

•
•
•
•
•
•
•
•
•
•

Estructura y Tecnología de Computadores

Módulo E. El subsistema de E/S

Tema 9. Acceso directo a memoria (DMA) y procesadores de E/S

José Manuel Mendías Cuadros
Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid

• • • • • • •

2

•

contenidos

1. Acceso directo a memoria
Necesidad del DMA. El controlador de DMA. Etapas de una transferencia por DMA. Tipos de transferencias: por ráfaga y por robo de ciclo.
2. El controlador de DMA
Estructura. Registros del DMAC. Señales de control del periférico. Funcionamiento de una transferencia. Controlador DMA de varios canales.
3. Ejemplo
Controlador de DMA i82C37.
4. Procesadores de E/S
Función. Esquema típico de conexión. Tipos de procesadores de E/S: multiplexor, selector y multiplexor por bloques. Evolución de los procesadores de E/S.

estructura y tecnología de computadores

• • • • • • •

1. acceso directo a memoria

Necesidad de DMA

- ☒ La E/S con espera de respuesta o por interrupciones resulta inadecuada para periféricos de alta velocidad, sobre todo si hay que transferir una gran cantidad de datos
 - ☒ **Ejemplo periférico lento**
 - Procesador a 200 MHz (tiempo ciclo = 5 ns.; Ciclo medio por instrucción: CPI = 2 ciclos)
 - ⇒ Una instrucción tarda en promedio $2 \times 5 \text{ ns} = 10 \text{ ns}$ ⇒ el computador puede ejecutar ~100 MIPS
 - Queremos imprimir un fichero de 10 Kbytes en una impresora láser de 20 páginas por minuto
 - 1 página ≅ 3.000 caracteres (1 carácter = 1 byte)
 - ⇒ La impresora imprime 60.000 caracteres por minuto = 1 Kbyte/s
 - a) E/S con espera de respuesta
 - La CPU entra en un bucle y envía un nuevo byte cada vez que la impresora está preparado para recibirlo
 - ⇒ La impresora tarda 10 s en imprimir 10 Kbyte
 - ⇒ **La CPU está ocupada con la operación de E/S durante 10 s**
(en ese tiempo la CPU podría haber ejecutado 1000 millones de instrucciones)
 - b) E/S por interrupciones
 - La impresora genera una interrupción cada vez que está preparada para recibir un nuevo byte
 - ⇒ Suponemos que la RTI tiene 10 instrucciones (salvar contexto, comprobar estado, transferir byte, restaurar contexto, RTE)
 - ⇒ Para transferir 10 Kbyte tenemos que ejecutar 10.000 veces la RTI
 - ⇒ hay ejecutar 100.000 instrucciones para atender al periférico ⇒ la CPU tarda 0,001 s
 - ⇒ **La CPU está ocupada con la operación de E/S durante 0,001 s**
- CONCLUSIÓN**
- La E/S por interrupciones reduce en 10.000 veces el tiempo que la CPU está ocupada gestionando la impresora

estructura y tecnología de computadores

• • • • • • • •

1. acceso directo a memoria

Necesidad de DMA (cont.)

- ☒ **Ejemplo periférico rápido**
 - Procesador a 200 MHz (tiempo ciclo = 5 ns.; Ciclo medio por instrucción: CPI = 2 ciclos)
 - ⇒ Una instrucción tarda en promedio $2 \times 5 \text{ ns} = 10 \text{ ns}$ ⇒ el computador puede ejecutar ~100 MIPS
 - Disco con velocidad de transferencia de 10 Mbytes/s (1 byte cada 2×10^{-7} seg)
 - Queremos transferir un fichero de memoria a disco de 10 Mbytes
 - a) E/S con espera de respuesta
 - La CPU entra en un bucle y envía un nuevo byte cada vez que el disco está preparado para recibirlo
 - ⇒ El disco tarda 1 seg en recibir un fichero de 10 Mbyte
 - ⇒ **La CPU está ocupada con la operación de E/S durante 1 s**
(en ese tiempo la CPU podría haber ejecutado 200 millones de instrucciones)
 - b) E/S por interrupciones
 - El disco genera una interrupción cada vez que está preparado para recibir un nuevo byte
 - ⇒ Suponemos que la RTI tiene 10 instrucciones (salvar contexto, comprobar estado, transferir byte, restaurar contexto, RTE)
 - ⇒ Para transferir 10 Mbytes tenemos que ejecutar 10^7 veces la RTI
 - ⇒ hay ejecutar 100 millones de instrucciones para atender al periférico ⇒ la CPU tarda 1 s
 - ⇒ **La CPU está ocupada con la operación de E/S durante 1 s**
- CONCLUSIÓN**
- La E/S por interrupciones no mejora el tiempo que la CPU está ocupada en atender al periférico

La técnica de DMA permite la transferencia de datos entre un periférico y la memoria **sin intervención de la CPU** (salvo en la fase de inicialización de los parámetros de la transferencia)

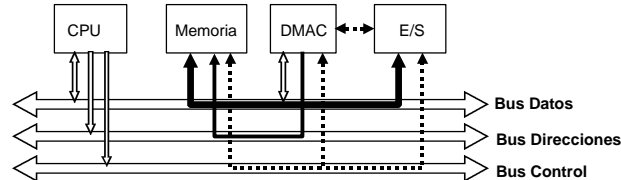
estructura y tecnología de computadores

• • • • • • • •

1. acceso directo a memoria

El controlador de DMA (DMAC)

- ☒ El controlador de DMA es un dispositivo capaz de controlar una transferencia de datos entre un periférico y memoria sin intervención de la CPU



- ☒ El DMAC debe actuar como máster del bus durante la transferencia DMA y debe ser capaz de
 - Solicitar el uso del bus mediante las señales y la lógica de arbitraje necesarias
 - Especificar la dirección de memoria sobre la que se realiza la transferencia
 - Generar las señales de control del bus
 - ⇒ Tipo de operación (lectura/escritura)
 - ⇒ Señales de sincronización de la transferencia

estructura y tecnología de computadores

• • • • • • •

1. acceso directo a memoria

Etapas de una transferencia por DMA

- ☒ **Inicialización de la transferencia**
 - La CPU debe enviar al interfaz del periférico y al DMAC los parámetros de la transferencia
 - Inicialización del interfaz (Bus master: CPU - Bus slave: Interfaz)
 - ⇒ Nº de bytes a transferir
 - ⇒ Tipo de transferencia (lectura/escritura)
 - ⇒ Otra información de control (pista, sector, etc.)
 - Inicialización del controlador DMA (Bus master: CPU - Bus slave: DMAC)
 - ⇒ Nº de bytes o palabras a transferir
 - ⇒ Tipo de transferencia (lectura/escritura)
 - ⇒ Dirección de memoria inicial para la transferencia
 - ⇒ Nº de canal (para DMAs con varios canales)
 - Después de la inicialización la CPU retorna a sus tareas y ya no se preocupa más de la evolución de la transferencia
- ☒ **Realización de la transferencia**
 - Cuando el periférico está listo para realizar la transferencia se lo indica al DMAC
 - El DMAC pide el control del bus y se realiza la transferencia entre el periférico y la memoria
 - ⇒ **Bus máster:** DMAC + Periférico - **Bus slave:** Memoria
 - ⇒ Después de la transferencia de cada palabra se actualizan los registros del DMA
 - ✓ Nº de bytes o palabras a transferir
 - ✓ Dirección de memoria
- ☒ **Finalización de la transferencia**
 - El DMAC libera el bus y devuelve el control a la CPU
 - El DMAC suele activar una señal de interrupción para indicar a la CPU la finalización de la operación de E/S solicitada

estructura y tecnología de computadores

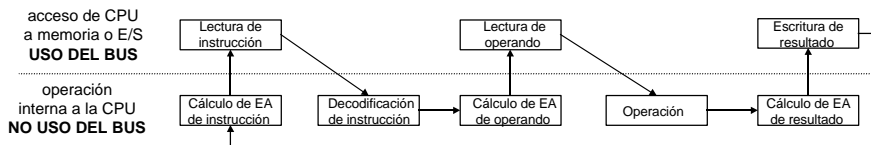
• • • • • • •

1. acceso directo a memoria

Tipos de transferencias

⊗ Problema que puede presentar el DMA

- Puede degradar el rendimiento de la CPU si el DMAC hace uso intensivo del bus
 - ⇒ Si el bus está ocupado en una transferencia DMA, la CPU no puede acceder a la memoria para leer instruc.
- Este problema se reduce con el uso de memoria cache
 - ⇒ La mayor parte del tiempo, la CPU lee instruc. de la cache, por lo que no necesita usar el bus de memoria
 - ⇒ El DMAC puede aprovechar estos intervalos en los que la CPU está leyendo instrucciones de la cache (y por tanto no usa el bus de memoria) para realizar las transferencias
- En caso de computadores sin cache
 - ⇒ El procesador no utiliza el bus en todas las fases de la ejecución de una instrucción
 - ⇒ El DMAC puede aprovechar las fases de ejecución de una instrucción en las que la CPU no utiliza el bus para realizar sus transferencias



⊗ Conclusión

- Si el DMAC sólo toma el control del bus durante los intervalos de tiempo en los que la CPU no hace uso del mismo ⇒ *el rendimiento del sistema no sufrirá degradación alguna*
- Se distinguen, por tanto, dos tipos de transferencias
 - Por ráfagas (burst)
 - Por robo de ciclo (cycle-stealing)

estructura y tecnología de computadores

• • • • • • • •

1. acceso directo a memoria

Tipos de transferencias (cont.)

⊗ Transferencia DMA modo ráfaga

- El DMAC solicita el control del bus a la CPU
- Cuando la CPU concede el bus el DMAC no lo libera hasta haber finalizado la transferencia de todo el bloque de datos completo
- **VENTAJAS:**
 - ⇒ La transferencia se realiza de forma rápida
- **DESVENTAJAS:**
 - ⇒ Durante el tiempo que dura la transferencia la CPU no puede utilizar el bus con memoria, lo que puede degradar el rendimiento del sistema

⊗ Transferencia DMA modo robo de ciclo

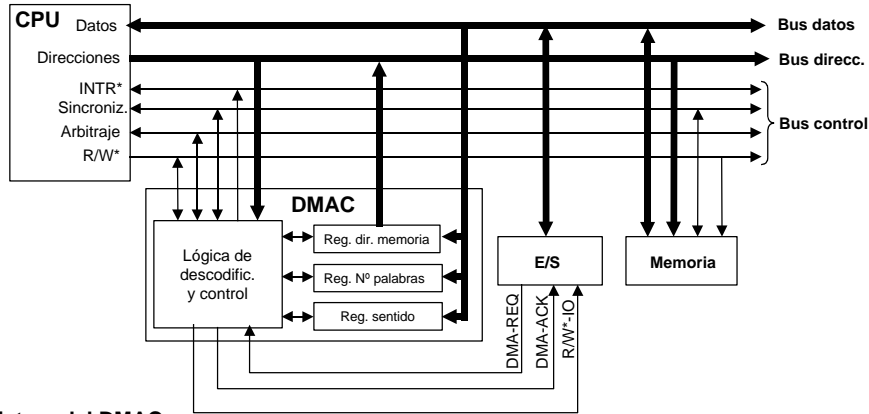
- El DMAC solicita el control del bus a la CPU
- Cuando la CPU concede el bus al DMAC, se realiza la transferencia de una única palabra y después el DMAC libera el bus
- El DMAC vuelve a solicitar el control del bus tantas veces como sea necesario hasta haber finalizado la transferencia del bloque completo
- La CPU cede el control del bus durante los ciclos que hace uso del mismo
- **VENTAJAS:**
 - ⇒ No se degrada el rendimiento del sistema
- **DESVENTAJAS:**
 - ⇒ La transferencia tarda más tiempo en llevarse a cabo

estructura y tecnología de computadores

• • • • • • • •

2. el controlador de DMA

Estructura hardware de un DMAC



Registros del DMAC

- ☒ **Reg. dir. memoria:** almacena la dir. inicial de memoria y se incrementa/decrementa después de transferir cada palabra
- ☒ **Reg. N° palabras:** almacena el número de palabras a transferir y se decrementa después de transferir cada palabra
- ☒ **Reg. sentido:** almacena el sentido de la transferencia (lectura o escritura)

estructura y tecnología de computadores

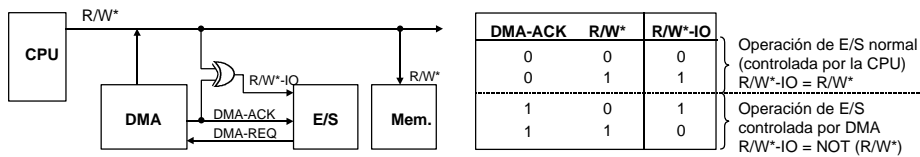
• • • • • • • •

2. el controlador de DMA

Señales de control del periférico

- ☒ **DMA-REQ:** solicitud de servicio DMA
 - La activa el periférico para indicar al DMAC que está listo para transmitir/recibir
- ☒ **DMA-ACK:** Concesión del servicio DMA
 - La activa el DMAC para indicar al periférico que puede realizar la transferencia
 - Antes de activar esta señal el DMAC debe estar en posesión del bus
- ☒ **R/W*-IO:** Sentido de la transferencia para el periférico
 - El periférico no puede utilizar la misma señal de R/W* que la memoria, ya que la transferencia tiene sentidos opuestos desde el punto de vista de uno y otro dispositivo
 - **Operación DMA de lectura (memoria @ periférico)**
 - ⇒ Para la memoria es una lectura: R/W* = 1
 - ⇒ Para el periférico es una escritura: R/W*-IO = 0
 - **Operación DMA de escritura (periférico @ memoria)**
 - ⇒ Para la memoria es una escritura: R/W* = 0
 - ⇒ Para el periférico es una lectura: R/W*-IO = 1

Posible esquema de conexión de R/W* y R/W*-IO



estructura y tecnología de computadores

• • • • • • • •

2. el controlador de DMA

Funcionamiento de una transferencia DMA

☒ Fase de inicialización

- La CPU le indica al interfaz del periférico la operación a realizar
- La CPU accede a los registros del DMAC para indicar los parámetros de la transferencia
 - ⇒ Dirección inicial de memoria → *Reg. dir. memoria*
 - ⇒ Número de palabras a transferir → *Reg. Nº palabras*
 - ⇒ Sentido de la transferencia → *Reg. Sentido*
- La CPU regresa a sus tareas

☒ Fase de transferencia

- Cuando el periférico está listo para transmitir/recibir lo indica al DMAC activando la señal **DMA-REQ**
- El DMA solicita el control del bus mediante las líneas de arbitraje
- El DMA recibe la concesión del bus y activa la señal **DMA-ACK** para indicar al periférico que puede iniciar la transferencia.
- El DMAC debe generar y procesar las señales del bus adecuadas
 - ⇒ Dirección de memoria sobre la que se realiza la transferencia
 - ⇒ Señales de sincronización de la transferencia (master-syncro, slave-syncro, etc.)
 - ⇒ Señales de lectura/escritura (R/W* y R/W*-IO)
- Después de transferir cada palabra el DMAC debe actualizar sus registros
 - ⇒ Decrementar el registro de nº de palabras
 - ⇒ Incrementar/decrementar el reg. de direcciones de mem. (según sean direcciones crecientes o decrecientes)
- En transferencias de robo de ciclo
 - ⇒ Se libera el bus después de transferir cada palabra y se vuelve a solicitar para transferir la siguiente
- En transferencias en modo ráfaga
 - ⇒ El DMAC mantiene el control del bus hasta que se ha transferido el bloque completo

☒ Fase de finalización de la transferencia

- Cuando el registro de nº de palabras llega a cero el DMAC activa la señal de interrupción para indicárselo a la CPU

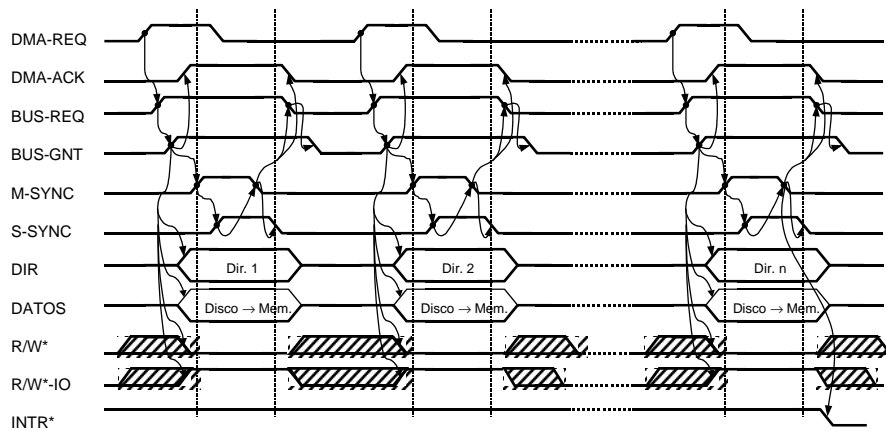
estructura y tecnología de computadores

• • • • • • • •

2. el controlador de DMA

Ejemplo de una transferencia DMA

- ☒ Transferencia modo robo de ciclo de disco a memoria (leer disco - escribir memoria)
- ☒ Protocolo de bus asíncrono y protocolo de arbitraje de dos hilos

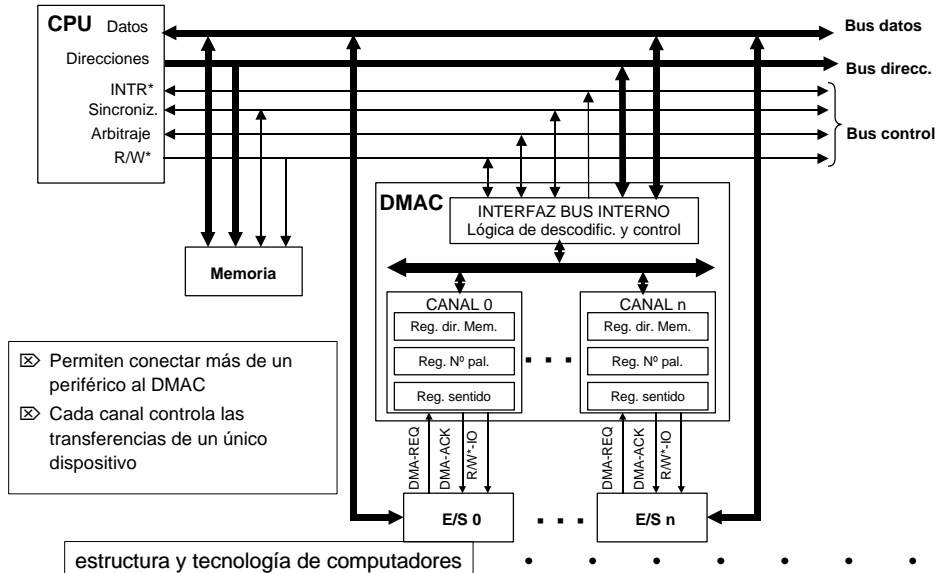


estructura y tecnología de computadores

• • • • • • • •

2. el controlador de DMA

Controlador DMA de varios canales



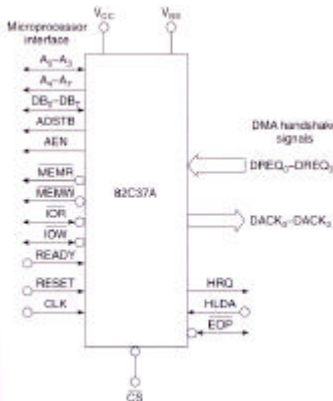
- ☒ Permiten conectar más de un periférico al DMAC
- ☒ Cada canal controla las transferencias de un único dispositivo

3. ejemplo: controlador de DMA i82C37

Controlador de DMA i82C37

- ☒ Diseñado para el i80386 (versión actualizada del i8237 e i82237 diseñados para i8086 e i80286 respect.)
- ☒ Controlador de 4 canales enmascarables selectivamente
- ☒ Trabaja con direcciones de 16 bits y puede realizar transferencias de 8 y 16 bits.
- ☒ Protocolo de transferencia semisíncrono y protocolo de arbitraje de 2 hilos (similar al i8086)

Líneas del controlador



DREQ0-3 (DMA Request): Señales de petición de servicio DMA (4 canales)

DACK0-3 (DMA Ack.): Señales de concesión de servicio DMA (4 canales)

HRQ (Hold Request): Señales de petición del control del bus

HLDA (Hold Ack.): Señales de concesión del control del bus

EOP* (End of Process): Señal bidireccional con 2 funciones

Puede activarla la CPU para abortar una transferencia DMA

Puede activarla el DMAC para informar a la CPU el fin de la transferencia

CS* (Chip Select): La activa la CPU para programar el DMAC en la inicializ.

CLK (Clock): Señal de reloj común a la CPU, DMAC, memoria y periféricos

READY: La activa la memoria o el periférico cuando no pueden realizar la transferencia DMA en el tiempo mínimo establecido

RESET: Reinicia el DMAC poniendo a cero los registros programables por la CPU

ADSTB (Address Strobe): Se utiliza para almacenar la dirección en un latch (registro) externo

AEN (Address Enable): Se utiliza para capacitar la salida del latch donde está almacenada la dirección y volcar la dirección sobre el bus

estructura y tecnología de computadores

3. ejemplo: controlador de DMA i82C37

Líneas del controlador (cont.)

MEMR*-MEMW* (memory read - memory write): La activa el DMAC para indicar que la transferencia es de tipo:

- ⇒ MEMR*: lectura de memoria (mem→perif). Se activa simultáneamente a IOW*
- ⇒ MEMW*: escritura en memoria (perif→mem). se activa simultáneamente a IOR*

IOR*-IOW* (I/O read - I/O write): Señales bidireccional con 2 funciones

- a) Durante la fase de inicialización (señales de entrada al DMAC)
La activa la CPU para indicar que se va a leer (IOR*) o escribir (IOW*) alguno de los registros del DMAC
- b) Durante la fase de transferencia DMA (señales de salida del DMAC)

La activa el DMAC para indicar al periférico que la transferencia es de tipo:

- ⇒ IOR*: lectura de E/S (perif→mem). Se activa simultáneamente a MEMW*
- ⇒ IOW*: escritura en E/S (mem→perif). Se activa simultáneamente a MEMR*

DB0-DB7 (Data bus): Señales con 2 funciones

- a) Durante la fase de inicialización
La usa la CPU para leer o escribir sobre los registros del DMAC (registros de 16 bits ↔ necesita dos accesos)
- b) Durante la fase de transferencia DMA
Las utiliza el DMAC para especificar los 8 bits más signif. de la dir. de mem. sobre la que se realiza la transfer.

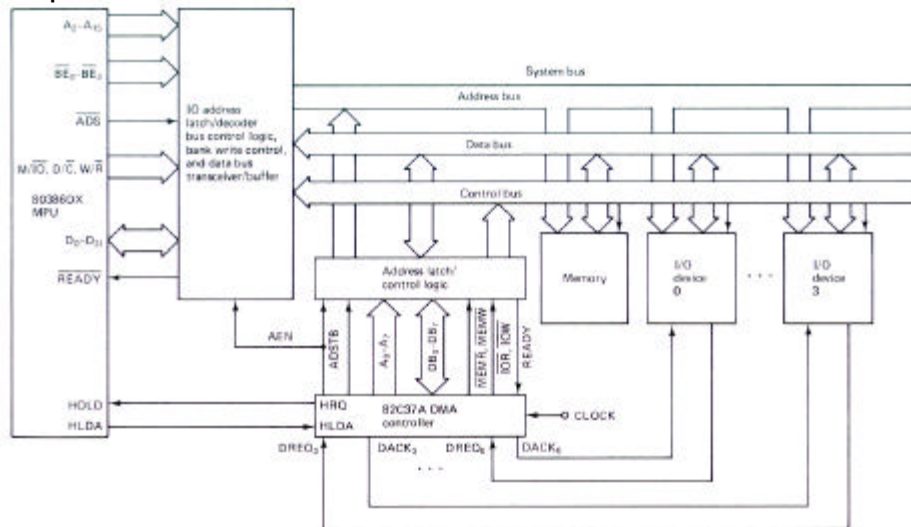
A0-A7 (Address): Señales con 2 funciones

- a) Durante la fase de inicialización
La CPU usa las líneas A0-A3 para indicar cual de los registros del DMAC va a leer o escribir
- b) Durante la fase de transferencia
Las utiliza el DMAC para especificar los 8 bits menos signif. de la dir. de mem. sobre la que se realiza la transfer.

estructura y tecnología de computadores

3. ejemplo: controlador de DMA i82C37

Esquema de conexión del controlador DMA i82C37



estructura y tecnología de computadores

3. ejemplo: controlador de DMA i82C37

Funcionamiento del controlador

- ☒ El controlador puede estar en dos estados: inactivo y activo
- ☒ **Estado inactivo (idle)**
 - El DMAC está en estado inactivo siempre que no hay ningún canal solicitando servicio DMA
 - En este estado la CPU puede leer o programar los registros del DMAC
 - ⇒ CPU activa la señal CS*
 - ⇒ Mediante las líneas A0-A3 la CPU selecciona el registro del DMAC
 - ⇒ Mediante IOR*/IOW* la CPU indica si se va a leer o escribir el registro seleccionado
- ☒ **Estado activo (busy)**
 - El DMAC pasa a estado activo cuando se activa alguna de las señales DREQi
 - Entonces el DMAC solicita el control del bus y cuando la CPU lo concede se realiza la transferencia
 - La transferencia puede ser de 2 tipos: simple o de bloque
- ☒ **Modo de transferencia simple (robo de ciclo)**
 - Cuando el DMAC controla el bus se transfiere una única palabra (de 8 o 16 bits según el ancho programado)
 - Después de cada palabra el DMAC libera el bus y vuelve a solicitarlo hasta que se finaliza la transferencia completa
 - El periférico puede desactivar su señal DREQi una vez que detecta DACKi activada
 - ⇒ El bus se libera después de cada palabra aunque el periférico mantenga activada la petición DREQi
- ☒ **Modo de transferencia de bloque (modo ráfaga)**
 - Cuando el DMAC toma el control del bus se transfiere un bloque completo hasta que el cont. de palabras llega a cero
 - El periférico puede desactivar su señal DREQi una vez que detecta DACKi activada

estructura y tecnología de computadores



3. ejemplo: controlador de DMA i82C37

Programación del controlador DMA i82C37

- ☒ **Registros de direcciones y contadores de palabras (16 bits cada uno)**
 - Dos registros de direcciones (base y actual) y dos registros contadores de palabras (base y actual) por cada canal
 - ⇒ 16 bits cada uno
 - **Registro de dirección base y registro contador base**
 - ⇒ Almacenan la dirección inicial de la transferencia y el nº inicial de palabras a transferir
 - ⇒ Su contenido no se modifica durante toda la transferencia
 - ⇒ Permiten realizar varias transferencias similares sin necesidad de reprogramar el DMAC
 - **Registro de dirección actual y registro contador actual**
 - ⇒ Almacenan la dirección de la siguiente palabra a transferir y el nº de palabras que quedan por transferir
 - ⇒ Se actualizan después de la transferencia de cada palabra
 - ⇒ Cuando el contador de palabras actual llega a cero indica que la transferencia ha finalizado
- ☒ **Registro de órdenes (8 bits)**
 - Permiten a la CPU programar el funcionamiento del DMAC. Las principales funciones que pueden programarse son:
 - ⇒ **Activación del DMAC:** DMAC activado o desactivado
 - ⇒ **Tipos de prioridad de los canales:** fija o rotante
 - ⇒ **Forma de las señales DREQi:** activadas a alta o a baja
 - ⇒ **Forma de las señales DACKi:** activadas a alta o a baja
- ☒ **Registros de modo (uno por canal, 6 bits cada uno)**
 - Permiten a la CPU programar el funcionamiento de cada canal. Las principales funciones son:
 - ⇒ **Sentido de la transferencia:** lectura o escritura
 - ⇒ **Modo de transferencia:** simple o de bloque
 - ⇒ **Orden de direcciones de memoria:** creciente o decreciente

estructura y tecnología de computadores



3. ejemplo: controlador de DMA i82C37

Programación del controlador DMA i82C37 (cont.)

☒ Registro de máscara (4 bits)

- Cada canal tiene un asociado un bit de máscara programable por la CPU
- Si el bit de máscara de un canal está activado ese canal no puede atender peticiones DMA por su línea DREQ_i

☒ Registro de estado (8 bits)

- Permiten a la CPU consultar el estado del DMAC
 - ⇒ Bit 0/1/2/3 → Canal 0/1/2/3 ha alcanzado el final de la cuenta
 - ⇒ Bit 4/5/6/7 → Solicitud de servicio por canal 0/1/2/3 (DREQ0/DREQ1/DREQ2/DREQ3 activada)

Transferencias DMA

☒ Las etapas en una transferencia DMA son las siguientes

1. El periférico activa DREQ_i para solicitar servicio DMA
2. El DMAC solicita el control del bus activando HRQ
3. La CPU concede el uso del bus activando HLDA
4. El DMAC activa DACK_i
5. Se realiza la transferencia de una palabra
 - ⇒ Tras la activación de DACK_i cada transferencia individual dura como mínimo 3 ciclos de reloj
 - ⇒ Si el periférico o la memoria necesitan más tiempo deben desactivar la señal READY
6. Se actualizan los registros de dirección actual y contador de palabras actual
 - ⇒ Si la transferencia es modo simple el DMAC libera el uso del bus y se vuelve al paso 1
 - ⇒ Si la transferencia es modo de bloque se vuelve al paso 5

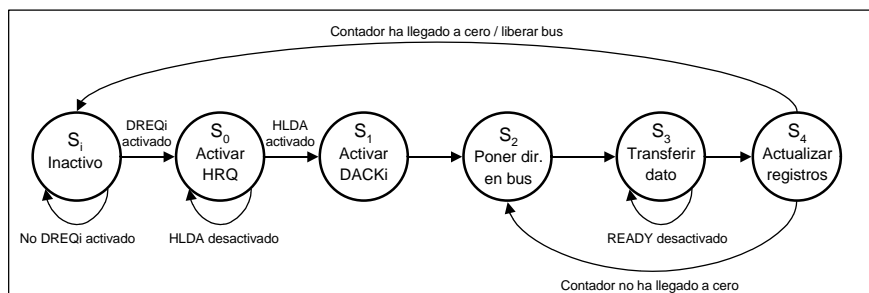
estructura y tecnología de computadores

• • • • • • •

3. ejemplo: controlador de DMA i82C37

Diagrama de estados de la transferencia DMA

Transferencia modo bloque



Transferencia modo simple

- ☒ Similar salvo que después de transmitir cada palabra se salta siempre de S₄ a S₁

estructura y tecnología de computadores

• • • • • • •

4. procesadores de E/S

Revisión de métodos de E/S

- ☒ **E/S programada con espera de respuesta**
 - La CPU ejecuta un bucle de espera hasta que el periférico está preparado para la transferencia
 - Durante una operación de E/S, la CPU está completamente dedicada a la atención del periférico
- ☒ **E/S por interrupciones**
 - El periférico es capaz de interrumpir a la CPU para indicar que está preparado
 - La CPU puede realizar otras tareas y solamente atiende al periférico cuando se produce una interrupción
 - La CPU es la encargada de controlar la operación de E/S mediante la ejecución de la RTI
- ☒ **Acceso directo a memoria (DMA)**
 - La CPU sólo debe preocuparse de inicializar el DMAC indicándole los parámetros de la transferencia
 - Una vez inicializado, el DMAC se encarga de controlar totalmente la transferencia
 - Durante la transferencia de E/S la CPU puede dedicarse a otras tareas

Procesadores de E/S (PE/S) o canales

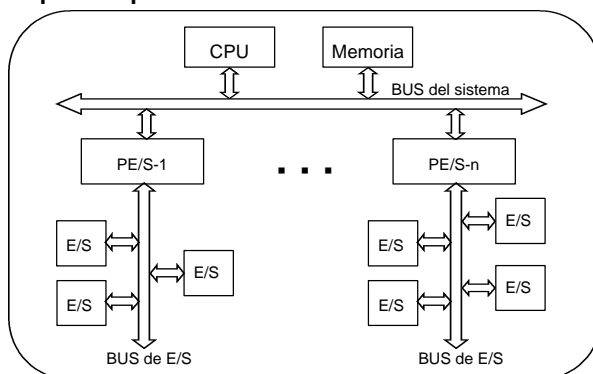
- ☒ Un PE/S o canal es un procesador auxiliar dedicado a realizar las operaciones de E/S
 - Disponen de un repertorio de instrucciones limitado y especializado en operaciones de E/S
 - Son capaces de controlar totalmente las operaciones de E/S mediante la ejecución de instrucciones de E/S
- ☒ Las instrucciones o programas de E/S se almacenan en la memoria principal del computador
 - Para iniciar una transferencia de E/S la CPU únicamente indica al procesador de E/S la dirección de memoria donde comienza el programa de E/S
 - El procesador de E/S ejecuta estas instrucciones y controla totalmente la transferencia de datos
 - ⇒ La CPU queda completamente liberada de las operaciones de E/S

estructura y tecnología de computadores

• • • • • • •

4. procesadores de E/S

Esquema típico de conexión



Tipos de procesadores de E/S

- ☒ Dependiendo de su funcionamiento y del tipo de periféricos que se conectan se distinguen tres tipos de PE/S o canales:
 - Canal multiplexor
 - Canal selector
 - Canal multiplexor por bloques

estructura y tecnología de computadores

• • • • • • •

4. procesadores de E/S

Tipos de procesadores de E/S (cont.)

☒ Canal multiplexor

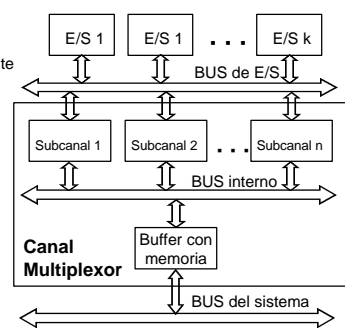
- Diseñado para trabajar con dispositivos de baja velocidad
- Puede manejar la transferencia de varios dispositivos simultáneamente

➢ Idea básica

- ⇒ El intercambio de datos entre el canal y memoria se realiza mucho más rápidamente que la transferencia entre el canal y cada uno de los periféricos individualmente
- ⇒ El canal puede atender de modo alternativo a los distintos periféricos durante cortos intervalos de tiempo en los que se transmite una palabra (multiplexación en el tiempo)

➢ Estructura

- ⇒ Está formado por varios subcanales, cada uno de los cuales atiende a un periférico
- ⇒ El nº de subcanales determina el nº de periféricos que pueden atenderse de forma simultánea



☒ Canal selector

- Diseñado para trabajar con dispositivos de alta velocidad
- Puede atender a varios dispositivos pero no de forma simultánea
- ⇒ Cuando se inicia una transferencia con un periférico no se puede atender a ningún otro hasta que la transferencia haya finalizado por completo

☒ Canal multiplexor por bloques

- Diseñado para trabajar con dispositivos de alta velocidad que transmiten su información en bloques separados
- El canal controla las transferencias de varios dispositivos de forma multiplexada en el tiempo pero por bloques
- ⇒ El canal atiende de forma alternativa a los distintos periféricos durante intervalos tiempo en los que se transmite un bloque de información

estructura y tecnología de computadores

4. procesadores de E/S

Evolución de los procesadores de E/S

☒ Procesadores de E/S o canales de propósito específico

- Los procesadores de E/S clásicos son procesadores específicamente diseñados para realizar operaciones de E/S
- Disponen de un conjunto reducido de instrucciones, que incluyen:
 - ⇒ Tipos de datos y modos de direccionamiento simples
 - ⇒ Instrucciones simples de aritmética básica (suma, incremento, decremento, complemento,...)
 - ⇒ Instrucciones de movimiento de datos (MOVE, IN, OUT, ...), que soportan movimientos de bloque
 - ⇒ Instrucciones específicas de control de periféricos (avance, rebobinado, etc.)

➢ Ejemplos

- ⇒ IBM 370 (diseñado para mainframes basados en el IBM 360)
- ⇒ Intel 8089 (procesador de E/S compatible con i8086)

☒ Microprocesadores de propósito general

- Actualmente es más común utilizar microcontroladores de E/S basados en microproc. RISC de propósito general

➢ Ejemplo: i890 RP IOP

- ⇒ Microcontrolador de E/S integrado basado en el procesador intel 80890 (procesador RISC de 32 bits)
- ⇒ Integra, entre otros, los siguientes elementos:
 - ✓ Procesador (+ cache de datos e intrucc.)
 - ✓ Controlador de memoria local
 - ✓ Bus interno de 32 bits
 - ✓ Soporte para conexión de 2 buses PCI
 - ✓ Controlador de interrupciones avanzado
 - ✓ Tres canales DMA independientes

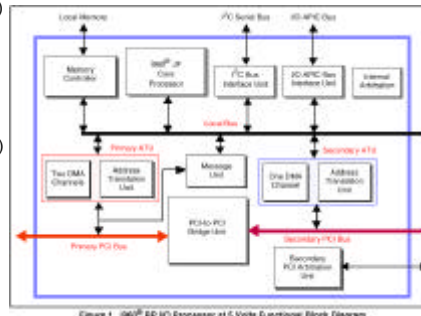


Figure 1. i890 RP IOP Processor at 5 Volts Functional Block Diagram

estructura y tecnología de computadores