8255

(Programmable peripheral interface)

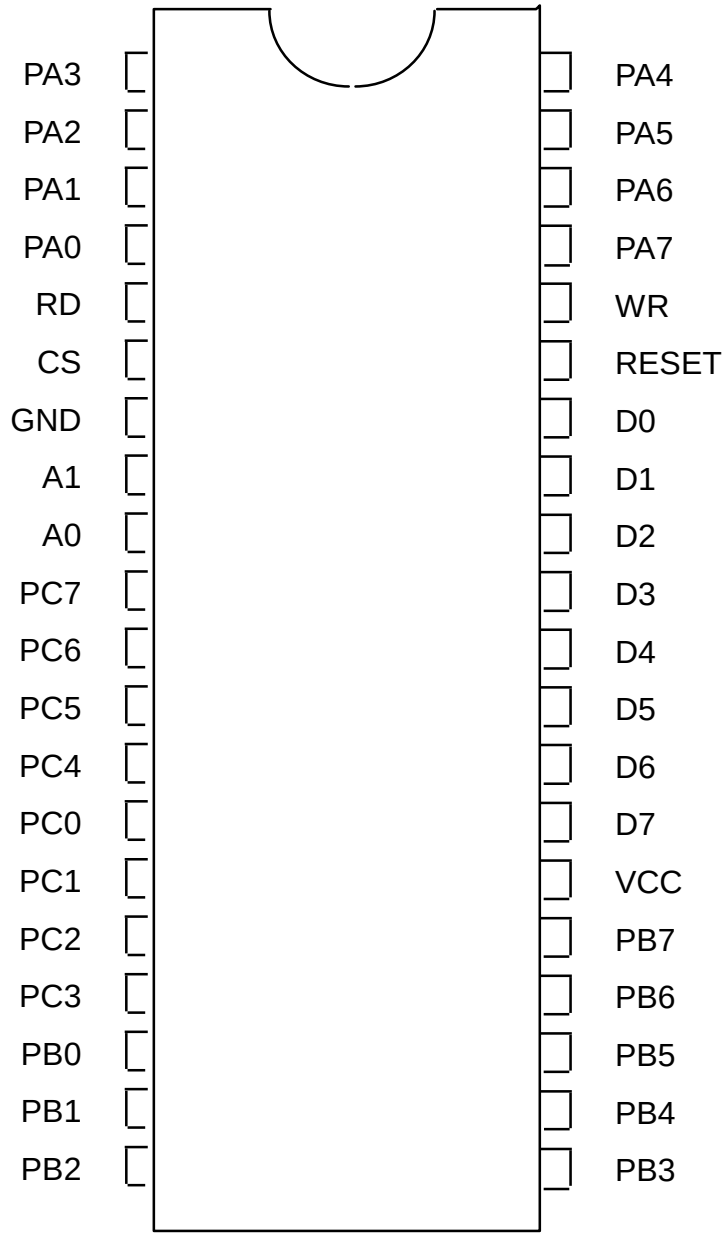
M. Rebaudengo - M. Sonza Reorda – E. Sanchez

Politecnico di Torino
Dip. di Automatica e Informatica

The programmable peripheral interface

- **The 8255 implements a parallel I/O interface for 8085 and 8086 –based systems**
- **It was initially produced usign a DIP 40-pin LSI chip**
- **The 8255 allows to manage I/O of different data types: bit, nibble, and byte.**
- **The 8255 device is fully configurable via software**
- **It is possible to have 3 independent I/O ports of 1 byte.**

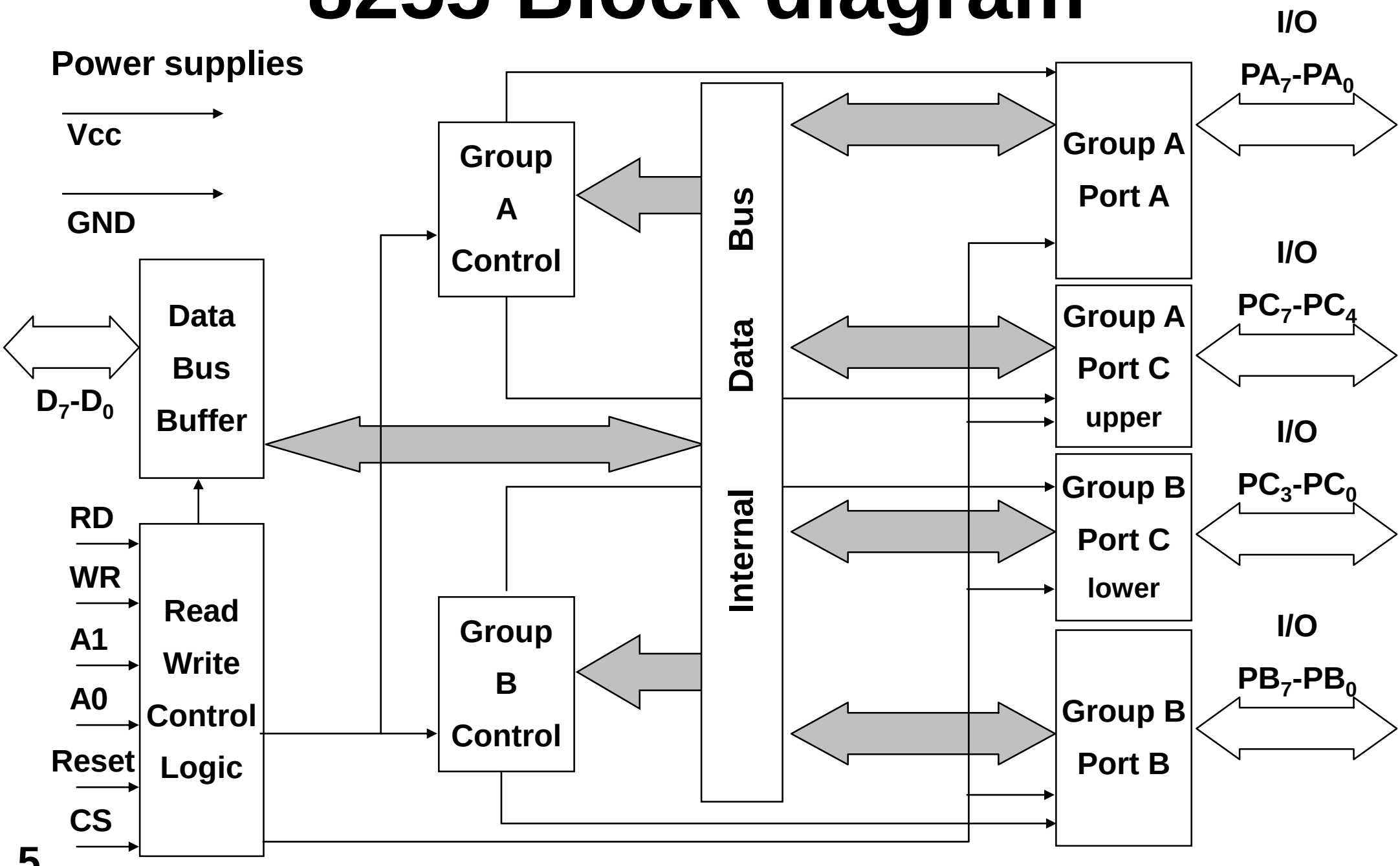
The 8255 chip



Pin description

- **$D_{0\text{-}7}$:** Data Bus
- **$PA_{0\text{-}7}$:** PORT A
- **$PB_{0\text{-}7}$:** PORT B
- **$PC_{0\text{-}7}$:** PORT C
- **$\overline{RD}$:** Read Control
- **$\overline{WR}$:** Write Control
- **$\overline{CS}$:** Chip Select
- **VCC:** System Power
- **GND:** System Ground
- **$A_{0\text{-}1}$:** Address
- **RESET:** Reset

8255 Block diagram



8255 logic model

The programmer's model of the 8255 is based on 4 8-bit registers corresponding to three registers that access to the programmable ports, and one control register.

Data transfer is made by accessing in reading and writing mode to the 3 port registers.

The control register allows to configure every one of the ports in the device, and it is only accessible in writing mode.

The 4 8255 registers can be accessed through $D_{0\text{-}7}$, and using the address bits A_0 and A_1.

Control signals

- **CS (*chip select*): a low level signal enables the CPU-8255 data transfer**
- **RD (*read*): a low level signal enables the 8255 to send to the CPU a port data value or the 8255 status value using the data bus.**
- **WR (*write*): a low level signal enables the 8255 to receive from the data bus a data or the control register value written by the CPU.**
- **RESET: a high level value resets the 8255: the three ports are configured as input ports in mode 0.**

Accessing to the device ports

A_0 and A_1 controls the access to the three 8255 ports as well as the Control Word Register (*CW*).

A_0 and A_1 are usually connected to the least significant bits of the address bus.

A_1	A_0	*Register*
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Control Register

8255 Input/Output Operation

A1	A0	RD	WR	CS	
0	0	0	1	0	Port A ⇒ Data Bus
0	1	0	1	0	Port B ⇒ Data Bus
1	0	0	1	0	Port C ⇒ Data Bus
0	0	1	0	0	Data Bus ⇒ Port A
0	1	1	0	0	Data Bus ⇒ Port B
1	0	1	0	0	Data Bus ⇒ Port C
1	1	1	0	0	Data Bus ⇒ Control
X	X	X	X	1	Data Bus ⇒ 3-State
X	X	1	1	0	Data Bus ⇒ 3-State
1	1	0	1	0	Not available
X	X	0	0	X	Not available

8255 modes of operation

The 8255 ports can be configured in 3 different operative modes:

- **Mode 0: *Basic Input/Output***
- **Mode 1: *Strobed Input/Output***
- **Mode 2: *Bidirectional Bus*.**

At the reset time, the three 8255 ports are initialized as input ports in mode 0.

Ports groups

The 24 I/O 8255 pins are split in two groups counting 12 bits each:

- **Group A: Port A and Port C (most significant or upper bits)**
- **Group B: Port B and Port C (least significant or lower bits).**

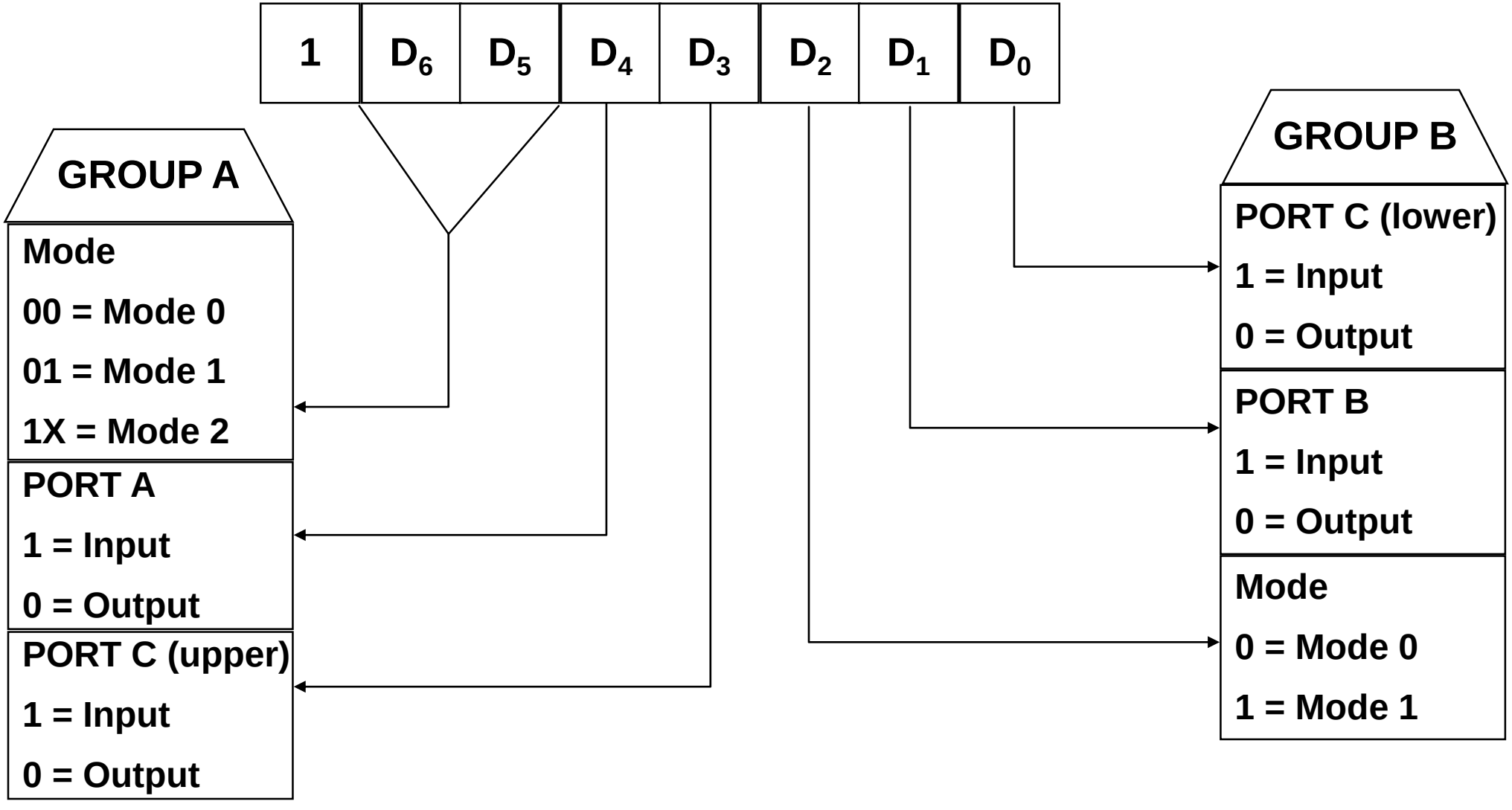
Control word

The CW register is written when the CPU writes in the 8255 control register.

Two different functions are available:

- **Configuring the three 8255 ports**
- **Writing a logic value in a single bit of the Port C.**

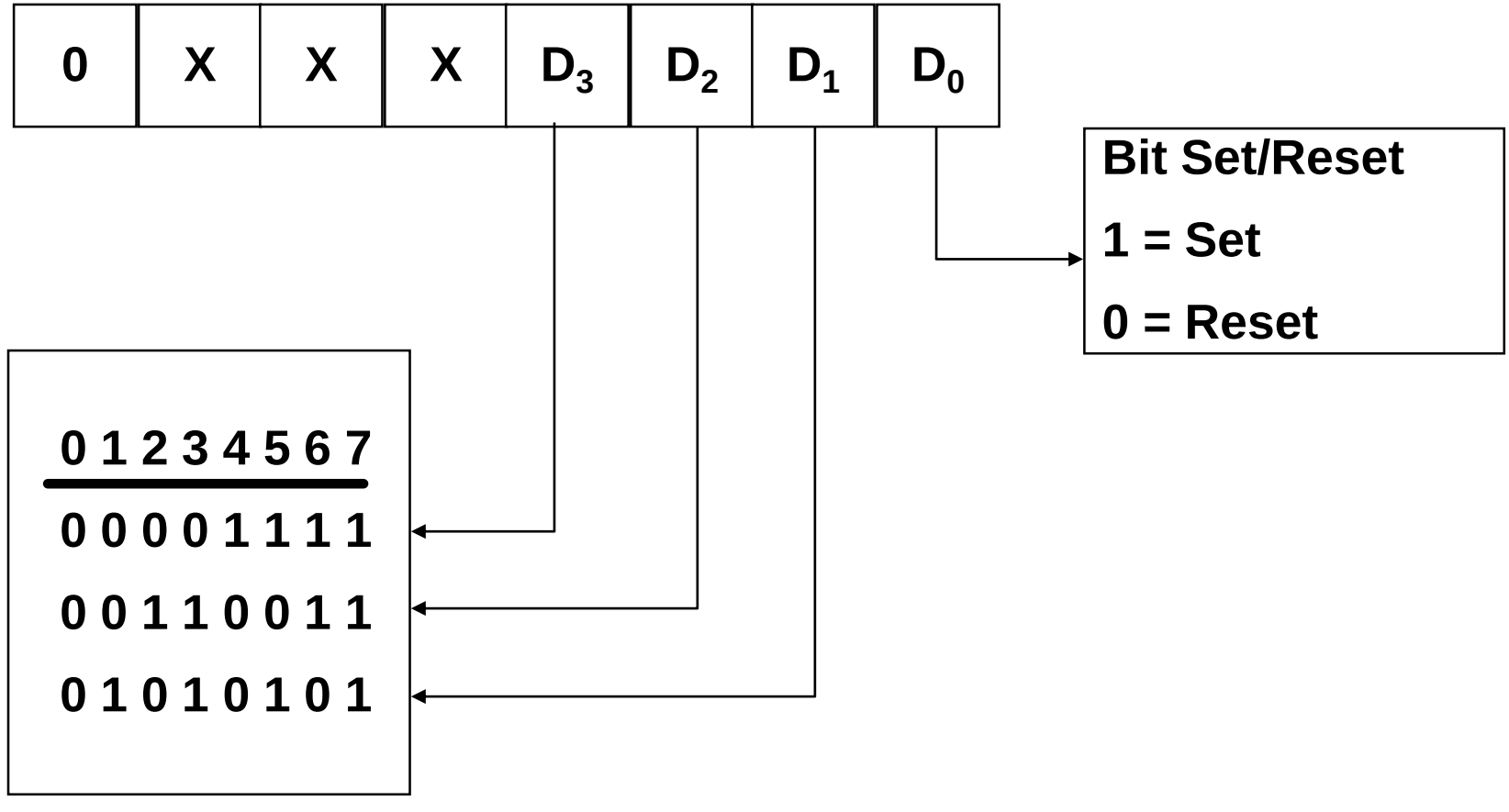
Programming the 8255



Single Bit Set/Reset

A writing operation in the Control Register can force the value on every single bit in the C port.

Single Bit Set/Reset



Interrupt Control

In modes 1 and 2, some control signals in the port C can be used to send an interrupt request to the CPU.

These signals can be enabled or disabled by setting or resetting the internal *interrupt-enable* flip-flop (INTE), through the port C *bit set/reset* operation.

INTE enables the interrupt when the related port C bit is set to 1.

Mode 0
(*Basic Input/Output*)

This operation mode allows to configure all the 8255 ports as Input / Output ports.

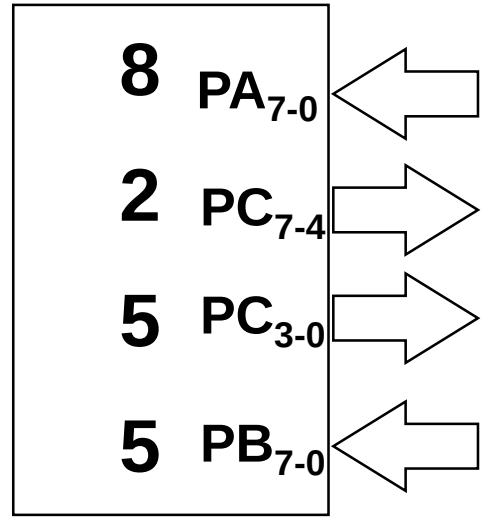
In this mode, no handshaking protocol neither handshake signals are allowed.

Mode 0

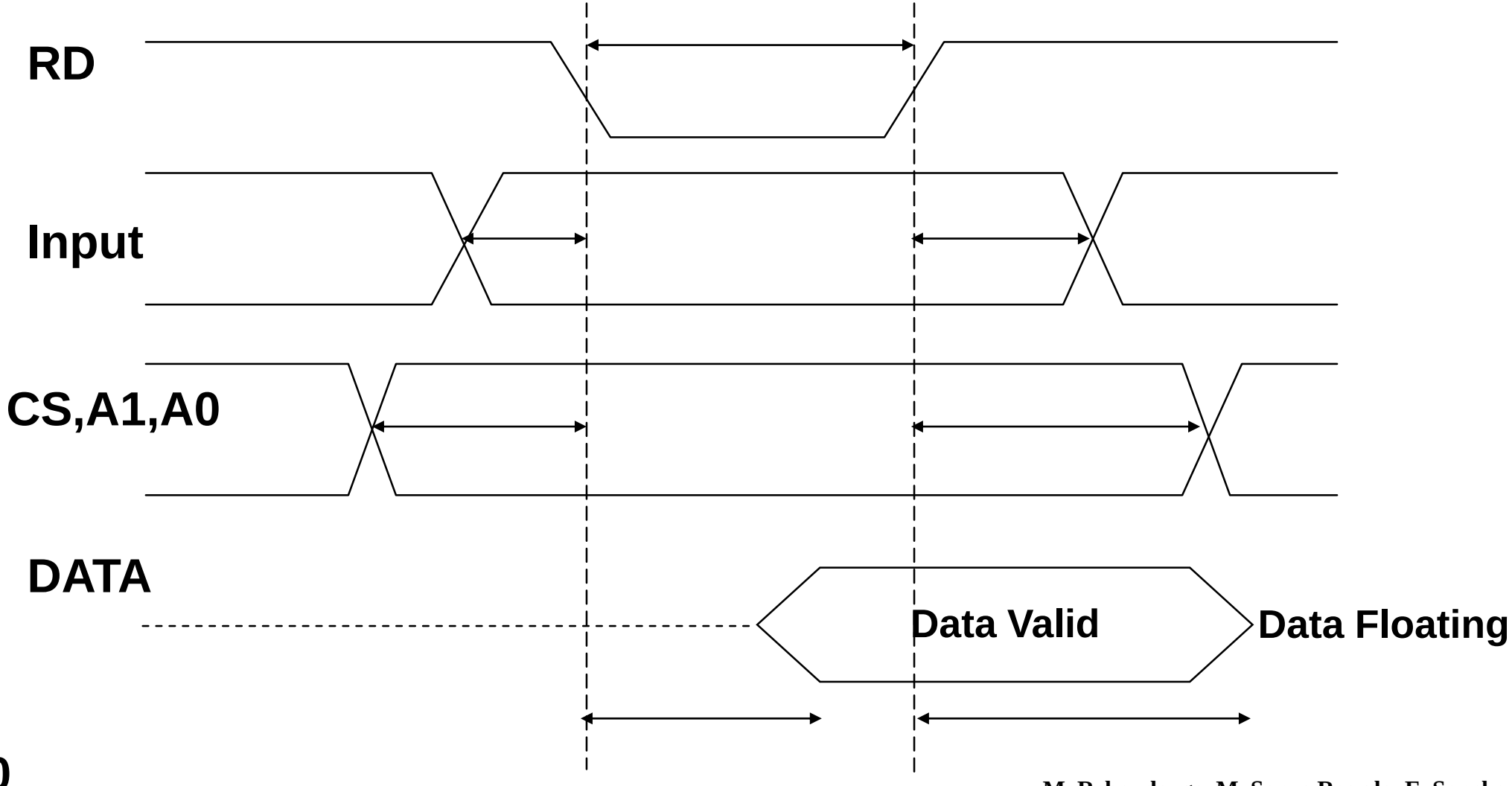
- **Configurable to obtain 2 8-bit ports (ports A and B) and 2 4-bit ports (port C).**
- **Any 8255 port can be configured as input or output**
- **Output ports are latched**
- **Input ports are not latched**
- **A total of 16 different I/O configurations.**

Example

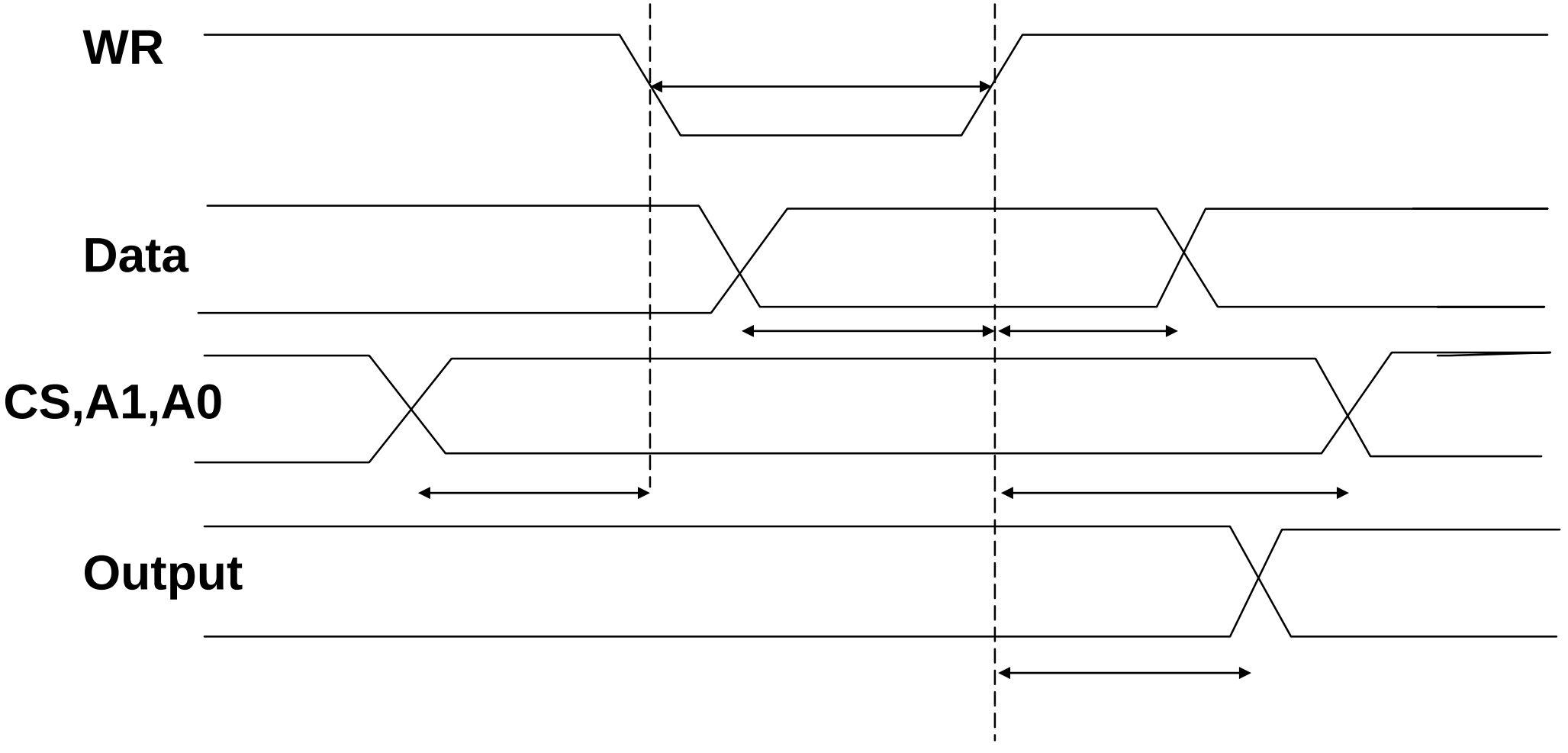
D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	0



Mode 0 - Input



Mode 0 - Output



Mode 1
(*Strobed Input/Output*)

Data transfer in this mode is supported by some handshaking signals that allow handling interrupts.

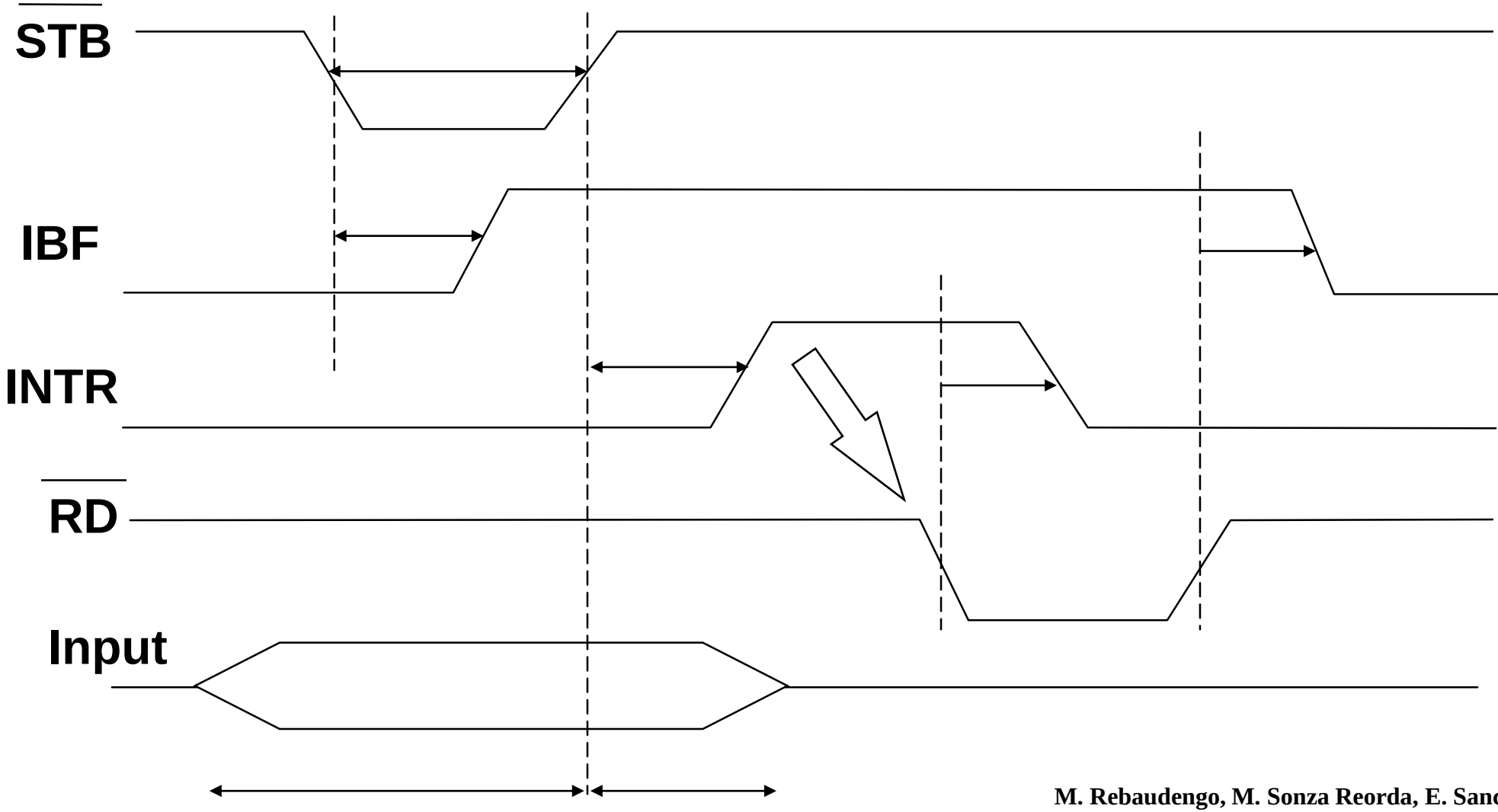
Mode 1

- **2 possible Groups**
- **Every group is composed of an 8-bit data port and a control 4-bit port**
- **Data ports can be configured as Input or Output**
- **Input and Output are latched.**

Control signals for Mode 1 in Input

- **STB (*Strobe Input*): a signal low value load the data in the input latch**
- **IBF (*Input Buffer Full*): a high level value indicates that the value is loaded in the input latch. It can be used as an acknowledge signal.**
- **INTR (*Interrupt Request*): a high level value can be used as an interrupt request to the CPU.**
- **$INTE_A$ (*Interrupt Enable* per group A): controlled by PortC[4] bit.**
- **$INTE_B$ (*Interrupt Enable* per group B): controlled by PortC[2] bit.**

Timing diagram



Functioning mode

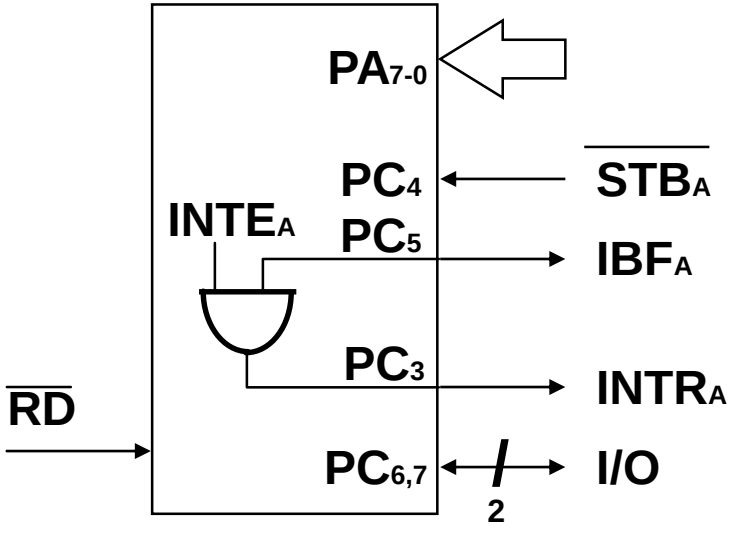
- **IBF is set to one when STB is low, and it is reset again at the raising edge of RD.**
- **INTR is set to one once STB is high, IBF is high, and INTE is set to one; then, INTR is reset in the falling edge of RD.**

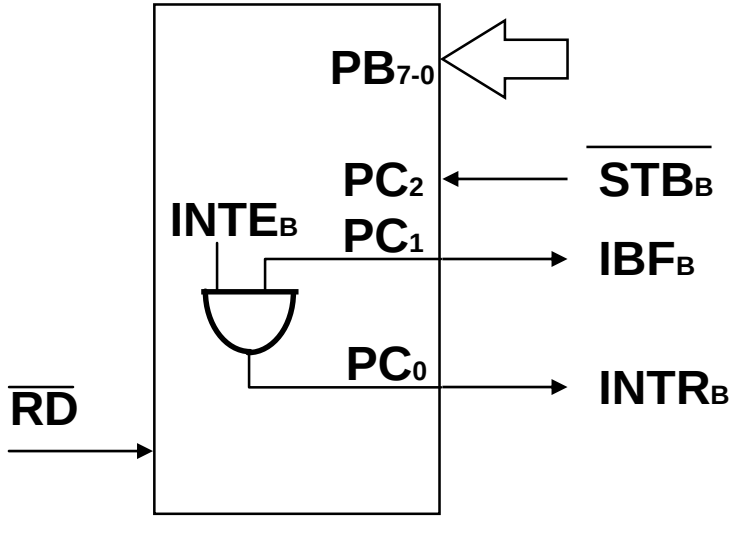
Mode 1: Input

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	1	1/0	X	X	X

PC 6,7
1=Input
0=Output

D7	D6	D5	D4	D3	D2	D1	D0
1	X	X	X	X	1	1	X

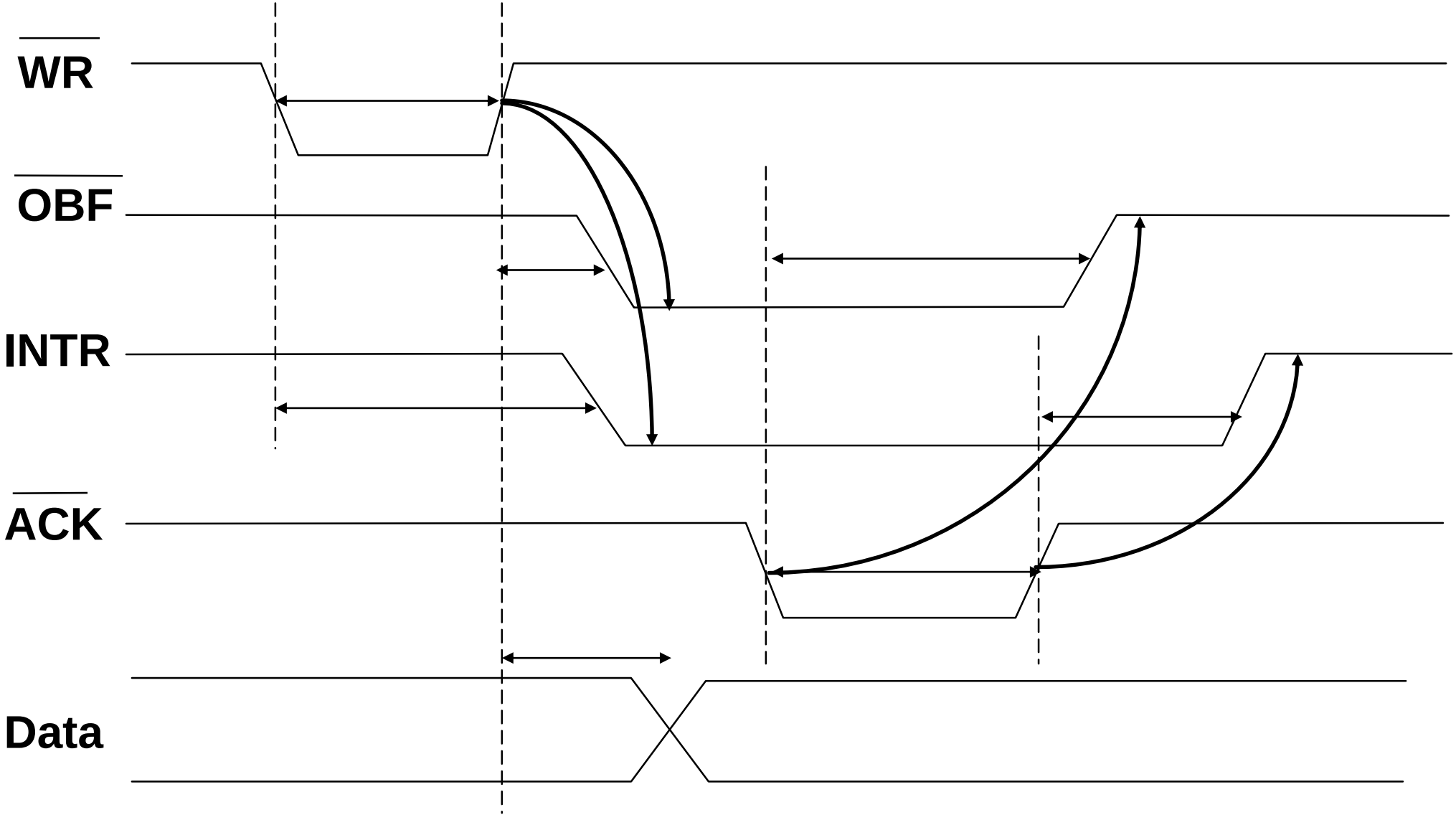




Control signals for Mode 1 in Output

- **OBF (*Output Buffer Full*): a low value signals that the CPU wrote a data in the Port.**
- **ACK (*Acknowledge Input*): a low value indicates the 8255 that a data was received by the external peripheral.**
- **INTR (*Interrupt Request*): a high value can be used as an interrupt request to the CPU.**
- **$INTE_A$ (*Interrupt Enable* per Group A): controlled by the bit PortC[6].**
- **$INTE_B$ (*Interrupt Enable* per Group B): controlled by the bit PortC[2].**

Timing diagram



Functioning mode

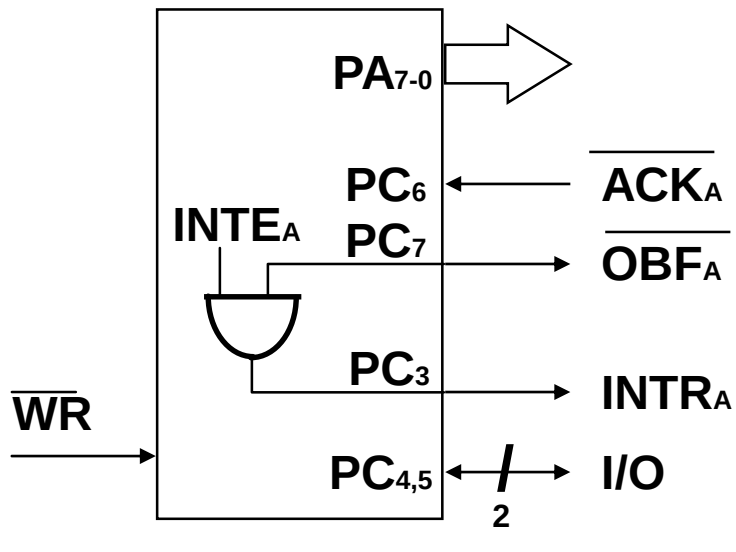
- **OBF is set during the raising edge of WR and it is reset when the ACK goes to 0.**
- **INTR is reset at the falling edge of WR, and it is set to one when ACK is a high, OBF is high, and INTE is set to one.**

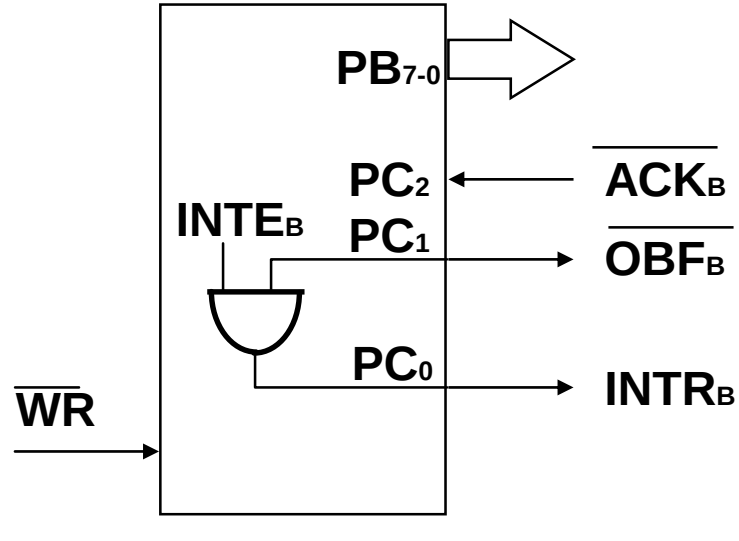
Mode 1: Output

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	0	1/0	X	X	X

PC 4,5
1=Input
0=Output

D7	D6	D5	D4	D3	D2	D1	D0
1	X	X	X	X	1	0	X





Mode 2
(*Bidirectional I/O*)

This functioning mode allows to configure the 8255 as a unique bidirectional I/O port supported by handshake signals.

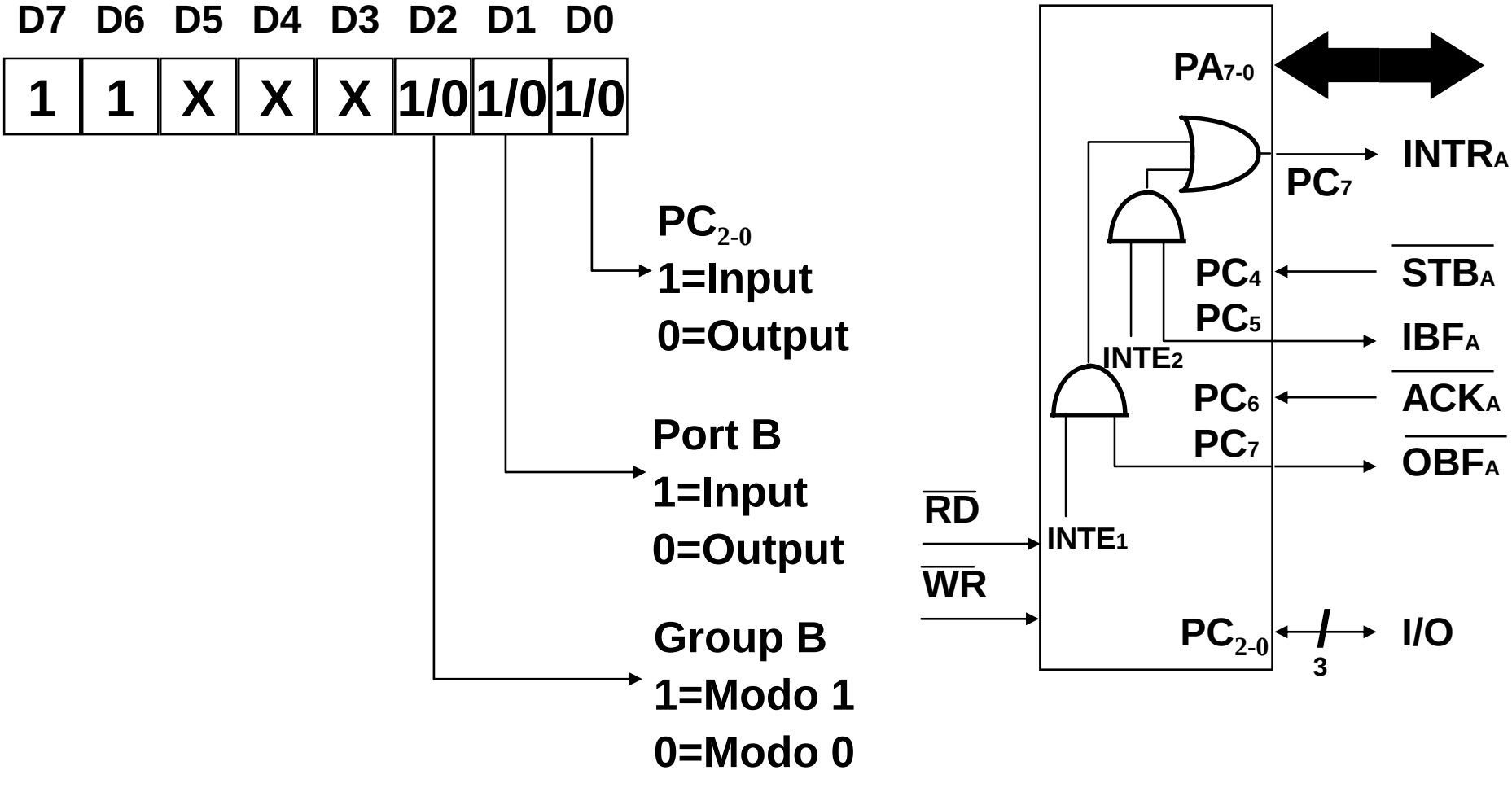
Mode 2

- **Only available for Group A.**
- **One 8-bit bidirectional port (Port A) and a control port using that uses 5 bits of Port C.**
- **Input and Output are latched.**

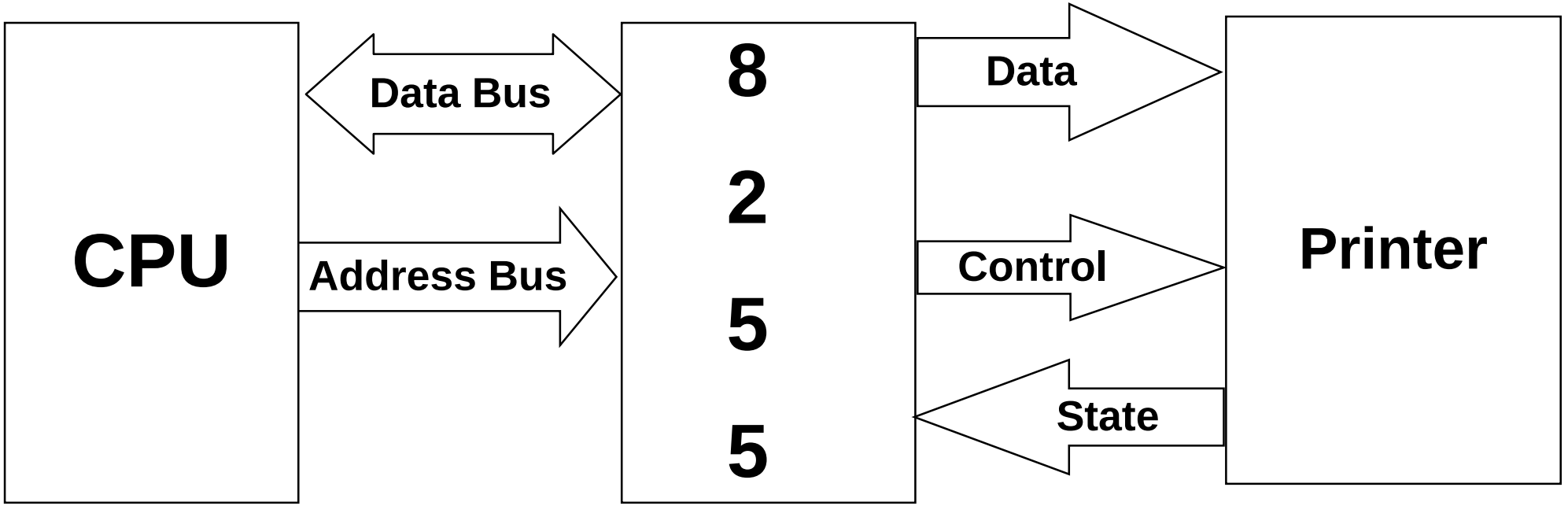
Control signals

- **INTR: a high value can be used as an interrupt request to the CPU.**
- **OBF: a low value indicates that the CPU wrote a value in the Port A.**
- **ACK: a low value enables to send a data value.**
- **STB: a low value load the data into the input latch.**
- **IBF: a high value indicates that the data is loaded in the input latch.**
- **$INTE_1$ (Output Interrupt Enable): controlled by the bit PC[6].**
- **$INTE_2$ (Input Interrupt Enable): controlled by the bit PC[4].**

Mode 2


D7 D6 D5 D4 D3 D2 D1 D0
1 1 X X X 1/0 1/0 1/0
PC2-0
1=Input
0=Output
Port B
1=Input
0=Output
Group B
1=Modo 1
0=Modo 0
PA7-0
INTRA
PC7
PC4
STBA
PC5
IBFA
INTE2
PC6
ACKA
PC7
OBFA
RD
INTE1
WR
PC2-0
3
I/O

Example: PC-printer interface



Centronics Parallel port

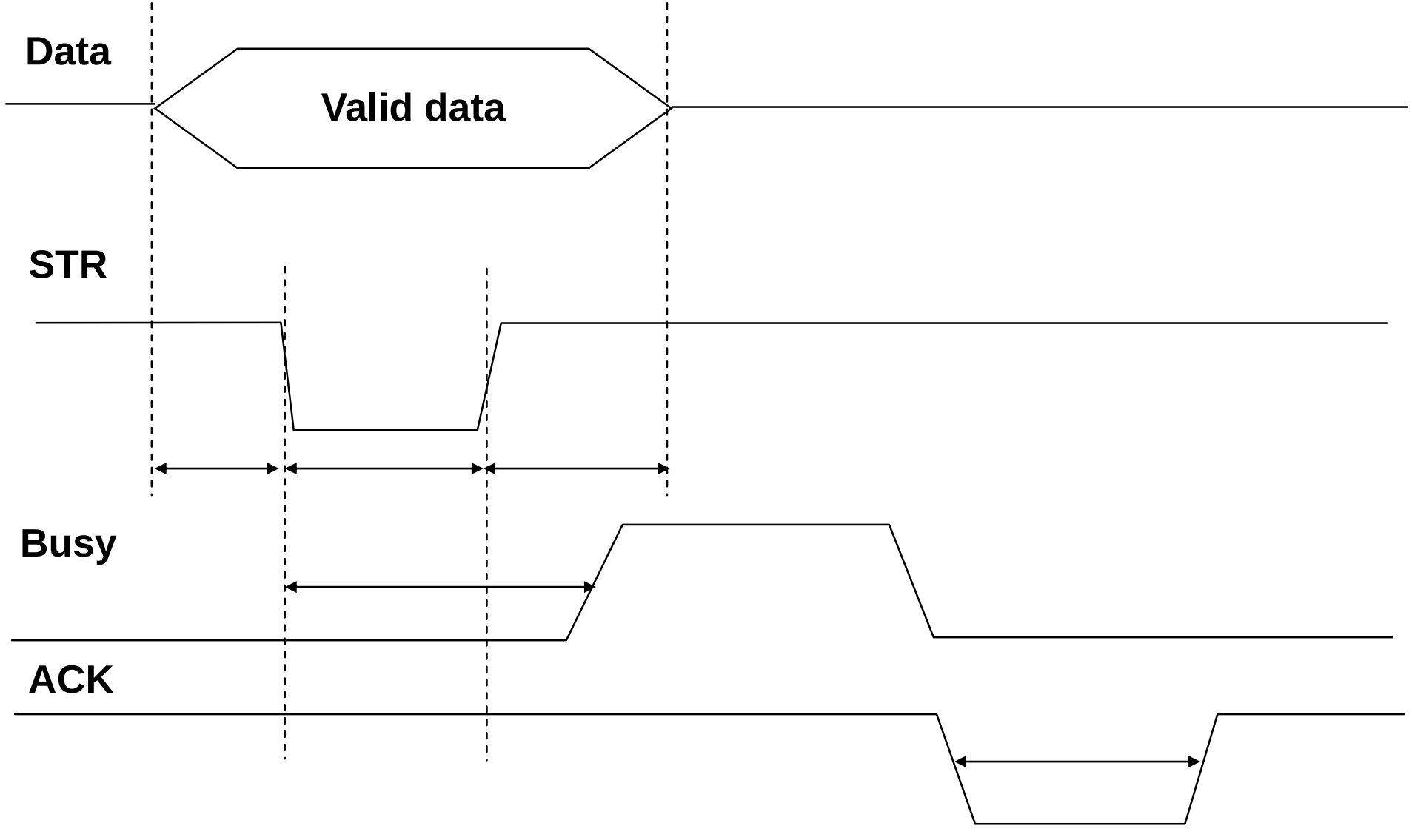
The parallel port Centronics is composed of the following pins:

- **8 Data bits;**
- **STROBE: a low value lasting more than 1us writes one data byte in the printer;**
- **ACK: the printer sends a low level value indicating that the data was received;**
- **BUSY: a high value indicates that the printer cannot receive any new data;**
- **AUTOFDX: asks the printer to go to the next line;**

Centronics Parallel port (II)

- **PAPER EMPTY: a high level value indicates that the paper is missing;**
- **ON LINE: a high level value indicates that the printer is available;**
- **HI: the printer sends a high level value while being turned on;**
- **RESET: cancel the current job by cleaning the printing buffer.**
- **ERROR: a low level value indicates that there is a printer error.**

Communication protocol



8255-Centronics interface

Port A: Data word

Port B: Printer status word

Port C: Printer control word.

Port B
Printer status word

- **PB7: BUSY (the printer cannot accept new data)**
- **PB6: ACK (data reception, printer enable)**
- **PB5: PE (paper empty)**
- **PB4: SLCT (On line)**
- **PB3: ERR (error signal).**

Port C
Printer control word

- **PC0: STB (strobe for transfering data)**
- **PC1: AUTOFDX (*autofeed*, go to the enxt line)**
- **PC2: INIT (printer initialization)**
- **PC3: SLCTIN (enables the printer to acept new data).**

Program

```
prDATA    EQU  0378h      ; address LPT1
prSTAT    EQU  prDATA+1   ; port B
prCTRL    EQU  prDATA+2   ; port C
DELAY     EQU  100        ; delay
sERR      EQU  08h        ; enabled LOW
sSEL      EQU  10h        ; enabled HIGH
sPE       EQU  20h        ; enabled HIGH
sACK      EQU  40h        ; enabled LOW
sBUSY     EQU  80h        ; enabled LOW
cSTB      EQU  01h        ; enabled HIGH
cAUTO     EQU  02h        ; enabled HIGH
cINIT     EQU  04h        ; enabled LOW
cSEL      EQU  08h        ; enabled HIGH
```

```
            .MODEL    small
            .STACK
            .DATA
msg         DB   'Hello world',0Dh,0Ah,0
            .CODE
            .STARTUP
            MOV  BX, OFFSET msg
            XOR  SI, SI
next:       MOV  AL, [BX][SI]
            CMP  AL,  0
            JE   done
            CALL pr_al      ;prints a character
            INC  SI
            JMP  next
done:       .EXIT
```

```
pr_al           PROC
                PUSH DX
                PUSH AX
                MOV  AH, AL     ; writes AL in AH
                MOV  DX, prSTAT
pr_n_ready:     IN   AL, DX     ; read the printer
                                ; status
                TEST AL, sERR
                JZ   pr_error   ; error
                TEST AL, sBUSY
                JZ   pr_n_ready ; busy
                TEST AL, sSEL
                JZ   pr_n_ready ; on line
```

```
            MOV  AL, AH
            MOV  DX, prDATA
            OUT  DX, AL     ; sends the data
            MOV  CX, DELAY
cycle1:     LOOP cycle1
            MOV  DX, prCTRL
            IN   AL, DX
            OR   AL, cSTB   ; set strobe bit
            OUT  DX, AL
            MOV  CX, 2*DELAY
cycle2:     LOOP cycle2
            AND  AL, not cSTB    ; reset strobe bit
            OUT  DX, AL
            MOV  CX, DELAY
cycle3:     LOOP cycle3
```

```
pr_done:   POP  AX
           POP  DX
           RET
pe_error:  JMP  pr_done
pr_al      ENDP
           END
```