



TEMA III

Unidad de entrada-salida



3.1 Dispositivos externos

3.2 Controlador de E/S

3.2.1 Funciones del controlador de E/S

3.2.2 Estructura del controlador de E/S

3.2.3 Estructura del sistema de E/S

3.3 E/S controlada por programa

3.3.1 Órdenes de E/S

3.3.2 Instrucciones de E/S

3.4 E/S por interrupciones

3.4.1 Clasificación de las interrupciones

3.4.2 Origen de las interrupciones

3.4.3 Número de líneas de interrupción

3.4.4 Control de la CPU sobre la interrupción

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

3.4.6 Niveles de interrupción

3.4.7 Controlador de interrupciones

3.4.8 Ejemplos de controladores de interrupciones

3.5 Acceso directo a memoria (DMA)

3.5.1 Controlador de DMA

3.5.2 Transferencia de datos mediante DMA

3.5.3 Configuración del DMA

3.6 Procesador de E/S (PE/S)

3.6.1 Características de los PE/S

3.6.2 Clasificación de los PE/S



Entrada/salida

- La unidad de E/S proporciona un método de comunicación eficaz entre el sistema central y el usuario.
- Los motivos por los que no se conectan los periféricos al bus directamente son:
 - **La gran variedad de tipos y métodos de operación**, por lo que no es práctico incorporar dentro de la UCP la lógica que se necesita para controlar un conjunto tan amplio de dispositivos.
 - **La velocidad de transferencia** de datos de los periféricos suelen ser menor que la de la memoria con la UCP.
 - Los periféricos utilizan **formatos y longitudes** de palabra distintas a las del computador.



3.1 Dispositivos externos

- Un dispositivo externo se conecta al computador a través de un enlace con un **controlador de E/S**.
- La comunicación con el controlador de E/S se efectúa mediante señales de:
 - Datos,
 - Control y de
 - Estado del dispositivo.



3.2 Controlador de E/S

- Es el módulo responsable del control de uno o más dispositivos externos y del intercambio de datos entre dichos periféricos con la memoria principal o con los registros de la UCP
- 3.2.1 Funciones del controlador de E/S
 - Decodificación de la orden
 - Intercambio de datos
 - Información del estado
 - Reconocimiento de la dirección

3.2.2 Estructura del controlador de E/S

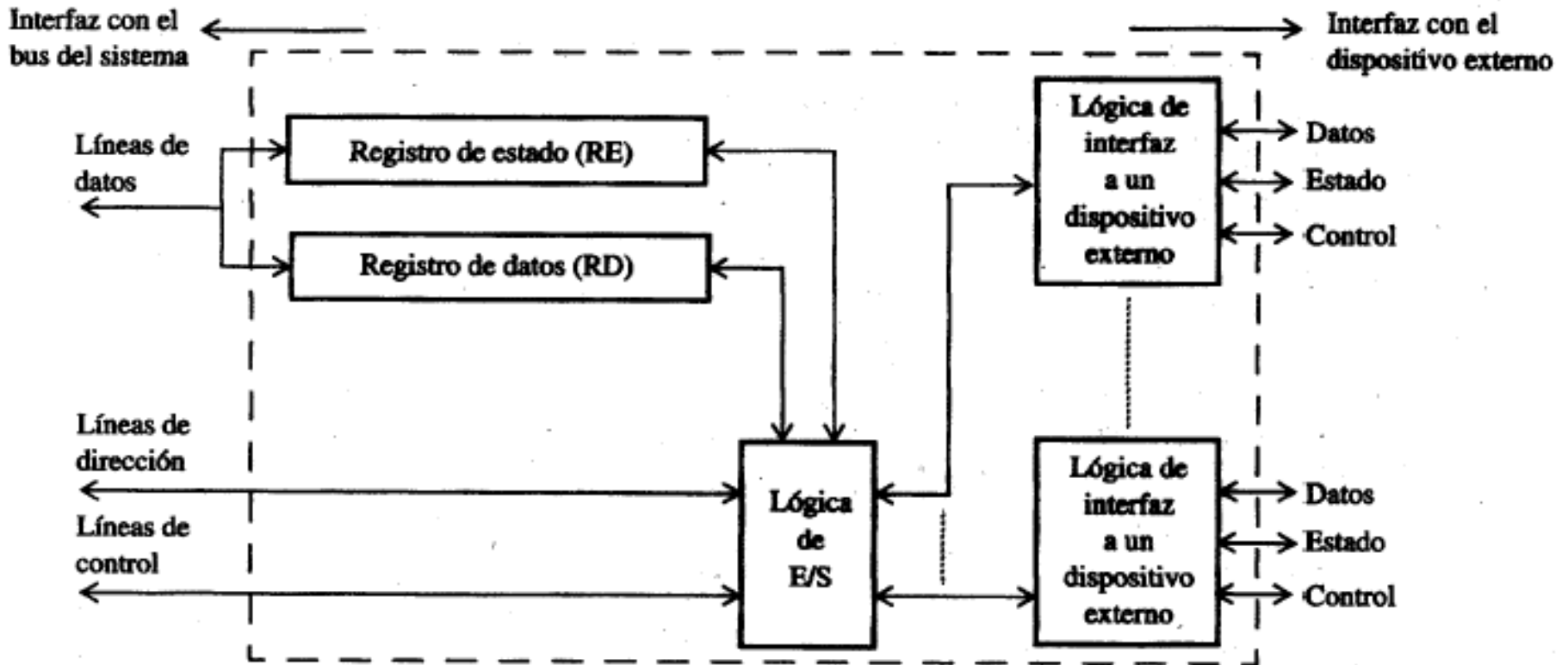


Figura 3.3: Diagrama de bloques de un controlador de E/S



3.2.3 Estructura del sistema de E/S

■ Modos de interconexión E/S – CPU

- Buses independientes para memoria y sistema
- Bus común para memoria y sistema
 - Líneas de control separadas (E/S aislada de memoria)
 - Líneas de control común (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)

3.3 E/S controlada por programa

	<i>Sin interrupciones</i>	<i>Con interrupciones</i>
<i>Transferencia de E/S a memoria a través de la CPU</i>	E/S controlada por programa	E/S con interrupciones
<i>Transferencia de E/S a memoria directa</i>		Acceso directo a memoria (DMA)

Tabla 3.2: Mecanismos básicos de E/S

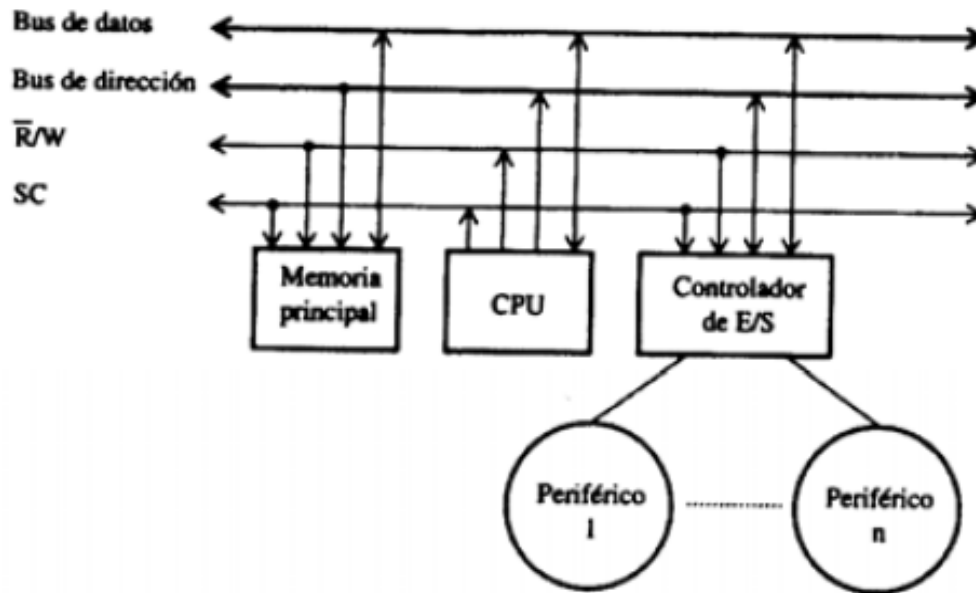
- La CPU está pendiente en todo momento (sondeo constante mediante bucles) del estado de la E/S.
- **Inconvenientes**
 - Pérdida de tiempo en bucle de espera



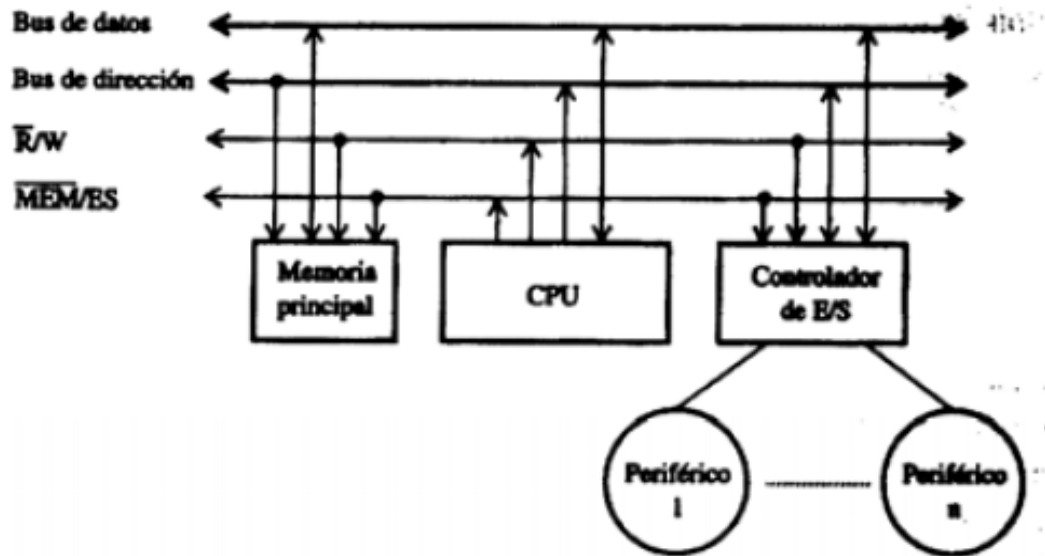
3.3.2 Instrucciones de E/S


- Cuando la UCP, la memoria principal y la unidad de E/S comparten un bus común
 - E/S localizada en memoria:
 - Hay un único espacio de direcciones para las posiciones de memoria y los dispositivos de E/S.
 - Se reserva un segmento del espacio de direcciones para los registros internos de los controladores
 - Mismas instrucciones para memoria y periférico
 - E/S aislada:
 - El bus del sistema dispone, además de las líneas de control de lectura y escritura en memoria, de líneas de control específicas de E/S para acceder a los periféricos
 - El rango completo de direcciones está disponible para ambos

Localizada en memoria memoria



Aislada de memoria





5.- Indique si las siguientes afirmaciones sobre *E/S localizada en memoria* son verdaderas:

I. Sería posible realizar operaciones directamente sobre los puertos de E/S de datos sin necesidad de moverlos explícitamente por programa a algún registro de la CPU.

II. Puede evitar gastar espacio de direcciones empleando una señal de control especial para indicar que está realizando una operación de E/S localizada en memoria.

A) I: sí, II: sí. B) I: sí, II: no. C) I: no, II: sí. D) I: no, II: no.

La afirmación I es verdadera. Ver ventaja b) de la página 59 del Problema 3-1.

La afirmación II es verdadera. Ver desventaja a) de la página 60 del Problema 3-1.



3.4 E/S por interrupciones

- La idea básica es eliminar el bucle de espera.
- La UCP envía una orden de E/S al periférico y prosigue la tarea que estaba ejecutando
- Cuando el periférico está preparado para intercambiar información, fuerza una interrupción en la tarea que realiza la UCP para que atienda la operación de E/S



Tratamiento de una interrupción:

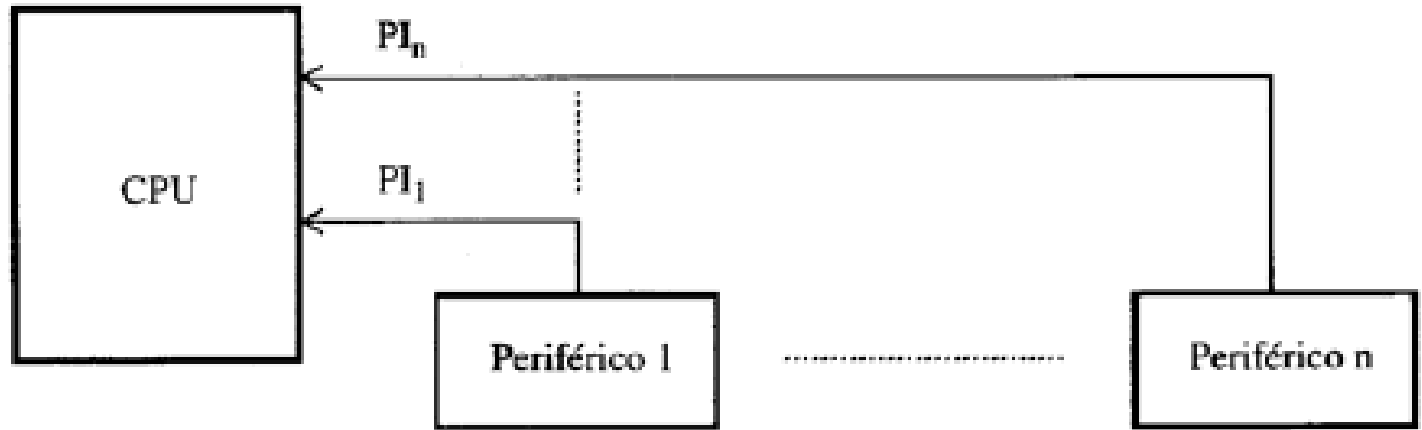
1. Activar el sistema de interrupciones, validando las que sea necesario.
2. Periférico activa la interrupción (PI).
3. CPU termina la instrucción en curso, salva el PC (Contador de programa) y algunos de los otros registros usados por el programa.
4. CPU inhibe interrupciones y carga en el PC el vector 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 y se continúa con la ejecución del programa interrumpido.

Origen de las interrupciones	<p>a) Externa. Las provoca un periférico</p> <p>b) Interna. Las provoca la CPU (ej. dividir por 0)</p>
Número de líneas de interrupción	<p>a) 1 línea. Solo 1 línea de petición de interrupción PI</p> <p>b) Múltiples líneas. PI_1, PI_2, \dots, PI_n</p>
Control de la CPU sobre la interrupción	<p>a) Enmascarables. La CPU puede desactivarlas</p> <p>b) No enmascarables. La CPU no puede desactivarlas</p>
Identificación de la fuente de la interrupción	<p>a) Múltiples líneas. PI_1, PI_2, \dots, PI_n</p> <p>b) Encuesta. La interrupción se identifica por programa</p> <p>c) Vectorizadas. La interrupción identifica al periférico</p>
Gestión de la prioridad de la interrupción	<p>a) Por software. Un programa determina la prioridad</p> <p>b) Por hardware. Un circuito determina la prioridad</p>
Niveles de interrupción	<p>a) Nivel único. La interrupción no puede interrumpirse</p> <p>b) Multinivel. Anidamiento de interrupciones</p>

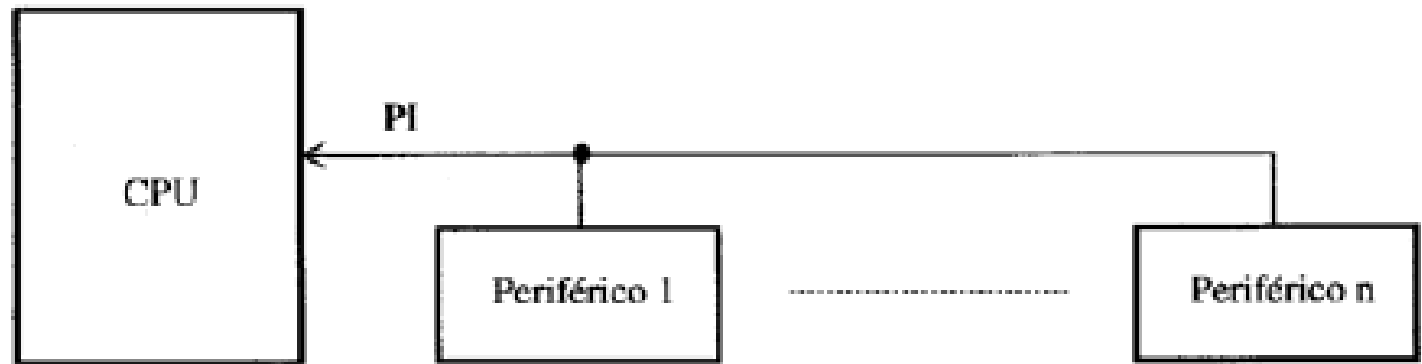
Tabla 3.3: Clasificación de las interrupciones

3.4.3 Número de líneas de interrupción

- *Múltiples líneas*
- *Línea única:*



a) La CPU posee varias entradas de interrupciones



b) La CPU posee una única entrada de interrupción



3.4.4 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 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”.
- Un **registro de estados** que posee tantos bits como interrupciones existan, bits es “1” si la interrupción es autorizada por el registro de máscaras y un “0” en caso contrario.

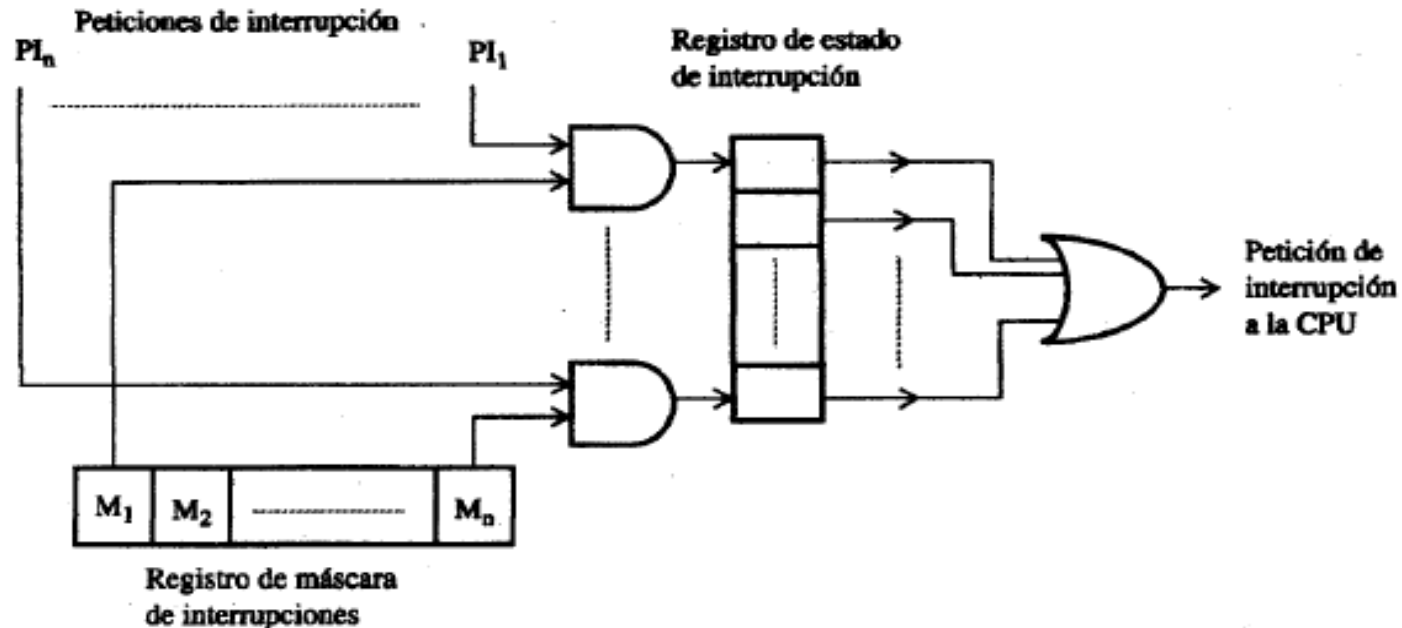
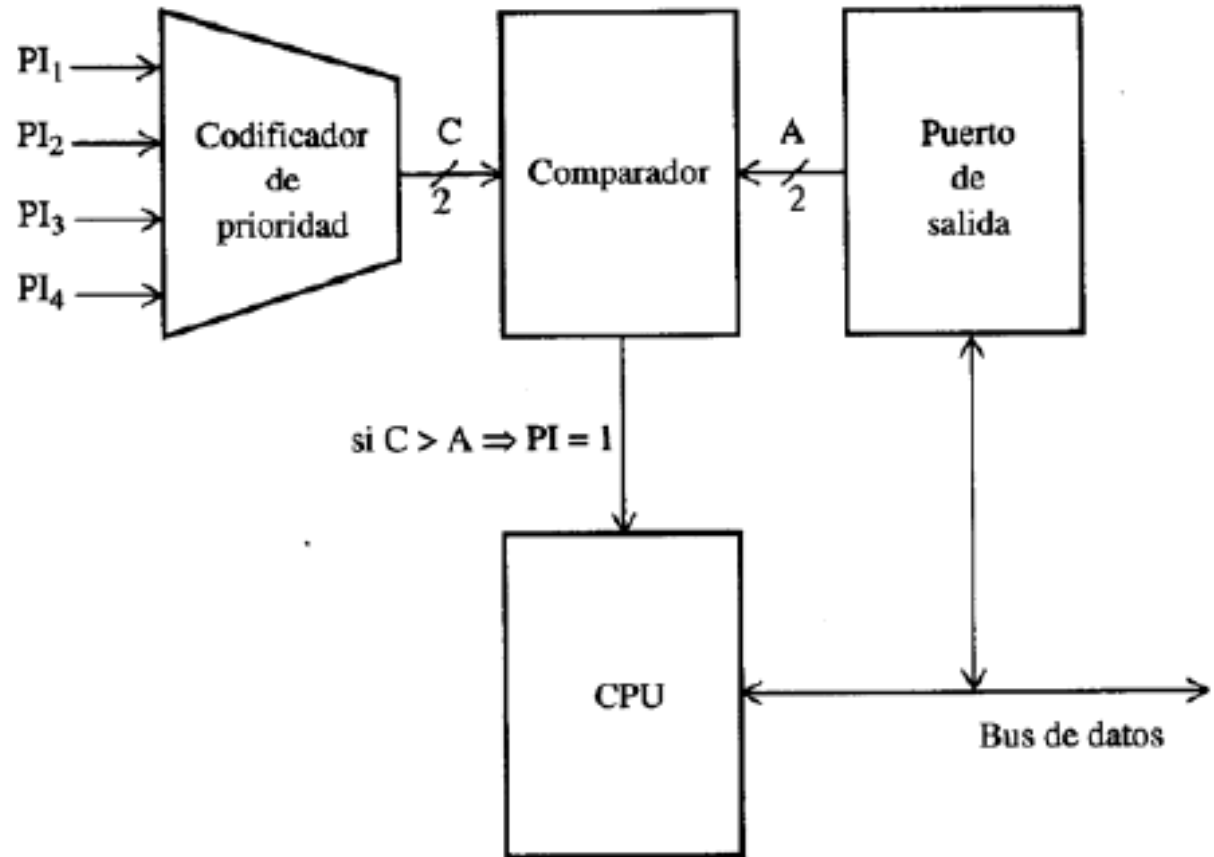



Figura 3.16: Enmascaramiento individual de cada entrada de interrupción

Enmascaramiento por nivel

- Se ordenan atendiendo a un criterio de prioridad.
- La CPU fija un nivel
 - Las que estén en un nivel superior son atendidas



Enmascaramiento de interrupciones por nivel



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

■ ***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.

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 3.4: Tabla de verdad del circuito de interrupciones encadenadas

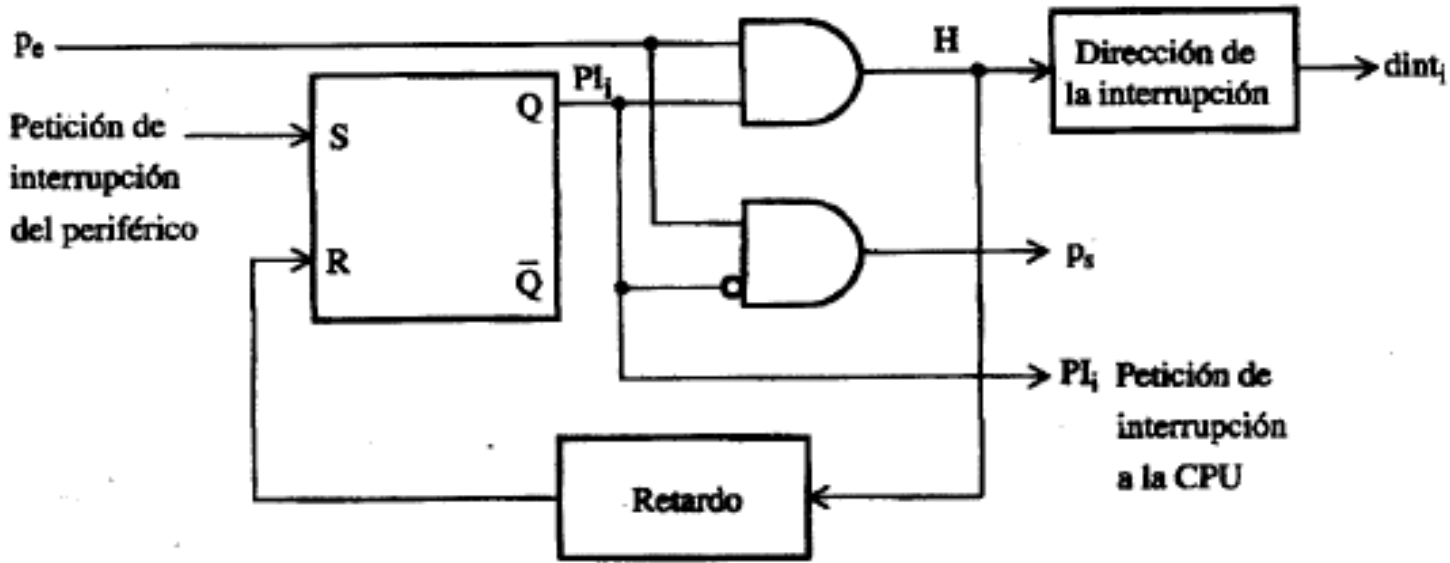
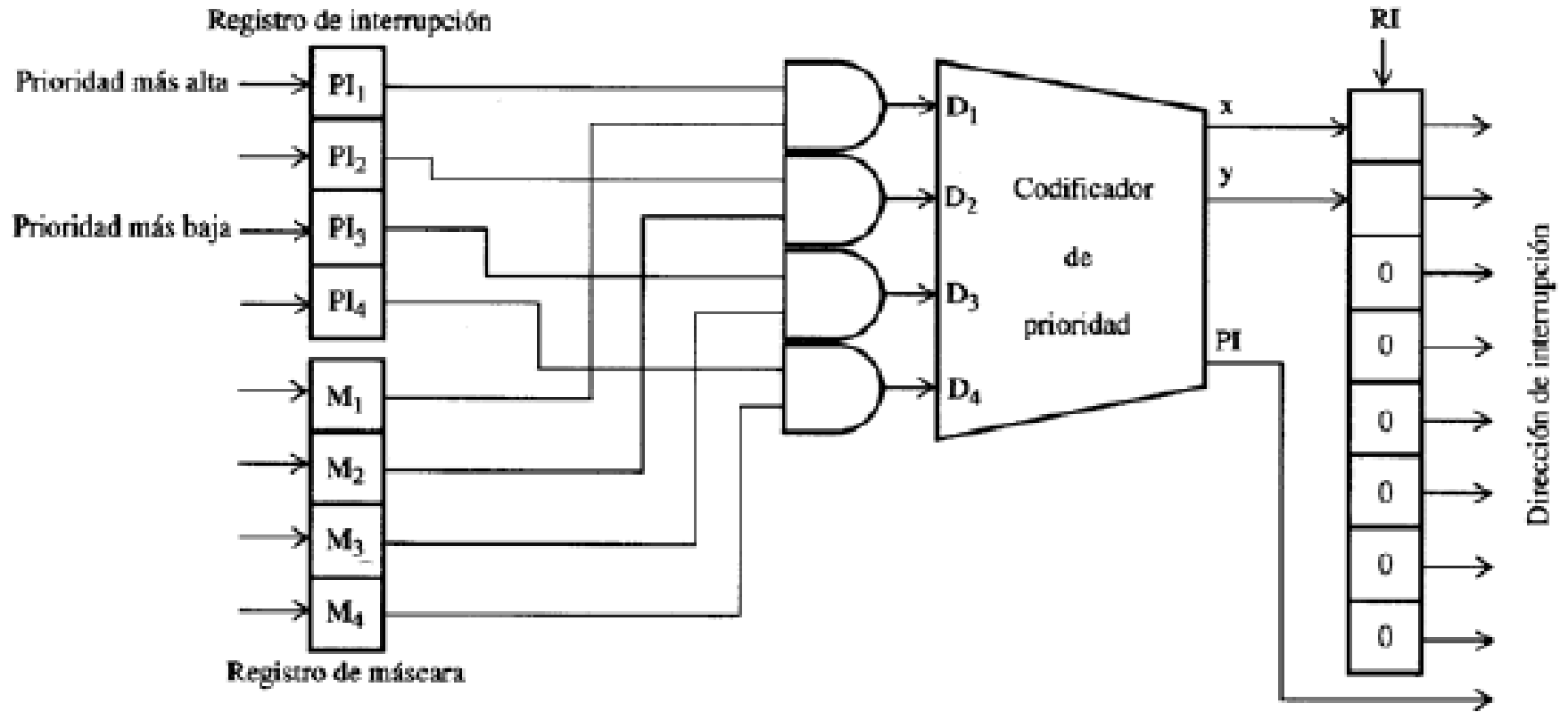


Figura 3.20: Etapa iésima de la interrupción por encadenamiento

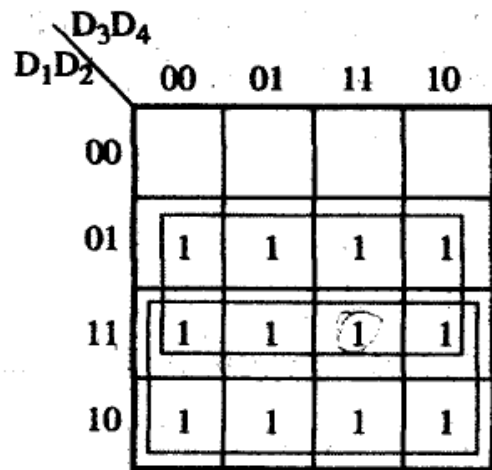


Mediante hardware paralelo

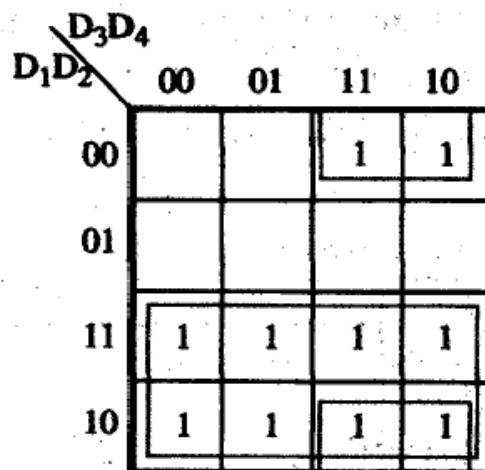
- Se utiliza un registro de interrupción cuyos bits se controlan por las señales de interrupción Pli de cada periférico.
- La dirección de respuesta a la interrupción se obtiene mediante un codificador de prioridad que genera un número en función de la interrupción activada.



Hardware paralelo de interrupciones con prioridad



$$y = D_1 + D_2$$



$$x = D_1 + D_3 \bar{D}_2$$

Figura 3.22: Diagrama de Karnaugh del codificador de prioridad

D ₁	D ₂	D ₃	D ₄	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



En la Figura 3.23 se muestra el circuito lógico que implementa al codificador de prioridad mediante las funciones booleanas siguientes:

$$x = D_1 + D_3 \overline{D_2}$$
$$y = D_1 + D_2$$
$$PI = D_1 + D_2 + D_3 + D_4$$

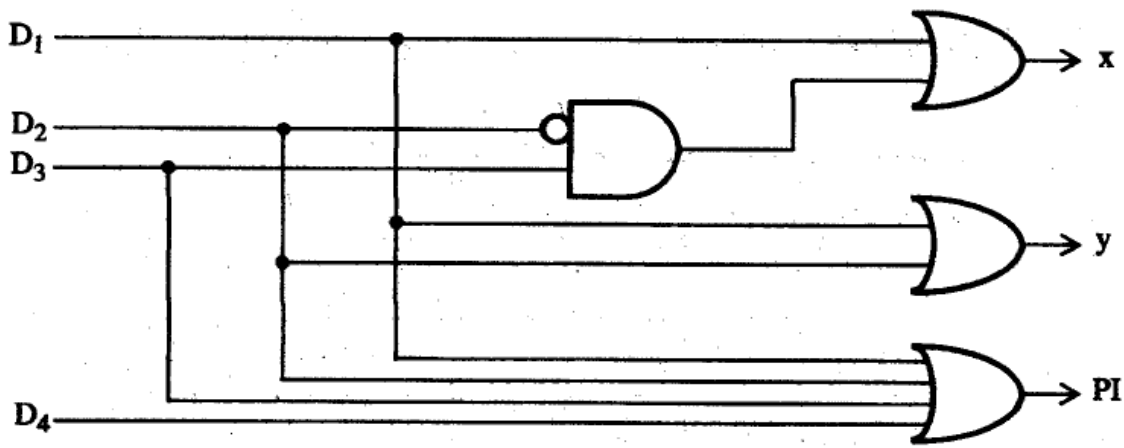


Figura 3.23: Diagrama lógico del codificador de prioridad de 4 entradas

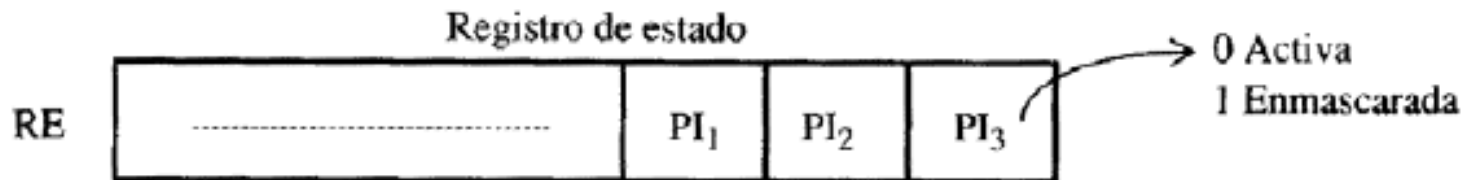
3.4.6 Niveles de interrupción

■ Sistema de un único nivel de interrupción

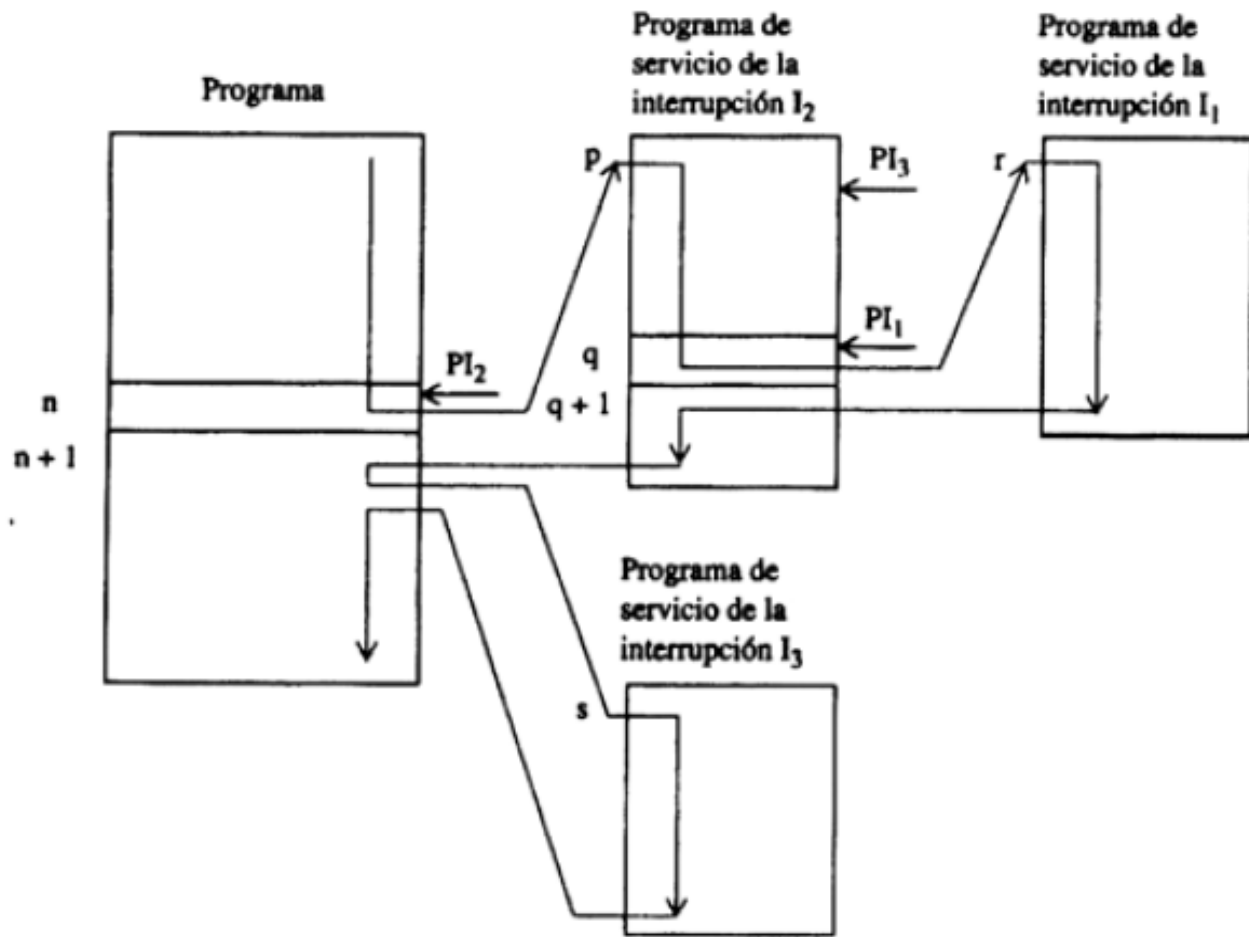
- Las interrupciones se desactivan cuando se está atendiendo una

■ **Sistema de interrupciones *multinivel***

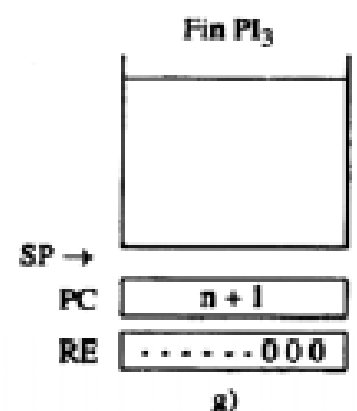
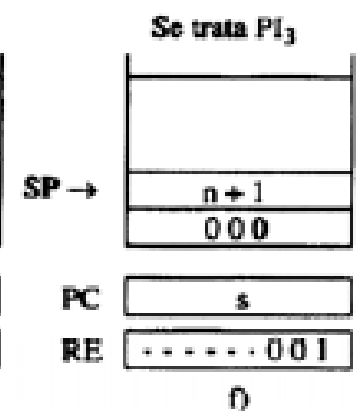
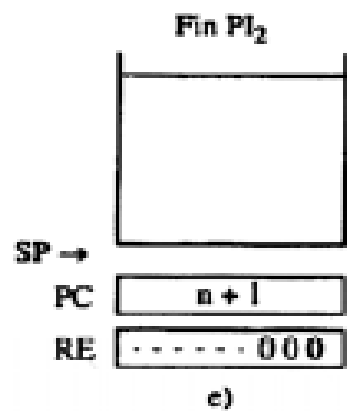
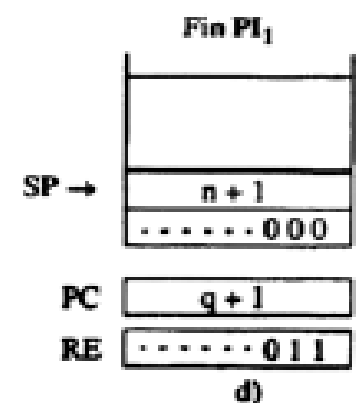
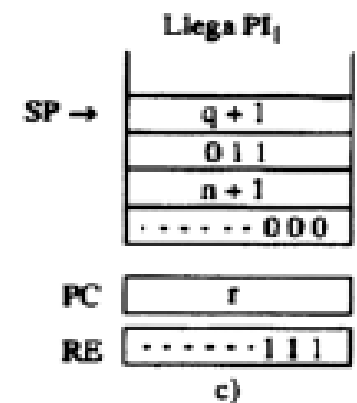
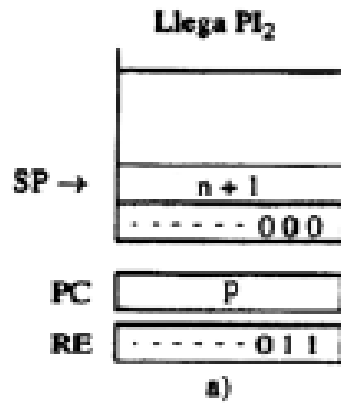
- Si es necesario atender a una interrupción de mayor prioridad a pesar de estar atendiendo a otras
- Permite ANIDAR interrupciones
 - Permite que interrupciones de mayor prioridad interrumpen la ejecución de las que tienen menor prioridad



Registro de estado con múltiples líneas de interrupción



Evolución de la pila y de los registros de la CPU

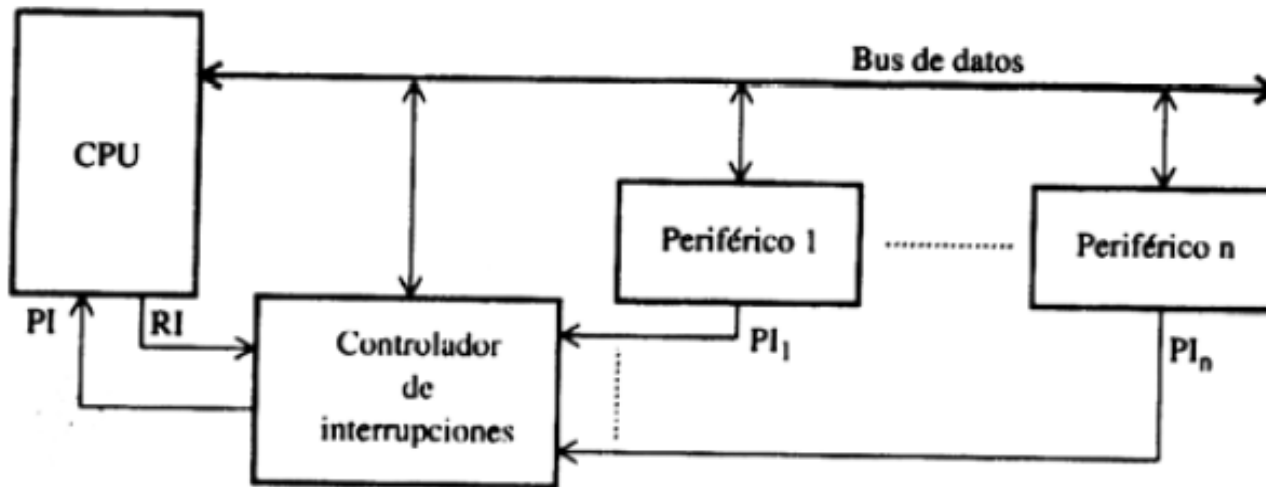




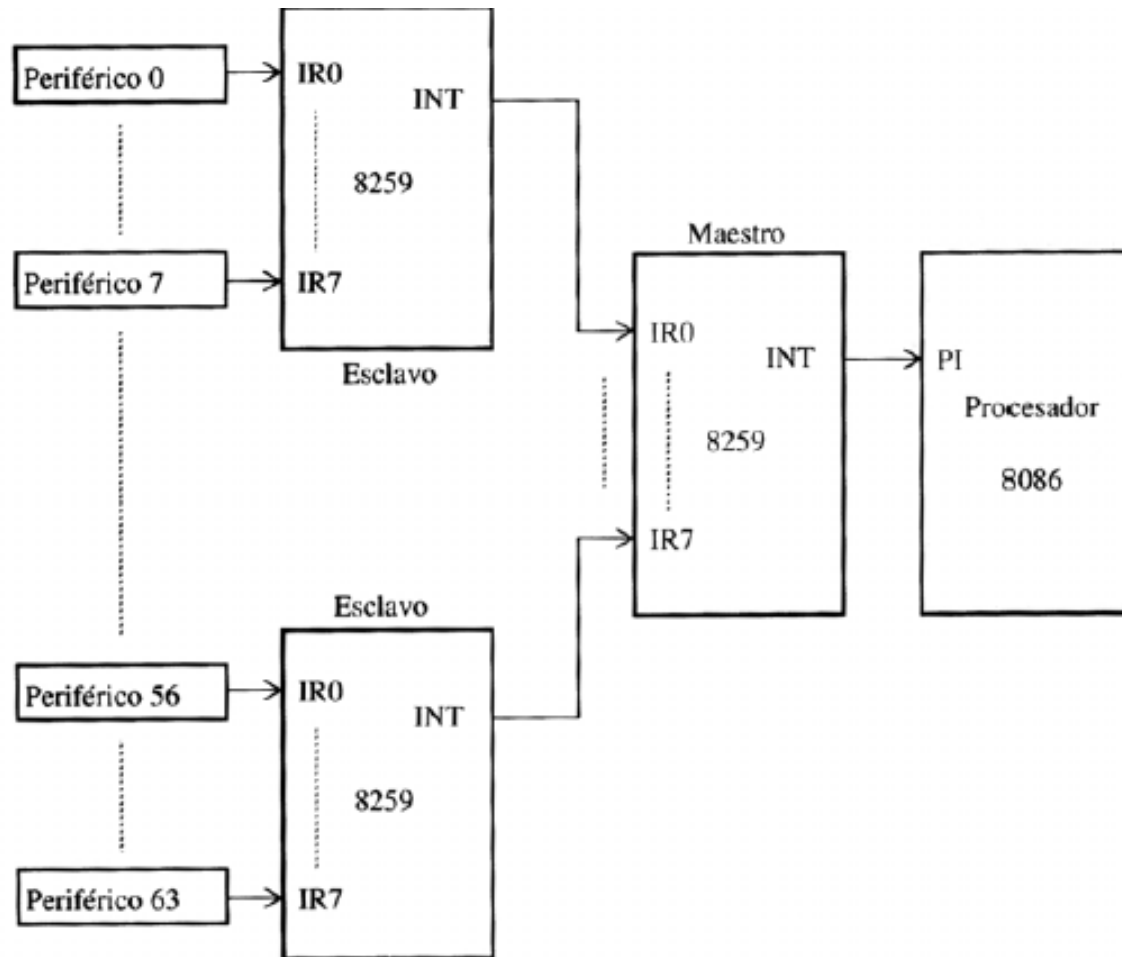
3.4.7 Controlador de interrupciones

- Las funciones que realiza el PIC son:
 - Identificar la fuente de la interrupción.
 - Establecer las prioridades de cada periférico.
 - Activar o desactivar de forma selectiva las peticiones de interrupción recibidas.
 - Enviar a la UCP información sobre la petición de interrupción y cuál es el periférico que debe ser atendido.

Controlador de interrupciones



Controlador 8259A



Considere un codificador de prioridad con 4 entradas (I_0, I_1, I_2, I_3) y 3 salidas (S_0, S_1, PI) cuyo funcionamiento viene descrito por la tabla de la verdad adjunta

I_0	I_1	I_2	I_3	S_0	S_1	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

Indique cuál de las afirmaciones siguientes es correcta

- A) $S_0 = I_0 + I_1$
- B) $S_1 = I_0 + I_2 \cdot I_1'$
- C) Las dos anteriores son correctas.
- D) Ninguna de las anteriores

I_0	I_1	I_2	I_3	S_0	S_1	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

Salida S_0

		I_2I_3			
		00	01	11	10
I_0I_1	00				
	01	1	1	1	1
	11	1	1	1	1
	10	1	1	1	1

Salida S_1

		I_2I_3			
		00	01	11	10
I_0I_1	00			1	1
	01				
	11	1	1	1	1
	10	1	1	1	1

Luego las funciones lógicas de S_0 y S_1 son:

$$S_0 = I_0 + I_1$$

$$S_1 = I_0 + I_2 \bar{I}_1$$

Luego la respuesta correcta es la **C**.

3.5 Acceso directo a memoria (DMA)

- **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.

Bus de dirección

Bus de datos

Registro contador
de palabras

Registro de
dirección

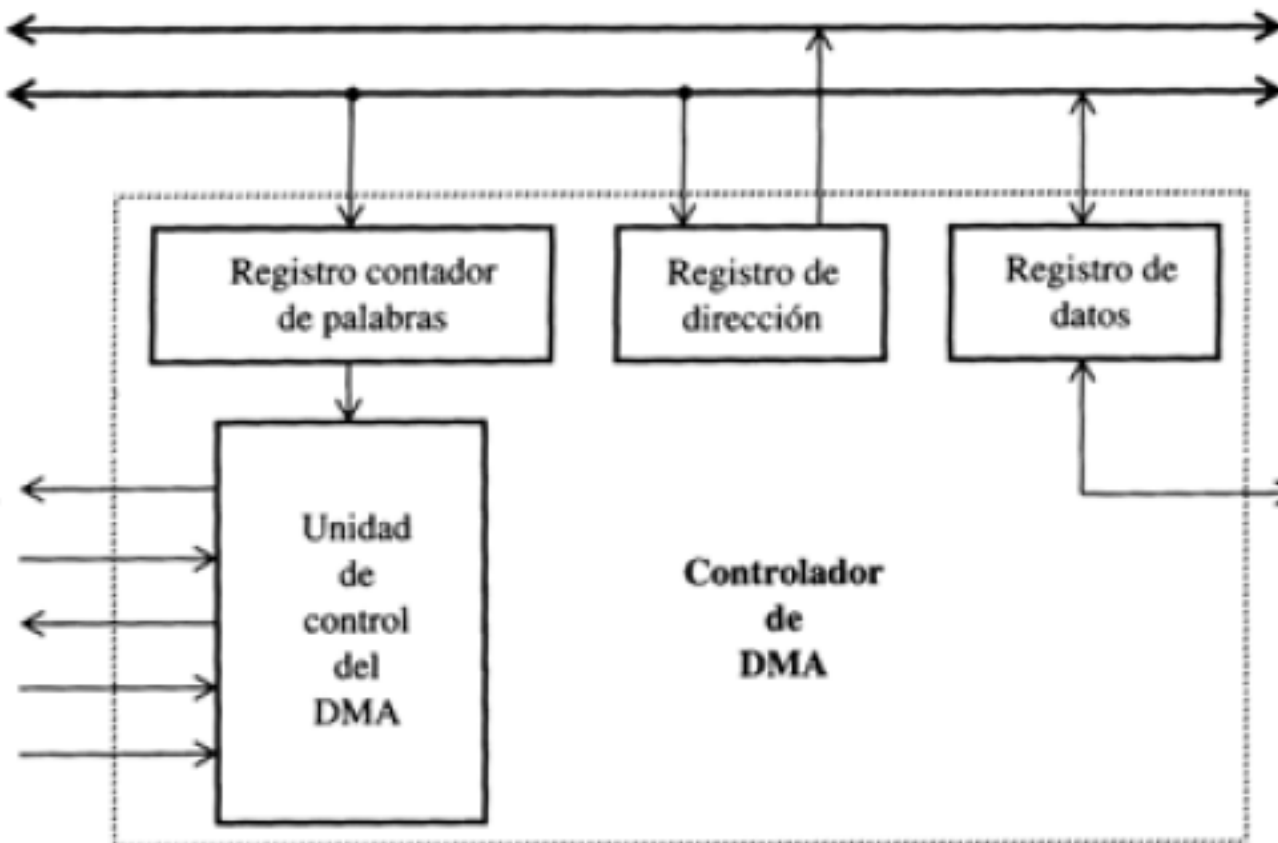
Registro de
datos

Unidad
de
control
del
DMA

Controlador
de
DMA

Periférico

Petición de DMA
Reconocimiento
de DMA
Interrupción
Lectura
Escritura



Por ráfagas

- Cuando el DMA toma el control del bus no lo libera hasta haber transmitido el bloque de datos completo
- La velocidad de transferencia es máxima, pero la UCP está inactiva durante periodos grandes.



2007 6.-Se dispone de un computador que funciona a una velocidad de 10^8 ciclos por segundo y en el que, en promedio, una instrucción emplea 10 ciclos máquina y cada operación de lectura o escritura de una palabra de memoria tarda 2 ciclos máquina.

Determinar la máxima velocidad de transferencia de datos si en este sistema se emplea DMA con estrategia de transferencia *por ráfagas*.

- 10^7 palabras/seg.
- 2×10^7 palabras/seg.
- 6.66×10^6 palabras/seg.
- Ninguna de las anteriores

SOLUCIÓN

6. Véase el apartado 3.5.2 del texto de teoría y el problema 3-16.

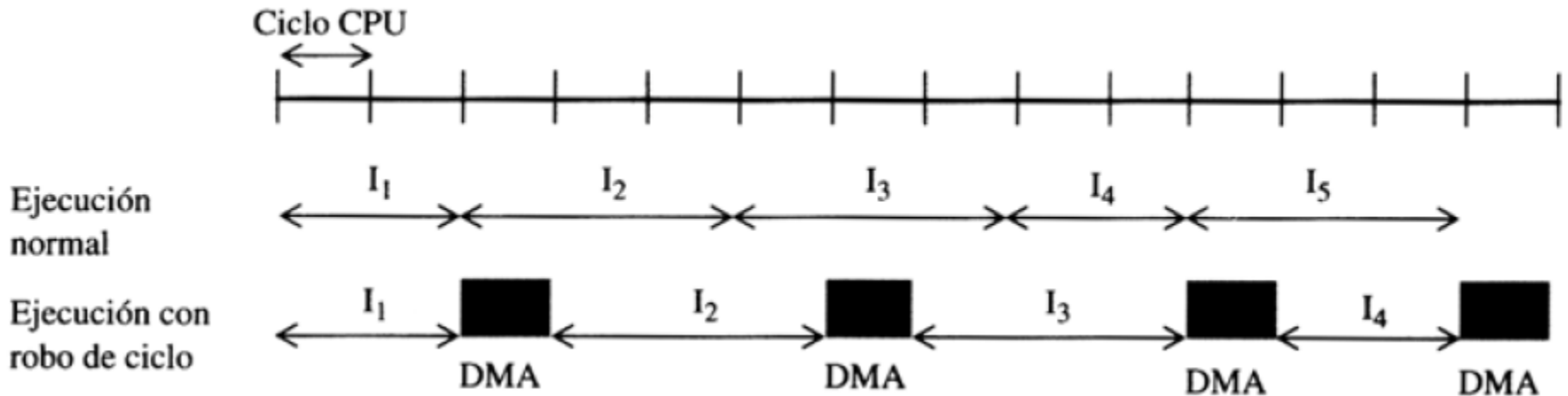
El DMA por ráfagas no libera el bus hasta haber transmitido el bloque de datos. Así pues, la velocidad máxima es:

$$(10^8 \text{ ciclos/seg}) / (2 \text{ ciclos/palabra}) = 5 \times 10^7 \text{ palabras/seg.}$$

Respuesta: D (Ninguna de las anteriores)

Por robo de ciclos

- Toma el control durante un ciclo, transfiere una palabra y lo libera
- Reduce al máximo la velocidad de transmisión y la interferencia del controlador sobre la UCP



EJERCICIO

6.- Un controlador de DMA está transmitiendo, mediante robo de ciclos, caracteres a memoria desde un periférico.

La CPU realiza la búsqueda de las instrucciones a una velocidad de 1 millón de instrucciones por segundo.

Si el porcentaje en que se reduce la velocidad del procesador debido al controlador DMA es de 0.89%,

¿a qué velocidad estará transmitiendo el DMA?

Suponga una longitud de palabra de 8 bits y

que la CPU emplea en promedio 2 ciclos para ejecutar una instrucción.

A) 132000 bits/seg

B) 142400bits/seg

C) 123700 bits/seg

• Ninguna de las anteriores

SOLUCIÓN:

Véase el problema 3-15 del libro de problemas. Si v es la velocidad pedida, el número de palabras que se transmiten por segundo es:

La penalización que el número de instrucciones no ejecutadas supone para la CPU es del 0.89%, por lo tanto:

$$10^6 \text{ instrucciones/seg} \times 2 \text{ ciclo/instrucción} = 2 \times 10^6 \text{ ciclos /seg}$$

el DMA utiliza el 0.89 % de los ciclos por tanto son:

$$(2 \times 10^6 \times 0.89) / 100 = 2 \times 0,89 \times 10^4 \text{ ciclos/segundo};$$

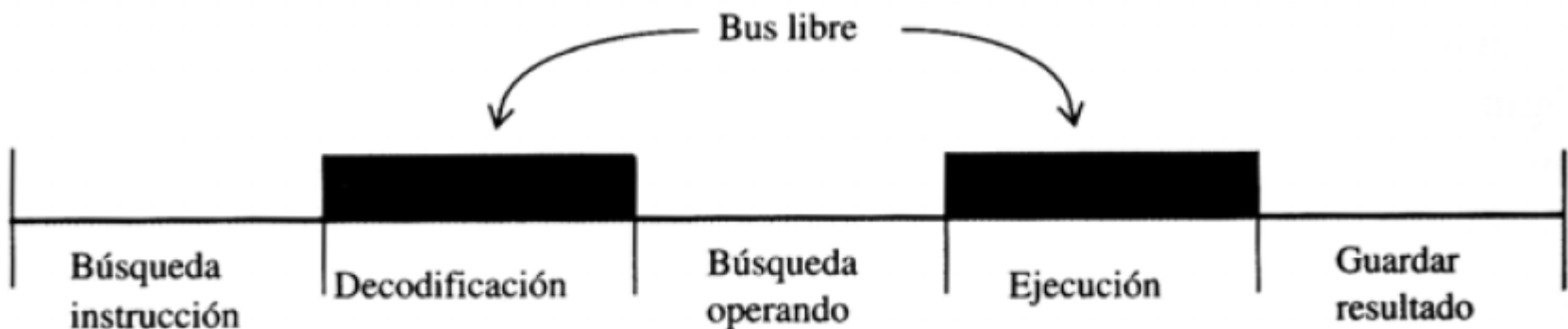
En cada ciclo transmite 8 bits

$$V = 2 \times 10^4 \times 0,89 \times 8 = 142.400 \text{ bit/seg}$$

Respuesta: B

Transparente

- Elimina totalmente la interferencia entre el controlador DMA y la UCP.
- El DMA sólo roba ciclos cuando la UCP no está utilizando el sistema.



EJERCICIO

Un controlador de DMA está transmitiendo palabras a memoria mediante la estrategia *transparente*.

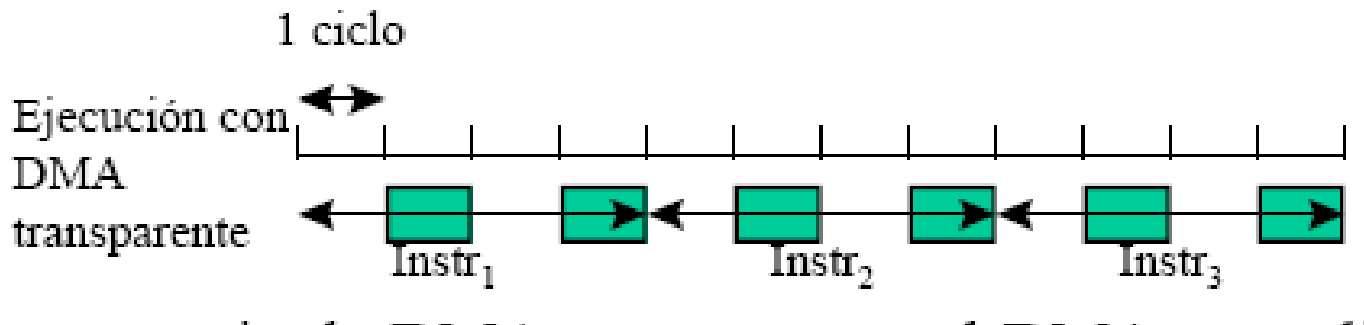
Cuando la CPU dispone en exclusiva del bus las instrucciones emplean, en promedio, 4 ciclos de reloj en ser ejecutadas, aunque en 2 de ellos no se requiere el acceso al bus.

El controlador de DMA tarda 1 ciclo de reloj en transmitir una palabra.

Si la frecuencia de reloj del computador es de 10^6 ciclos/seg,

¿cuánto tiempo tardará el controlador de DMA en mover un bloque de datos compuesto por 10^6 palabras?

DMA transparente, el DMA se realiza en aquellas fases de la ejecución de una instrucción en que la CPU no hace uso del bus



$$R = \frac{2(\text{ciclos} / \text{instr})}{1(\text{ciclo} / \text{pal})} = 2(\text{pal} / \text{instr})$$

La velocidad de ejecución es:

$$v_{\text{ejec}} = \frac{f(\text{ciclos} / \text{seg})}{4(\text{ciclos} / \text{inst})} = \frac{10^6}{4} = 2.5 \cdot 10^5 (\text{instr} / \text{seg})$$

La velocidad de transferencia es:

$$v_T = R \cdot v_{\text{ejec}} = 2(\text{pal} / \text{inst}) \cdot 2.5 \cdot 10^5 (\text{inst} / \text{seg}) = 5 \cdot 10^5 (\text{pal} / \text{seg})$$

Por otra parte, la velocidad de transferencia se define como

$$v_T = \frac{N}{T}$$

donde N es el número de palabras que se transmiten y T el tiempo que tardan en transmitirse. Despejando T de la expresión anterior

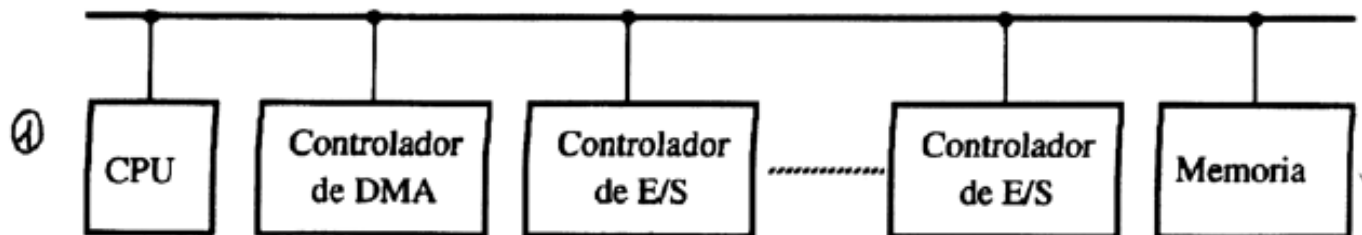
$$T = \frac{N}{v_T}$$

y sustituyendo los valores de N (dado en el enunciado) y v_T , se obtiene:

$$T = \frac{10^6}{5 \cdot 10^5} = 2 \text{ s}$$

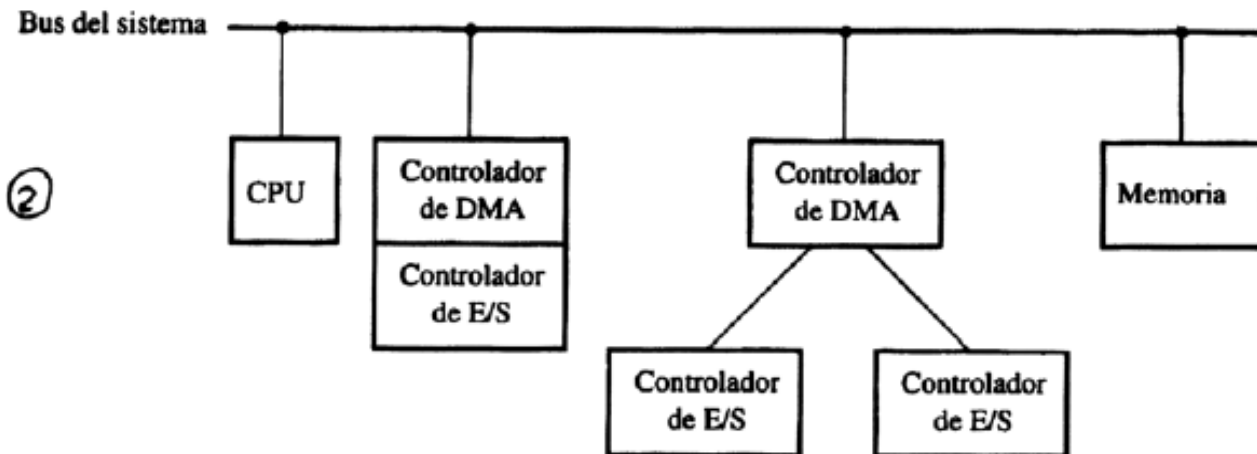
Configuración de la DMA

- Bus único con DMA conectado
 - Todos los módulos comparten el mismo bus del sistema.
 - El controlador de DMA, que actúa como sustituto de la UCP, utiliza E/S controlada por programa para intercambiar datos entre la memoria y un periférico a través del DMA.
 - Esta configuración es económica y poco eficaz, ya que cada transferencia de palabra consume dos ciclos de bus.



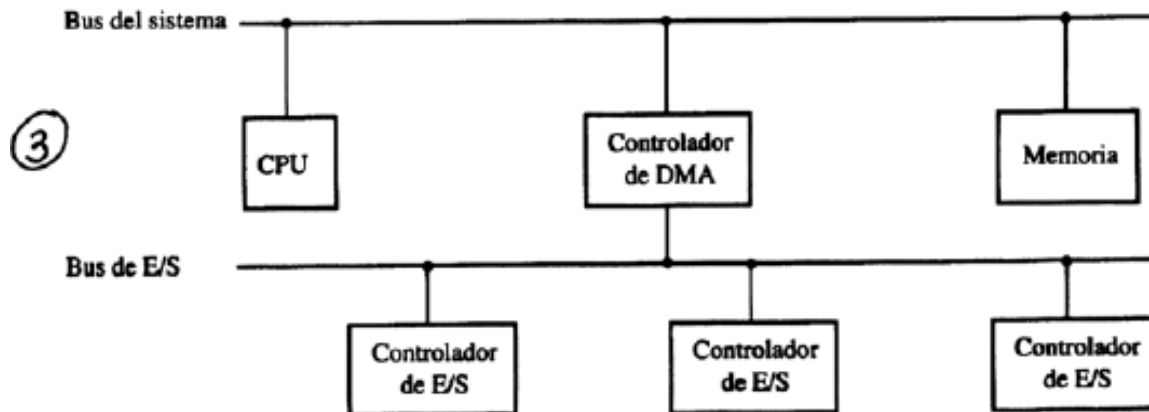
Bus único con DMA integrado

- Las funciones de DMA y E/S se integran, lo que supone que hay un camino entre el DMA y uno o más controladores de E/S que no incluyen el bus del sistema.



Controlador DMA conectado a bus de E/S

- Se utiliza en bus de E/S para conectar los controladores de E/S al controlador de DMA.





Cuestión:

Si TP, TI y TD denotan el tiempo de CPU que se necesita para realizar una transferencia de una sola palabra empleando E/S controlada por programa, con interrupciones y con DMA respectivamente, se verifica la relación:

- A) $TP < TI < TD$
- B) $TP < TD < TI$
- C) $TD < TI < TP$
- D) Ninguna de las anteriores

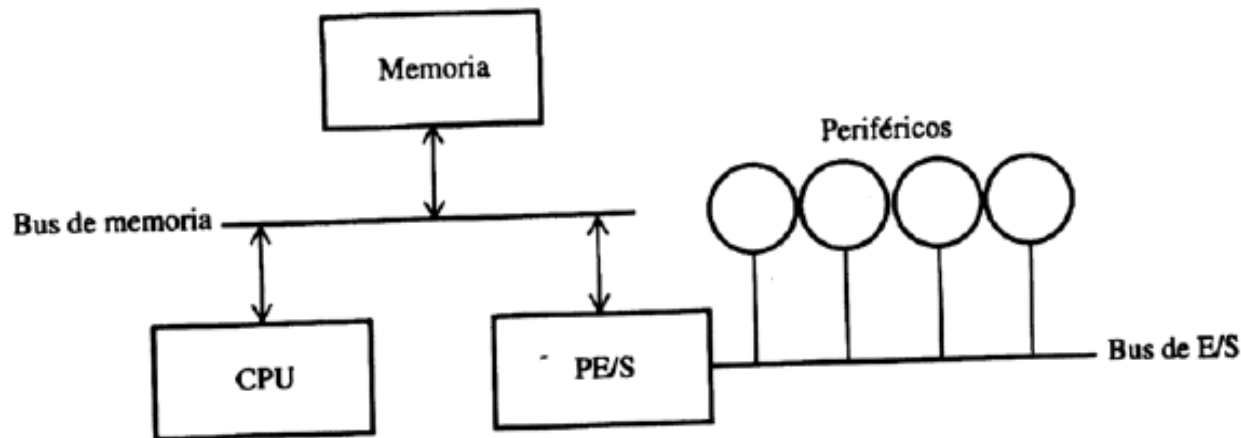
Respuesta:

El tiempo de CPU es mayor en la E/S controlada por programa, inferior en la E/S por interrupciones y la técnica que emplea menos tiempo de la CPU es la E/S por DMA.

Respuesta: C($TD < TI < TP$)

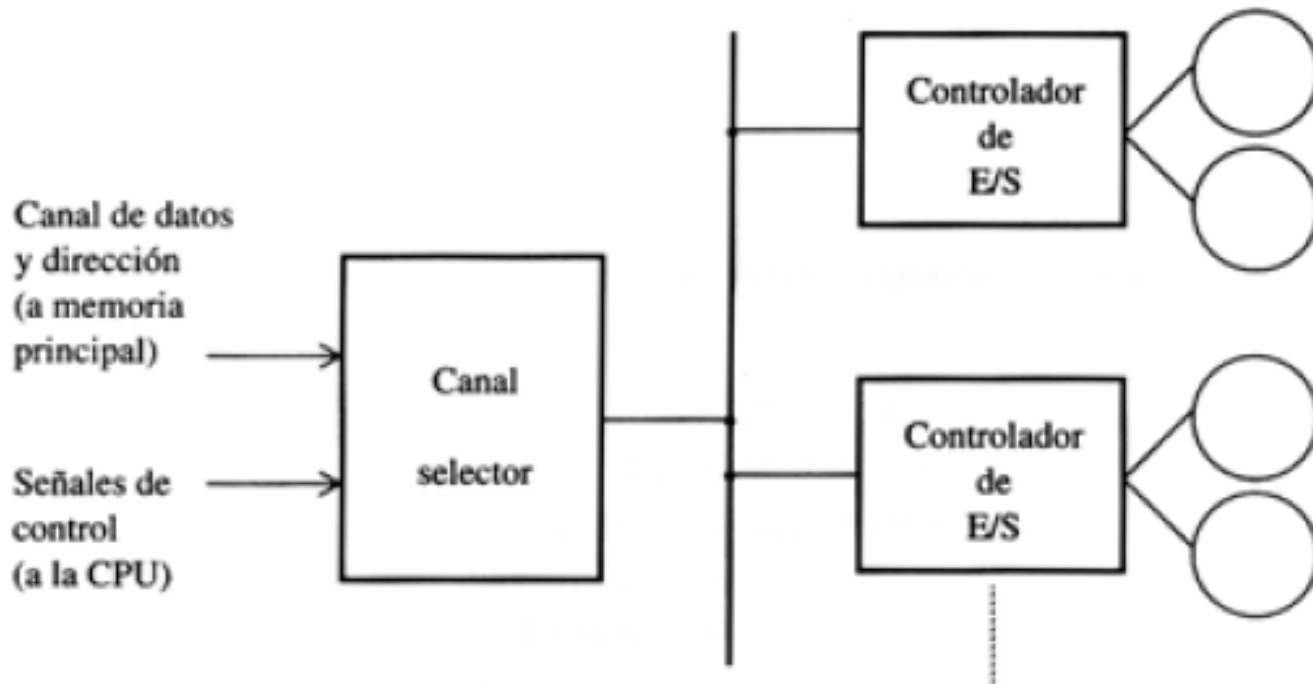
3.6 Procesador de E/S (PE/S)

- El PE/S representa una extensión del concepto de DMA
- Un PE/S tiene capacidad de ejecutar instrucciones de E/S.
- La UCP inicia una transferencia de E/S al dar la orden al PE/S para ejecutar un programa en memoria, el cual especifica:
 - El periférico que interviene
 - La zona de memoria utilizada
 - Las prioridades y
 - Las acciones a realizar en caso de error
- Aumento del rendimiento.



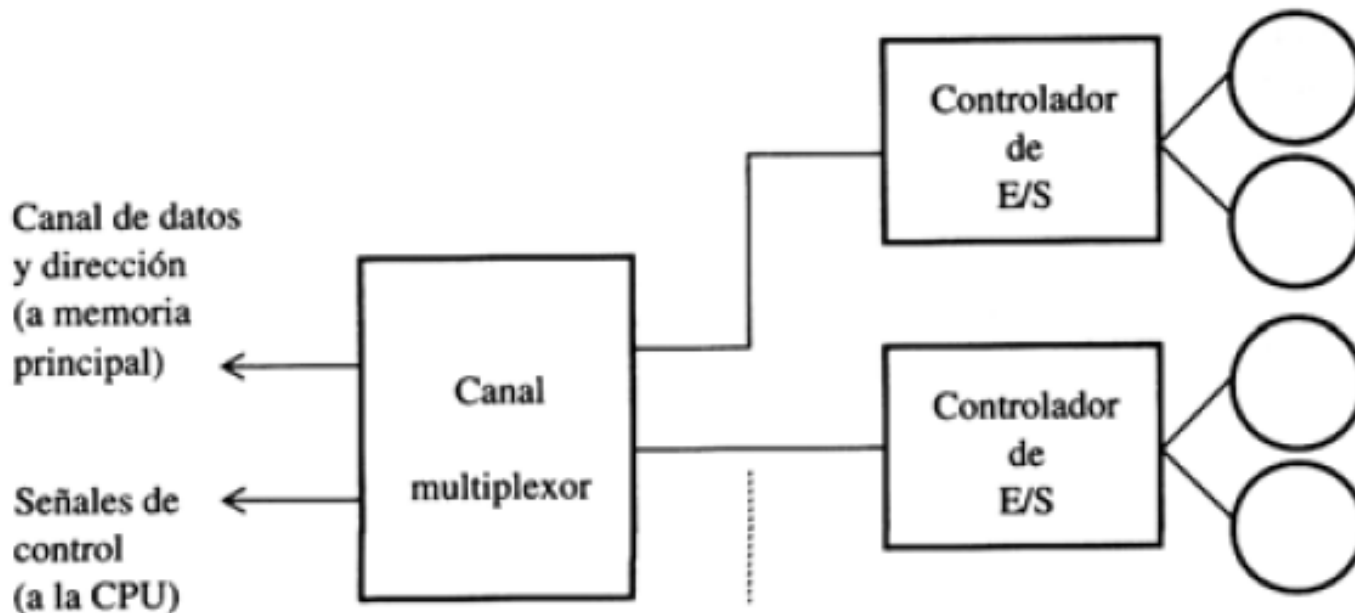
Canal selector

- El PE/S selecciona el periférico que tiene que comunicar y destina todo su tiempo a atenderle, no atendiendo al resto



Canal multiplexor

- El PE/S reparte el tiempo de comunicación de manera secuencial y en espacios cortos entre todos los periféricos, dando la sensación de que está tendiendo a todos de modo paralelo



■ 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 disco800 Kbytes/seg
 - Unidad de cinta magnética.....200 Kbytes/seg
 - Impresora8 Kbytes/seg
 - Lectora de tarjeta1,5 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

3.- Un procesador dispone de ocho líneas de interrupción (numeradas del 0 al 7) y una política en la que las interrupciones con un número bajo tienen mayor prioridad sobre aquellas de número más alto.

El procesador comienza sin interrupciones pendientes y se produce la siguiente secuencia de interrupciones: 4, 7, 1, 3, 0, 5, 6, 4, 2, 1. Suponga que la gestión de una interrupción tarda el tiempo suficiente para que se produzcan dos nuevas interrupciones y que las interrupciones no se pueden interrumpir entre sí. El orden en que se gestionan las interrupciones es:

- A) 4 1 0 3 2 1 4 5 6 7
- B) 4 7 1 3 0 5 6 4 2 1
- C) 0 1 1 2 3 4 4 5 6 7
- D) Todas las afirmaciones anteriores son falsas.

Una forma de resolver este problema es crear una lista de interrupciones pendientes.

Tabla 3.1: Gestión de las interrupciones.

Interrupción atendida	Interrupciones solicitadas	Interrupciones pendientes
4	7, 1	7, 1
1	3, 0	7, 3, 0
0	5, 6	7, 3, 5, 6
3	4, 2	7, 5, 6, 4, 2
2	1	7, 5, 6, 4, 1

Respuesta: A