

# *Introducción a la Computación*

---

## *Capítulo 7*

### *Entrada/Salida*

## *Problemas Entrada/Salida*

---

- Existe una amplia variedad periféricos
  - Se les entrega distintas cantidades de datos
  - Funcionan a distintas velocidades
  - Tienen diferentes formatos
- Todos son mas lentos que la CPU y la RAM
- Son necesarios los **Módulos de E/S**

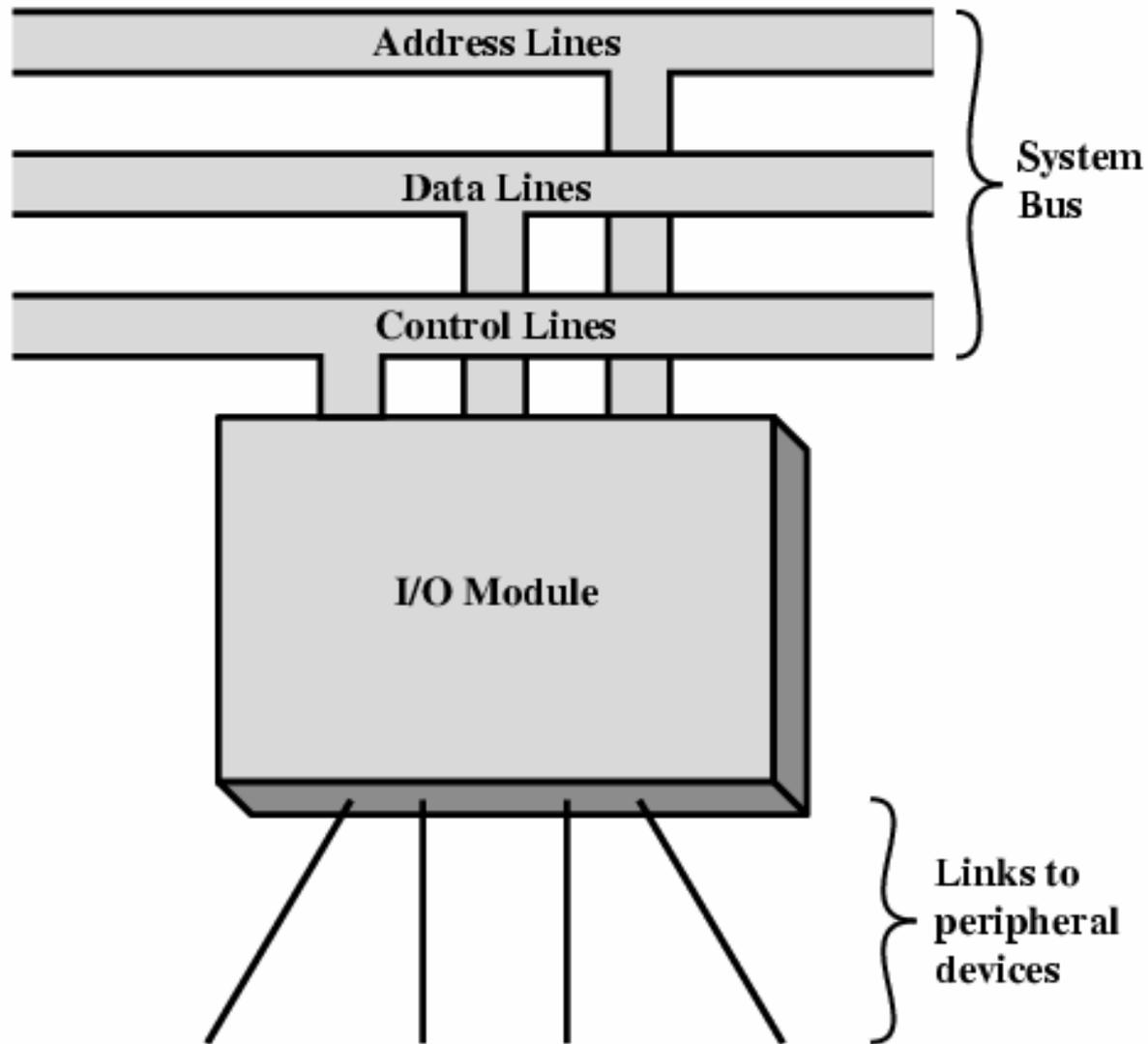
## *Funciones del Módulo de E/S*

---

- Interfase entre CPU y Memoria
- Interfases entre uno o más periféricos

# *Modelo genérico del Módulo de E/S*

---



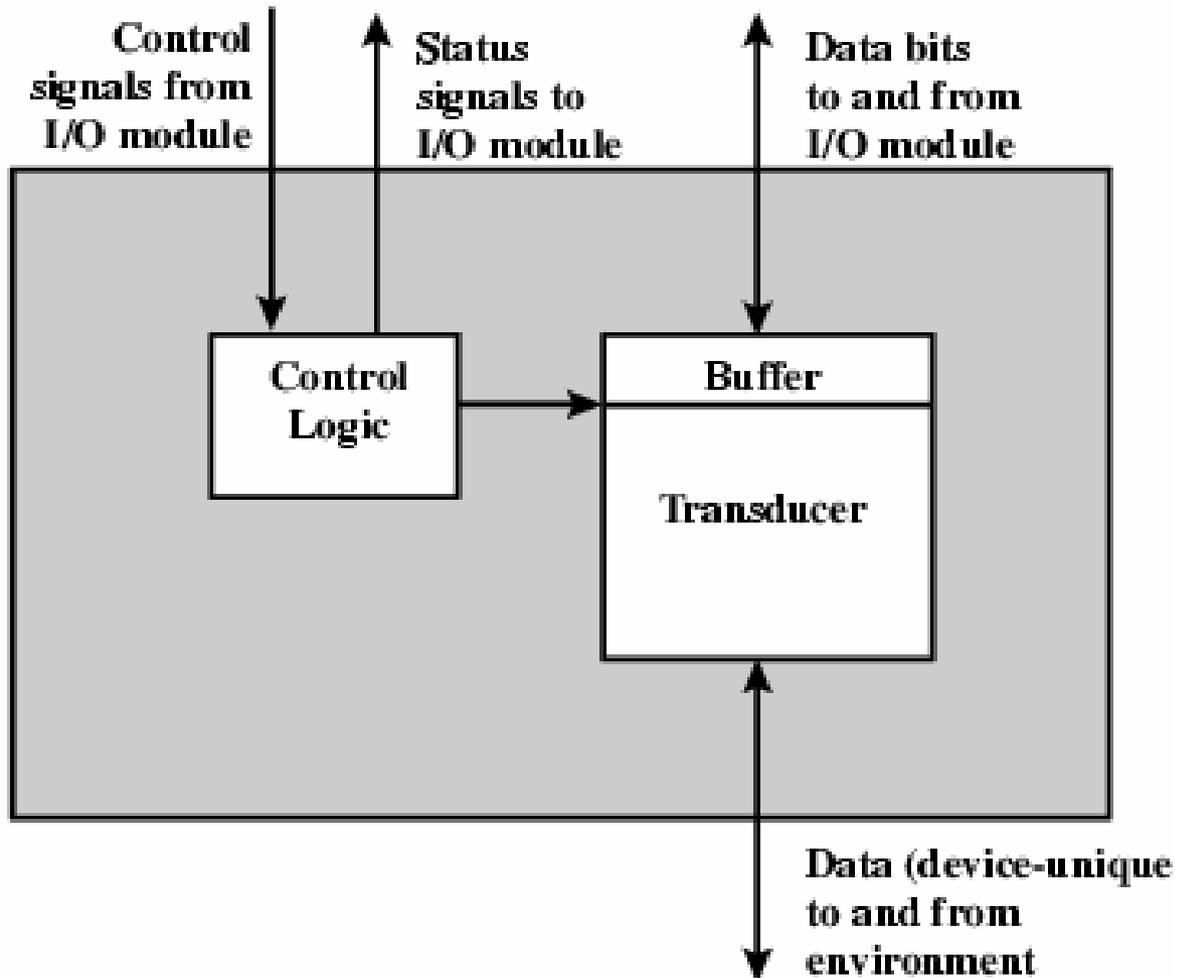
# *Dispositivos Externos*

---

- De interacción con humanos
  - Pantalla, impresora, teclado
- De interacción con máquinas
  - Monitoreo y control
- Comunicación
  - Modem
  - Placa de Red (NIC)

## *Diagrama de bloque del dispositivo externo*

---



## *Funciones del Módulo E/S*

---

- Control y temporización
- Comunicación con la CPU
- Comunicación con el Dispositivo
- Almacenamiento temporal de Datos
- Detección de Errores

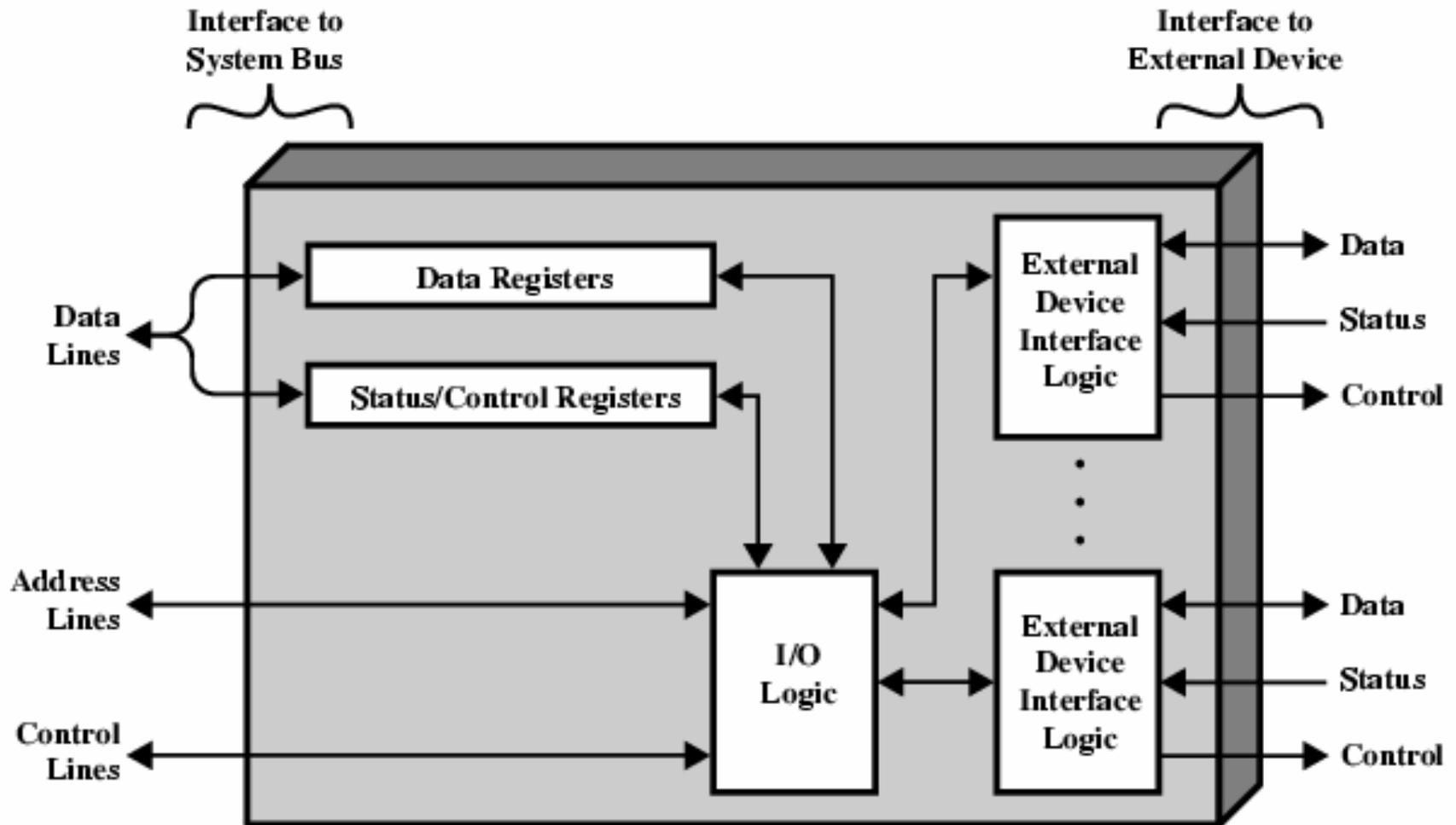
## *Etapas E/S*

---

- La CPU verifica el estado del módulo E/S
- El módulo E/S devuelve su estado
- Si está listo, la CPU envía una petición para transferencia de datos
- Módulo obtiene datos del dispositivo
- El Módulo transfiere esos datos a la CPU
- Variables para DMA, etc.

# Diagrama del Módulo E/S

---



## *Decisiones en el Módulo E/S*

---

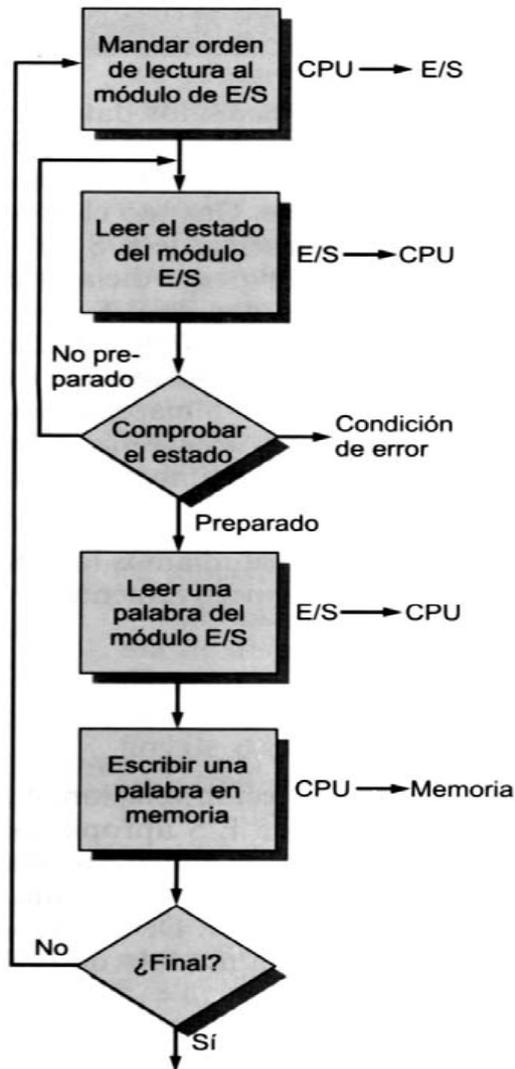
- Esconder o revelar propiedades del dispositivo a la CPU
- Soporte de uno o varios dispositivos
- Funciones de control del dispositivo o dejarlas a CPU
- Decisiones del SO
  - Ej. Unix trata todo dispositivo como archivo

## *Técnicas de control de la E/S*

---

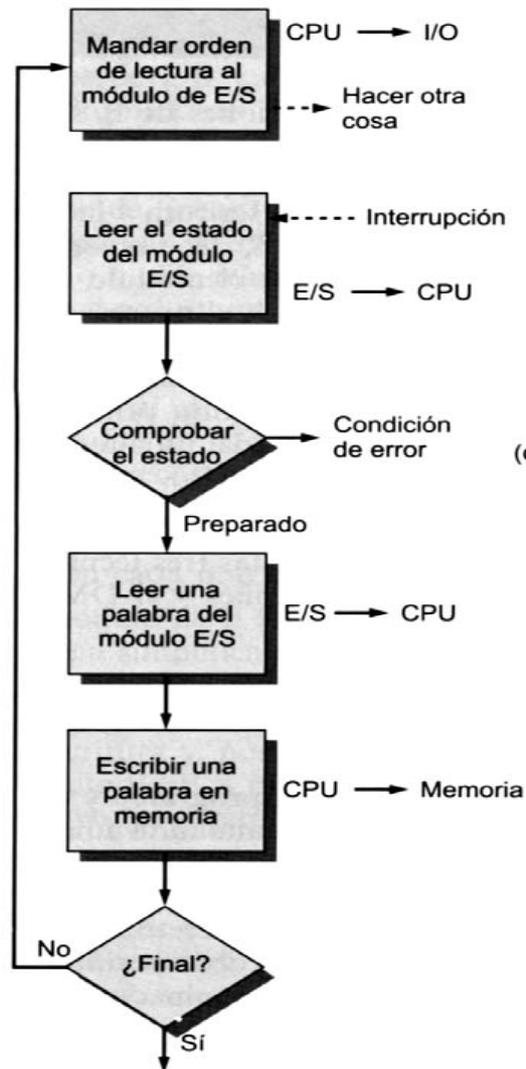
- Programada o Polling (Encuesta)
- Controlada por Interrupciones
  - Con Acceso Directo a Memoria (DMA)

# Las 3 técnicas para la Entrada de un bloque de datos



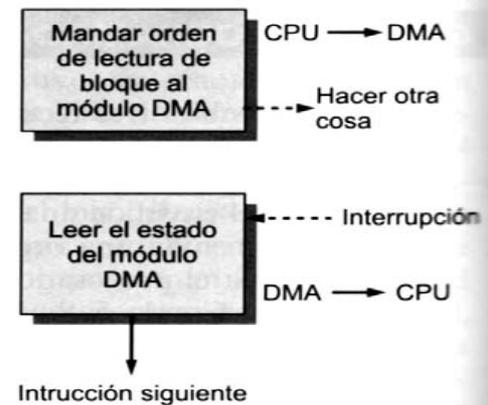
Instrucción siguiente

(a) E/S programada



Instrucción siguiente

(b) E/S mediante interrupciones



(c) Acceso directo a memoria

## *E/S Programada*

---

- La CPU tiene el control directo sobre la E/S
  - Censa el estado para verificar atención
  - Lee y escribe comandos
  - Transfiere datos
- La CPU espera a que el Módulo de E/S complete su operación
- Sigue haciendo lo mismo para cada dispositivo, uno por uno, cíclicamente (Polling)
- Se pierde tiempo de CPU

## *Detalle de la E/S Programada*

---

- La CPU pide una operación de E/S
- El Módulo la realiza
- El Módulo configura sus bits de estado
- La CPU verifica periódicamente estos bits
- El Módulo no informa directamente a la CPU
- EL Módulo no interrumpe a la CPU
- La CPU debe esperar o volver mas tarde

## *Comandos de E/S*

---

- La CPU usa (líneas de) **direcciones** para
  - **Identificar** el módulo (y dispositivo si hay mas por módulo)
- La CPU utiliza comandos (líneas de **datos**) para
  - **Control** – le dice al módulo que hacer
    - Ej. Encender el motor del disquete
  - **Verificación** – estado
    - Ej. Energía? Error?
  - **Lectura/Escritura**
    - El Módulo transfiere datos usando un buffer de y hacia el dispositivo

## *Direccionamiento de los Dispositivos de E/S*

- En E/S programada los datos transferidos se hacen de manera similar a la de la Memoria (desde el punto de vista de la CPU)
- Cada dispositivo tiene un único identificador
- Los comandos de la CPU contienen este identificador (direcciones)

# *Mapeo de la E/S*

---

- E/S mapeada en memoria
  - Los dispositivos y la Memoria comparten el mismo espacio de direcciones
    - Se utilizan las mismas líneas de control
  - La E/S se ve como la memoria (para Lectura/Escritura)
  - No hay instrucciones especiales para E/s
- E/S aislada
  - Memoria y E/S tienen distintos espacios
  - Utilizan líneas de control separadas
  - Instrucciones especiales para E/S

## *E/S controlada por Interrupciones*

---

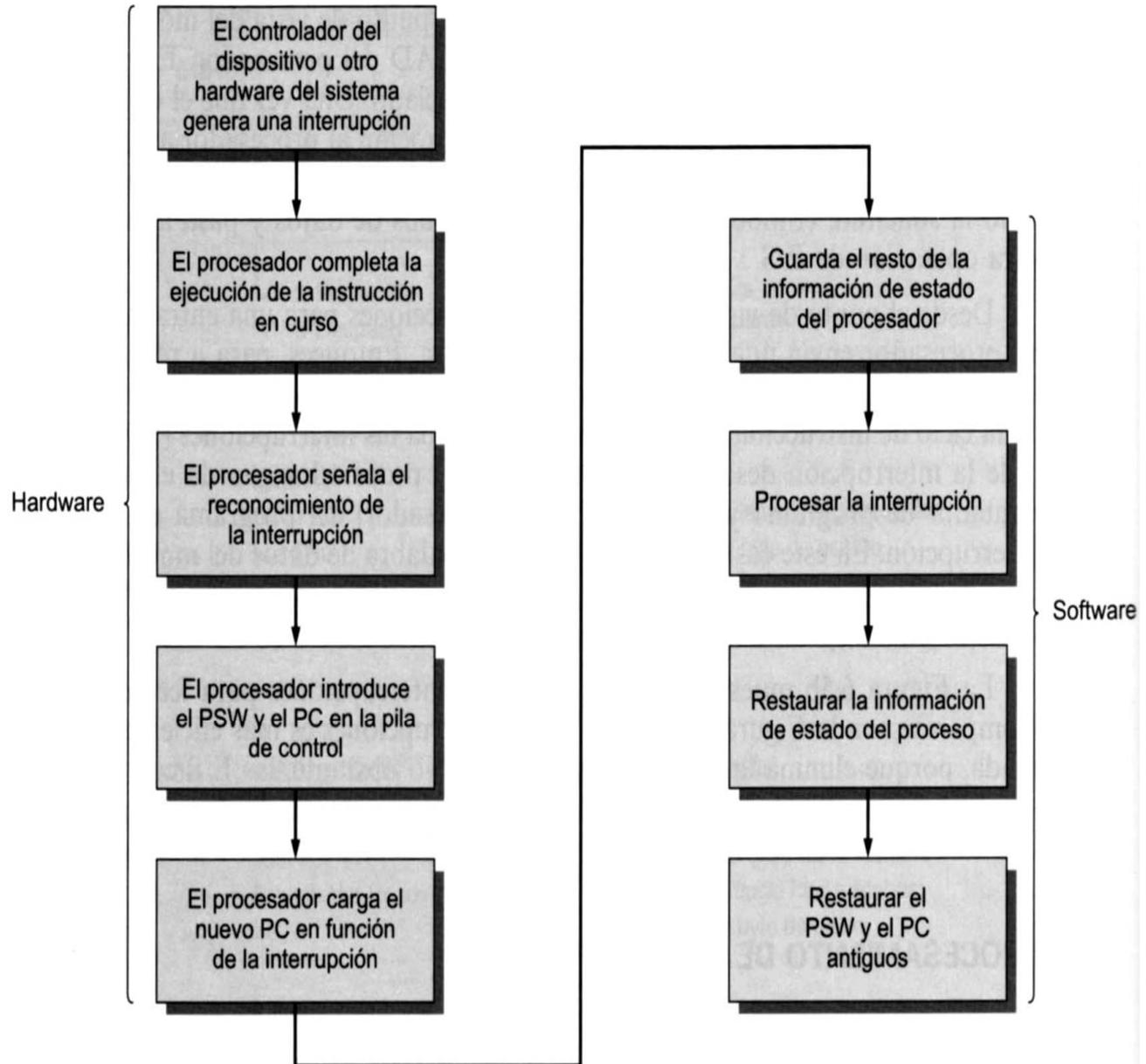
- Evita que la CPU espere por un dispositivo
- Evita el chequeo reiterado del dispositivo
- El Módulo de E/S interrumpe la CPU cuando esta listo

# *Operación Básica de la E/S por Interrupciones*

---

- La CPU pide una lectura y continúa con otra tarea.
- Mientras... el Módulo obtiene datos desde el dispositivo mientras la CPU hace otra cosa.
- Entonces el Módulo interrumpe la CPU.
- La CPU atiende al módulo y pide los datos.
- El Módulo se los transfiere.

# Procesamiento Interrupción

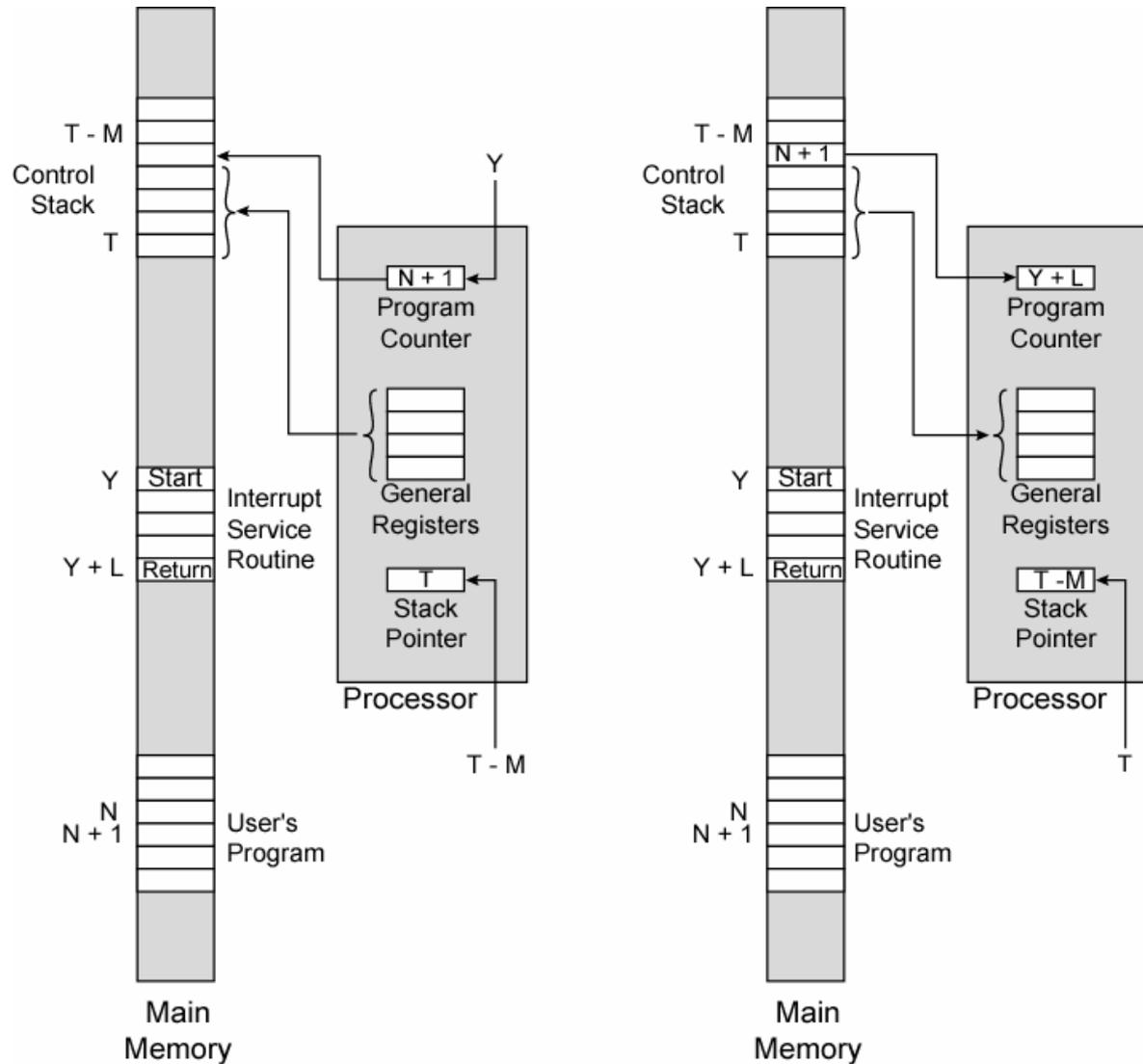


## *Punto de vista de la CPU*

---

- Emite un pedido de lectura
- Hace otra trabajo
- Verifica por interrupciones al final de cada ciclo de instrucción
- Si hay un pedido:
  - Salva el contexto (registros)
  - Procesa la interrupción
    - Busca datos y almacena

# Cambios en memoria y registros



(a) Interrupt occurs after instruction at location N

(b) Return from interrupt

## *Cuestiones de Diseño*

---

- ¿Como identificar el Módulo que pide la Interrupción?
- ¿Como tratar con múltiples interrupciones?
  - Ej. Cuando un servicio de atención de interrupción es interrumpido

## *Identificación del Módulo (1)*

---

- Cada Módulo tiene su línea de Interrupción
  - PC
  - Limita el número de dispositivos
- Encuesta por Software
  - La CPU verifica cada dispositivo
  - Lento

## *Identificación del Módulo (2)*

---

- Daisy Chain o Encuesta por Hardware
  - El reconocimiento de interrupción es devuelto a la cadena
  - El módulo responde poniendo su **vector** en el BUS
  - La CPU usa el **vector** para identificar la rutina que atenderá el pedido
- Arbitraje de BUS (Busmastering)(vectorizada)
  - El Módulo debe pedir el BUS antes de levantar el pedido de Interrupción, al recibir el reconocimiento pone su vector en el BUS
  - Ej. PCI & SCSI

## *Múltiples Interrupciones*

---

- Cada línea de Interrupción tiene asignada una prioridad
- Las de Alta prioridad pueden interrumpir las de Menor prioridad
- Si se utiliza bus mastering solo el maestro actual puede interrumpir

## *Ejemplo - PC Bus*

---

- 80x86 tiene una sola línea de Interrupción
- Los sistemas basados en el 8086 usaban un solo controlador de interrupciones 8259A
- El 8259A podía manejar 8 líneas de interrupción

## *Secuencia de Eventos*

---

- El 8259A acepta la interrupción
- El 8259A determina la prioridad
- El 8259A activa la línea INTR del 8086
- La CPU reconoce el pedido (ACK)
- El 8259A coloca el vector correspondiente en el BUS de Datos
- La CPU procesa la interrupción

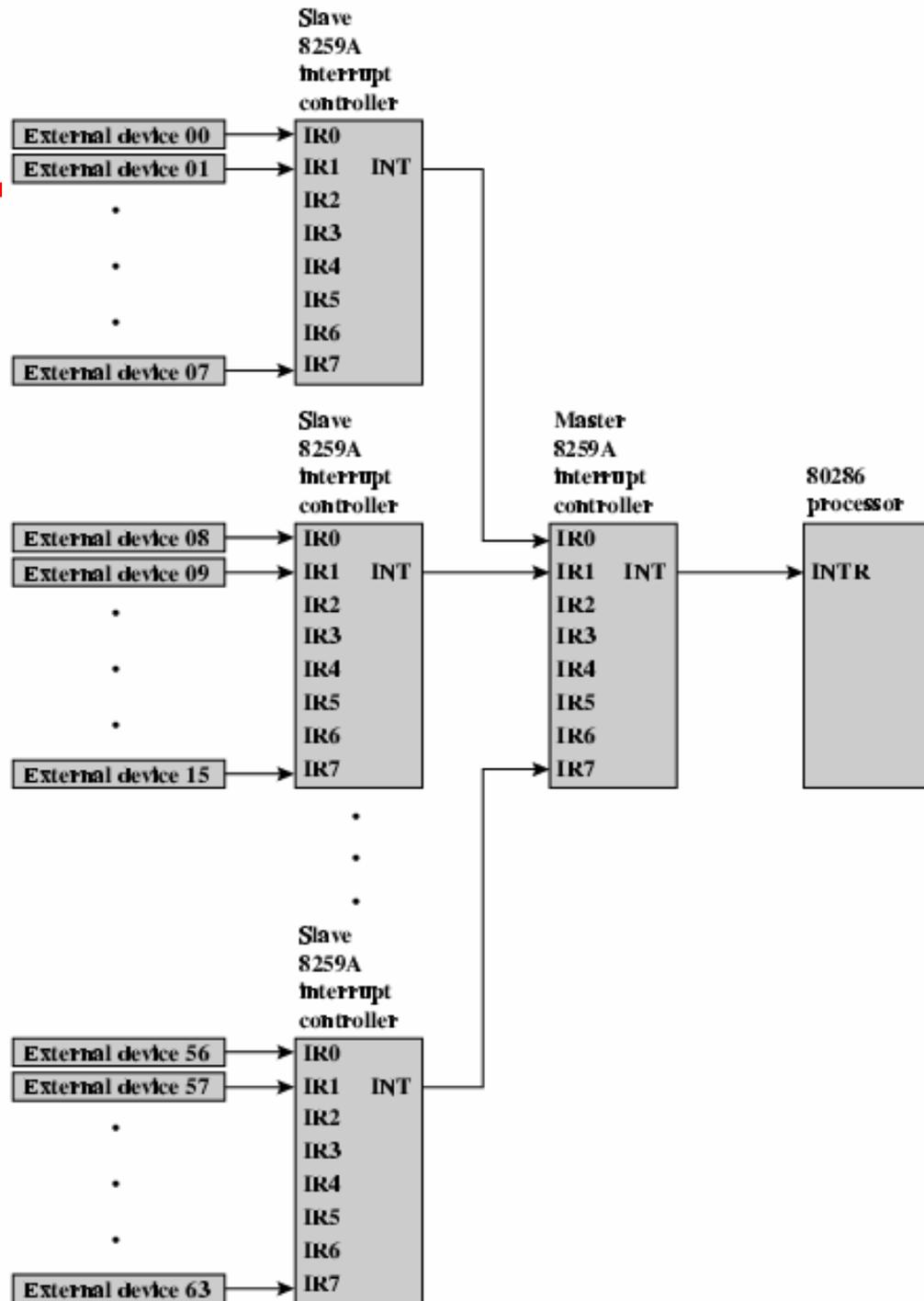
## *Sistema de Interrupciones del Bus ISA*

---

- El Bus ISA tiene dos 8259A en cascada
- El enlace es vía la línea de interrupción 2
- Esto da 15 líneas
- La IRQ 9 es utilizada para rutear los pedidos que intenten usar la IRQ 2
  - Compatible hacia atrás
- Está incorporado en el chip set

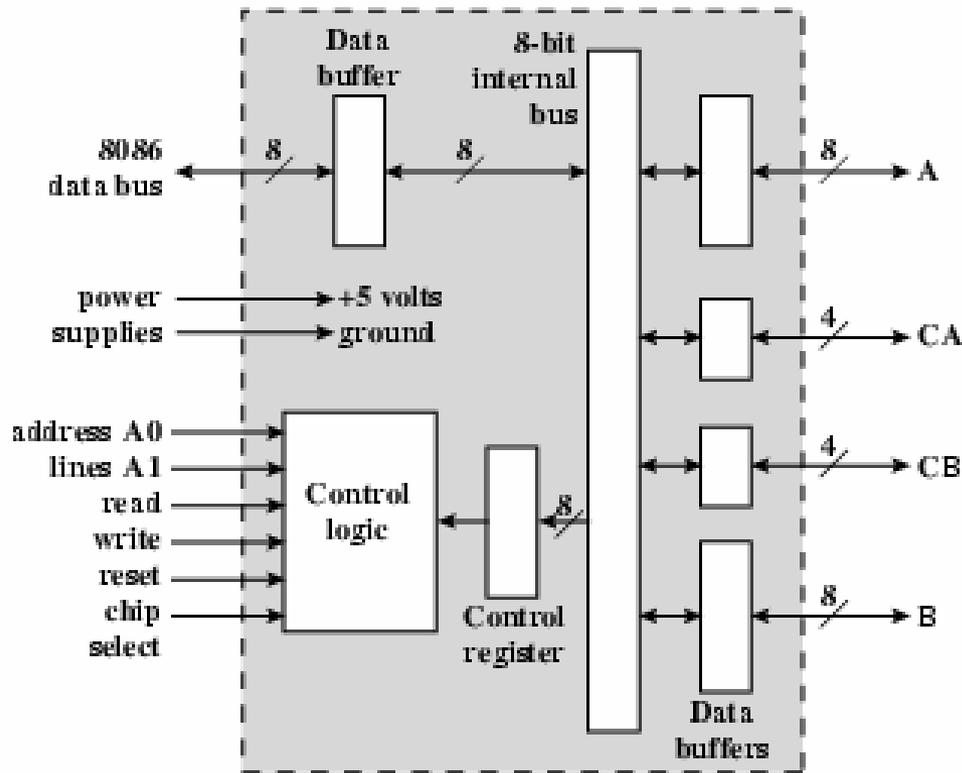
---

# Controlador de Interrupciones 82C59A

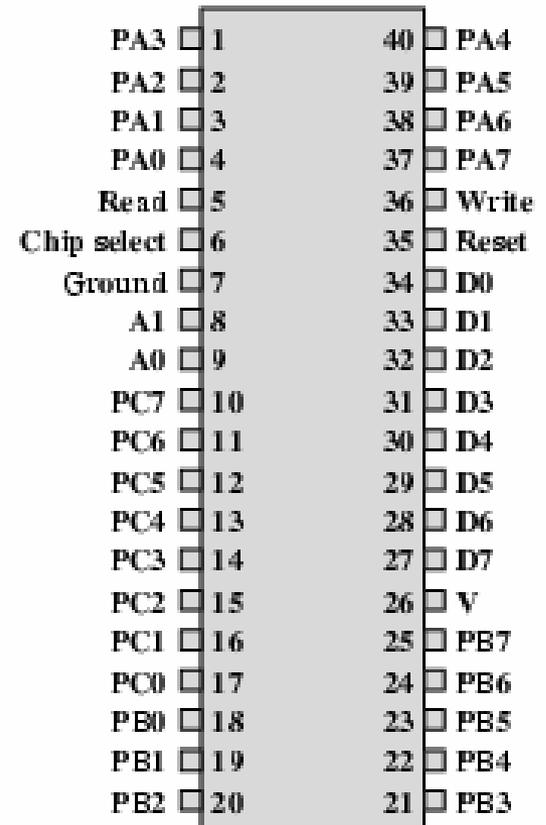


# Intel 82C55A

## Interfase Programable de Periféricos



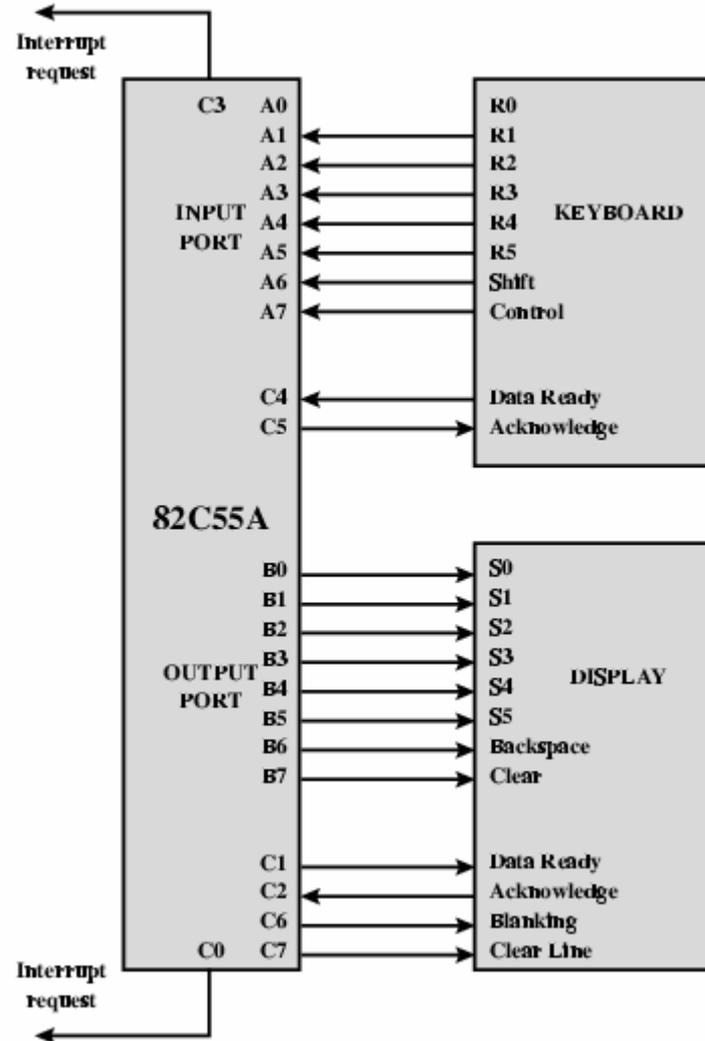
(a) Block diagram



(b) Pin layout

# Uso del 82C55A Para controlar Teclado /Display

---



## *Acceso Directo a Memoria*

---

- Ya sea la E/S Programada o por Interrupciones, siempre interviene la CPU
  - Limita la velocidad de transferencia
  - La CPU está siempre involucrada
- La respuesta es DMA

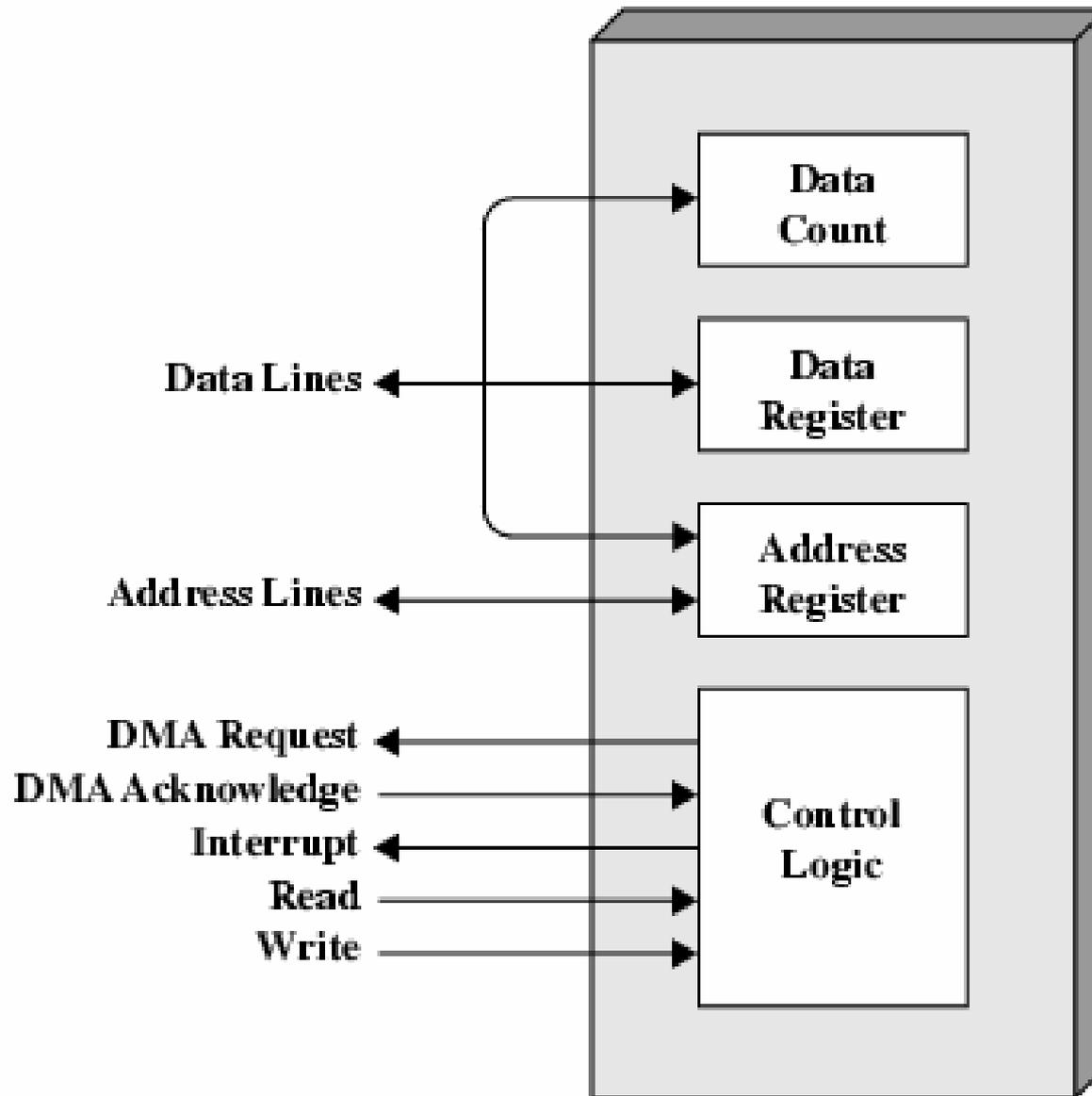
## *Función DMA*

---

- Módulo **Adicional** (hardware) en el bus
- EL controlador DMA toma el lugar de la CPU para la E/S

# *Diagrama del Módulo DMA*

---



## *Operación DMA*

---

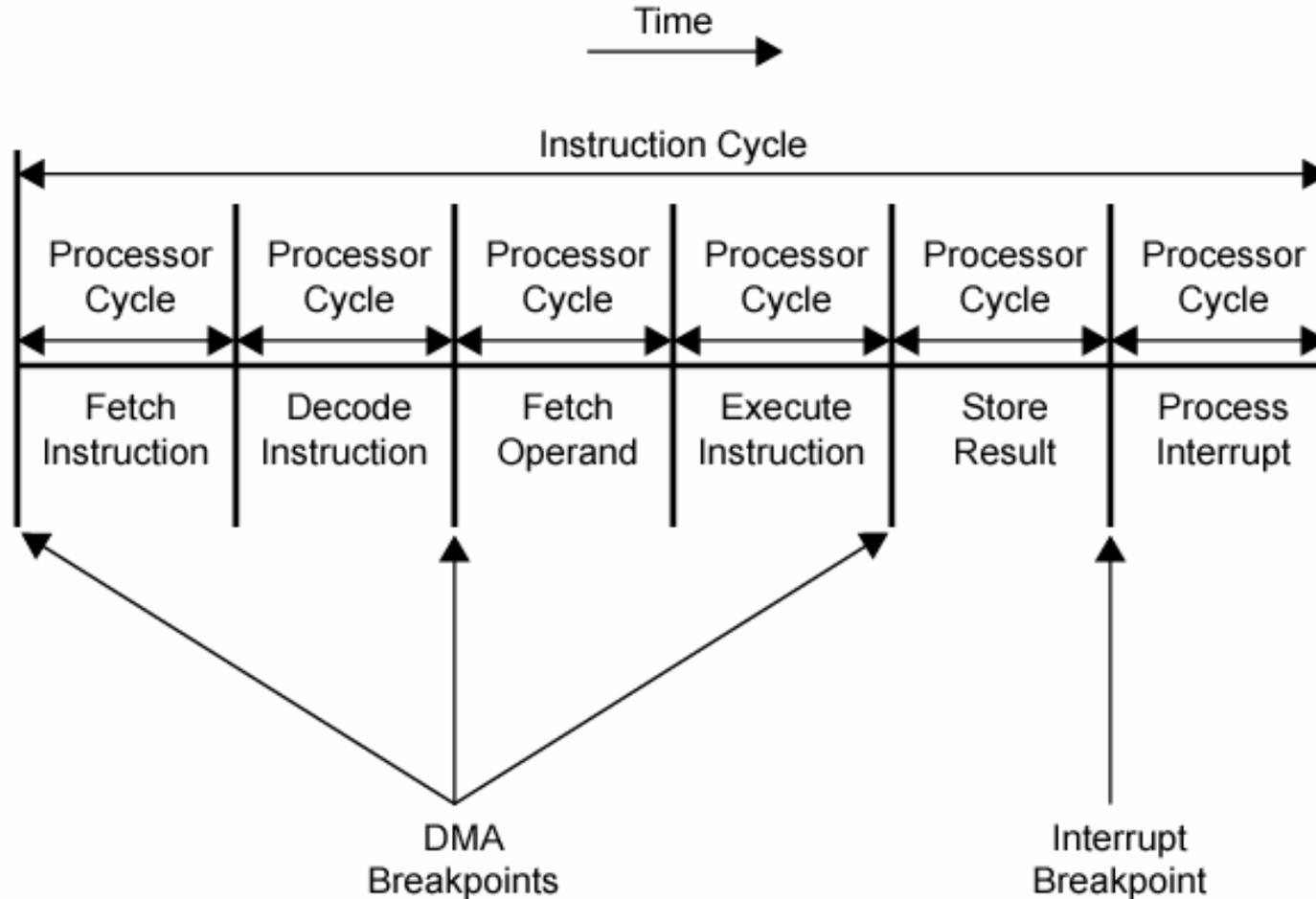
- La CPU le transfiere al controlador DMA :
  - Si se trata de Lectura o Escritura
  - La dirección del Dispositivo
  - La dirección de comienzo del bloque de memoria para los datos a transferir
  - La cantidad de datos a transferir
- La CPU atiende otras tareas
- EL controlador DMA realiza la transferencia
- El controlador DMA interrumpe la CPU cuando ha concluido su trabajo

## *Robo de Ciclo en Transferencia DMA*

---

- El controlador necesita tomar el control del bus durante un ciclo
- Transfiere una palabra de los datos
- No es como la interrupción
  - La CPU no necesita salvar su contexto
- La CPU se detiene justo antes de acceder al bus
  - Antes de la busca de un operando o de la escritura de datos
- Aunque hace que la CPU sea mas lenta pero es mejor que si lo hiciera la misma CPU

# Puntos de ruptura de ciclo de instrucción



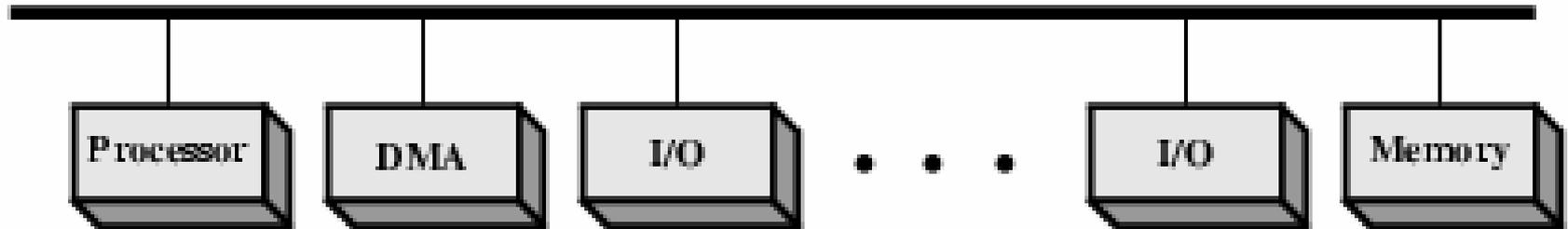
## *Por otro lado...*

---

- ¿Que efecto tiene el cache de memoria en el DMA?
- Pista: ¿Cuántos buses disponibles hay en el sistema?

# Configuraciones DMA (1)

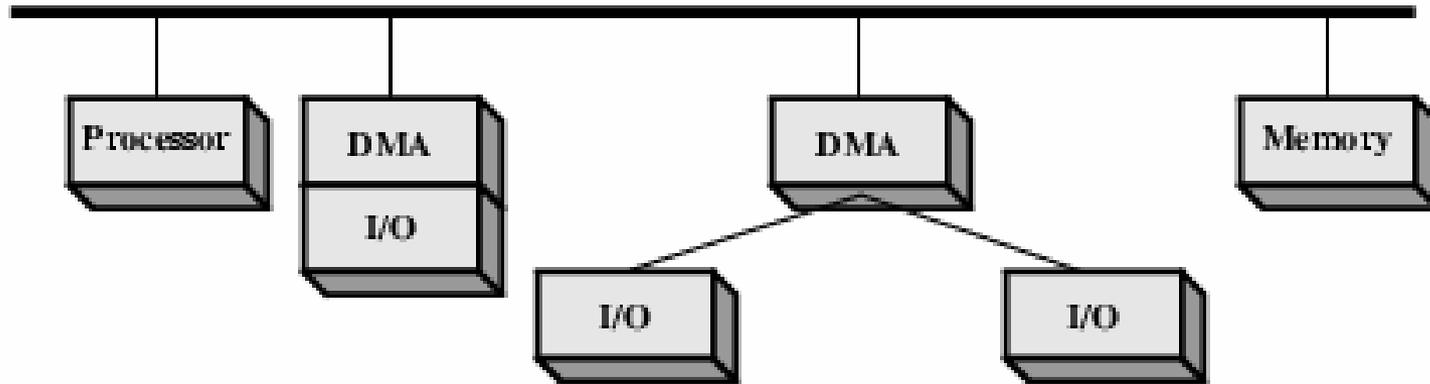
---



- Un solo Bus, Modulo separado
- Cada transferencia usa el bus dos veces
  - E/S a DMA luego de DMA a memoria
- La CPU está suspendida dos veces

## Configuraciones DMA (2)

---

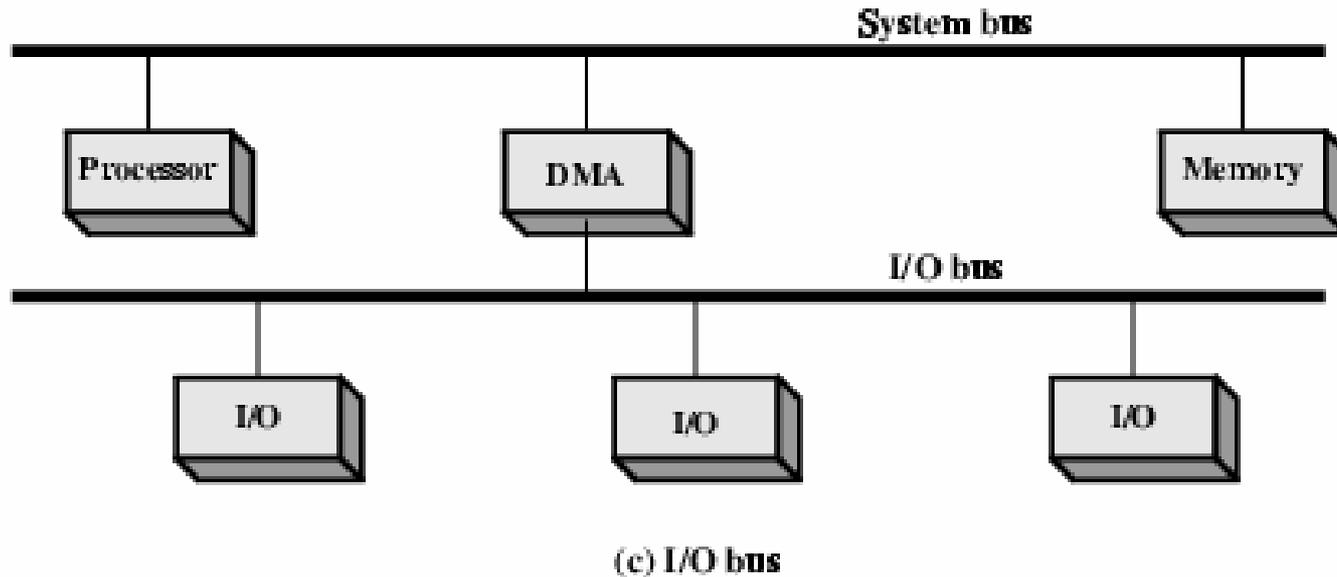


(b) Single-bus, Integrated DMA-I/O

- Un solo Bus, Controlador DMA Integrado
- El controlador puede manejar mas de 1 dispositivo
- Cada transferencia usa el bus una sola vez
  - DMA a memoria
- La CPU está suspendida una sola vez

## Configuraciones DMA (3)

---



- Bus separado de E/S
- El Bus suporta todos los dispositivos habilitados para DMA
- Cada transferencia usa el bus una vez
  - DMA a memoria
- CPU está suspendida una vez

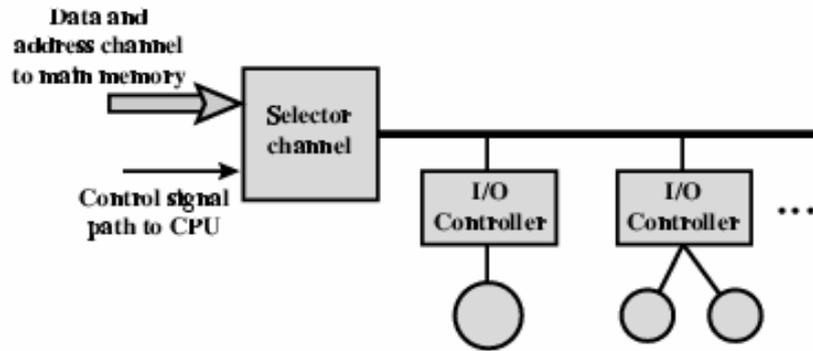
## *Canales de E/S*

---

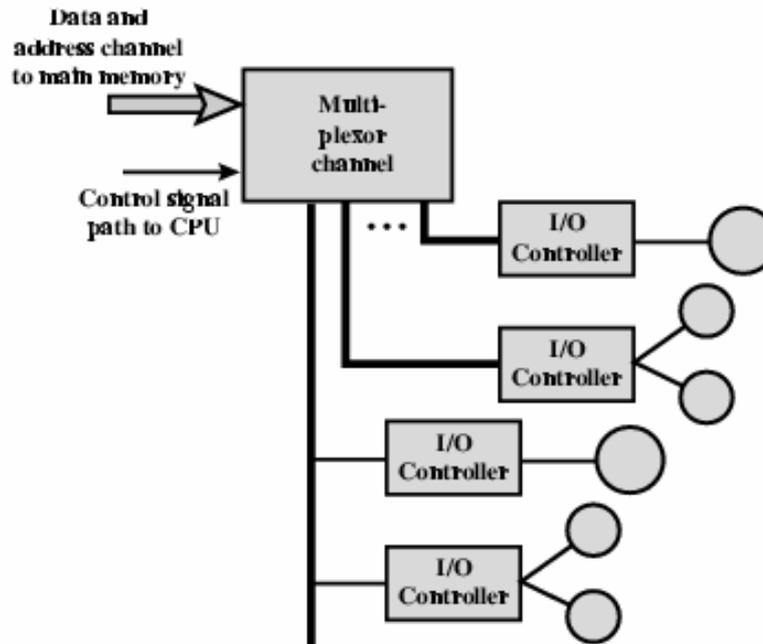
- Los dispositivos de E/S son cada vez mas sofisticados
  - Ej. Placas de video 3D
- La CPU instruye al controlador E/S para la transferencia
- El controlador realiza toda la transferencia
- Mejora la velocidad
  - Le quita la carga a la CPU
  - Un procesador dedicado es mas rápido

# Arquitectura de Canales de E/S

---



(a) Selector



(b) Multiplexor

## *Interfaz externa*

---

- Conexión de varios dispositivos
- ¿Serie o Paralelo?
- ¿Procesador dedicado/memoria/buses?
- Ej. FireWire, InfiniBand, USB, SCSI, SATA, PATA

# *Interfaces*

---

- Firewire IEEE 1394x
- InfiniBand
- USB
- SCSI
- IDE, EIDE, PATA, SATA
- RS232, RS422, RS485
- Ethernet - WiFi
- Bluetooth
- VGA-DVI