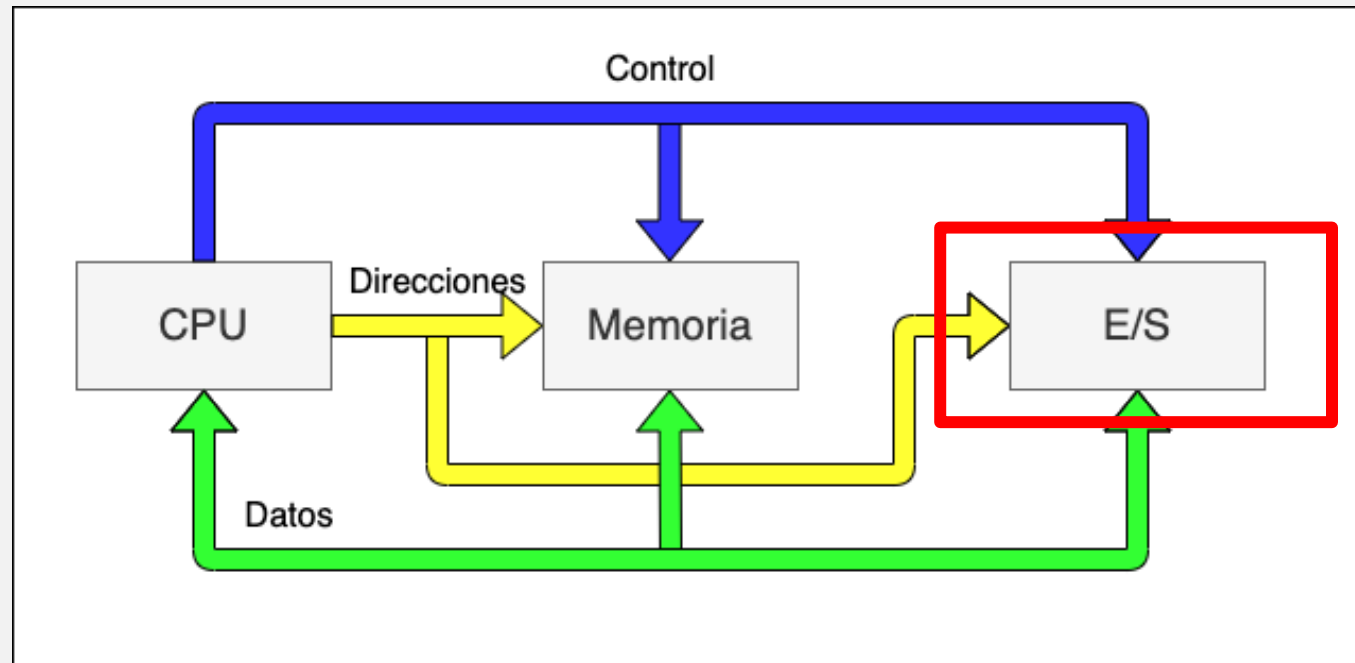


# TEMA 6: PROGRAMACIÓN DE ENTRADA/SALIDA

**Objetivo:** El alumno explicará los conceptos utilizados para la programación de entrada y salida.

# DISPOSITIVOS ENTRADA/SALIDA



# DISPOSITIVOS ENTRADA/SALIDA

- **Dispositivos**

## **Entrada**

- Teclado
- Mouse
- Touchpad
- Scanner
- Unidad de disco

## **Salida**

- Monitor
- Impresora
- Memorias

# DISPOSITIVOS ENTRADA/SALIDA

- **Puertos**
  - Ethernet
  - Video: VGA - HDMI - Display Port - DVI
  - Audio: IN - OUT - MIC
  - USB
  - IDE - SATA
  - PCI - PCIe
  - PS/2
  - DB9 (Serial port)
  - DB25 (Parallel port)

## OPERACIÓN DE DIRECCIONES DE E/S

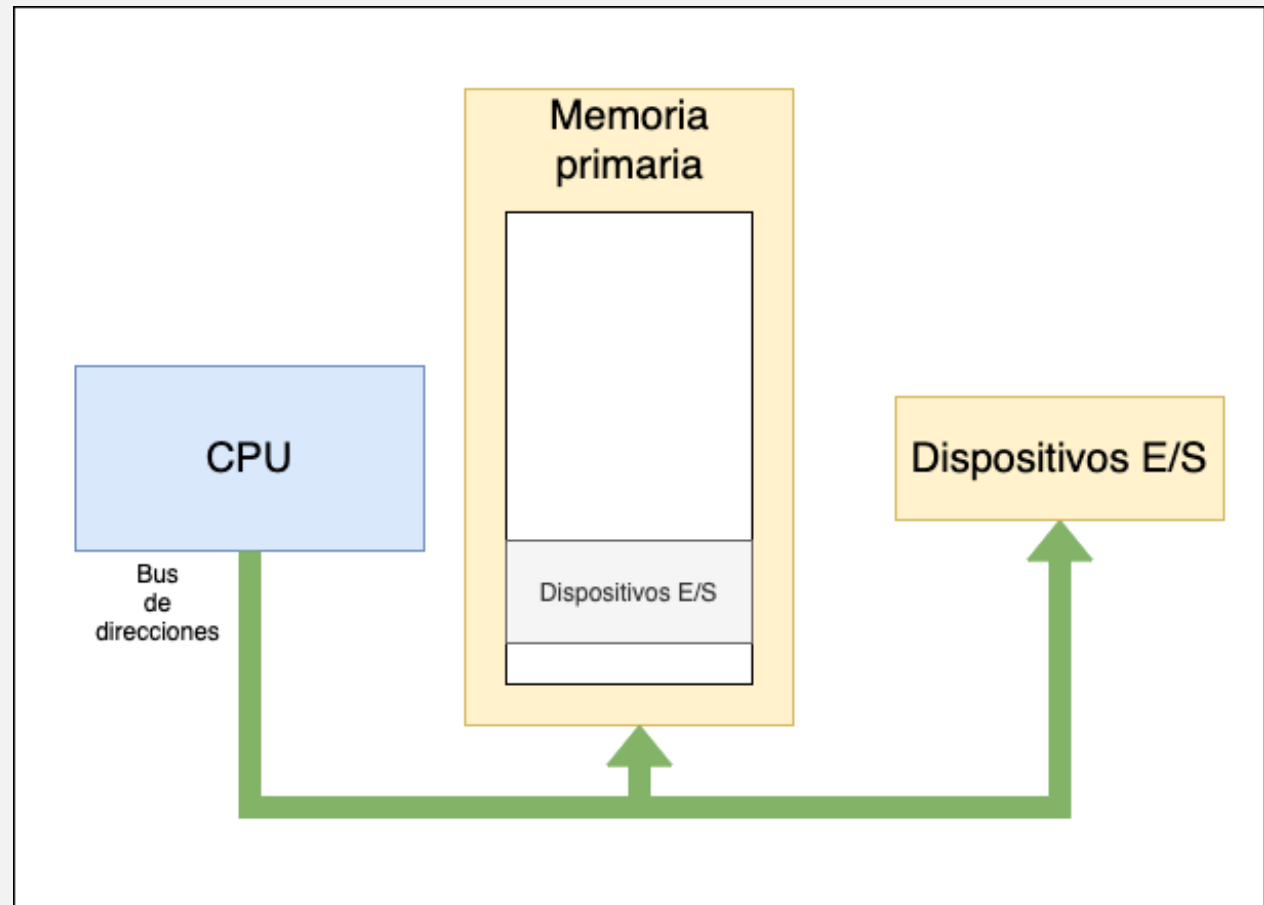
Se utilizan 2 formas de realizar operaciones entre periféricos y CPU:

- **E/S mapeado por memoria**
- **E/S mapeado por puerto**

# OPERACIÓN DE DIRECCIONES DE E/S

## E/S mapeado por memoria

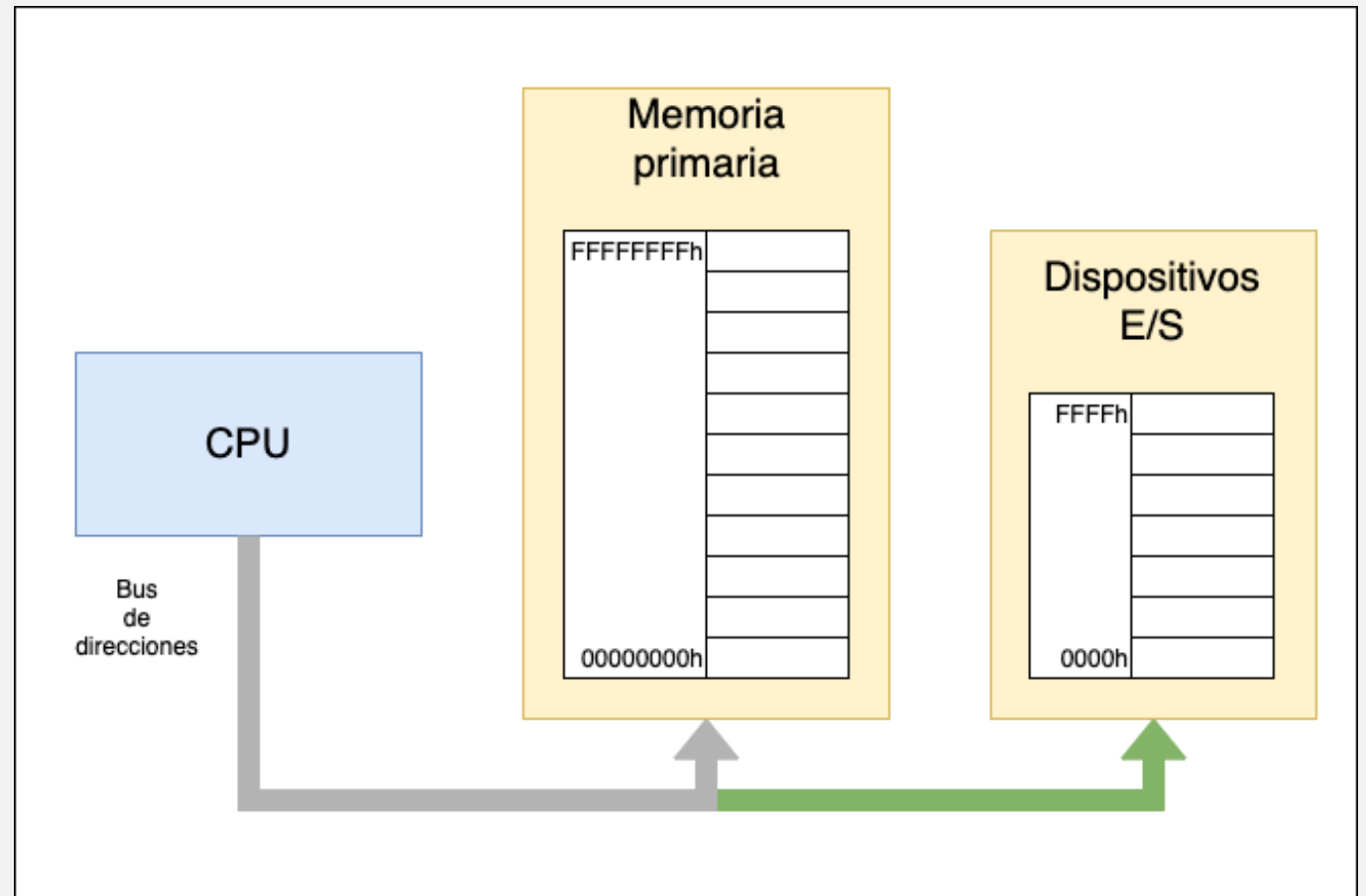
- Utiliza el mismo espacio para direcciones de memoria y periféricos. La memoria y los periféricos están mapeados a valores de dirección, así que una dirección accedida por el CPU puede hacer referencia a una localidad de memoria RAM o a la memoria de un dispositivo E/S.
- Las áreas de memoria utilizadas por los periféricos deben estar reservadas.



# OPERACIÓN DE DIRECCIONES DE E/S

## E/S mapeado por puerto

- Utiliza instrucciones especiales del CPU diseñadas para realizar operaciones de E/S. Estas instrucciones son IN y OUT.



# PUERTOS

- En una computadora, un puerto de Entrada/Salida (E/S) utiliza una dirección de memoria para transferir datos. A este procedimiento de comunicación con periféricos se le denomina **direccionamiento de E/S**.
- Una dirección de E/S hace referencia a un espacio de memoria separado en los periféricos, similar a periféricos mapeados por memoria que utilizan bloques de memoria.
- Los periféricos utilizan dos métodos: una dirección de E/S para pasar señales de control y dirección, y memoria para transferencia de datos.



# PUERTOS

- Existe un espacio de direcciones de 64 KB para direccionamiento de E/S, aunque comúnmente sólo se utiliza menos de 1 KB.
- Cada dispositivo que utiliza una dirección de E/S contiene pocos bytes de memoria colocados en un rango de direcciones predeterminados.
- Los espacios de E/S son números de pequeños bancos de memoria dispersos entre diferentes dispositivos.
- Si cada dispositivo está colocado en una dirección diferente, el CPU puede transmitir señales al dispositivo apropiado sin conflictos

## OPERACIÓN DE DIRECCIONES DE E/S

- Si un programa necesita enviar un byte a algún puerto, emite una instrucción OUT al CPU con la dirección del puerto. El CPU notifica al bus de control activar el espacio de E/S, y el bus de direcciones envía la señal de la localidad apropiada del dispositivo. Entonces el CPU envía los datos sobre el bus de datos a ese puerto.

### **Puerto serie**

- |        |      |
|--------|------|
| - COM1 | 3F8h |
| - COM2 | 2F8h |
| - COM3 | 2E8h |
| - COM4 | 2E0h |

### **Puerto paralelo**

- |        |      |
|--------|------|
| - LPT1 | 378h |
| - LPT2 | 278h |
| - LPT3 | 3BCh |

# OPERACIÓN DE DIRECCIONES DE E/S

## **Instrucciones IN y OUT**

- IN – Lectura de dispositivos E/S
- OUT – Escritura de dispositivos E/S

# INSTRUCCIÓN IN

- **Función**
- Lee en AL o AX un dato de 8 o 16 bits, respectivamente, de la dirección del puerto especificado.
- **Sintaxis**

in   destino,   origen  
      {AL,AX}   {imm8/DX}

Registro  
destino.

Dirección del puerto:  
Puede ser un valor inmediato (constante)  
de 8 bits o el registro DX de 16 bits.

## Ejemplos:

in al, 60h   ;lee de teclado y guarda dato de  
              ;8 bits en AL

mov dx, 03F8h ;dirección de puerto serie en DX  
in ax, dx   ;lee de puerto direccionado por DX  
              ;y guarda dato de 16 bits en AX

## Banderas afectadas:

Ninguna

# INSTRUCCIÓN OUT

- **Función**
- Escribe un dato de AL (8 bits) o AX (16 bits) en la dirección del puerto especificado en el destino.
- **Sintaxis**

out destino, origen  
{imm8/DX} {AL,AX}

Registro  
origen.

Dirección del puerto:

Puede ser un valor inmediato (constante) de 8 bits o el registro DX de 16 bits.

## Ejemplo:

```
mov dx,03F8h ;dirección de puerto serie en DX
out dx,ax     ;escribe el contenido de AX en el puerto
              ;direccionado por DX
```

## Banderas afectadas:

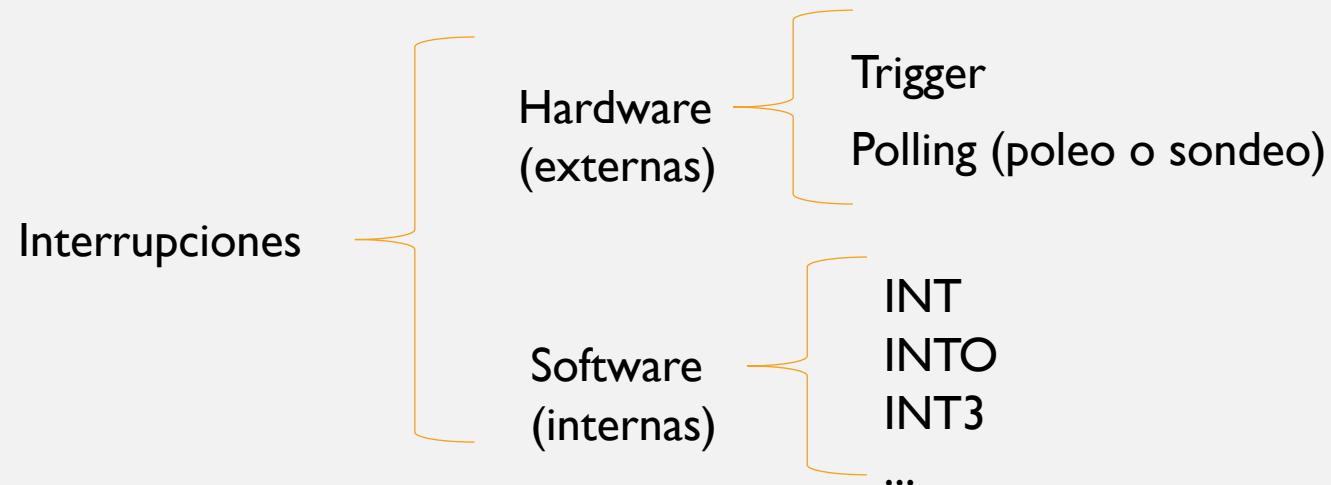
Ninguna

# INTERRUPCIONES

- Una interrupción **es una operación que suspende la ejecución de un programa** de manera que **el sistema pueda realizar una acción especial**.
- Cuando se presenta una interrupción, el sistema suspende la ejecución de un programa y atiende la interrupción mediante una **rutina o procedimiento de interrupción**. Esta rutina **se ejecuta y**, por lo regular, **regresa el control del procesamiento al programa** que fue interrumpido, el cual continúa su ejecución.
- El BIOS maneja las interrupciones 00h – 1Fh, y DOS 20h – 3Fh.

# INTERRUPCIONES

- **Interrupciones externas:** son provocadas por dispositivos externos al microprocesador.
- **Interrupciones internas:** ocurren como resultado de la ejecución de una instrucción de interrupción INT, o una operación de división que cause desbordamiento, ejecución en modo de paso a paso (debugger) o una petición para una operación externa como E/S de disco.



# VECTORES DE INTERRUPCIÓN

- Los vectores de interrupción y la tabla de vectores son elementos imprescindibles para comprender interrupciones.
- La tabla de vectores de interrupción se encuentra en los primeros 1024 bytes de memoria, en las direcciones 00000000h – 000003FFh.
- Esta tabla contiene 256 vectores de interrupción, de 4 bytes distintos.
- Un vector de interrupción contiene la dirección (segmento y desplazamiento) del procedimiento de servicio de interrupciones.

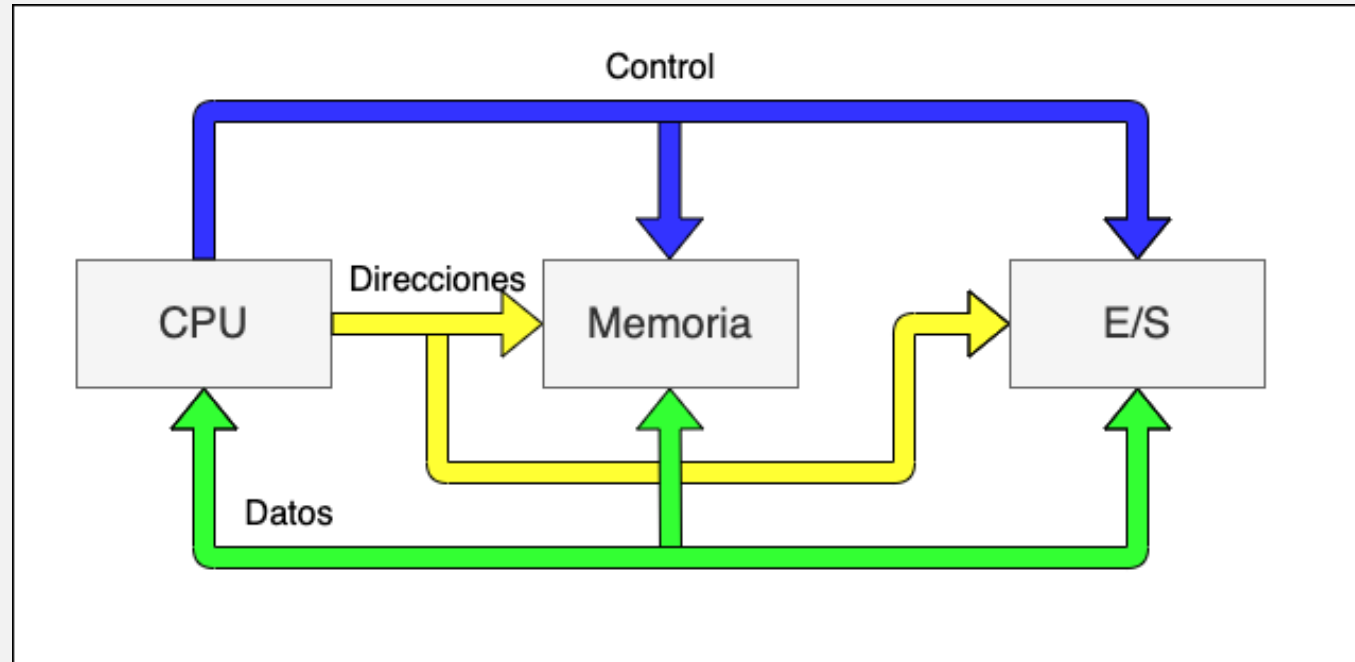


## DMA – ACCESO DIRECTO A MEMORIA

- Es una técnica que permite que un dispositivo de E/S tenga acceso a la memoria principal del sistema mientras el microprocesador está temporalmente deshabilitado.
- Esto permite transferencia directa de datos entre memoria y el dispositivo E/S, aunque a una velocidad limitada por la memoria o el dispositivo de E/S.

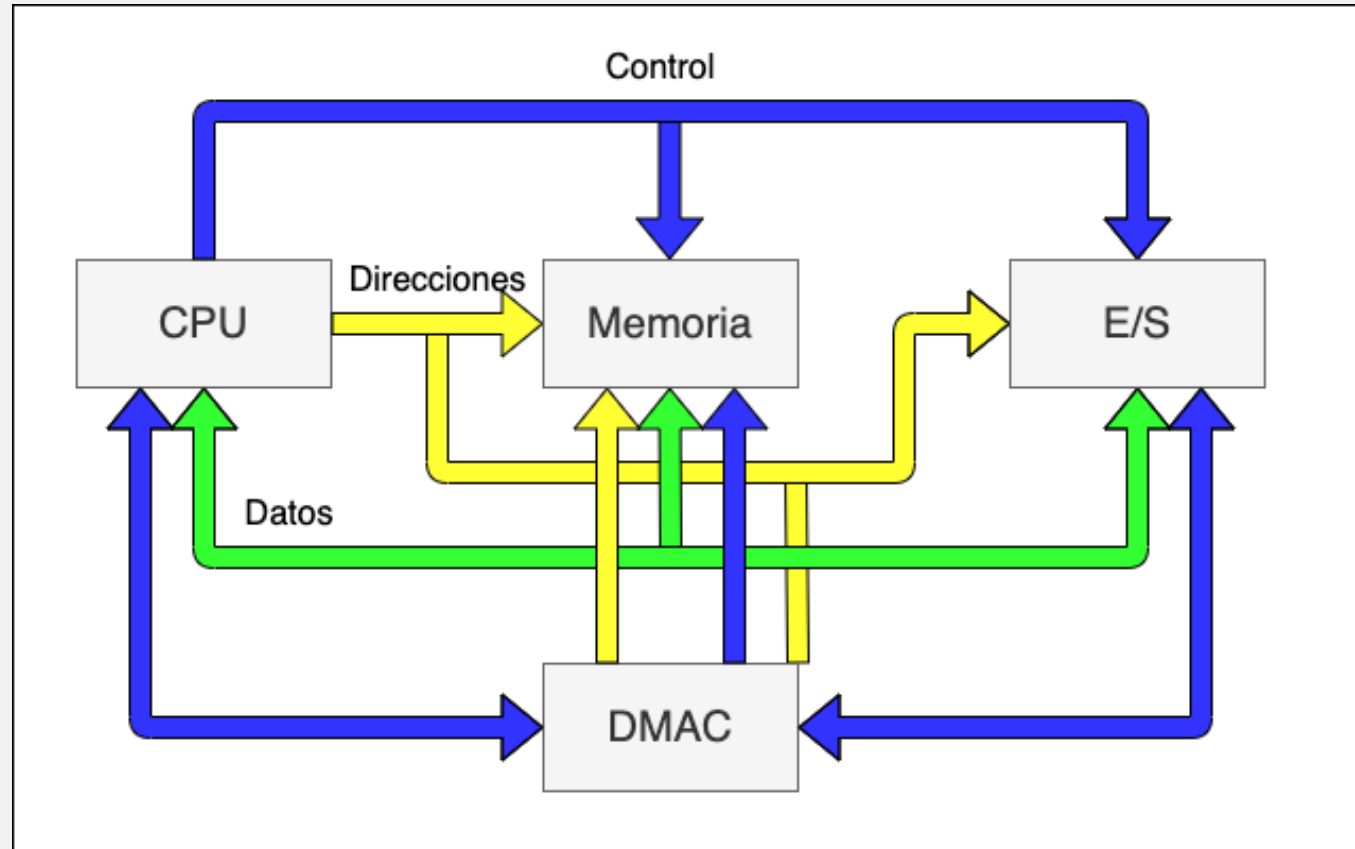
# DMA – ACCESO DIRECTO A MEMORIA

Lo que sabíamos hasta el momento...



# DMA – ACCESO DIRECTO A MEMORIA

Se introduce un Controlador de DMA

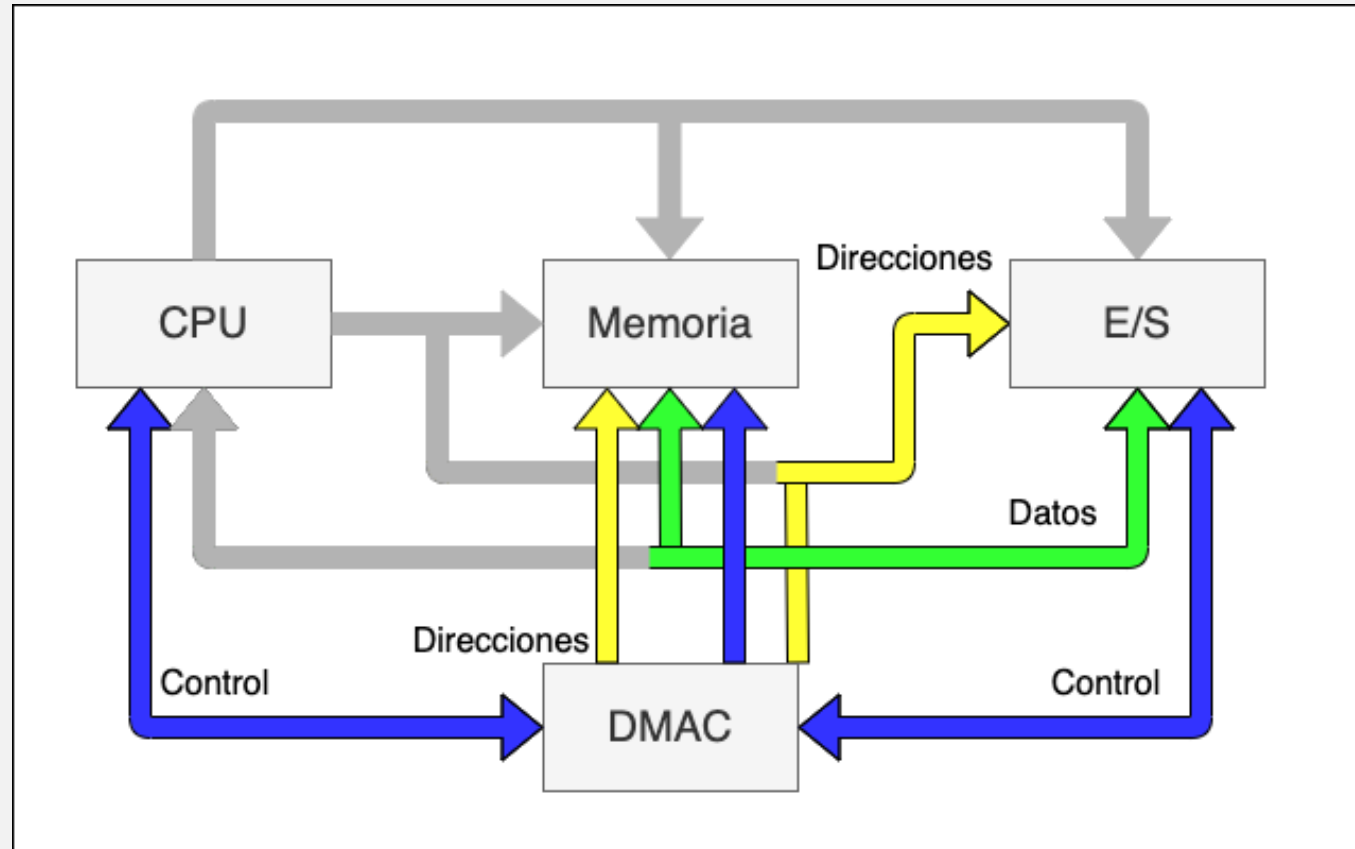


## DMA – ACCESO DIRECTO A MEMORIA

- El controlador de DMA proporciona señales e información de direcciones a la memoria y a los dispositivos E/S durante la transferencia DMA.
- El DMAC funciona como un microprocesador de propósito específico.

# DMA – ACCESO DIRECTO A MEMORIA

Se deshabilitan los buses desde el CPU y el control lo tiene el DMAC

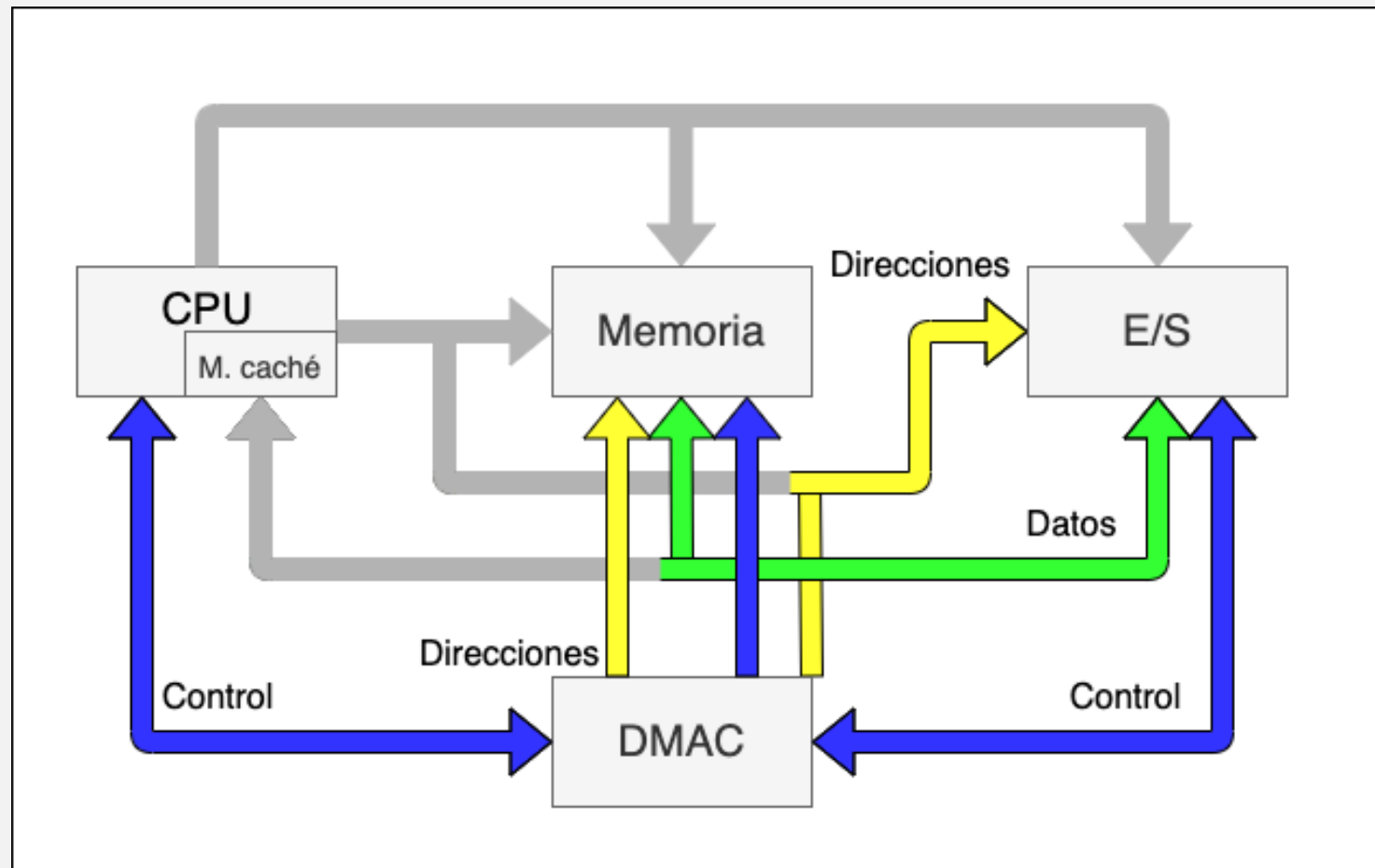


# DMA – ACCESO DIRECTO A MEMORIA

- **Problemas de memoria caché**

Dado que la memoria caché se encuentra en el CPU, se debe tener especial cuidado al manejar cambios de memoria usando DMA. Si el CPU accede a una localidad de memoria afectada por DMA, es necesario que se refresque la caché después de realizar DMA.

## DMA – ACCESO DIRECTO A MEMORIA PROBLEMAS EN MEMORIA CACHÉ



## DMA – ACCESO DIRECTO A MEMORIA PROBLEMAS EN MEMORIA CACHÉ

- La memoria caché almacena una localidad de memoria y su contenido para un acceso más rápido.
- Si una localidad de memoria del sistema se modificó por DMA, el CPU no está enterado de ese cambio.
- Es posible que la memoria caché contenga información obsoleta.
- Se debe actualizar la información de las localidades de memoria de la caché, este proceso puede ser tardado.