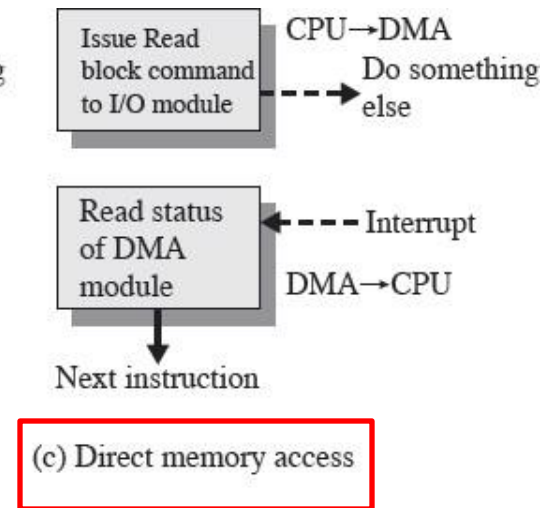
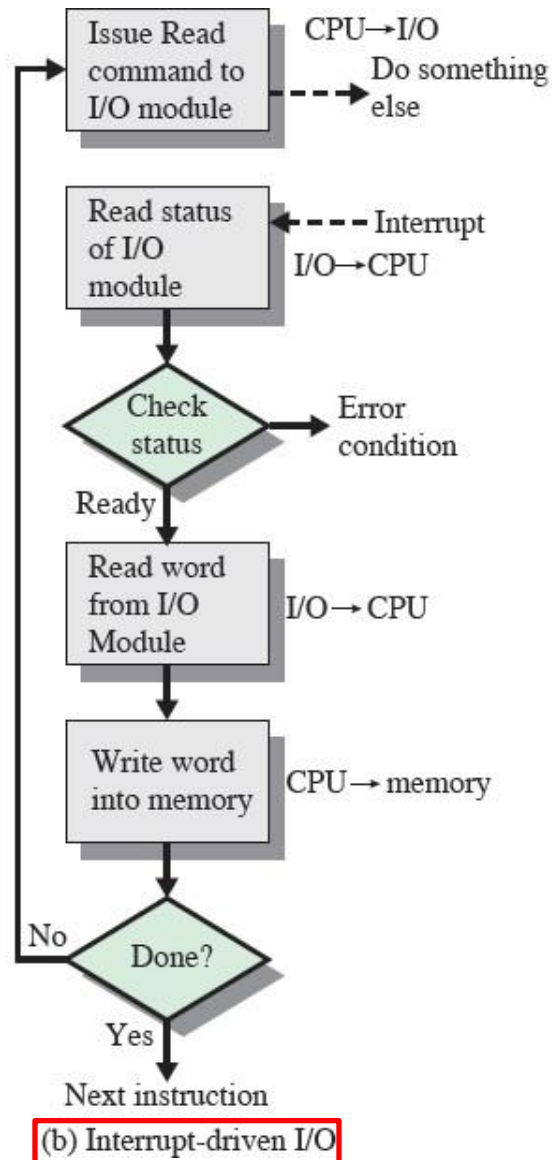
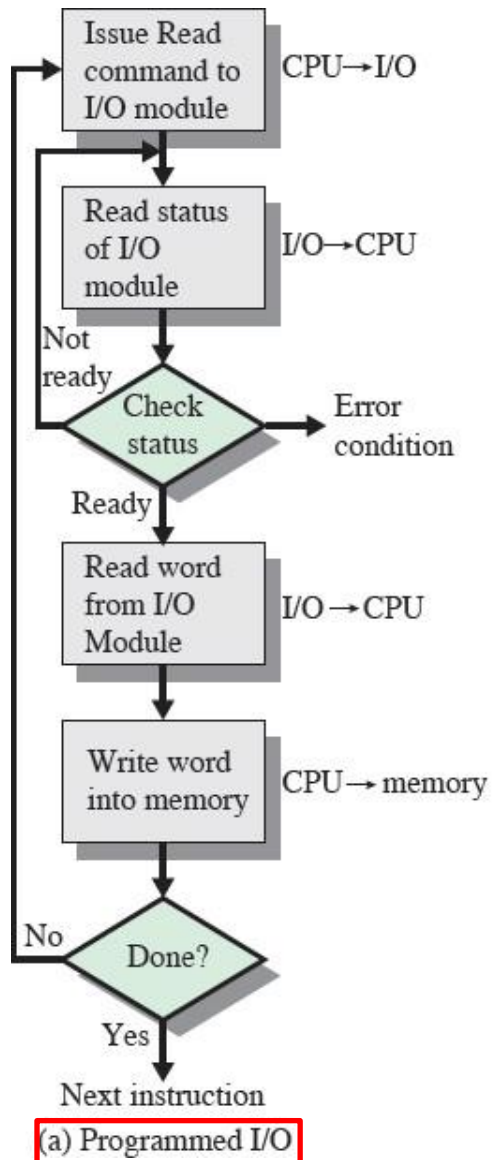


# Entrada / Salida III: Técnicas de E/S II

## Tres técnicas para la entrada de un bloque de datos



# Acceso directo a memoria (DMA)

El tratamiento por interrupción resulta inadecuado para la gestión de dispositivos de alta velocidad de transferencia.

## Ejemplo:

Velocidad de transferencia  $\rightarrow 10^5$  bytes /seg

Rutina de servicio de interrupción  $\rightarrow 100 \mu s = 10^{-4}$  seg

Para procesar  $10^5$  bytes  $\rightarrow 10^5 * 10^{-4} = 10$  seg

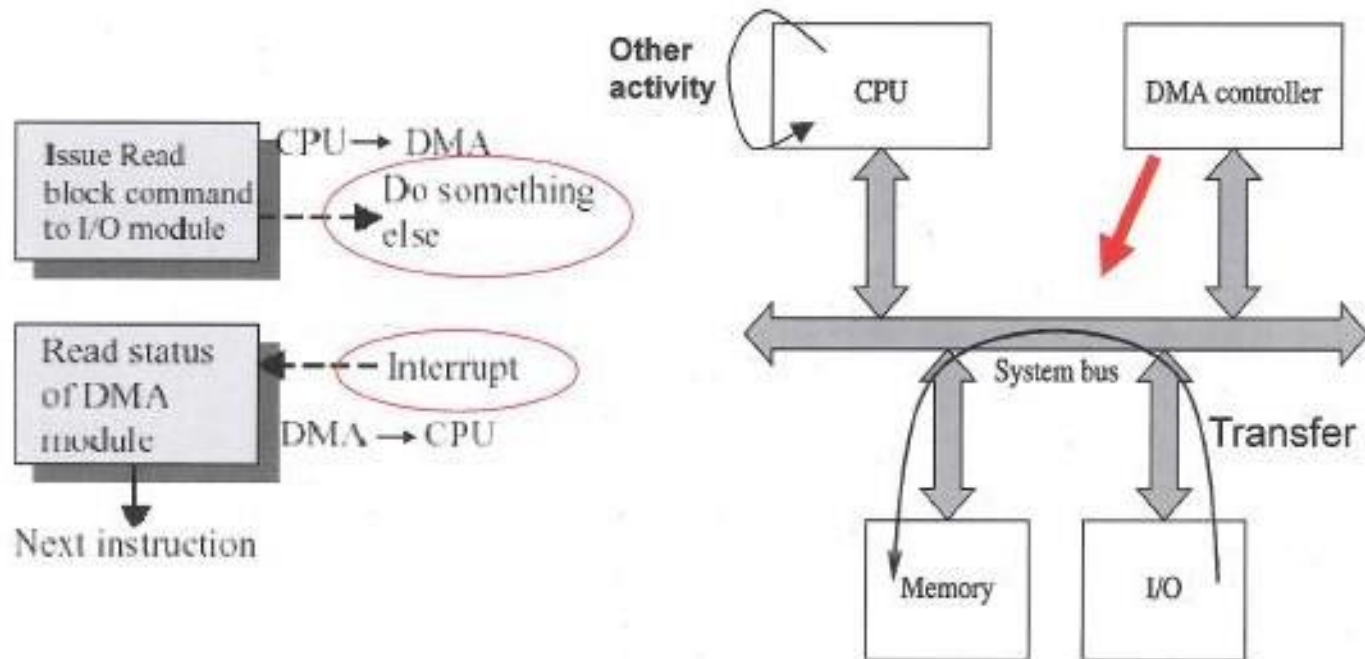
- Una rutina de servicio de interrupción puede contener muchas instrucciones ajenas a la transferencia del dato en sí.



Solución

**DMA**: Técnica que permite la transferencia de datos entre la memoria de un computador y sus periféricos sin intervención de la CPU.

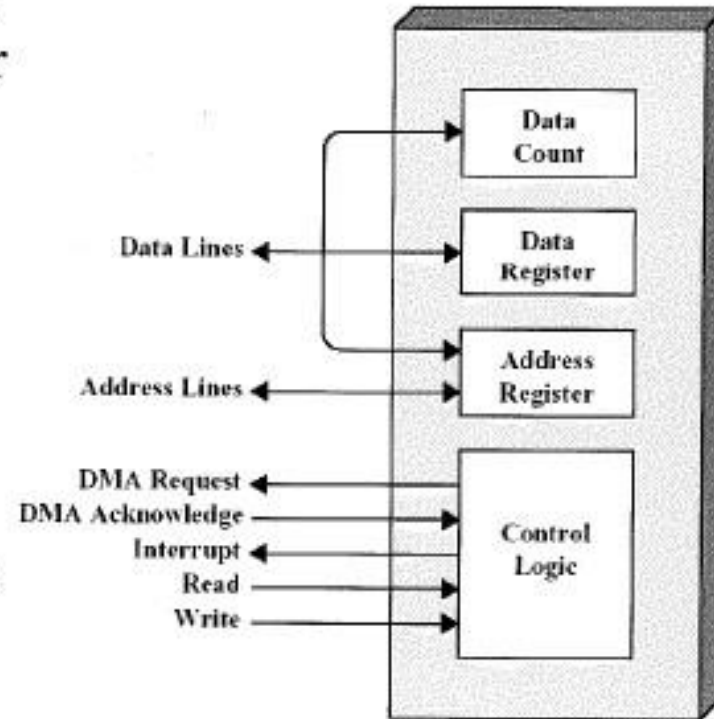
# E/S con DMA



- Interrupciones capturadas por controlador DMA
- Transferencia de datos mediante controlador DMA
  - CPU gestiona transferencias a nivel de bloques de datos

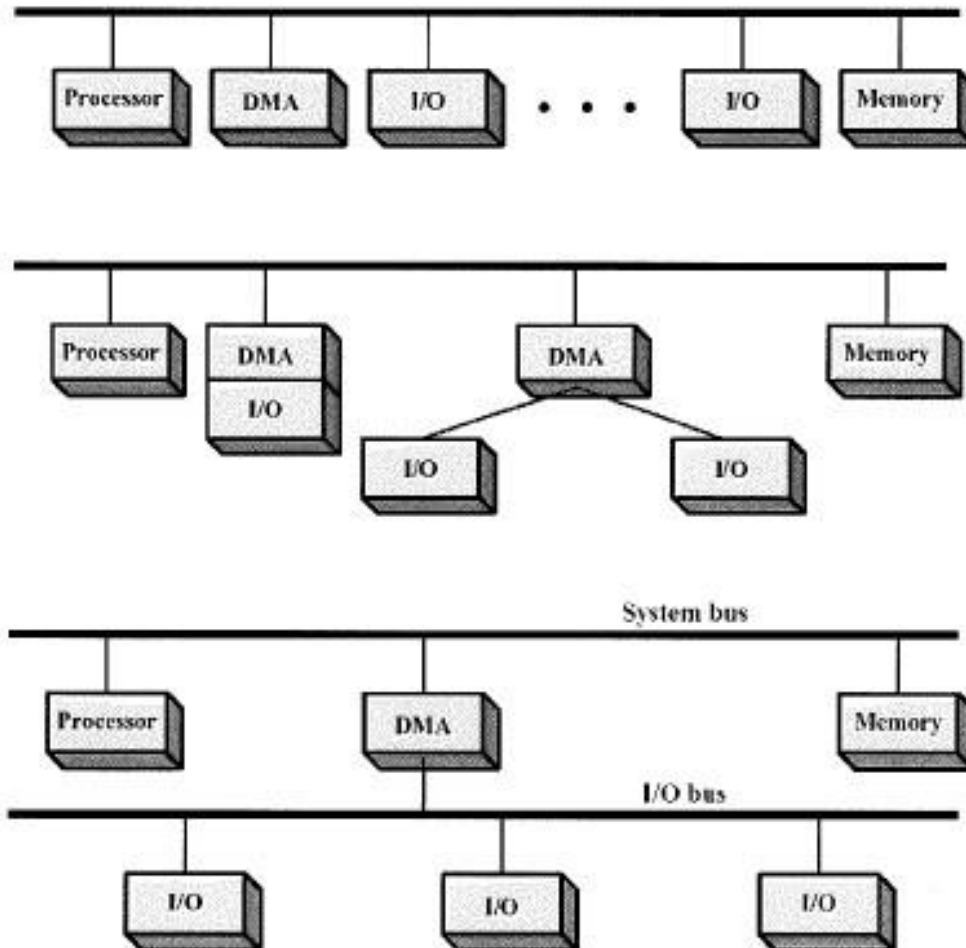
# Transferencia directa a memoria (DMA)

- Controlador de DMA: dispositivo que lleva a cabo transferencias de datos entre memoria y controlador de E/S de forma automática
- El procesador sólo ha de programar la transferencia
  - Dirección de memoria
  - Puerto del Controlador E/S
  - N° y Tamaño de Datos
  - Lectura / Escritura
- El procesador es avisado (con una interrupción) cuando la transferencia ha finalizado (correctamente, o con error)



# Configuraciones con DMA

- Bus único vs. Bus separado
- Control. DMA Independiente vs. Integrado
- Diferencias:
  - N° de usos del bus por transferencia
  - Límite en n° de controladores de E/S por DMA



Todos los controladores comparten el mismo bus

El controlador de DMA integra los dispositivos periféricos

Bus de E/S independiente

# Requerimientos de un sistema con DMA

- a) El procesador y el DMA pueden actuar como «masters»
  - Mecanismo de arbitraje del bus
  - Habitualmente el árbitro es el **procesador** utilizando acciones de *Request* y *Acknowledge*
- b) El DMA debe ser capaz de generar direcciones y señales de «timing» → Lógica necesaria para ello (registros, circuitos de control)

# Comportamiento funcional

1. El procesador envía al DMA los datos (es decir, el *comando*) que definen la transferencia a realizar (DMA SLAVE).

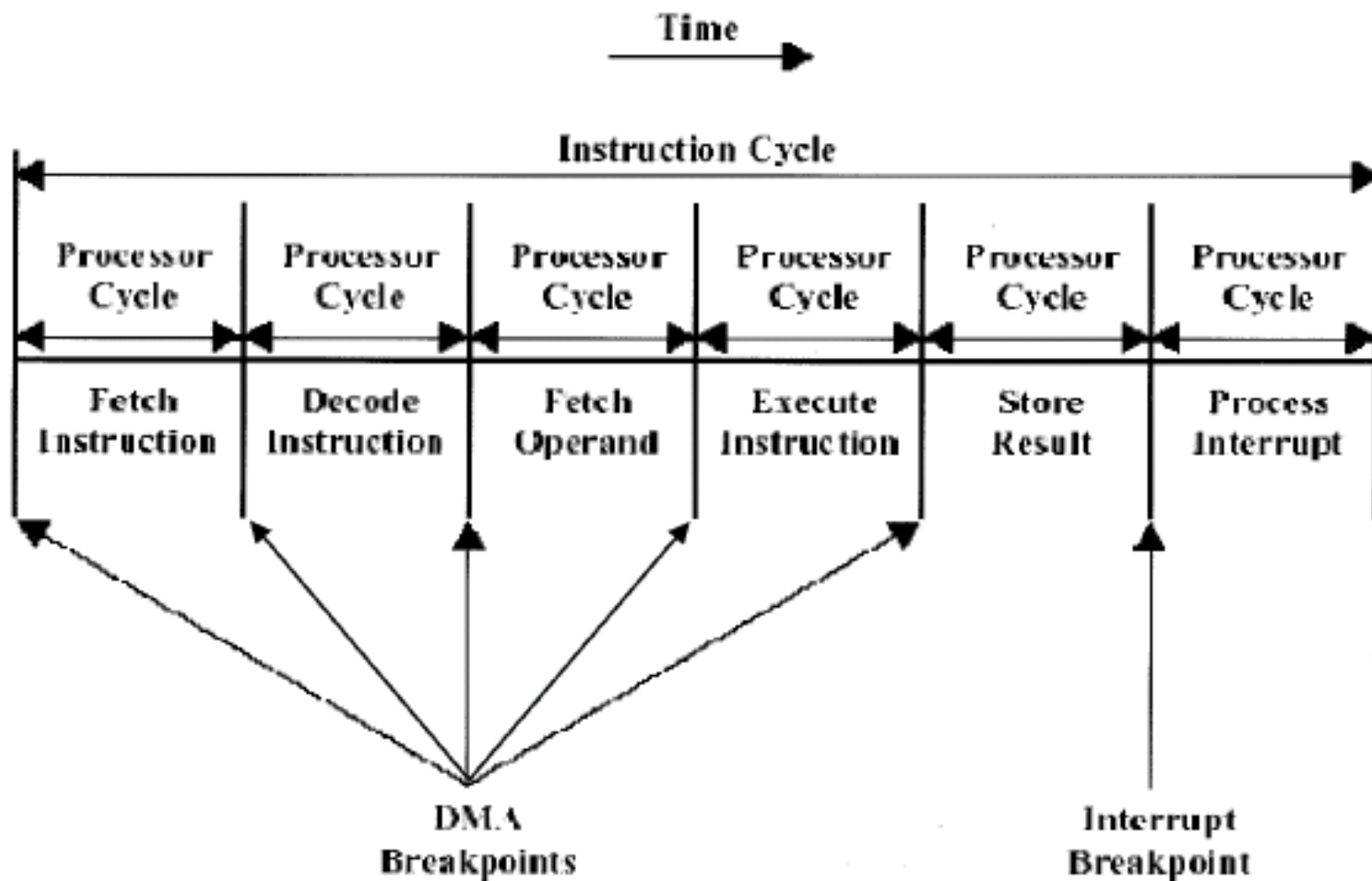
Estos datos son la DIRECCIÓN INICIAL DE MEMORIA, el NÚMERO DE PALABRAS, el SENTIDO, la DIRECCIÓN DEL PERIFÉRICO.

El procesador maneja al DMA como un periférico más.

2. El procesador retorna a otras tareas y el DMA prepara la transmisión.
3. Cuando el DMA necesita acceder a memoria, pide al árbitro el control del bus. El árbitro (el procesador) cede el bus (DMA MASTER) y el DMA accede al bus y genera señales necesarias. Finalmente, el DMA actualiza sus registros internos.
4. Cuando el DMA completa la operación de E/S pedida advierte al procesador, normalmente mediante interrupción.
5. El procesador reconoce y trata la interrupción. Si es necesario, el procesador ordena una nueva operación.



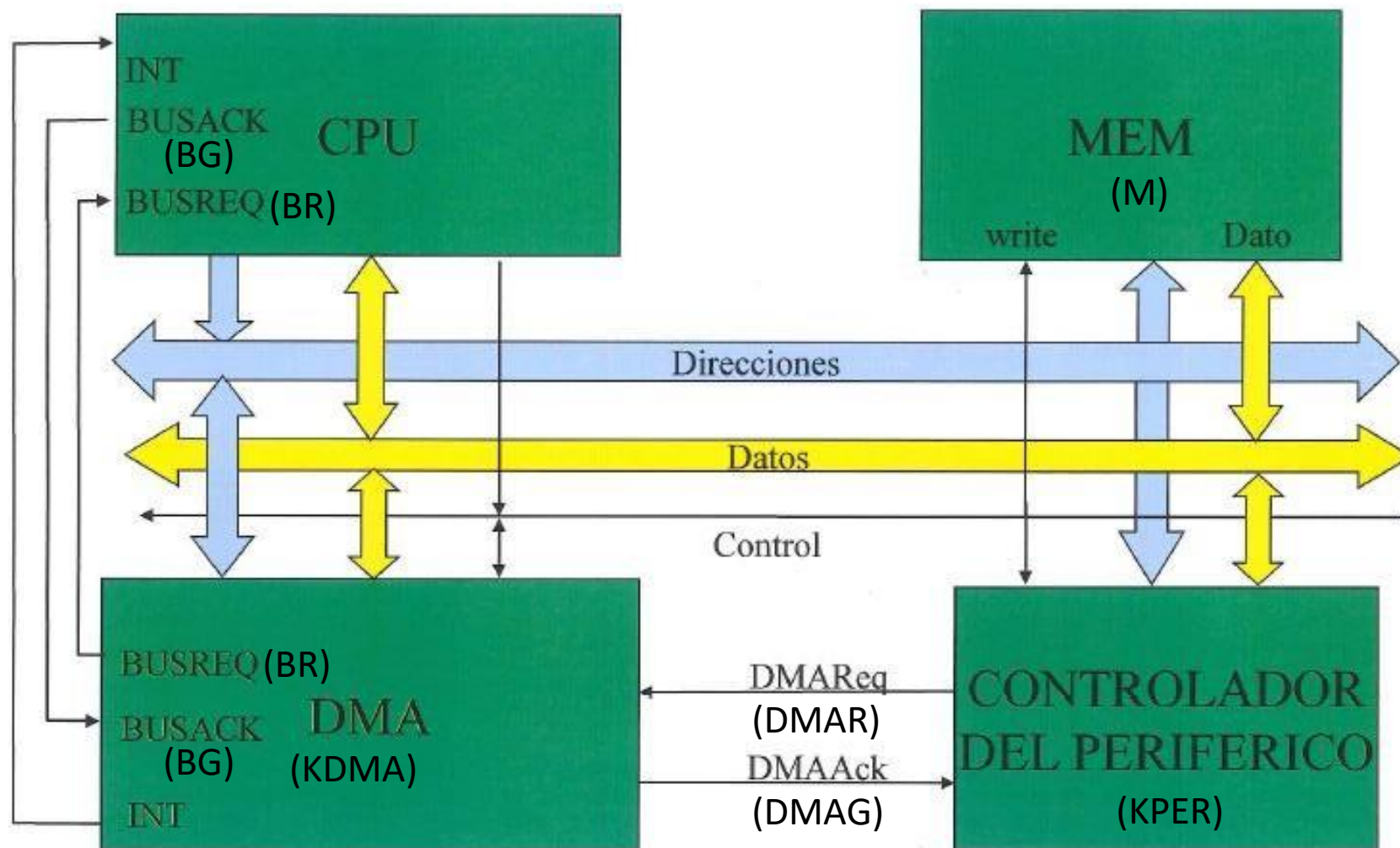
# Puntos de ruptura durante la ejecución



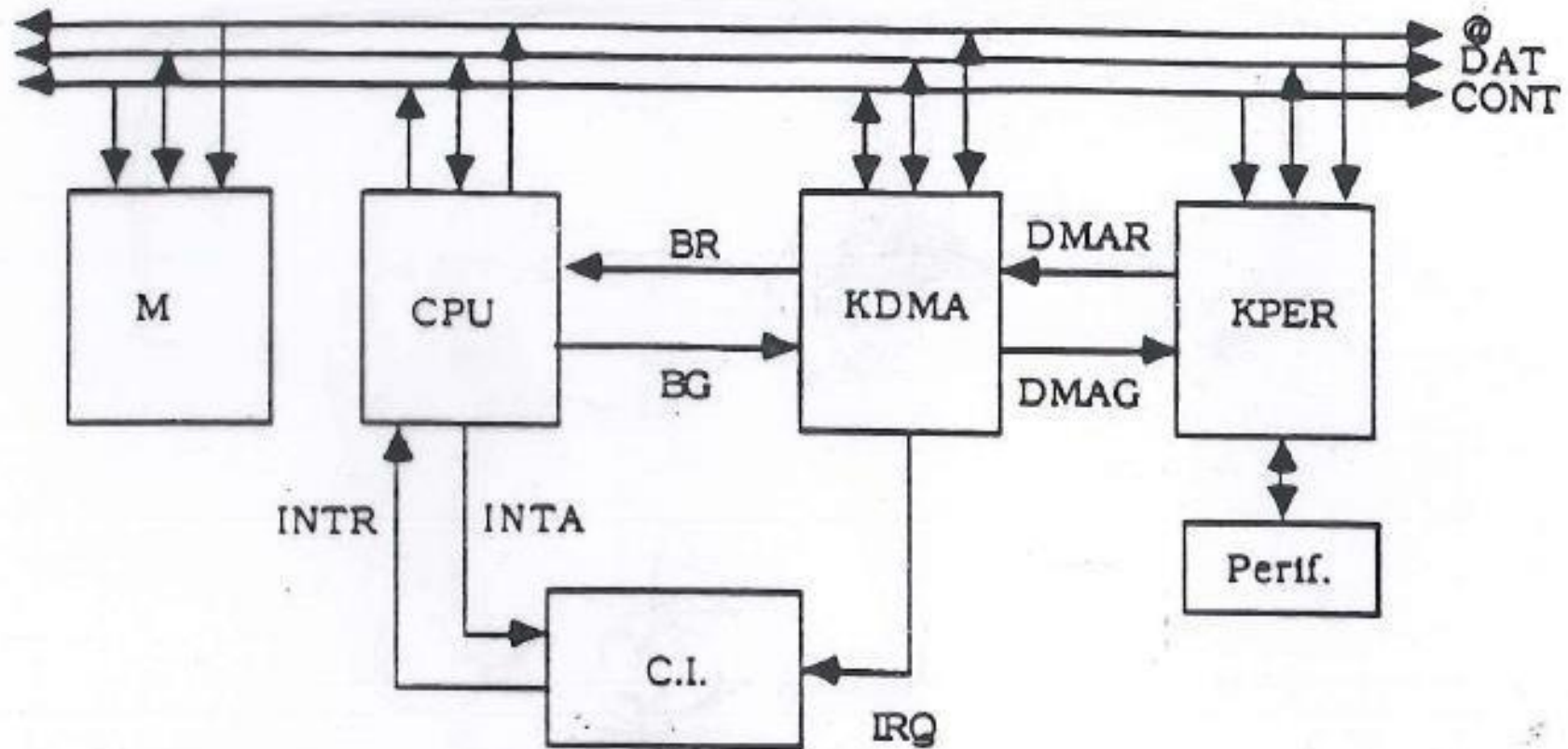
# DMA Breakpoints

Una petición de DMA puede ser atendida en cualquier ciclo de la ejecución de una instrucción (el DMA no altera los registros del procesador. **Nótese que no es una interrupción !**). Téngase en cuenta que el procesador no necesita usar el bus en todas las fases de la ejecución de una instrucción. Durante la decodificación y la ejecución de una instrucción, el bus está libre.

# La interface externa del DMA



# Conexión CPU – Controlador de DMA – Controlador de interrupciones



La señal DMAR es activada por el periférico para indicar al controlador que está listo para recibir o enviar un nuevo dato. Cuando esto ocurre, el KDMA activa la señal BR para indicar a la CPU que desea utilizar los buses. Aquí se produce la sincronización necesaria entre CPU y KDMA.

La CPU, en respuesta a BR, después de acabar su acceso a memoria (en caso de que estuviera realizando alguno) activa la señal BG para indicar al KDMA que puede tomar el control de los buses. En este momento, la CPU pone sus salidas en alta impedancia de forma que es como si se desconectara de los buses.

Cuando KDMA recibe BG coloca en el bus de direcciones la dirección adecuada, activa la señal R/W adecuadamente y activa DMAG para indicar al periférico que puede dejar el valor leído en el bus de datos (caso de lectura) o tomar el valor del bus de datos (en el caso de escritura).

Cuando el periférico completa la operación, baja las señales DMAR y en respuesta a ésta el KDMA baja BR para que la CPU se entere de que puede retomar el control de los buses.

Notar que la CPU no tiene por qué estar parada durante este proceso. Efectivamente, no puede acceder a memoria, pero puede realizar cualquier otro trabajo como decodificación, cálculo de dirección, operaciones aritméticas, etc.

# Modos de transferencia del DMA

## 1. Ciclo robado (Cycle Stealing)

- El DMA pide el uso del bus a la CPU y espera a que se lo conceda
- Transfiere UNA única palabra
- Devuelve el uso del bus a la CPU

Cuando el DMA toma el control del bus lo retiene durante un solo ciclo. Transmite una palabra y libera el bus.

El DMA «roba» ciclos al procesador.

## 2. Modo bloque o ráfaga (Burst)

- El DMA pide el uso del bus a la CPU
- Transfiere TODO un bloque de datos de forma continua
- Devuelve el uso del bus a la CPU

Cuando el DMA toma el control del bus no lo libera hasta haber transmitido el bloque de datos pedido.



## Modos de transferencia: uso del bus



Ciclo robado (Dispositivo lento\*)



Ciclo robado (Dispositivo rápido)



Transferencia de bloque o modo ráfaga (Dispositivo rápido)



CPU



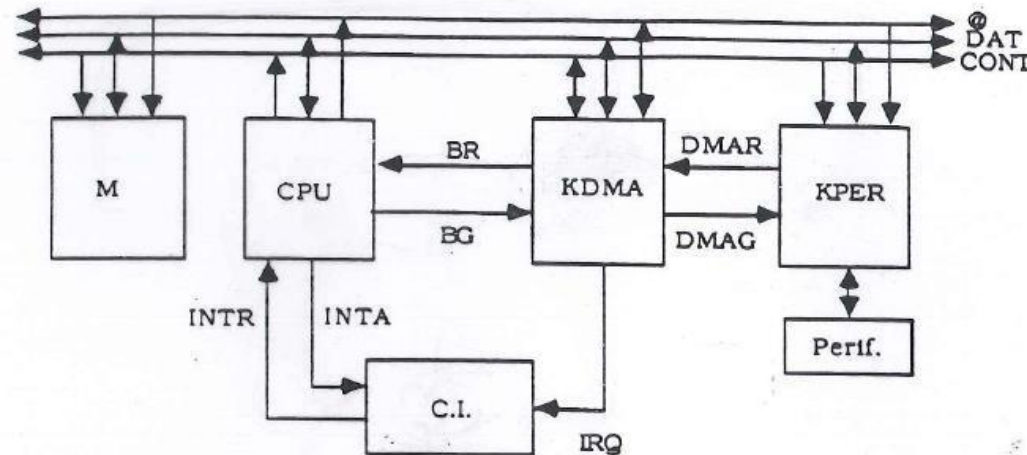
DMA



IDLE

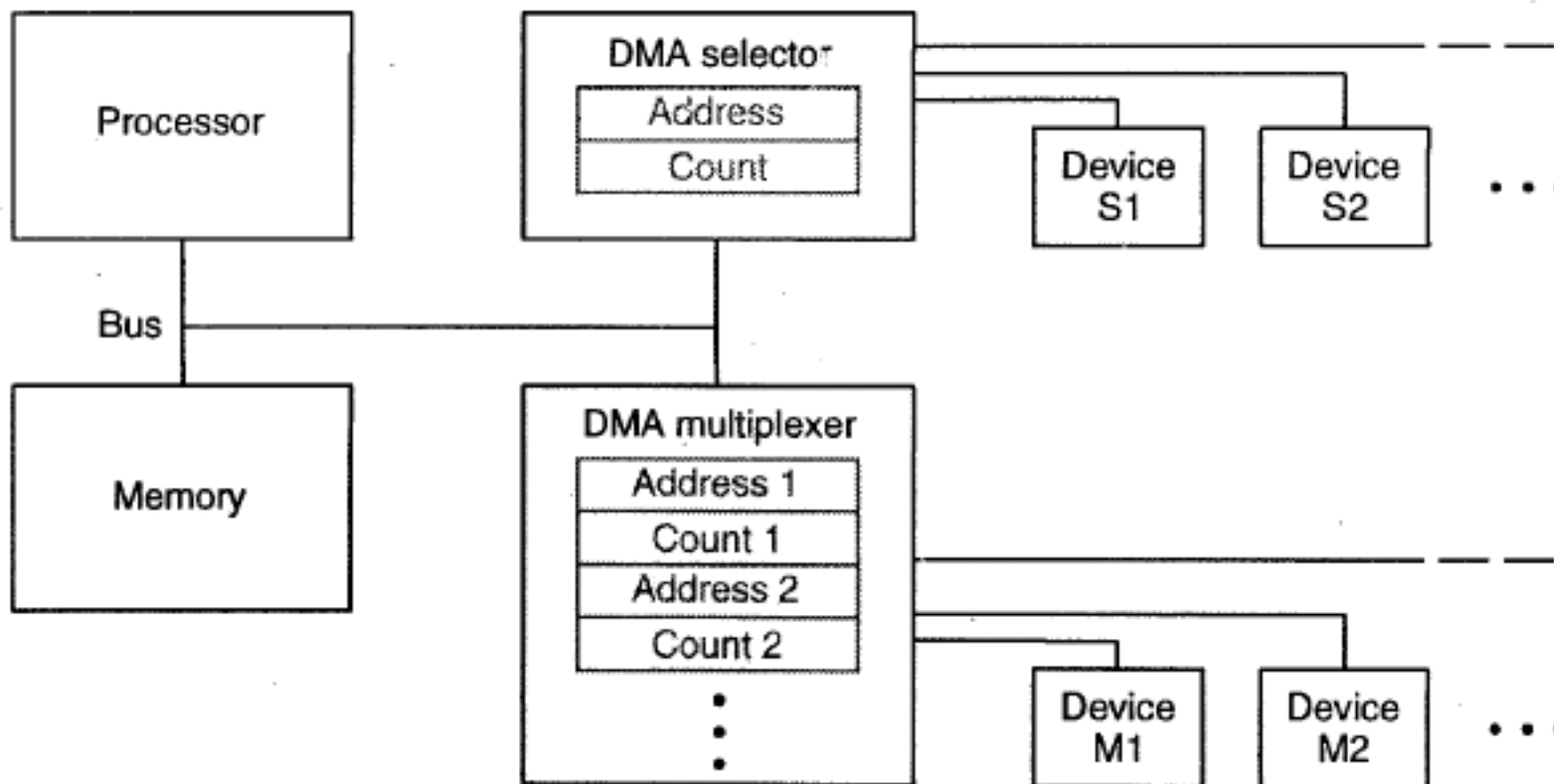
Cuando ya se finaliza la transferencia, ya sea por robo de ciclos o ráfaga, se debe producir una sincronización entre la CPU y el DMA por **interrupción**.

Cuando acaba la transferencia del bloque, el KDMA interrumpe a la CPU. La rutina de atención a la interrupción comprueba si toda ha ido bien, consultando el registro de estado del KDMA o del KPER. Es importante notar que la interrupción puede ser generada tanto por el KDMA como por el KPER.





# Canales DMA multiplexor y selector



# Evaluación de E/S por DMA

Programación del DMA:	30 instr. (240 ciclos)
Petición y Obtención del bus:	10 ciclos
Transferencia de un byte por el bus:	4 ciclos
Transferencia de $k$ bytes (modo ráfaga):	$k \cdot 4$ ciclos
Tiempo Respuesta Interrupción:	59 ciclos
Tiempo ejecución RSI al finalizar DMA:	10 instr. (80 ciclos)

## Tiempo CPU para TRANSFERENCIA de 1 Byte por DMA

$$240 + 10 + 4 + 59 + 80 = 393 \text{ Ciclos de reloj}$$

## Tiempo CPU para TRANSFERENCIA de N Bytes por DMA

$$240 + (10+4) \cdot N + 59 + 80 = 379 + 14 \cdot N \text{ ciclos}$$

## Tiempo CPU para N Bytes por DMA con ráfagas de $k$ bytes

$$240 + (10+4 \cdot k) \cdot N/k + 59 + 80 = 379 + 10 \cdot N/k + 4 \cdot N \text{ ciclos}$$

Es el tiempo máximo que se le roba a la CPU. En ocasiones, el robo del bus no frena del todo a la CPU, ya que no lo necesita inmediatamente para la tarea que está realizando

# Comparación Rendimiento

$N = 512 \text{ BYTES}$ ,  $M = 50$ ,  $1 \text{ CLK} = 100 \text{ ns}$  (10 Mhz)

	<u>CONS.</u>	<u>INTERRUP</u>	<u>DMA</u>	<u>DMA (<math>k=8</math>)</u>
<u>DISP. LENTO</u>				
$t_{\text{CPU}} =$	79,25 ms	13,15 ms	$(379+14 \cdot N)=0,7547\text{ms}$	$(379+10 \cdot N/8+4 \cdot N)=0,3067\text{ms}$
Ancho Banda:	6,46 KB/s	6,46 KB/s	6,46 KB/s	6,46 KB/s
% utiliz. CPU:	100%	16,59%	0,95%	0,39%

## DISP. más RÁPIDO gestionable

$t_{\text{CPU}}=$ ms	3,37 ms	13,15 ms	$(14 \cdot N) = 0,7168 \text{ ms}$	$(10 \cdot N/8 + 4 \cdot N) = 0,2688$ ms
Ancho Banda:	151,92 KB/s	38,91 KB/s	714,28 KB/s	1904,76 KB/s
% utiliz. CPU:	100%	100%	100%	100%

# Comparación Rendimiento

## DISPOSITIVO LENTO (6,46 KB/s)



## DISPOSITIVO más RÁPIDO POSIBLE

