
Arquitectura 2023

Explicación 2

Interrupciones

Objetivos:

Comprender la utilidad de las interrupciones por software y por hardware y el funcionamiento del Controlador de Interrupciones Programable (PIC).

Escribir programas en el lenguaje assembler del simulador MSX88.

Ejecutarlos y verificar los resultados, analizando el flujo de información entre los distintos componentes del microprocesador.

Interrupciones

- Mecanismo mediante el cual se puede interrumpir el procesamiento normal de la CPU (*Ejecución secuencial de instrucciones de un programa*)
 - Pueden ser de origen interno o externo a la CPU.
-

¿Porqué interrumpir?

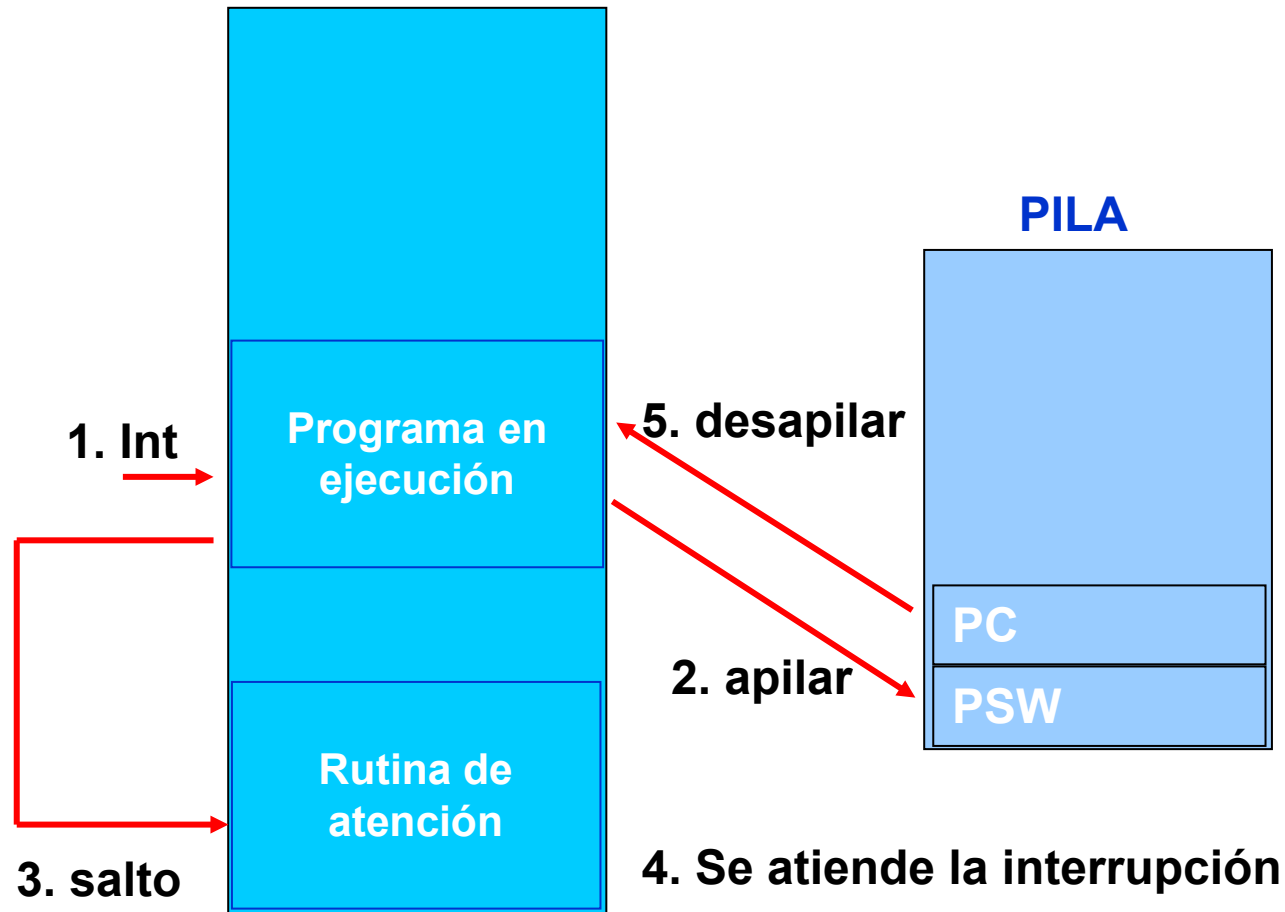
- Por resultado de una ejecución de una instrucción.
*Ej: Desbordamiento aritmético (“overflow”),
División por cero*
- Por un temporizador interno del procesador.
Permite al S.O. realizar ciertas funciones de manera regular.
- Por una operación de E/S.
Ej: Para indicar la finalización normal de una operación.
- Por un fallo de hardware.
Ej: error de paridad en la memoria.

Jerarquía de interrupciones

Si hay múltiples fuentes, se establece un orden de prioridad

- No enmascarables: No pueden ignorarse.
Indican eventos de alta prioridad.
 - Enmascarables: Pueden ser ignoradas.
Se puede inhibir su ocurrencia mediante instrucciones especiales.
-

¿Cómo trabajan las interrupciones? (I)



Implementación

El procesador tiene una única entrada de pedido de interrupciones.

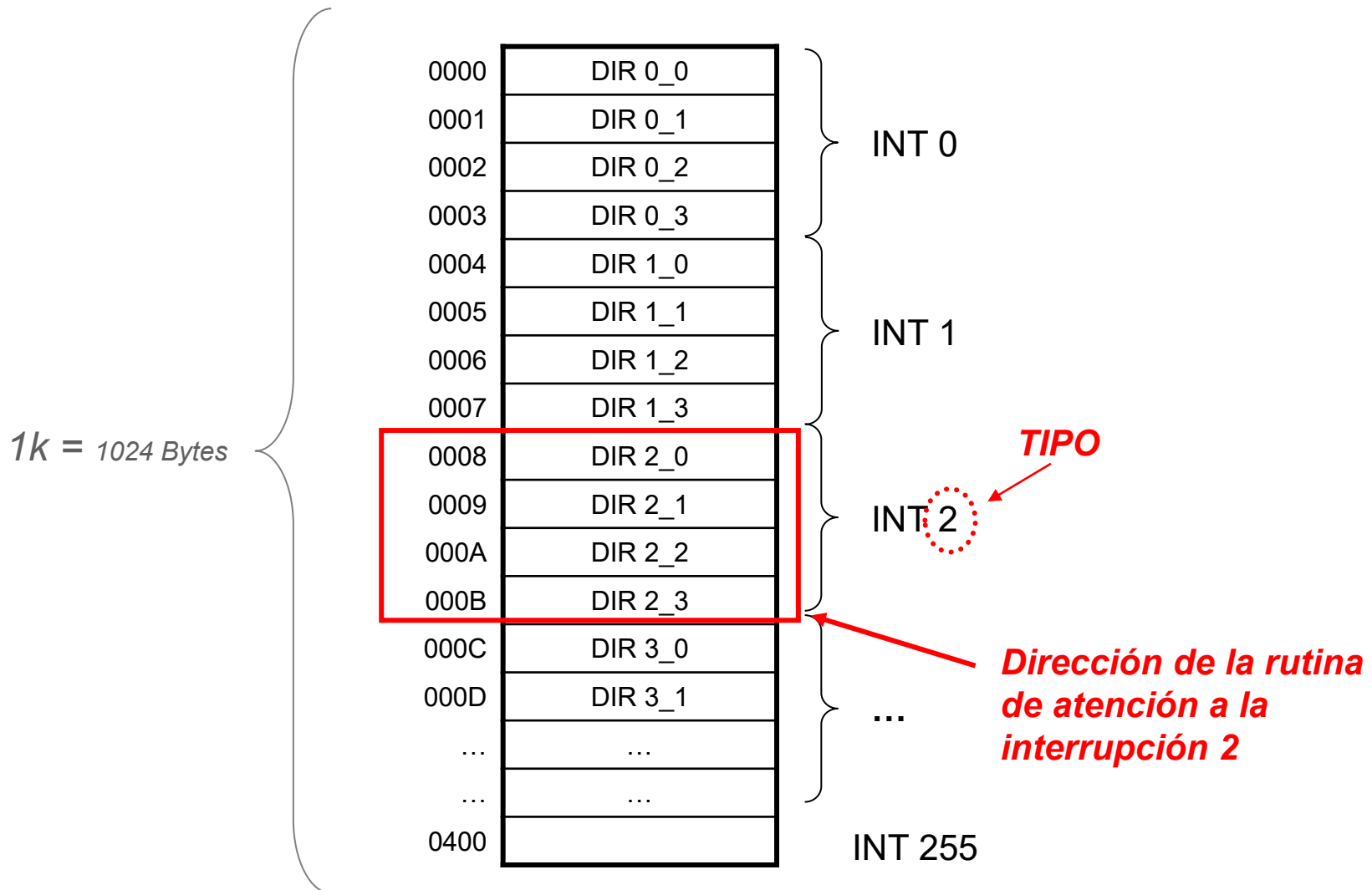
Si existen varias fuentes de interrupciones
¿Cómo se puede solucionar?

Interrupciones vectorizadas I

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

Lo puede poner el periférico directamente, o bien un controlador de interrupciones (*que se ocupa de todo*).

Vector de interrupciones

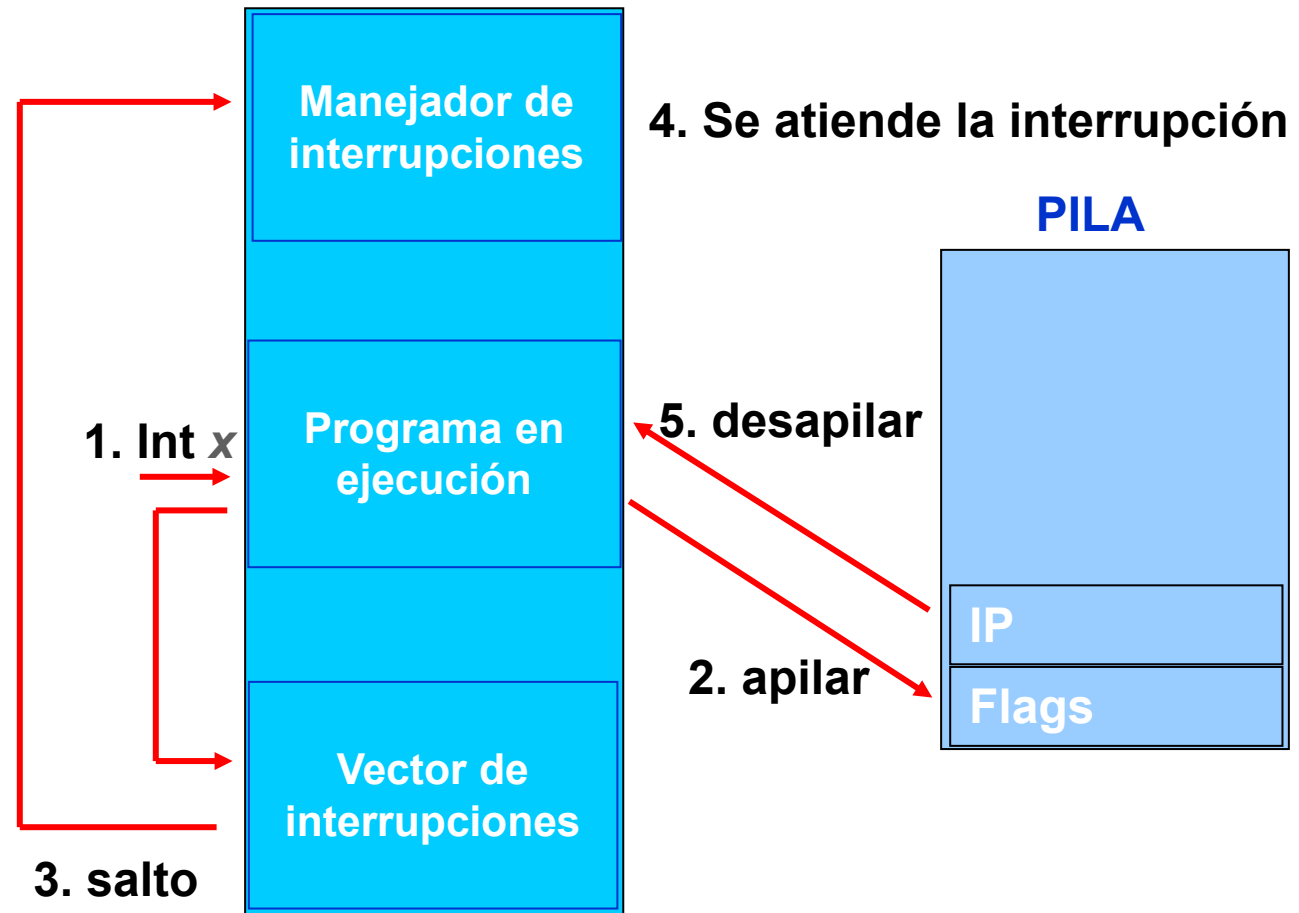


Interrupciones vectorizadas II

- Nexo entre tipo de interrupción (0...255) y el procedimiento designado para atenderla.
- Cada entrada es una doble palabra (4 bytes).
- Dirección del procedimiento que brinda el servicio.

*Ej: **0000yyyy**, donde **yyyy** es la dirección lógica/física.*

¿Cómo trabajan las interrupciones? (II)



Interrupciones por software

Vectores preasignados

- **INT 0 (*HLT*)**
- **INT 3 (*debug*)**
- **INT 6 (*leer caracter*)**
 - *BX tiene la dirección donde se almacena el carácter*
- **INT 7 (*imprimir un string*)**
 - *BX tiene la dir donde comienza el string*
 - *AL cantidad de caracteres a imprimir*

Ej 1

1) Escritura de datos en la pantalla de comandos.

Implementar un programa en el lenguaje assembler del simulador MSX88 que muestre en la pantalla de comandos un mensaje previamente almacenado en memoria de datos, aplicando la interrupción por software INT 7.

ORG 1000H

```
MSJ DB "ARQUITECTURA DE COMPUTADORAS "  
    DB "FACULTAD DE INFORMATICA "  
    DB 55H  
    DB 4EH  
    DB 4CH  
    DB 50H  
FIN DB ?
```

BX

10 00H

AX

00 39H

ORG 2000H

```
MOV BX, OFFSET MSJ  
MOV AL, OFFSET FIN-OFFSET MSJ  
INT 7  
INT 0  
END
```

ARQUITECTURA DE COMPUTADORAS FACULTAD DE INFORMATICA UNLP

Ej 4

Lectura de datos desde el teclado.

Escribir un programa que solicite el ingreso de un número (de un dígito) por teclado e inmediatamente lo muestre en la pantalla de comandos, haciendo uso de las interrupciones por software INT 6 e INT 7.

```
ORG 1000H
MSJ DB "INGRESE UN NUMERO:"
FIN DB ?
ORG 1500H
NUM DB ?
ORG 2000H
MOV BX, OFFSET MSJ
MOV AL, OFFSET FIN-OFFSET MSJ
INT 7
MOV BX, OFFSET NUM
INT 6
MOV AL, 1
INT 7
MOV CL, NUM
INT 0
END
```

BX 1500H

AX 00 01

INGRESE UN NUMERO:

4

NUM 1500 34H

CX 34H

Interrupciones por hardware

Son las generadas por dispositivos de E/S.

Son las “*verdaderas*” interrupciones.

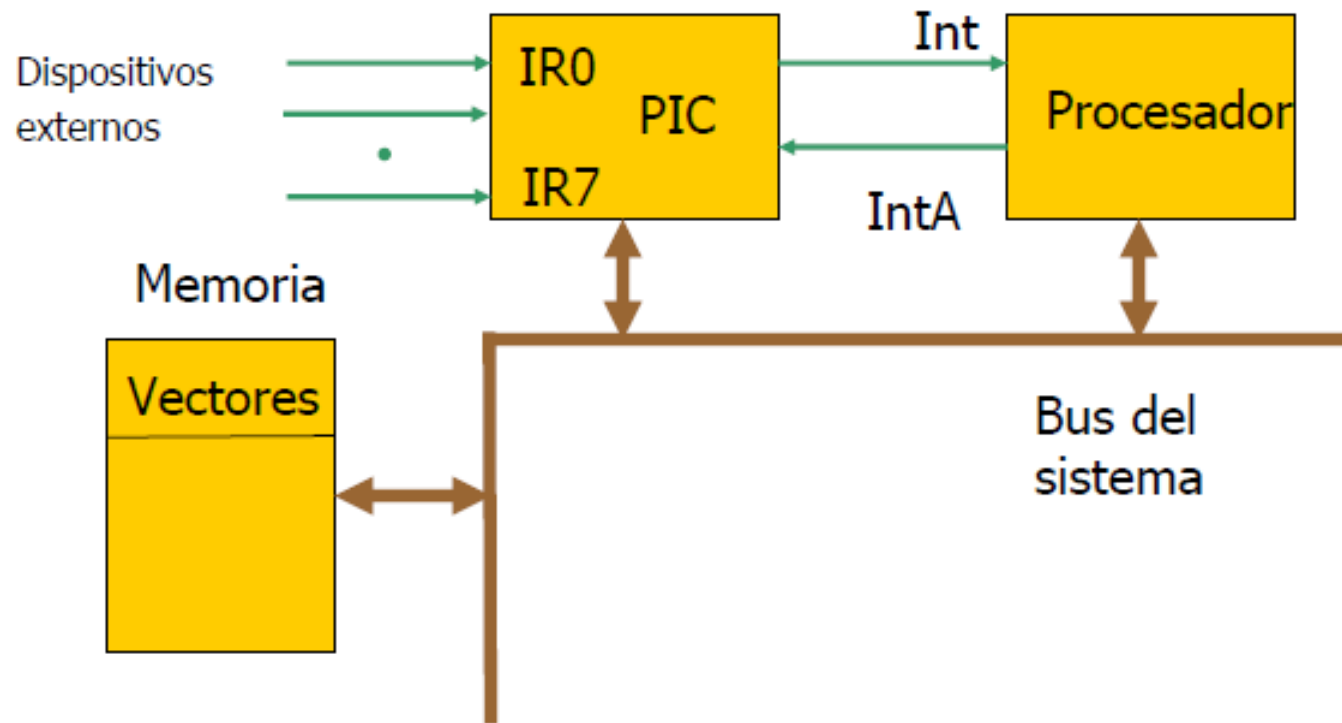
No están relacionadas con el proceso en ejecución en ese momento.

Son conocidas como ***interrupt request***.

El sistema de cómputo debe ser capaz de manejar estos eventos externos “no planeados” ó “asincrónicos”.

Dispositivo controlador de interrupciones (PIC)

- Periférico encargado de administrar los pedidos de interrupción procedentes de los demás periféricos.



PIC - Registros internos

EOI: para comandos

*Para fin de interrupción
escribir 20H*

IMR: máscara de int
enmascara con '1'

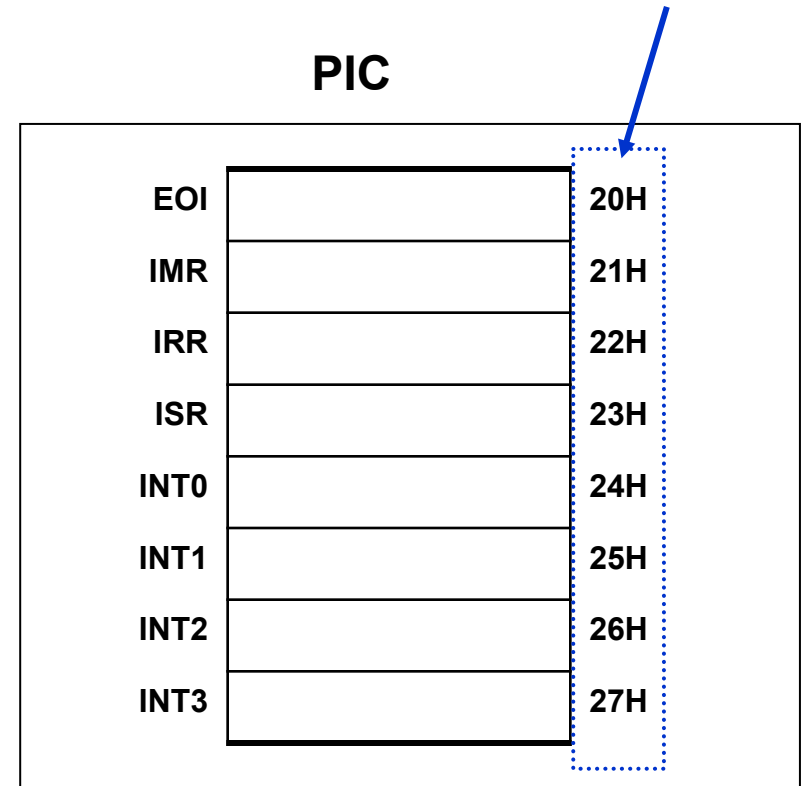
IRR: petición de int
Indica con bit en 1

ISR: int en servicio
Indica con bit en 1

INT0...INT7

c/u con su vector

Memoria de E/S



*Son accedidos con operaciones lectura y escritura en el espacio de E/S
(IN y OUT).*

PIC – Asignación de líneas

Las líneas del PIC están asignadas por defecto a algunos periféricos

INT0 → *tecla F10*

INT1 → *Timer*

INT2 → *Handshake*

INT3 → *DMA*

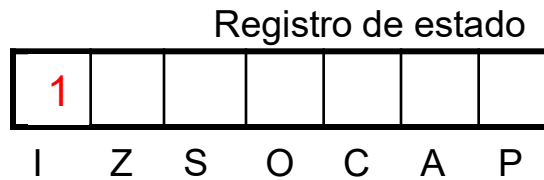
Ej 10

Interrupción por hardware: tecla F10

Escribir un programa que, mientras ejecuta un lazo infinito, cuente el número de veces que se presiona la tecla F10 y acumule este valor en el registro DX.

Ej 10

Tipo int elegido → 10



Vector	
38	...
39	...
40	00
41	00
42	00
43	30
44	...
45	...

```
PIC EQU 20H
EOI EQU 20H
N_F10 EQU 10

ORG 40
IP_F10 DW RUT_F10
```

AX 10

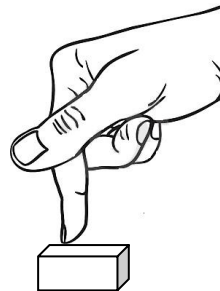
DX 1

ORG 2000H

```
CLI
MOV AL, 0FEH
OUT PIC+1, AL ; PIC: registro IMR
MOV AL, N_F10
OUT PIC+4, AL ; PIC: registro INT0
MOV DX, 0
STI
LAZO: JMP LAZO
```

ORG 3000H

```
RUT_F10: PUSH AX
INC DX
MOV AL, EOI
OUT EOI, AL ; PIC: registro EOI
POP AX
IRET
END
```



PIC		
EOI	20H	20H
IMR	0FEH = 11111110	21H
IRR		22H
ISR		23H
INT0	10	24H
INT1		25H
INT2		26H
INT3		27H

TIMER (*temporizador*)

Posee dos registros de 8 bits.

- ❑ **COMP:** *registro de comparación que determina el módulo de la cuenta del timer.*
- ❑ **CONT:** *registro contador, muestra la cuenta de los pulsos de la señal aplicada a la entrada del periférico. Cuando coincide su valor con el del registro COMP provoca una señal de salida.*
- Direcciones de registros: 10H y 11H
- Frecuencia: 1 Hz

Ej 12

Interrupción por hardware: TIMER.

Implementar a través de un programa un reloj segundero que muestre en pantalla los segundos transcurridos (00-59 seg) desde el inicio de la ejecución.

Ej12

TIMER

Cont: 10H

Comp: 11H

1
1

TIMER EQU 10H
 PIC EQU 20H
 EOI EQU 20H
 N_CLK EQU 10

ORG 40

IP_CLK DW RUT_CLK

ORG 1000H

SEG DB 30H
 DB 30H
 FIN DB ?

ORG 3000H

RUT_CLK: PUSH AX
 INC SEG+1
 CMP SEG+1, 3AH
 JNZ RESET
 MOV SEG+1, 30H
 INC SEG
 CMP SEG, 36H
 JNZ RESET
 MOV SEG, 30H
 RESET: INT 7
 MOV AL, 0
 OUT TIMER, AL
 MOV AL, EOI
 OUT PIC, AL
 POP AX
 IRET

ORG 2000H

CLI
 MOV AL, 0FDH
 ; PIC: registro IMR
 OUT PIC+1, AL
 MOV AL, N_CLK
 ; PIC: registro INT1
 OUT PIC+5, AL
 MOV AL, 1
 ; TIMER: registro COMP
 OUT TIMER+1, AL
 MOV AL, 0
 ; TIMER: registro CONT
 OUT TIMER, AL
 MOV BX, OFFSET SEG
 MOV AL, OFFSET FIN-OFFSET SEG
 STI
 LAZO: JMP LAZO
 END

PIC

EOI	20H	20H
IMR	0FDH = 11111101	21H
IRR		22H
ISR		23H
INT0		24H
INT1	10	25H
INT2		26H
INT3		27H

AX

20H

BX

10 00 H

01

02

...

10

Vector

39	...
40	00
41	00
42	00
43	30
44	...
45	...

...

1000H	31H
1001H	30H
1002H	?
1003H	
1004H	
1005H	

Registro de estado

1						
I	Z	S	O	C	A	P

¿Preguntas?