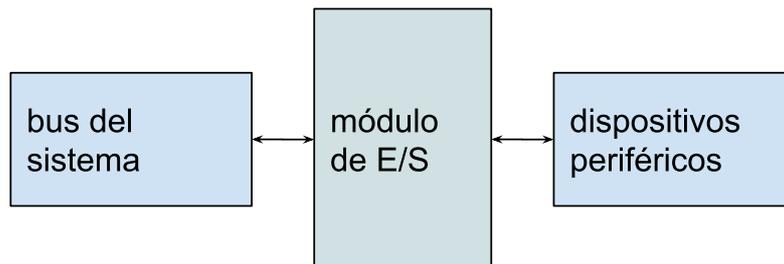


Capítulo 7 - Entrada/Salida

Cada módulo de E/S interactúa con el bus del sistema o el interruptor central, y controla uno o más dispositivos periféricos. Un módulo de E/S contiene lógica para realizar una función de comunicación entre los periféricos y el bus.



¿Por qué no se conectan directamente los periféricos al bus?

→ Hay muchos periféricos con varios métodos de operación, por lo que sería poco práctico incorporar la lógica necesaria dentro del procesador para controlar una gama de dispositivos.

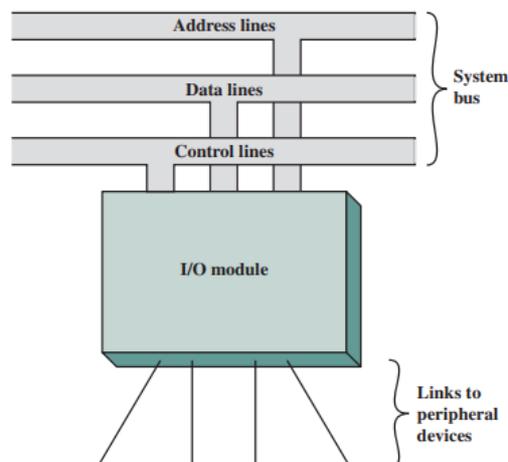
→ La tasa de transferencia de datos de los periféricos a menudo es mucho más lenta que la de la memoria o el procesador, por lo tanto, tampoco es práctico usar el bus de sistema de alta velocidad para comunicarse directamente con un periférico.

→ Por otro lado, la tasa de transferencia de datos de algunos periféricos es más rápida que la de la memoria o el procesador, la diferencia llevaría a ineficiencias si no se gestiona adecuadamente.

→ Los periféricos a menudo usan diferentes formatos de datos y longitudes de palabra que la computadora a la que están conectados, por lo tanto, se requiere un módulo de E/S, este módulo tiene 2 funciones principales:

⇒ Interfaz con el procesador y la memoria a través del bus del sistema o el interruptor central.

⇒ Interfaz con uno o más dispositivos periféricos mediante enlaces de datos adaptados.



7.1 Dispositivos externos

Las operaciones de E/S se realizan a través de una amplia variedad de dispositivos externos que proporcionan un medio de intercambio de datos entre el entorno externo y la computadora.

Un **dispositivo externo** se conecta a la computadora a través de un módulo de E/S. La conexión entre la computadora y el módulo de E/S, se utiliza para intercambiar control, estados y datos.

Clasificación de dispositivos externos

- De interacción con humanos: posibilitan la comunicación con el usuario de la computadora. (ej. impresora)
- De interacción con máquinas: posibilitan la comunicación con el equipo, es decir, con otros componentes de la computadora. (ej. disco rígido, lectoras de CD's y DVD's)
- De comunicación: posibilitan la comunicación con dispositivos remotos. (ej. placas de red)

Estos últimos (dispositivos de comunicación) permiten a la computadora el intercambio de datos con dispositivos remotos, como un dispositivo legible por humanos, como un terminal, un dispositivo legible por máquinas, o incluso otra computadora.

La interfaz hacia el módulo de E/S es en forma de *señales de control, estados y datos*. Las **señales de control** determinan la función que el dispositivo realizará, como enviar datos al módulo de E/S, aceptar datos del módulo de E/S, informar sobre el estado o realizar alguna función de control particular para el dispositivo. Los **datos** están en forma de un **conjunto de bits para enviar o recibir** del módulo de E/S. Las **señales de estado** indican el estado del dispositivo (ready/not ready para mostrar si el dispositivo está listo para la transferencia de datos).

La **lógica de control** asociada con el dispositivo controla la operación del dispositivo en respuesta a la dirección del módulo de E/S. El **transductor** convierte los datos de forma **eléctrica a otras formas de energía** durante la **salida** y de **otras formas a eléctrica** durante la **entrada**. Típicamente, el buffer está asociado con el transductor para mantener temporalmente los datos que se transfieren entre los módulos de E/S y el entorno externo.

7.2 Módulos de E/S

Función de un módulo

Las principales funciones o requerimientos de un módulo de E/S caen en las siguientes categorías:

- ⇒ Control y temporizado.
- ⇒ Comunicación con el procesador.
- ⇒ Comunicación con los dispositivos.
- ⇒ Almacenamiento temporal de datos.
- ⇒ Detección de errores.

→ Durante cualquier período de tiempo, el procesador debe comunicarse con uno o más dispositivos externos en patrones impredecibles, dependiendo de la necesidad de E/S del programa. Los recursos internos, como la memoria principal y el bus del sistema, deben ser compartidos entre varias actividades, incluida la E/S de datos. Por lo tanto, la **función de E/S** incluye un requisito de **control y temporización**, para **coordinar el flujo de tráfico** entre los recursos internos y los dispositivos externos.

El **control de transferencia** de datos desde un dispositivo externo hacia el procesador puede involucrar los siguientes pasos:

- 1) El procesador le pregunta al módulo de E/S para chequear el estado del dispositivo conectado.
- 2) El módulo de E/S retorna el estado del dispositivo.
- 3) Si el dispositivo está operativo y listo para transmitir, el procesador pide la transferencia de datos, mediante un comando al módulo de E/S.
- 4) El módulo de E/S obtiene la unidad de datos del dispositivo externo.
- 5) Los datos se transfieren desde el módulo de E/S hacia el procesador.

→ Si el sistema emplea un bus, entonces cada una de las interacciones entre el procesador y el módulo de E/S implica:

- Decodificación de comandos: El módulo de E/S acepta los comandos desde el procesador, típicamente enviados como señal en el bus de control
- Datos: Los datos son intercambiados entre el procesador y el módulo de E/S a través del bus de datos.
- Reportes de estado: Debido a que los periféricos son muy lentos, es importante saber el estado del módulo de E/S.
- Reconocimiento de direcciones: Así como cada palabra de memoria tiene una dirección, también cada dispositivo de E/S. Además, un módulo de E/S debe reconocer dirección única para cada dispositivo externo que controla.

Por otro lado, el módulo de E/S debe ser capaz de llevar a cabo la **comunicación entre los dispositivos**. Esta comunicación implica comandos, información de estados y datos.

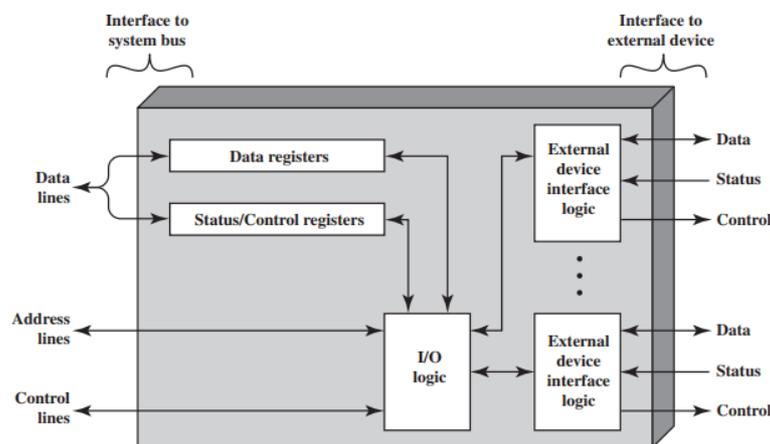
Una tarea esencial de un módulo de E/S es el **almacenamiento temporal de datos**. Mientras la tasa de transferencia dentro y fuera de la memoria principal o el procesador es bastante alta, la tasa de transferencia es muchísimo menor para muchos de los dispositivos externos y cubre un amplio rango. Los datos provenientes de la memoria principal son enviados a un módulo de E/S en ráfaga, son almacenados temporalmente en el módulo de E/S y luego son enviados al dispositivo externo a su tasa de datos. Por el contrario, los datos se almacenan en el búfer para no ocupar la memoria con una operación de transferencia lenta. Así, el módulo de E/S debe ser capaz de operar a la velocidad del dispositivo y de la memoria. Del mismo modo, si el dispositivo de E/S opera con una tasa de transferencia más alta que la tasa de acceso a memoria, entonces el módulo de E/S realiza la operación de almacenamiento en el búfer necesaria.

Finalmente, un módulo de E/S a menudo es responsable de la **detección de errores** y para informar posteriormente errores al procesador. Una clase de errores incluye fallos mecánicos y eléctricos informados por el dispositivo. Otra clase consiste en cambios no intencionales en el patrón de bits mientras se transmite desde el dispositivo al módulo de E/S. A menudo

se utiliza algún tipo de código de detección de errores para detectar errores de transmisión. Un ejemplo simple es el uso de un bit de paridad en cada carácter de datos.

Estructura de un módulo de E/S

El módulo se conecta con el resto de la computadora a través de un conjunto de líneas de señales. Los datos que se transfieren desde y hacia el módulo de E/S son almacenados temporalmente en uno o más registros de datos. También puede haber uno o más registros de estado que brindan la información del estado actual. Un registro de estados también puede funcionar como un registro de control para aceptar información de control detallada desde el procesador. La lógica dentro del módulo interactúa con el procesador a través de un conjunto de líneas de control. El **procesador** usa las **líneas de control** para **emitir comandos** hacia el **módulo de E/S**. El módulo de E/S también debe ser capaz de reconocer y generar direcciones asociadas con los dispositivos de control. Cada módulo de E/S tiene una única dirección o, si controla más de un dispositivo externo, un único conjunto de direcciones. Finalmente, el módulo de E/S contiene lógica específica para la interfaz con cada dispositivo que controla.



7.3 E/S programada

Con la E/S programada, los datos son intercambiados entre el procesador y el módulo de E/S. El procesador ejecuta un programa que le da un control directo de la operación de E/S, incluyendo la detección del estado del dispositivo, enviando un comando de lectura o escritura y transfiriendo los datos. Cuando el procesador emite un comando al módulo de E/S, debe esperar hasta que la operación de E/S se complete. Si el procesador es más rápido que el módulo de E/S, es una pérdida de tiempo del procesador.

En la E/S programada, así como en la que es basada en interrupciones, el procesador es responsable de extraer los datos desde la memoria principal para la salida y guardar datos en la memoria principal para la entrada.

→ Cuando el procesador se encuentra ejecutando un programa y encuentra una instrucción relacionada a E/S, ejecuta la instrucción emitiendo un comando al correspondiente módulo de E/S. Con la E/S programada, el módulo de E/S llevará a cabo la acción requerida y luego

colocará los bits apropiados en el registro de estado de E/S. El **módulo de E/S** no hace más nada para avisar al procesador, en particular, **no interrumpe al procesador**. Así, es responsabilidad del procesador chequear periódicamente el estado del módulo de E/S hasta que encuentre que la operación se completó.

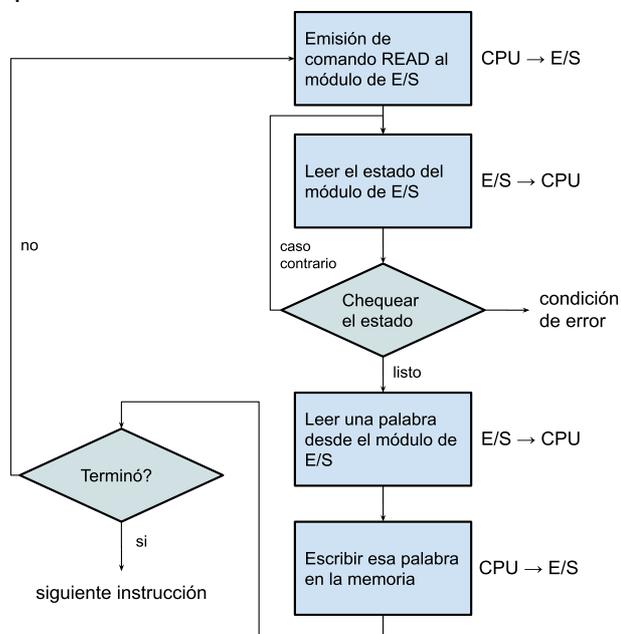
Comandos de E/S

Para ejecutar una instrucción relacionada con E/S, el procesador emite una dirección, especificando el módulo de E/S y el dispositivo externo específicos, y el comando de E/S. Existen **4 tipos de comandos de E/S** que le módulo de E/S puede recibir cuando es direccionado por el procesador:

- **Control:** Se utiliza para activar un dispositivo externo y decirle qué hacer. Estos comandos están adaptados al tipo de dispositivo periférico en particular.
- **Testeo (test):** Es utilizado para testear varias condiciones de estado asociadas con el módulo de E/S y sus periféricos. El procesador querrá saber si el periférico de interés está encendido y disponible para su uso, además de saber si la última operación de E/S se ha completado y si se han producido errores.
- **Lectura (read):** Hace que el módulo de E/S obtenga un ítem de datos desde un dispositivo externo y que lo almacene temporalmente en el buffer. El procesador puede obtener los ítems de datos, solicitando al módulo de E/S que los coloque en el bus de datos.
- **Escritura (write):** Hace que el módulo de E/S tome un ítem de dato desde el bus de datos y luego los transmita al dispositivo externo.

Instrucciones de E/S

Con la E/S programada, hay una estrecha correspondencia entre las instrucciones relacionadas con la E/S, que el procesador recupera desde la memoria, y los comandos de E/S que el procesador emite al módulo de E/S para ejecutar las instrucciones. Es decir, las instrucciones se pueden mapear fácilmente en comando de E/S, y a menudo existe una simple relación uno a uno. La forma de la instrucción depende de la manera en que se direccionan los dispositivos externos.



Típicamente, va a haber muchos dispositivos de E/S conectados a través de los módulos de E/S al sistema. A cada dispositivo se le asigna un identificador único o una dirección. Cuando el procesador emite un comando de E/S, el comando contiene la dirección del dispositivo deseado. Así, cada módulo de E/S debe interpretar las líneas de dirección para determinar si el comando es para él.

Cuando el procesador, la memoria principal, y el módulo de E/S comparten un bus en común, dos modos de direccionamiento son posibles: **memoria mapeada y aislada**. Con **E/S mapeada a memoria**, hay un solo espacio para las locaciones de memoria y los dispositivos de E/S. El procesador trata los registros de estados y los registros de datos de los módulos E/S como locaciones de memoria y usa las mismas instrucciones máquina para accederlos a ambos.

Con la E/S mapeada a memoria, una sola línea para *lectura* y una sola línea para *escritura*. Alternativamente, en la **E/S aislada**, el bus puede estar equipado con lecturas y escrituras en memoria además de líneas de comando de E/S. La línea de comando especifica si la dirección se refiere a una ubicación de memoria o a un dispositivo de E/S.

En un sistema con E/S aislada, la memoria y los dispositivos de E/S no compiten por las mismas direcciones y, por lo tanto, pueden ser accedidos de manera independiente y simultánea. Esto aumenta la eficiencia y la velocidad de la computadora, ya que permite que el sistema operativo realice operaciones de E/S y de memoria de manera simultánea y sin conflictos.

Con E/S aislada, los puertos de E/S son accesibles solo con comandos especiales, los cuales activan las líneas de comandos de E/S en el bus.

7.4 E/S con interrupciones

El problema con la E/S programada es que el procesador tiene que esperar mucho tiempo para que el módulo de E/S esté listo para recibir o transmitir datos. El procesador, mientras espera, debe preguntar, repetidas veces, el estado del módulo de E/S, lo cual reduce altamente el rendimiento de todo el sistema.

Una alternativa es que el procesador emita un comando de E/S a un módulo y que luego siga realizando otras tareas. El módulo de E/S va a interrumpir al procesador para solicitar el servicio cuando esté listo para intercambiar datos con el procesador. Luego, el procesador realiza la transferencia de datos, como antes, y luego reanuda su procesamiento anterior.

⇒ Desde el punto de vista del **módulo de E/S**, para la **entrada** recibe un comando de lectura (READ) desde el procesador. Luego, el módulo de E/S procede a leer los datos desde el dispositivo externo. Una vez que los datos están en el registro de datos del módulo, el módulo señala una interrupción al procesador desde la línea de control. Luego, el módulo de E/S espera hasta que los datos son solicitados por el procesador, cuando se realiza la solicitud, el módulo coloca los datos en el bus de datos y luego queda disponible para otra operación de E/S.

⇒ Desde el punto de vista del **procesador**, para la entrada, el procesador emite un comando de **lectura**, luego sigue realizando otras tareas. Al final de cada ciclo de instrucción, el procesador se fija si hubo alguna interrupción. Cuando el módulo de E/S realiza la interrupción, el procesador guarda el contexto del programa actual y procesa la

interrupción. En este caso, el procesador lee la palabra de datos desde el módulo de E/S y la guarda en memoria para luego retomar el contexto en el que estaba trabajando.

→ La E/S con interrupciones es más eficiente que la E/S programa porque quita la necesidad de esperar. Sin embargo, la E/S con interrupciones sigue consumiendo mucho tiempo del procesador porque cada palabra de datos que va desde la memoria hasta el módulo de E/S, o viceversa, debe pasar por el procesador.

Problemas de diseño

¿Cómo determina el procesador qué módulo realizó la interrupción?

→ Para identificar qué dispositivo interrumpió, hay 4 alternativas:

1) Múltiples líneas de interrupción

La alternativa más sencilla es tener múltiples líneas de interrupción entre el procesador y los módulos de E/S. Sin embargo, resulta poco práctico dedicar más de unas líneas de bus o pines del procesador como líneas de interrupción. En consecuencia, incluso si se usan varias líneas, es probable que cada línea tenga varios módulos de E/S conectadas, entonces debería utilizarse alguna de las otras 3 alternativas en cada línea.

2) Consulta de software (software polling)

Otra alternativa es la consulta de software. Cuando el procesador detecta una interrupción, va hacia una rutina de servicio de interrupción que consulta a cada módulo de E/S quién de ellos realizó la interrupción. La consulta puede ser en forma de líneas de comando por separado, en este caso, el procesador envía un comando y coloca la dirección de un módulo particular en las líneas de dirección, el módulo responde positivamente si realizó la interrupción. Alternativamente, cada módulo puede contener un registro direccionable de estados, en este caso, el procesador lee el estado de cada módulo de E/S para identificar al que interrumpió.

→ La desventaja de la consulta de software es el tiempo que consume.

3) Conexión en cadena (Daisy chain)

Una alternativa más eficiente es la conexión en cadena, la cual proporciona una consulta a nivel de hardware. Cuando el procesador detecta una interrupción. Para las interrupciones todos los módulos de E/S comparten una línea en común donde solicitan la interrupción. La línea de reconocimiento de la interrupción está conectada a través de los módulos, cuando el procesador detecta la interrupción envía una señal que se propaga a través de los módulos de E/S hasta que consigue el módulo que realizó la solicitud de interrupción. El módulo que realizó la solicitud responde colocando su ubicación en las líneas de datos, la cual el procesador utiliza como puntero y evita que se necesite ejecutar primero una rutina general de servicio de interrupción.

4) Arbitraje de bus

Con el arbitraje de bus, un módulo de E/S primero debe obtener el control del bus antes de poder realizar la solicitud de la interrupción. Así, sólo un módulo puede levantar la línea de solicitud a la vez. Cuando el procesador detecta la interrupción, responde en la línea de reconocimiento de interrupción y luego el módulo coloca su ubicación en la línea de datos.

7.5 E/S con DMA

La E/S programada y la E/S con interrupciones tienen inconvenientes, la tasa de transferencia es limitada por la velocidad en la que trabaja el procesador y, además, el procesador está ocupado manejando la transferencia de E/S.

Hay cierto equilibrio entre estos dos inconvenientes. Considerando la transferencia de un bloque de datos, al utilizar una E/S programada simple, el procesador está dedicado a la tarea de E/S y puede mover datos a una tasa bastante alta, a costa de no hacer nada más. La E/S con interrupciones libera en cierta medida al procesador a expensas de la tasa de transferencia de E/S.

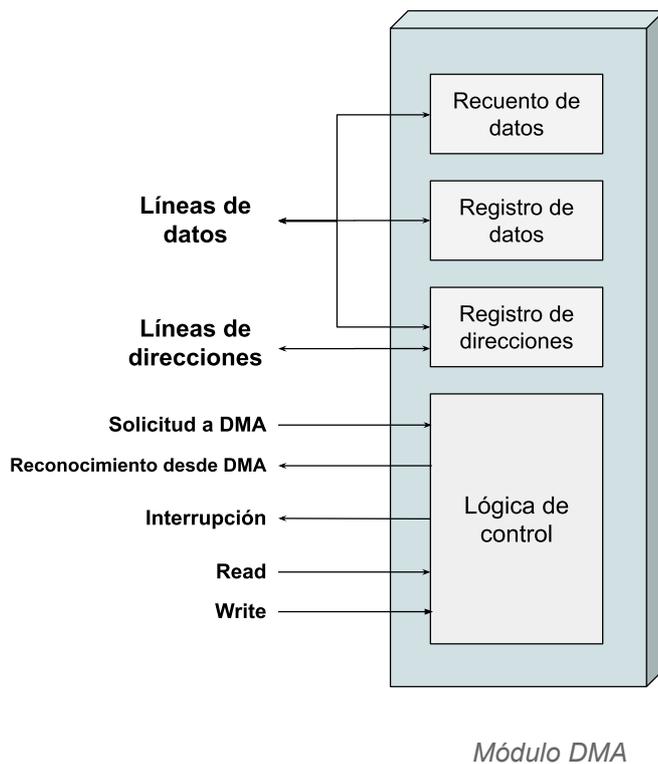
Cuando se deben mover grandes volúmenes de datos se requiere una técnica más eficiente, por caso, el **acceso directo a la memoria (DMA)**.

Función del DMA

DMA involucra un módulo adicional en el bus del sistema. El módulo de DMA es capaz de imitar al procesador y, en efecto, tomar control del sistema desde el procesador. Necesita hacer esto para poder transferir los datos desde y hacia memoria a través del bus del sistema. Para este propósito, el módulo de DMA debe usar el bus solo cuando el procesador no lo necesite, o debe forzar al procesador para que deje de usarlo temporalmente. Esta última, es más común y se conoce como **robo de ciclos**, porque el módulo DMA en efecto, roba un ciclo de bus

→ Cuando el procesador desea leer o escribir un bloque de datos, emite un comando al módulo DMA, enviando al módulo DMA la siguiente información

- Si se pide una *lectura* o *escritura*, usando la línea de control para lectura o escritura entre el procesador y el módulo de E/S.
- La dirección del dispositivo de E/S involucrado, comunicado en las líneas de datos.
- La locación inicial en memoria para leer desde o escribir hacia, comunicada con las líneas de datos y almacenadas por el módulo DMA en su registro de direcciones
- El número de palabras a ser leídas o escritas, nuevamente comunicadas por las líneas de datos y guardadas en el registro de conteo de datos



El procesador luego, continúa con otro trabajo. Ha delegado esta operación de E/S al módulo DMA. El módulo DMA transfiere un bloque entero de datos, palabra por palabra, directamente desde o hacia memoria, sin ir a través del procesador. Cuando la transferencia se completa, el módulo DMA envía una señal de interrupción al procesador. Así, el procesador está involucrado sólo en el inicio y en el fin de la transferencia.

En cada caso, el procesador está suspendido solo antes que necesite usar el bus. El módulo DMA, luego, transfiere una palabra y le devuelve el control al procesador. Notar que esto no es una interrupción, el procesador no guarda el contexto y luego hace otra cosa. Más bien, el procesador se detiene durante un ciclo de bus. La idea es hacer que el procesador se ejecute más lentamente. Sin embargo, para transferencia de E/S de múltiples palabras, DMA sigue siendo mucho más eficiente que la E/S con interrupciones o programada.

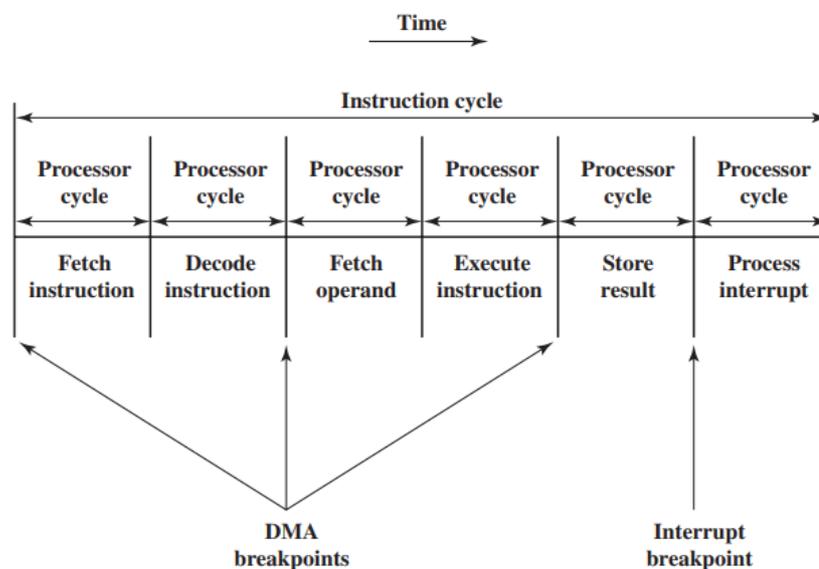
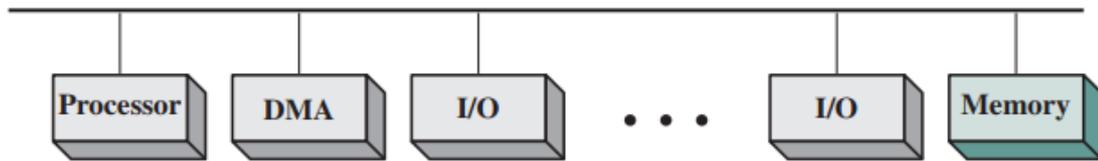
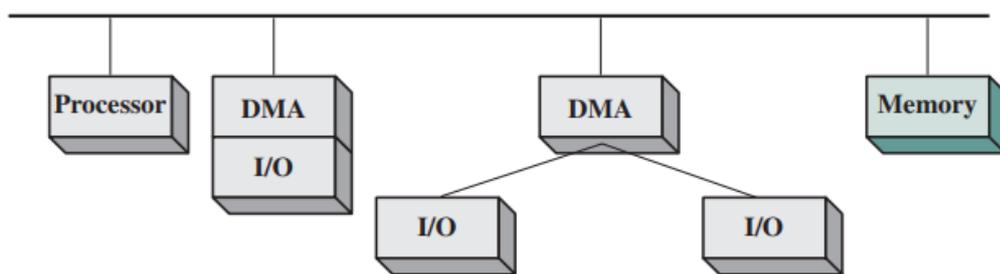


Figure 7.13 DMA and Interrupt Breakpoints during an Instruction Cycle



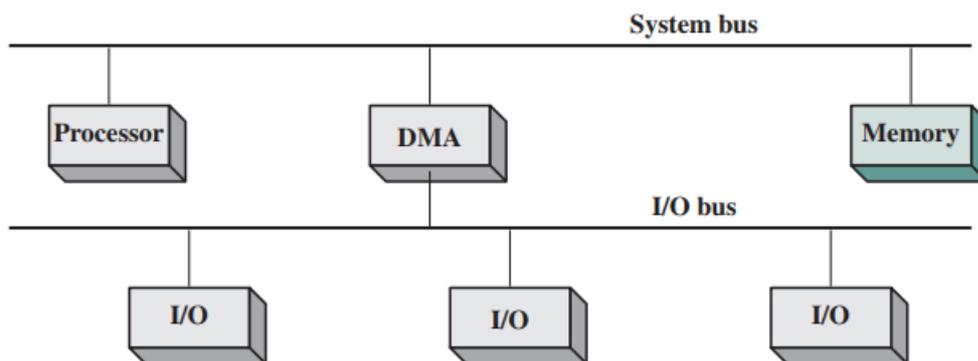
(a) Single-bus, detached DMA

El mecanismo DMA puede configurarse de varias formas, en este ejemplo (a) todos los módulos comparten el mismo bus del sistema. El módulo DMA, actuando como procesador sustituto, usa la E/S programada para intercambiar datos entre la memoria y el módulo de E/S a través del módulo DMA. Esta configuración es poco costosa y claramente ineficiente. Como E/S programada controlada por el procesador, cada transferencia consume 2 ciclos de bus.



(b) Single-bus, integrated DMA-I/O

El número de ciclos de bus requeridos se puede reducir sustancialmente integrando las funciones de DMA y de E/S. Esto quiere decir que hay un camino entre el módulo de E/S y uno o más módulos de E/S que no incluye al bus del sistema. La lógica de DMA puede ser parte de un módulo de E/S o debe ser un módulo por separado que controla uno o más módulos de E/S.



(c) I/O bus

Este concepto puede ser llevado a un siguiente nivel, conectando los módulos de E/S al módulo DMA usando un bus de E/S. Esto reduce el número de interfaces de E/S en el módulo DMA a uno y proporciona una configuración que puede expandirse fácilmente.

En estos dos casos (b y c), el bus del sistema que el módulo de E/S comparte con el procesador y la memoria, es únicamente usado por el módulo DMA para intercambiar datos con la memoria. El intercambio de datos entre el módulo DMA y los módulos de E/S toma lugar fuera del bus del sistema.

7.6 Procesadores y canales de E/S

La evolución de la función de E/S

A medida que los sistemas han evolucionado, hubo un patrón de creciente complejidad y sofisticación de los componentes individual, en ninguna parte es más evidente que en la función de E/S. Los pasos de evolución pueden resumirse de la siguiente forma:

- 1) El CPU controla directamente un dispositivo externo. Esto se ve en los dispositivos simples controlados por el microprocesador.
- 2) Se añade un controlador o módulo de E/S. El CPU usa la E/S programada sin interrupciones. Con este paso, el CPU se separa de los detalles específicos de las interfaces de los dispositivos externos.
- 3) La misma configuración se utiliza pero se implementan las interrupciones. El CPU no necesita perder tiempo esperando para que una operación de E/S se realice, lo cual incrementa la eficiencia.
- 4) El módulo de E/S tiene acceso directo a memoria desde el DMA. Puede mover un bloque de datos desde/hacia memoria sin involucrar al CPU, exceptuando el principio y el final de la transferencia.
- 5) El módulo de E/S mejora para convertirse en un procesador por sí mismo, con un set de instrucciones especial adaptado a la E/S. El CPU dirige al procesador de E/S para ejecutar un programa de E/S en memoria. El procesador de E/S recupera y ejecuta estas instrucciones sin intervención del CPU, esto permite que el CPU especifique una secuencia de actividades de E/S y ser interrumpido sólo cuando la secuencia entera se realice.
- 6) El módulo de E/S tiene una memoria local propia y es una computadora en sí. Con esta arquitectura un set largo de dispositivos de E/S pueden ser controlados con una participación mínima del CPU.

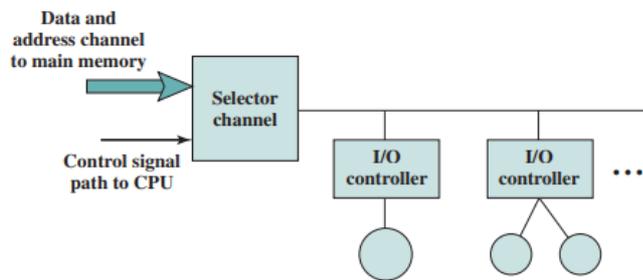
El CPU es cada vez más liberado de las tareas relacionadas con E/S, mejorando el desempeño. Con los dos últimos pasos (5,6) un cambio mayor ocurre con la **introducción de un módulo de E/S capaz de ejecutar un programa**. Para el paso 5), el módulo de E/S es usualmente conocido como un canal de E/S. Para el paso 6), el término procesador de E/S es más usado. Sin embargo, ambos términos se aplican para ambos casos.

Características de los canales de E/S

Los **canales de E/S representan una extensión del concepto de DMA**. Un canal de E/S tiene la capacidad de ejecutar instrucciones de E/S, lo que le da control total sobre las operaciones de E/S. En un sistema con tales dispositivos, el CPU no ejecuta instrucciones de E/S. Esas instrucciones son almacenadas en la memoria principal para ser ejecutadas por un procesador de propósito especial en el propio canal de E/S. Por lo tanto, el CPU inicia una transferencia de E/S instruyendo al canal de E/S para que ejecute un programa en memoria. El programa especifica el/los dispositivo/s, el/las área/s de memoria para el almacenamiento, prioridad y acciones que deben tomarse para ciertas condiciones de error. El canal de E/S sigue estas instrucciones y controla la transferencia de datos.

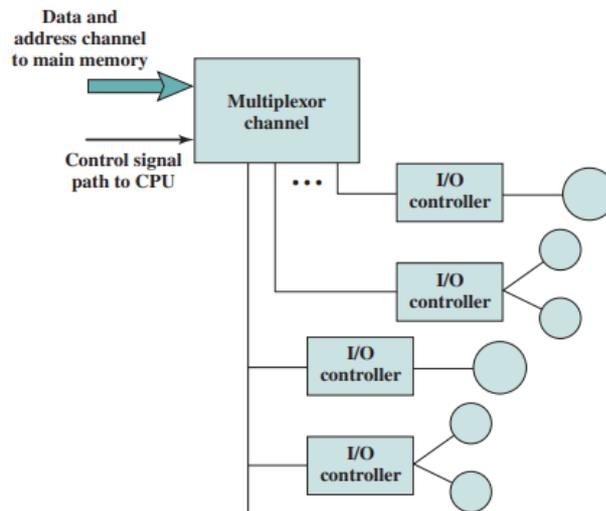
Dos tipos de canales de E/S son comunes:

- Un *canal selector* controla múltiples dispositivos de alta velocidad y, en cualquier momento, está dedicado a la transferencia de datos con uno de esos dispositivos. Por lo tanto, el canal de E/S selecciona un dispositivo y lleva a cabo la transferencia de datos. Cada dispositivo, o un pequeño conjunto de dispositivos, es manejado por un *controlador*, o módulo de E/S, lo cual es muy parecido a los módulos de E/S de los que hablamos anteriormente. Así, el canal de E/S trabaja en lugar del CPU para controlar estos controladores de E/S.



(a) Selector

- Un *canal multiplexor* puede manejar la E/S con varios dispositivos al mismo tiempo. Para los dispositivos lentos, un *byte multiplexor* acepta o transmite caracteres lo más rápido posible a múltiples dispositivos. Para dispositivos rápidos, un bloque multiplexor intercala los bloques de datos desde varios dispositivos.



(b) Multiplexor

Figure 7.15 I/O Channel Architecture