



Arquitectura de Computadoras

Curso 2020 – Prof. Jorge Runco

Ejemplos: interrupciones por software y hardware

| | |
|-------|-----|
| 0012H | 00H |
| 0013H | 00H |
| 0014H | 00H |
| 0015H | 30H |
| 0016H | 00H |

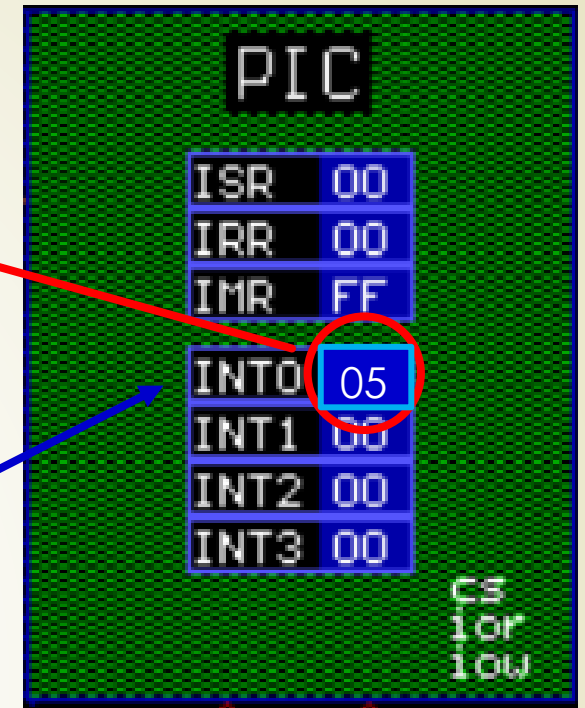
ORG 20 (4*5)
DW nombre

ORG 3000H

nombre:

IRET

ORG 2000H
OUT INTO, 05H
END



EJ10TP2

0028 00 30

3000 50
3001 FF C2
3003 B0 20
3005 E6 20
3007 58
3008 CF

```

1 PIC EQU 20H
2 EOI EQU 20H
3 N_F10 EQU 10
4
5 ORG 40
6 IP_F10 DW RUT_F10
7
8 ORG 3000H
9 RUT_F10: PUSH AX
10 INC DX
11 MOV AL, EOI
12 OUT EOI, AL
13 POP AX
14 IRET
15

```

```

16
17 2000 FA
18 2001 B0 FE
19 2003 E6 21
20 2005 B0 0A
21 2007 E6 24
22 2009 BA 00 00
23 200C FB
24 200D E9 0D 20
25

```

```

ORG 2000H
CLI
MOV AL, 0FEH
OUT PIC+1, AL
MOV AL, N_F10
OUT PIC+4, AL ;
MOV DX, 0
STI
LAZO: JMP LAZO
END

```

IMR 11111110

INT0 00001010

Ejemplo interrupciones por hardware

Ejemplo interrupciones por hardware

EJ5TP2

PILA

| | |
|-------|-----|
| 7FFEh | 10H |
| 7FFFh | 20H |
| 8000H | 00H |

SUBROUTINA

| | |
|-------|-----|
| 3000H | 50H |
| 3001H | FFH |
| 3002H | C2H |
| 3003H | B0H |

PRINCIPAL

| | |
|-------|-----|
| 200CH | 00H |
| 200DH | E9H |
| 200EH | 0DH |
| 200FH | 20H |
| 2010H | 00H |

CPU

4x10=40=28H

SP 7FFEh

IP 3000H

VECTORES (MEM)

| | |
|-------|-----|
| 0027H | 00H |
| 0028H | 00H |
| 0029H | 30H |
| 002AH | 00H |

PIC

1111111 0

IMR

INT0 10

INT1

INT2

INT3

F10

BUS DE DATOS

EJ5TP2

```

        ORG 1000H
NUM      DB ?
MSJ1     DB "CARACTER NO VÁLIDO"
FIN1     DB ?
MSJ2     DB "CARACTER VÁLIDO"
FIN2     DB ?
    
```

```

        ORG 3000H
ES_NUM: MOV AH, [BX]
        SUB AH, 30H
        CMP AH, 10
        JNC NO_ES
        MOV AH, 0FFH
        JMP FIN
NO_ES:  MOV AH, 00H
FIN:    RET
    
```

NO_ES_NUM:

AFUERA:

```

        ORG 2000H
        MOV BX, OFFSET NUM
        INT 6
        CALL ES_NUM
        CMP AH, 00H
        JZ NO_ES_NUM
        MOV BX, OFFSET MSJ2
        MOV AL, OFFSET FIN2 - OFFSET MSJ2
        INT 7
        JMP AFUERA
        MOV BX, OFFSET MSJ1
        MOV AL, OFFSET FIN1 - OFFSET MSJ1
        INT 7
        HLT
        END
    
```

Modificar el programa anterior agregando una subrutina llamada ES_NUM que verifique si el caracter ingresado es realmente un número. De no serlo, el programa debe mostrar el mensaje “CARACTER NO VALIDO”. La subrutina debe recibir el código del caracter por referencia desde el programa principal y debe devolver vía registro el valor 0FFH en caso de tratarse de un número o el valor 00H en caso contrario. Tener en cuenta que el código del “0” es 30H y el del “9” es 39H.

EJ5TP2

```
ORG 2000H
MOV BX, OFFSET NUM
INT 6
CALL ES_NUM
CMP AH, 00H
JZ NO_ES_NUM
MOV BX, OFFSET MSJ2
MOV AL, OFFSET FIN2 - OFFSET MSJ2
INT 7
JMP AFUERA
NO_ES_NUM: MOV BX, OFFSET MSJ1
MOV AL, OFFSET FIN1 - OFFSET MSJ1
INT 7
AFUERA:  HLT
END
```

En BX se carga con la dirección donde se almacena el carácter ingresado.

El carácter ingresado por teclado se almacenó en NUM (BX). Ahora llamamos a la rutina que comprueba si es un número. Si es un número AH=FF y si no AH=00

EJ5TP2

```
ORG 3000H
ES_NUM: MOV AH, [BX]
        SUB AH, 30H
        CMP AH, 10
        JNC NO_ES
        MOV AH, 0FFH
        JMP FIN
NO_ES:  MOV AH, 00H
FIN:    RET
```

BX 1000H

AX 0000H

AX

Supongamos que el carácter ingresado es un 8.

Si es un número el resultado de la cuenta estará entre 0 y 9. Al restar 10 habrá borrow (Carry=1). Si no es un número no hay borrow (Carry=0).

Estas 4 instrucciones funcionan como un if
→ if (no hay carry) then AH=00 else AH=FF

EJ5TP2

```
ORG 3000H
ES_NUM: MOV AH, [BX]
        SUB AH, 30H
        CMP AH, 10
        JNC NO_ES
        MOV AH, 0FFH
        JMP FIN
NO_ES:  MOV AH, 00H
FIN:    RET
```

Si hay carry no salta (es un número).
Hace el else AH=FF. El JMP hace falta para no ejecutar el then también.

Si no hay carry salta a NO_ES (no es un número).

AX FF 00H

AX 00 00H

Estas 4 instrucciones funcionan como un if
→ if (no hay carry) then AH=00 else AH=FF


```
ORG 2000H
MOV BX, OFFSET NUM
INT 6
CALL ES_NUM
CMP AH, 00H
JZ NO_ES_NUM
MOV BX, OFFSET MSJ2
MOV AL, OFFSET FIN2 - OFFSET MSJ2
INT 7
JMP AFUERA
NO_ES_NUM: MOV BX, OFFSET MSJ1
            MOV AL, OFFSET FIN1 - OFFSET MSJ1
            INT 7
AFUERA:    HLT
            END
```

EJ5TP2

```
ORG 2000H
MOV BX, OFFSET NUM
INT 6
CALL ES_NUM
CMP AH, 00H
JZ NO_ES_NUM
MOV BX, OFFSET MSJ2
MOV AL, OFFSET FIN2 - OFFSET MSJ2
INT 7
JMP AFUERA
NO_ES_NUM: MOV BX, OFFSET MSJ1
           MOV AL, OFFSET FIN1 - OFFSET MSJ1
           INT 7
AFUERA:    HLT
           END
```

Es un if que analiza el valor devuelto por la subrutina en AH. Llama a INT 7 para mostrar un cartel en la pantalla. Recordemos que antes de llamar a la función que imprime en pantalla, los parámetros pasados son BX= dirección donde empieza el mensaje y AL= cantidad de elementos a mostrar.