

## Tema 4: SISTEMAS DE ENTRADA/SALIDA

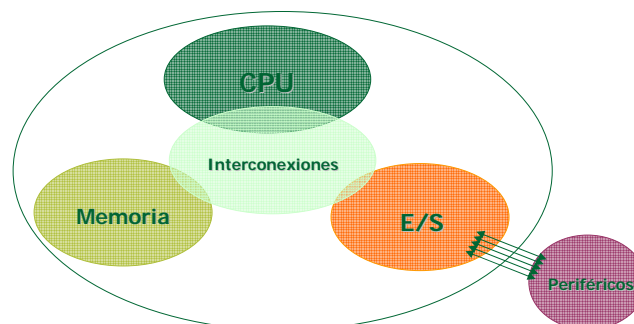
### ÍNDICE

1. Introducción.
2. Módulos de Entrada/Salida: comunicación CPU - Periféricos.
3. Mapa de Entrada/Salida: común y separada.
4. Introducción a los métodos de programación de E/S.
  - E/S programada.
  - E/S por interrupciones.
  - Acceso directo a memoria.
5. Canales y procesadores de E/S.
6. Ejemplo de dispositivo de E/S: puerto paralelo (*interfaz centronics*).

1

### Introducción (I)

- Un computador no puede estar formado sólo por la CPU y la memoria.
- Para darle alguna utilidad debe **comunicarse con el mundo exterior** (periféricos) a través del subsistema de entrada/salida.



1. Introducción

2

## Introducción (II)

La **misión principal** del subsistema de E/S es **adaptar los dispositivos externos** antes de conectarlos al bus del sistema.

### ¿Por qué no se conectan directamente al bus del sistema?

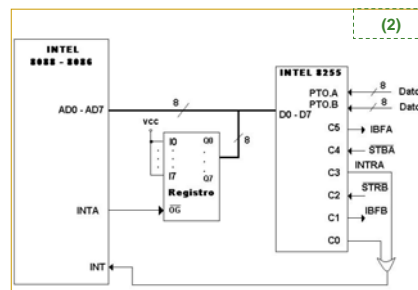
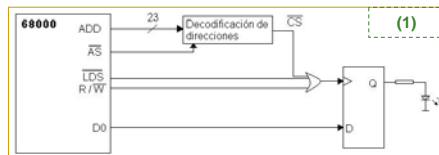
- La velocidad de transmisión de datos de los periféricos es siempre menor que la de la memoria y la CPU. Es necesario amortiguar esa diferencia de velocidades.
- Debido a la gran diversidad de periféricos no es posible incorporar toda la lógica necesaria en el computador para manejar cada uno de éstos.
- Los tamaños y formatos de datos de los periféricos son diferentes a los del resto del computador.

#### 1. Introducción

3

## Implementación del sistema de E/S

- Hay dos **formas de implementar la E/S** en un computador:
  1. Construir a medida el subsistema utilizando elementos básicos. (Ejemplo: *circuito para que al escribir un '1' en una dirección de memoria se ilumine un LED*).
  2. o bien incorporar y programar dispositivos estándares más complejos y genéricos (Ej. 8255).



#### 1. Introducción

4

## Funciones Básicas del sistema de E/S

- **Direcccionamiento:** Selección del dispositivo correspondiente de entre todos los dispositivos presentes en el sistema.
- **Sincronización:** Permitir que la CPU y la memoria (alta velocidad transferencia de datos) se puedan comunicar con los dispositivos de E/S (baja velocidad) sincronizando los envíos de datos entre ambos. (Inicio de la transferencia).
- **Transferencia:** El sistema de E/S debe tener toda la circuitería y señales de comunicación apropiadas para llevar a cabo la comunicación con cada uno de los dispositivos del sistema. (Método de transferencia).

### 1. Introducción

5

## Comunicación CPU - Periféricos

La E/S se implementa mediante periféricos.

- **Periférico:**
  - Elemento que permite la transferencia de información entre la CPU y el mundo exterior.
  - Interfaz que traduce la información asíncrona y analógica del mundo exterior a la información síncrona y codificada del computador.
  - Dos partes: **módulo de E/S (controlador) y dispositivo externo.**

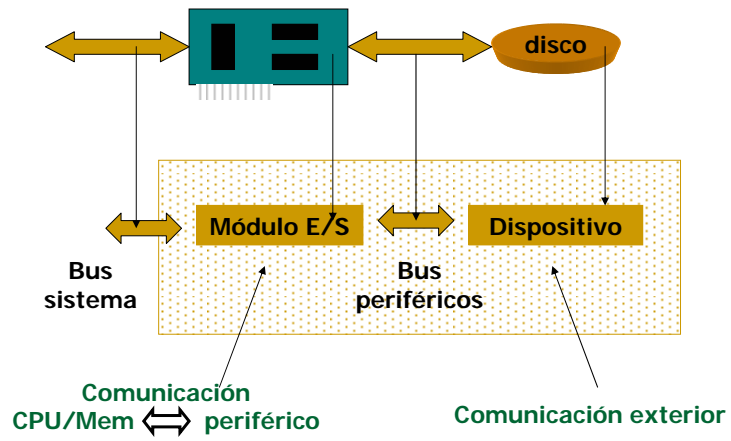


Fig. Periférico genérico

### 2. Periférico

6

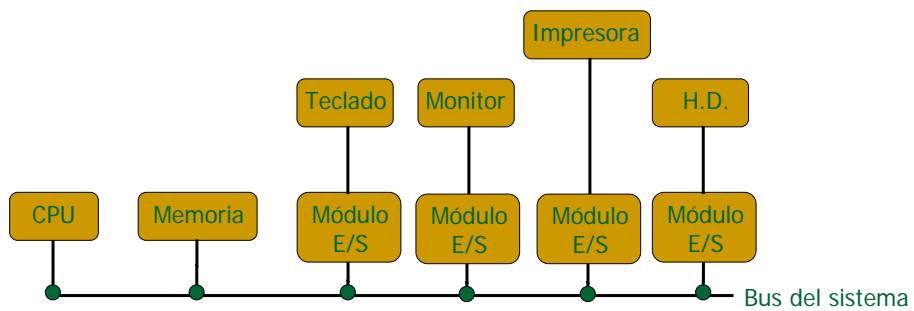
## Comunicación CPU - Periféricos



2. Periférico

7

## Comunicación CPU - Periféricos



2. Periférico

8

## Módulo de E/S

### Funciones

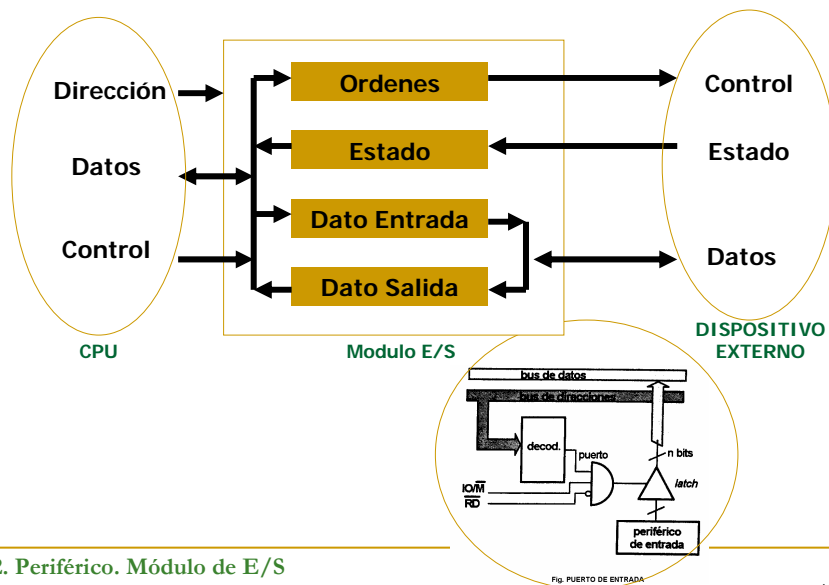
Elemento que permite la conexión de un dispositivo externo al bus del sistema.

- Control y Temporización:**
  - Coordina el tráfico entre dispositivos internos (memoria, CPU) y externos (impresora, monitor, ...).
- Comunicación con la CPU:**
  - Reconocer la dirección de la CPU que identifica al dispositivo externo
  - Recepción y decodificación de mandatos (comandos) desde la CPU
  - Transferencia de datos entre la CPU y el dispositivo externo
  - Información sobre el estado del periférico
- Comunicación con el/los dispositivo/s (órdenes, estado, datos).**
- Almacenamiento temporal de datos.**
- Detección de errores.**

2. Periférico. Módulo de E/S

9

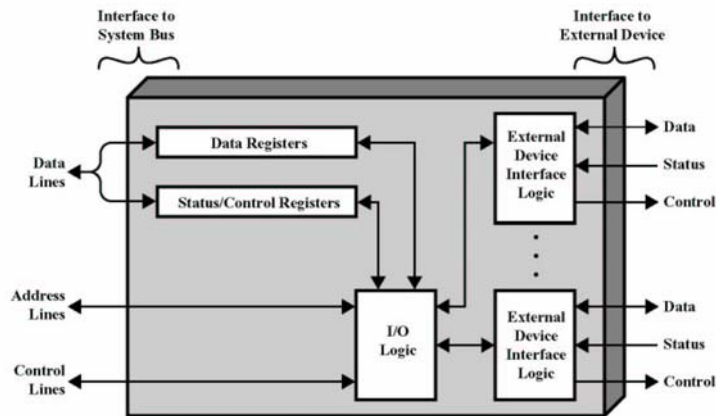
## Comunicación CPU – Dispositivo externo



2. Periférico. Módulo de E/S

10

## Comunicación CPU – Dispositivos externos

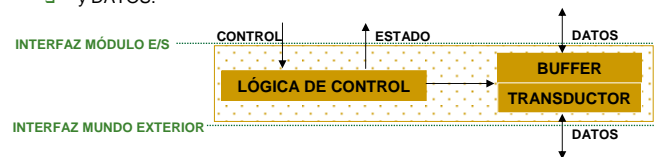


### 2. Periférico. Módulo de E/S

11

## Dispositivo Externo

- Elemento que permite la transferencia de información entre la CPU y el mundo exterior.
  - La forma de comunicación con el módulo de E/S se hace mediante:
    - SEÑALES DE CONTROL (mandatos/comandos enviados por la CPU),
    - ESTADO (información sobre el funcionamiento del dispositivo)
    - y DATOS.



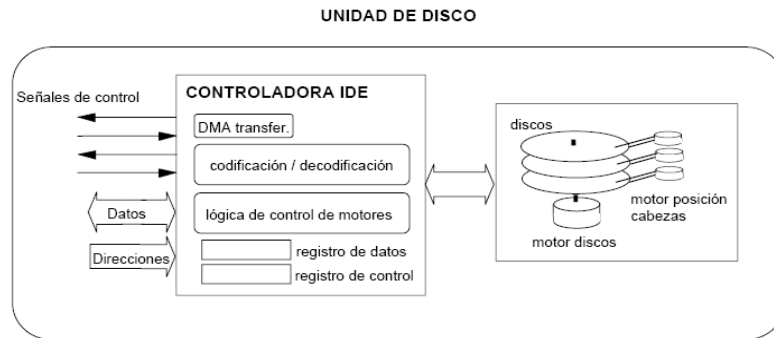
- Funciona a modo de interfaz, transformando la información asincrónica y analógica del mundo exterior a la información sincrónica y codificada del computador (transductor).
- Internamente suele haber una serie de *buffers* que son capaces de almacenar temporalmente datos o información adicional de contexto.

### 2. Periférico. Dispositivo externo

12

## Ejemplos de periféricos

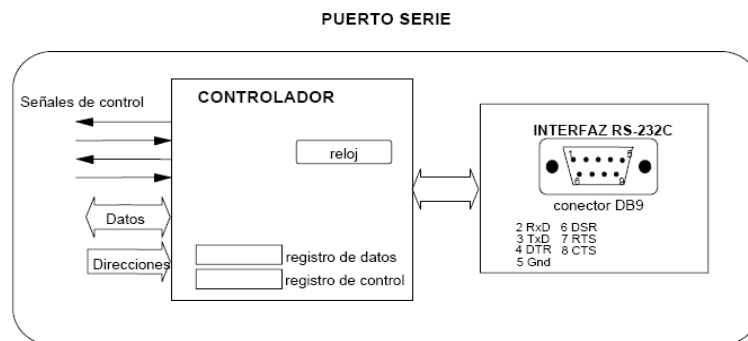
### Unidad de Disco



13

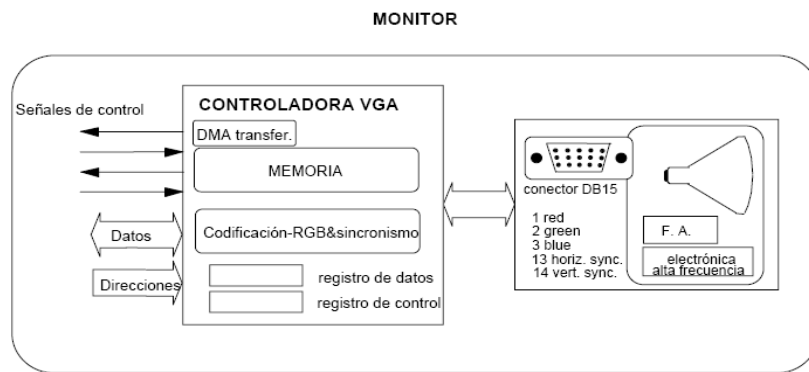
## Ejemplos de periféricos

### Puerto Serie



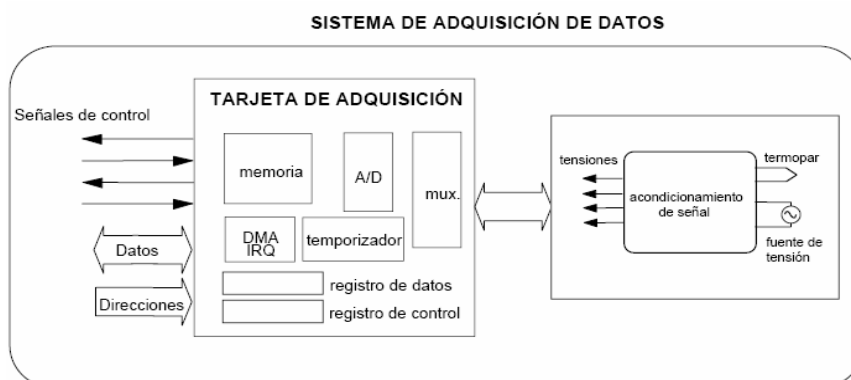
14

## Ejemplos de periféricos Monitor



15

## Ejemplos de periféricos Sistema de Adquisición de Datos



16

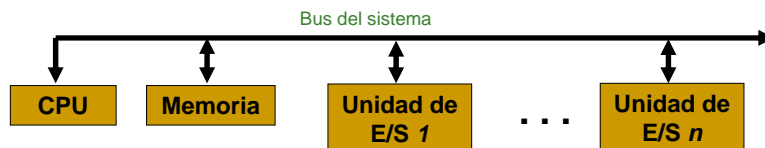


## MAPA DE E/S:

### E/S común y separada

Según el modo de seleccionar el periférico y el acceso a sus registros de control, datos y estado:

- **E/S común o asignada/mapeada en memoria**
  - El acceso a estos módulos se realiza de igual modo a como se accede a un dato de memoria principal.
  - Los periféricos se integran en el computador como si fueran parte de la memoria → Comunicarse con un módulo de E/S es leer y escribir en memoria. Ej: 68000.
  - VENTAJA: Se aprovecha la potencia del juego de instrucciones.
  - INCONVENIENTE: Se desperdicia parte del espacio de direcciones.



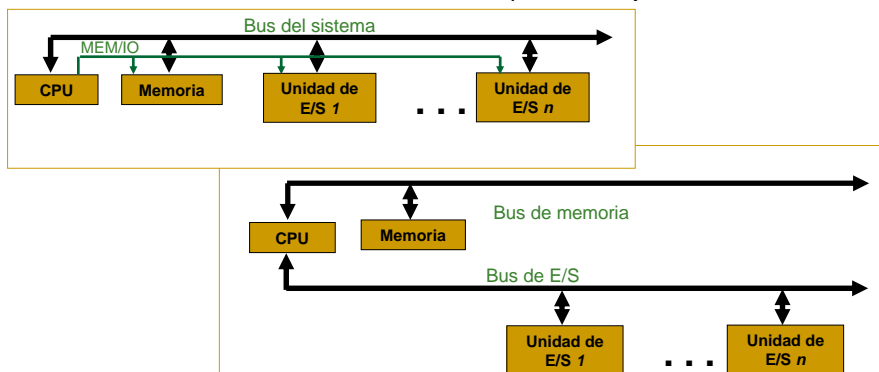
3. Mapa de E/S

17

## MAPA DE E/S:

### E/S común y separada

- **E/S aislada o separada**
  - El acceso a la E/S está contemplado en la arquitectura.
  - Existen dos mapas de memoria separados: uno para memoria y otro para E/S → Existen señales e instrucciones específicas. Ej: intel 80x86.



3. Mapa de E/S

18



## Métodos de programación de E/S.

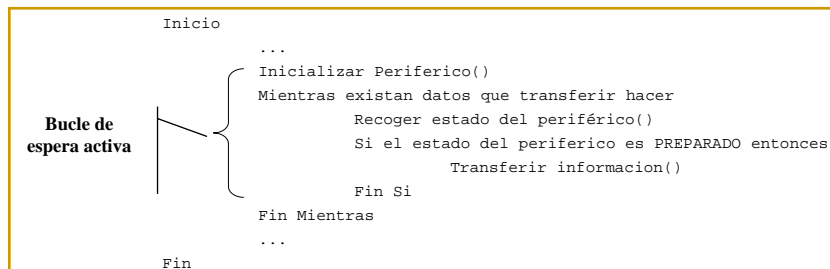
- E/S Programada (Encuesta o *pooling*).
- E/S por interrupciones.
- Acceso directo a memoria (DMA).

### 4. Métodos de programación de E/S

21

## E/S Programada (Encuesta o *pooling*)

- La **CPU** se dedica por completo a realizar la operación de E/S:
  - realiza la inicialización, la comprobación de estado y la transferencia.
- La **CPU** tiene el **control absoluto** de la operación de E/S.
- Ventaja: Hardware mínimo.
- Inconvenientes:
  - Malgasta tiempo de proceso (interroga continuamente al periférico).
  - Dificultad para atender varios periféricos.

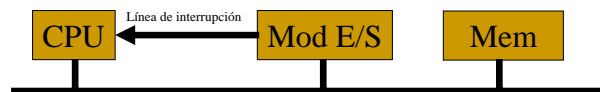


### 4. Métodos de programación de E/S. E/S PROGRAMADA

22

## E/S por interrupciones

- La E/S le indica a la CPU cuando está preparada para transferir datos (genera una interrupción a la CPU), activando una línea especial conectada a la CPU (**línea de interrupción**).

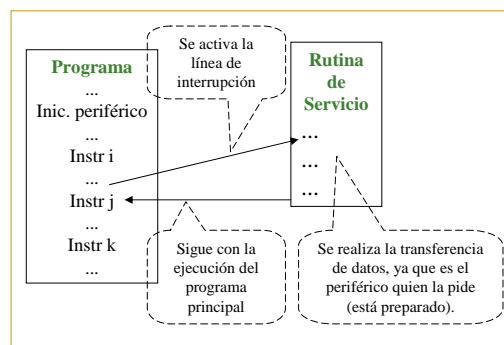


### 4. Métodos de programación de E/S. E/S POR INTERRUPCIONES

23

## Funcionamiento

- El procesador ejecuta instrucciones de un programa. Al finalizar cada instrucción comprueba si se ha producido una interrupción.
- En caso afirmativo se salva el estado actual del programa (contador del programa y registros) y se salta a ejecutar la **rutina de servicio** correspondiente.



- La rutina de servicio efectúa las operaciones apropiadas en la E/S para realizar la transferencia de datos solicitada.
- Al finalizar la rutina de servicio se recupera el estado de la CPU y se continúa ejecutando el programa que se estaba ejecutando antes de la interrupción.

### 4. Métodos de programación de E/S. E/S POR INTERRUPCIONES

24

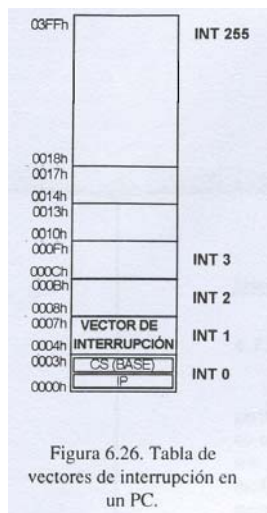
## Clasificación de Interrupciones

- Según FUENTE que produce la interrupción:
  - **Interrupciones HARDWARE**
    - INTERNAS (producidas por la CPU)
    - EXTERNAS (producidas por los dispositivos de E/S)
      - Vectorizadas
      - No vectorizadas
  - **Interrupciones SOFTWARE** (producidas por la ejecución de instrucciones de la CPU).
- Según MODO DE CONOCER el **VECTOR DE INTERRUPCIÓN** (la dirección donde se encuentra la rutina de servicio de la interrupción):
  - **Interrupciones AUTOVECTORIZADAS**: el vector de interrupción es FIJO.
  - **Interrupciones VECTORIZADAS**: el vector de interrupción lo suministra el propio periférico

### 4. Métodos de programación de E/S. E/S POR INTERRUPCIONES

25

## Interrupciones Vectorizadas



- Con el vector de interrupción se indexa una tabla de vectores en memoria principal para obtener la dirección de la rutina de servicio a la que hay que saltar.

Figura 6.26. Tabla de vectores de interrupción en un PC.

### 4. Métodos de programación de E/S. E/S POR INTERRUPCIONES

26

## Conexión de varios periféricos

- Generalmente existen **VARIOS PERIFÉRICOS** conectados (y no uno sólo) que pueden realizar interrupciones.
- Es necesario:
  - Decidir cómo se conectan los periféricos a la CPU.
    - Una sola línea de interrupción.
    - Varias líneas de interrupción.
  - Establecer prioridades.
  - Determinar para cada periférico su vector de interrupciones.

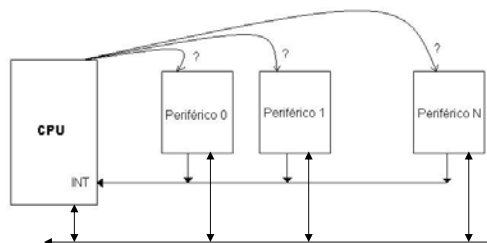
### 4. Métodos de programación de E/S. E/S POR INTERRUPCIONES

27

## Una sola línea de interrupción

- Todos los periféricos interrumpen por la misma línea.
- El vector de interrupción es fijo y común a todos los periféricos.
- Mediante *polling* (Consulta software del Registro de Estado) la CPU identifica el periférico y desactiva la interrupción.
- La prioridad viene determinada por el orden de la encuesta.

Inconveniente:  
Tiempo consumido  
en la consulta software.

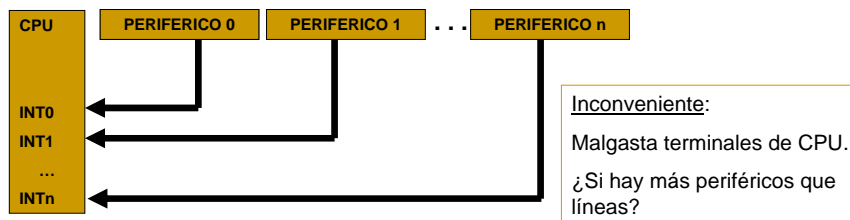


### 4. Métodos de programación de E/S. E/S POR INTERRUPCIONES

28

## Varias líneas de interrupción

- Cada periférico tiene su línea de interrupción.
- Cada línea tiene su propio vector de interrupción asociado.
- La CPU determina la prioridad.

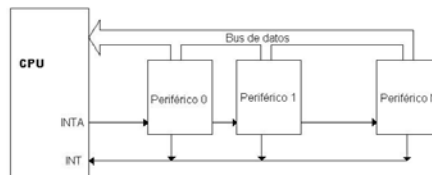


### 4. Métodos de programación de E/S. E/S POR INTERRUPCIONES

29

## Daisy-chain o conexión en cadena

- Una única línea INT para todos los periféricos.
- El vector de interrupciones es fijo.
- La CPU envía la señal INTA, como reconocimiento de la interrupción.
- Si el periférico que recibe la señal INTA, no generó la interrupción, entonces la transmite al siguiente elemento de la cadena.
- La señal INTA se propaga de un periférico a otro y el periférico de mayor prioridad desactiva la petición de interrupción INT.
- La prioridad viene determinada por la posición del periférico en la cadena.

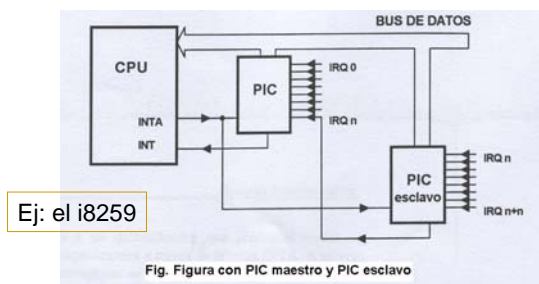


### 4. Métodos de programación de E/S. E/S POR INTERRUPCIONES

30

## Controlador de interrupciones (PIC, *programmable interrupt controller*)

- Permite conectar varios periféricos a una misma línea de interrupciones.
  - Enmascarar interrupciones por software
  - Definir vectores de interrupciones asociados a cada interrupción
  - Gestionar prioridades.



### 4. Métodos de programación de E/S. E/S POR INTERRUPTIONES

31

Num.	Dirección	Contenido
00	000 - 003	CPU: División entre cero
01	004 - 007	CPU: Un sólo paso
02	008 - 009	CPU: NMI (Error en elemento RAM)
03	00C - 00F	CPU: Se ha llegado al punto de interrupción (breakpoint)
04	010 - 013	CPU: Desbordamiento numérico
05	014 - 017	Hard copy (copia permanente, salida impresa)
06	018 - 01B	Instrucción desconocida (sólo 80286)
07	01D - 01F	Reservado
08	020 - 023	IRQ0: Timer (llamada 18,2 vez/seg.)
09	024 - 027	IRQ1: Teclado
0A	028 - 02B	IRQ2: Segundo 8259 (sólo AT)
0B	02C - 02F	IRQ3: Puerto de conexión en serie 2
0C	030 - 033	IRQ4: Puerto de conexión en serie 1
0D	034 - 037	IRQ5: Disco duro
0E	038 - 03B	IRQ6: Disquete
0F	03C - 03F	IRQ7: Impresora
10	040 - 043	BIOS: Funciones vídeo
11	044 - 047	BIOS: Averiguar configuración
12	048 - 04B	BIOS: Averiguar tamaño de la memoria RAM
13	04C - 04F	BIOS: Funciones disquetes/discos duros
14	050 - 053	BIOS: Acceso a puerto de conexión en serie
15	054 - 057	BIOS: Funciones casete/amplicadas
16	058 - 05B	BIOS: Preguntas teclado
17	05C - 05F	BIOS: Acceso a impresora en paralelo
18	060 - 063	Llamada del ROM-BASIC
19	064 - 067	BIOS: Botar sistema (Alt+Ctrl+Del)
1A	068 - 06B	BIOS: Preguntar Hora/Fecha
1B	06C - 06F	Tecla Break (no Ctrl-C) pulsada
1C	070 - 073	Se llamará 08 después de cada INT
1D	074 - 077	Dirección de la tabla de parámetros vídeo
1E	078 - 07B	Dirección de la tabla de parámetros disquete
1F	07C - 07F	Dirección del carácter bit muestra
20	080 - 083	DOS: Finalizar el programa
21	084 - 087	DOS: Llamar función DOS
22	088 - 08B	Dirección rutina final de programa DOS
23	08C - 08F	Dirección rutina Ctrl-Break del DOS
24	090 - 093	Dirección de la rutina de error del DOS
25	094 - 097	DOS: Leer disquete/disco duro
26	098 - 09B	DOS: Escribir disquete/disco duro
27	09C - 09F	DOS: Finalizar prog., quedar residente
28	0A0 - 0A3	DOS: El DOS está desocupado
29	0A4 -	DOS: Reservado
2E		0BB
2F	0BC - 0BF	DOS: Multiplexor
30	0C0 -	DOS: Reservado
32	0CB	
33	0CC - 0CF	Funciones del controlador del ratón
34	0D0 -	DOS: Reservado
40	0FF	
41	104 - 107	Dirección de la tabla del disco duro 1
42	108 -	Reservado
45	117	
46	118 - 11B	Dirección de la tabla del disco duro 2
47	11C -	Los programas de aplicación pueden
49	127	ocuparlos como quieran
4A	128 - 12B	Llegado tiempo de alarma (sólo AT)
4B	12C - libre:	Los programas de aplicación pueden
58	16F	ocuparlos como quieran
5C	170 - 173	Funciones del NETBIOS
5D	174 -	libre: Los programas de aplicación pueden
66	19B	ocuparlos como quieran
67	19C - 19F	Funciones administrador de la memoria EMS
68	1A0 -	libre: Los programas de aplicación pueden
6F	1BF	ocuparlos como quieran
70	1C0 - 1C3	IRQ08: Reloj de tiempo real (sólo AT)
71	1C4 - 1C7	IRQ09: (sólo AT)
72	1C8 - 1CB	IRQ10: (sólo AT)
73	1CC - 1CF	IRQ11: (sólo AT)
74	1D0 - 1D3	IRQ12: (sólo AT)
75	1D4 - 1D7	IRQ13: 80287 NMI (sólo AT)
76	1D8 - 1DB	IRQ14: Disco duro (sólo AT)
77	1DC - 1DF	IRQ15: (sólo AT)
78	1E0 -	No se utilizan
7F	1FF	
80	200 -	Se utilizan dentro del
F0	3C3	interprete Basic
F1	3C4 -	
FF	3CF	No se utilizan

32



## Acceso Directo a Memoria (DMA)

### Ejemplo de E/S con periférico lento

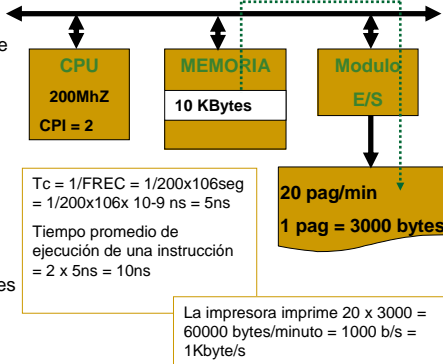
Imprimir un archivo de 10KB en una impresora láser de 20 páginas por minuto.

#### E/S programada

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

#### E/S por interrupciones

- La impresora genera una interrupción cada vez que está preparada.
- Si la Rutina de Interrupción tiene 10 instrucciones (salvar contexto, comprobar estado, transferir byte, restaurar contexto, rti), para transferir 10 Kbytes se ejecuta  $10.000 \times 10$  instrucciones.
- La CPU está ocupada **0,001 s con la operación de E/S**.



#### 4. Métodos de programación de E/S. DMA

33

## Acceso Directo a Memoria (DMA)

### Ejemplo de E/S con periférico rápido

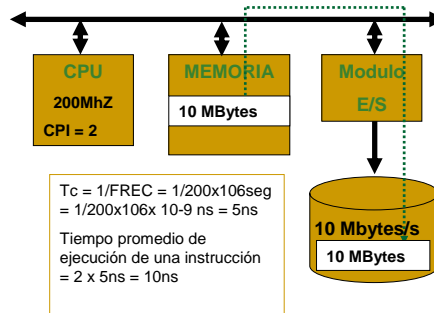
Transferir un archivo de 10MB de la memoria a disco.

#### E/S programada

- La CPU entra en un bucle y envía un nuevo byte cada vez que el disco está preparado.
- El disco tarda **1s** en recibir 10 Mbytes.
- La CPU está ocupada **1 s con la operación de E/S** (la CPU podría haber ejecutado 100 millones de instrucciones).

#### E/S por interrupciones

- El disco genera una interrupción cada vez que está preparado.
- Si la Rutina de Interrupción tiene 10 instrucciones (salvar contexto, comprobar estado, transferir byte, restaurar contexto, rti), para transferir 10Mbytes se ejecuta  $10.000000 \times 10$  instrucciones.
- La CPU está ocupada **1 s con la operación de E/S**.



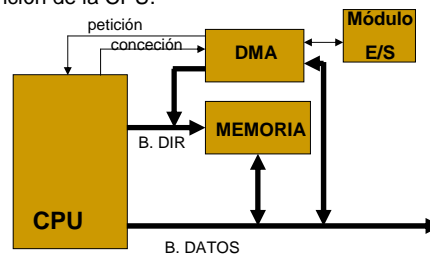
#### 4. Métodos de programación de E/S. DMA

34

## Acceso Directo a Memoria (DMA)

### Motivación y Definición

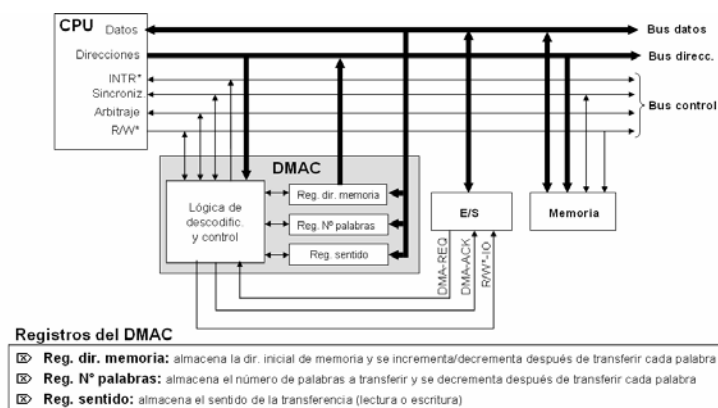
- La E/S programada y la E/S por interrupción tienen un **impacto negativo** sobre la **actividad del procesador** y la **velocidad de E/S**:
  - E/S programada: La CPU transfiere datos a alta velocidad al precio de no hacer nada.
  - E/S por interrupciones: Libera en parte al procesador a expensas de reducir la velocidad de E/S (sobrecarga por el cambio de contexto).
- **DMA (Direct Memory Access): LIMITA AL MÁXIMO LA INTERVENCIÓN DE LA CPU.**
  - Es un dispositivo capaz de controlar una transferencia de datos entre un periférico y memoria sin intervención de la CPU.



#### 4. Métodos de programación de E/S. DMA

35

## Estructura hardware de un DMAC



#### Registros del DMAC

- ☒ **Reg. dir. memoria:** almacena la dir. inicial de memoria y se incrementa/decrementa después de transferir cada palabra
- ☒ **Reg. N° palabras:** almacena el número de palabras a transferir y se decrementa después de transferir cada palabra
- ☒ **Reg. sentido:** almacena el sentido de la transferencia (lectura o escritura)

#### 4. Métodos de programación de E/S. DMA

36

## Acceso Directo a Memoria

### Funcionamiento (I)

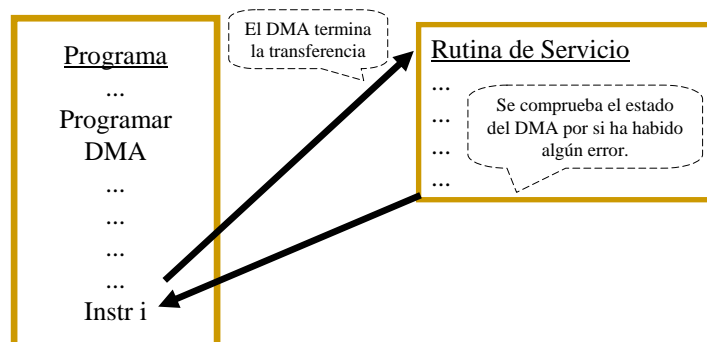
- Inicialización de transferencia:** la CPU envía los parámetros de la transferencia:
  - inicializa interfaz del periférico: nº de bytes a transferir, tipo de transferencia (R/W)...
  - inicializa DMA: nº bytes/palabras, tipo de transferencia (R/W), dirección de memoria inicial para transferencia; nº de canal, si DMA tiene más de un canal...
- La CPU retorna a su tarea** y se olvida de la evolución de la transferencia.
- Se realiza la transferencia:**
  - Cuando el periférico está preparado se lo avisa a DMAC.
  - El DMAC pide el control del bus: actúa como **master del bus** durante la transferencia,
  - especifica la dirección de memoria sobre la que realiza la transferencia,
  - y realiza transferencia entre periférico y memoria. Genera las señales de control:
    - Tipo de operación (R/W).
    - Señales de sincronización de la transferencia.
- Finaliza la transferencia:**
  - DMAC libera el bus
  - y devuelve el control a la CPU (generalmente con una interrupción)

#### 4. Métodos de programación de E/S. Funcionamiento DMA

37

## Acceso Directo a Memoria

### Funcionamiento (II)



#### 4. Métodos de programación de E/S. Funcionamiento DMA

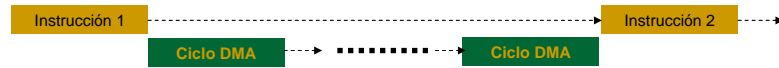
38

## Acceso Directo a Memoria

### Control del bus

#### Ráfagas:

- El DMA toma control del bus y no lo libera hasta transmitir un bloque de datos completo.
- Mayor velocidad de transferencia pero puede dejar sin actividad a la CPU durante periodos grandes de tiempo.



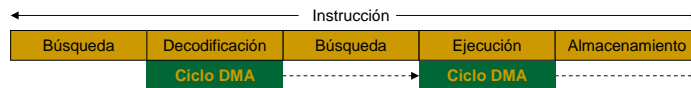
#### Robo de ciclos (más usual):

- EL DMA toma control del bus y lo retiene durante un solo ciclo. (Transmite una palabra y libera el bus).



#### Transparente:

- El DMA accede al bus solo en los ciclos en los que la CPU no lo utiliza. (En diferentes fases de ejecución de las instrucciones).
- La ejecución del programa no se ve afectada en su velocidad de ejecución.



#### 4. Métodos de programación de E/S. DMA. Control del bus.

39

## Acceso Directo a Memoria

### Control del bus. Ejemplo de Robo de Ciclos

Solución mas económica.

El DMA cada vez que quiere tomar el control del bus del sistema para realizar la transferencia de un dato, lo solicita a la CPU mediante la señal HOLD.

La CPU concede el control del bus mediante la señal HLDA. Después de realizar la transferencia el controlador de DMA lo comunica a la CPU mediante las señales de control correspondientes.

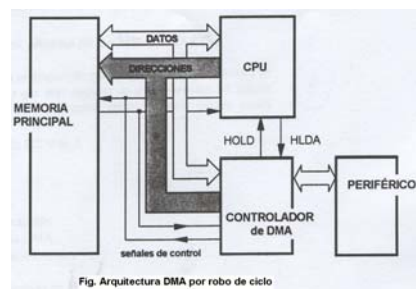


Fig. Arquitectura DMA por robo de ciclo

#### 4. Métodos de programación de E/S. DMA. Control del bus.

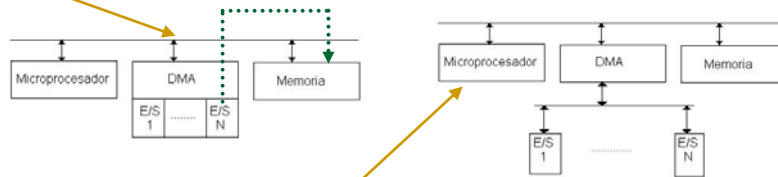
40

## Conexión del DMA al sistema

1. **Bus único, DMA independiente:** Actúa como una CPU de E/S. Necesita un ciclo para acceder al módulo de E/S o periférico y otro para acceder a memoria.



2. **Bus único, DMA y E/S integradas:** alternativa para reducir un ciclo de bus en la transferencias.



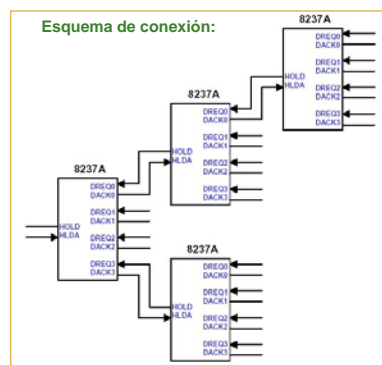
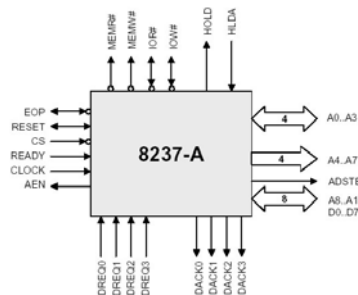
3. **Bus de E/S:** es una variación de la anterior que permite hacer la arquitectura más escalable.

## 4. Métodos de programación de E/S. DMA

41

## Ejemplo de DMAC: 8237

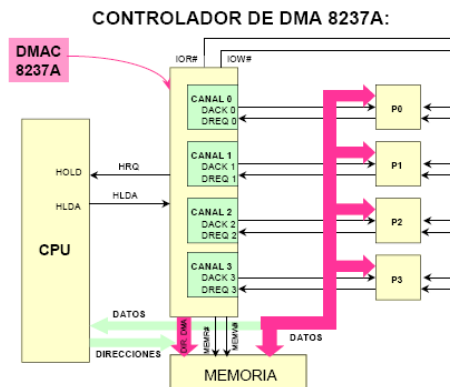
- Ejemplo de DMA: **i8237**. Posee 4 "canales" (procesadores de DMA) programables con tres modos diferentes y además se puede poner en cascada con otros i8237.



## 4. Métodos de programación de E/S. DMA

42

## Ejemplo de DMAC: Conexión CPU - 8237



### 4. Métodos de programación de E/S. DMA

43

## Evolución de la E/S

- La CPU controla directamente los periféricos.
- Se agrega un **módulo de E/S o controlador**.
- Aparece la gestión de **E/S por interrupción**.
- El módulo de E/S provee el acceso directo a memoria (**DMA**).
- El módulo de E/S tiene su propio procesador con su pequeño conjunto de instrucciones (**canal de E/S**).
- El módulo además tiene su memoria local o sea se convierte en una computadora en sí mismo (**procesador de E/S**).

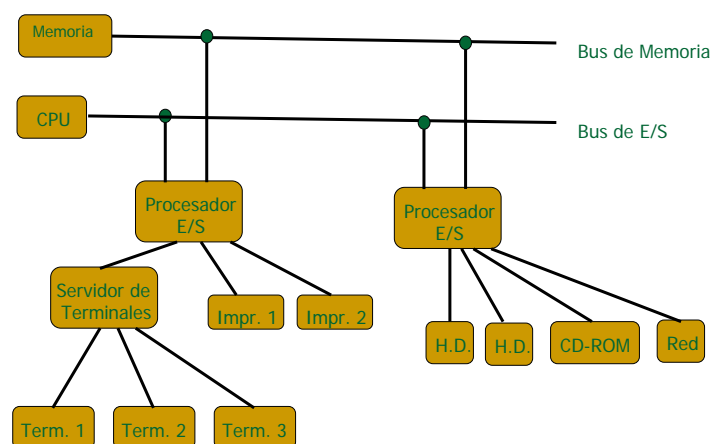
### 5. Canales y procesadores de E/S.

44

## Características de los canales de E/S

- Los canales representan una extensión al concepto 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.
- Instrucciones almacenadas en memoria principal que serán ejecutadas por un procesador especial en el canal.
- La CPU inicia la transferencia de E/S instruyendo al canal para ejecutar el programa que está en memoria.
- Este programa especifica dispositivos áreas de memoria a usar, prioridades y acciones ante errores.
- El canal siguiendo las instrucciones controla la transferencia de datos.

## Canales y procesadores de E/S

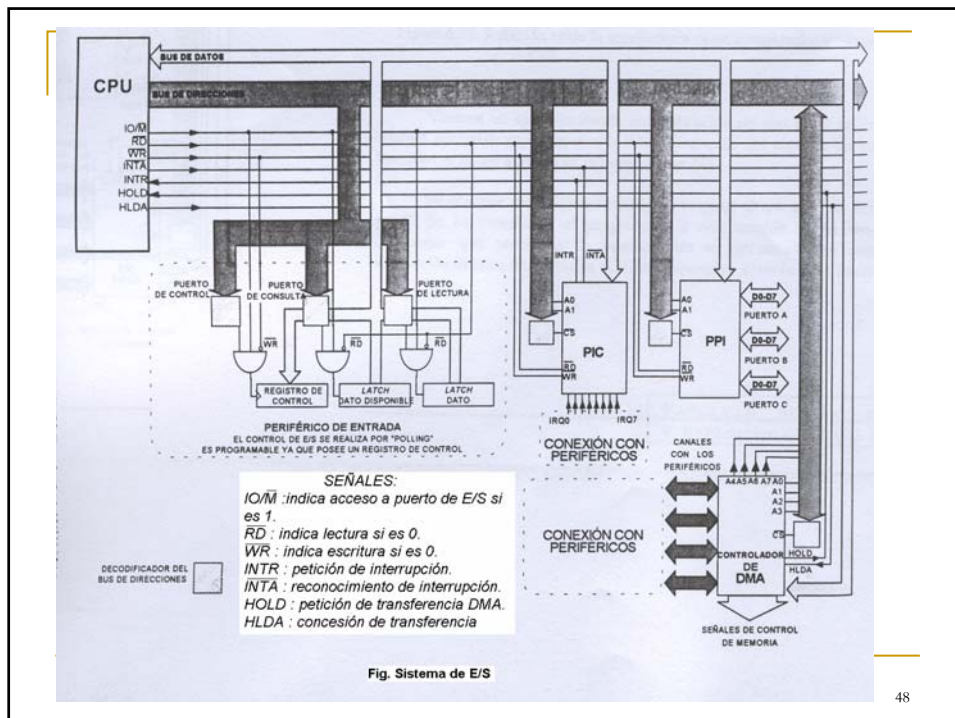


## Funcionamiento

- Para realizar una transferencia de E/S, la CPU primero ha de indicar qué canal de E/S ejecuta un determinado programa.
- La CPU también debe definir el área de almacenamiento temporal, establecer una prioridad y establecer las correspondientes acciones en caso de error. El programa a ejecutar está cargado en memoria principal y puede contener instrucciones propias sólo procesables por el canal de E/S.
- Después de terminar la operación de E/S, el canal de E/S deja el resultado en un área de memoria y a continuación genera una interrupción para indicar que ha acabado.

### 5. Canales y procesadores de E/S.

47



48



## Ejemplo de dispositivo de E/S.

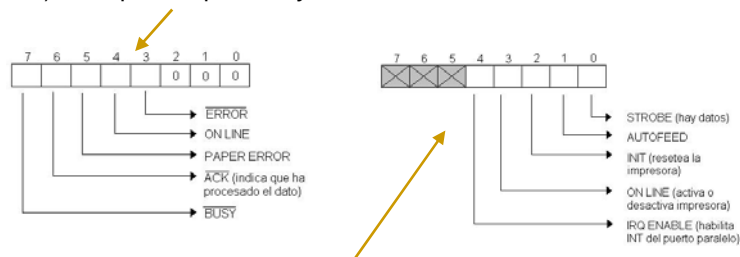
### Puerto paralelo (interfaz *centronics*).

- Es un controlador de E/S que está pensado, en principio, para ser usado solamente por impresoras.
- Para ello habría que utilizar un cable adaptado al bus de puerto paralelo diferente al *Centronics*, que es el que se utiliza habitualmente.
- Un PC tiene dos puertos paralelos llamados LPT1 y LPT2 a partir de las direcciones o puertos de E/S 378 (LPT1) y 278 (LPT2).

**NOTA:** ¡¡ no confundir puerto de E/S (dirección) con puerto hardware (puerto serie, puerto paralelo, ...) !!

### Puerto paralelo (interfaz *centronics*).

- Registro de datos: es de sólo escritura. Direcciones 378 (LPT1) y 278 (LPT2). Compuesto por un *byte*.
- Registro de estado: es de sólo lectura. Direcciones 379 (LPT1) y 279 (LPT2). Compuesto por un *byte*.



- Registro de control: es de lectura/escritura. Direcciones 37A (LPT1) y 27A (LPT2). Compuesto por un *byte*.