

PRACTICA 3

Entrada/Salida y DMA

Objetivos: Comprender la comunicación entre el microprocesador y los periféricos externos (luces, microconmutadores e impresora). Configurar la interfaz de entrada/salida (PIO) y el dispositivo de handshaking (HAND-SHAKE) para el intercambio de información entre el microprocesador y el mundo exterior. Comprender el funcionamiento del Controlador de Acceso Directo a Memoria (CDMA) incluido en el simulador MSX88. Configurarlos para la transferencia de datos memoria-memoria y memoria-periférico en modo bloque y bajo demanda. Escribir programas en el lenguaje assembly del simulador MSX88. Ejecutarlos y verificar los resultados, analizando el flujo de información entre los distintos componentes del sistema.

- 1) Encendido/apagado de las luces (periférico de salida) mediante la barra de microconmutadores (periférico de entrada), ambos comunicados con el microprocesador a través de los puertos paralelos de la PIO. Implementar un programa en el lenguaje assembly del simulador MSX88 que configure la PIO para leer el estado de los microconmutadores y escribirlo en la barra de luces. El programa se debe ejecutar bajo la configuración P1 C0 del simulador. Los microconmutadores se manejan con las teclas 0-7.

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

ORG 2000H
MOV AL, 0FFH ; PA entradas (Microconmutadores)
OUT CA, AL
MOV AL, 0    ; PB salidas (Luces)
OUT CB, AL
POLL: IN  AL, PA
      OUT PB, AL
      JMP POLL
      END
```

- 2) Encendido/apagado sincronizado de las luces. Implementar un contador que incremente la cuenta en uno una vez por segundo y la visualice a través de las luces conectadas a uno de los puertos paralelos del simulador. Ejecutar en configuración P1 C0.

PIC	EQU 20H		
TIMER	EQU 10H		
PIO	EQU 30H		
N_CLK	EQU 10		
	ORG 40		
IP_CLK	DW RUT_CLK		
	ORG 1000H		
INICIO	DB 0		
	ORG 2000H		ORG 3000H
	CLI	RUT_CLK:	INC INICIO
	MOV AL, 0FDH		CMP INICIO, 0FFH
	OUT PIC+1, AL		JNZ LUCES
	MOV AL, N_CLK		MOV INICIO, 0
	OUT PIC+5, AL	LUCES:	MOV AL, INICIO
	MOV AL, 1		OUT PIO+1, AL
	OUT TIMER+1, AL		MOV AL, 0
	MOV AL, 0		OUT TIMER, AL
	OUT PIO+3, AL		MOV AL, 20H
	OUT PIO+1, AL		OUT PIC, AL
	OUT TIMER, AL		IRET
	STI		END
LAZO:	JMP LAZO		

- 3) Escribir un programa que encienda una luz a la vez, de las ocho conectadas al puerto paralelo del microprocesador a través de la PIO, en el siguiente orden: 0-1-2-3-4-5-6-7-6-5-4-3-2-1-0-1-2-3-4-5-6-7-6-5-4-3-2-1-0-1-... Cada luz debe estar encendida durante un segundo. Ejecutar en la configuración P1 C0 del simulador.

Uso de la impresora a través de la PIO

- 4) . Escribir un programa que envíe datos a la impresora a través de la PIO. La PIO debe cumplir las funciones de temporización que requiere la impresora para la comunicación. Ejecutar en configuración P1 C1 del simulador y presionar F5 para mostrar la salida en papel. El papel se puede blanquear ingresando el comando BI.

```
PIO      EQU 30H

        ORG 1000H
MSJ      DB "ARQUITECTURA DE      "
        DB "COMPUTADORAS"
FIN      DB ?

        ORG 2000H
MOV AL, 0FDH                ; INICIALIZACION PIO PARA IMPRESORA
OUT PIO+2, AL
MOV AL, 0
OUT PIO+3, AL
IN AL, PIO
AND AL, 0FDH
OUT PIO, AL                ; FIN INICIALIZACION
MOV BX, OFFSET MSJ
MOV CL, OFFSET FIN-OFFSET MSJ
POLL:   IN AL, PIO
AND AL, 1
JNZ POLL
MOV AL, [BX]
OUT PIO+1, AL
IN AL, PIO                ; PULSO 'STROBE'
OR AL, 02H
OUT PIO, AL
IN AL, PIO
AND AL, 0FDH
OUT PIO, AL                ; FIN PULSO
INC BX
DEC CL
JNZ POLL
INT 0
END
```

- 5) Escribir un programa que solicite el ingreso de cinco caracteres por teclado y los envíe de a uno por vez a la impresora a través de la PIO a medida que se van ingresando. No es necesario mostrar los caracteres en la pantalla. Ejecutar en configuración P1 C1.

<pre>PIO EQU 30H ORG 1000H NUM_CAR DB 5 CAR DB ? ; SUBROUTINA DE INICIALIZACION ; PIO PARA IMPRESORA ORG 3000H INI_IMP: MOV AL, 0FDH OUT PIO+2, AL MOV AL, 0 OUT PIO+3, AL IN AL, PIO AND AL, 0FDH OUT PIO, AL RET</pre>	<pre>; SUBROUTINA DE GENERACIÓN ; DE PULSO 'STROBE' ORG 4000H PULSO: IN AL, PIO OR AL, 02H OUT PIO, AL IN AL, PIO AND AL, 0FDH OUT PIO, AL RET</pre>
---	---

```

; PROGRAMA PRINCIPAL
      ORG 2000H
      PUSH AX
      CALL INI_IMP
      POP AX
      MOV BX, OFFSET CAR
      MOV CL, NUM_CAR
LAZO:  INT 6
POLL:  IN  AL, PIO
      AND AL, 1
      JNZ POLL
      MOV AL, [BX]
      OUT PIO+1, AL
      PUSH AX
      CALL PULSO
      POP AX
      DEC CL
      JNZ LAZO
      INT 0
      END

```

- 6) Escribir un programa que solicite ingresar caracteres por teclado y que recién al presionar la tecla F10 los envíe a la impresora a través de la PIO. No es necesario mostrar los caracteres en la pantalla. Ejecutar en configuración P1 C1 del simulador.

Uso de la impresora a través del dispositivo de hand-shaking por consulta de estado.

- 7) Escribir un programa que envíe datos a la impresora a través del HAND-SHAKE. La comunicación se debe establecer por consulta de estado (polling). Ejecutar en configuración P1 C2.

```

HAND  EQU 40H
      ORG 1000H
MSJ    DB "FACULTAD DE          "
      DB "INFORMATICA"
FIN     DB ?

      ORG 2000H
      IN  AL, HAND+1
      AND AL, 7FH
      OUT HAND+1, AL
      MOV BX, OFFSET MSJ
      MOV CL, OFFSET FIN-OFFSET MSJ
POLL:  IN  AL, HAND+1
      AND AL, 1
      JNZ POLL
      MOV AL, [BX]
      OUT HAND, AL
      INC BX
      DEC CL
      JNZ POLL
      INT 0
      END

```

Uso de la impresora a través del dispositivo de hand-shaking por interrupción.

- 8) Escribir un programa que envíe datos a la impresora a través del HAND-SHAKE. La comunicación se debe establecer por interrupciones emitidas desde el HAND-SHAKE cada vez que la impresora se desocupa. Ejecutar en configuración P1 C2.

PIC	EQU 20H		
HAND	EQU 40H		
N_HND	EQU 10		
IP_HND	DW RUT_HND	ORG 40	ORG 1000H
		MSJ	DB "UNIVERSIDAD "
			DB "NACIONAL DE LA PLATA"
		FIN	DB ?

<pre> ORG 3000H RUT_HND: PUSH AX MOV AL, [BX] OUT HAND, AL INC BX DEC CL MOV AL, 20H OUT PIC, AL POP AX IRET </pre>	<pre> ORG 2000H MOV BX, OFFSET MSJ MOV CL, OFFSET FIN-OFFSET MSJ CLI MOV AL, 0FBH OUT PIC+1, AL MOV AL, N_HND OUT PIC+6, AL MOV AL, 80H OUT HAND+1, AL STI LAZO: CMP CL, 0 JNZ LAZO IN AL, HAND+1 AND AL, 7FH OUT HAND+1, AL INT 0 END </pre>
--	---

- 9) Escribir un programa que solicite el ingreso de cinco caracteres por teclado y los almacene en memoria. Una vez ingresados, que los envíe a la impresora a través del HAND-SHAKE, en primer lugar tal cual fueron ingresados y a continuación en sentido inverso. Implementar dos versiones, una por consulta de estado y otra por interrupción, en lo que se refiere a la comunicación entre el HAND-SHAKE y el microprocesador.

10) **DMA.** Transferencia de datos memoria-memoria.

Escribir un programa que copie una cadena de caracteres almacenada a partir de la dirección 1000H en otra parte de la memoria, utilizando el CDMA en modo de transferencia por bloque. La cadena original se debe mostrar en la pantalla de comandos antes de la transferencia. Una vez finalizada, se debe visualizar en la pantalla la cadena copiada para verificar el resultado de la operación. Ejecutar el programa en la configuración P1 C3.

<pre> PIC EQU 20H DMA EQU 50H N_DMA EQU 20 ORG 80 IP_DMA DW RUT_DMA ORG 1000H MSJ DB "FACULTAD DE" DB " INFORMATICA" FIN DB ? NCHAR DB ? ORG 1500H COPIA DB ? ; rutina aten interrupción del CDMA ORG 3000H RUT_DMA: MOV AL, 0FFH ;inhabilita.. OUT PIC+1, AL ;interrupc de PIC MOV BX, OFFSET COPIA MOV AL, NCHAR INT 7 MOV AL, 20H OUT PIC, AL ; EOI IRET </pre>	<pre> ORG 2000H CLI MOV AL, N_DMA OUT PIC+7, AL ; reg INT3 de PIC MOV AX, OFFSET MSJ OUT DMA, AL ; dir comienzo .. MOV AL, AH ; del bloque .. OUT DMA+1, AL ; a transferir MOV AX, OFFSET FIN-OFFSET MSJ OUT DMA+2, AL ; cantidad .. MOV AL, AH ; a .. OUT DMA+3, AL ; transferir MOV AX, OFFSET COPIA OUT DMA+4, AL ; dir destino .. MOV AL, AH ; del .. OUT DMA+5, AL ; bloque MOV AL, 0AH ; CDMA en transfer.. OUT DMA+6, AL ; mem-mem por bloque MOV AL, 0F7H OUT PIC+1, AL ; habilita INT3 STI MOV BX, OFFSET MSJ MOV AL, OFFSET FIN-OFFSET MSJ MOV NCHAR, AL INT 7 ; mensaje original MOV AL, 7H OUT DMA+7, AL ; arranque Transfer INT 0 END </pre>
---	---

Cuestionario:

- Analizar minuciosamente cada línea del programa anterior.
- Explicar qué función cumple cada registro del CDMA e indicar su dirección.

- 5