

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 VonSim. 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 VonSim que muestre en la pantalla de comandos un mensaje previamente almacenado en memoria de datos, aplicando la interrupción por software INT 7.

```

        ORG 1000H
MS      DB  "ARQUITECTURA DE COMPUTADORAS-"
J
        DB  "FACULTAD DE INFORMATICA-"
        DB  55H
        DB  4EH
        DB  4CH
        DB  50H
FI      DB  ?
N

        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 VonSim, 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
MS      DB  "INGRESE UN NUMERO:"
J
FI      DB  ?
N

        ORG 1500H
NU      DB  ?
M

        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:

- a) Con referencia a la interrupción INT 7, ¿qué se almacena en los registros BX y AL? *La dirección en BX desde donde empieza a leer y cuántos caracteres va a imprimir en AL*
- b) Con referencia a la interrupción INT 6, ¿qué se almacena en BX? *La dirección donde se guardará el carácter leído*
- c) En el programa anterior, ¿qué hace la segunda interrupción INT 7? ¿qué queda almacenado en el registro CL? *Imprime caracteres en pantalla. Queda guardado el carácter leído*

5) Modificar el programa anterior agregando una subrutina llamada ES_NUM que verifique si el carácter 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.

10)

a) Los registros del PIC (Programmable Interrupt Controller) son componentes esenciales en sistemas computacionales que manejan interrupciones. A continuación, se explica la función de los registros del PIC

- EOI (End of Interrupt): El registro EOI se usa para señalar al PIC que se ha completado el manejo de una interrupción. La dirección del EOI es 20H

- IMR (Interrupt Mask Register): El IMR se utiliza para habilitar o deshabilitar las líneas de interrupción específicas. Cada bit en el IMR corresponde a una línea de interrupción específica, y un bit en 1 indica que la interrupción está habilitada, mientras que un bit en 0 la deshabilita. La dirección de IMR es 21H

- IRR (Interrupt Request Register): Este registro almacena las interrupciones que han sido solicitadas por los dispositivos periféricos, pero aún no se han reconocido o atendido. La dirección de IRR es 22H

- ISR (Interrupt Service Register): Este registro indica qué dispositivo está siendo atendido. La dirección de ISR varía según el PIC. La dirección es 23H

-INT0-INT3: Estos registros contienen los IDs asignados a cada dispositivo. INT0 contiene el ID de la tecla F10 y se encuentra en la dirección 24H, INT1 contiene el ID del timer y se encuentra en la dirección 25H, INT2 contiene el ID del handshake y se encuentra en la dirección 26H, INT3 contiene el ID del CDMA y se encuentra en la dirección 27H.

b) Los registros programables en el PIC son el EOI y el IMR. La instrucción OUT se utiliza para programar estos registros.

c) Las instrucciones CLI (Clear Interrupt Flag) y STI (Set Interrupt Flag) son instrucciones que controlan la habilitación y deshabilitación de las interrupciones en la CPU:

CLI: La instrucción CLI se utiliza para borrar (desactivar) el bit de bandera de interrupción (IF) en el registro de banderas (FLAGS) de la CPU. Cuando se borra el bit IF, las interrupciones no se procesan y se desactivan.

STI: La instrucción STI se utiliza para establecer (activar) el bit de bandera de interrupción (IF) en el registro de banderas (FLAGS) de la CPU. Cuando se establece el bit IF, las interrupciones se habilitan y se procesan si se presenta una solicitud de interrupción.

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:

- Cómo funciona el TIMER y cuándo emite una interrupción a la CPU.
- 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.

12) Su función principal es generar interrupciones periódicas a la unidad central de procesamiento (CPU) a intervalos regulares. Estas interrupciones son útiles para una variedad de propósitos, como medir el tiempo, realizar tareas periódicas y controlar el flujo del programa.

Registro de Contador (TCNT - Timer Counter): Este registro almacena el valor actual del contador del TIMER. Puedes leer y escribir en este registro para conocer o establecer el estado actual del contador. Dirección 10H

Registro de Comparación (OCR - Output Compare Register): Este registro almacena el valor de comparación que se utiliza para generar la interrupción cuando el contador alcanza este valor. Puedes programar este registro para definir cuándo se disparará la interrupción. Dirección 11H

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