

PRÁCTICA 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.

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  ?

    ORG 2000H
    MOV BX, OFFSET MSJ
    MOV AL, OFFSET FIN-OFFSET MSJ
    INT 7
    INT 0
    END

```

2) Escribir un programa que muestre en pantalla todos los caracteres disponibles en el simulador MSX88, comenzando con el caracter cuyo código es el número 01H.

3) * Escribir un programa que muestre en pantalla las letras del abecedario, sin espacios, intercalando mayúsculas y minúsculas (AaBb...), sin incluir texto en la memoria de datos del programa. Tener en cuenta que el código de "A" es 41H, el de "a" es 61H y que el resto de los códigos son correlativos según el abecedario.

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

```

Responder brevemente:

- Con referencia a la interrupción INT 7, ¿qué se almacena en los registros BX y AL?
- Con referencia a la interrupción INT 6, ¿qué se almacena en BX?
- En el programa anterior, ¿qué hace la segunda interrupción INT 7? ¿qué queda almacenado en el registro CL?

5) 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 carácter 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.

6) * Escribir un programa que solicite el ingreso de un número (de un dígito) por teclado y muestre en pantalla dicho número expresado en letras. Luego que solicite el ingreso de otro y así sucesivamente. Se debe finalizar la ejecución al ingresarse en dos vueltas consecutivas el número cero.

7) * Escribir un programa que efectúe la suma de dos números (de un dígito cada uno) ingresados por teclado y muestre el resultado en la pantalla de comandos. Recordar que el código de cada carácter ingresado no coincide con el número que representa y que el resultado puede necesitar ser expresado con 2 dígitos.

8) Escribir un programa que efectúe la resta de dos números (de un dígito cada uno) ingresados por teclado y muestre el resultado en la pantalla de comandos. Antes de visualizarlo el programa debe verificar si el resultado es positivo o negativo y anteponer al valor el signo correspondiente.

9) Escribir un programa que aguarde el ingreso de una clave de cuatro caracteres por teclado sin visualizarla en pantalla. En caso de coincidir con una clave predefinida (y guardada en memoria) que muestre el mensaje "Acceso permitido", caso contrario el mensaje "Acceso denegado".

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.

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

        ORG 40
IP_F10   DW  RUT_F10

        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
```

Explicar detalladamente:

- La función de los registros del PIC: ISR, IRR, IMR, INT0-INT7, EOI. Indicar la dirección de cada uno.
- Cuáles de estos registros son programables y cómo trabaja la instrucción OUT.
- Qué hacen y para qué se usan las instrucciones CLI y STI.

11) Escribir un programa que permita seleccionar una letra del abecedario al azar. El código de la letra debe generarse en un registro que incremente su valor desde el código de A hasta el de Z continuamente. La letra debe quedar seleccionada al presionarse la tecla F10 y debe mostrarse de inmediato en la pantalla de comandos.

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.

```

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

                END

```

Explicar detalladamente:

- a) Cómo funciona el TIMER y cuándo emite una interrupción a la CPU.
- b) La función que cumplen sus registros, la dirección de cada uno y cómo se programan.

13) Modificar el programa anterior para que también cuente minutos (00:00 - 59:59), pero que actualice la visualización en pantalla cada 10 segundos.

14) * Implementar un reloj similar al utilizado en los partidos de básquet, que arranque y detenga su marcha al presionar sucesivas veces la tecla F10 y que finalice el conteo al alcanzar los 30 segundos.

15) Escribir un programa que implemente un conteo regresivo a partir de un valor ingresado desde el teclado. El conteo debe comenzar al presionarse la tecla F10. El tiempo transcurrido debe mostrarse en pantalla, actualizándose el valor cada segundo.

Nota: Los ejercicios marcados poseen una resolución propuesta