

# **Conceptos de Arquitectura de Computadoras**

---

## **Clase 3**

### **Entrada/Salida**

# Problemas de Entrada/Salida

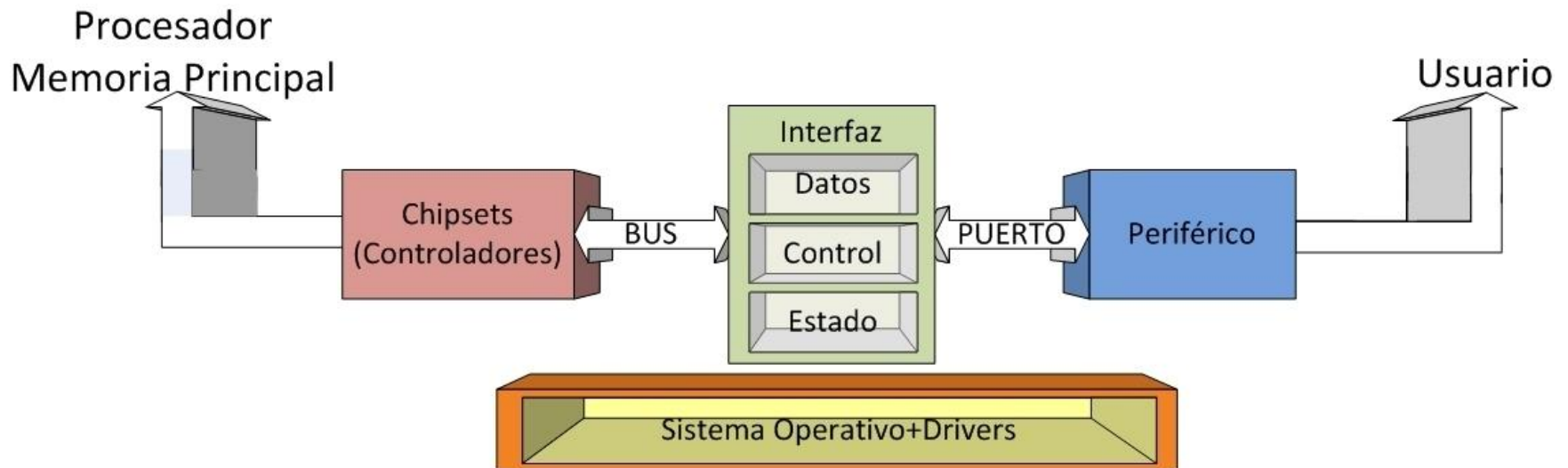
---

- Gran variedad de periféricos con varios métodos de operación.
  - Trasmisión de diferentes cantidades de datos.
  - A diferentes velocidades.
  - Usan diferentes formatos de dato y tamaño de palabra.
- Todos más lentos que la CPU y la RAM.
- Necesidad de módulos de E/S (con alguna “inteligencia”)

# Módulo de E/S

---

- Realiza la interfaz entre el procesador y la memoria (bus) y los periféricos.
- Pueden manejar uno o más periféricos.



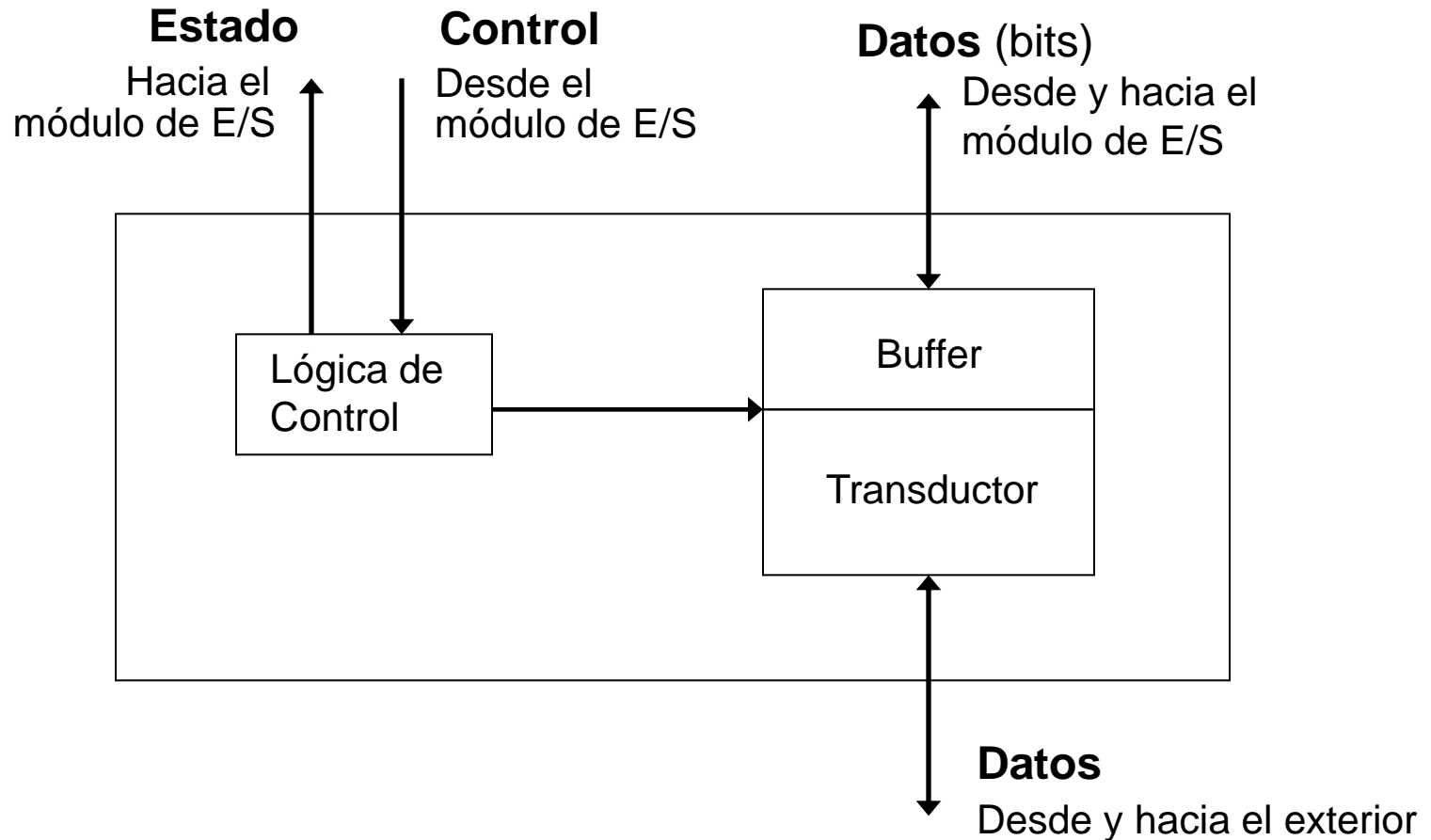
# Dispositivos externos

---

- e/s básicos
  - monitor/pantalla, mouse, teclado
- almacenamiento
  - disco duro, CD, DVD
- impresión
  - impresora, escáner
- comunicación con dispositivos remotos
  - modem, acceso/interfaz de red
- multimedia
  - micrófono, parlantes
- automatización y control
  - Sensores, alarmas, adquisición de datos

# Dispositivo externo tipo

---



# Características de un puerto

---

- Interfase entre el periférico y el módulo de E/S
- Señales de Control, Estado y Datos
  - Señal de Control: función a realizar
    - Ej: INPUT ó READ, OUTPUT ó WRITE
  - Señal de Estado: READY/NOT READY
  - Control lógico: manejo de direccionamiento
  - Transductor: conversión del datos
  - Buffer: adaptación (1, 8 o 16 bits)

# Código ASCII

00 NUL	10 DLE	20 SP	30 0	40 @	50 P	60 `	70 p
01 SOH	11 DC1	21 !	31 1	41 A	51 Q	61 a	71 q
02 STX	12 DC2	22 "	32 2	42 B	52 R	62 b	72 r
03 ETX	13 DC3	23 #	33 3	43 C	53 S	63 c	73 s
04 EOT	14 DC4	24 \$	34 4	44 D	54 T	64 d	74 t
05 ENQ	15 NAK	25 %	35 5	45 E	55 U	65 e	75 u
06 ACK	16 SYN	26 &	36 6	46 F	56 V	66 f	76 v
07 BEL	17 ETB	27 '	37 7	47 G	57 W	67 g	77 w
08 BS	18 CAN	28 (	38 8	48 H	58 X	68 h	78 x
09 HT	19 EM	29 )	39 9	49 I	59 Y	69 i	79 y
0A LF	1A SUB	2A *	3A :	4A J	5A Z	6A j	7A z
0B VT	1B ESC	2B +	3B ;	4B K	5B [	6B k	7B {
0C FF	1C FS	2C ,	3C <	4C L	5C \	6C l	7C
0D CR	1D GS	2D -	3D =	4D M	5D ]	6D m	7D }
0E SO	1E RS	2E .	3E >	4E N	5E ^	6E n	7E ~
0F SI	1F US	2F /	3F ?	4F O	5F _	6F o	7F DEL

NUL	Null	FF	Form feed	CAN	Cancel
SOH	Start of heading	CR	Carriage return	EM	End of medium
STX	Start of text	SO	Shift out	SUB	Substitute
ETX	End of text	SI	Shift in	ESC	Escape
EOT	End of transmission	DLE	Data link escape	FS	File separator
ENQ	Enquiry	DC1	Device control 1	GS	Group separator
ACK	Acknowledge	DC2	Device control 2	RS	Record separator
BEL	Bell	DC3	Device control 3	US	Unit separator
BS	Backspace	DC4	Device control 4	SP	Space
HT	Horizontal tab	NAK	Negative acknowledge	DEL	Delete
LF	Line feed	SYN	Synchronous idle		
VT	Vertical tab	ETB	End of transmission block		

# Funciones de un módulo de E/S

---

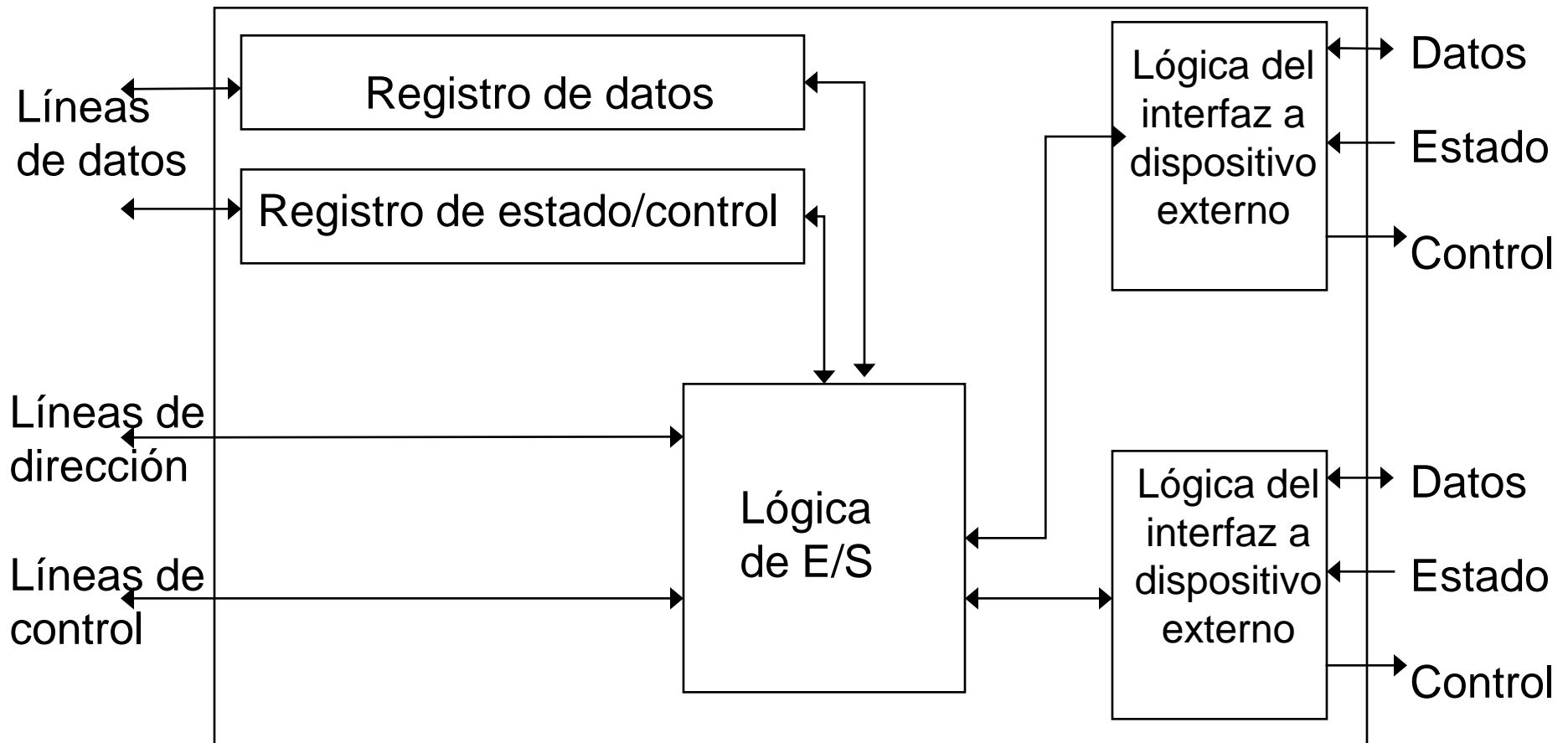
- Control y temporización de uno o más dispositivos externos
- Interpretar las órdenes que recibe de CPU y transmitirlas al periférico
- Comunicación con la CPU (registros) y Memoria
- Controlar las transferencias de datos entre CPU y el periférico (convertir formatos, adaptar velocidades)
- Comunicación con los dispositivos (periféricos)
- Informar a la CPU del estado del periférico
- Almacenamiento temporal (buffering) de datos
- Detección de errores



# Diagrama en bloques de un módulo de E/S

Interfaz al bus del sistema

Interfaz a dispositivos externos



# Capacidades de un módulo de E/S

---

- Ocultar las propiedades del dispositivo a la CPU.
  - Ej: temporizados, formatos, electromecanismos ...etc
- Ocuparse de uno o varios dispositivos.
- Controlar o no las funciones del dispositivo.
  - Canales de E/S ó procesador de E/S (manejo de parte importante de la carga del procesamiento). Presentes en Mainframes.
  - Controlador de E/S ó controlador de dispositivo (manejo primitivo). Presentes en microcomputadoras.

# Operación de Entrada ó Salida

---

Requiere:

- Direccionamiento
  - E/S mapeada en memoria
  - E/S aislada
- Transferencia de información
  - Lectura ó escritura
- Gestión de la transferencia
  - Mecanismos de sincronización y control de la transferencia de datos

# Direccionamiento de E/S

---

- E/S asignada en memoria (memory-mapped)
  - Dispositivos de E/S y memoria comparten un único espacio de direcciones.
  - E/S se parece a la memoria de lectura/escritura.
  - No hay órdenes específicas para E/S.
    - Variedad de órdenes de acceso a memoria (programación eficiente)
- E/S aislada
  - Espacios de direcciones separados.
  - Necesidad de líneas especiales de E/S y de memoria.
  - Órdenes específicas para E/S.
    - Conjunto limitado de instrucciones.

# Técnicas de gestión de E/S

---

- E/S Programada con espera de respuesta
- E/S con interrupciones
- E/S con acceso directo a memoria (DMA)

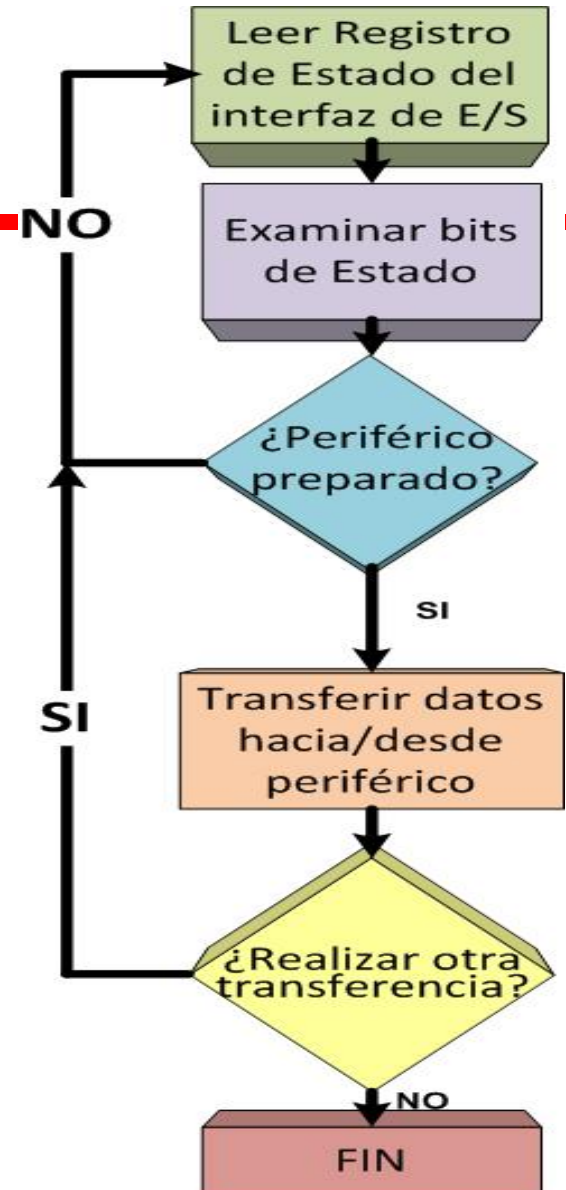
# E/S programada

---

- Intercambio de datos entre la CPU y el módulo
- La CPU tiene control directo sobre la operación de E/S
  - Comprobación del estado del dispositivo
  - Envío de comandos de lectura/escritura
  - Transferencia de datos
- La CPU espera que el módulo E/S termine la operación
- Por lo tanto la CPU permanece ociosa durante un período de tiempo (no deseable)

# Detalles de la E/S programada

- La CPU solicita la operación de E/S al módulo.
- El módulo E/S realiza la operación.
- El módulo E/S activa los bits de estado del dispositivo direccionado y espera.
- La CPU comprueba periódicamente el estado de esos bits, hasta que detecta que la operación fue completada.
- En caso contrario la CPU espera y vuelve a comprobarlo más tarde.



# Órdenes (comandos) de E/S

---

- La CPU emite una dirección
  - Especifica el módulo (y el dispositivo si hay más de uno por módulo)
- La CPU da una orden
  - Control: indica al módulo qué hacer
    - Ejemplo: rebobinar una cinta magnética.
  - Test: comprueba el estado del módulo y sus periféricos
    - Ejemplo: ¿está conectado? ¿hubo algún error?
  - Lectura/Escritura
    - Transfiere datos desde o hacia el dispositivo por el bus de datos.



# E/S con interrupciones

---

- La CPU no tiene que esperar la finalización de la tarea de E/S, puede seguir procesando.
- No se repite la comprobación de los estados de los módulos
- El módulo envía un pedido de interrupción a la CPU cuando está listo nuevamente.

# **E/S mediante interrupciones: ¿qué hace la CPU ???**

---

- La CPU envía una orden de lectura (READ).
  - El módulo E/S obtiene los datos del periférico mientras que la CPU realiza otro trabajo.
- La CPU chequea si hay pedidos de interrupciones pendientes al final de cada ciclo de instrucción.
  - El módulo E/S emite un pedido de interrupción a la CPU.
- La CPU detecta el pedido, guarda el contexto, interrumpe el proceso y realiza la gestión de la interrupción.
- La CPU solicita los datos.
  - El módulo E/S transfiere los datos.

# Cuestiones de diseño

---

- ¿Cómo saber qué dispositivo ha provocado la interrupción?
- Con múltiples interrupciones ¿Cómo elegir la interrupción que se debe atender? ¿Establecemos prioridades?
  - una rutina de interrupción que a su vez es interrumpida.

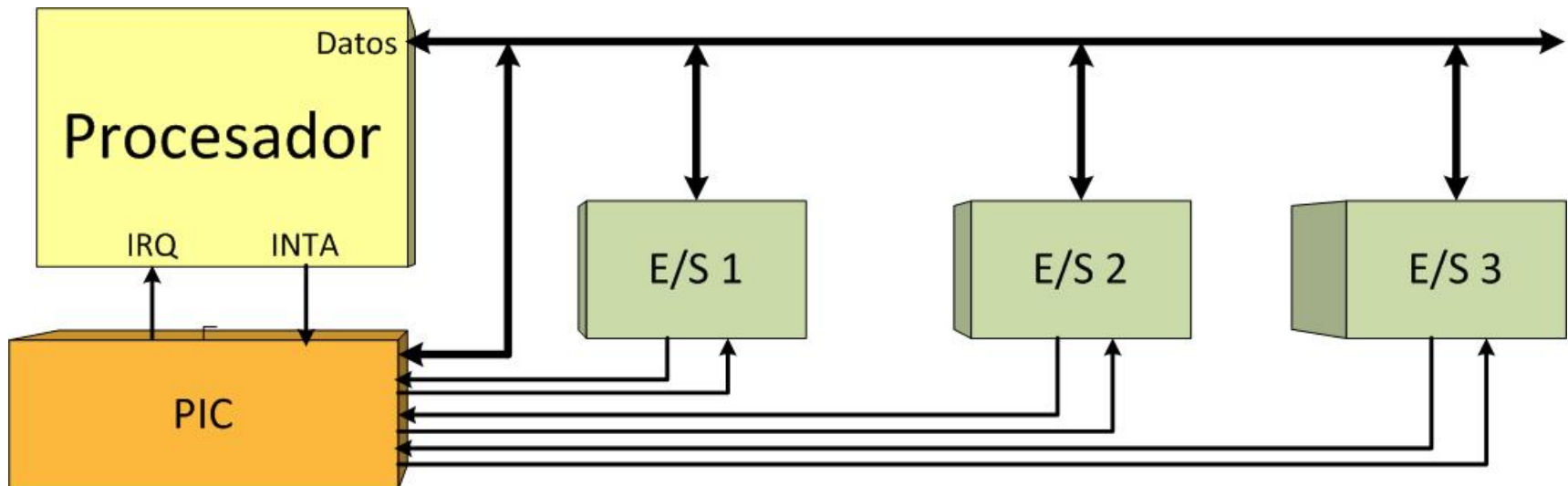
# Identificación del módulo que interrumpe

---

- Diferentes líneas para cada módulo
  - Computadoras personales (PC)
  - Limita el número de dispositivos
- Consulta software (Poll o encuesta)
  - Ocurrido un pedido de interrupción la CPU consulta a cada módulo para determinar quien fue el demandante.
  - Resulta lento.
- Conexión en cadena (daisy chain) "hard poll"
  - La línea de reconocimiento de interrupción se conecta encadenando los módulos, la línea de pedido es compartida.
  - Una vez enviada la confirmación de parte de la CPU el módulo responderá colocando un vector (palabra), en el bus, que lo identifica.
  - La CPU emplea el vector como puntero para acceder a la rutina de servicio.

# Interrupciones múltiples

- Todas las líneas de interrupción tienen un orden de prioridad
- Las líneas con más prioridad pueden interrumpir a las líneas con menor prioridad.
- Si existe un maestro del bus, solo él puede interrumpir.



# Estructuras de interrupciones (1)

---

- El Intel 8086 tiene sólo una línea de petición de interrupción (INTReq) y por lo tanto una sola de confirmación (INTAck).
- Se deberá utilizar un árbitro o gestor de interrupciones externo, el 8259A (PIC).
- Este chip tiene 8 líneas de interrupción, por lo tanto podrá manejar 8 módulos de E/S.
  - Usando conexión en cascada se puede gestionar hasta 64 módulos.

# Estructuras de interrupciones (2)

---

- Puede ser útil tener una Interfase de Periféricos Programable (PIO).
- El Intel 8255A es un chip para usar también en entorno 8086.
- Es un módulo de E/S de propósito general.
- Posee 24 líneas de E/S programable vía los registros de control.
- Usado para una variedad de periféricos.
  - Ej: Teclado/Pantalla

# Análisis

---

- Las operaciones de E/S mediante interrupciones son más efectivas que las programadas.
- Pero ambas necesitan la intervención directa de la CPU.
  - La velocidad de transferencia es limitada.
  - La CPU permanece ocupada mucho tiempo durante la operación.
- ¿Qué sucede si el volumen a transferir es grande?



# Ejemplo E/S con periférico lento

---

- Procesador a 200 MHz (tiempo ciclo reloj = 5 ns; Ciclos por instrucción CPI = 2 , en promedio)
  - Una instrucción tarda en promedio  $2 \times 5 \text{ ns} = 10 \text{ ns} \Rightarrow$  la computadora puede ejecutar  $\sim 100 \text{ Mips}$
- Queremos imprimir un archivo de 10 Kbytes en una impresora láser de 20 páginas por minuto
  - 1 página  $\cong 3.000$  caracteres (1 carácter = 1 byte)
  - La impresora imprime 60.000 caracteres por minuto = 1 Kbyte/s

## Ej. con periférico lento (2)

---

### a) E/S con espera de respuesta

- La CPU entra en un bucle y envía un nuevo byte cada vez que la impresora está preparada para recibirlo
  - La impresora tarda 10 seg en imprimir 10 Kbytes
  - **La CPU está ocupada con la operación de E/S durante 10 seg.**  
(en ese tiempo la CPU podría haber ejecutado 1000 millones de instrucciones)

# Ej. con periférico lento (3)

---

## b) E/S con interrupciones

- La impresora genera una interrupción cada vez que está preparada para recibir un nuevo byte.
  - Si la gestión de interrupción (ATI) tiene 10 instrucciones (salvar contexto, comprobar estado, transferir byte, restaurar contexto, rti)
  - Para transferir 10 Kbytes tenemos que ejecutar 10.000 veces la ATI
    - ⇒ ejecutar 100.000 instrucciones para atender al periférico ⇒ la CPU tarda 0,001 seg
- **La CPU está ocupada con la operación de E/S durante 0,001 seg.**

# Conclusión con periférico lento

---

- La E/S por interrupciones reduce en 10.000 veces el tiempo que la CPU está ocupada gestionando la impresora.

# Ejemplo E/S con periférico rápido

---

- Procesador a 200 MHz (tiempo ciclo reloj = 5 ns; Ciclos por instrucción CPI = 2 ciclos, en promedio)
  - Una instrucción tarda en promedio  $2 \times 5 \text{ ns} = 10 \text{ ns} \Rightarrow$  la computadora puede ejecutar  $\sim 100$  Mips
- Queremos transferir un archivo de memoria a disco de 10 Mbytes.
  - El disco posee una velocidad de transferencia de 10 MB/s (1 byte cada  $10^{-7}$  seg ó 100 nanoseg)

## Ej. con periférico rápido (2)

---

### a) E/S con espera de respuesta

- La CPU entra en un bucle y envía un nuevo byte cada vez que el disco está preparado para recibirlo
  - El disco tarda 1 seg. en recibir un archivo de 10 Mbytes
  - **La CPU está ocupada con la operación de E/S durante 1 seg.**

(en ese tiempo la CPU podría haber ejecutado 100 millones de instrucciones)

# Ej. con periférico rápido (3)

---

## b) E/S con interrupciones

- El disco genera una interrupción cada vez que está preparado para recibir un nuevo byte
  - Si la gestión de interrupción (ATI) tiene 10 instrucciones (salvar contexto, comprobar estado, transferir byte, restaurar contexto, rti)
  - Para transferir 10 Mbytes tenemos que ejecutar  $10^7$  veces la ATI
    - ⇒ ejecutar 100 millones de instrucciones para atender al periférico ⇒ la CPU tarda 1 seg.
- **La CPU está ocupada con la operación de E/S durante 1 seg.**

# Conclusión con periférico rápido

---

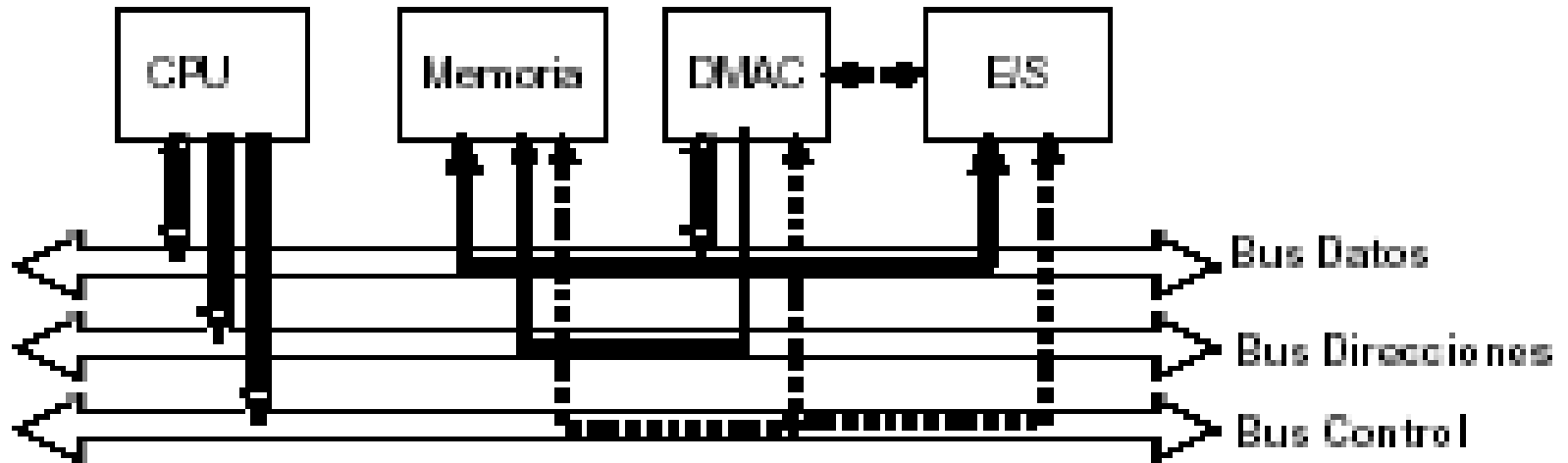
- La E/S por interrupciones no mejora el tiempo que la CPU está ocupada en atender al periférico.



# Acceso directo a memoria (DMA)

---

El controlador de DMA es un dispositivo capaz de controlar una transferencia de datos entre un periférico y memoria sin intervención de la CPU.



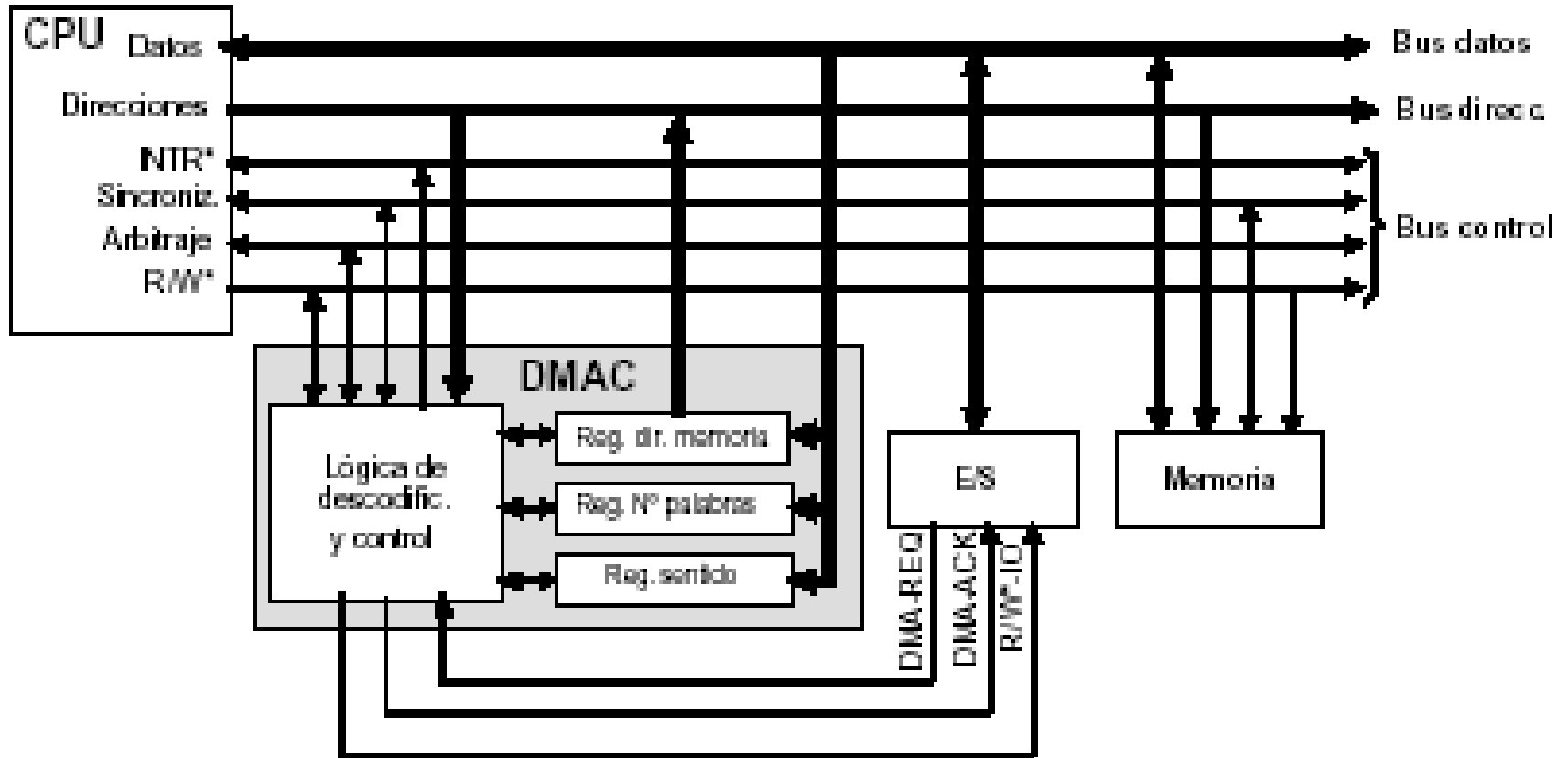
# Controlador de DMA

---

El Controlador de DMA (DMAC) debe actuar como maestro del bus durante la transferencia DMA y debe ser capaz de

- Solicitar el uso del bus mediante las señales y la lógica de arbitraje necesarias
- Especificar la dirección de memoria sobre la que se realiza la transferencia
- Generar las señales de control del bus
  - Tipo de operación (lectura/escritura)
  - Señales de sincronización de la transferencia

# Estructura de un DMAC



# Etapas de una transferencia DMA

---

## Inicialización de la transferencia

- La CPU debe enviar al interfaz del periférico y al DMAC los parámetros de la transferencia

### **Inicialización del interfaz (Bus master: CPU-Bus slave: Interfaz)**

- N° de bytes a transferir
- Tipo de transferencia (lectura/escritura)
- Otra información de control (pista, sector, etc.)

### **Inicialización controlador DMA (Bus master: CPU-Bus slave: DMAC)**

- N° de bytes o palabras a transferir
  - Tipo de transferencia (lectura/escritura)
  - Dirección de memoria inicial para la transferencia
  - N° de canal (para DMAs con varios canales)
- Después de la inicialización la CPU retorna a sus tareas y ya no se preocupa más de la evolución de la transferencia.

# Etapas de una transf. DMA (2)

---

## Realización de la transferencia

- Cuando el periférico está listo para realizar la transferencia se lo indica al DMAC
- El DMAC pide el control del bus y se realiza la transferencia entre el periférico y la memoria
  - **Bus master:** DMAC + Periférico - **Bus slave:** Memoria
  - Después de la transferencia de cada palabra se actualizan los registros del DMAC
    - N° de bytes o palabras a transferir
    - Dirección de memoria

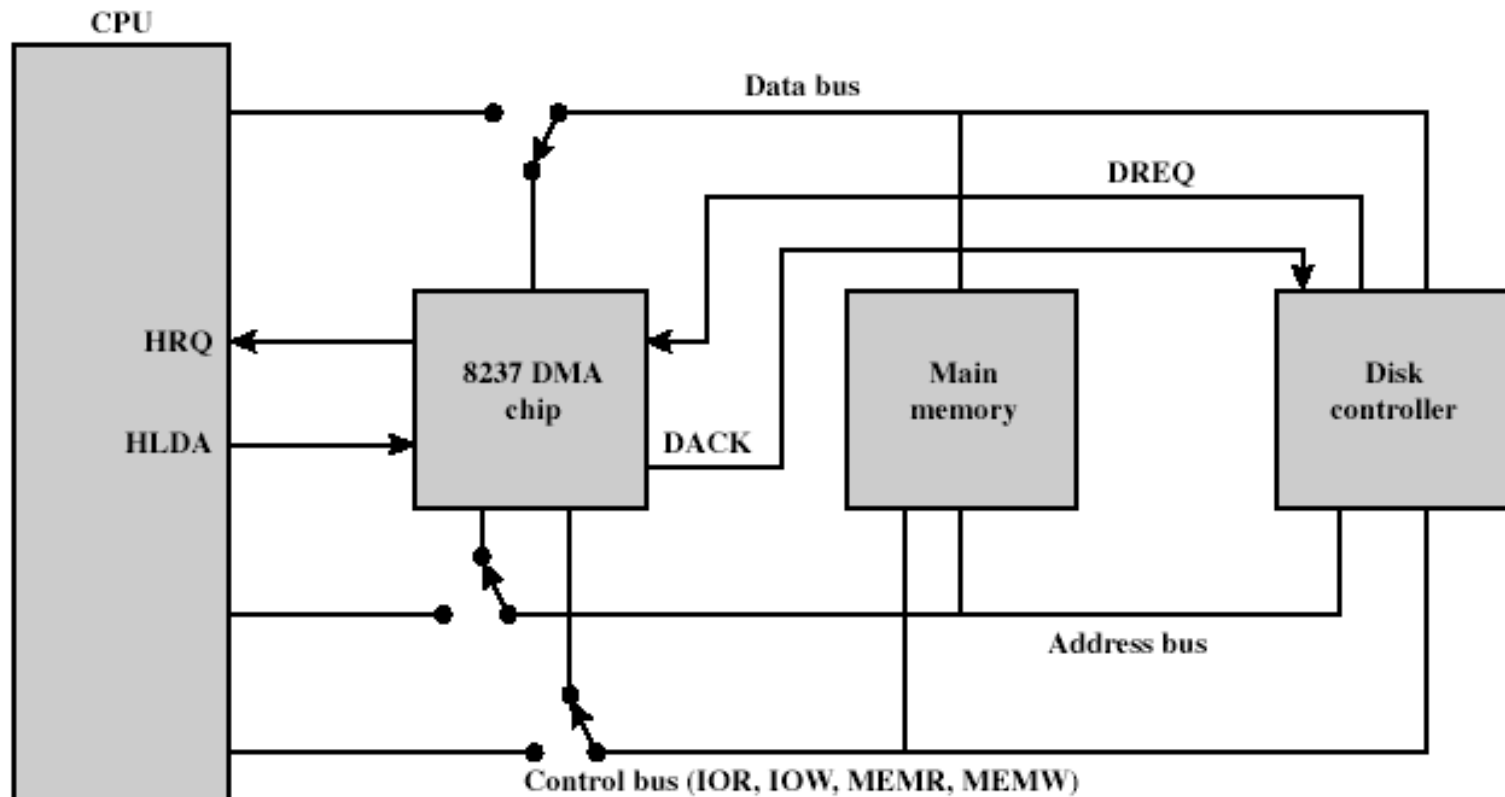
# **Etapas de una transf. DMA (3)**

---

## **Finalización de la transferencia**

- El DMAC libera el bus y devuelve el control a la CPU
- El DMAC suele activar una señal de interrupción para indicar a la CPU la finalización de la operación de E/S solicitada

# Bus del sistema y DMA



DACK = DMA acknowledge  
DREQ = DMA request  
HLDA = HOLD acknowledge  
HRQ = HOLD request

# Problema que puede haber

---

- Se puede degradar el rendimiento de la CPU si el DMAC hace uso intensivo del bus
  - Si el bus está ocupado en una transferencia DMA, la CPU no puede acceder a memoria para leer instrucc. / datos
- El problema se reduce con el uso de memoria cache
  - La mayor parte del tiempo, la CPU lee instruc. de la cache, por lo que no necesita usar el bus de memoria.
  - El DMAC puede aprovechar estos intervalos en los que la CPU está leyendo instrucciones de la cache (y por tanto no usa el bus de memoria) para realizar las transferencias.



# Problema que puede haber (2)

---

- En caso de computadores sin cache
  - El procesador no utiliza el bus en todas las fases de la ejecución de una instrucción.
  - El DMAC puede aprovechar las fases de ejecución de una instrucción en las que la CPU no utiliza el bus para realizar sus transferencias.

# Tipos de transferencias

---

- Si el DMAC sólo toma el control del bus durante los intervalos de tiempo en los que la CPU no hace uso del mismo **el rendimiento del sistema no sufrirá degradación alguna**
- Se distinguen dos tipos de transferencias:
  - Por ráfagas (burst)
  - Por robo de ciclo (cycle-stealing)

# DMA modo ráfaga

---

- El DMAC solicita el control del bus a la CPU
- Cuando la CPU concede el bus, el DMAC no lo libera hasta haber finalizado la transferencia de todo el bloque de datos completo.
- **VENTAJAS:**
  - La transferencia se realiza de forma rápida.
- **DESVENTAJAS:**
  - Durante el tiempo que dura la transferencia la CPU no puede utilizar el bus con memoria, lo que puede degradar el rendimiento del sistema.

# DMA modo robo de ciclo

---

- El DMAC solicita el control del bus a la CPU.
- Cuando la CPU concede el bus al DMAC, se realiza la transferencia de una única palabra y después el DMAC libera el bus.
- El DMAC solicita el control del bus tantas veces como sea necesario hasta finalizar la transferencia del bloque completo
- **VENTAJAS:**
  - No se degrada el rendimiento del sistema.
- **DESVENTAJAS:**
  - La transferencia tarda más tiempo en llevarse a cabo.

# DMA modo robo de ciclo (2)

---

- Para la CPU no es una interrupción.
  - El procesador no debe guardar el contexto.
- Si bien el trabajo de la CPU es lento, no será tanto como si ella realizara la transferencia.
- Por lo tanto, para transferencia de E/S de múltiples palabras, es la técnica más eficiente.

# Canales de E/S

---

Los dispositivos de E/S son cada vez más sofisticados

- Ej: tarjetas gráficas 3D.
- Evolución:
  1. La CPU controla directamente los periféricos.
  2. Se agrega un módulo de E/S o controlador.
  3. Idem 2 más llamado de interrupción.
  4. El módulo de E/S provee el acceso directo a memoria (DMA).
  5. El módulo de E/S tiene su propio procesador con su pequeño conjunto de instrucciones.
  6. El módulo además tiene su memoria local o sea se convierte en una computadora en sí mismo.

# Características de Canales de E/S

---

- Representan una extensión al concepto de DMA
  - Tienen la habilidad de ejecutar instrucciones de E/S
- Completo control de la transferencia de datos
  - por lo tanto la CPU no ejecuta instrucciones de E/S
- Programa almacenado en memoria principal
- La CPU inicia la transferencia de E/S
  - Ordena ejecutar el programa que está en memoria
  - El programa especifica dispositivos, áreas de memoria a usar, prioridades y acciones ante errores

# Tipos de canales de E/S (1)

---

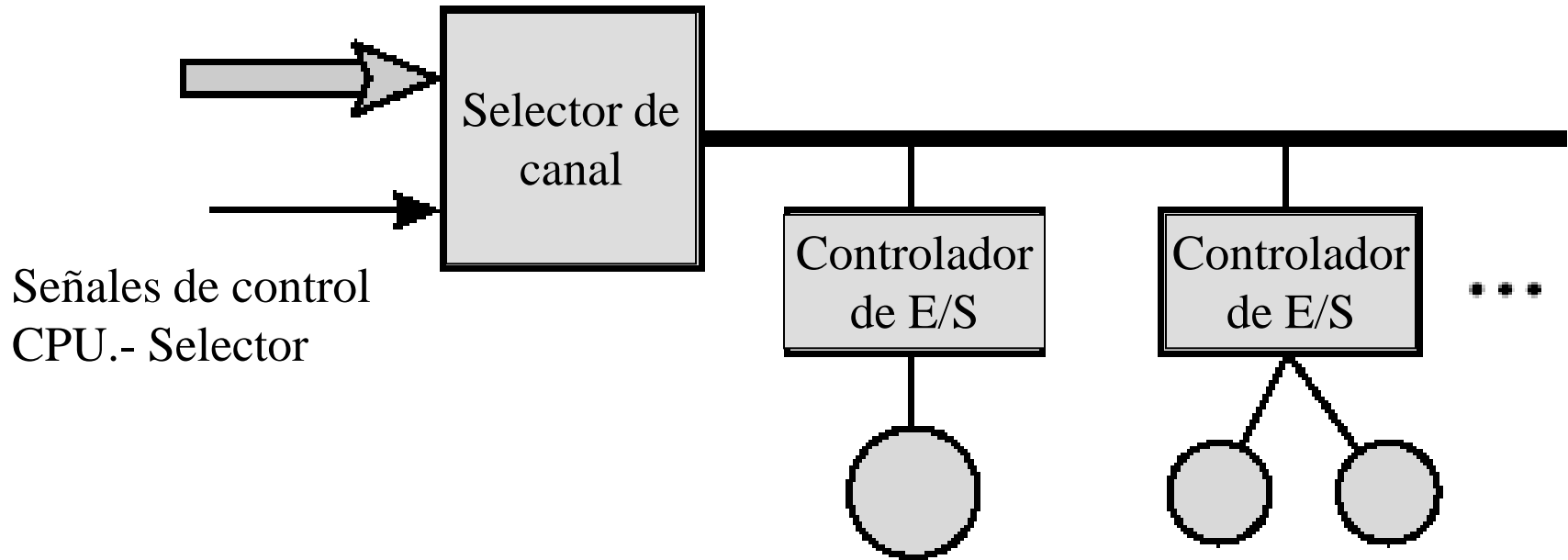
- Selector
  - Controla varios dispositivos de alta velocidad y uno por vez, por lo tanto el canal se dedica para la transferencia de datos de ese dispositivo.
  - El canal selecciona un dispositivo y efectúa la transferencia.
  - Los dispositivos son manejados por un controlador o módulo de E/S
  - Por lo tanto el canal de E/S ocupa el lugar de la CPU en el control de esos controladores.



# Selector

---

Datos y direcciones  
canal a mem.ppal.



# Tipos de canales de E/S (2)

---

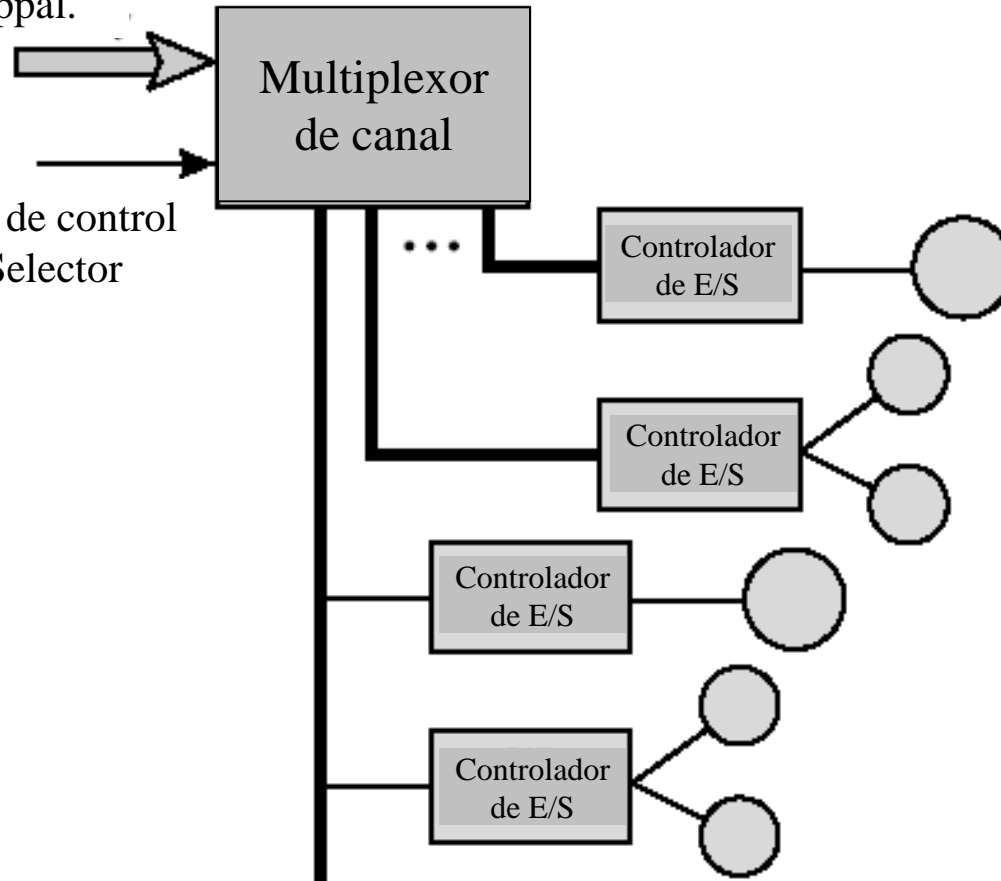
- Multiplexor
  - Puede manejar E/S con varios dispositivos a la vez.
  - Multiplexor de bytes:
    - Acepta y transmite caracteres.
  - Multiplexor de bloques:
    - Intercala bloques de datos desde distintos dispositivos.

# Multiplexor

---

Datos y direcciones  
canal a mem.ppal.

Señales de control  
CPU.- Selector



# Lecturas básicas

---

- Capítulo 6
  - W. Stallings, 5º Ed.
- <http://www.pcguide.com/ref/mbsys/res/irq/func.htm>
- <http://www.pcguide.com/>