

# Arquitectura de Computadoras

FECHA

Interrupciones: Mecanismo mediante el cual se puede interrumpir el procesamiento normal de la CPU.

- ↳ En casi todos los casos implica: transferir el control a un gestor que:
- Salve el estado del procesador
  - Corrija o responda a la causa que ocasionó la interrupción
  - Restablece el estado original del procesador
  - Retorne la ejecución normal del programa interrumpido.

## Jerarquía de Int.

No enmascarables: No pueden ignorarse (eventos peligrosos)

Enmascarables: Pueden ser ignoradas mediante instrucciones que inhiban la solicitud.

## Interrupciones por Hardware

Son generadas por dispositivos de E/S. El sistema de cómputo tiene que manejar estos eventos asincrónicos.

Traps/Excepciones: Interrupciones por Hard creadas por el procesador ante ciertos eventos (Fallo de Prog, de Hard, etc)

## Interrupciones de Software

Muchos procesadores tienen instrucciones explícitas que afectan

al procesador de la misma manera que las interrupciones por hardware.

• Generalmente ~~se~~ se usan para hacer llamadas a funciones del S.O.

• No requieren conocer la dir. de la rutina en tiempo de ejecución.

## Ciclo de Interrupciones

• Se comprueba si se ha solicitado alguna interrupción.

(Indicada por Flags de petición de int.)

• Si no hay señal se copia la siguiente instrucción.

• Si hay pedido de interrupción pendiente:

1. Se suspende la ejecución del programa en curso.

2. Guarda su contexto (prox instrucción y estado)

3. Carga el PC con la dirección de comienzo de una rutina de gestión de int.; Se inhiben otras Int.!

4. Finalizada la rutina de gestión, el procesador retoma la ejecución del programa del usuario en el punto de int.

Las instrucciones se manejan en un orden secuencial estricto. Si se hubiera generado una interrupción mientras se daba a otra, se mantendrá en espera y se examinará luego una vez que se hayan habilitado nuevamente.

## Definir Prioridades

Una interrupción de prioridad más alta puede interrumpir a un gestor de interrupción de prioridad menor (En la cola de Int., no en ejecución)

Cuando se ha gestionado la interrupción de prioridad más alta, el procesador vuelve a las interrupciones previas

## Interrupciones Multinivel

Cada dispositivo está conectado físicamente al CPU

Sencillo pero caro

## Línea Única de Interrupción

Una sola entrada física de pedido de Int. a la que se conectan todos los dispositivos

Se debe preguntar a cada dispositivo si ha producido el pedido de interrupción (Polling)

## Interrupciones Vectorizadas

El dispositivo que quiere interrumpir además de la señal de pedido de interrupción debe colocar en el bus de datos un identificador (vector)

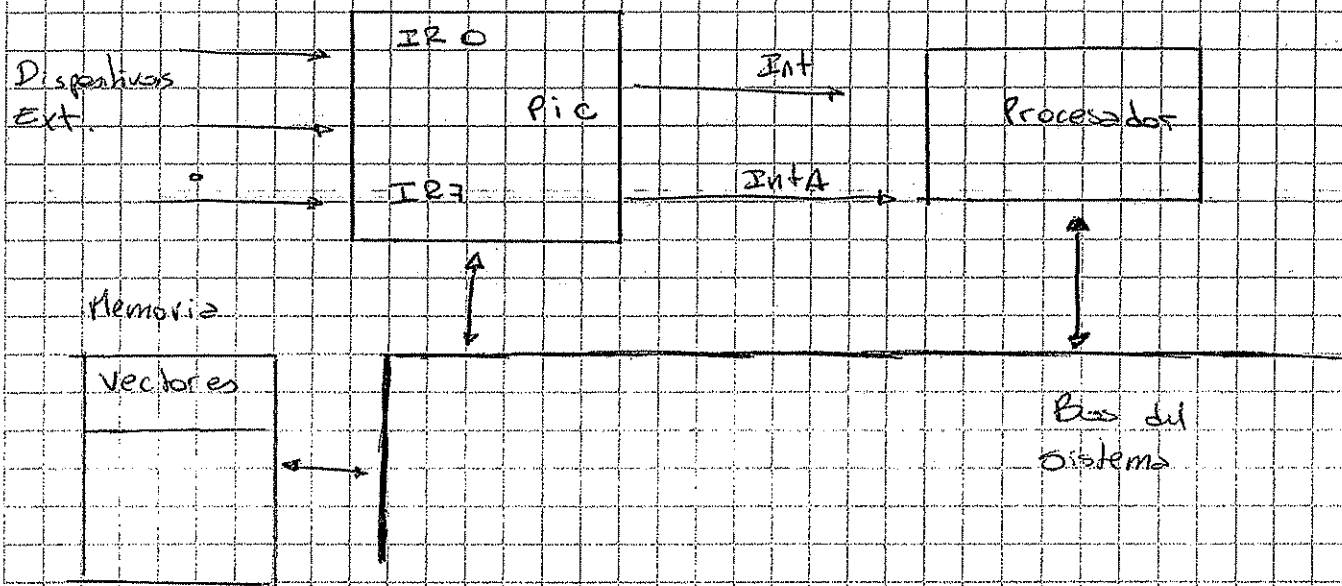
• lo coloca el periférico

o

• lo coloca el controlador de Int.



## Conexionado



## Tabla de Vectores de Interrupción

Es el nexo entre tipo de interrupción (0. 255) y el procedimiento designado para atenderla.

Cada entrada es un doble palabra (4 bytes).

### Vectores preasignados

- Tipo 0, finaliza ejecución del programa.
- Tipo 3, depuración.
- Tipo 6, lectura de entrada std. (Bx)
- Tipo 7, escritura de salida std. (Bx y AL)

## Registros Internos del PIC

- EOI: Para fin de Int escribir 20H.
- IMR: máscara de Int, enmascara con 1.
- IRR: Petición de Int, indica con bit 1.
- ISR: Int. en servicio, indica con bit 1.
- Int0 .. Int7: C/U con su vector.

— Están situados a partir de 20H y son accedidos con operaciones lectura y escritura en el espacio de E/S (in, out)

### Interrupciones por Hard Assignada

- Int0 = F10
- Int1 = Timer
- Int2 = Handshake
- Int3 = DMA

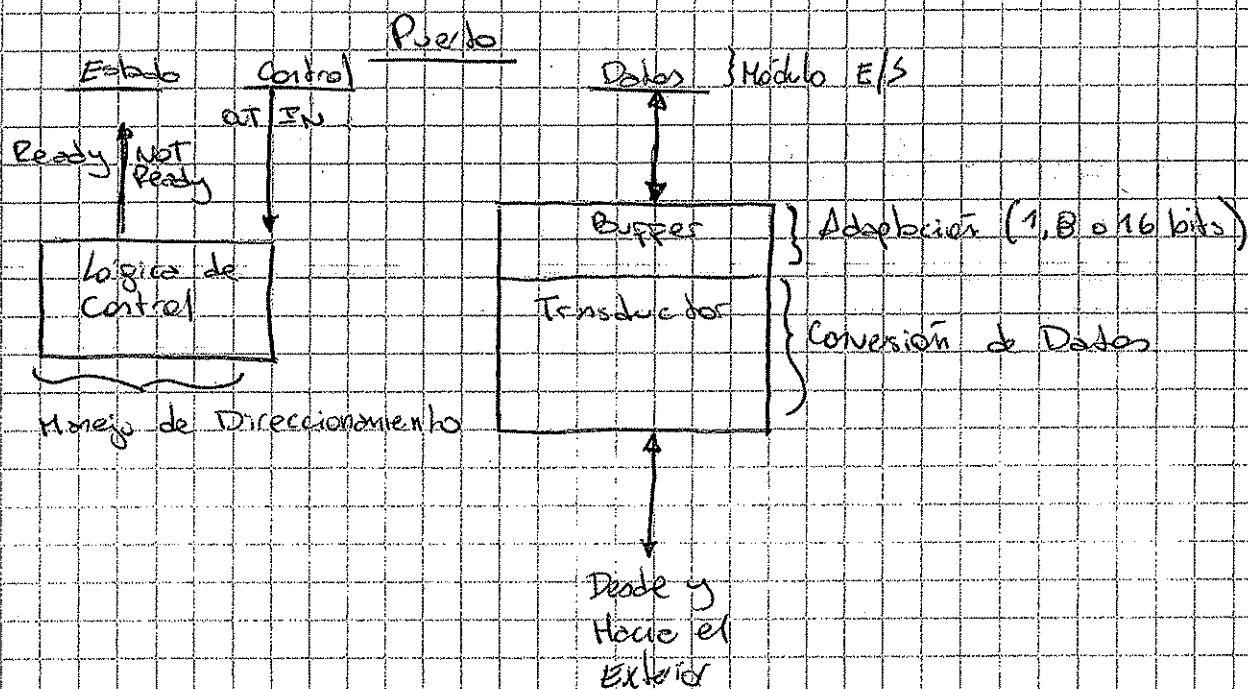
→ Resto no se usa.

## Problemas de Entrada/Salida

- Gran Variedad de Periféricos (dist. Cantidad de Datos, dif. velocidades, etc)
- Todos mas lentos que la CPU

Se necesitan módulos de E/S con "inteligencia" que realicen la interfaz entre el procesador y la memoria (bus) y los periféricos.

Poder manejar 1 o mas periféricos.



### Funciones Módulo E/S

- Control y temporización de 1 o más dispositivos ext.
- Interpretar el p.e. que recibe de la CPU y transmitirlos al periférico
- los controla y realiza la conversión de formatos, adaptación de velocidades
- lo p.e. haga falta para p.e. sea interpretada
- Almacenamiento temporal de datos (Buffering)
- Detección de errores

### Técnicas de Gestión de E/S

- Programada con espera de Respuesta
  - a) Intercambio de Datos el CPU y Módulo
  - b) CPU controla directamente la operación  
(comprueba estado, envía info y comandos)
  - c) La CPU espera que el módulo termine

¡ MUCHO TIEMPO OCIOSO!

### a. Con Interrupciones

- La CPU no tiene que esperar la finalización de la tarea de E/S.
- No se repite la comprobación de los estados.
- El módulo envía pedido de interrupción.

### Acciones de la CPU

- Envía una orden de lectura. (El módulo E/S obtiene datos de periférico, mientras CPU hace otra cosa)
- Chequea si hay pedidos de interrupciones perdidos al final del ciclo de instrucción.
- Detecta el pedido, guarda el contexto, interrumpe el proceso y realiza la gestión de la interrupción.
- Solicita Datos.

### c. Como se reconoce el dispositivo que solicita la interrupción?

- a) Dif. líneas para c/mod (Mala opción)
- b) Poll: CPU consulta cada módulo p/ saber quien fue (Lento).
- c) Hard Poll (Daisy Chain): La línea de reconocimiento de interrupción se conecta encadenando los módulos, la línea de pedidos es compartida. Una vez enviada la confirmación de parte de la CPU el módulo ~~se~~ ~~responde~~ colocando un vector, en el bus, que lo identifica. La CPU utiliza el vector como puntero para acceder a la rutina de servicio.

Interrupciones Múltiples: Todas las líneas tienen un orden de prioridad, las que tienen mayor prioridad pueden interrumpir a las líneas con menor prioridad. Si existe un maestro del Bus, solo él puede interrumpir.

PIO : Interfaz de E/S de propósito general, que presenta 2 puertos paralelos de 8 bits, denominados A y B que se pueden programar individualmente (Puede configurarse como entradas).

PA y PB = Contienen el dato presente en A y B

CA y CB = Control, permiten programar cada bit como entrada (1) o salida (0).

$\left. \begin{array}{l} 30H = PA \\ 31H = PB \\ 32H = CA \\ 33H = CB \end{array} \right\}$  Tienen 2 configs.: Se conectan a los leds correspondientes

$\left. \begin{array}{l} PA0 : \text{Busy} \\ PA1 : \text{Strobe} \\ PB0 : 7 : \text{Datos} \end{array} \right\}$  Impresora

HAND : Interfaz de periferia que cumple la temporización especificada en el Interfaz Centronics (BASURA), no admitiendo modo de programación alguno. Se comunica con la CPU a través del bit 7 de su registro de estado.

2 Registros internos = Dato

EST = INT X X X X X STR Busy  
                    └──┬──┘      └┬┘      └┬┘  
                    Basura      Inten.      D A

→ Lectura = 0 no activa INT

1 INT se activa cuando Busy no este Activa

Escritura = 0 inhibe la linea INT

1 INT se activara s/ Busy.



Timer = Contador de eventos, que realiza una cuenta ascendente los pulsos de la señal aplicada a su entrada INT, restándose el valor inicial de cuenta al final de la misma.

2 Reg. Internos = comp = Determina el módulo de la cuenta Timer  
Cont = Registro contador, que muestra la cuenta de los pulsos de la señal aplicada a la entrada INT del periférico. La coincidencia de su valor con el del registro anterior provoca la activación de la salida OUT.

PIC (Controlador de Interrupciones) = Maneja hasta 8 peticiones de INT independientes al mismo tiempo. Tiene la capacidad de mantener memorizadas las interrupciones secundarias mientras el procesador da servicio a la más prioritaria. (Prioridad 0...7)

Al final de la rutina de servicio de la interrupción habia que mandar al registro de comandos del controlador el comando final de la interrupción (EOI) representado por el valor 20H

Registros internos de 8 bits

• ISR = Reg. Interrupción de Serv. Indica cual int. está siendo atendida, mediante la puesta a 1 del bit asociado.

• IRR = Reg. Petición (Request) Int. Almacena las int. demandadas hasta el momento. 1 o 0 cuando es atendida

• IMR = Mascara de Int. Permite el enmascaramiento selectivo de cada una de las entradas de int. mediante la puesta 1 de su bit asociado.

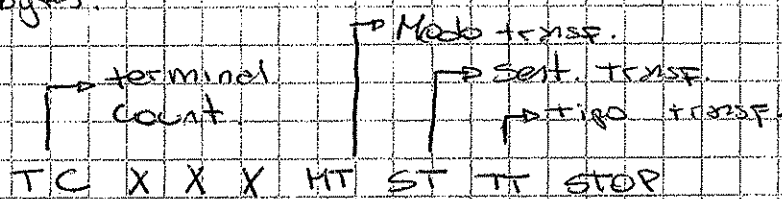
• INT 0 • INT 7 = Cada uno contiene el valor del vector de int. correspondiente a la entrada del mismo nombre.

• EOI = Solo escritura. La CPU deberá mandar el comando de final de interrupción, representado por el valor 20H.

Controlador DMA: Posibilita la transferencia de datos de 8 bits memoria - memoria o memoria - periférico y a la inversa, sin intervención directa de la CPU y en robo de ciclo (La CPU le cede los buses) siendo el tamaño máximo del bloque a transferir 64 kbytes.

Registros:

1) CTBL =



Escritura. CO

CO → 0 NA  
CO → 1 CPU debe estar la transf. en curso

S0H } Dir. F

S1H }

S2H } Cont. Oct

S3H }

S4H } Dir. Dest.

S5H }

S6 } Tipo

S7 } Transf.

C1 → 0 peri - mem, o inversa.

C1 → 1 mem - mem

C2 → 0, peri - mem

C2 → 1, mem - per

} Solo tiene sentido si C1 = 0

C3 → 0, transf. por demanda

C3 → 1, transf. por bloque (transmite datos sin prr hasta que se agota el bloque a transferir)

En lectura - Co

- 0, transf. en curso
- 1, transf. detenida x CPU

• C7

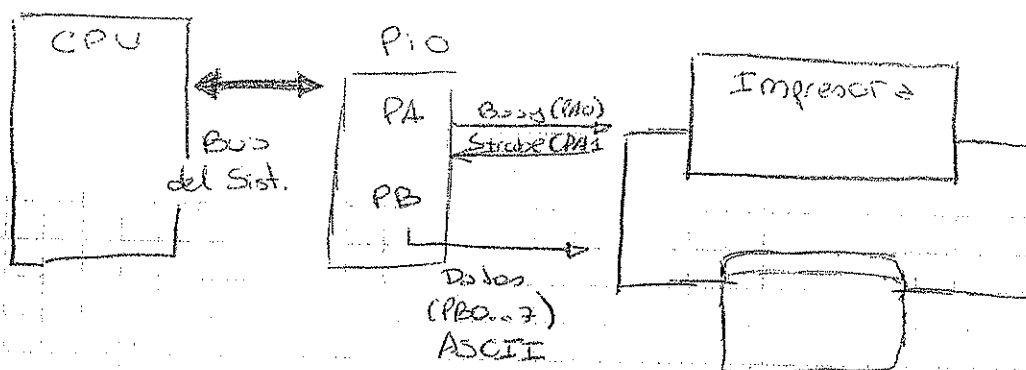
- 0, transf. no finalizada
- 1, ya finalizada

- 2) RF = Reg. de Direcciones fuente. En transf. mem-per (Inversa) se carga en él la Dir. del bloque de memoria a transferir o recibir. (Mem-Mem, Idem ~~en nombre~~)
- 3) RD = Reg. Direcciones destino. Solo tiene sentido en Mem-Mem.
- 4) Cont = Registro contador. Indica el número de octetos a transferir.

Los registros del CMA se ubican a partir de 50H

### Comunicación CPU - Impresora

#### 1) CPU - PIO - Impresora



```

Poll: IN AL, PA }
      AND AL, 1 } Preguntas
      JNZ Poll   } Impresora es libre

      MOV AL, [BX] }
      OUT PB, AL   } Enviar Dato

      MOV AL, 0FH }
      OUT PA, AL   } Pulso de Strobe
      MOV AL, 0H
      OUT PA, AL

      INC BX
      DEC CL }
      JNZ Poll } ¿Ultimo char?
  
```

### Inicialización PIO Impresora

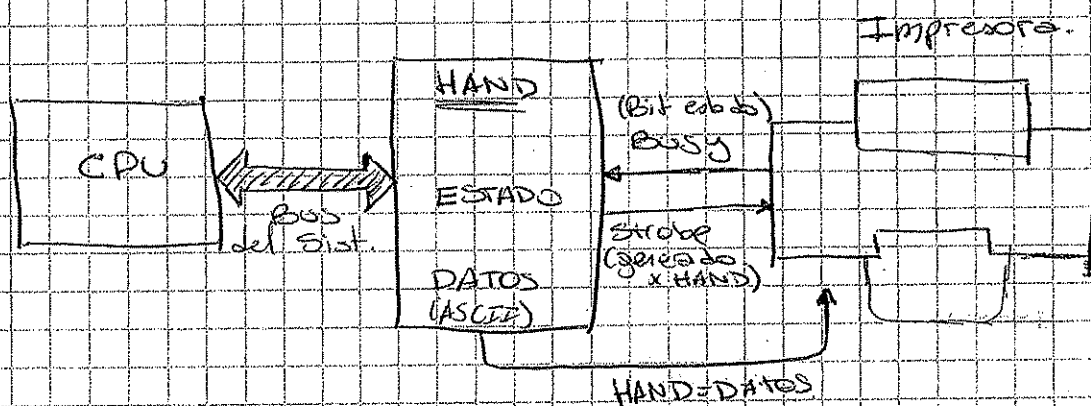
```

CA EQU 32H
CB EQU 33H

ORG 2000H

MOV AL, 0FH
OUT CA, AL
MOV AL, 0
OUT CB, AL
AND AL, 0FH
OUT PIO, AL
  
```

### 2) a) CPU - HAND - IMPRESORA (Poll)



```

    IN AL, HAND+1
    AND AL, 7FH ; Pone a "0" a INT
    OUT HAND+1, AL

Poll: IN AL, HAND+1
      AND AL, 1
      JNZ Poll } libre?

      MOV AL, [BX]
      OUT HAND, AL } Envia Datos

      INC BX
      DEC CL
      JNZ Poll } Último char?
  
```

2/b) CPU - HAND - IMPRESSORA (Interrupção)

PIC EQU 20H                      ORG 40H  
HAND EQU 40H                  IRHAND DW BUT-HAND  
N-HAND EQU 10

Org. ZOOCH.

# MENSAJE

CCI

MOV AL, OFBH } HABILITO en  
OUT PIC+1, AL } IMR INT x HAND

```
MOV AL, N_HAND } Índice a INT.6
OUT PIC16, AL } Espacio vector
```

MOV AL, 80H  
OUT HAND+1, AL } Envía 10000000 → Línea INT Activada  
// interrupciones.

513

Lazot CMP CL, O  
JNZ Lazot

```
IN AL, HAND+1 } Detenga possibilidade  
AND AL, 7FH } de p/ Impresso  
OUT HAND+1, AL } interrompe.  
INT 0  
END
```

```
ORG 3000H
D:  PUSH AX
    MOV AL, [BX]
    OUT HAND, AL
    INC BX
    DEC CL
    MOV AL, 204H } EOI
    OUT PIC, AL
    POP AX
    IRET
```

LO IMPORTANTE!

### 3) CPU - DMA - Impresora (por demanda)

```
PIC EQU 20H
DMA EQU 50H
HAND EQU 40H
N-DMA EQU 20
```

```
ORG 80
IP-DMA DW 255-DMA
```

```
ORG 1000H
MENSAJE
```

```
ORG 2000H
CLI
MOV AL, N-DMA } Reg INT 3 del PIC
OUT PIC+7, AL }
MOV AX, OFFSET MSJ }
OUT DMA, AL } Dir comienzo del
MOV AL, AH } bloque a transferir
OUT DMA+1, AL }
```

```
MOV AX, OFFSET FIN - OFFSET MSJ }
OUT DMA+2, AL } Cantidad a
MOV AL, AH } transferir
OUT DMA+3, AL }
```

SSH y SSH No se usa xp no tiene sentido indicar destino en men - periférico

```
MOV AL, 4
OUT DMA+6, AL } 0100, Inicia control DMA
```

```
MOV AL, F7H
OUT PIC+1, AL -> HABILITA INT 3
OUT DMA+7, AL -> ARRANQUE TRANSFER
```

```
MOV AL, 80H
OUT HAND+1, AL
STI
```

```
ORG 2000H
```

```
PUT DMA: MOV AL, 0
OUT HAND+1, AL
MOV FLAG, 1
MOV AL, 0FFH
OUT PIC+1, AL } Inhabilita
MOV AL, 20H } Interrupción
OUT PIC, AL } Hand y PIC.
IRET
```

```
LOZO: CMP FLAG, 1
JNZ LOZO
INT 0
END
```