

# Arquitectura de computadoras



2020

**Explicación Práctica 3**

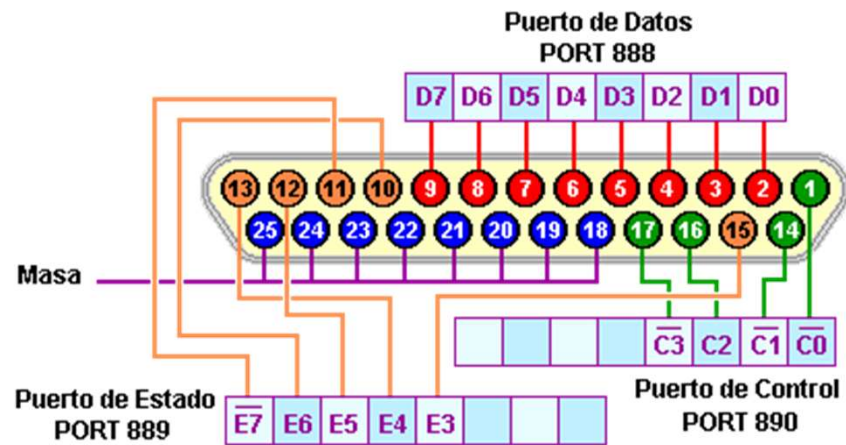
Entrada / Salida

# Entrada / Salida

- ▶ Gran variedad de periféricos con varios métodos de operación, pero todos **son más lentos que la CPU y la RAM**
- ▶ Necesidad de una **interface** entre el procesador, la memoria y los periféricos
- ▶ MSX88 utiliza lo que se llama **E/S aislada** → las direcciones de E/S, llamadas **puertos**, están separadas de la memoria
  - ▶ **Ventaja:** no existe memoria de E/S mapeada en memoria principal (no se desperdicia espacio)
  - ▶ **Desventaja:** para transferir datos se debe usar instrucciones especiales como **IN** y **OUT**.

# Puerto paralelo (Centronics)

- ▶ Registro de **control** (bidireccional): 4 bits.
- ▶ Registro de **estado** (entrada): 5 bits.
- ▶ Registro de **datos** (bidireccional): 8 bits.



# PIO (puerto paralelo E/S)

- ▶ Son 2 puertos 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.

<b>PIO</b>		
<b>Datos</b> {	30H	<b>PA</b>
	31H	<b>PB</b>
<b>Configuración</b> {	32H	<b>CA</b>
	33H	<b>CB</b>

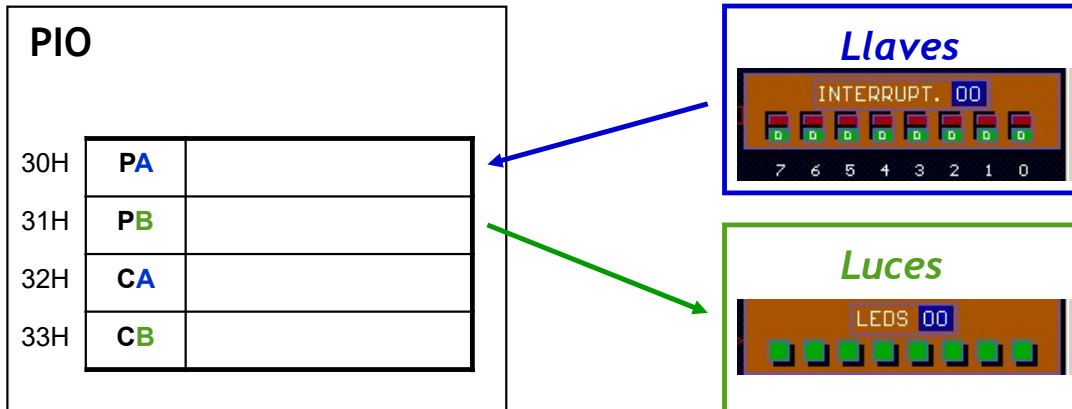
Un bit en **0** en CA/CB selecciona como **salida** a la línea correspondiente en PA/PB

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

# Ejercicio PIO-luces/llaves

Encendido/apagado de las **luces** (periférico de salida) mediante la barra de microconmutadores o **llaves** (periférico de entrada), ambos comunicados con el microprocesador a través de los puertos paralelos de la PIO.

Implementar un programa 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. Las llaves se manejan con las teclas 0-7.



# Ejercicio PIO-luces/llaves

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

**Llaves**

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

**Luces**

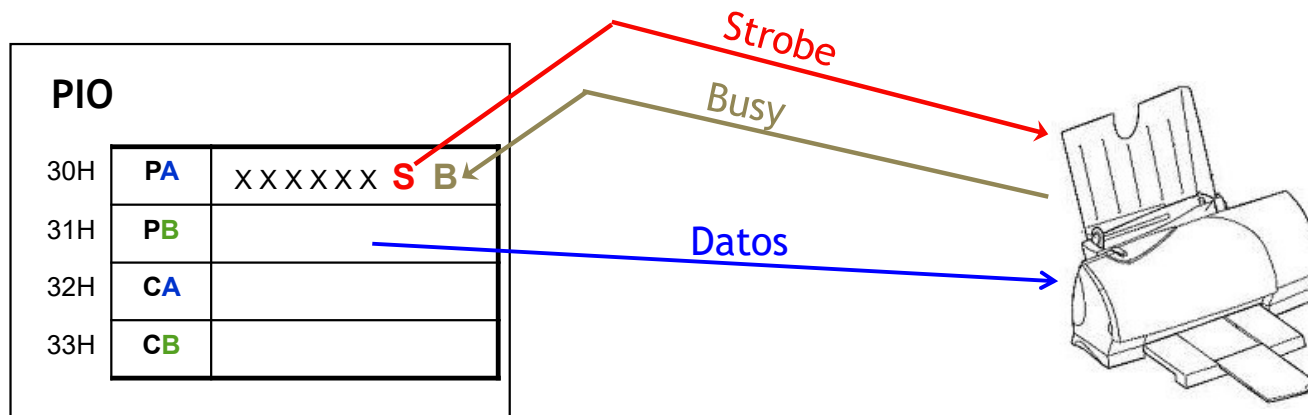
1	1	0	0	0	0	1	1
7	6	5	4	3	2	1	0

**PIO**

30H	<b>PA</b>	
31H	<b>PB</b>	
32H	<b>CA</b>	
33H	<b>CB</b>	

# Impresora - PIO

$PA_0$  : línea **BUSY** (*entrada*)  
 $PA_1$  : línea **STROBE** (*salida*)  
 $PB_0...PB_7$  : líneas de **DATOS** (*salidas*)



# Ejercicio PIO-imp

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**.



# Ejercicio PIO-imp

PIO EQU 30H

ORG 1000H

MSJ DB "ORG. Y ARQ. DE COMPUTADORAS"

FIN DB ?

ORG 2000H

MOV AL, 0FDH

OUT PIO+2, AL

MOV AL, 0

OUT PIO+3, AL

IN AL, PIO

AND AL, 0FDH

OUT PIO, AL

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

OR AL, 02H

OUT PIO, AL

IN AL, PIO

AND AL, 0FDH

OUT PIO, AL

INC BX

DEC CL

JNZ POLL

INT 0

END

CA = 1111 1101 → B= Entrada S=Salida

CB = 0000 0000 → PB=Salida

Fuerzo Strobe a 0

Inicializo BX y CL

Leo estado de la linea BUSY y no  
avanzo hasta que no este libre (0)

Coloco el carácter a imprimir  
en el puerto de datos (PB)

Fuerzo Strobe a 1

Fuerzo Strobe a 0

BX apunto al siguiente caracter

