

PRÁCTICA 3

Interrupciones por Hardware

Objetivos: Comprender la utilidad de las interrupciones por software y 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. Escribir programas en el lenguaje assembly del simulador VonSim y el MSX88. Ejecutarlos y verificar los resultados, analizando el flujo de información entre los distintos componentes del sistema.

Parte 1: Interrupciones por Hardware

- 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	interrupción atendida o no	No aplica
21h	IMR	int. habilitadas y deshabilitadas	1111 1101 (0 habilitada, 1 deshabilitada)
22h	IRR	q disp. solicita interrumpir	0000 0100
23h	ISR	q disp. esta siendo atendido	0000 0011
24h	INT 0	F10	15
25h	INT 1	Timer	10
26h	INT 2	Handshake	25

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

Dispositivos	IMR	INT 0	INT 1	INT 2	Dirección Vector
F10	1111 1110	5	No importa	No importa	20
F10	1111 1110	8	No importa	No importa	32
Timer	1111 1101	10	No importa	No importa	40
HANDSHAKE	1111 1011	5	No importa	No importa	20
Ninguno	1111 1111	-	-	-	-
F10 y Timer	1111 1100	10	20	No importa	40 y 80
Handshake y Timer	1111 1001	4	8	No importa	16 y 32
Handshake, Timer y F10	1111 1000	5	10	15	20,40 y 60

- 3) **Tecla F10** 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 loop infinito. Completar las instrucciones faltantes para que el programa funcione correctamente.

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

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

- ¿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?
 - ¿Por qué se usa el valor 0FEH en el programa?
 - ¿Qué instrucciones le indican al PIC que la interrupción ha terminado?
 - ¿En qué dirección se encuentra la subrutina que atiende las interrupciones del F10?
- 4) **Tecla F10 con pantalla y terminación**
- Escribir un programa que muestre en pantalla “Vamos las interrupciones!” cada vez que se presiona la tecla F10. El programa nunca termina.
 - Idem A pero ahora el programa termina luego de que se muestra 5 veces el mensaje.

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.

- 5) **Timer, tres usos: periódico infinito, periódico finito, y única vez**
- Escribir un programa que muestre el mensaje “Vamos las interrupciones!” una vez cada 2 segundos y no termine nunca.
 - Modificar a) para que termine a los 10 segundos.
 - Modificar a) para que solo imprima una vez, a los 10 segundos, y luego termine.
 - 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,
- 6) **F10 con Timer**
- 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 apreta F10 debe mostrarse en pantalla “7 6 5 4 3 2 1 0” en los 7 segundos siguientes

Pista: Usar un flag de estado “INICIO” que empieza con el valor 0 (Falso) y se pone en 1111 1111 (Verdadero) cuando se presiona F10

- 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.

- 7) **HANDSHAKE con interrupciones**
- 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.
 - 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.
 - 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.

Parte 2: Ejercicios de repaso para el parcial

- 1) **F10 y Luces** Escribir un programa que al presionar F10 encienda todas las luces, y al presionarlo nuevamente las apague. El programa nunca termina
- 2) **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
- 3) **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.
- 4) **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.
- 5) **Luces, llaves y opciones** Escribir un programa que deberá utilizar las luces y llaves, con estos tres casos:
 - A. Verificar si todas llaves están apagadas. Si es así, mostrar en pantalla el mensaje "Fin de programa" y finalizar el mismo. Caso contrario, hacer tanto B como C.
 - B. Actualizar las luces a su estado opuesto. Por ejemplo, si las llaves están en el estado "00011010" las luces tendrán el estado "11100101".
 - C. Si la primera llave (la del bit menos significativo) está prendida, mostrar en pantalla el mensaje "Arquitectura de Computadoras: ACTIVADA". Las funciones "A", "B" y "C" deben implementarse utilizando tres subrutinas independientes. La subrutina A debe devolver 1 si hay que finalizar el programa y 0 de lo contrario.
- 6) **Llaves y mensajes**
 - a) Escribir un programa que continuamente verifique el estado de las llaves. Si están prendidas la primera y la última llave, y el resto están apagadas (patrón 10000001), se debe mostrar en pantalla el mensaje "ACTIVADO". En caso contrario, no se debe mostrar nada.
 - b) Modificar a) para que el mensaje se imprima una sola vez cada vez que detecte ese patrón de bits. Por ejemplo, si el programa lee la siguiente secuencia de patrones: 00010101 → 10010000 → **10000001** → **10000001** → **10000001** → 10010001 → **10000001** → **10000001** → 10010101 → 01110001
Entonces solo deberá imprimir "ACTIVADO" dos veces.
- 7) **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.
- 8) **CriptoLlaves (Llaves, Luces, Timer):** Escriba un programa de VonSim que permita jugar al CriptoLlaves. El usuario debe adivinar un patrón secreto de 8 bits que está almacenado en la memoria del programa. Para ello, debe manipular las llaves hasta que el patrón de bits de las llaves sea exactamente igual al del patrón secreto. Como ayuda para el usuario, si el estado de una llave acierta al bit correspondiente, el programa debe prender el led correspondiente. Por ejemplo, si el patrón es 0100 0101 y las llaves están en el estado 1110 0100, deben prenderse las luces de los bits 1, 2, 3, 4 y 6. Como no acertó a todos los bits, el usuario no ha adivinado el patrón y debe continuar jugando. El programa termina cuando el usuario acierta todos los bits del patrón, o cuando pasen 9 segundos. Para que el usuario sepa cuánto tiempo le queda, debe mostrarse en pantalla una cuenta regresiva desde '9' a '0'. En caso de que el usuario haya acertado a todos los bits, mostrar el mensaje "GANASTE". En caso de que el usuario no haya acertado a ningún bit, mostrar el mensaje "PERDISTE". Para esta funcionalidad, implemente una subrutina llamada **INFORMAR** que reciba por registro y valor el byte secreto y el byte de estado de las llaves. Debe utilizar también una subrutina para configurar el PIO, y otra para configurar el **TIMER**.
- 9) **Dispositivo nuevo con PIO** Escribir un programa para VonSim que envíe la cadena de caracteres "Universidad Nacional de La Plata" a un dispositivo nuevo, conectado a los 8 bits del puerto PB. Este dispositivo recibe la cadena de a un carácter a la vez. Para que el dispositivo reconozca que se va a enviar un dato, antes de enviar un carácter debe enviar el valor 0. 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 0 → Envío de la "A" → Envío de 0 → Envío de la "S" → Envío de 0 → Envío de la "D" → Envío de 0 → Envío de la "F"

Pista: para comunicarse con el dispositivo no es necesario realizar una consulta de estado ni hacerlo mediante interrupciones, el protocolo solo requiere que envíe un 0 y un carácter de forma alternada.