

CAPITULO 3

SUBSISTEMAS DE ENTRADA / SALIDA

3.1 - Características de los subsistemas de E/S: [2][31]

Un subsistema de E/S consiste en interfases de E/S y dispositivos periféricos. Un diagrama típico de esta arquitectura puede verse en la Fig. 3.1.

La interfase de E/S controla la operatoria de los dispositivos conectados a ella. Las operaciones de control (por ejemplo rebobinado, posicionamiento, etc.) se arrancan mediante comandos emitidos por la CPU. El conjunto de comandos que se ejecutan para completar la transacción de E/S se denomina driver.

Las funciones de la interfase son almacenar los datos y realizar las conversiones que se le requieran. También detecta errores en la transmisión y es capaz de reiniciar la transacción en casos de error. Más aún, la interfase puede testear, arrancar y detener el dispositivo según las directivas impartidas por la CPU. En algunos casos la interfase puede consultar a la CPU si algún dispositivo está requiriendo atención urgente.

En la figura 3.2 podemos ver la clase de hardware que se encuentra en el bus de E/S.

Existen distintos tipos de comandos que circulan por el bus, a saber:

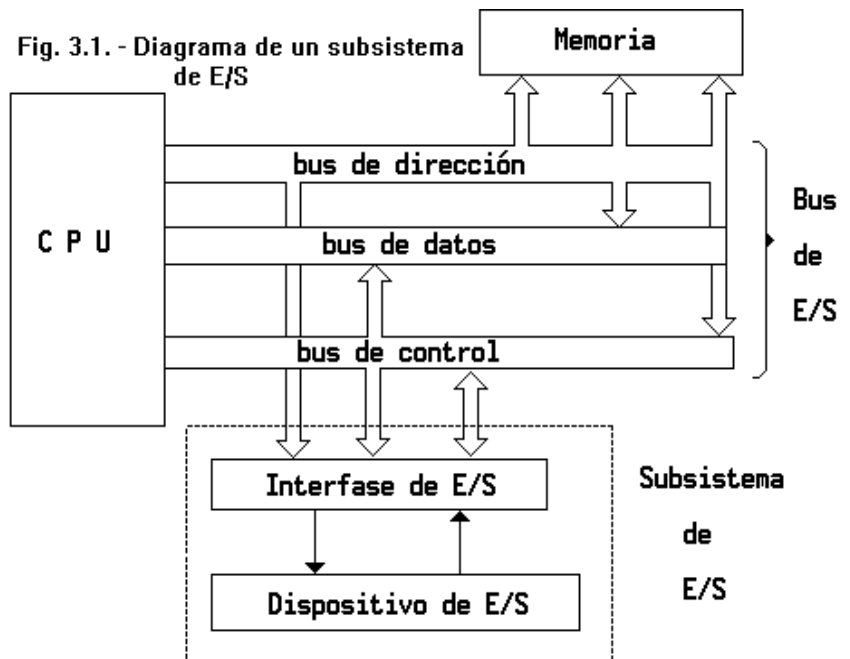
- De control: son para activar el periférico y decirle que debe hacer (por ej. rebobinar una cinta); varían según cada tipo de periférico.
- De verificación: verifican las diversas condiciones de estado en la interfase o en el periférico (por ej., una vez seleccionada la ruta la CPU puede desear verificarla para ver si existe energía (power on) o que el periférico esté en línea (on-line).
- Salida de datos: Hace que la interfase responda tomando un ítem de datos del bus.
- Entrada de datos: la interfase recibe un ítem de datos del periférico y lo coloca en su propio registro separador, avisa a la CPU, la que emite el comando de entrada de datos el cual transfiere el contenido de ese registro al bus de donde es tomado por la CPU y almacenado en su registro acumulador.

Ejemplo: Salida de datos a una unidad de cinta.

El computador arranca la unidad de cinta emitiendo un comando de control. El procesador entonces monitorea el estado de la cinta por medio de comandos de verificación.

Cuando la cinta está en posición correcta, el computador emite un comando de salida de datos.

La interfase responde a la dirección y a las líneas de comando y transfiere los datos de la línea de datos del bus de E/S a su registro separador. La interfase se comunica entonces para aceptar un nuevo ítem de datos para almacenar en la cinta.



3.2 - Modos de transferencia [2]

Los subsistemas de E/S pueden clasificarse según qué tanto esté involucrada la CPU en la transacción de E/S. Una transacción de E/S puede transferir un único bit, byte, palabra, o bloque de bytes de información entre el dispositivo de E/S y la CPU, o entre el dispositivo de E/S y la memoria principal.

3.2.1 - Transferencia de datos bajo control de programa [2]

La arquitectura más simple de E/S es aquella en la cual el procesamiento se realiza en forma secuencial. En tales sistemas, la CPU ejecuta programas que inicializan, chequean el estado del dispositivo, realizan la

transferencia de los datos, y terminan las operaciones de E/S. En este caso, las transacciones de E/S son realizadas utilizando un programa-manejador de E/S. Muchas computadoras proveen esta opción ya que requiere un hardware mínimo.

Sin embargo, la CPU permanece en un ciclo de programa testeando el dispositivo hasta el momento en que este indique su disponibilidad, lo cual trae como desventaja además de que se degrada el uso de la CPU que

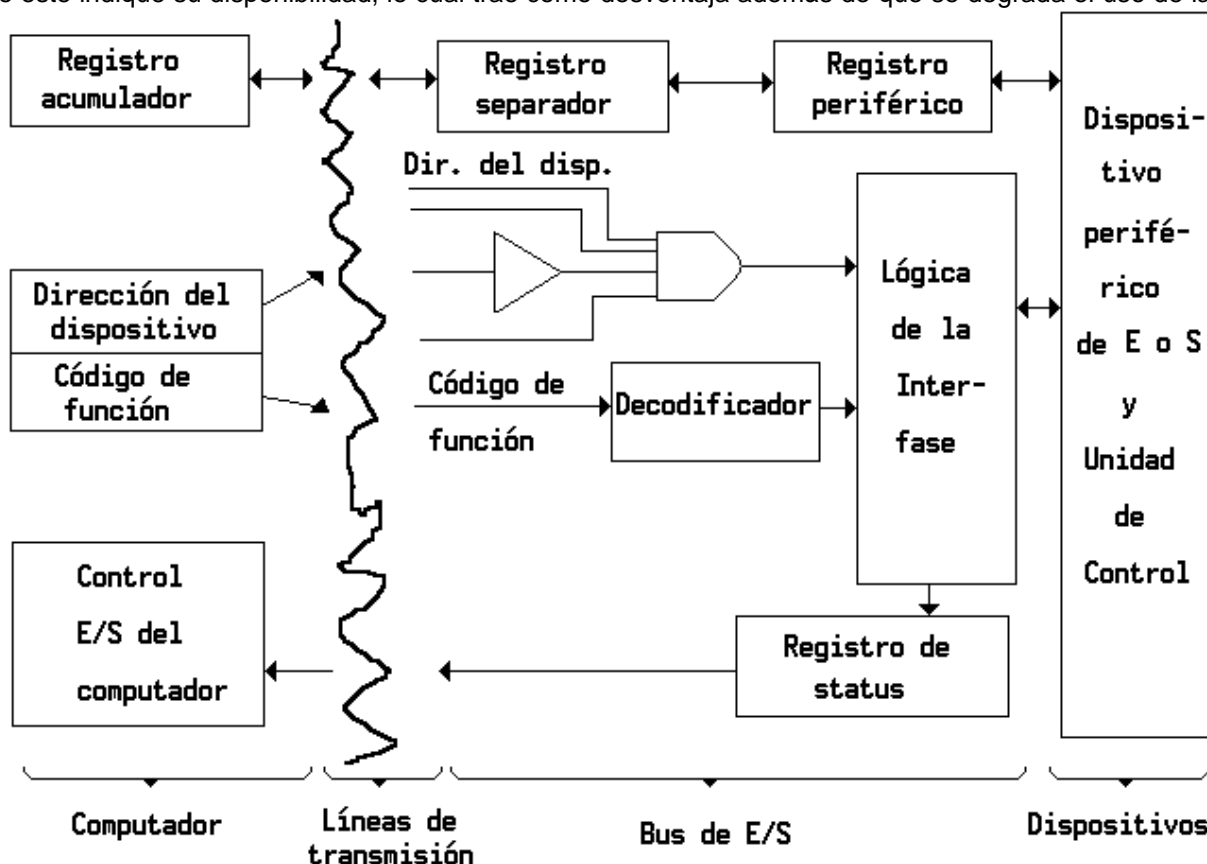


Fig. 3.2. - Hardware dentro de un bus de E/S.

el tiempo requerido para transferir una unidad de información entre memoria y el dispositivo de E/S es de una magnitud varias veces superior al tiempo promedio del ciclo de instrucción. Una posible solución a esta degradación es permitir la concurrencia entre el procesamiento de la CPU y las E/S.

3.2.2 - Transferencia de datos con mecanismo de interrupción [2]

A medida que el nivel de concurrencia aumenta, la complejidad del hardware deberá crecer también para poder satisfacer los requerimientos de transferencia de datos.

Un esquema utiliza un "pseudo" programa-manejador de E/S. En este esquema la CPU inicia la transacción de E/S y reasume su tarea normal. Cuando el dispositivo tiene el dato listo, por ejemplo una operación de input, la unidad de control del dispositivo notifica a la CPU la presencia del dato en su buffer.

La CPU puede entonces atender al dispositivo para ubicar el dato y transferirlo a memoria (nótese que en este caso es la misma CPU la que coloca el dato en la memoria una vez que esta disponible desde el medio externo).

Una descripción similar puede hacerse sobre una operación de output.

La señal de notificación a que nos referimos es un requerimiento de **interrupción**. La capacidad de interrupción libera a la CPU de estar constantemente chequeando el estado del dispositivo.

Debido a que una interrupción puede producirse en forma asincrónica durante un ciclo de instrucción, muchos procesadores permiten que la instrucción se complete antes de atender la interrupción. La CPU indica su disponibilidad para atender interrupciones mediante una señal de habilitación-de-interrupción. Esta señal notifica al dispositivo si la CPU se encuentra en estado no-interruptible. Cuando se produce la interrupción la CPU notifica al dispositivo el hecho de que toma conocimiento de dicha interrupción enviándole una señal de recibida al controlador del dispositivo.

3.2.3 - Transferencia directa a memoria (DMA) [2][31]

El último grado de concurrencia en procesamiento de E/S puede alcanzarse si el controlador del dispositivo es lo suficientemente inteligente como para realizar la transacción de E/S entre el dispositivo y la memoria principal sin la intervención de la CPU.

Este paralelismo es muy efectivo cuando se transmite un bloque de datos. Esto requiere que el controlador sea capaz de generar una secuencia de direcciones de memoria. Sin embargo *la CPU es aún responsable de iniciar la transferencia del bloque*.

Veamos por ejemplo una secuencia típica de operaciones necesarias para transferir un bloque de datos entre un dispositivo y memoria. La CPU inicializa el buffer en la memoria principal que recibirá el bloque de datos luego de que se completa la transacción de E/S. La dirección del buffer y su tamaño son transmitidos al controlador del dispositivo, así como la dirección del bloque de datos dentro del dispositivo. Entonces la CPU ejecuta un comando especial de inicio de E/S (start I/O) que provoca que el subsistema comience a transferir.

Mientras la transferencia se produce, la CPU queda liberada para poder atender a sus cálculos lo que mejora la performance general del sistema. Cuando se completó la transferencia se notifica a la CPU.

Nótese que ya que la CPU y el controlador comparten la memoria principal, el dispositivo debe periódicamente "robar" ciclos de memoria para depositar los datos en ella.

El mecanismo de "robo" de ciclos es muy efectivo debido a que los dispositivos son muy lentos respecto de la CPU. Cuando existe conflicto entre el dispositivo y la CPU se le da prioridad al dispositivo sobre la CPU debido a que es un componente de mayor tiempo crítico.

Este tipo de esquema de transferencia de datos se denomina Acceso Directo a Memoria (**Direct Access Memory - DMA**).

La diferencia principal entre una transferencia controlada por programa de E/S y DMA, es que *el DMA no emplea los registros de la CPU* (por ejemplo para llevar la cuenta de los bytes que se van transfiriendo).

La transferencia es hecha en la interfase de la DMA primero verificando si la unidad de memoria no está siendo utilizada por la CPU y posteriormente la DMA "roba" un ciclo de memoria para acceder una palabra de memoria.

El controlador de E/S utilizado generalmente para operaciones DMA se denomina "canal de datos de E/S". Un canal de datos es un procesador de E/S que puede manejar muchos periféricos a través de un DMA y una facilidad de interrupción.

3.2.4 - IOP's (Input Output Processor) [1][2]

Nótese que la facilidad DMA no concede el control total de la transacción de E/S al subsistema de E/S.

El subsistema de E/S puede asumir el completo control de las transacciones de E/S si se utiliza una unidad especial llamada "procesador de E/S" (IOP).

El IOP tiene acceso directo a memoria principal y contiene una cantidad de canales independientes de E/S de datos. Puede ejecutar programas de E/S y puede realizar algunas transacciones independientes de E/S entre memoria y los dispositivos o entre dos dispositivos, sin intervención de la CPU.

Los canales proveen una vía de comunicación entre el IOP y las unidades de control de los dispositivos y los dispositivos.

El sistema de bus-IOP comunica los procesadores de E/S y los dispositivos de E/S entre sí, en tanto que

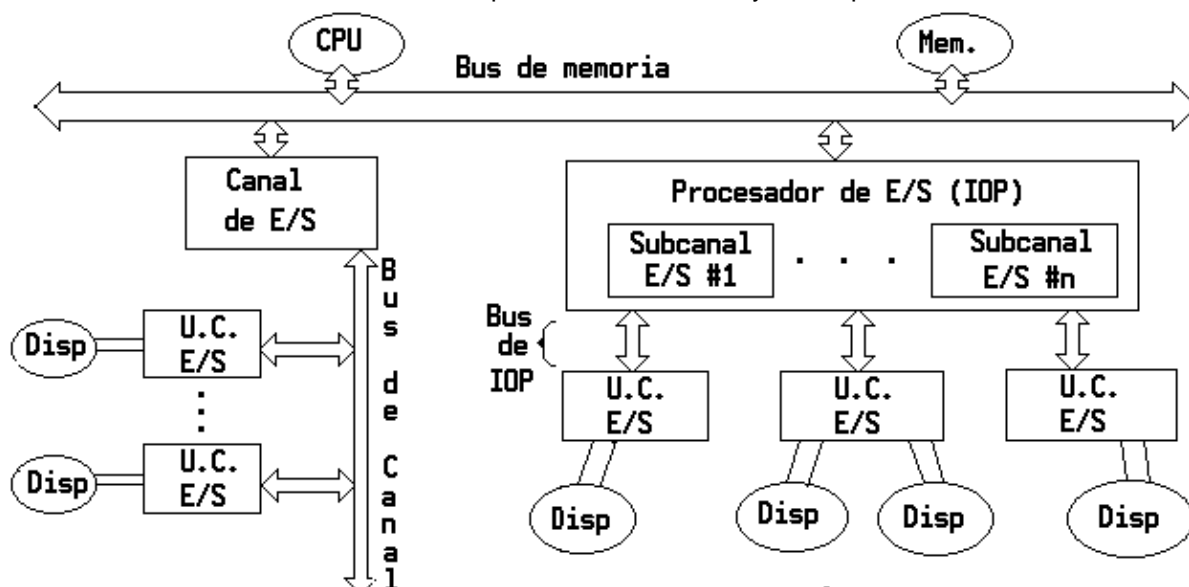


Fig. 3.3. - Arquitectura típica de un IOP.

el sistema de bus de memoria comunica el IOP con la memoria principal. Los canales de E/S pueden existir solos sin un IOP.

En la forma más simple, y cuando existen solos, los canales pueden ser pequeños procesadores que realizan operaciones DMA para un pequeño conjunto de dispositivos. Si el canal se incorpora en un IOP, es esencialmente un componente pasivo sin capacidad de procesamiento lógico sobre lo que le pertenece. Cuando el canal posee capacidad de procesamiento es utilizado muy a menudo como un IOP.

Los canales stand-alone son utilizados en muchos mainframes, como por ejemplo la IBM 370; los IOPs se utilizan en sistemas tales como la CDC 6600 y en microcomputadoras Intel de 8 y 16 bits. En la figura 3.3 puede verse una arquitectura típica de un sistema con IOP y canales stand-alone.

Para realizar una transacción de E/S la CPU transmite una señal de start y la dirección del dispositivo al canal del cual pende dicho dispositivo. Estando la ruta libre el canal obtiene entonces la dirección de comienzo del programa de canal para poder ejecutarlo, y lo ejecuta para realizar la transacción de E/S; de otra forma el requerimiento debe ser encolado o debe notificarse a la CPU respecto de la no disponibilidad del dispositivo. En la figura 3.4 puede verse el mecanismo de comunicación entre un IOP y la CPU.

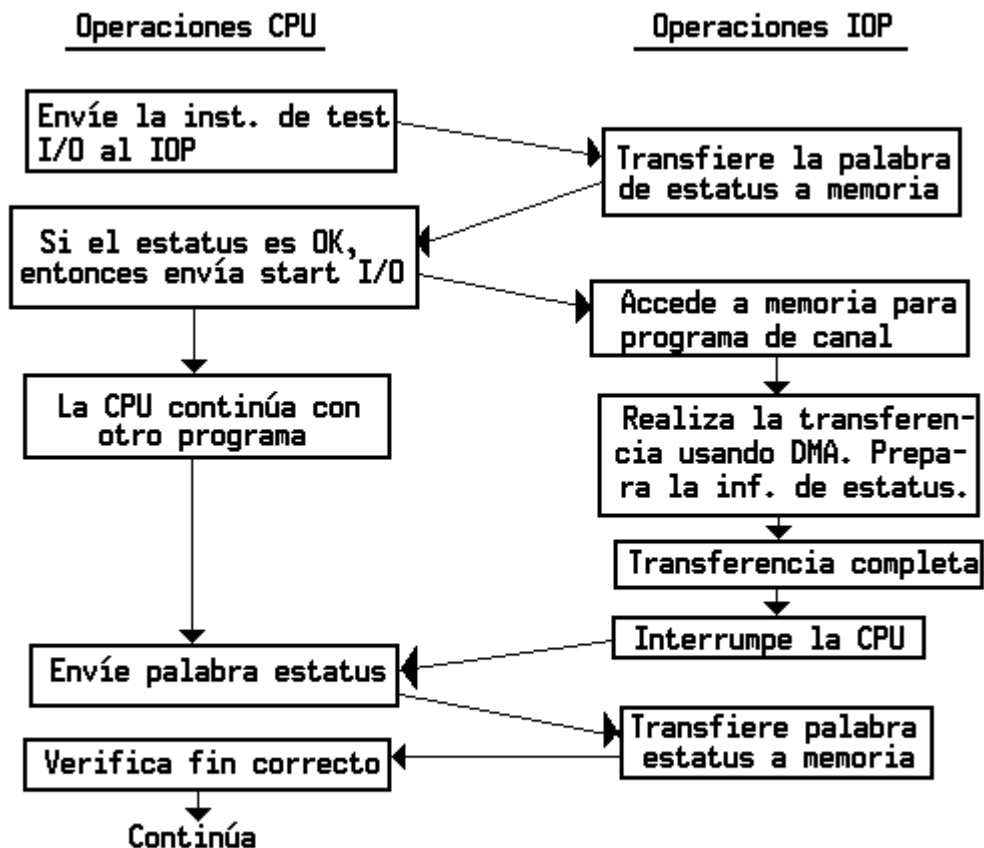


Fig. 3.4. - Mecanismo de comunicación entre CPU - IOP.

3.3 - Canales selectores y multiplexores: su arquitectura[2][31]

Algunos canales de E/S están permanentemente conectados a un único dispositivo (es decir, se necesita un canal por cada dispositivo) mientras que otros pueden estar conectados a la vez con varios dispositivos, a los que pueden accionar uno por vez (canales selectores) o con simultaneidad (canales multiplex).

Un canal selector es un IOP diseñado para manejar una sola transacción de E/S a la vez en forma dedicada. Una vez que el dispositivo es seleccionado, el conjunto de operaciones de E/S para tal transacción se ejecuta hasta finalizar, antes de que pueda iniciarse la próxima transacción. La tasa de transmisión máxima de un canal selector es típicamente del orden de los 1 a 3 megabytes/s.

Un canal multiplexor es un IOP que puede controlar algunas transacciones diferentes de E/S concurrentemente. En este caso, las transferencias de datos son intercaladas (multiplexadas) en el tiempo íntegramente en la interfase de E/S.

Estos tipos de canales pueden dividirse en: block multiplexor o byte multiplexor. El byte multiplexor se utiliza para dispositivos lentos, y el block multiplexor se usa para dispositivos de velocidad media-alta.

El multiplexor consiste en un conjunto de subcanales cada uno de los cuales actúa como un canal selector de baja velocidad. Cada subcanal contiene un buffer, un registro de dirección de dispositivo y flags (banderas, señales) que pueden indicar el estado u operaciones de control.

Sin embargo, los subcanales comparten el control global del canal. En la modalidad multiplex el control del canal recorre cíclicamente las flags de cada subcanal. Si la señal está encendida, el subcanal es seleccionado para transferir un carácter o un bloque. Se chequea la modalidad del subcanal para determinar la dirección de la transferencia. Cuando el carácter o bloque ha sido transferido se examina el siguiente subcanal. El block multiplexor intercala bloques, así como el byte multiplexor intercala caracteres.

Por ejemplo, supongamos que se requieren tres transacciones de E/S para obtener X, Y y Z. Supongamos que cada transacción requiere transmitir un conjunto de n caracteres (cada carácter es 1 byte) X1 a Xn; Y1 a Yn y Z1 a Zn.

Si estas transacciones son iniciadas en un canal selector la transacción aparece de esta forma:

X1,...,Xn,Y1,...,Yn,Z1,...,Zn

En un canal multiplexor de tipo byte multiplexor, con por lo menos tres subcanales, puede aparecer así:

X1,Y1,Z1,...,Xn,Yn,Zn

Si el canal es del tipo block multiplexor de k caracteres por bloque (k menor a n), la transferencia aparece de la siguiente forma:

X1,...,Xk,Y1,...,Yk,Z1,...,Zk,Xk+1,...

3.4 - Configuraciones de E/S en un sistema con memoria cache [2]

Hay dos formas básicas de conectar el subsistema de E/S a una cache.

En el primero el canal de E/S puede ser conectado a la cache. De esta forma la cache es compartida por el procesador y los canales como muestra la figura 3.5.

El canal compite con el procesador por accesos a la cache. Un canal de E/S es mucho más lento que el procesador, luego conectar el canal a la cache no mejora significativamente la performance de la transferencia de E/S.

En este esquema de conexión existe un incremento del tráfico entre la memoria principal y la cache, debido a la actualización de la memoria principal con la información transferida por el canal, éste debe levantar desde memoria principal las instrucciones del correspondiente programa de canal por medio de la cache, además disminuye el espacio disponible de cache a los procesos.

Una configuración alternativa es conectar el canal directamente a memoria, como se ve en la figura 3.6.

En este caso el canal compite con la cache para acceder a memoria principal. Aún subsiste el conflicto entre la CPU y el canal para el acceso a memoria principal, pero solamente cuando la CPU detecta la falta de la información en la cache.

Existe aún un problema de coherencia entre la información en la cache y la información accedida en memoria principal por el canal.

Una forma posible de solucionar este problema es obligar al canal a verificar si la palabra de memoria que está accediendo se encuentra en memoria cache y testear el bit de cambio de esa palabra en la cache. Luego podrá determinar si es necesario una actualización del dato en memoria principal desde la memoria cache antes de recuperar la dirección deseada.

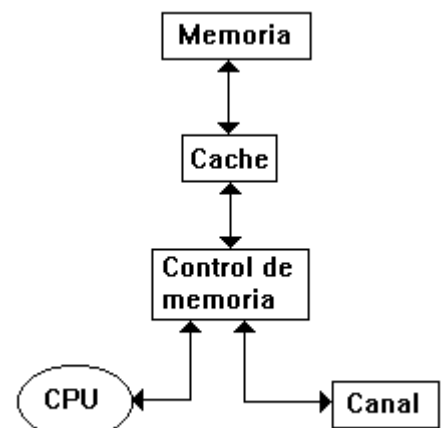


Fig. 3.5. - Canal accede a cache.

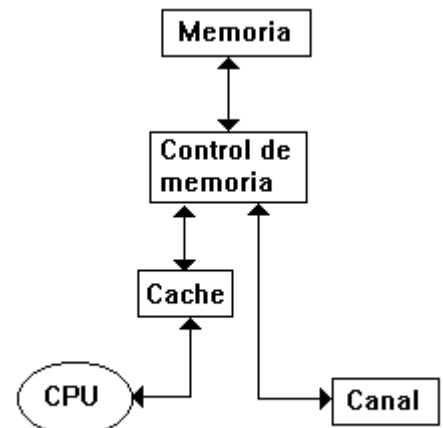


Fig. 3.6. - Canal accede a memoria.

EJERCICIOS

1) Dibuje esquemáticamente como se compone un subsistema de E/S. Indique claramente qué clase de información circula por cada uno de los buses de interconexión.

2) Indique las diferencias en los siguientes métodos de transferencia de información dentro de un subsistema de E/S :

- Transferencia de datos bajo control de programa
- Transferencia de datos con mecanismo de interrupción
- Transferencia directa a memoria
- Transferencia por Procesadores de E/S

Indique claramente en cuál de estos métodos aparece por primera vez el elemento hardware **canal**.

- 3) Tiene sentido tener IOP's en un sistema monoprocesador ? Justifique.
- 4) Cómo funciona un canal byte-multiplexor ? Y un canal block-multiplexor ?
- 5) Cuáles son las dos formas de conectar una memoria cache en un subsistema de E/S dado ?

3.1 - Características de los subsistemas de E/S: [2][31]	1
3.2 - Modos de transferencia [2]	1
3.2.1 - Transferencia de datos bajo control de programa [2]	1
3.2.2 - Transferencia de datos con mecanismo de interrupción [2]	2
3.2.3 - Transferencia directa a memoria (DMA) [2][31]	3
3.2.4 - IOP's (Input Output Processor) [1][2]	3
3.3 - Canales selectores y multiplexores: su arquitectura[2][31]	4
3.4 - Configuraciones de E/S en un sistema con memoria cache [2]	5