

## TEMA II. SUBSISTEMAS DE ENTRADA/SALIDA.

### INDICE

- 2.1 Introducción.
- 2.2 Jerarquía en operaciones de E/S.
  - 2.2.1 Transferencia elemental
  - 2.2.2 Transferencia de bloques.
  - 2.2.3 Operación de entrada salida.
- 2.3 Sincronización Periféricos-UCP.
- 2.4 Control de periféricos.
- 2.5 Elementos implicados en operaciones de entrada salida.
- 2.6 Mecanismos de Transferencia elemental.
  - 2.6.1 Instrucciones de E/S. E/S programada.
  - 2.6.2 Instrucciones de control de estado.
- 2.7 Gestión de Prioridades.
- 2.8 Gestión Interrupciones.
- 2.9 Accesos concurrentes con UCP.
  - 2.9.1 Acceso directo a memoria.
  - 2.9.2 Canales de entrada/salida.

## 2.1 Introducción.

Los sistemas (o subsistemas) de entrada/salida tienen una gran importancia por ser los responsables de la interacción del computador con los dispositivos periféricos. Algunos de estos son los que nos permiten comunicarnos con el computador.

Su correcto diseño es de vital importancia pues incide directamente en la velocidad de funcionamiento de la CPU y de los programas que necesitan usar estos periféricos.

El sistema de entrada salida, puede ser visto como un usuario más que hace uso de la CPU y, en algunos casos, con mayor prioridad que el usuario humano del computador.

Esto nos lleva a introducir diversos mecanismos de interrupción, asignación de prioridades, rutinas de tratamiento de interrupciones, señales de control y sincronización del periférico y de las comunicaciones.

Es imposible dar una visión completa de la E/S sin hablar de Sistemas Operativos. Muchas de las operaciones de entrada salida se resuelve mediante el software del SO correspondiente.

Para comprender la complejidad de la entrada/salida es necesario tener en cuenta que la velocidad de los periféricos es substancialmente menor que la velocidad de la CPU, además, en la mayoría de los casos, el ancho de palabra es de un octeto, lo cual no coincide con el ancho de palabra de los procesadores actuales. Además, los periféricos, por tener un funcionamiento autónomo por definición, pueden tener interacciones simultáneas con la CPU dando lugar a problemas de concurrencia.

El conjunto de operaciones relacionadas con la E/S se reparte entre el controlador del periférico, la UCP y los programas de E/S.

## 2.2 JERARQUÍA DE OPERACIONES DE E/S.

Este conjunto de operaciones a realizar se pueden clasificar en tres tipos:

- *Transferencia Elemental.* Se establece para la transmisión de un bit, un byte o una palabra. Esta información puede ser un Dato o información de control.
- *Transferencia de bloque.* Se establece para la transferencia de un conjunto de datos, generalmente el tamaño del bloque es proporcional a algún tamaño físico, por ejemplo, la longitud de un sector de una unidad de disco.
- *Operación de E/S.* Es una operación completa que contempla, generalmente, un número entero de transferencias de bloques.

### 2.2.1 Transferencia Elemental.

- Puede requerir una conversión serie/paralelo o viceversa.
- Se realiza a la velocidad del dispositivo físico de interconexión, generalmente un BUS. La una duración entre 10 y 100ns.

- La UCP ha de estar diseñada para activar las señales de control del periférico, así como tener capacidad para leer la información de estado del periférico.
- Hay distintos mecanismos físicos de interconexión. Se diferencian en las señales de sincronización que emplean y en los cronogramas resultantes.

### 2.2.2 Transferencia de Bloque.

- Exige la sincronización correcta entre la UCP y el periférico en el sentido de que ambos difieren notablemente en sus velocidades de proceso.
- Ha de llevarse cuenta de los octetos o palabras transmitidas.
- Ha de ser capaz de soportar la detección de errores (control de paridad, polinomiales...). En caso de error ha de ser capaz de retransmitir.
- Hace uso de buffer temporales (zonas de memoria intermedias).
- Este tipo de transferencias suelen sacar el máximo provecho del ancho de banda, dando lugar al concepto de **“ancho de banda sostenido”**.

### 2.2.3 Operación de entrada/salida.

- Gestiona y supervisa las transferencias de bloques y el estado del periférico.
- Realiza el tratamiento de los errores y lleva la cuenta de bloques transferidos.
- Realiza las conversiones de códigos.
- Inicia y termina la comunicación.
- Una operación completa de Entrada/Salida que conlleve una transferencia elemental o una transferencia de bloque, como las aquí descritas, recibe el nombre de **“transacción”**.
- EL tiempo total de una operación de Entrada/Salida recibe el nombre de **“Latencia”** o **“Elapsed Time”**

## 2.3 SINCRONIZACIÓN PERIFÉRICO-UCP.

*Básicamente consiste en establecer un mecanismo que permita adecuar de forma eficiente las diferentes velocidades entre ambos elementos.*

Hay dos formas de realizar esta tarea:

- La UCP pregunta periódicamente por el estado del periférico para ver si ha de enviar o recibir algún dato (**Polling** o **sondeo**). Tiene el inconveniente que mientras dure la operación de E/S la UCP está dedicada en exclusiva al periférico.
- La UCP es avisada por el controlador del periférico mediante una interrupción. El inconveniente radica en que aumenta el grado de complejidad de la operación de E/S al tener que arbitrar un sistema de interrupciones y prioridades.

## 2.4 CONTROL DE LOS PERIFÉRICOS.

Para el control del periférico podemos distinguir dos clase de señales:

- Señales de estado.
- Señales de control.

Estas señales están agrupadas en registros de estado y de control, normalmente en el controlador del periférico.

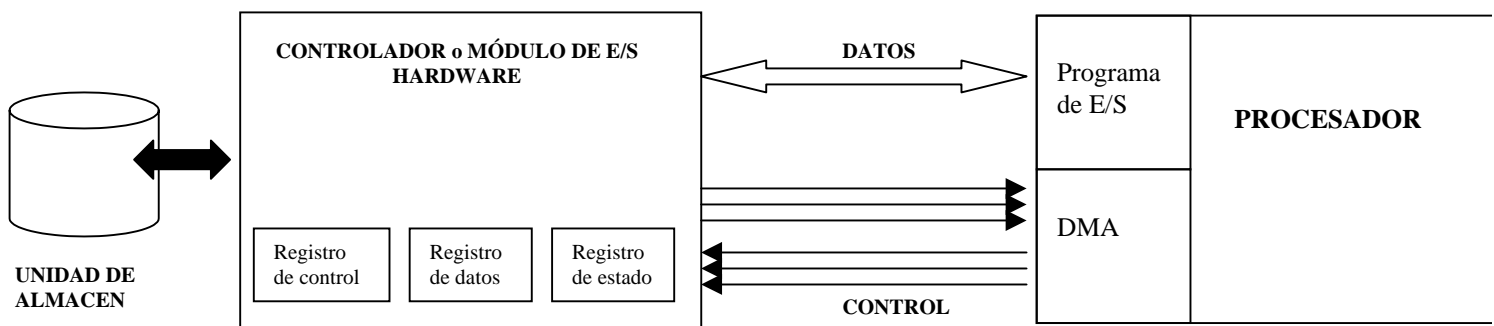
Señales típicas de estado:

- Nuevo dato disponible.
- Operación terminada.
- Periférico encendido o apagado.
- Periférico operando.
- Periférico operativo o no.
- Error de operación.
- Error de paridad.

Señales típicas de control:

- Encender o pagar.
- Seleccionar opciones.
- Saltar página.
- Buscar marca.
- Ubicar brazo.
- Etc.

## 2.5 ELEMENTOS CARACTERÍSTICOS DE E/S.



El dispositivo periférico, en este caso, se ha elegido una unidad de almacenamiento como por ejemplo un disco duro.

El periférico necesita de una cierta circuitería controladora, denominada “controlador del periférico”.

El periférico, en este caso un disco, consta de:

- Motor que hace girar el disco a velocidad constante.
- Motor que acciona el brazo.
- Elementos electrónicos de lectura y escritura.
- Elementos electrónicos de accionamiento de motores.

El circuito controlador consta de:

- Registro de control.
- Registro o registros de datos. (Búferes).
- Registro de estado.

## 2.6 MECANISMOS DE TRANSFERENCIA ELEMENTAL.

Existen dos: Instrucciones de entrada salida y acceso directo a memoria.

### 2.6.1 Instrucciones de E/S.

Son instrucciones máquina, como otras del repertorio de instrucciones. Contienen:

- **Información de dirección** de intercambio con periférico. Se selecciona el registro del controlador del periférico destino u origen del dato, o la dirección de la zona de intercambio de información entre periférico y UC.
- Si es transferencia del computador hacia el periférico, hay información de su dirección.
- **Tipo de operación.** Lectura o escritura.
- **Temporización o Sincronización** Establecimiento de camino físico y Cronograma de instrucción de E/S

#### Direccionamiento.

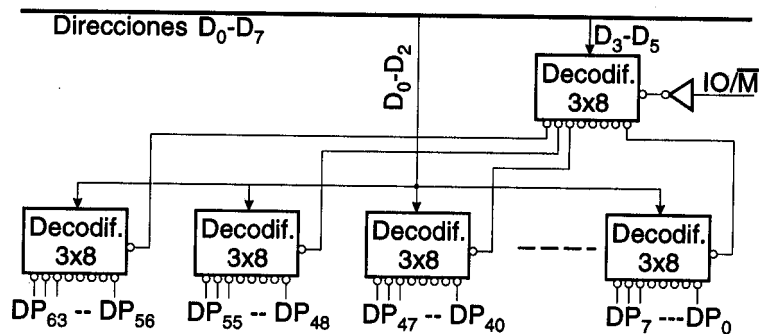
La información de dirección ha de servir para seleccionar, no sólo el registro, sino además, el periférico correspondiente.

El conjunto de direcciones de E/S conforman el llamado **mapa de direcciones de entrada salida**. Una dirección recibe el nombre de **puerta (o puerto) de entrada/salida**. Estas mismas direcciones pueden hacer referencia a posiciones de memoria principal en caso de que el periférico carezca de registros o buffer.

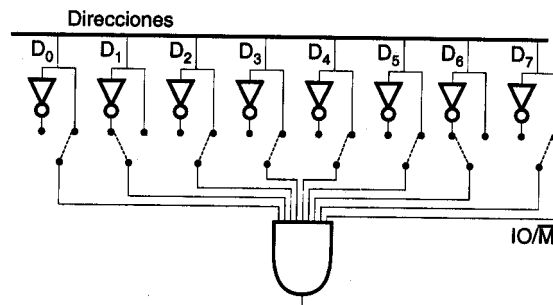
Un periférico puede disponer de una o más direcciones de entrada salida en función de su número de registros de control, dato o estados direccionables.

Hay dos métodos de decodificación de direcciones:

- Decodificación centralizada.



- Decodificación de cada puerta de entrada salida.



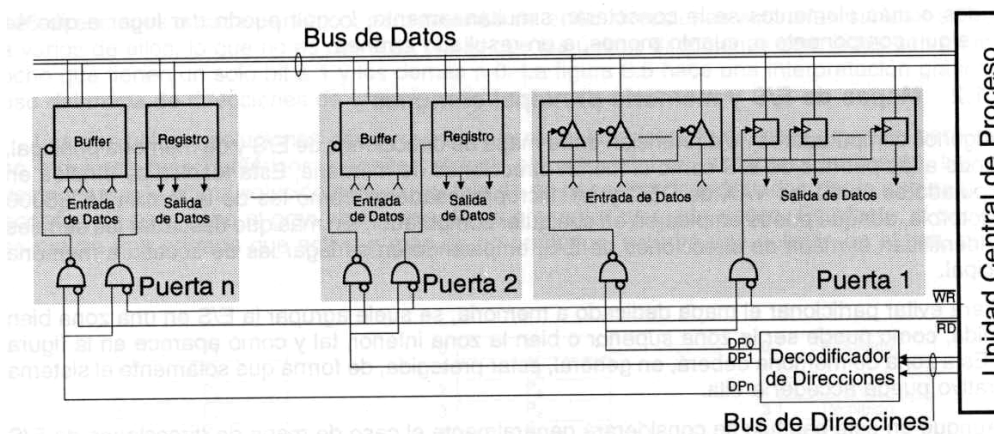
Señales de datos.

Los datos de cada periférico se pueden transmitir o, o bien, por una arquitectura en bus, o bien, por una arquitectura multiplexada.

#### 2.6.1.1 Transferencia de datos con arquitectura en Bus.

## Arquitectura en BUS

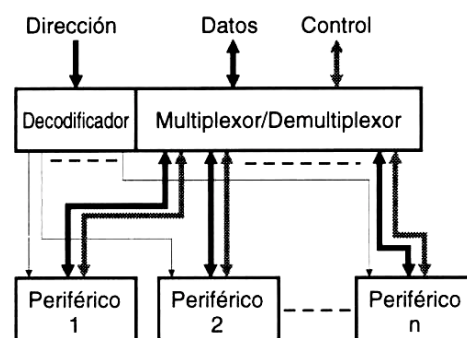
Los datos pueden ir o por líneas bi-direccionales o por dos conjuntos de líneas unidireccionales. A continuación podemos observar un ejemplo de bus bi-direccional con selección de puerto centralizada.



El bus de datos puede ser el mismo que el interno del procesador, pero con una etapa aisladora capaz de independizar los datos correspondientes a interrupciones de entrada salida de datos del repertorio de instrucciones del procesador.

#### 2.6.1.2 Transferencia de dato con multiplexación.

Las líneas de datos de cada periférico están independizadas. A continuación podemos observar un ejemplo de conexión multiplexada en datos hacia o desde el periférico:

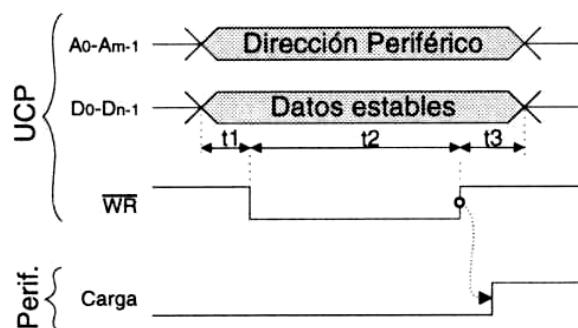


#### Señales de Control.

Las señales de control indican si la operación es de lectura del periférico (ENTRADA) o escritura en el mismo (SALIDA) además de sincronizar los procesos.

En cuanto a la sincronización, hay tres métodos:

- **Transferencia Síncrona:** Es la unidad de control la que marca el ritmo de la conexión a través de sus flancos de subida y bajada. Es la unidad de control quien lleve la iniciativa.

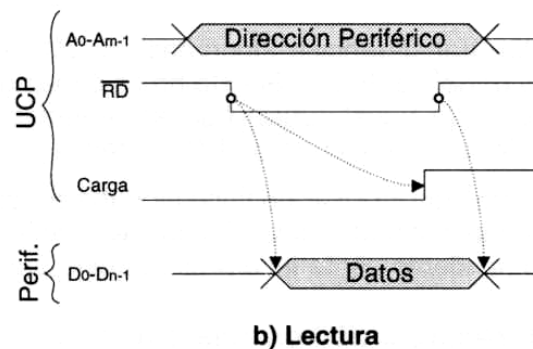


#### a) Escritura

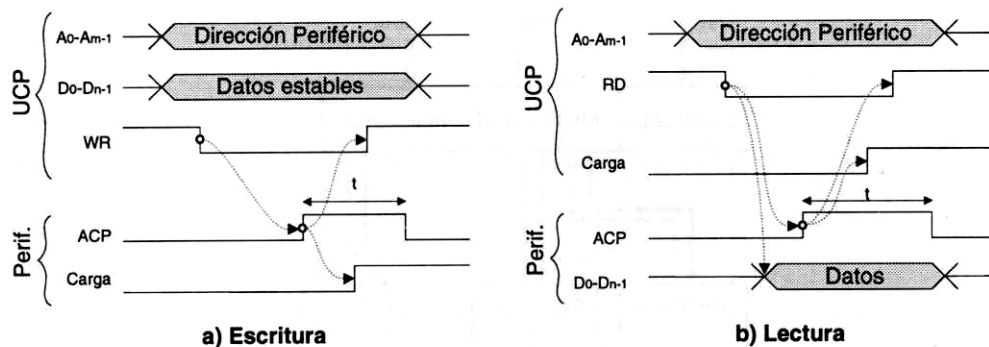
En operación de **escritura**, la unidad de control pone los datos en el bus y activa a baja la señal de escritura. Una vez que los datos se han estabilizado en el bus de datos, la señal de escritura se pone en alta

provocando de forma sincronizada, el flanco de carga del registro del periférico. De forma sincronizada significa, con la llegada del siguiente ciclo de reloj. Aunque esta señal no esté dibujada, es imprescindible en las transferencias síncronas. También puede haber, como ya vimos en el tema anterior, una señal de ACK por parte del esclavo, a intervalo de tiempo fijo, indicando que inicia la lectura.

En **lectura**. La activación a baja de la señal de lectura del periférico por parte de la UC, provoca que el periférico ponga los datos en el bus de datos. Y simultáneamente, la señal de carga en algún registro gobernado por la unidad de control. Cuando la unidad de control pone en alta la señal de lectura, entonces el periférico retira los datos del bus. En este caso, también los ciclos de reloj son imprescindibles. Puede haber un ACK por parte del periférico indicando que ha iniciado la puesta de datos en el BUS.



- Transferencia asíncrona. Es el periférico el que, de forma asíncrona, acepta (ack, acknowledge), la operación indicada por la unidad de control. Podemos distinguir sin interbloqueo y con interbloqueo.

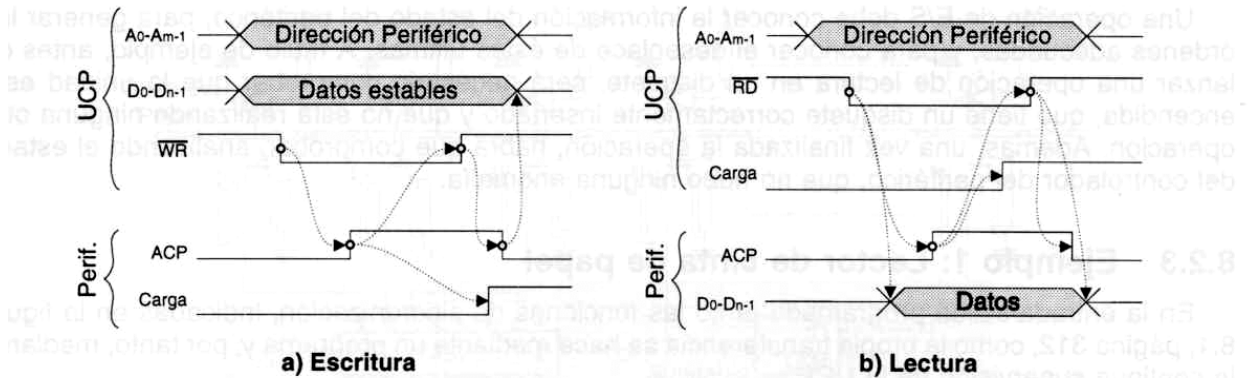


Cuando no hay interbloqueo, el periférico emite un ack cuando está listo para dar un dato o tomarlo. Es una señal activa a alta y asíncrona, que desencadena la señal de carga del registro correspondiente. La operación dura un tiempo máximo suficiente para que se establezcan los datos en destino. Esto significa que el funcionamiento independiente del periférico no bloquea la UCP una vez que esta recibe el ACK. Ahora bien, no es ninguna garantía de éxito debido a que, a partir de la llegada del ACK, la operación está temporizada, por lo que la UCP puede retirar los datos del bus en caso de escritura (o el periférico en caso de lectura) aunque el periférico no haya terminado de leer.



No hay garantía de finalización correcta.

En la transferencia asíncrona con interbloqueos, el tiempo de la operación viene fijado por el encadenamiento de flancos, incluyendo la finalización completa de la transferencia.



Cuando el periférico da el ACK, la unidad de control pone en alta las señales de control que indican lectura o escritura. A su vez, la puesta en alta de estas señales, marca el fin de la puesta de datos en el bus. En este caso está garantizada la operación.

En las transferencias asíncronas, el computador puede quedar bloqueado esperando el ack.

En las transferencias asíncronas con interbloqueo, además, puede quedar bloqueado esperando los flancos de finalización.

El éxito de estas temporizaciones requiere el empleo de módulos Software del Sistema Operativo que aborten las operaciones por "Time out" (tiempo de espera excedido)

### 2.6.2. Instrucciones de control de estado.

El estado y control del periférico puede ser gestionado mediante dos mecanismos diferentes:

- El periférico ha de tener una puerta de E/S para los datos de estado y de control, y otra puerta para datos. De esta forma, cuando quiere saber el estado, lee esa puerta y cuando quiere mandar órdenes, escribe en la misma puerta
- El periférico tiene puertas de E/S para control y estado.

En ambos casos, pueden existir instrucciones de entrada/salida del tipo TEST, para conocer el estado y CONTROL. Estas instrucciones han de especificar la dirección del puerto que se desea leer.

## 2.7 GESTIÓN DE PRIORIDADES.

Cuando varios periféricos desean hacer uso de la UCP, es necesario establecer un mecanismo de prioridades.

Básicamente hay dos sistemas:

- Gestión Distribuida. Entre los peticionarios se ponen de acuerdo .
- Gestión Centralizada. Hay maestro que es el que adjudica el servicio.

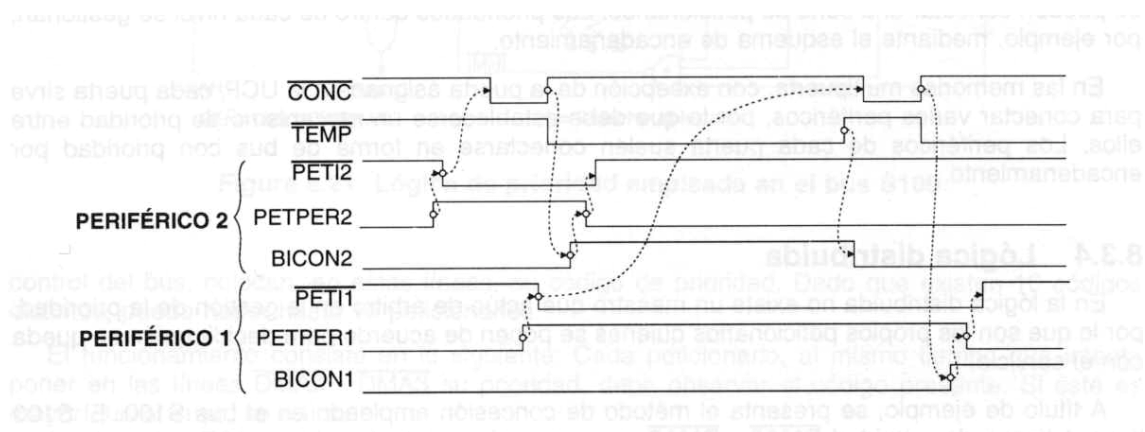
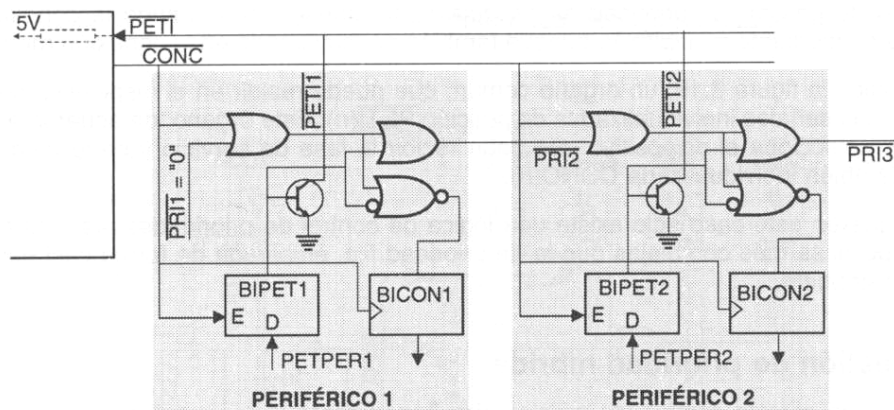
Ejemplos:

### **Encadenamiento o daisy-chain.**

Los periféricos están dispuestos en cadena de mayor a menor prioridad. El más cercano físicamente a la UCP es el que más prioridad tiene.

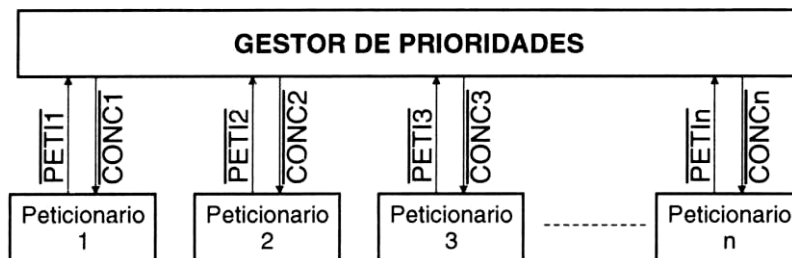
Se permiten varias peticiones simultáneas.

La concesión del servicio está centralizada pero el arbitraje de quien se queda con la concesión depende de la posición del periférico.



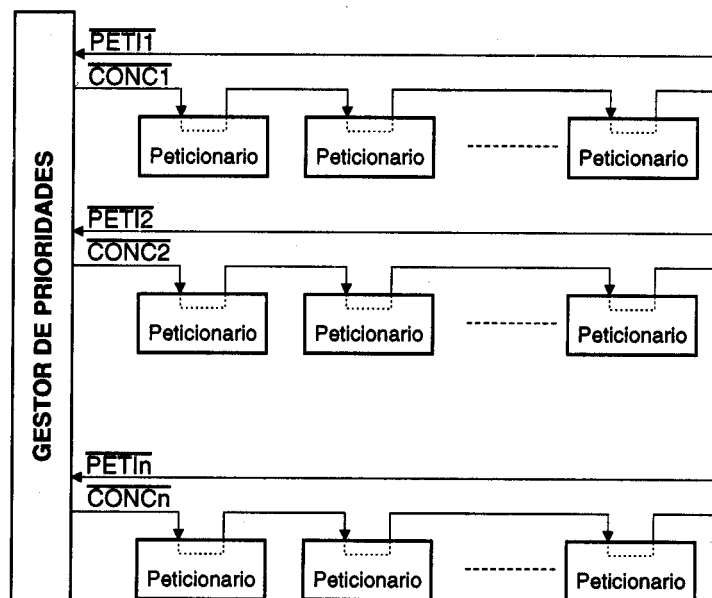
### ***Gestor de prioridad centralizada con señales individuales.***

Hay gestor de prioridades que analiza las peticiones y arbitra las concesiones de forma individualizada.



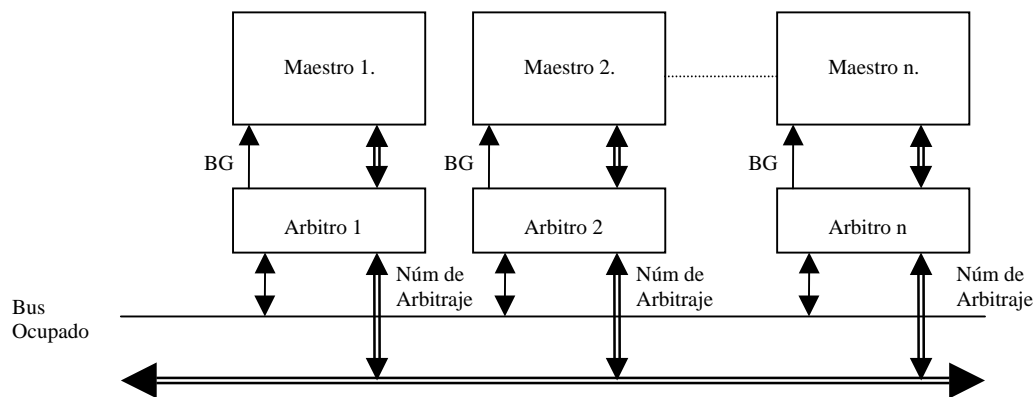
### ***Gestión de prioridades híbrida.***

Hay varias líneas de conexión para los periféricos. En cada línea se tiene una gestión por encadenamiento.



### ***Lógica distribuída.***

Cada periférico tiene un identificador que coincide con su prioridad. Cuando solicita el servicio, pone su identificador a la vez que compara con el identificador de otro, si lo hubiera. Si tiene menos prioridad, se retira. (Ya visto en el tema I de Buses).



## **2.8 GESTIÓN DE INTERRUPTIONES.**

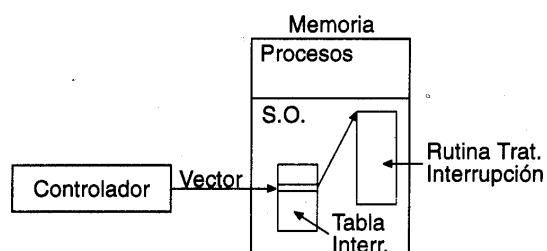
La interrupción es una señal externa emitida por un dispositivo ajeno al procesador y que provoca una bifurcación a una dirección diferente de las del programa que se encuentra en ejecución.

### ***Interrupciones vectorizadas***

En este caso, los periféricos que interrumpen tienen que suministrar la dirección en donde se encuentra la rutina de tratamiento de la interrupción.

Los mecanismos de obtención de esa dirección son tres:

- Es una dirección fija que corresponde con un programa, que a su vez discrimina el tipo de interrupción y proporciona la dirección de la rutina que realiza el tratamiento.
- La dirección la proporciona el dispositivo que interrumpe.
- El dispositivo que interrumpe proporciona parte de la dirección. Es el caso de tratamiento vectorizado.



Los problemas que ha de resolver el mecanismo de tratamiento de interrupciones son:

- Identificación del peticionario.
- Selección de la rutina de tratamiento.
- Desactivación de la solicitud de interrupción.
- Superposición de peticiones.
- Resolución de prioridades.
- Anidamiento de interrupciones.

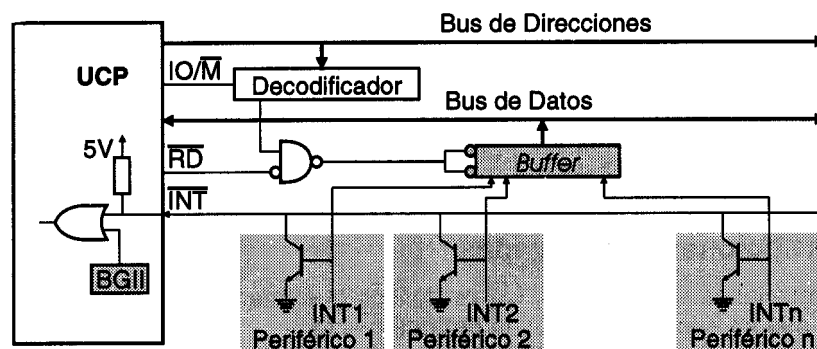
**Ejemplo: Línea de interrupción única.**

Es una línea que implementa un OR cableado. De forma que varios dispositivos puedan activar la petición simultáneamente.

La UCP, una vez que recibe la petición de interrupción, inhabilita otras solicitudes mediante un bi-estable, llamado bi-estable general de inhibición de interrupciones.

Dado que no hay más señales, se salta a una dirección fija.

Para identificar el dispositivo, la UCP inicia un ciclo de polling (muestreo).



Este sistema se puede emplear para gestionar prioridades en daisy-chain sin mas que añadir una línea de aceptación de la interrupción.

**Mecanismos de enmascaramiento de interrupciones.**

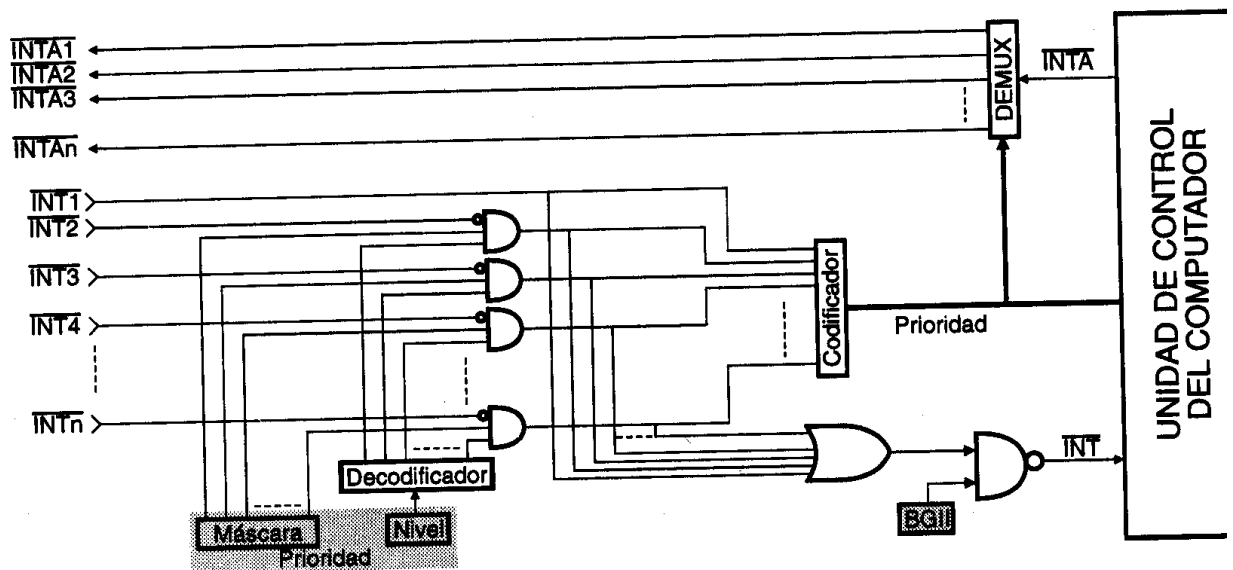
Cuando se produce una interrupción, pueden ocurrir que se nos den las siguientes situaciones:

- Que nos interese tratar esa interrupción sin atender a ninguna otra mientras tanto.

- Que el proceso de tratamiento de interrupción, pueda ser, a su vez, interrumpido según quien interrumpa.
- Que pueda ser interrumpido siempre.

Para ello hay tres mecanismos que se pueden combinar en una solución única:

- Biestable general de inhabilitación de interrupciones.
- Máscara que inhabilita alguna interrupción.
- Nivel de interrupción. Inhabilita interrupciones de menor prioridad.



### ***Tratamiento común a todas las interrupciones:***

- La UCP detecta presencia de señal de interrupción y la acepta.
- Salva el registro o palabra de estado.
- Salva el Contador de programa.
- Pone el nivel de interrupción en el máximo o activa el bi-estable de inhibición general.
- Obtiene la dirección de bifurcación.
- Bifurca.
- Realiza el tratamiento.
- Restaura el contenido de palabra de estado y contador de programa.

## 2.9 E/S CONCURRENTES CON LA UNIDAD DE CONTROL DE PROCESO.

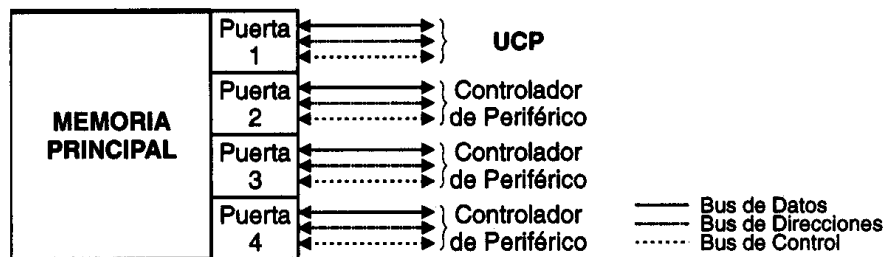
### 2.9.1 Acceso directo a memoria.

El controlador del periférico realiza accesos a memoria sin el concurso de la unidad de control. Para ello es necesario que el controlador disponga de un bloque de información de control (bloque de control) alojado en sus buferes con información del tipo: número y tamaño de datos a enviar o recibir, y direcciones a leer o escribir.

Hay dos mecanismos:

### **Memoria Multipuerta.**

La memoria permite accesos simultáneos con buses de datos y direcciones independientes. En caso de colisión se producen retardos.



Son memorias caras.

### ***Robo de Ciclo.***

La UCP, tras recibir una petición de interrupción del periférico, le cede el control de los buses y la memoria. La secuencia de operaciones es la siguiente:

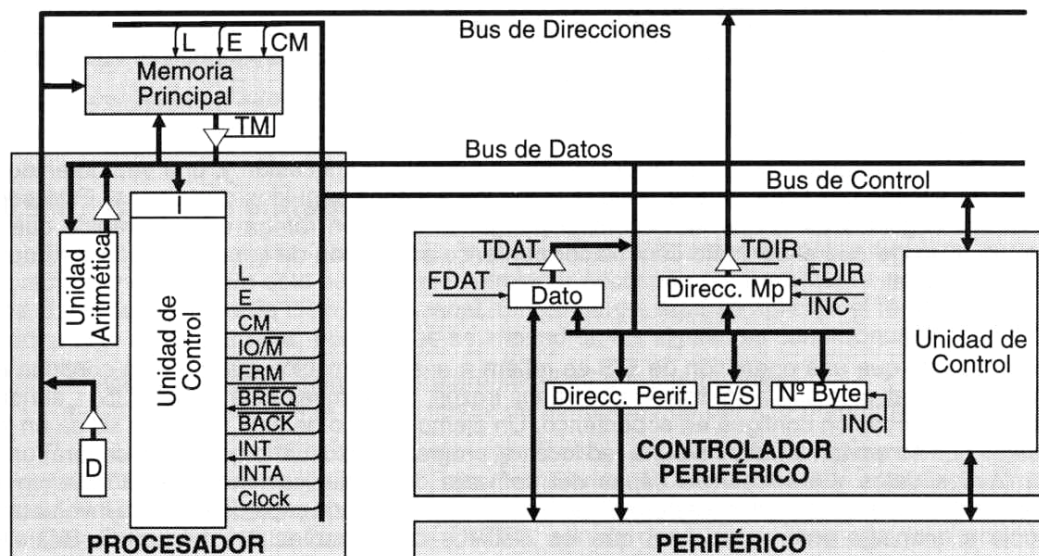
El controlador del periférico solicita a la UCP el acceso a memoria principal (solicita el robo de uno o varios ciclos). (Señal **"Bus Request"**).

La UCP contesta con la señal “**Bus acknowledge**”. Y pone en alta impedancia sus buses y señales de acceso a memoria. También le transmite las direcciones a las que debe acceder el controlador del periférico.

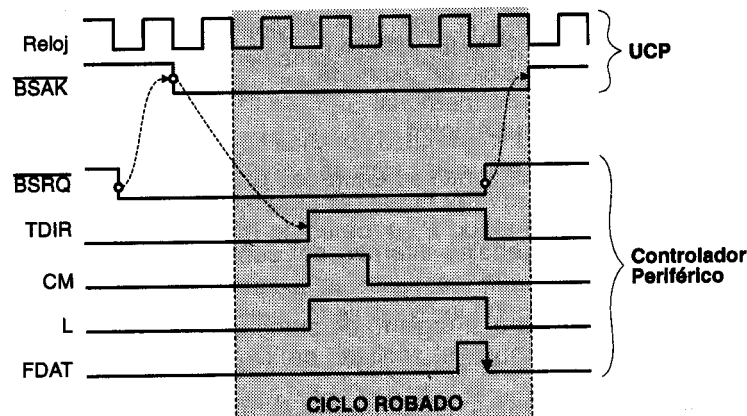
La unidad de control del periférico hace los accesos como los haría la unidad de control del procesador. Puede realizar una transferencia por cada ciclo robado o realizar una transferencia en **ráfaga**, trayéndose bloques completos.

La unidad de control del periférico desactiva la señal “**Bus Request**”.

La unidad de control del procesador desactiva “**Bus Acknowledge**”.



El cronograma correspondiente podría ser:



Independientemente del mecanismo, podemos distinguir cuatro tipos de DMA:

- **DMA simple.** Cuando la transferencia sólo se puede hacer de una palabra por cada robo de ciclo o acceso. Teniendo que negociar un nuevo acceso DMA por cada palabra a transferir.
- **DMA en ráfaga.** Una vez negociado y concedido el acceso, se tiene en exclusiva mientras haya palabras que enviar. Estas se envían una de tras de otra sin que medien negociaciones (ráfaga).
- **DMA disperso.** En el caso de ráfaga, las direcciones de memoria accedidas son consecutivas, no así en el DMA disperso. En este caso, el bloque de información de control del DMA ha de contener las direcciones que serán accedidas, o bien, en lectura (gathering=recolección), o bien, en escritura (scattering=dispersión).
- **DMA gobernado por memoria.** En este caso el bloque de control no se envía por la UCP a los buferes del controlador de E/S directamente, sino que se encuentra escrito en memoria, a donde accede el controlador para leer la información de control que necesitará. Esto permite encadenar operaciones de E/S solapadamente con la ocurrencia de una.



### 2.9.2 Canales de entrada salida.

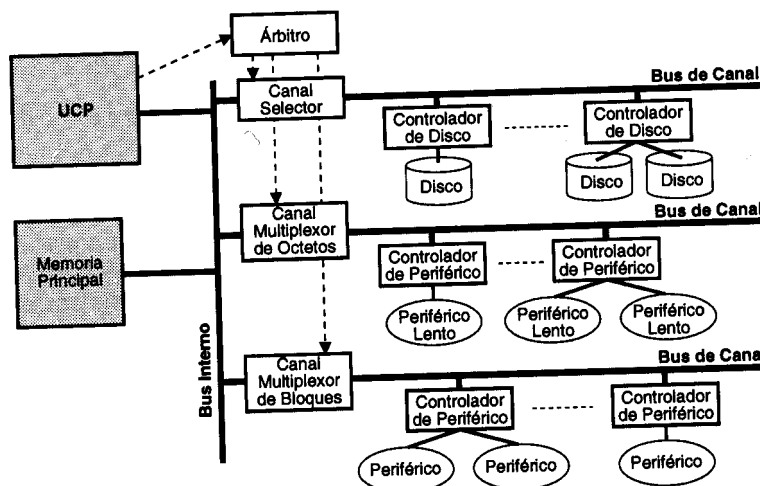
Es una ampliación del concepto de DMA. El canal es un **procesador** que se conecta al computador principal como si fuera un periférico, aunque en realidad es un procesador especializado en ejecutar instrucciones de entrada/salida para familias completas de periféricos. La CPU inicia la operación de E/S indicándole al canal que ejecute un programa de E/S que está en su memoria.

Hay canales **selectores** y canales **multiplexores**.

Los selectores se emplean con periféricos de alta velocidad permitiendo sólo una operación al tiempo.

Los canales multiplexores se emplean con periféricos lentos manteniendo varias operaciones de entrada salida simultáneas.

El gráfico siguiente muestra un ejemplo de estructura de E/S del IBM 370/360 (modelo un tanto antiguo), con canales de ambos tipos:



### **EJERCICIOS DE ENTRADA/SALIDA.**

1. Un módulo DMA está transfiriendo caracteres a memoria mediante robo de ciclo desde un dispositivo que transfiere a 9600 bps. La CPU capta instrucciones a una velocidad de 1 Millón de instrucciones por segundo (1MIPS). ¿En qué medida se reduce la velocidad del procesador debido a la actividad de DMA? .
2. Un computador de 32 bits tiene dos canales selectores y un canal multiplexor. Cada canal selector soporta dos discos magnéticos y dos unidades de cinta magnética. El canal multiplexor tiene conectadas dos impresoras de línea, dos lectoras de tarjetas y 10 terminales VDT.

Suponga las siguientes velocidades de transferencia:

- Unidad de disco 800KB/s.
- Unidad de cinta magnética 200 KB/s.
- Impresora de línea 6,6 KB/s.
- Lector de tarjetas 1,2 KB/s.
- VDT 1KB/s.

Estime la velocidad total de transferencia de E/S en el sistema.

3. Un computador está constituido por un procesador y un dispositivo D de E/S, conectado a la memoria principal M a través de un bus compartido de una palabra. El procesador puede ejecutar un máximo de 1 millón de instrucciones por segundo. Por término medio, las instrucciones necesitan cinco ciclos máquina, tres de los cuales utilizan el bus de memoria. Una operación de lectura o escritura en memoria emplea un ciclo máquina. Suponga que el procesador se encuentra ejecutando continuamente ejecutando programas en segundo plano que requieren el 95% de la velocidad de ejecución de sus instrucciones, pero ninguna operación de E/S. Asuma que un ciclo de procesador es igual a un ciclo de BUS. En un momento dado, el dispositivo de E/S se utiliza para transferir bloques muy grandes de datos entre la memoria principal M y D.
  - Si la entrada salida es programada, y cada transferencia de una palabra requiere que el procesador ejecute dos instrucciones, estime la máxima velocidad (en palabras por segundo) de transferencia de datos de entrada salida posible a través de D.
  - Estime la misma magnitud si se utiliza DMA.
4. En un bus SCSI , cada dispositivo de E/S negocia con el computador anfitrión para determinar la velocidad de transferencia a ráfagas a utilizar (usualmente la velocidad más rápida de las mutuamente soportadas). Asuma que la velocidad de transferencia a ráfagas que permite el computador anfitrión es 20MB/s. Suponga que todos los dispositivos tienen los buffers suficientes para poder mantener su velocidad de transferencia sostenida, incluso cuando compiten por tiempo de bus.
  - Asuma que un dispositivo de cinta con una velocidad de transferencia sostenida de 500KB/s y una velocidad de ráfaga de 4 MB/s está conectado al SCSI. Desea conectar discos al mismo bus, cada uno con una velocidad de transferencia sostenida de 6MB/s y una velocidad de ráfaga de 20 MB/s. ¿Cuántos dispositivos de disco puede conectar al bus si quiere que todos los dispositivos puedan

- funcionar simultáneamente y a toda su velocidad? ¿Cuál sería la utilización del bus en este caso?
- Mejoremos ahora el modelo. Asumamos que el dispositivo de cinta necesita 4 ms del tiempo del bus para cada transferencia, y que el tamaño máximo de transferencia son 64KB. Los dispositivos de disco también necesitan 4 ms de bus en cada transferencia, pero pueden transferir hasta 256 KB por petición. Vuelva a calcular la utilización del bus que hace cada dispositivo, y la utilización total. ¿Sigue siendo el bus adecuado para el número de dispositivos indicado en a)?
5. Asuma que un procesador de 16 bits y dos de ocho bits deben conectarse a un bus del sistema. Considere los siguientes detalles:
- Todos los microprocesadores tienen el hardware necesario para cualquier tipo de transferencia: E/S programada, E/S mediante interrupciones, y DMA.
  - Todos los microprocesadores tienen un bus de direcciones de 16 bits.
  - Hay dos tarjetas de memoria de 64 KB cada una conectadas al bus. El diseñador desea que se comparta la mayor cantidad de memoria posible.
  - El bus del sistema permite hasta un máximo de cuatro líneas de interrupción y una de DMA. Haga las suposiciones adicionales que necesite y:
    - i. Establezca las especificaciones del bus en términos de número de líneas.
    - ii. Explique como es la interfaz de los dispositivos que han de conectarse al bus.
6. Se desea dotar a un edificio de 100 sensores de “efecto doppler” (Señal de salida 0V. O 5V) Se quieren controlar mediante un procesador de ancho de palabra de 8 bits.

a) Diseñar la E/S del sistema.

7. Disponemos de un pequeño computador con los siguientes elementos:

A<sub>0</sub> – A<sub>7</sub>, 8 líneas de dirección para seleccionar el periférico.

D<sub>0</sub> – D<sub>7</sub>, 8 Líneas de datos.

L Una línea para lectura.

E Una línea para escritura.

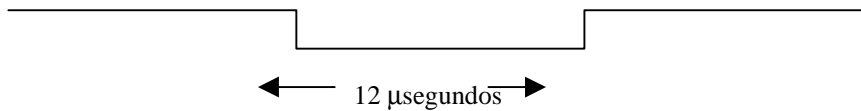
El computador dispone de las instrucciones IN dirección y OUT dirección, para leer o escribir un periférico, el cual intercambia 1 byte con el registro R1 del computador.

Se desean supervisar 4000 sensores. Diseñar el sistema de E/S necesario. Supongamos que cada sensor contiene un biestable con la información que se desea leer.

8. Un microprocesador controla 4 periféricos conectados de la siguiente forma:

Periférico	Dirección de control	Dirección de datos.
1	0	1
2	2	3
3	4	5
4	6	7

El accionamiento de cada uno de estos periféricos requiere enviar, por el bit 0 de la palabra de control, una señal de la siguiente forma:



Suponiendo que cada instrucción dura 300 ns, para instrucciones de no acceso a memoria, y de  $250 + (n * 200)$  ns, para las que sí acceden:

- A) Sería útil utilizar interrupciones para poder generar la señal de control anterior.
- B) Elaborar un programa que permita accionar dos periféricos con una diferencia de tiempo  $< 12$  microsegundos.