

Almacenamiento y resto de E/S

Diseño de un Sistema de E/S básico

Katia Leal Algara, URJC

Juan González Gómez, URJC

Introducción

- El sistema de E/S permite la comunicación con el mundo exterior
- La transferencia de información entre la CPU y el mundo exterior se realiza a través de dispositivos llamados **periféricos**
- **CPU**
 - Funcionamiento **síncrono**
 - Información **digital**
- **Mundo exterior**
 - Funcionamiento **asíncrono**
 - Información **analógica**

Introducción

- Gran olvidado en el campo de la Arquitectura de Computadores
- Sin embargo, por mucho que se optimice el funcionamiento del procesador y de la jerarquía de memoria, el **sistema de E/S** puede suponer un gran cuello de botella afectando negativamente al rendimiento global del sistema

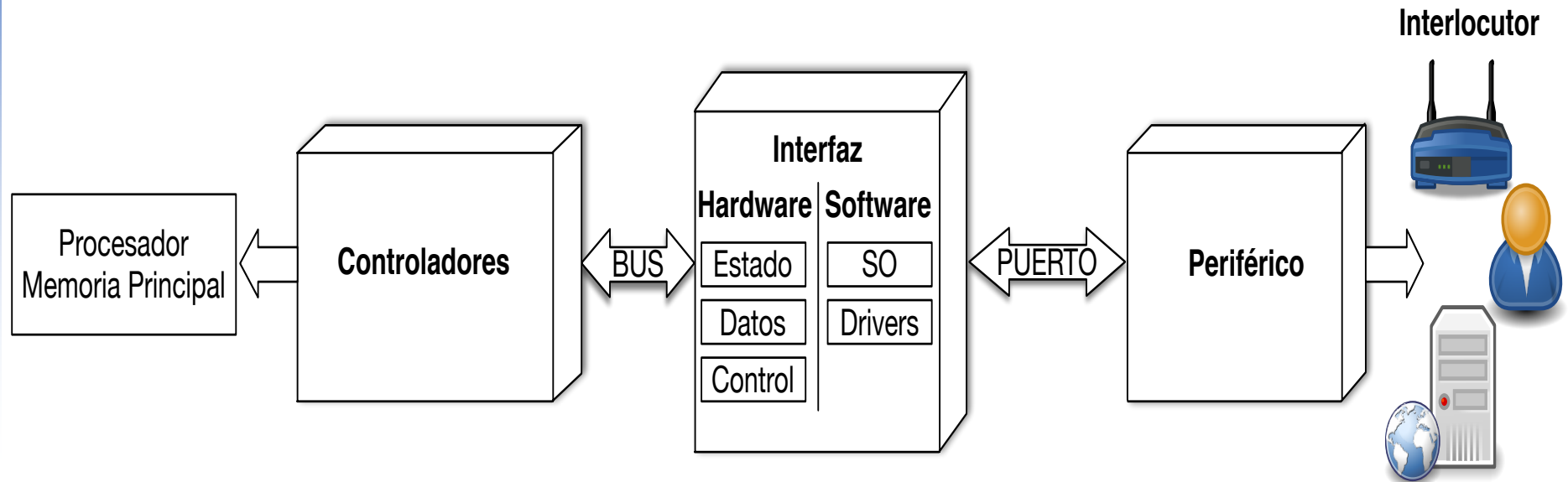
Objetivos

- Dispositivos de E/S
- Buses
- **Gestión del sistema de E/S**
 - **Direccionamiento**
 - **Transferencia de información**
 - **Gestión de la transferencia**

Elementos del sistema de E/S

- **Dispositivo/Periférico**
 - Mecánico, electromecánico o electromagnético
 - Se relaciona con el exterior
- **Puerto**
 - Permite al dispositivo conectarse al interfaz
- **Interfaz hardware/software**
- **Controladores**

Mecanismo completo de una operación de E/S



- Funciones del sistema de E/S
 - Direcccionamiento
 - Transferencia de información
 - Gestión de la transferencia

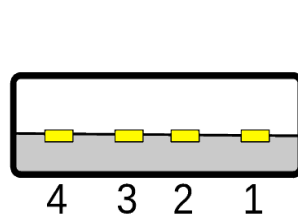
Periféricos

- Comportamiento
 - **Entrada** (lectura)
 - **Salida** (escritura)
 - **Almacenamiento** (lectura-escritura)
- Interlocutor
 - **Humano**
 - **Máquina**
- Velocidad de transferencia de datos
 - Velocidad pico a la que se pueden transferir datos entre un dispositivo de E/S y la memoria principal o el procesador

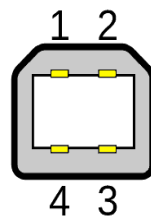
Puertos

- Son un caso especial de periféricos:
 - Canales de E/S de datos entre la CPU y el exterior de acuerdo a protocolos
 - De estos puertos cuelgan otros dispositivos de E/S
 - Permite al dispositivo conectarse con su interfaz
- Puertos serie o paralelo
 - Conexiones más utilizadas en el pasado
 - Prácticamente en desuso
- Conexiones actuales
 - Micro USB, fibra óptica, conexiones ethernet, conexiones inalámbricas (WiFi o Bluetooth)
- Puertos USB, *Universal Serial Bus* (Puerto Serie Universal)
 - Permite conectar prácticamente cualquier tipo de dispositivo

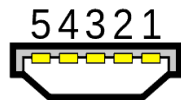
Puertos USB



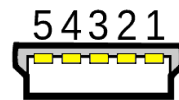
Type A



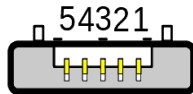
Type B



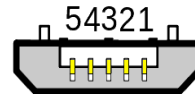
Mini-A



Mini-B



Micro-A



Micro-B



USB 2.0 Type A Plug



USB 2.0 Type A Jack



USB 2.0 Type B Plug



USB 2.0 Type B Jack



USB 2.0 Mini Type B Plug (4 Position)



USB 2.0 Type B Jack (4 Position)



USB 2.0 Mini Type B Plug (5 Position)



USB 2.0 Type B Jack (5 Position)



USB 3.0 Type A Plug



USB 3.0 Type A Jack



USB 3.0 Type B Plug



USB 3.0 Type B Jack



USB 2.0 Micro Type B Plug



USB 2.0 Micro Type B Jack



USB 3.0 Micro Type B Plug



USB 3.0 Micro Type B Jack

Periféricos

Device	Behavior	Partner	Data rate (Mbit/sec)
Keyboard	input	human	0.0001
Mouse	input	human	0.0038
Voice input	input	human	0.2640
Sound input	input	machine	3.0000
Scanner	input	human	3.2000
Voice output	output	human	0.2640
Sound output	output	human	8.0000
Laser printer	output	human	3.2000
Graphics display	output	human	800.0000–8000.0000
Modem	input or output	machine	0.0160–0.0640
Network/LAN	input or output	machine	100.0000–1000.0000
Network/wireless LAN	input or output	machine	11.0000–54.0000
Optical disk	storage	machine	80.0000
Magnetic tape	storage	machine	32.0000
Magnetic disk	storage	machine	240.0000–2560.0000

Interfaz hardware/software

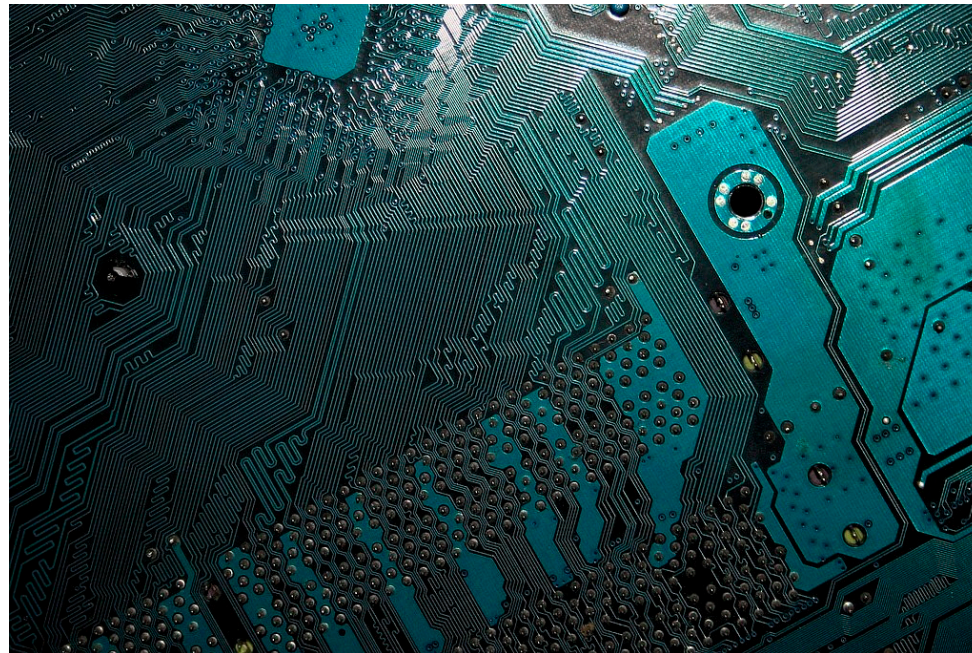
- Funciones principales
 - Interpretar las órdenes del procesador y transmitir las al periférico
 - Transferencia de datos
 - Informar al procesador del estado del periférico
- Parte software
 - Ligada a los drivers y al SO
- Ubicada en una tarjeta
- Clasificación
 - Serie o paralelo
 - Generales o específicos
- Conectado mediante el bus de E/S con un controlador ubicado en un chipset o hub

Controlador

- Misión del controlador
 - Transferencia de información con el procesador o con la memoria según los mecanismos de gestión de E/S disponibles en la arquitectura

Buses de E/S

- Sistema digital que transfiere datos entre los componentes de un computador
 - Siempre hay un **maestro** que dirige la transferencia y un **esclavo** que obedece las órdenes
 - Formado por cables o líneas en un circuito impreso



Buses de E/S: *función*

■ **Bus de control**

- Gobierna el uso y acceso a las líneas de datos y de direcciones
- Transmite tanto órdenes como información de temporización
- Consigue que no haya colisión de información en el sistema

■ **Bus de direcciones**

- La MP es direccionable: cada celda tiene su propia dirección
- Las direcciones son un número que selecciona una celda de memoria dentro de la MP o en el espacio de direcciones de E/S
- Es un canal del procesador independiente del bus de datos donde se establece la dirección de memoria del dato en tránsito
- La cantidad de memoria que se puede direccionar depende de la cantidad de bits que conforman el bus de direcciones

■ **Bus de datos**

- Intercambio de datos entre la CPU y el resto del sistema

Buses de E/S: *transferencia*

- Clasificación según transferencia de la información en **Serie** o **Paralelo**
 - Para distancias cortas el bus paralelo, para largas el serie
- **Bus paralelo**
 - Los datos son enviados por bytes al mismo tiempo, con la ayuda de varias líneas que tienen funciones fijas
 - Usado de manera intensiva, desde el bus del procesador, los buses de discos duros, tarjetas de expansión y de vídeo, hasta impresoras
- **Bus serie**
 - Los datos son enviados bit a bit, y luego se reconstruyen
 - Antes conectaban dispositivos lentos (teclado o ratón), actualmente se usan para conectar dispositivos rápidos como discos duros, unidades de estado sólido, tarjetas de expansión e incluso para el bus del procesador
 - Multicanal: se transmite por varios buses serie simultáneamente!
- Antes eran habituales los buses paralelos, quedando los buses serie dedicados para funciones de menor entidad y dispositivos lentos
- Ahora se están reemplazando los buses paralelos por buses serie multicanal, más difíciles de implementar, pero con velocidades de transferencia más elevadas y longitudes de cable mayores

Buses de E/S: *modo de operación*

■ **Protocolos de transferencia**

- Determinan el tipo de comunicación que se establece entre los dispositivos conectados al bus

■ **Protocolos de sincronización**

- Determinan el principio y el fin de cada transferencia de información

■ **Protocolos de arbitraje**

- Controlan el acceso al bus cuando más de un dispositivo puede actuar como maestro
- Evitan interbloqueos

Diseño de buses de E/S

- **Protocolos de transferencia**
 - Existen tantos como buses de E/S
- **Protocolos de sincronización**
 - Síncronos
 - Asíncronos
 - Semisíncronos
- **Protocolos de arbitraje**
 - Centralizados
 - Distribuidos

Diseño de buses de E/S: *sincronización*

■ Buses síncronos:

- Transferencias gobernadas por una señal de reloj compartida por los dispositivos conectados al bus
- Cada transferencia se realiza en un número fijo de ciclos de reloj
- Adaptación al dispositivo más lento
- El bus no puede ser muy largo

■ Buses asíncronos

- No existe señal de reloj
- Los dispositivos se sincronizan por medio de señales de control
- Menos eficientes

■ Buses semisíncronos

- Síncronos para dispositivos rápidos
- Asíncronos para dispositivos lentos

Diseño de buses de E/S: *arbitraje*

■ Protocolos centralizados

- El control del acceso es responsabilidad de un único dispositivo, el **árbitro**
- Conexiones más habituales
 - Estrella
 - Encadenado

■ Protocolos distribuidos

- La responsabilidad se comparte entre todos los dispositivos conectados al bus

Diseño de jerarquías de buses

- No se pueden conectar todos los dispositivos a un mismo bus
 - Disminución del rendimiento
 - Incompatibilidad
- Mejor utilizar una jerarquía de buses
 - Bus del sistema
 - Bus de memoria
 - Bus de expansión

Diseño de jerarquías de buses

■ Bus del sistema

- Conecta el procesador con el resto del sistema
- Corto, rápido, propietario (no estandarizado)
- Optimizado para arquitecturas y diseños específicos: conectan un número fijo de dispositivos de prestaciones conocidas

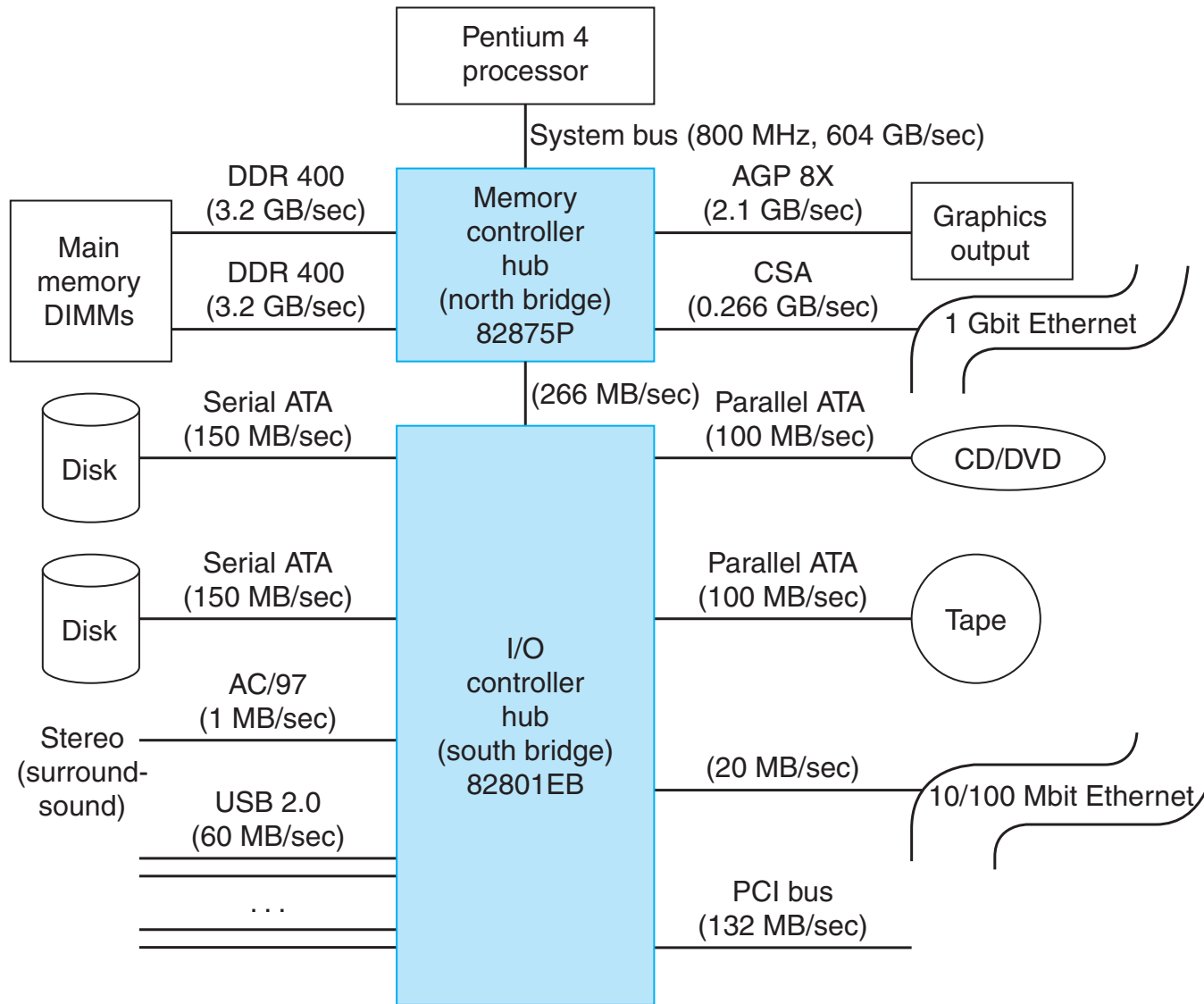
■ Bus de memoria

- Conecta la memoria principal con el controlador de memoria
- Mismas características que el bus del sistema

■ Bus de expansión

- Accesibles por el usuario
- Conecta un número indeterminado de dispositivos, de prestaciones desconocidas y muy diferentes entre sí
- Largos, lentos, abiertos (estandarizados)
- Ejemplo: bus PCI

Diseño de jerarquías de buses



Gestión del sistema de E/S

- Tareas necesarias para transferir datos desde un *dispositivo* hasta el espacio de direcciones de memoria de un *proceso de usuario*
- ¿Cómo se transforma una petición de usuario de E/S en un comando para el dispositivo y cómo se comunica dicho comando?
- ¿Cómo se transfieren los datos hacia y desde memoria?
- ¿Cuál es el papel del SO?

Gestión del sistema de E/S

- El SO hace de interfaz entre el hardware y el programa que realiza la petición de E/S
- Responsabilidades del SO:
 - Permitir que los programas **compartan** el sistema de E/S
 - Los sistemas de E/S utilizan **interrupciones** para comunicar información sobre el estado de las operaciones de E/S
 - **Ocultar** la complejidad del control de bajo nivel de un dispositivo de E/S

Gestión del sistema de E/S

- El SO debe garantizar que un programa sólo accede a la porción de un dispositivo de E/S para la que tiene permiso
- El SO proporciona abstracciones para acceder a los dispositivos de E/S
- El SO gestiona las interrupciones de la misma manera que maneja las excepciones generadas por un programa
- El SO trata de proporcionar un acceso equitativo a los dispositivos de E/S compartidos

Gestión del sistema de E/S

- Para realizar estas funciones, el SO necesita de tres tipos de comunicación:
 1. El SO debe ser capaz de **ordenar comandos (direccionar)** a los dispositivos: read, write, disk seek ...
 2. El dispositivo debe poder **notificar al SO la finalización** de una operación de E/S o un error (sincronización)
 3. Los datos se deben **transferir entre la memoria y el dispositivo** de E/S

Gestión del sistema de E/S: *direccionamiento*

- El procesador debe ser capaz de:
 - **Seleccionar** el dispositivo con el que quiere realizar la transferencia
 - **Identificar** al dispositivo que le hace alguna solicitud
- **Soluciones:**
 - E/S mapeada en memoria
 - Instrucciones especiales para E/S

Gestión del sistema de E/S: *direccionamiento*

- El procesador debe ser capaz de:
 - **Seleccionar** el dispositivo con el que quiere realizar la transferencia
 - **Identificar** al dispositivo que le hace alguna solicitud
- **Soluciones:**
 - E/S mapeada en memoria
 - E/S aislada

Gestión del sistema de E/S: *direccionamiento*

■ E/S mapeada en memoria

- El sistema de E/S y la MP comparten el *mismo espacio de direcciones*
- A cada periférico les corresponden varias direcciones del mapa de E/S
 - A estas direcciones de E/S se les llama *puertos* (no confundir con los puertos serie, paralelo o USB)
- Las operaciones de E/S se realizan con instrucciones de *load* y *store*

■ Ventajas e inconvenientes

- Se reduce la MP que se puede emplear para almacenar instrucciones y datos
- Se simplifica el repertorio de instrucciones del procesador
- Se pueden realizar todo tipo de operaciones (aritméticas, de manipulación de bits, etc)

Gestión del sistema de E/S: *direccionamiento*

- **Ejemplo de E/S mapeada en memoria:** una operación de escritura sobre un dispositivo se utiliza para ordenar un comando
 - El procesador pone el dato y la dirección en el bus de memoria, que son ignorados por el controlador de memoria al tratarse de una dirección reservada a la E/S
 - El controlador del dispositivo sí que reconoce la dirección en el bus, graba el dato y lo transmite al dispositivo como un comando
- **Protección:** los programas de usuario no pueden realizar operaciones de E/S directamente
 - **Las direcciones de E/S están protegidas por el mecanismo de traducción de direcciones**
- En cada evento, la dirección utilizada codifica la identidad del dispositivo

Gestión del sistema de E/S: *direccionamiento*

■ E/S aislada

- El sistema de E/S y la MP utilizan un *espacio de direcciones diferente*
- Una línea en el bus de direcciones especifica si el direccionamiento es de memoria o de E/S
 - Un puerto puede tener la misma dirección que una posición válida de memoria
- Las operaciones de E/S se realizan con *instrucciones específica* (tipo *in* y *out*)

■ Ventajas e inconvenientes

- La MP se emplea sólo para almacenar instrucciones y datos
- Se complica el repertorio de instrucciones del procesador

■ Protección: **sólo se permite la ejecución de instrucciones específicas de E/S en modo supervisor**

Gestión del sistema de E/S: sincronización

- Asincronía y dispositivos lentos
 - Resolver problemas de temporización
- Se debe tener en cuenta la temporización para:
 - el *inicio* de la transferencia
 - la *transferencia* propiamente dicha
- Según el grado de intervención de la CPU tenemos:
 - **E/S programada**
 - La CPU tiene todo el protagonismo, inicia y lleva a cabo la transferencia
 - **E/S por interrupciones**
 - La CPU ejecuta la transferencia, pero el inicio lo solicita el periférico
 - **Acceso directo a memoria (DMA)**
 - La transferencia la realiza un controlador especializado

Gestión del sistema de E/S: *sincronización*

- **E/S programada:** la CPU inicia y ejecuta la transferencia
- La transferencia puede ser:
 - **Incondicional**
 - La CPU no comprueba si el periférico está disponible
 - Se pueden hacer lecturas o escrituras erróneas
 - **Condicional**
 - La CPU primero comprueba si el dispositivo está disponible
 - Sondeo de un registro de estado o ***polling***
- Desventajas ***polling***
 - El procesador no hace trabajo útil durante el bucle
 - El programa que se estaba ejecutando se detiene
 - Dificultades para atender a varios periféricos al mismo tiempo
 - Técnica que no se utiliza prácticamente nunca, salvo en algunos sistemas de tiempo real embebidos

Gestión del sistema de E/S: *sincronización*

■ E/S por interrupciones

- La CPU ejecuta la transferencia, pero el inicio lo solicita el periférico
- No hay un bucle de espera activo
- La petición de servicio se realiza mediante una señal de control específica, activando una *línea de petición de interrupción (IRQ)*
- Después, se ejecuta una Rutina de Tratamiento de Interrupción (**RTI**)

■ Las interrupciones son como las excepciones, salvo que:

1. **Una interrupción es asíncrona** con respecto a la ejecución de una instrucción, no está asociada con ninguna instrucción y no puede evitar que la instrucción se complete. Antes de comenzar la ejecución de una nueva instrucción, la unidad de control comprueba si existen interrupciones de E/S pendientes
2. Además de saber que se ha producido una interrupción, necesitamos conocer más información, como la **identidad del dispositivo que ha generado la interrupción**, así como la prioridad de las peticiones de interrupción

Gestión del sistema de E/S: *sincronización*

■ E/S por interrupciones: procesador secuencial

1. Un periférico activa la línea de petición de interrupción (IRQ)
2. Tras ejecutar la última instrucción, se comprueba si hay interrupciones pendientes
3. En caso afirmativo, se almacena en la pila el contexto de ejecución, se deshabilitan las interrupciones y se salta a la RTI:
 - Se informa al periférico que se ha reconocido su interrupción, vía software accediendo el registro de estado del interfaz, o vía hardware, mediante la señal INTA. El periférico desactiva IRQ
 - Se salvan en pila los registros de datos y direcciones utilizados por la RTI
 - Se realiza la operación de E/S con el periférico
 - Se restauran los registros de datos y de direcciones
 - Se ejecuta la instrucción de retorno de interrupción
4. Se habilitan las interrupciones, se restaura el contexto y se retorna al programa que se estaba ejecutando antes de atender la interrupción

Gestión del sistema de E/S: *sincronización*

- **E/S por interrupciones: procesador segmentado**
 - Se estarán ejecutando varias instrucciones al mismo tiempo
 - Considerando las interrupciones como excepciones
- Con muchos dispositivos, ¿cómo identificar la fuente?
 - Identificación software por *polling*
 - La RTI examina uno por uno los bits de estado de cada periférico hasta hallar el que tiene activado el bit de petición de interrupción
 - Se ejecuta su RTI
 - Mecanismo de prioridades: en caso de peticiones simultáneas, se atiende primero al periférico que se consulta en primer lugar
 - Identificación hardware por vectores
 - El periférico envía un código o número de vector al partir del cual se calcula la dirección de memoria de la RTI
 - Mecanismo de prioridades más flexible, pero número de dispositivos limitado por el número de bits del vector (con 4 bits 16 dispositivos)
 - Usar **códigos de grupo** + identificación dentro del grupo por *polling*

Gestión del sistema de E/S: *sincronización*

■ E/S por interrupciones: PIC

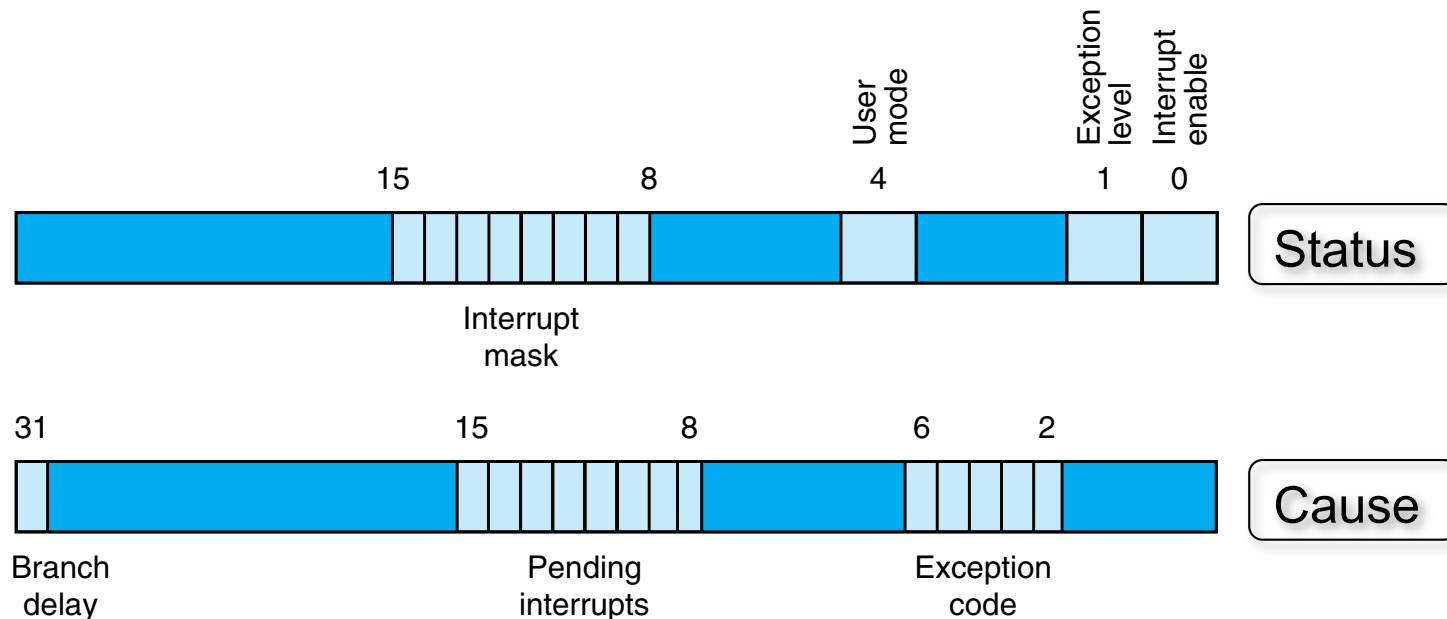
- *Peripheral Interface Controller*
- El PIC original se desarrolló en 1975 para mejorar el rendimiento del sistema liberando a la CPU de parte de la gestión de la E/S
- Se suelen usar dos o más controladores conectados en cascada

■ Interrupciones multinivel

- Existen varias líneas o niveles de petición de interrupción
- A cada línea se pueden conectar varios dispositivos
- Peticiones simultáneas por la misma línea
 - Polling o vector de interrupciones
- Peticiones simultáneas por líneas distintas
 - Codificador de prioridades dentro del PIC
 - Se atiende primero a la línea más prioritaria
- Registro de máscara de interrupción
- Anidamiento de interrupciones

Gestión del sistema de E/S: *sincronización*

- **E/S por interrupciones**, niveles de prioridad de interrupción
- Registros clave
 - **Status**: determina quién puede interrumpir. Si el bit de interrupción está a 1, el nivel de interrupción correspondiente está habilitado, a 0, el nivel está deshabilitado
 - **Cause**: el campo con el código de la excepción le indica al SO el motivo de la interrupción



Gestión del sistema de E/S: *sincronización*

- **E/S por interrupciones:** pasos en el manejo de una interrupción
 1. AND lógico *máscara de interrupción* e *interrupciones pendientes*:
interrupt mask AND pending interrupts
 2. **Seleccionar la interrupción de mayor prioridad:** las que están más a la izquierda tienen una prioridad mayor y se corresponden con los dispositivos más rápidos y con un mayor ancho de banda
 3. **Salvar** la máscara de interrupciones del registro de estado
 4. Cambiar la máscara de interrupciones para deshabilitar todas aquellas interrupciones con una prioridad menor o igual
 5. Salvar el estado del procesador necesario para manejar la interrupción
 6. Para permitir interrupciones de mayor prioridad, poner a 1 el bit de habilitación de interrupción
 7. **Llamar a la correspondiente rutina de tratamiento de interrupción**
 8. Antes de **restaurar** el estado, poner el bit de habilitación de interrupción a 0 para restaurar la máscara de interrupción

Gestión del sistema de E/S: *sincronización*

- **E/S por interrupciones:**
 - Interrupt Priority Levels, *IPL*
- El IPL es un mecanismo inventado por el SO
- Forma parte de la información de un proceso
- A cada proceso del sistema se le asigna un IPL
- Con el nivel más bajo de IPL
 - todas las interrupciones están permitidas
- Con el nivel más alto de IPL
 - todas las interrupciones están bloqueadas
- Incrementar o decrementar el IPL implica modificar la máscara de interrupciones

Gestión del sistema de E/S: *sincronización*

- **Acceso Directo a Demoria**
 - Direct Memory Access, **DMA**
 - La transferencia la realiza un controlador especializado
- Permite la transferencia entre un periférico y la MP sin intervención del procesador, salvo en la inicialización
- Sólo tienen sentido para **dispositivos con una ancho de banda grande**, cuando se mueven bloques de datos
- La transferencia no se lleva a cabo mediante instrucciones, sino por la activación de las señales de control de acceso a memoria
- Las transferencias por DMA implican capturar los buses de memoria por parte del controlador de DMA que los gobierna al margen de la CPU

Gestión del sistema de E/S: *transferencia*

- **Transferencia por *polling*:** el procesador carga los datos de los registros del dispositivo y los almacena en memoria
- **Transferencia por interrupciones:** el SO transfiere datos por medio de un número pequeño de bytes desde o hacia los dispositivos
 - Mientras se completan las transferencias, el SO realiza otras tareas
 - Cuando el último byte de una petición de E/S ha sido transferido y la operación de E/S ha finalizado, el SO informa al programa
- El SO y el procesador realizan todo el trabajo
- Estos métodos son **apropiados para dispositivos con un ancho de banda pequeño**

Gestión del sistema de E/S: *transferencia*

■ Transferencia por *DMA*

1. **Inicialización de la transferencia** - Se inicializa el DMA indicando:
 - El identificador del dispositivo (puerto del periférico)
 - La operación a realizar en el dispositivo (lectura/escritura)
 - La dirección de MP fuente o destino de los datos
 - El número de bytes a transferir
2. **Realización de la transferencia** - El DMA comienza la operación en el dispositivo y arbitra la interconexión:
 - Cuando el dato está disponible, lo transfiere
 - El DMA proporciona la dirección de MP para la lectura o escritura
 - Si la petición requiere más de una transferencia, la unidad de DMA genera la siguiente dirección de memoria e inicia la siguiente transferencia
 - La unidad de DMA puede completar una transferencia entera sin necesidad de molestar al procesador
3. **Finalización de la transferencia** - Una vez que la transferencia por DMA ha finalizado, el controlador interrumpe al procesador

Gestión del sistema de E/S: *transferencia*

■ Transferencia por *DMA*

- En aquellas ocasiones en las que el procesador tenga que acceder a MP, si esta está ocupada realizando una transferencia por DMA, **el procesador tendrá que esperar**
- Sin embargo, en la mayoría de los casos, **la utilización de MC evita tener que acceder a la MP**, dejando el ancho de banda de la memoria libre para ser utilizado por los dispositivos de E/S

■ Modos de transferencia por DMA

■ Transferencia DMA modo ráfaga

- Una vez el DMAC obtiene el control del bus, no lo libera hasta transferir el bloque completo
- Transferencia rápida, pero durante ese tiempo el procesador no tiene acceso a la MP, degradación del sistema

■ Transferencia DMA modo robo de ciclo

- El DMAC solicita, transfiere una palabra y libera el bus tantas veces como sea necesario hasta finalizar la transferencia del bloque
- Transferencia más lenta, pero el sistema no se degrada

Gestión del sistema de E/S: *transferencia*

- Transferencia por **DMA**: *¿direcciones virtuales o físicas?*
 - **DMA Virtual**, utiliza direcciones virtuales
 - Podemos sobrepasar los límites de una página
 - DMAC tiene un pequeño número entradas con las traducciones de dirección virtual a física para una transferencia
 - Este conjunto de entradas se lo proporciona el SO cuando se inicia la operación de E/S
 - **DMA Físico**, utiliza direcciones físicas
 - El SO ubica en MP las páginas implicadas en la transferencia en marcos consecutivos para no pasar el límite al incrementar o decrementar la dirección de memoria
 - En ambos casos, las páginas origen o destino de la transferencia deben bloquearse para no ser reemplazadas hasta que la transferencia no termine

Gestión del sistema de E/S: transferencia

- **Transferencia por *DMA*: jerarquía de caché**
- Como mínimo, pueden existir 3 copias de la misma información:
 - en MC, MP y MV
- Como las peticiones de la unidad de DMA van directamente a la memoria, **el contenido de una posición de memoria desde el punto de vista del procesador y del DMAC pueden ser diferentes**
- Situaciones:
 - Una transferencia del disco a la memoria puede provocar que algunas posiciones que están en caché queden desactualizadas. Una lectura posterior por parte del procesador de dichas ubicaciones en caché devolverá un contenido desactualizado
 - Del mismo modo, si la caché utiliza ***post-escritura***, el DMAC al leer directamente de la memoria obtendría un valor desactualizado
- A estos problemas se los denomina **problemas de coherencia**

Gestión del sistema de E/S: transferencia

■ Transferencia por *DMA*: soluciones jerarquía de caché

1. Hacer que todo el tráfico de E/S pase por la caché
 - Las lecturas verán el valor más actualizado
 - Las escrituras actualizarán el dato en la caché
 - Caro y puede tener un impacto negativo en el rendimiento del procesador
2. ***Invalidaciones selectivas*** de caché por parte del SO para una lectura de E/S o forzar el volcado de la caché
 - ***cache flushing***
 - Requiere menos soporte hardware y puede ser más eficiente si el software hace su trabajo de forma sencilla y eficiente
3. Proporcionar un **mecanismo hardware** qué, de forma selectiva, invalide o fuerce el volcado de la caché
 - La invalidación por hardware para asegurar la coherencia de la caché es típica en sistemas multiprocesador
 - Esta misma técnica se puede utilizar para el sistema de E/S