Universidad Nacional de Entre Ríos

Tecnicatura universitaria en desarrollo web

Interrupciones

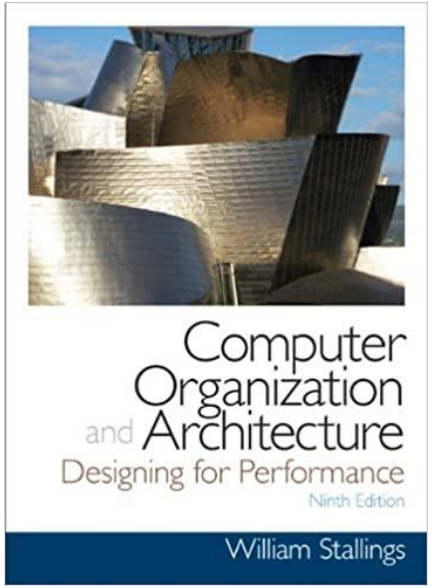
Semana 7 – Arquitectura de computadoras

Docente: José Ruiz {jose.ruiz@uner.edu.ar}

UNER virtual

Esta presentación esta basada en el libro de:

- William Stallings, Computer Organization and Architecture, 9th Edition, 2017.



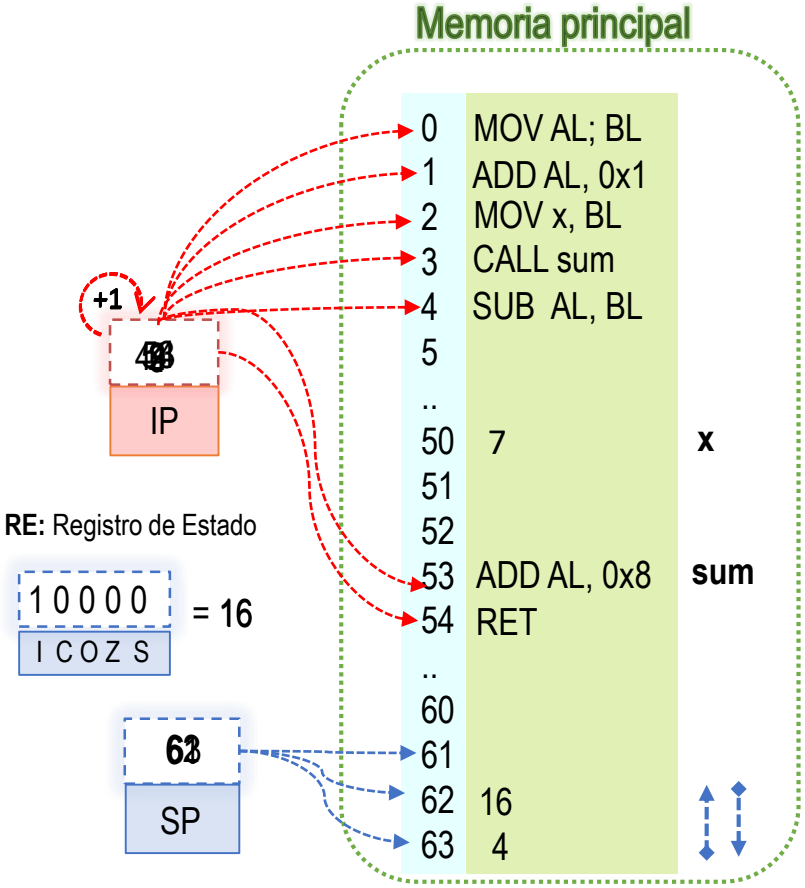
Archivos presentación y ejemplos se alojan en:

https://github.com/ruiz-jose/tudw-arq.git

Interrupciones

- Instrucción CALL/RET
- Arranque del sistema
- Vector de interrupciones
- Tipos de interrupciones:
  - Interrupción Software: Instrucción INT/IRET
  - Interrupción Hardware
  - Excepción

Instrucción CALL/RET



CALL dirección → llamada a una rutina

RE: Registro de Estado

1 0 0 0 0 = 16

I C O Z S

CALL sum

a) PUSH *IP*: {[SP] ← IP ; SP ← SP – 1}
b) PUSH *RE*: {[SP] ← RE ; SP ← SP – 1}
c) IP ← dirección

RET → retorno al programa llamador.

RET

a) POP *RE*: {SP ← SP + 1 ; RE ← [SP]}
b) POP *IP*: {SP ← SP + 1 ; IP ← [SP]}

Arranque del sistema

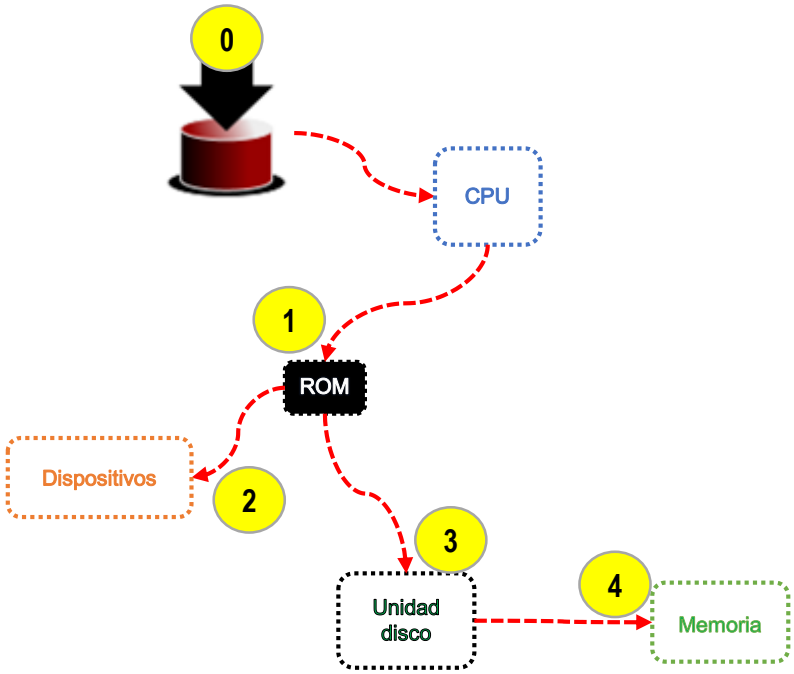
0. Presionamos el botón de encendido, llega energía y arranca el sistema

1. El CPU comienza a ejecutar el **BIOS (Basic Input Output System),** que se encuentra en la memoria **ROM** de la placa madre

2. La BIOS se encargar de correr una serie de diagnósticos llamados **POST (Power On Self Test)**

3. Busca la Unidad «booteable», y copia el «bootloader» a la memoria y comienza ejecutarlo

4. El «bootloader» carga el sistema operativo en memoria y sede el control al mismo.



Vector de interrupciones

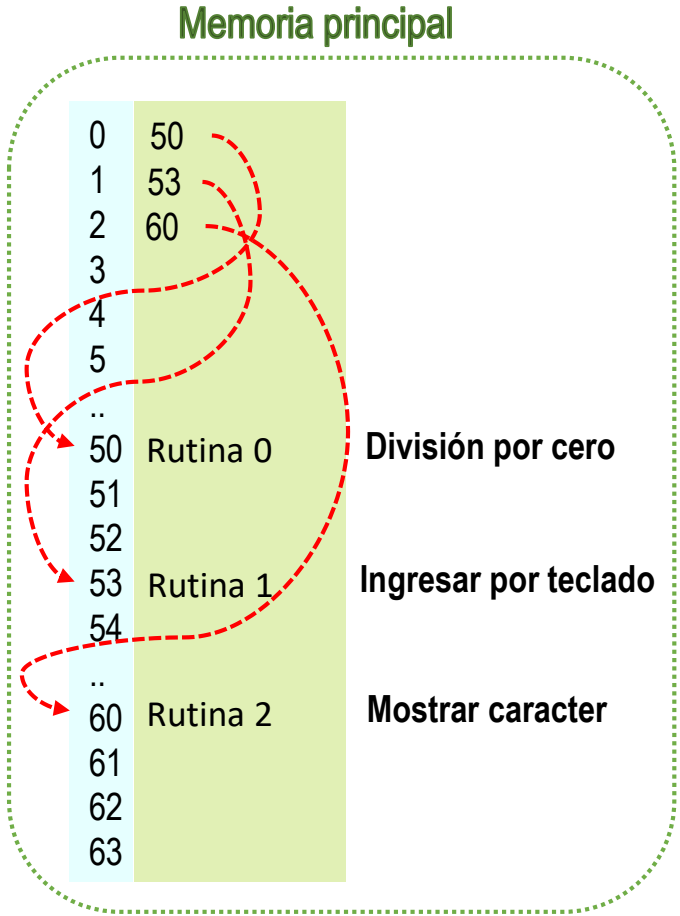
Cuando la computadora se enciende, el BIOS y el SO establecen una tabla de rutinas de tratamiento de interrupciones en las primeras localidades de memoria.

- **Tabla de vectores de Interrupción**

  Refleja las direcciones de las distintas rutinas de tratamiento de interrupciones.

¿Quién invoca la rutina de tratamiento de interrupción?

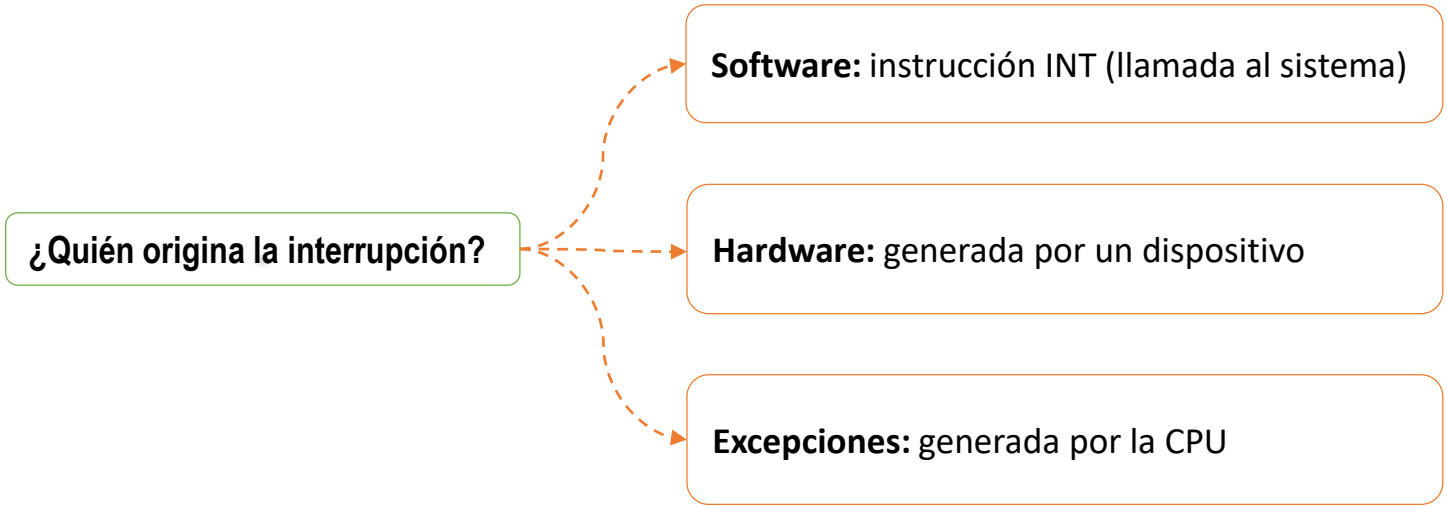
- Interrupción Software: Instrucción INT/IRET
- Interrupción Hardware
- Excepción



Tipos de interrupción

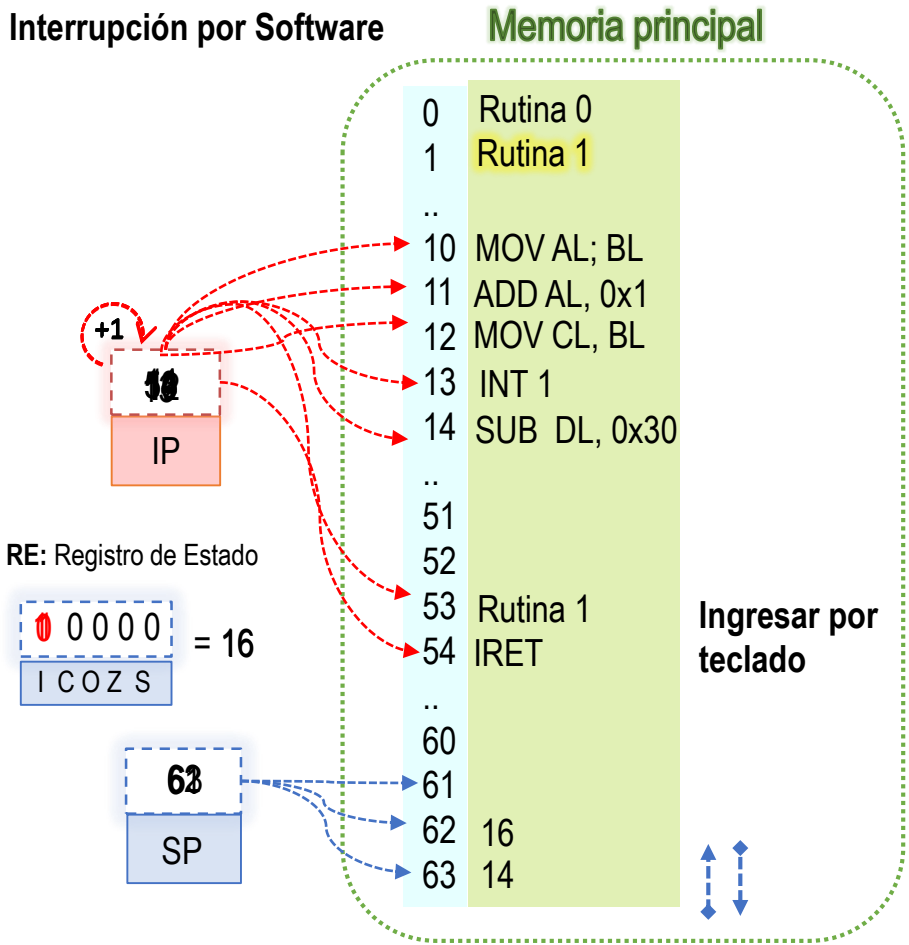
Tipos de interrupciones

Según su origen se clasifican en:



Interrupción por Software Instrucción INT/IRET

- **Interrupción por Software**



¿Cómo se invoca a estas rutinas?

INT Nro_INT → llamada a una rutina de interrupción.

INT 1 → Llama a la rutina 1 del vector de interrupciones devuelve en DL el código ASCII del dato ingresado

Nuevo flag I:

que indica si el procesador puede ser interrumpido o no.

RE: Registro de Estado

1 0000 = 16

I C O Z S

INT 1 →

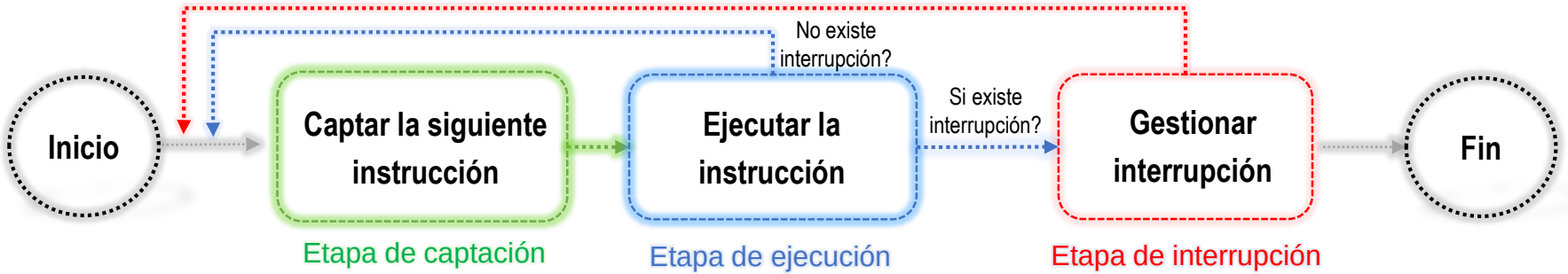
a) PUSH *IP*: {[SP] ← IP ; SP ← SP – 1}
b) PUSH *RE*: {[SP] ← RE ; SP ← SP – 1}
c) I ← 0
d) IP ← [1]

IRET → retorno al programa llamador.

IRET →

a) POP *RE*: {SP ← SP + 1 ; RE ← [SP]}
b) POP *IP*: {SP ← SP + 1 ; IP ← [SP]}
c) I ← 1

Ciclo de instrucción – Etapas de captación, ejecución e interrupción

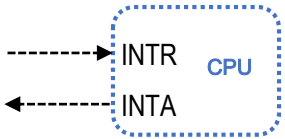


Interrupciones por hardware

Interrupciones por hardware

El procesador 8086
Incluye dos nuevas señales:

- Entrada: INTR (Solicitud interrupción)
- Salida: INTA (Interrupción reconocida)



¿Qué pasa cuando un dispositivo interrumpe al CPU?

Cuando se produce una interrupción si el dispositivo de E/S activa la señal de interrupción (INTR) y el flag I vale 1, termina de ejecutar la instrucción en curso y realiza atómicamente los siguientes pasos:

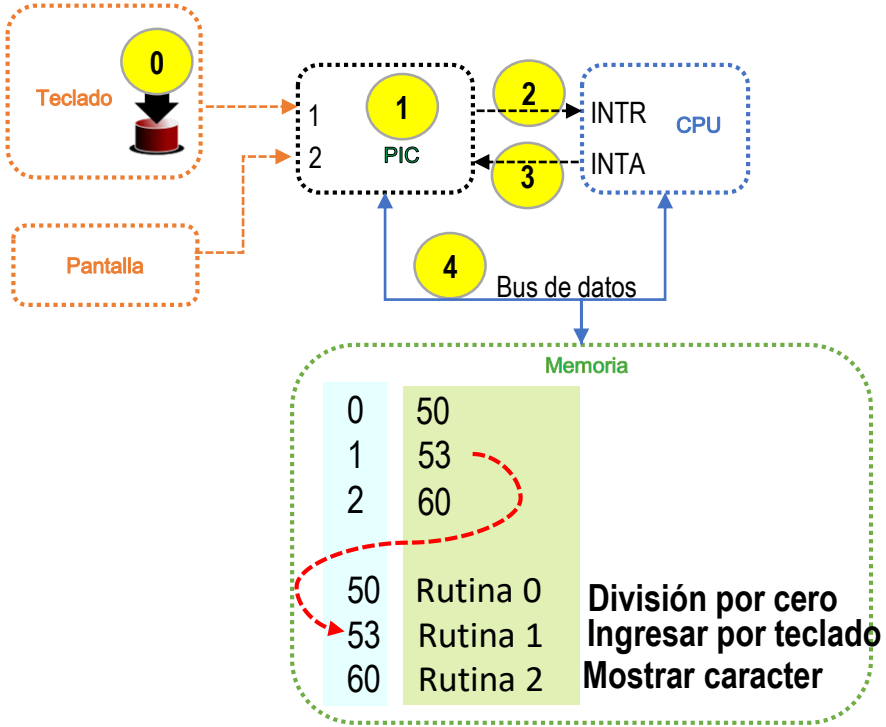
a) PUSH *IP*: {[SP] ← IP ; SP ← SP – 1}
b) PUSH *RE*: {[SP] ← RE ; SP ← SP – 1}
c) I ← 0 : para evitar que el procesador vuelva a interrumpirse
d) INTA: Activa la señal INTA para indicarle al dispositivo que atenderá su pedido.
e) IP ← [dirección rutina servicio de interrupción]

Nota: el servicio de interrupción es responsable de preservar el valor actual de los registros y restaurar el registro de estado y PC, mediante la instrucción de retorno de interrupción IRET

Interrupciones por hardware

¿Cómo identificar el dispositivo que genero la Interrupción?

0. Presionamos una tecla
1. El PIC (Programmable Interrupt Controller) controla que no se este atendiendo otra interrupción y solicita mediante la señal INTR al CPU ser atendido.
2. CPU recibe la solicitud
3. CPU verifica si va a atender la interrupción, en caso afirmativo avisa mediante la señal INTA que va atenderla.
4. El PIC envía por el bus de datos el Nro 1 correspondiente a la rutina del vector de interrupciones que tiene que invocar el CPU.

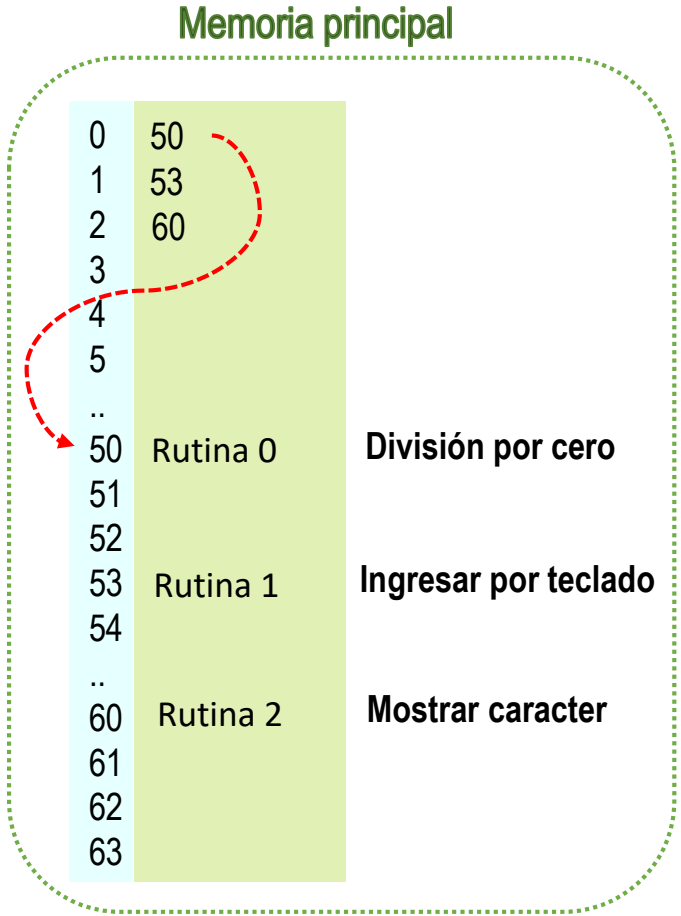


Excepciones (división por cero)

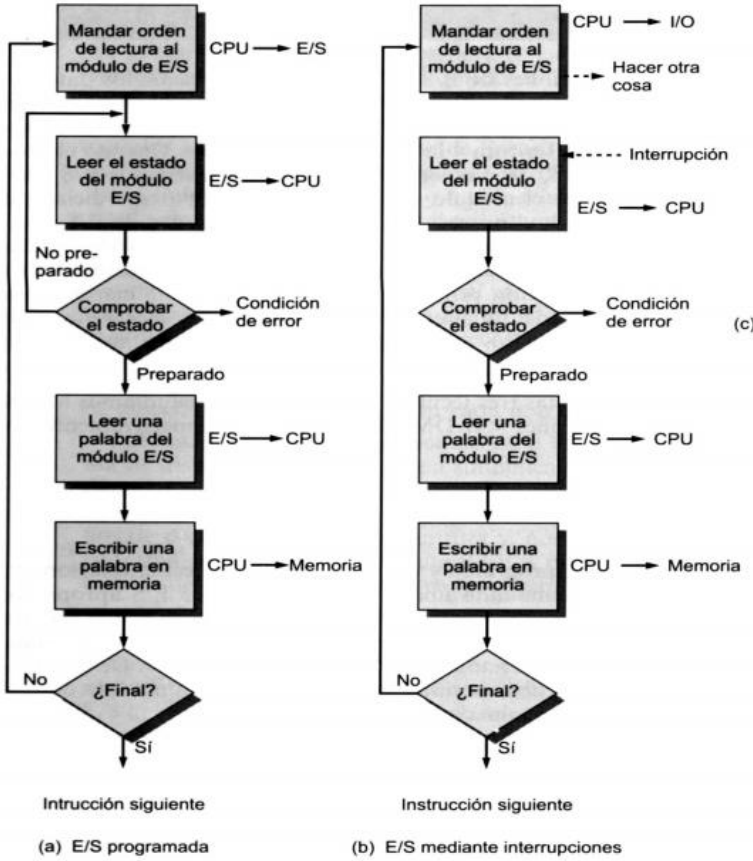
DIV (división por cero)

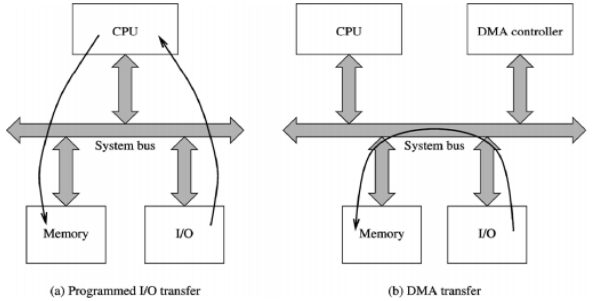
```
MOV AX, 203   ; AX = 00CBh
MOV BL, 0
DIV BL
RET
```

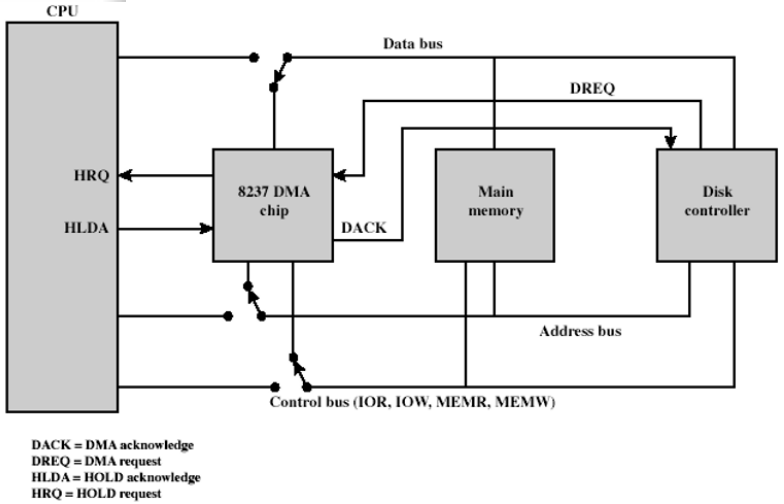
Al producirse una división por cero el CPU invoca a la rutina 0 del vector de interrupciones



Técnicas de transferencia de información entre Modulo de E/S-CPU







Técnicas de transferencia de información entre Modulo de E/S-CPU

CPU a 200 MHz (tiempo ciclo reloj = 5 ns; Ciclos por instrucción CPI = 2 , en promedio)
• Una instrucción tarda en promedio 2 x 5 ns = 10 ns => la computadora puede ejecutar ~100 Mips

La impresora imprime 1000 bytes/s por segundos
• Si queremos imprimir un archivo de 10.000 bytes

E/S programada
La CPU entra en un bucle y envía un nuevo byte cada vez que la impresora está preparada para recibirlo
• La impresora tarda 10 seg en imprimir 10.000 bytes
• **La CPU está ocupada con operaciones de E/S durante 10 seg.**

E/S con interrupciones
La impresora genera una interrupción cada vez que está preparada para recibir un nuevo byte.
• Si la rutina de tratamiento de interrupción tiene 10 instrucciones (salvar contexto, comprobar estado, transferir byte, restaurar contexto, retornar al programa anterior)
• Para transferir 10.000 bytes tenemos que ejecutar 10.000 veces la rutina de tratamiento de interrupción
es como ejecutar 100.000 instrucciones (10.000 * 10) → para atender al periférico la CPU tarda 0,001 seg
• **La CPU está ocupada con la operación de E/S durante 0,001 seg.**

Conclusión
• Si el dispositivo es lento la E/S por interrupciones reduce en 10.000 veces el tiempo que la CPU está ocupada gestionando la impresora.

Preguntas?