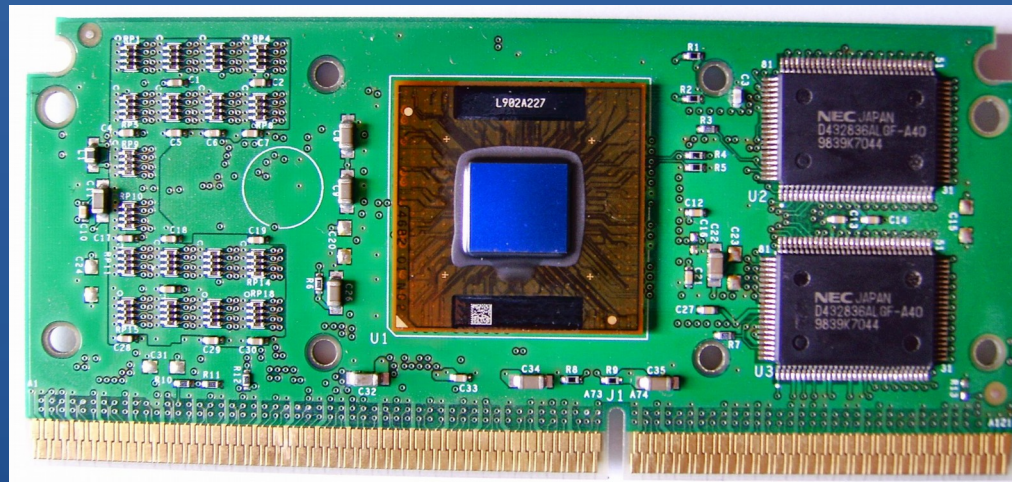
Esquema de interconexión de un bus



Realización física de una arquitectura de bus



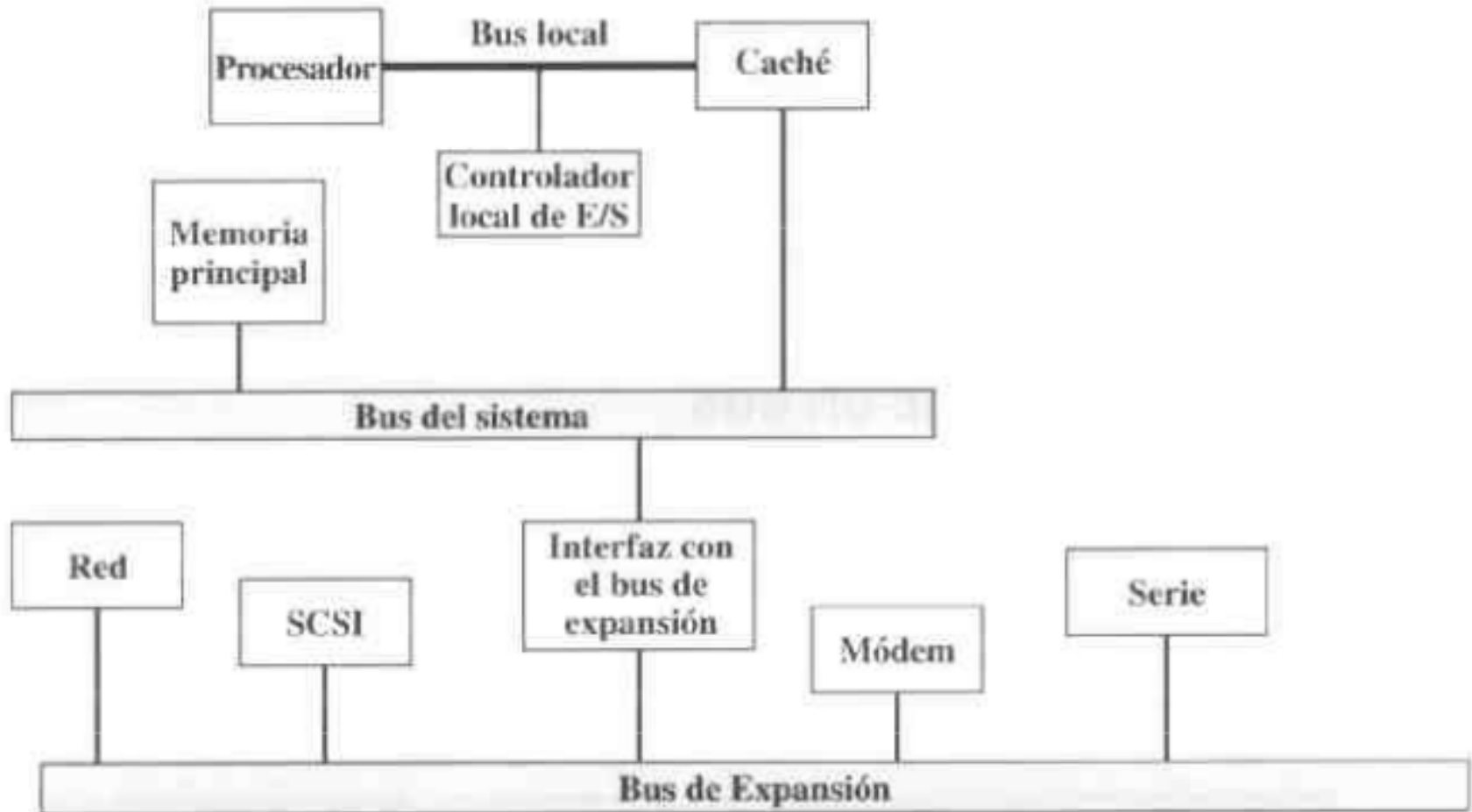
Realización física de una arquitectura de bus



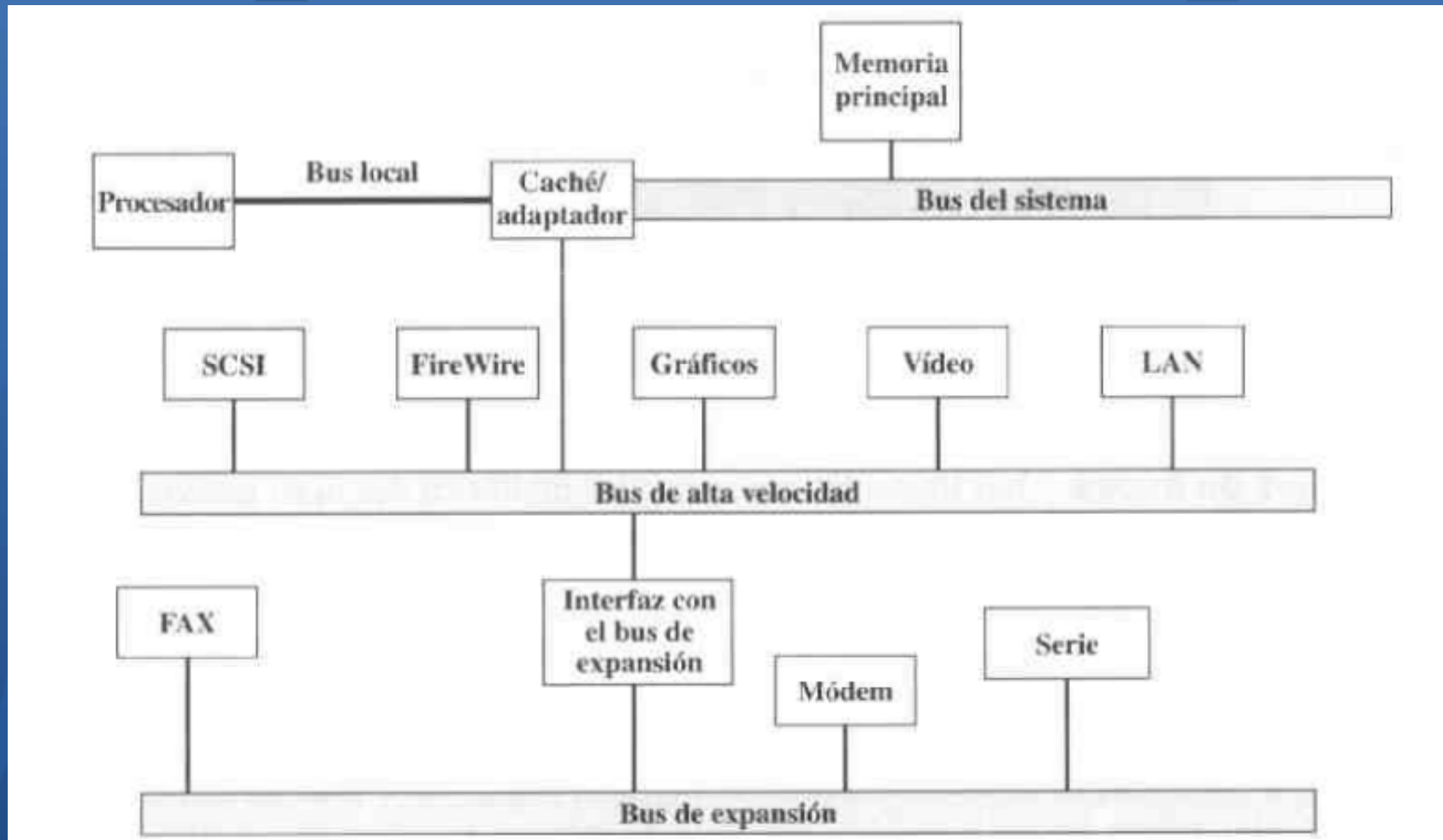
Problemas de buses simples

- Presentan problemas de retrasos en la programación
- Muchos sistemas utilizan buses múltiples para solucionar estos problemas

Modelo tradicional (ISA) de Bus



Configuración de alto rendimiento de Bus



Tipos de buses

- Dedicados
 - Separan datos y direcciones
- Multiplexados
 - Lineas compartidas
 - Linea de control indicando si es un dato o una dirección
 - Ventajas: pocas lineas
 - Desventajas
 - Control complejo

Arbitraje de buses

- Más de un modulo puede controlar un bus
- Ejemplo CPU y Memoria
- Sólo un modulo puede controlar al tiempo al bus
- El arbitraje puede ser centralizado o distribuido

Arbitraje de buses

- Centralizado

- Un hardware controla el bus
 - Controlador de bus
 - Arbitro
- Puede ser parte de la CPU o no

- Distribuido

Cada modulo reclama el control del bus

- Existe una lógica de control en todos los modulos

Control de tiempo

- Coordinación de los eventos del bus
- Sincrono
 - Los eventos son determinados por señales de reloj
 - Todos los dispositivos pueden leer el reloj

Diagrama de tiempo sincrono

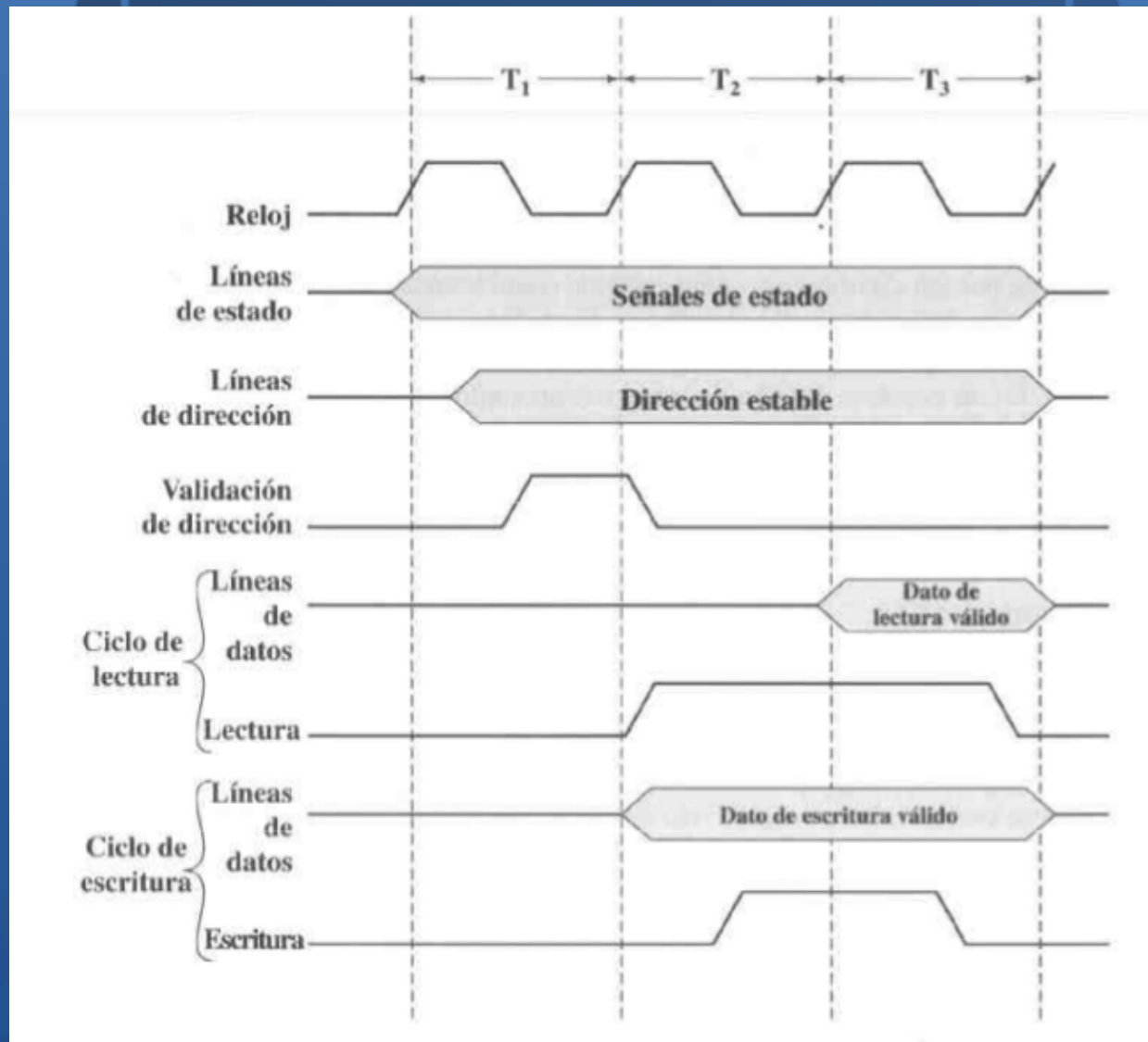


Diagrama de tiempo asincrono

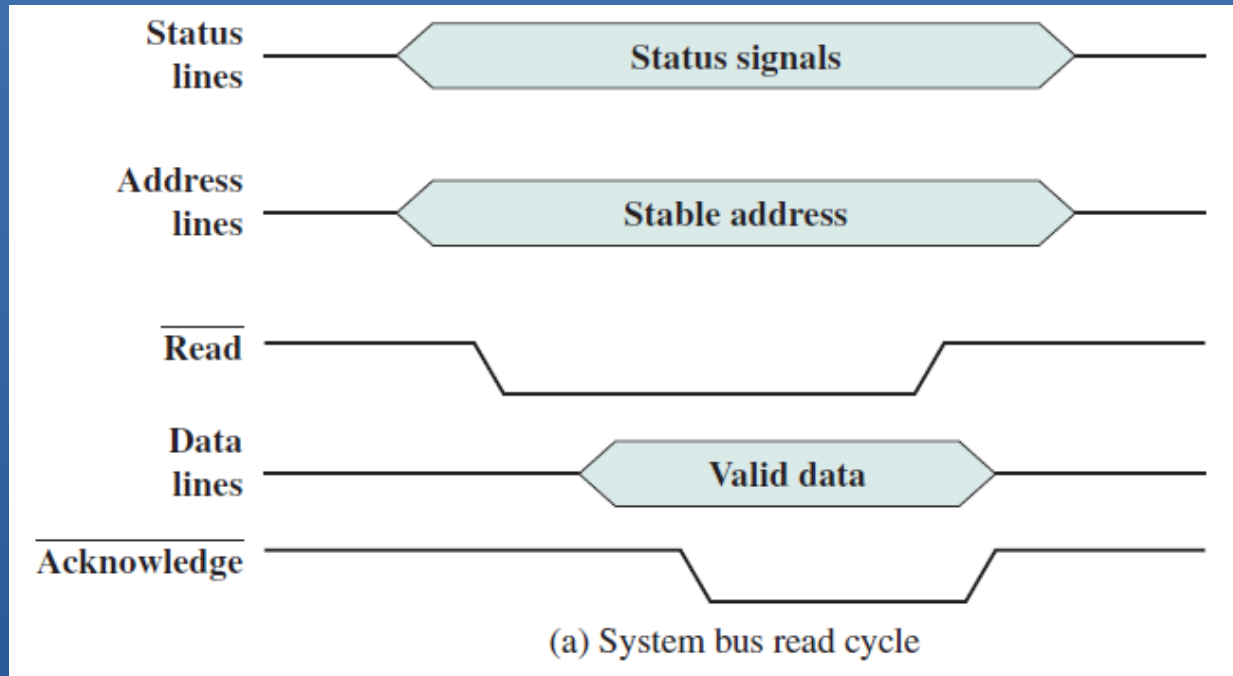
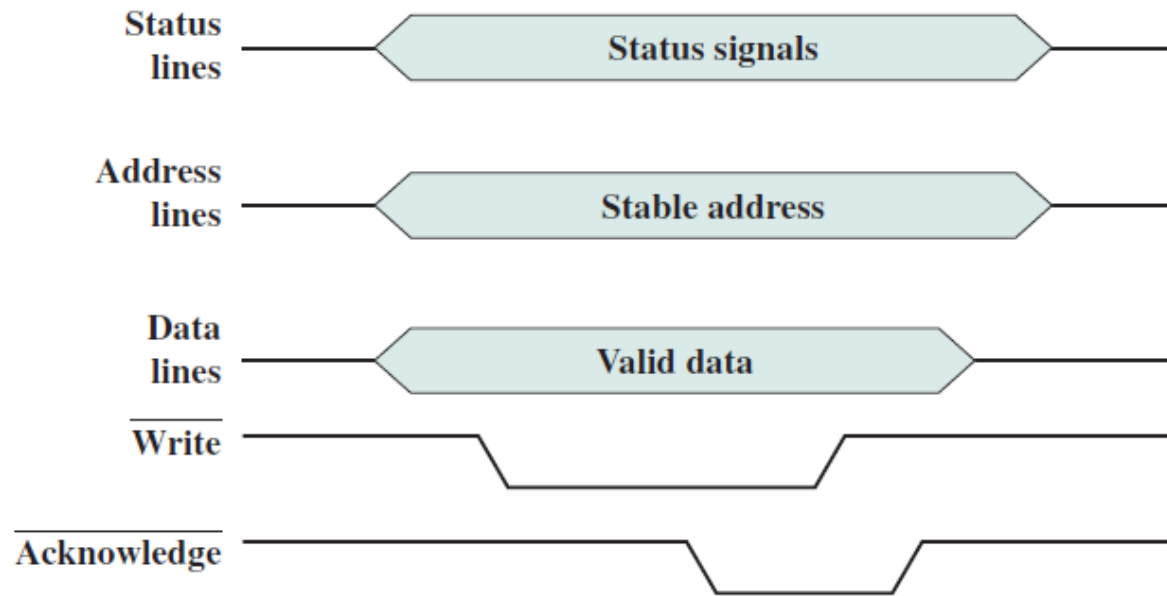


Diagrama de tiempo asincrono



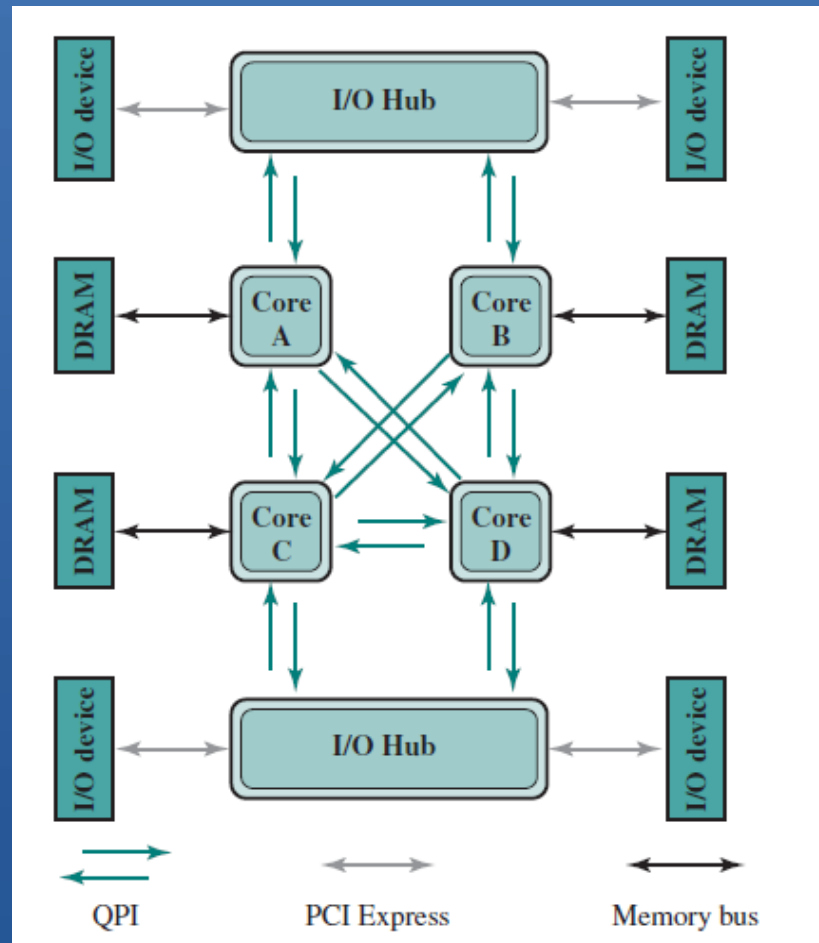
(b) System bus write cycle

- 
- Interconexión con buses

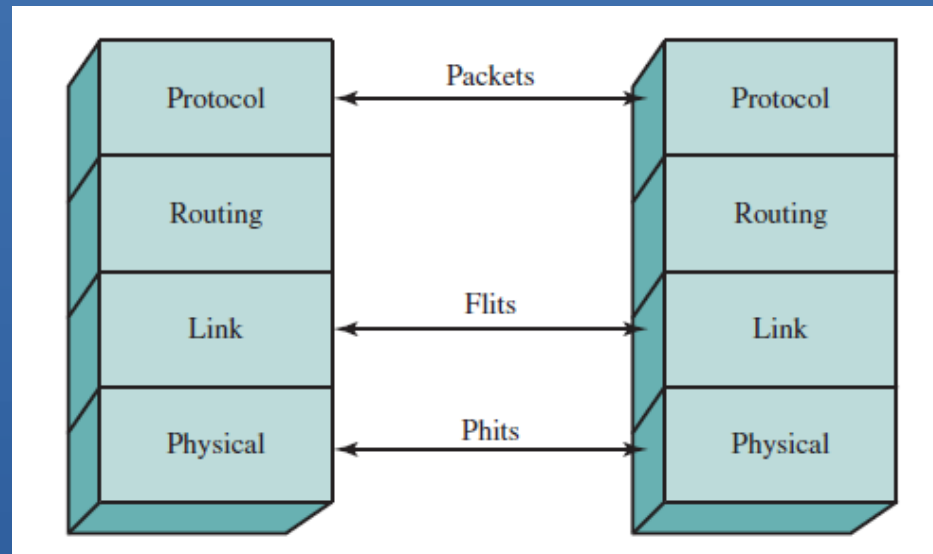
Interconexión

- Introducido por Intel en el 2008 (QPI)
- Múltiples Conexiones directas
- Inspirado en protocolos por capas (Redes)
- Datos transferidos por paquetes

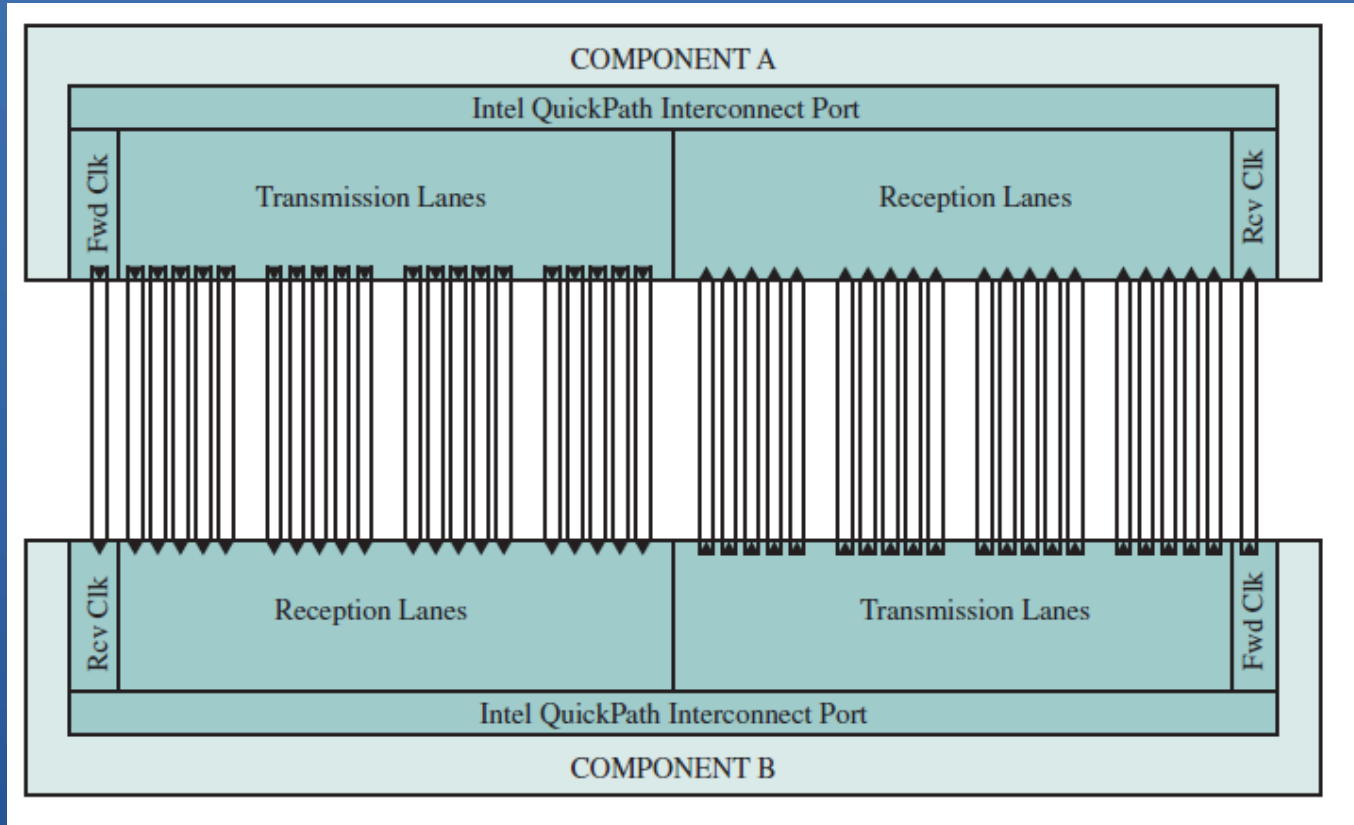
Interconexión



Interconexión



Interconexión

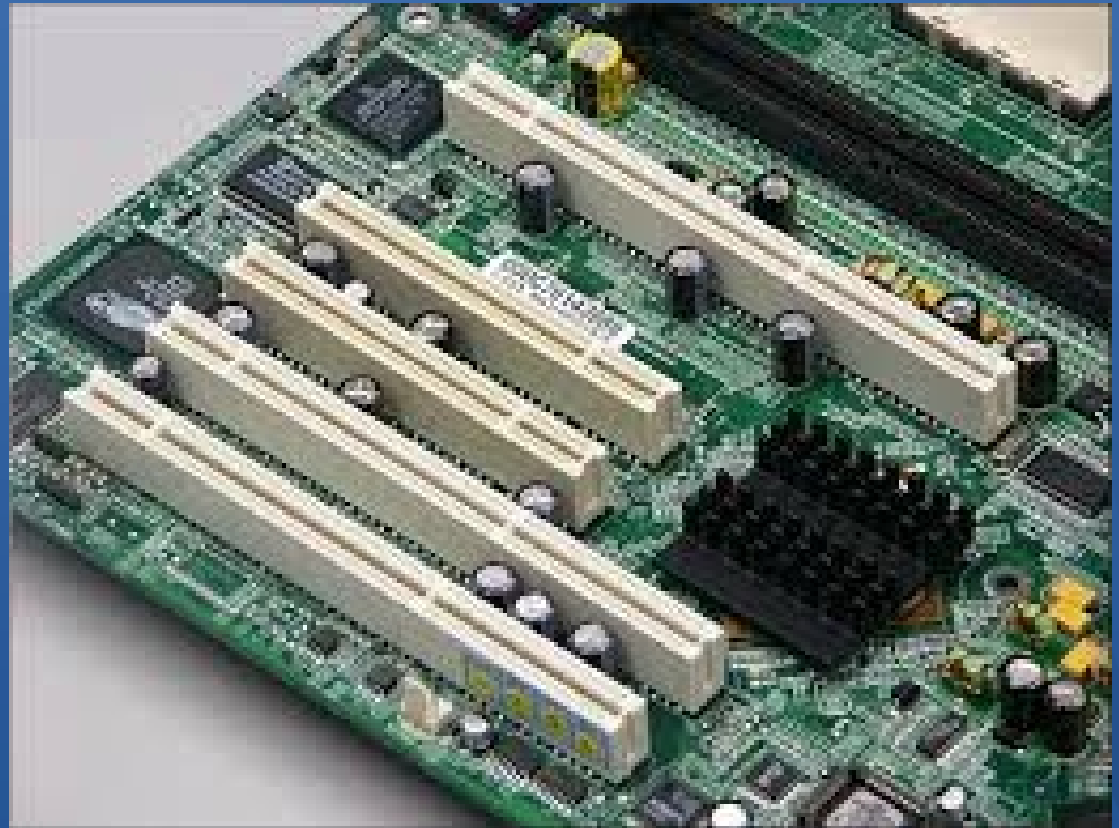




PCI Express

Bus PCI

- Conexiones de componentes perifericos
- 32 o 64 bit
- 50 lineas



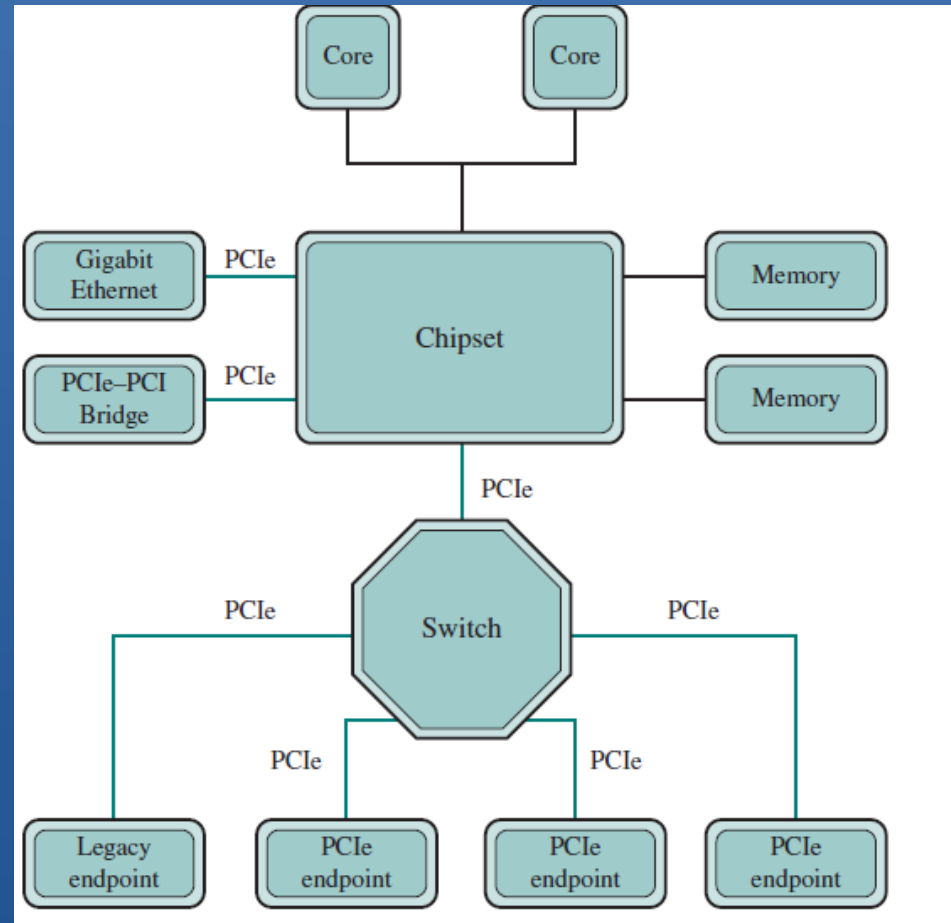
Bus PCI

Velocidades:

Versión	Velocidad
V1	250MB/s (por puerto) 4 GB/s (Total)
V2	500MB/s (por puerto) 8GB/s (Total)
V3	985MB/s (por puerto) 15.75 GB/s (Total)
V4	1969 MB/s (por puerto) 31.51 GB/s (Total)

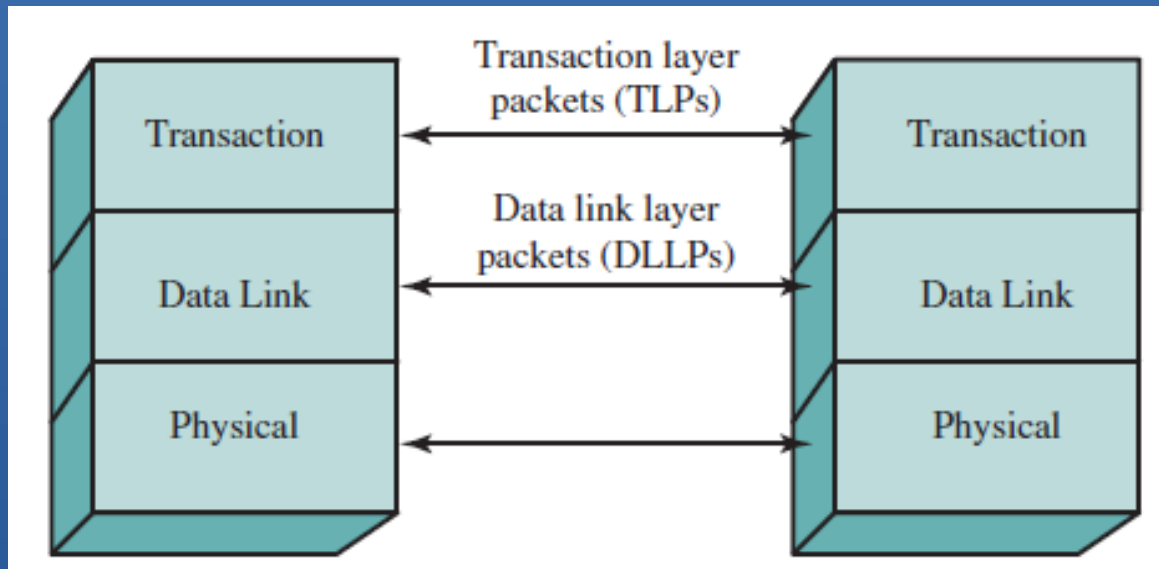
Bus PCI

- Uso típico



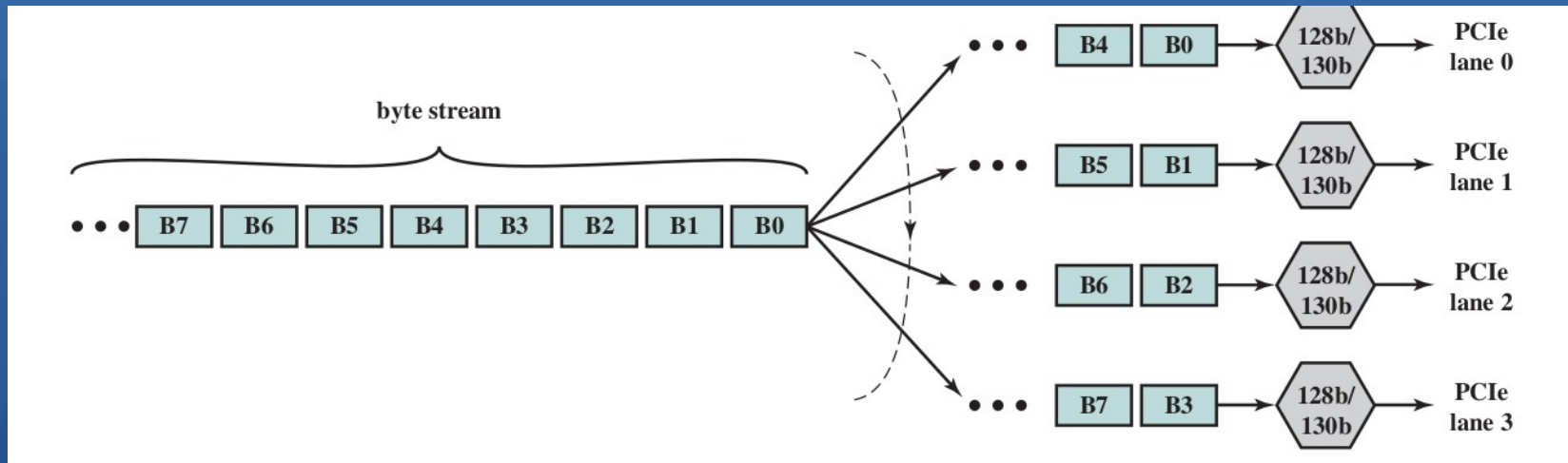
Bus PCI

- Protocollo PCIe



Capa física PCIe

- Similar a QPI (Conexión punto a punto)
- Cada puerto provee 1,4,6,16 o 32 líneas
- Trabaja con 128 bits al tiempo



Lineas de bus PCI

- Lineas del sistema
 - Incluyen reloj y reset
- Direcciones y datos
 - 32 lineas multiplexadas
 - Lineas de interrupciones
- Interface de control
- Lineas de error

Referencias

William Stallings. 2013. Computer Organization and Architecture: Designing for Performance (9th Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA. Chapter 3

Gracias

¿Preguntas?

Próxima clase: Memoria cache