

# PRÁCTICA 3 - RESOLUCIÓN

## *Interrupciones por Hardware*

**Objetivos:** Comprender la utilidad de las interrupciones por hardware y el funcionamiento del Controlador de Interrupciones Programable (PIC). Utilizar el TIMER, F10, y el dispositivo de handshaking (HAND-SHAKE) mediante interrupciones para el intercambio de información entre el microprocesador y el mundo exterior.

### Parte 1: PIC y Vector de Interrupciones

- 1) **Registros del PIC** ★ Dados los siguientes registros del PIC indique su función. Además, para el valor de ejemplo, indique qué significa en términos del funcionamiento del programa

Dirección	Registro	Función	Valor de Ejemplo
20h	EOI	<b>Fin de Interrupción.</b> Solo para escribir, se debe mandar el comando de fin de interrupción cuando se terminó de servir la interrupción. (Valor 20h)	No aplica
21h	IMR	<b>Registro de Mascara de Interrupciones.</b> Permite enmascarar selectivamente las interrupciones que va a recibir el PIC. Cada bit de este registro representa una línea de interrupción (bit 0 se asocia a la entrada INT0 ... bit 7 se asocia a la entrada INT7). Si el valor es puesto en 1, esa interrupción estará inhibida.	1111 1101  Esta habilitada la interrupción INT1 (bit 1 en 0). Esa interrupción está conectada al Timer.
22h	IRR	<b>Registro de Petición de Interrupciones.</b> Almacena las interrupciones pendientes. Cada bit de este registro representa una línea de interrupción (bit 0 se asocia a la entrada INT0 ... bit 7 se asocia a la entrada INT7). Cuando la interrupción se satisface, el bit de dicha línea se pone a cero.	0000 0011  Se recibió y están pendientes de ser atendidas las interrupciones INT0 e INT1. Esa interrupción está conectada a la tecla F-10 y al Timer.

23h	ISR	<b>Registro de Interrupción en Servicio.</b> El bit que representa la línea de interrupción que se está atendiendo estará en 1 interrupción (bit 0 se asocia a la entrada INT0 ... bit 7 se asocia a la entrada INT7), el resto en 0. Si no hay interrupciones atendiéndose todos los bits valen 0.	0000 0100 La CPU está ejecutando el manejador de interrupción INT2. Esa interrupción está conectada al <i>HandShake</i> .
24h	INT0	<b>ID de Línea INT0.</b> Almacena el ID de la interrupción asociada al dispositivo <i>F10</i> para buscar en el vector de interrupciones la dirección de comienzo de la subrutina que lo atiende.	15  Es el número de vector que se le enviará a la CPU para manejar la interrupción. Si multiplico por 4 ( $15 \times 4 = 60$ ) nos da la dirección de memoria donde debe estar la dirección de rutina que maneja la interrupción correspondiente a <i>F10</i> .
25h	INT1	<b>ID de Línea INT1.</b> Almacena el ID de la interrupción asociada al dispositivo <i>Timer</i> para buscar en el vector de interrupciones la dirección de comienzo de la subrutina que lo atiende.	10  Es el número de vector que se le enviará a la CPU para manejar la interrupción. Si multiplico por 4 ( $10 \times 4 = 40$ ) nos da la dirección de memoria donde debe estar la dirección de rutina que maneja la interrupción correspondiente al <i>Timer</i> .
26h	INT2	<b>ID de Línea INT2.</b> Almacena el ID de la interrupción asociada al dispositivo <i>Handshake</i> para buscar en el vector de interrupciones la dirección de comienzo de la subrutina que lo atiende.	25  Es el número de vector que se le enviará a la CPU para manejar la interrupción. Si multiplico por 4 ( $25 \times 4 = 100$ ) nos da la dirección de memoria donde debe estar la dirección de rutina que maneja la interrupción correspondiente al <i>Handshake</i> .

- 2) **Configuración del PIC** ★ Se desean configurar algunas interrupciones mediante el PIC. Complete la tabla indicando en cada caso los registros o direcciones que faltan para que funcionen las interrupciones, o el dispositivo que se está usando.

Dispositivos	IMR	INT0	INT1	INT2	Dirección Vector
F10	1111 1110	5	No importa	No importa	20

F10	1111 1110	8	No importa	No importa	32
Timer	1111 1101	No importa	10	No importa	40
HANDSHAKE	1111 1011	No importa	No importa	5	20
Sin interrupciones	1111 1111	No importa	No importa	No importa	-
F10 y Timer	1111 1100	10	20	No importa	40 y 80
TIMER y HANDSHAKE	1111 1001	No importa	4	8	16 y 32
F10, TIMER y HANDSHAKE	1111 1000	5	10	15	20, 40, 60

Nota: En el caso de haber más de un dispositivo en uso, indicarlos en el mismo orden en las columnas "Dispositivos" y "Dirección Vector". En el caso en que una columna no deba configurarse, indicarla como "No importa".

## Parte 2: Interrupciones con la tecla F10, Timer y Handshake

- 3) **Contador de pulsaciones** ★ El siguiente programa cuenta el número de veces que se presiona la tecla F10 y acumula este valor en el registro DX. El programa nunca termina, ya que ejecuta un ciclo infinito. Completar las instrucciones faltantes para que el programa funcione correctamente.

<pre>EOI EQU 20H IMR EQU 21H INT0 EQU 24h  N_F10 EQU 15  ORG 60 IP_F10 DW RUT_F10</pre>	<pre>ORG 2000H CLI MOV AL, 0FEH OUT IMR, AL MOV AL, N_F10 OUT INT0, AL MOV DX, 0 STI LAZO: JMP LAZO</pre>	<pre>ORG 3000H RUT_F10: PUSH AX INC DX MOV AL, EOI OUT EOI, AL POP AX IRET  END</pre>
---	---	---

Luego de completarlo, verifique su funcionamiento en el simulador, y explicar detalladamente:

- A. ¿Qué hacen las instrucciones CLI y STI? ¿Qué podría suceder si no están las mismas y la persona que usa el programa presiona F10 rápidamente durante la configuración del PIC?

La instrucción **CLI** inhibe las interrupciones enmascarables. la instrucción **STI** las vuelve a habilitar. Las interrupciones enmascarables son las interrupciones gestionadas por el PIC.

Si no están las mismas mientras configuramos el PIC, la CPU podría recibir un pedido de interrupción de un vector que aún no este configurado (en este caso el registro INT0), eso podría hacer que la CPU busque la rutina manejadora de la interrupción en la dirección equivocada y se termine ejecutando código en otra dirección de memoria.

- B. ¿Por qué se usa el valor 0FEH en el programa?

El valor 0FEH lo usamos para configurar el registro **IMR** del PIC, para habilitar solamente la interrupción **F10** del simulador.

- C. ¿Qué instrucciones le indican al PIC que la interrupción ha terminado?

Se debe escribir el valor 20h en el registro EOI para indicar al PIC que se terminó de procesar la interrupción. Esto lo hacemos con las instrucciones:

```
MOV AL, EOI
OUT EOI, AL
```

D. ¿En qué dirección se encuentra la subrutina que atiende las interrupciones del F10?

La subrutina que atiende la interrupción se encuentra a partir de la dirección 3000h.

#### 4) Tecla F10 con pantalla y terminación ★★

A) Escribir un programa que muestre en pantalla “Vamos las interrupciones!” cada vez que se presiona la tecla F10. El programa nunca termina.

```
EOI EQU 20H
IMR EQU 21H
INT0 EQU 24h

N_F10 EQU 15

ORG 60
IP_F10 DW RUT_F10

ORG 1000H
TEXT0 DB "Vamos las interrupciones!"
FIN_TEXT0 DB ?

ORG 3000H
RUT_F10: PUSH AX
        PUSH BX
        MOV BX, OFFSET TEXT0
        MOV AL, OFFSET FIN_TEXT0 - OFFSET TEXT0
        INT 7
        MOV AL, EOI
        OUT EOI, AL
        POP BX
        POP AX
        IRET

ORG 2000H
CLI
MOV AL, 0FEH
OUT IMR, AL
MOV AL, N_F10
OUT INT0, AL
STI
LAZO: JMP LAZO

END
```

B) Idem A pero ahora el programa termina luego de mostrar 5 veces el mensaje (es decir, luego de que presione 5 veces la tecla F10).

**Pista:** cuando se ejecuta la 5ta interrupción, deben deshabilitarse las mismas para evitar reaccionar a próximas pulsaciones de la tecla F10. Utilizar una variable global para registrar cuántas veces falta mostrar el mensaje antes de terminar.

```
EOI EQU 20H
IMR EQU 21H
INT0 EQU 24h

    ORG 1000H
TEXT0 DB "Vamos las interrupciones!"
FIN_TEXT0 DB ?
CANT DB 0

    ORG 2000H
    CLI
    MOV AL, 0FEH
    OUT IMR, AL
    MOV AL, N_F10
    OUT INT0, AL
    STI
LAZO: CMP CANT, 5
      JNZ LAZO
      INT 0

    ORG 3000H
RUT_F10: PUSH AX
        PUSH BX
        MOV BX, OFFSET TEXT0
        MOV AL, OFFSET FIN_TEXT0 - OFFSET TEXT0
        INT 7
        INC CANT
        CMP CANT, 5
        JNZ SIGUE
        MOV AL, 0FFh
        OUT IMR, AL
SIGUE: MOV AL, EOI
        OUT EOI, AL
        POP BX
        POP AX
        IRET

END
```

5) **Timer, tres usos: periódico infinito, periódico finito, y única vez** ★★

- a) Escribir un programa que muestra el mensaje "Vamos las interrupciones!" una vez cada 2 segundos y no termine nunca.

```
EOI EQU 20H
IMR EQU 21H
INT1 EQU 25h

CONT EQU 10h
COMP EQU 11h

N_TIMER EQU 20
```

```
ORG 80
IP_F10 DW RUT_TIMER

    ORG 1000H
TEXTO DB "Vamos las interrupciones!"
FIN_TEXTO DB ?

    ORG 2000H
    CLI
    MOV AL, 0FDH
    OUT IMR, AL
    MOV AL, N_TIMER
    OUT INT1, AL
    MOV AL, 2
    OUT COMP, AL
    MOV AL, 0
    OUT CONT, AL
    STI
LAZO: JMP LAZO

    ORG 3000H
RUT_TIMER: PUSH AX
    PUSH BX
    MOV BX, OFFSET TEXT0
    MOV AL, OFFSET FIN_TEXT0 - OFFSET TEXT0
    INT 7
    MOV AL, EOI
        OUT EOI, AL
    MOV AL, 0
    OUT CONT, AL
    POP BX
        POP AX
    IRET

END
```

b) Modificar a) para que termine a los 10 segundos.

```
EOI EQU 20H
IMR EQU 21H
INT1 EQU 25h

CONT EQU 10h
COMP EQU 11h

N_TIMER EQU 20

ORG 80
IP_F10 DW RUT_TIMER

    ORG 1000H
TEXT0 DB "Vamos las interrupciones!"
FIN_TEXT0 DB ?
CANT DB 0

    ORG 2000H
    CLI
```

```

        MOV AL, 0FDH
        OUT IMR, AL
        MOV AL, N_TIMER
        OUT INT1, AL
        MOV AL, 2
        OUT COMP, AL
        MOV AL, 0
        OUT CONT, AL
        STI
LAZO:   CMP CANT, 5

        JNZ LAZO
        INT 0

        ORG 3000H
RUT_TIMER:  PUSH AX
            PUSH BX
            MOV BX, OFFSET TEXTO
            MOV AL, OFFSET FIN_TEXTO - OFFSET TEXTO
            INT 7
            INC CANT
            CMP CANT, 5
            JNZ SIGUE
            MOV AL, 0FFh
            OUT IMR, AL
SIGUE:    MOV AL, EOI
            OUT EOI, AL
            MOV AL, 0
            OUT CONT, AL
            POP BX
            POP AX
            IRET
END

```

c) Modificar a) para que solo imprima una vez, a los 10 segundos, y luego termine.

```

EOI EQU 20H
IMR EQU 21H
INT1 EQU 25h

CONT EQU 10h
COMP EQU 11h

N_TIMER EQU 20

ORG 80
IP_F10 DW RUT_TIMER

        ORG 1000H
TEXTO DB "Vamos las interrupciones!"
FIN_TEXTO DB ?
TERMINA DB 0

        ORG 3000H

```

```

RUT_TIMER: PUSH AX
            PUSH BX
            MOV BX, OFFSET TEXT0
            MOV AL, OFFSET FIN_TEXT0 - OFFSET TEXT0
            INT 7

            INC TERMINA
            MOV AL, 0FFh
            OUT IMR, AL
SIGUE:      MOV AL, EOI
            OUT EOI, AL

            MOV AL, 0
            OUT CONT, AL
            POP BX
            POP AX
            IRET

ORG 2000H
CLI
            MOV AL, 0FDH
            OUT IMR, AL
            MOV AL, N_TIMER
            OUT INT1, AL
            MOV AL, 10
            OUT COMP, AL
            MOV AL, 0
            OUT CONT, AL
            STI
LAZO:      CMP TERMINA, 0
            JZ LAZO
            INT 0

END

```

- d) **Desafío:** Modificar a) para que el primer mensaje se imprima luego de 1 segundo, el segundo luego de 2 segundos, el tercero luego de 3, y así sucesivamente, hasta que se espere 255 en el último mensaje, y el programa termine

```

EOI EQU 20H
IMR EQU 21H
INT1 EQU 25h

CONT EQU 10h
COMP EQU 11h

N_TIMER EQU 20

ORG 80
IP_F10 DW RUT_TIMER

ORG 1000H
TEXT0 DB "Vamos las interrupciones!"
FIN_TEXT0 DB ?
ESPERA DB 1

ORG 3000H
RUT_TIMER: PUSH AX
            PUSH BX
            MOV BX, OFFSET TEXT0

```



```

        MOV AL, OFFSET FIN_TEXTO - OFFSET TEXTO
        INT 7

            INC ESPERA
            MOV AL, ESPERA
            OUT COMP, AL
            CMP AL, 0
            JNZ SIGUE
            MOV AL, 0FFh
            OUT IMR, AL
SIGUE:   MOV AL, EOI
            OUT EOI, AL

        MOV AL, 0
        OUT CONT, AL
        POP BX
            POP AX
            IRET

    ORG 2000H
        CLI
        MOV AL, 0FDH
        OUT IMR, AL
        MOV AL, N_TIMER
        OUT INT1, AL
        MOV AL, ESPERA
        OUT COMP, AL
        MOV AL, 0
        OUT CONT, AL
        STI
    LAZO: CMP ESPERA, 0
            JNZ LAZO
            INT 0

    END

```

#### 6) F10 con Timer

- a) **Conteo regresivo** ★★: Escribir un programa que implemente un conteo regresivo a partir de un valor (de 1 a 9) 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. Por ejemplo, si se ingresa el valor 7, cuando se aprieta F10 debe mostrarse en pantalla “7 6 5 4 3 2 1 0” en los 7 segundos siguientes

**Pista:** Una vez leído el dígito se puede activar solamente la interrupción F10, y cuando se invoque la subrutina F10 activar el Timer para que a partir de ese momento se muestre el conteo regresivo.

```

    CONT EQU 10H
    COMP EQU 11H
    PIC   EQU 20H
    EOI   EQU 20H
    IMR   EQU 21H
    INT0  EQU 24h
    INT1  EQU 25H

    N_F10 EQU 10
    N_CLK EQU 15

    ORG 40

```

```
IP_F10    DW RUT_F10

        ORG 60
IP_CLK    DW    RUT_CLK

        ORG 1000H
CONTADOR  DB 39H
FINAL     DB 0

        ORG 3000H
RUT_CLK:  PUSH AX
          PUSH BX
          MOV BX, OFFSET CONTADOR
          MOV AL, 1
          INT 7
          CMP CONTADOR, 30h
          JNZ SIGUE
          MOV AL, 0FFH
          OUT IMR, AL
          MOV FINAL, 1
SIGUE:    DEC CONTADOR
          MOV AL, 0
          OUT CONT, AL
          MOV AL, EOI
          OUT EOI, AL
          POP BX
          POP AX
          IRET

        ORG 3200H
RUT_F10:  PUSH AX
          MOV AL, 0FDH
          OUT IMR, AL
          MOV AL, 0
          OUT CONT, AL
          MOV AL, 1
          OUT COMP, AL
          MOV AL, EOI
          OUT EOI, AL
          POP AX
          IRET

        ORG 3400H
LEE_NUMERO: MOV BX, OFFSET CONTADOR
REPITE:    INT 6
          MOV AL, CONTADOR
          CMP AL, 30H
          JS REPITE
          CMP AL, 39H
          JNS REPITE
          RET

        ORG 2000H
          CALL LEE_NUMERO
          CLI
          MOV AL, 0FEH
```

```

        OUT IMR, AL
        MOV AL, N_F10
        OUT INT0, AL
        MOV AL, N_CLK
        OUT INT1, AL
        STI
BUCLE:  CMP FINAL, 0
        JZ BUCLE
        INT 0

```

- b) **Conteo regresivo con pausa** ★★ Escribir un programa que cuente de forma regresiva desde 9 hasta 0, mostrando el tiempo en pantalla.. Luego de que termine de contar, si se presiona F10 debe comenzar de nuevo la cuenta desde 9 hasta 0. Además, si en algún momento se presiona la tecla F10 durante el conteo, el mismo debe pausarse, y reanudarse sólo cuando se presiona nuevamente F10.

**Pista:** Utilizar un flag de estado “PAUSA” que se controla mediante la tecla F10, y que la subrutina del timer puede consultar para saber si debe decrementar el contador y mostrar en pantalla.

```

CONT EQU 10H
COMP EQU 11H
PIC EQU 20H
EOI EQU 20H
IMR EQU 21H
INT0 EQU 24h
INT1 EQU 25H

N_F10 EQU 10
N_CLK EQU 15

    ORG 40
IP_F10 DW RUT_F10

    ORG 60
IP_CLK DW RUT_CLK

    ORG 1000H
CONTADOR DB 39H
PAUSA DB 0
TERMINO_CONTEO DB 0

    ORG 3000H
RUT_CLK: PUSH AX
          PUSH BX
          MOV BX, OFFSET CONTADOR
          MOV AL, 1
          INT 7
          CMP CONTADOR, 30h
          JNZ SIGUE
          MOV AL, 0FEH
          OUT IMR, AL
          MOV TERMINO_CONTEO, 1
SIGUE:   DEC CONTADOR
          MOV AL, 0
          OUT CONT, AL
          MOV AL, EOI
          OUT EOI, AL
          POP BX

```

```

                POP AX
                IRET

ORG 3200H
RUT_F10: PUSH AX
            CMP TERMINO_CONTEO, 1
            JNZ NO_REPETIR
            MOV TERMINO_CONTEO, 0
            MOV CONTADOR, 39H
            MOV AL, 0
            OUT CONT, AL
            MOV AL, 0FCh
            JMP CONT_F10

NO_REPETIR: CMP PAUSA, 0
            JZ PAUSAR
            MOV PAUSA, 0
            MOV AL, 0FCH
            JMP CONT_F10

PAUSAR: MOV PAUSA, 1
        MOV AL, 0
        OUT CONT, AL
        MOV AL, 0FEh
CONT_F10: OUT IMR, AL
        MOV AL, EOI
        OUT EOI, AL
        POP AX
        IRET

ORG 2000H
CLI
MOV AL, 0FCH
OUT IMR, AL
MOV AL, N_F10
OUT INT0, AL
MOV AL, N_CLK
OUT INT1, AL
MOV AL, 0
OUT CONT, AL
    MOV AL, 1
OUT COMP, AL
STI
BUCLE: JMP BUCLE

END

```

7) **Impresión con HANDSHAKE mediante interrupciones** ★★

- a) Escribir un programa que imprime "UNIVERSIDAD NACIONAL DE LA PLATA" en la impresora a través del HAND-SHAKE. El envío de los caracteres se realiza por **interrupciones** emitidas desde el HAND-SHAKE cada vez que detecta que la impresora se desocupa.

```

IMR    EQU 21H
EOI    EQU 20H
INT2   EQU 26H

HAND   EQU 40H
EST    EQU 41H

```

```
DATO EQU 40H
N_HND EQU 10

        ORG 40
        DW RUT_HND

        ORG 1000H
MSJ      DB "UNIVERSIDAD NACIONAL DE LA PLATA"
FIN      DB ?

        ORG 2000H
        MOV BX, OFFSET MSJ
        MOV CL, OFFSET FIN - OFFSET MSJ
        CLI
        MOV AL, 0FBH
        OUT IMR, AL
        MOV AL, N_HND
        OUT INT2, AL
        MOV AL, 80H
        OUT EST, AL
        STI
LAZO:    CMP CL, 0
        JNZ LAZO
        IN AL, EST
        AND AL, 7FH
        OUT EST, AL
        INT 0

        ORG 3000H
RUT_HND: PUSH AX
        MOV AL, [BX]
        OUT DATO, AL
        INC BX
        DEC CL
        JNZ FINAL
        MOV AL, 0FFH
        OUT IMR, AL
FINAL:   MOV AL, EOI
        OUT EOI, AL
        POP AX
        IRET

END
```

b) Probar el programa utilizando:

- 1) Una **velocidad de reloj** del 100%, y una **velocidad de impresión** del 25%
- 2) Una **velocidad de reloj** del 25%, y una **velocidad de impresión** del 100%

¿En qué caso sería mejor utilizar consulta de estado en lugar de interrupciones?

En el caso de una CPU lenta y una impresora rápida (caso 2) la consulta de estado resulta más conveniente. La complejidad que añaden las interrupciones no se justifica si la espera es poca.

- c) Modificar a) de modo que el string a imprimir se lea desde teclado. El string a leer tiene una longitud fija de 10 caracteres, y se lee de forma completa *antes* de comenzar la impresión. Tener en cuenta entonces que las interrupciones deben esperar a que se ingrese todo el string.

```
DATO EQU 40h
EST EQU 41h
IMR EQU 21h
INT2 EQU 26h
EOI EQU 20h

    ORG 40
    DW RUT_HAND

ORG 3000H
RUT_HAND: MOV AL, [BX]
           OUT DATO, AL
           INC BX
           DEC CL
           JNZ FIN
           IN AL, EST
           AND AL, 07FH
           OUT EST,AL
           MOV AL, 0FFh
           OUT IMR, AL
FIN:      MOV AL, EOI
           OUT EOI, AL
           IRET

    ORG 1000H
NUM_CAR DB 5
CAR      DB ?

    ORG 2000H
    ; PROGRAMA PRINCIPAL
    MOV BX, OFFSET CAR
    MOV CL, NUM_CAR
LAZO:    INT 6
         INC BX
         DEC CL
         JNZ LAZO
         MOV BX, OFFSET CAR
         MOV CL, NUM_CAR
         CLI
         MOV AL, 0FBh
         OUT IMR, AL
         MOV AL, 10
         OUT INT2, AL
         IN AL, EST
         OR AL, 080h
         OUT EST, AL
         STI

ESPERA:  CMP CL, 0
         JNZ ESPERA
         IN AL, EST
         AND AL, 07FH
         OUT EST,AL
         INT 0

END
```

- d) Modificar b) de modo que cuando se presione F10 se cancele la impresión. En tal caso, deben desactivarse las interrupciones del HANDSHAKE para evitar que se envíen más caracteres a imprimir.

```
IMR    EQU 21H
EOI    EQU 20H
INT0   EQU 24H
INT2   EQU 26H

HAND   EQU 40H
EST    EQU 41H
DAT0   EQU 40H
N_HND  EQU 10
N_F10  EQU 15

        ORG 40
        DW  RUT_HND

        ORG 60
        DW  RUT_F10

        ORG 1000H
MSJ     DB "UNIVERSIDAD NACIONAL DE LA PLATA"
FIN     DB ?
TERMINO DB 0

        ORG 2000H
MOV BX, OFFSET MSJ
MOV CL, OFFSET FIN - OFFSET MSJ
CLI
MOV AL, 0FAH
OUT IMR, AL
MOV AL, N_F10
OUT INT0, AL
MOV AL, N_HND
OUT INT2, AL
MOV AL, 80H
OUT EST, AL
STI
LAZO:   CMP TERMINO, 0
        JZ LAZO
        INT 0

        ORG 3000H
RUT_HND: PUSH AX
        MOV AL, [BX]
        OUT DAT0, AL
        INC BX
        DEC CL
        JNZ FINAL
        MOV AL, 0FFH
        OUT IMR, AL
        IN  AL, EST
        AND AL, 7FH
        OUT EST, AL
```

```
        MOV TERMINO, 1
FINAL:  MOV AL, EOI
        OUT EOI, AL
        POP AX
        IRET

        ORG 3200H
RUT_F10: PUSH AX
        MOV AL, 0FFH
        OUT IMR, AL
        IN AL, EST
        AND AL, 7FH
        OUT EST, AL
        MOV TERMINO, 1
        MOV AL, EOI
        OUT EOI, AL
        POP AX
        IRET
END
```



### Parte 3: Ejercicios de repaso para el parcial

- 1) **Imprimir mensaje leído, longitud arbitraria** ★. Escribir un programa que solicite ingresar caracteres por teclado y los almacene en la memoria. Al presionar la tecla F10, se envían a la impresora a través de la PIO. No es necesario mostrar los caracteres en la pantalla.

```
PA EQU 30h
PB EQU 31h
CA EQU 32h
CB EQU 33h

IMR EQU 21h
EOI EQU 20h
INT0 EQU 24h
N_F10 EQU 10

ORG 40
DW RUT_F10

ORG 1000h
FIN_LEER DB 0
LARGO DB 0
CADENA DB ?

ORG 3200h
RUT_F10: PUSH AX
        MOV AL, 0FFh
        OUT IMR, AL
        MOV FIN_LEER, 1
        MOV AL, EOI
        OUT EOI, AL
        POP AX
        IRET

ORG 3000h
LEE_CADENA: CLI
        MOV AL, 0FEh
        OUT IMR, AL
        MOV AL, N_F10
        OUT INT0, AL
        STI
        MOV BX, OFFSET CADENA
        MOV LARGO, 0
SIGUE: INT 6
        INC BX
        INC LARGO
        CMP FIN_LEER, 0
        JZ SIGUE
        RET

ORG 2000h
        CALL LEE_CADENA
        MOV AL, 0FDh ; strobe salida (0), busy entrada (1)
        OUT CA, AL
        MOV AL, 0 ; puerto de datos todo salida
```

```

    OUT CB, AL

    ; inicializo strobe en 0
    IN AL, PA
    AND AL, 11111101b
    OUT PA, AL

    MOV CL, LARGO
    MOV BX, OFFSET CADENA
    ; espero que busy=0
POLL:  IN AL, PA
    AND AL, 01h ; 1000 0000
    JNZ POLL
    ; se que busy es 0, mandar caracer
    MOV AL, [BX]
    OUT PB, AL
    ; mandar flanco ascendente de strobe
    IN AL, PA
    OR AL, 00000010b
    OUT PA, AL
    ; strobe en 0
    IN AL, PA
    AND AL, 11111101b
    OUTPA, AL

    INC BX
    DEC CL
    JNZ POLL
    INT 0

END

```

- 2) **F10 y Luces** ★★ Escribir un programa que al presionar F10 encienda todas las luces, y al presionarlo nuevamente las apague. El programa nunca termina.

```

PA EQU 30h
PB EQU 31h
CA EQU 32h
CB EQU 33h

IMR EQU 21h
EOI EQU 20h
INT0 EQU 24h
N_F10 EQU 10

ORG 40
DW RUT_F10

ORG 1000h
ESTADO DB 0

ORG 3200h
RUT_F10: PUSH AX
    XOR ESTADO, 0FFh
    MOV AL, ESTADO
    OUT PB, AL
    MOV AL, EOI

```

```

        OUT EOI, AL
        POP AX
        IRET

ORG 2000h
        CLI
        MOV AL, 0 ; puerto de datos todo salida
        OUT CB, AL
        OUT PB, AL
        MOV AL, 0FEh
        OUT IMR, AL
        MOV AL, N_F10
        OUT INT0, AL
        STI
LOOP: JMP LOOP
END

```

- 3) **Ruleta, F10 y azar** ★★ Escribir un programa que permita seleccionar un dígito al azar para jugar a la ruleta. Para eso, el programa principal debe iterar continuamente, cambiando un valor de un registro desde el '0' hasta el '9' (y volviendo al '0' luego del '9'). Cuando se presiona F10, la letra queda "seleccionada" en base al valor del registro, y debe mostrarse de inmediato en la pantalla. Luego el programa termina.

```

IMR EQU 21h
EOI EQU 20h
INT0 EQU 24h
N_F10 EQU 10

ORG 40
DW RUT_F10

ORG 1000h
FINALIZADO DB 0
NUM DB ?

ORG 3200h
RUT_F10: PUSH AX
        MOV FINALIZADO, 1
        MOV AL, 0FFh
        OUT IMR, AL
        MOV AL, EOI
        OUT EOI, AL
        POP AX
        IRET

ORG 2000h
        CLI
        MOV AL, 0FEh
        OUT IMR, AL
        MOV AL, N_F10
        OUT INT0, AL
        STI
        MOV DL, '0'
LOOP:   CMP FINALIZADO, 1
        JZ ELEGIDO
        INC DL

```

```

        CMP DL, '9'
        JNZ LOOP
        MOV DL, '0'
        JMP LOOP
ELEGIDO:  MOV BX, OFFSET NUM
          MOV [BX], DL
          MOV AL, 1
          INT 7
          INT 0
END

```

- 4) **F10, lectura e Impresora con PIO** ★★★★★ Escribir un programa que debe mostrar en pantalla el mensaje “PRESIONE F10 PARA COMENZAR” y una vez que el usuario presiona F10, leer de teclado un mensaje de 10 caracteres. Este mensaje debe luego ser enviado a través de la pila a una subrutina para imprimirse en la impresora mediante el PIO. La configuración del PIO también debe hacerse en una subrutina aparte.

```

PA EQU 30h
PB EQU 31h
CA EQU 32h
CB EQU 33h

EOI EQU 20h
IMR EQU 21h
INT0 EQU 24h

N_F10 EQU 10

ORG 40
DW RUT_F10

ORG 1000h
MSJ_EMPEZAR DB "PRESIONE F10 PARA COMENZAR"
FIN_MSJ_EMPEZAR DB ?
EMPEZAR DB 0
MENSAJE DB ?

ORG 3000h
CONFIG_PIO: MOV AL, 0FDh
            OUT CA, AL
            MOV AL, 0
            OUT CB, AL
            ; inicializo strobe en 0
            IN AL, PA
            AND AL, 1111101b
            OUT PA, AL
            RET

ORG 3200h
RUT_F10:  PUSH AX
          MOV EMPEZAR, 1
          MOV AL, EOI
          OUT EOI, AL
          POP AX
          IRET

```

```
ORG 3400h
LEER_CADENA: INT 6
             INC BX
             DEC CL
             JNZ LEER_CADENA
             RET

ORG 3600h
IMPRIMIR:    MOV BX, SP
             ADD BX, 2
             MOV CL, [BX]
             ADD BX, 2
             MOV BX, [BX]

POLL:        IN AL, PA
             AND AL, 01h ; 1000 0000
             JNZ POLL

             MOV AL,[BX]
             OUT PB, AL

             ; mandar flanco ascendente de strobe
             IN  AL, PA
             OR  AL, 00000010b
             OUT PA, AL
             ; strobe en 0
             IN AL, PA
             AND AL, 11111101b
             OUT PA, AL

             INC BX
             DEC CL
             JNZ POLL
             RET

ORG 2000h
MOV BX, OFFSET MSJ_EMPEZAR
MOV AL, OFFSET FIN_MSJ_EMPEZAR - OFFSET MSJ_EMPEZAR
INT 7
CLI
MOV AL, 0FEh
OUT IMR, AL
MOV AL, N_F10
OUT INT0, AL
STI
ESPERA:      CMP EMPEZAR, 0
             JZ  ESPERA

             CALL CONFIG_PIO

             MOV BX, OFFSET MENSAJE
             MOV CL, 10
             CALL LEER_CADENA

             MOV BX, OFFSET MENSAJE
```

```

        PUSH BX
        MOV CL, 10
        PUSH CX
        CALL IMPRIMIR
        POP CX
        POP AX
        INT 0

END

```

- 5) **Timer y Luces** ★★ Escribir un programa que debe leer el estado de las llaves y enviarlo a una subrutina. La subrutina debe llamarse **LUCES12** y recibe el estado por referencia a través de la pila. Si el estado indica que la llave 7 (la de más a la izquierda) está prendida entonces debe encender todas las luces por 12 segundos y luego apagar todas las luces. En caso contrario no debe hacer nada.

```

PA EQU 30h
PB EQU 31h
CA EQU 32h
CB EQU 33h

ORG 1000h
ESTADO_LUCES DB ?

ORG 3000h
LUCES12:    MOV BX, SP
            ADD BX, 2
            MOV BX, [BX]

            MOV AL, [BX]
            AND AL, 080h
            JZ FIN_LUCES12
            MOV AL, 0FFh
            OUT PB, AL
FIN_LUCES12: RET

ORG 2000h
            MOV AL, 0FFh
            OUT CA, AL
            MOV AL, 0
            OUT CB, AL

            IN AL, PA
            MOV ESTADO_LUCES, AL
            MOV AX, OFFSET ESTADO_LUCES
            PUSH AX
            CALL LUCES12
            POP AX
            INT 0

END

```

- 6) **F10, impresora con HS y dígitos** ★★★ Escribir un programa para imprimir en la impresora un conteo regresivo en base a un dígito ingresado por teclado. Por ejemplo, si el usuario ingresa “3”, se debe imprimir “3 2 1 0”. El programa comienza leyendo un carácter dígito con la subrutina **LEER\_DIGITO**. Luego espera a que el usuario presione F10 para comenzar la impresión, y llama a la subrutina **DESCENDER** que la implementa.
- LEER\_DIGITO** muestra en pantalla el mensaje “INGRESE UN NUMERO DEL 1 AL 9:” y lee un carácter de teclado. Si el carácter ingresado no corresponde al número solicitado, se debe volver a leer un carácter hasta que el usuario ingrese efectivamente un número del 1 al 9.

**DESCENDER** recibe el dígito por valor y por la pila, y envía a la impresora en forma descendente desde el número ingresado hasta el "0", utilizando el Handshake por Consulta de Estado.

```
EOI    EQU 20H
IMR    EQU 21H
INT0   EQU 24h

EST    EQU 41H
DATO   EQU 40H

N_F10  EQU 10

ORG 40
IP_F10  DW RUT_F10

ORG 1000H
CONTADOR      DB 39H
SEGUIR_ESPERANDO DB 0
MSJ_INGRESO    DB "INGRESE UN NUMERO DEL 1 AL 9:"
FIN_MSJ        DB ?

ORG 3200H
    RUT_F10: PUSH AX
            MOV SEGUIR_ESPERANDO, 0
            MOV AL, EOI
            OUT EOI, AL
            POP AX
            IRET

ORG 3000H
LEE_DIGITO: MOV BX, OFFSET MSJ_INGRESO
            MOV AL, OFFSET FIN_MSJ - OFFSET MSJ_INGRESO
            INT 7
            MOV BX, OFFSET CONTADOR
REPITE:     INT 6
            MOV DL, CONTADOR
            CMP DL, 31H
            JS REPITE
            CMP DL, 39H
            JNS REPITE
            RET

ORG 3100H
DESCENDER: MOV BX, SP
            ADD BX, 2
            MOV DL, [BX]
POLL:      IN AL, EST
            AND AL, 1
            JNZ POLL
            MOV AL, DL
            OUT DATO, AL
            INC BX
            DEC DL
            CMP DL, 2Fh
            JNZ POLL
```

```

                RET

    ORG 2000H
    IN AL, EST
    AND AL, 7fh
    OUT EST, AL
    CALL LEE_DIGITO
    CLI
    MOV AL, 0FEH
    OUT IMR, AL
    MOV AL, N_F10
    OUT INT0, AL
    STI
    MOV SEGUIR_ESPERANDO, 1
ESPERA: CMP SEGUIR_ESPERANDO, 1
    JZ ESPERA
    PUSH DX
    CALL DESCENDER
    POP DX
    INT 0
END

```

- 7) **Dispositivo nuevo con PIO** ★★★★★ Escribir un programa para VonSim que envíe la cadena de caracteres "Hola!" a un dispositivo nuevo, conectado a los 8 bits del puerto PB. Este dispositivo recibe la cadena de a un carácter a la vez, esperando 5 segundos entre cada carácter. Para indicar al dispositivo que finalizó el envío de datos, se debe enviar el código ascii 0 luego de enviar todos los caracteres del string. El programa debe finalizar cuando se han enviado todos los caracteres de la cadena, o cuando se presiona la tecla F10, cancelando el envío cancelando el envío de los caracteres que restan.

**Ejemplo** para enviar la cadena "ASDF": Envío de la "A" → 5 segundos de espera → Envío de la "S" → 5 segundos de espera → Envío de la "D" → 5 segundos de espera → Envío de la "F" → 5 segundos de espera → Envío de 0

```

PB EQU 31h
CB EQU 33h

EOI EQU 20h
IMR EQU 21h
INT0 EQU 24h
INT1 EQU 25h

CONT EQU 10h
COMP EQU 11h

N_CLK EQU 10
N_F10 EQU 15

ORG 40
DW RUT_CLK

ORG 60
DW RUT_F10

ORG 1000H
CADENA DB "Hola!"
FIN_CADENA DB ?

```



```
TIEMPO DB 0
TERMINAR DB 0

ORG 3000h
RUT_F10: PUSH AX
        MOV TERMINAR, 1
        MOV AL, EOI
        OUT EOI, AL
        POP AX
        IRET

ORG 3200h
RUT_CLK: PUSH AX
        MOV TIEMPO, 1
        MOV AL, 0
        OUT CONT, AL
        MOV AL, EOI
        OUT EOI, AL
        POP AX
        IRET

ORG 3400h
ESPERAR: CLI
        MOV AL, 0FCh
        OUT IMR, AL
        MOV AL, 0
        OUT CONT, AL
        MOV TIEMPO, 0
        STI
LOOP:   CMP TERMINAR, 1
        JZ SALIR
        CMP TIEMPO, 1
        JNZ LOOP

SALIR:  CLI
        MOV AL, 0FEh
        OUT IMR, AL
        STI
        RET

ORG 2000h
        CLI
        MOV AL, 0FCh
        OUT IMR, AL
        MOV AL, N_F10
        OUT INT0, AL
        MOV AL, N_CLK
        OUT INT1, AL
        MOV AL, 0
        OUT CONT, AL
        MOV AL, 5
        OUT COMP, AL
        MOV AL, 0
        OUT CB, AL
        STI
        MOV BX, OFFSET CADENA
```

```
MOV CL, OFFSET FIN_CADENA - OFFSET CADENA
```

```
SIGUE:    CMP TERMINAR, 1
```

```
        JZ FINALIZAR
```

```
        MOV AL, [BX]
```

```
        OUT PB, AL
```

```
        CALL ESPERAR
```

```
        INC BX
```

```
        DEC CL
```

```
        JNZ SIGUE
```

```
FINALIZAR: MOV AL, 0FFh
```

```
        OUT IMR, AL
```

```
        INT 0
```

```
END
```