

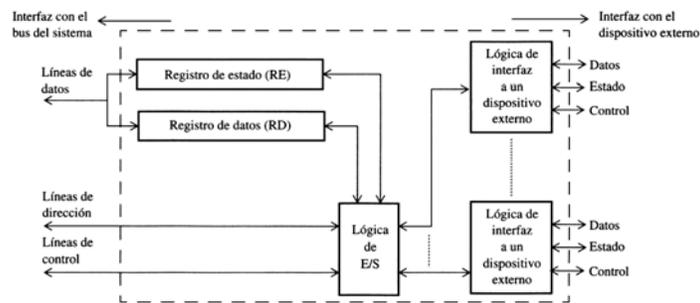
## UNIDAD DE ENTRADA SALIDA

### Conceptos

**Unidad de entrada-salida (E/S):** Elemento que proporciona un método de comunicación eficaz entre el sistema central y el periférico.

- Funciones** {
- Control y temporización
  - Comunicación con la CPU
  - Comunicación con el dispositivo externo
  - Almacén temporal de datos
  - Detección de errores

### Estructura del controlador E/S



- Modos de interconexión E/S ↔ CPU** {
- Buses independientes → *E/S aislada*
  - Bus común pero líneas de control separadas
  - Bus único → *E/S localizada en memoria*

- Técnicas de interacción E/S ↔ Sistema** {
- E/S controlada por programa
  - E/S por interrupciones
  - Acceso directo a memoria (DMA)

### E/S controlada por programa

La CPU está pendiente en todo momento (sondeo constante mediante bucles) del estado de la E/S.

- Inconvenientes** {
- Pérdida de tiempo en bucle de espera
  - Algunos programas no permiten estar en bucle
  - Si varios periféricos ⇒ muchos problemas

## E/S controlada por interrupciones

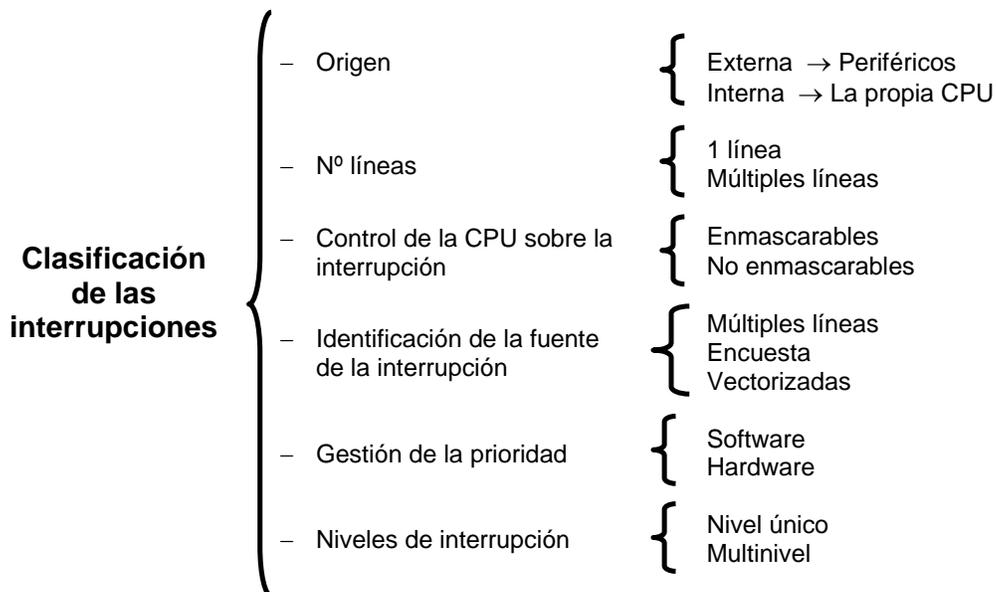
Elimina bucles de sondeo, de forma que entre CPU y unidad de E/S existe una línea de comunicación **HARDWARE** que provoca que cuando un periférico precisa de la atención de la CPU la unidad de E/S envía un nivel o flanco de tensión que provoca una interrupción en la CPU. Esta deja de hacer lo que estaba haciendo y salta a una subrutina (**vector interrupción**) de respuesta a dicho periférico.

A fin de cuentas tiene exactamente la misma función que una llamada a una subrutina, pero en lugar de efectuar dicha llamada vía instrucción recogida en el código del programa, dicha llamada se efectúa al activar un nivel de tensión en una determinada patilla.

Si se analiza el código del programa almacenado en la memoria de una aplicación se puede ver que están las dos partes, la correspondiente a la aplicación y a la interrupción, pero entre ambos códigos no se ve una relación software.

### Tratamiento de una interrupción:

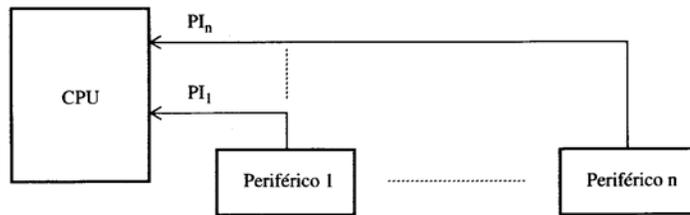
- 1.- Activar el sistema de interrupciones, validando las que sea necesario.
- 2.- Periférico activa la interrupción (PI).
- 3.- CPU termina la instrucción en curso, salva el PC (Contador de programa) y algunos de los otros registros usados por el programa.
- 4.- CPU inhibe interrupciones y carga en el PC el vector interrupción (dirección del programa de respuesta determinada por el fabricante del microprocesador), lo que implica que se salta a ejecutar la respuesta a la interrupción.
- 5.- Se informa al periférico del fin de respuesta para que desactive PI.
- 6.- Se reactiva el sistema de interrupciones.
- 7.- Se recupera el PC y los registros ⇒ se continua con la ejecución del programa interrumpido.



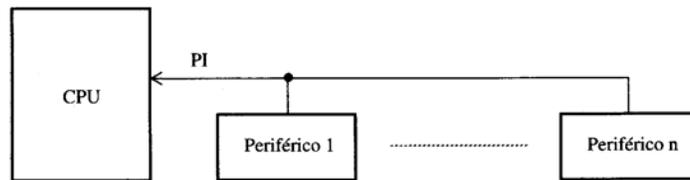
### Número de líneas de interrupción

De acuerdo a la identificación de la interrupción existen dos métodos:

- **Múltiples líneas:** En cuyo caso cada línea tiene asignado un **vector interrupción** (dirección de respuesta a cada línea de interrupción).
- **Línea única:** En cuyo caso para determinar la dirección de salto se puede utilizar el sistema de **encuesta** (pregunta mediante programa a todos y cada uno de los periféricos si ha sido él el productor de dicha interrupción) o **vectorizada** (cada periférico cuando genera una interrupción entrega a través del bus correspondiente la dirección del vector interrupción, de forma que la CPU recibe el aviso de interrupción e inmediatamente recoge del bus el vector interrupción).



a) La CPU posee varias entradas de interrupciones



b) La CPU posee una única entrada de interrupción

Identificación de la fuente de interrupción

### Control de la CPU sobre la interrupción

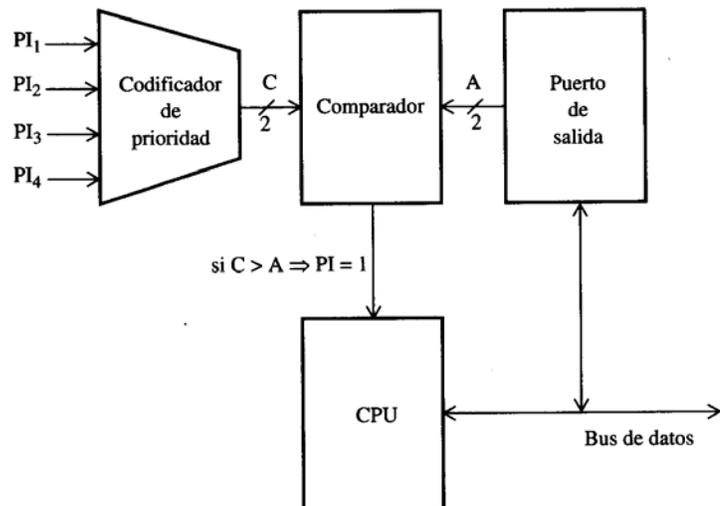
**Enmascarar:** forma de activar o desactivar de manera selectiva las interrupciones.

Procedimientos:

- Enmascaramiento individual.
- Enmascaramiento por nivel.

#### Enmascaramiento por nivel:

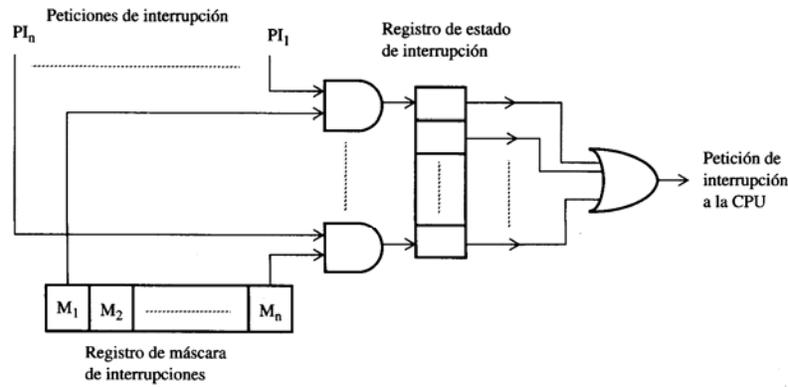
Las peticiones de interrupción se ordenan atendiendo a un criterio de prioridad. La CPU fija un nivel que puede ser modificado por programa. De forma que todas las interrupciones que estén en un nivel superior al especificado pueden ser atendidas, mientras que las que se encuentren en un nivel igual o inferior quedan prohibidas.



Enmascaramiento de interrupciones por nivel

**Enmascaramiento individual:**

A cada interrupción se le reserva un bit de un **registro de máscara**, de forma que se autorizan las interrupciones cuyo bit del registro de máscaras es "1" y se desautorizan las que tengan su bit de máscara a "0". Se proporciona un **registro de estados** que posee tantos bits como interrupciones existan, de forma que el valor que toma cada uno de estos bits es "1" si la interrupción es autorizada por el registro de máscaras y un "0" en caso contrario.



Enmascaramiento individual de cada entrada de interrupción

**Identificación de la fuente de interrupción y gestión de su prioridad**

Existen tres métodos de identificar la fuente de la interrupción:

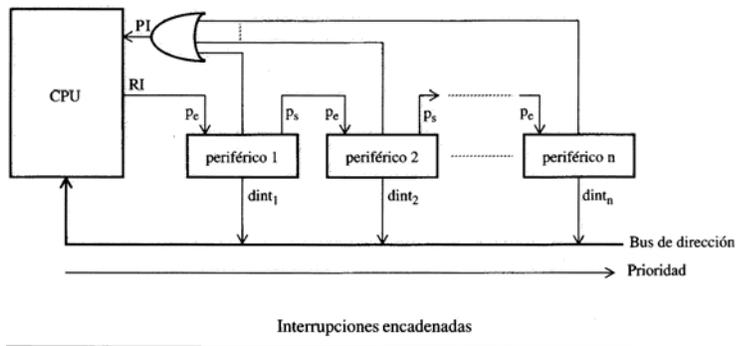
- Múltiples líneas
- Encuesta
- Vectorizadas

**Mediante encuesta:**

Consiste en realizar un programa que cuando detecte la producción de una interrupción pregunte a todos los periféricos si han sido el generador de la misma.

**Mediante encadenamiento:**

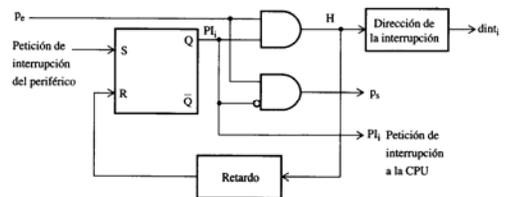
Daisy Chain consiste en conectar en serie todos los periféricos que solicitan la interrupción, de forma que los periféricos con más prioridad se colocan más próximos a la CPU



Interrupciones encadenadas

$P_e$	$PI_i$	$P_s$	H	Explicación
0	0	0	0	La señal de prioridad $p_e$ ha sido bloqueada ya por un periférico con mayor prioridad
0	1	0	0	
1	0	1	0	Propaga la señal de prioridad $p_s = p_e = 1$
1	1	0	1	Habilita $dint_i$ ( $h = 1$ ) y bloquea $p_s$ ( $p_s = 0$ )

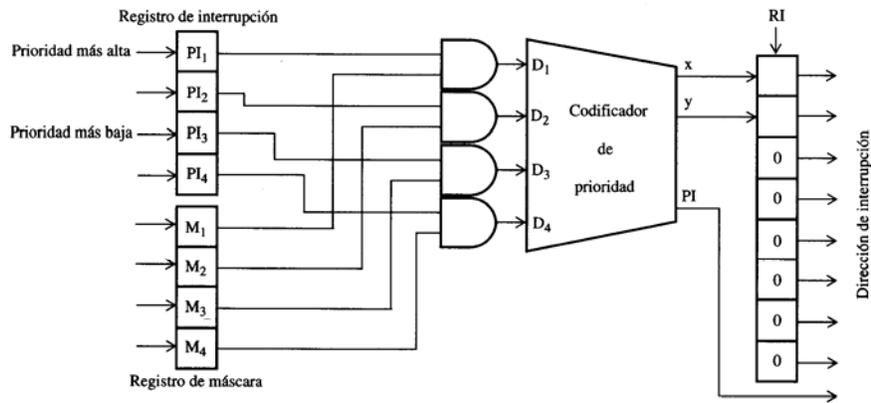
Tabla de verdad del circuito de interrupciones encadenadas



Etapla iésima de la interrupción por encadenamiento

**Mediante hardware paralelo:**

Se utiliza un registro de interrupción cuyos bits se controlan por las señales de interrupción PI<sub>i</sub> de cada periférico. La dirección se respuesta a la interrupción se obtiene mediante un codificador de prioridad que genera un número en función de la interrupción activada.



Hardware paralelo de interrupciones con prioridad

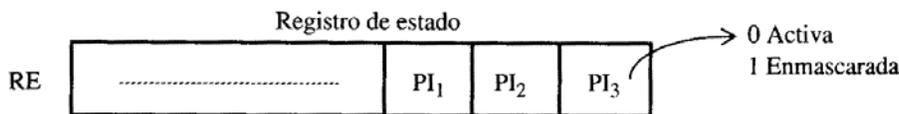
D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	y	x	PI
1	x	x	x	1	1	1
0	1	x	x	1	0	1
0	0	1	x	0	1	1
0	0	0	1	0	0	1
0	0	0	0	x	x	0

Tabla de verdad de un codificador de prioridad de 4 entradas

**Niveles de interrupción**

Hasta ahora se ha supuesto que el **sistema de interrupciones se desactiva** cuando se está **atendiendo** a una **interrupción** (independientemente de que se produzca una interrupción de mayor prioridad). Esto es debido a que solo se disponía de un único **nivel de interrupciones**. Esto es, dentro de un mismo nivel aunque una interrupción tenga mayor prioridad que otra, cuando se está atendiendo a una interrupción (ejecutando el programa de respuesta) no se puede atender a otras (aunque posean mayor prioridad) hasta que no se ha terminado la respuesta en curso.

En ocasiones es imprescindible atender a una interrupción a pesar de estar atendiendo a otras, en esos casos la CPU debe disponer de un **sistema** de interrupciones **multinivel** que permita alojar cada interrupción dentro de los diferentes niveles, de forma que mientras se está ejecutando una interrupción de un determinado nivel no se podrá atender a otras del mismo nivel o inferiores, pero permitirá interrumpir dicha respuesta si se produce una interrupción de nivel superior.



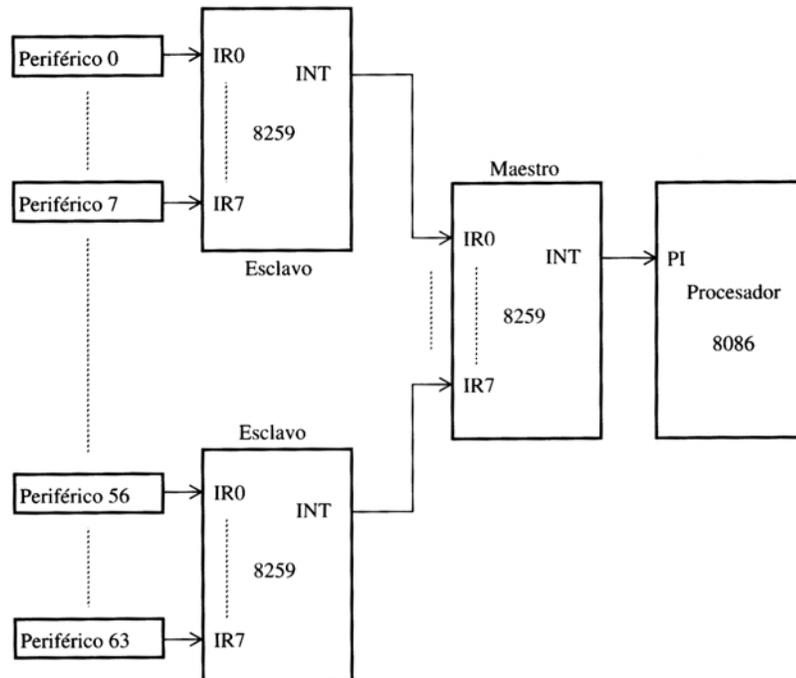
Registro de estado con múltiples líneas de interrupción

**ANIDAR = permite que interrupciones de mayor prioridad interrumpan la ejecución de las que tienen menor prioridad**

## Controlador de interrupciones:

Su misión es aumentar el número de líneas y encargarse de la gestión.

- Funciones**
- Identificar la fuente de interrupción.
  - Establecer las prioridades de cada periférico.
  - Activar/desactivar las interrupciones.
  - Enviar información a la CPU sobre la petición de interrupción y el periférico a atender.



**Problemas de E/S Controlado de interrupciones:**

**Problema 3-5**

Dadas las peticiones de interrupción ( $PI_i$ ), que se muestran en la Tabla 3.1, ¿qué programa de servicio de las interrupción estará ejecutando la CPU durante el intervalo de tiempo 0 - 100 ns?. Mostrar el cronograma de los valores de  $P_{ei}$  (prioridad de entrada el periférico i) y  $P_{si}$  (prioridad de salida del periférico i) para dicha secuencia de interrupciones. Se supondrá que el mecanismo de identificación de la prioridad de la interrupción es mediante encadenamiento y que hay conectados 4 periféricos. El sistema dispone de un hardware paralelo de interrupción con prioridad que permite que un periférico de mayor prioridad interrumpa a la CPU cuando está atendiendo a un periférico que tiene menor prioridad. Cada programa de servicio de las interrupciones, conjuntamente con la petición de la interrupción, necesita 20 ns para completarse. En el instante inicial (0 ns) arranca el programa principal (PP).

<b>Tiempo</b>	0 ns	10 ns	20 ns	45 ns	60 ns
<b>Acción</b>	Inicio PP	$PI_2$	$PI_1$	$PI_4$	$PI_3$

$$PI_1 > PI_2 > PI_3 > PI_4$$

Tabla 3.1: Instantes de solicitud de interrupción por diferentes dispositivos

**Solución:**

Instante de entrada de las interrupciones		$PI_2$	$PI_1$			$PI_4$		$PI_3$				
T(ns)	0	10	20	30	40	50	60	70	80	90	100	110
Programa en ejecución	PP	$PI_2$	$PI_1$		$PI_2$	$PI_4$	$PI_3$		$PI_4$	PP		
			Entra $PI_1$ pq tiene más prioridad y quedan pendientes 10ns de $PI_2$		Termina $PI_1$ y se reanuda $PI_2$ , en el 45ns aparece $PI_4$ pero tiene menor prioridad que $PI_2$ lo que hace que espere.	Se activa $PI_4$	Se interrumpe $PI_4$ (faltándole 10 ns) pq $PI_3$ tiene mayor prioridad y se atiende a $PI_3$		Termina $PI_3$ y se reanuda $PI_4$	Termina $PI_4$ y el conjunto de interrupciones y continua con el programa principal		

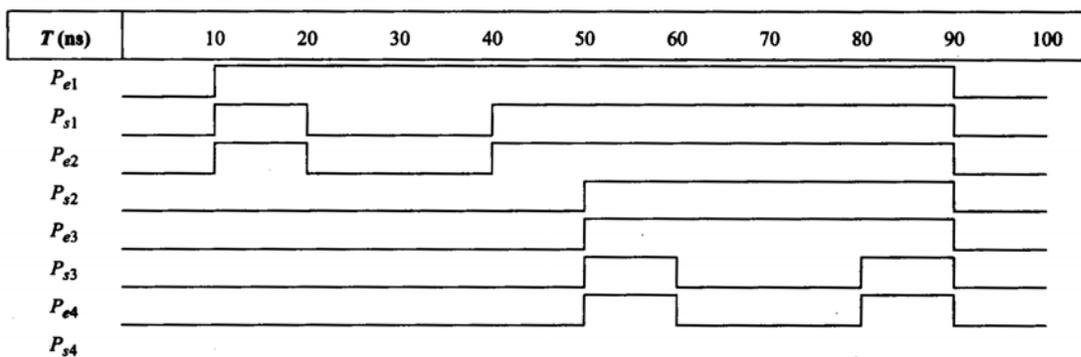


Tabla 3.4: Cronograma de  $P_{ei}$  y  $P_{si}$  para la secuencia de petición de interrupciones

**2002**

**Junio - 2ª semana**

7.- En un computador que usa E/S controlada por programa, el dispositivo de E/S tarda 6 mseg en tener disponible el dato solicitado. Suponiendo que el computador sólo se dedica a E/S, que el bucle de espera se implementa con una única instrucción, que la lectura de un dato y solicitud del siguiente dato por parte de la CPU lleva 10 instrucciones, y que cada instrucción de la CPU se ejecuta en 200 µseg, ¿qué tanto por ciento de su tiempo dedica la CPU al bucle de espera?

- A) 0%.      B) 99%.      C) 75%.      D) Ninguna de las anteriores

**2001**

**Junio      1ª Semana**

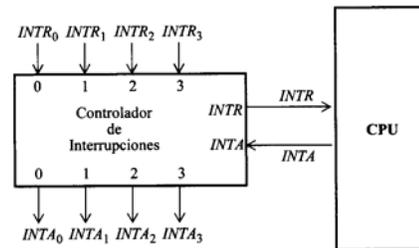
5.- En un sistema de E/S de un computador están conectados 2 discos duros, 1 impresora y 1 CDROM. El sistema de E/S gestiona los dispositivos mediante interrupciones **anidables** y además estos están agrupados por prioridades con **P1>P2>P3**, siendo la prioridad de los discos P1, la del CDROM P2 y la de la impresora P3. Se sabe además que las rutinas de interrupción llevan asociados unos tiempos de servicio, siendo éstos para los discos de 10 u.t. (unidades de tiempo), para el CDROM de 30 u.t. y para la impresora de 15 u.t. En un instante determinado (t =0 u.t.) se produce una petición de interrupción del primer disco duro, después en el instante t =3 u.t. llega una petición de interrupción de la impresora, posteriormente se produce una interrupción del CDROM en t = 5 u.t. y finalmente llega una petición de interrupción del segundo disco duro en t = 14 u.t. ¿En qué instante de tiempo finaliza la ejecución de la rutina de interrupción del CDROM?

- A) 30 u.t.      B) 45 u.t.      C) 50 u.t.      D) Ninguna de las anteriores

**Junio 2003 RESERVA**

6.- En la figura se presenta un diagrama con el circuito de gestión de prioridades sin enmascaramiento de un sistema de E/S (Prioridad P<sub>3</sub>>P<sub>2</sub>>P<sub>1</sub>>P<sub>0</sub>). La expresión lógica que define INTR es:

- A)  $INTR = INTR_3 + INTR_2 + INTR_1 + INTR_0$   
 B)  $INTR = INTR_3 + \overline{INTR_3}(INTR_2 + \overline{INTR_2}(INTR_1 + \overline{INTR_1}INTR_0))$   
 C)  $INTR = INTR_3 INTR_2 INTR_1 INTR_0$   
 D) Dos de las anteriores son ciertas.



**SEPTIEMBRE 2003**

3.-Suponga una CPU que dispone de 8 líneas de interrupción PI, i=1..8, con P<sub>1</sub>> P<sub>2</sub>>>P<sub>3</sub>>...>P<sub>8</sub>, que son enmascarables mediante 8 bits del registro de estado RE de la CPU (si el bit es 1 la petición está enmascarada y si es 0 está activa). El bit más significativo del registro corresponde a la petición de interrupción de mayor prioridad, el siguiente al segundo,... hasta el menos significativo que corresponde al de menor prioridad. La máscara de interrupciones es 6C, expresada en hexadecimal. Con esta máscara se solicita interrupción por las líneas 2 y 4 respectivamente (suponga que no está siendo atendida ninguna interrupción). La petición atendida en ese momento es:

- A) 2      B) 4      C) Cualquiera de las anteriores      D) Ninguna de las anteriores.

**PROBLEMAS**

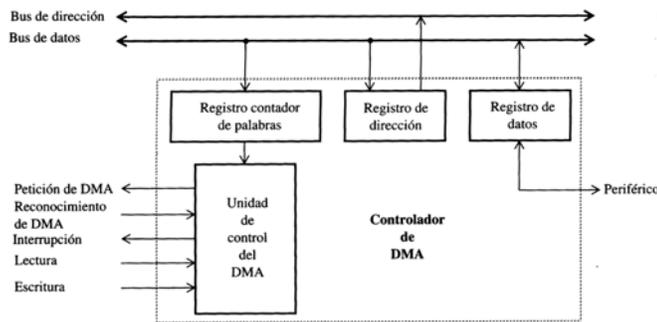
En la sección de problemas se han puesto los correspondientes a los resueltos en el libro de problemas:

**3.13 ..... Septiembre del 1999 y Junio del 2001 – 2ª semana**

## ACCESO DIRECTO A MEMORIA

- Inconvenientes de la E/S por interrupciones**
- La transferencia de datos limitada por la velocidad de la CPU
  - La CPU gestione la E/S ⇒ pérdida de rendimiento

El controlador de DMA (Acceso directo a memoria) transfiere directamente (sin pasar por la CPU) palabra a palabra el bloque completo de datos entre periférico y memoria. Participando la CPU solo al principio de la transmisión.



<b>Modos de transferencia</b>	- Por ráfagas	<p>Para la CPU y transmite el periférico</p> <p>DMA: Transferencia del bloque de datos</p>
	- Por robo de ciclos	<p>Roba un ciclo a la CPU por cada instrucción ejecutada</p> <p>DMA</p>
	- Transparente	<p>Espera a que la CPU deje libre el bus para transmitir</p> <p>Bus libre</p> <p>Búsqueda instrucción, Decodificación, Búsqueda operando, Ejecución, Guardar resultado</p>

- Configuración de la DMA**
- Bus único con DMA conectado
  - Bus único con DMA integrado
  - Controlador DMA conectado a bus de E/S

**Problemas de DMA:****Problema 3-15**

Un controlador de DMA esta transmitiendo, mediante robo de ciclos, caracteres a memoria desde un periférico a una velocidad de 19200 bps (bits/seg). Por su parte la CPU realiza la búsqueda de las instrucciones con una velocidad de 2 millones de instrucciones por segundo (2 MIP). ¿En qué porcentaje se reduce la velocidad del procesador debido al controlador de DMA?

Suponer una longitud de palabra de  $M$  bits y que la CPU emplea, en promedio,  $N$  ciclos para ejecutar una instrucción.

**Solución**

En la estrategia por *robo de ciclos* cuando el DMA toma el control del bus lo retiene durante un solo ciclo, transmite una palabra y libera el bus.

19200bps  $\Rightarrow$  19200/M palabras/sg  $\Rightarrow$  ciclos/sg que deja de ejecutar

$$\% = \frac{\text{Reducción}_n \text{ instrucciones}}{\text{Instrucciones sin DMA}} \times 100$$

$N = N^\circ$  medio de ciclos en ejecutar una instrucción  $\Rightarrow$  1 segundo  $\Rightarrow 2 \times 10^6 \times N$  ciclos de reloj sin DMA

SIN DMA  $\rightarrow 2 \times 10^6 \times N$  ciclos de reloj  $\rightarrow$  1 segundo

CON DMA  $\rightarrow 2 \times 10^6 \times N + 19200/M \rightarrow x$  segundos

$$x = \frac{2 \times 10^6 \times N + \frac{19200}{M}}{2 \times 10^6 \times N} = 1 + \frac{19200}{2 \times 10^6 \times N \times M}$$

$$\% = \frac{19200}{2 \times 10^6 \times N \times M} \times 100 = \frac{0,96}{N \times M} \%$$



## Exámenes Septiembre de 2000

2.- Un controlador de DMA está transmitiendo palabras a memoria mediante la estrategia *transparente*. Cuando la UCP dispone en exclusiva del bus las instrucciones emplean, en promedio, 4 ciclos de reloj en ser ejecutadas, aunque en 2 de ellos consecutivos no se requiere el acceso al bus. El controlador de DMA tarda 2 ciclos de reloj en transmitir una palabra. Si la frecuencia de reloj del computador es de  $200 \times 10^6$  ciclos/seg., ¿cuánto tiempo tardará el controlador de DMA en mover un bloque de datos compuesto por  $100 \times 10^6$  palabras?

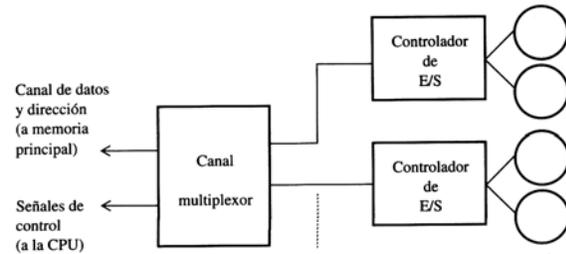
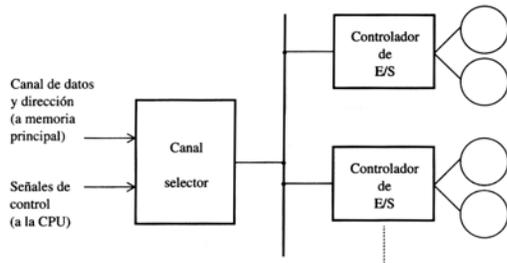
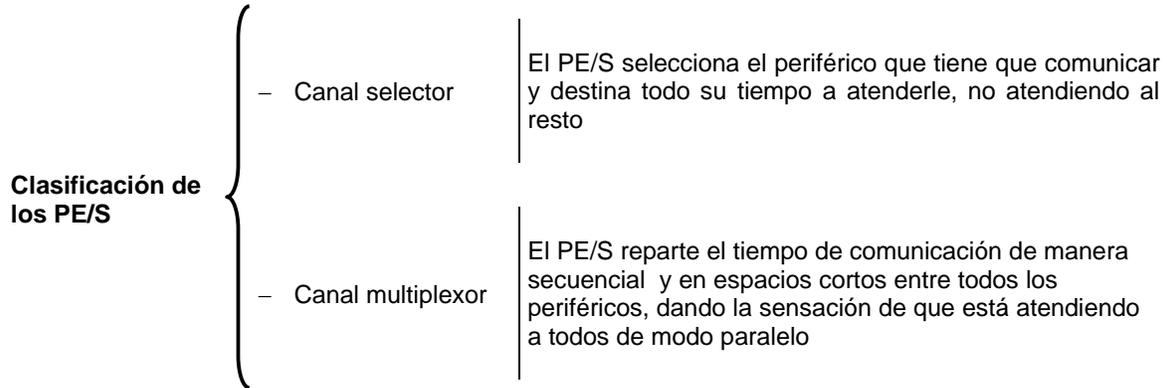
- A) 1 seg.      B) 2 seg.      C) 2.5 seg.      D) Ninguna de las anteriores.

**¿Y si la DMA tarda 3 ciclos de reloj en transmitir?**

## PROCESADOR DE ENTRADA SALIDA (PE/S)

Es una unidad de entrada salida a la que se le ha dotado de un microprocesador, formando dicha unidad un microcomputador.

Cuando se le solicita una comunicación con el periférico, el PE/S recibe los datos desde el procesador del sistema y ejecuta el **programa** correspondiente (**almacenado en memoria principal**) de manera autónoma e independiente al procesador principal del sistema, suponiendo ello un aumento del rendimiento.



**Problemas de PE/S:**

**Problema 3-25**

Un computador de 32 bits dispone de dos canales selectores y un canal multiplexor. Cada canal selector controla dos unidades de discos y dos unidades de cinta. Al canal multiplexor están conectados dos impresoras, dos lectoras de tarjetas y 10 pantallas. Para los diferentes periféricos se suponen las siguientes velocidades de transferencia:

- Unidad de disco ..... 800 Kbytes/seg
- Unidad de cinta magnética ..... 200 Kbytes/seg
- Impresora ..... 8 Kbytes/seg
- Lectora de tarjeta ..... 1,5 Kbytes/seg
- Pantalla..... 1 Kbytes/seg

Estimar la máxima velocidad de transferencia de E/S agregada en este sistema

**Solución**

Canal selector  $\Rightarrow$  selecciona un canal y realiza el envío  $\Rightarrow$  velocidad máxima será la del dispositivo más rápido (unidad disco)  
 Canal multiplexor  $\Rightarrow$  reparte la conexión entre todos  $\Rightarrow$  velocidad máxima será la suma de la de todos los dispositivos **si el bus lo permite en cuanto a velocidad.**

Canal selector  $\Rightarrow 2 \times 800 = 1600$  Kbytes/seg  
 Canal multiplexor  $\Rightarrow (2 \times 8) + (2 \times 1,5) + (10 \times 1) = 29$  Kbytes/seg

Velocidad total  $\Rightarrow$  suma de los dos canales  $\Rightarrow 1600 + 29 = 1629$  Kbytes/seg

**2001 Septiembre**

8.- Un computador dispone de un *canal multiplexor* que controla 5 unidades de disco y 5 unidades de cinta. Las unidades de disco tienen una velocidad de transferencia de  $6 \times 10^6$  bytes/seg., y las de cinta de  $10^5$  bytes/seg. Si para la transferencia de un byte del canal multiplexor a memoria principal se utiliza el bus del sistema durante 50 nseg, la máxima velocidad de transferencia agregada a este sistema es de:

- A)  $3.05 \times 10^7$  bytes/seg.    B)  $6 \times 10^6$  bytes/seg.    C)  $2 \times 10^7$  bytes/seg.    D) Ninguna de las anteriores.

**2000 Junio 2000 - 2ª semana**

5.- Un computador dispone de un *canal multiplexor* que controla 5 unidades de disco y 10 unidades de cinta. Las unidades de disco tienen una velocidad de transferencia de  $10^6$  bytes/seg., y las de cinta de  $2 \times 10^5$  bytes/seg. Si para la transferencia de un byte del canal multiplexor a memoria principal se utiliza el bus del sistema durante 200 nseg. ¿Cuál es la velocidad máxima de transferencia agregada a este sistema?

- A)  $7 \times 10^6$  bytes/seg    B)  $5 \times 10^6$  bytes/seg    C)  $10^6$  bytes/seg    D) Ninguna de las anteriores

**2000 Exámenes de Junio -1ª semana**

1.- Un computador dispone de un canal multiplexor que controla 5 unidades de disco y 10 unidades de cinta. Las unidades de disco tienen una velocidad de transferencia de  $10^6$  bytes/seg., y las de cinta de  $2 \times 10^5$  bytes/seg. Si para la transferencia de un byte del canal multiplexor a memoria principal se utiliza el bus del sistema durante 200 nseg. ¿Qué porcentaje máximo de tiempo estaría ocupado el bus del sistema en una transferencia de datos del canal multiplexor a memoria principal?

- A) 30%    B) 20 %    C) 14 %    D) Ninguna de las anteriores