
Arquitectura 2023

Explicación 3

Problema de Entrada/Salida

- Gran variedad de periféricos con varios métodos de operación.
 - Transmisión de diferentes cantidades de datos a diferentes velocidades.
 - Usan diferentes formatos de datos y tamaños de palabra.
 - Todos más lentos que la CPU y la RAM.
 - Necesidad de módulos de E/S (con alguna “inteligencia”)
 - Módulo E/S: Interfaz entre el procesador y la memoria (bus) y los periféricos.
-

Funciones de un módulo de E/S

- Control y temporización de uno o más dispositivos externos
 - Comunicación con la CPU y la memoria
 - Comunicación con los dispositivos (periféricos)
 - Almacenamiento temporal (*buffering*) de datos
 - Detección de errores
-

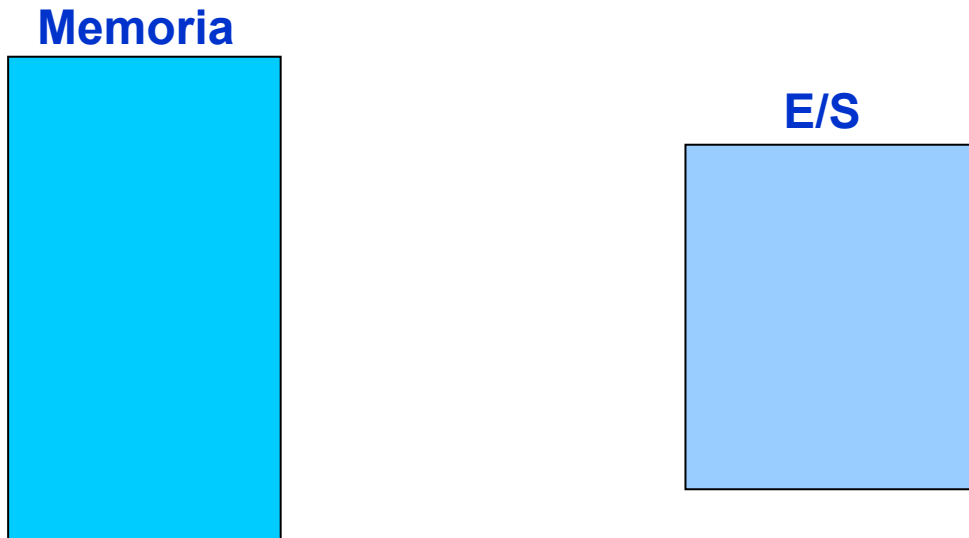
Entrada/Salida

Hay 2 métodos para plantear la E/S:

- ❑ E/S aislada
- ❑ E/S mapeada en memoria

E/S aislada

Las direcciones de E/S (llamadas **puertos**) están “separadas” de la memoria principal del sistema, en un espacio de direcciones distinto.



E/S aislada II

- **Ventaja:** se puede utilizar todo el espacio de direcciones de memoria.
 - **Desventaja:** para transferir datos entre el procesador y los dispositivos de E/S se deben utilizar instrucciones especiales como **IN** y **OUT**.
-

E/S mapeada en memoria

- Los módulos de E/S están mapeados en direcciones de memoria.
 - No se distingue una posición de E/S de una posición de memoria.
-

E/S mapeada en memoria II

- **Ventaja:** se puede usar el conjunto de instrucciones del procesador, porque todas las posiciones son tomadas como direcciones de memoria. No hay instrucciones específicas como **IN** y **OUT**.
- **Desventaja:** los módulos de E/S ocupan parte de la memoria

E/S en SX88

- Utiliza el modelo de memoria de E/S aislada.

Instrucciones:

- **IN AL, puerto** → *Lee un byte de la dirección de E/S llamada puerto.*
- **IN AL, DX** → *Lee un byte de la dirección de E/S almacenada en DX.*
- **OUT puerto, AL** → *Escribe un byte de AL en la dirección de E/S llamada puerto.*
- **OUT DX, AL** → *Escribe un byte de AL en la dirección de E/S contenida en DX.*

PIO (Puerto paralelo de E/S)

- Son 2 puertos paralelos de 8 bits: A y B.
- Se puede programar cada bit por separado como entrada ó salida.

Posee 4 registros internos de 8 bits:

- 2 de datos, PA y PB.
- 2 de control CA y CB para programar los bits de PA y PB.

| | |
|------------------------|--------|
| PIO | |
| Datos { | 30H PA |
| | 31H PB |
| Configuración { | 32H CA |
| | 33H CB |

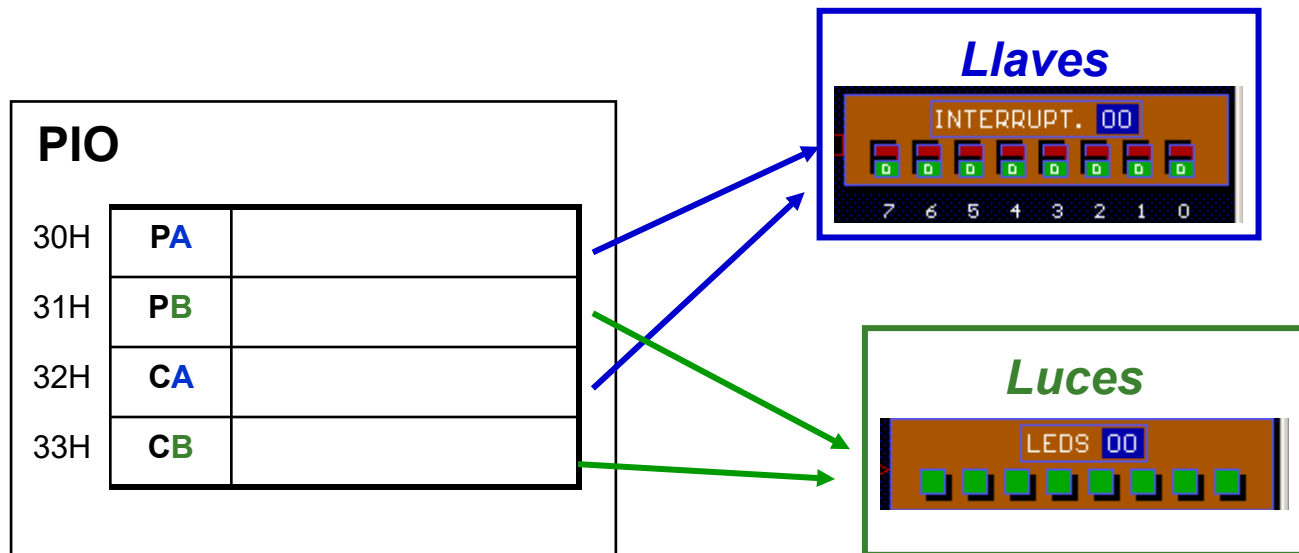
-Un bit en **0** en CA/CB
selecciona como salida a la línea
correspondiente en PA/PB

-Un bit en **1** en CA/CB
selecciona como entrada a la
línea correspondiente en PA/PB

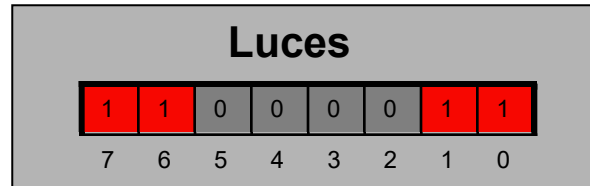
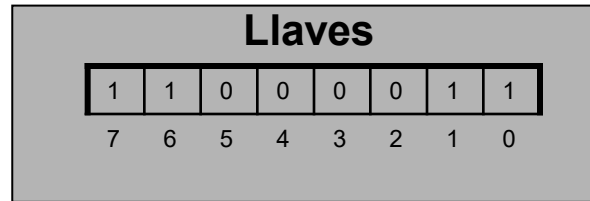
Ej. 1

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.



Ej. 1



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

ORG 2000H

MOV AL, 0FFH
OUT CA, AL
MOV AL, 0
OUT CB, AL

POLL: IN AL, PA
OUT PB, AL
JMP POLL

END

PIO

| | | |
|-----|----|-----------------|
| 30H | PA | C3H = 1100 0011 |
| 31H | PB | C3H = 1100 0011 |
| 32H | CA | FFH = 1111 1111 |
| 33H | CB | 0 = 0000 0000 |

AX

C3H

*PA (Microconmutadores) como **entradas** (1)*

*PB (Luces) como **salidas** (0)*

*Lee el estado de las llaves y
enciende las luces correspondientes*

Ej. 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.

Ej. 2

```
PIC EQU 20H
TIMER EQU 10H
PIO EQU 30H
N_CLK EQU 10
```

```
ORG 40
IP_CLK DW RUT_CLK
```

```
ORG 1000H
INICIO DB 2
```

```
ORG 2000H
```

```
CLI
```

```
MOV AL, 0FDH
OUT PIC+1, AL
MOV AL, N_CLK
OUT PIC+5, AL
```

```
MOV AL, 1
OUT TIMER+1, AL
```

```
MOV AL, 0
```

```
OUT PIO+3, AL
OUT PIO+1, AL
OUT TIMER, AL
```

```
STI
```

```
LAZO: JMP LAZO
```

```
ORG 3000H
```

```
RUT_CLK: INC INICIO
          CMP INICIO, 0FFH
```

```
          JNZ LUCES
          MOV INICIO, 0
```

```
LUCES:   MOV AL, INICIO
          OUT PIO+1, AL
```

```
          MOV AL, 0
          OUT TIMER, AL
```

```
          MOV AL, 20H
          OUT PIC, AL
```

```
          IRET
```

```
END
```

Si estan todas
prendidas las apago

Enciendo las luces
correspondientes

Inicializo el timer

AX

20H

PIC

| | | |
|------|-----------------|-----|
| EOI | 20H | 20H |
| IMR | FDH = 1111 1101 | 21H |
| IRR | | 22H |
| ISR | | 23H |
| INT0 | | 24H |
| INT1 | 10 | 25H |
| INT2 | | 26H |
| INT3 | | 27H |

TIMER

Cont: 10H

Comp: 11H

1

1

PIO

| | | |
|-----|----|---------------|
| 30H | PA | |
| 31H | PB | 2 = 0000 0010 |
| 32H | CA | |
| 33H | CB | 0 = 0000 0000 |

Luces

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

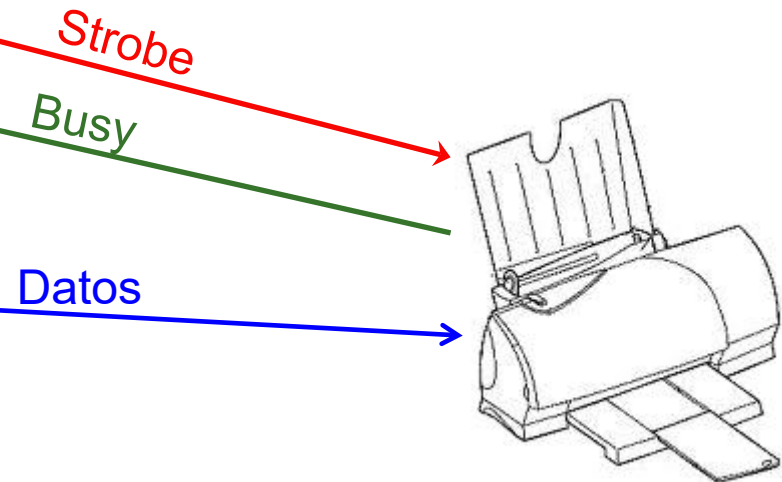
Impresora y la PIO

PA₀ : línea BUSY (*entrada*)

PA₁ : línea STROBE (*salida*)

PB₀...PB₇ : línea de datos (*salida*)

| PIO | | |
|-----|----|------------|
| 30H | PA | XXXXXX S B |
| 31H | PB | |
| 32H | CA | |
| 33H | CB | |



Ej. 4

Uso de la impresora a través de la PIO

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.

Ej. 4

PIO EQU 30H

ORG 1000H

MSJ DB "ARQUITECTURA DE "

DB "COMPUTADORAS"

FIN DB ?

ORG 2000H

MOV AL, 0FDH } CA = 1111 1101 → B= Entrada S=Salida
OUT PIO+2, AL

MOV AL, 0 } CB = 0000 0000 → PB=Salida
OUT PIO+3, AL

IN AL, PIO }
AND AL, 0FDH } Fuerza Strobe a 0

OUT PIO, AL }
MOV BX, OFFSET MSJ } Inicializo BX y
MOV CL, OFFSET FIN-OFFSET MSJ } CL

POLL: IN AL, PIO }
AND AL, 1 } Lee Busy y se queda en el
JNZ POLL } lazo mientras sea 1. (ocupada)

MOV AL, [BX] } Mando al puerto de datos
OUT PIO+1, AL } (PB) el carácter a imprimir

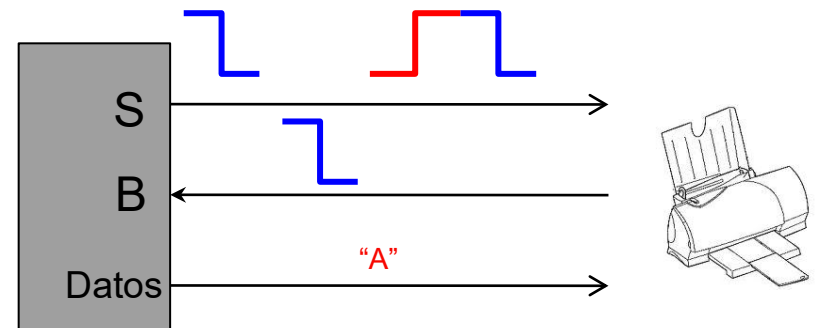
IN AL, PIO }
OR AL, 02H } Fuerzo Strobe a 1

OUT PIO, AL }
IN AL, PIO }
AND AL, 0FDH } Fuerzo Strobe a 0

OUT PIO, AL }
INC BX } BX apunto al siguiente
DEC CL } caracter
JNZ POLL }

INT 0

END

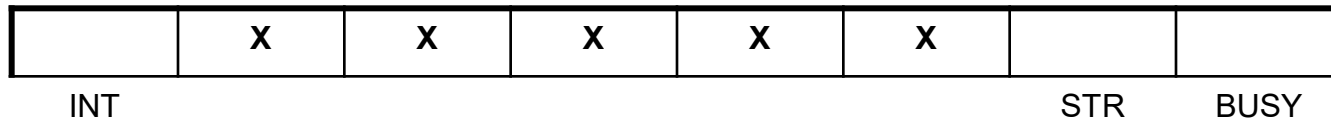


HAND

Interfaz con la impresora: cumple temporización necesaria para la norma Centronics.

Dos registros de 8 bits:

- ❑ **DATO:** D0.....D7 (40H)
- ❑ **EST:** registro de estado (41H)



En lectura :

BIT 0 en 0: línea BUSY desactivada

en 1: línea BUSY activada

BIT 1 en 0: línea STROBE desactivada

en 1: línea STROBE activada

BIT 2..6 No se usan

BIT 7 en 0: no se activará la línea INT

en 1: se activará la línea INT cuando BUSY no esté activa

En escritura :

BIT 0....6: no se usan

BIT 7 en 0 : inhibe de la línea INT. (Polling)

en 1 : se activa INT

Ej. 7

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

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.

Ej. 7

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
```

Se configura el HAND para polling
(7FH = 0111 1111)

Lee Busy y se queda en el lazo
mientras sea 1. (espera ocupada)

Envío el carácter a imprimir

Ej. 8

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

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.

Ej. 8

```
PIC      EQU 20H
HAND     EQU 40H
N_HND    EQU 10
```

ORG 40

```
IP_HND   DW   RUT_HND
```

ORG 1000H

```
MSJ DB  "UNIVERSIDAD "
      DB  "NACIONAL DE LA PLATA"
FIN  DB  ?
```

ORG 3000H

```
RUT_HND: PUSH AX
          MOV AL, [BX]
          OUT HAND, AL } Envía el carácter a imprimir
          INC BX
          DEC CL
          MOV AL, 20H
          OUT PIC, AL  } Fin de atención a la interrupción
          POP AX
          IRET
```

ORG 2000H

```
MOV BX, OFFSET MSJ
MOV CL, OFFSET FIN-OFFSET MSJ
CLI
MOV AL, 0FBH
OUT PIC+1, AL } IMR = 11111011 → INT2 (HAND)
MOV AL, N_HND
OUT PIC+6, AL } Asigno a INT2 el tipo 10
MOV AL, 80H
OUT HAND+1, AL } Configuro el HAND para interrupción
STI
LAZO: CMP CL, 0
      JNZ LAZO } Mientras tenga caracteres para imprimir se queda en lazo
      IN AL, HAND+1
      AND AL, 7FH
      OUT HAND+1, AL } Deshabilita las interrupciones
      INT 0
      END
```

¿Preguntas?