

Estructura de Computadores



Curso
2017-2018

Sistema de Entrada/Salida
E/S por DMA

Rendimiento de la E/S



Supongamos una CPU con:

- $f=200$ MHz $\rightarrow T = 5ns$
- CPI = 2
 - Una instrucción tarda en promedio: $2 \times 5ns = 10ns$
 - El computador puede ejecutar ≈ 100 MIPS

Vamos a estimar lo que tardaría una operación de E/S con dos dispositivos:

- Impresión de un fichero de 10KB en una impresora láser de 20 páginas por minuto
 - 1 página = 3.000 caracteres (1 carácter = 1 byte)
 - 20 pag/min = 60.000 bytes/minuto = 1000 byte/s
- Transferencia de 10MB a un disco con ancho de banda de 10 MB/s.

Dispositivo lento



- A** E/S con espera de respuesta
 - La CPU envía el documento byte a byte (con espera activa)
 - La impresora tarda 10s en imprimir 10 Kbytes
 - 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 instruc. (salvar contexto, comprobar estado, transferir byte, restaurar contexto, RTE)
 - Para transferir 10 Kbytes tenemos que ejecutar 10.000 veces la RTI
 - Hay que ejecutar 100.000 instrucciones para atender al periférico →la CPU tarda 0,001s
 - La CPU está ocupada con la operación de E/S durante 0,001s

Conclusión:

- La E/S por interrupciones reduce en 10.000 veces el tiempo que la CPU está ocupada gestionando la impresora

Dispositivo rápido



A E/S con espera de respuesta

- La CPU envía 1 byte cada vez que el disco está listo (espera activa)
- El disco tarda 1 seg en recibir un fichero de 10 Mbytes
- La CPU está ocupada con la operación de E/S durante 1 s
- En ese tiempo la CPU podría haber ejecutado 10^8 instrucciones

B E/S por interrupciones

- El disco solicita interrupción cada vez que está listo para recibir 1 byte
- Suponemos que la RTI tiene 10 instrucciones
- Para transferir 10 Mbytes tenemos que ejecutar 10^7 veces la RTI, es decir 100 millones de instrucciones
- La CPU tarda 1s, ocupada 1s con la operación de E/S

Conclusión

- La E/S por interrupciones no mejora el tiempo que la CPU está ocupada en atender al periférico rápido
- Necesitamos que otro dispositivo se encargue → Controlador DMA

Acceso directo a memoria (DMA)



DMA:

- Técnica que 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
- Con interrupciones se evita el bucle de espera pero la transferencia la lleva a cabo el procesador
 - Para transferir N palabras se generan N interrupciones
- Con DMA toda la transferencia la realiza el controlador de DMA
 - Solo una interrupción al final

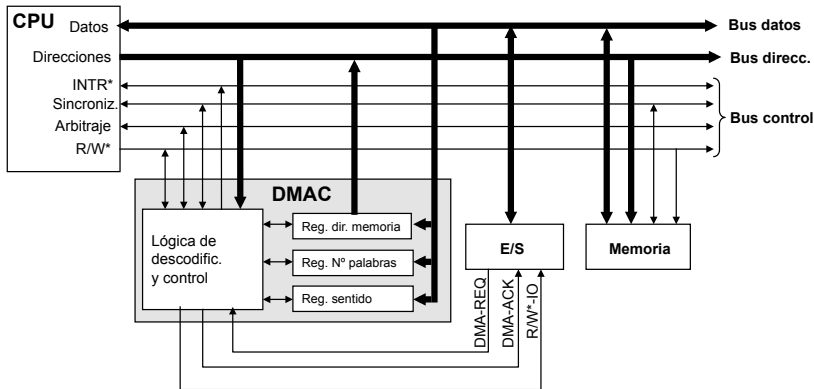
Controlador de DMA (DMAC):

- Dispositivo capaz de controlar una transferencia de datos entre un periférico y memoria, sin intervención de la CPU
- Debe actuar como máster del bus durante la transferencia DMA

Estructura del DMAC



- **Reg. dir. memoria:** dir. de memoria inicial
 - Se incrementa/decrementa después de transferir cada palabra
- **Reg. N^o palabras:** número de palabras a transferir
 - Se decrementa después de transferir cada palabra
- **Reg. sentido:** sentido de la transferencia (lectura o escritura)



Señales entre DMAC y periférico

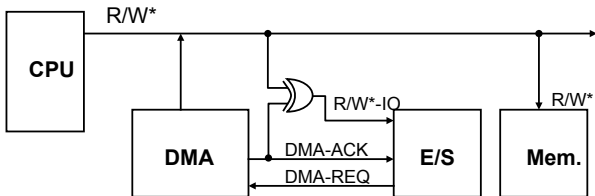


- **DMA-REQ**: solicitud de servicio DMA
 - Periférico indica al DMAC que está listo para transmitir/recibir
- **DMA-ACK**: Concesión del servicio DMA
 - DMAC indica al periférico que comienza la transferencia
 - El DMAC debe estar en posesión del bus
- **R/W*-IO**: Sentido de la transferencia para el periférico
 - Señal R/W* del bus indica el sentido de la transferencia para la memoria
 - Señal de R/W*-IO tiene valor opuesto
 - Lectura de memoria → Escritura en periférico
 - Escritura en memoria → Lectura de periférico

Generación de señales de control



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



DMA-ACK	R/W*	R/W*-IO	
0	0	0	Operación de E/S normal (controlada por la CPU) $R/W^*-IO = R/W^*$
0	1	1	
1	0	1	Operación de E/S controlada por DMA $R/W^*-IO = NOT(R/W^*)$
1	1	0	

Acceso directo a memoria

Transferencia en 3 etapas

- 1 Inicialización
- 2 Realización
- 3 Finalización



DMA: Inicialización



- 1** CPU configura el controlador de DMA (Bus Master: CPU, Bus Slave: DMAC)
 - N^o de bytes a transferir
 - Tipo de transferencia (lectura/escritura)
 - Dirección de memoria inicial
 - N^o de canal (para DMAs con varios canales)
- 2** CPU configura el controlador del periférico (Bus Master: CPU, Bus Slave: controlador del periférico)
 - N^o de bytes a transferir
 - Tipo de transferencia (lectura/escritura)
 - Otra información de control (pista, sector, ...)
- 3** Después de la inicialización la CPU retorna a sus tareas

DMA: Realización



- 1 Cuando el periférico está listo se lo indica al DMAC
 - Señal entre el DMAC y el controlador del periférico
- 2 El DMAC pide el control del bus
 - Arbitraje, el DMAC será el Master, la memoria el Slave
- 3 El DMAC realiza la transferencia, palabra por palabra
 - Después de transferir cada palabra se actualizan los registros del DMAC
 - N^o de bytes a transferir
 - Dirección de memoria

Modos de transferencia



- A** Modo ráfaga: el DMAC no suelta el bus hasta completar la transferencia
 - Transferencia rápida
 - La CPU no puede acceder a memoria durante la transferencia
 - Puede degradar el rendimiento del sistema
- B** Modo robo de ciclo: el DMAC suelta el bus después de transferir cada palabra
 - Se degrada poco el rendimiento del sistema
 - La transferencia tarda más tiempo en llevarse a cabo

DMA: finalización



- 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