

# Intel 8255

(interfaccia parallela programmabile)

M. Rebaudengo - M. Sonza Reorda

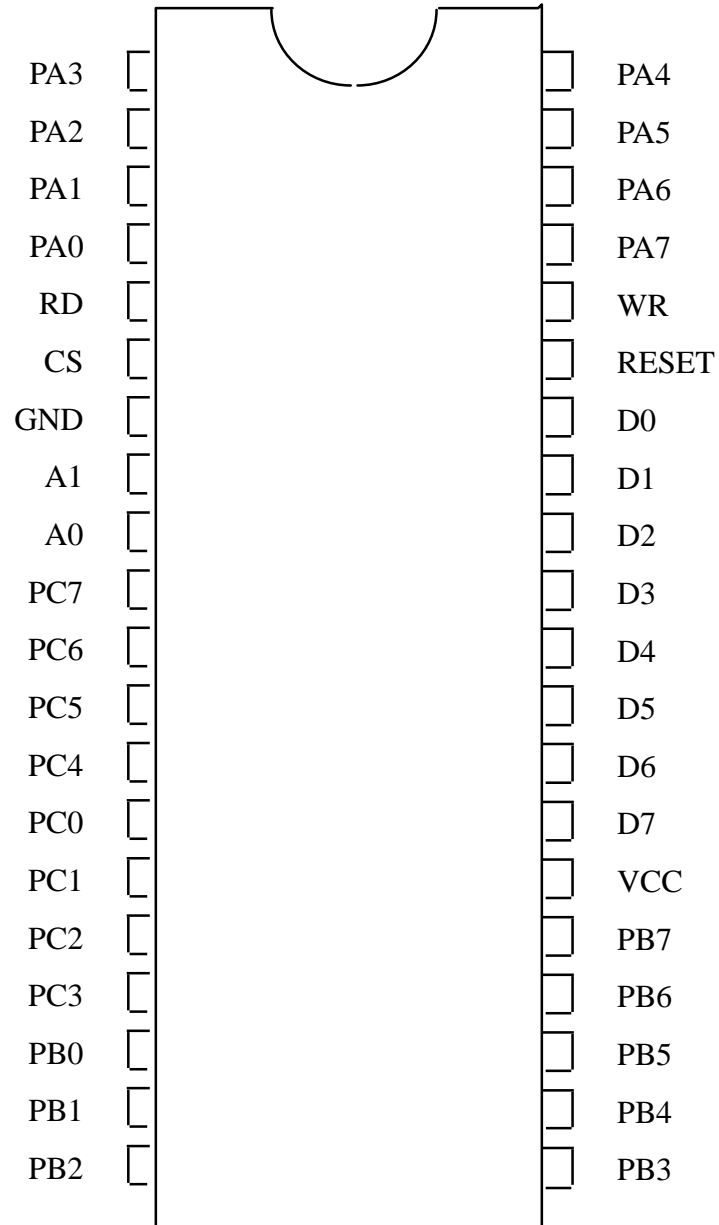
Politecnico di Torino  
Dip. di Automatica e Informatica



# Generalità

- **L'Intel 8255 implementa un'interfaccia di I/O parallela per sistemi delle famiglie 8085 e 8086**
- **Permette di eseguire input/output di bit, nibble e byte in modo completamente programmabile via software**
- **Gestisce fino a 3 porte di Input/Output indipendenti da 1 byte ciascuna**
- **È realizzato tramite un chip LSI di tipo DIP a 40 pin; oggi tali funzioni sono svolte (in modo compatibile) dalla circuiteria inclusa nel chipset**
- **È analogo ad altri dispositivi realizzati per altre famiglie di processori, quali**
  - **Motorola 6520 e 6522 per la famiglia 6502**
  - **Motorola 6820 per la famiglia 6800.**

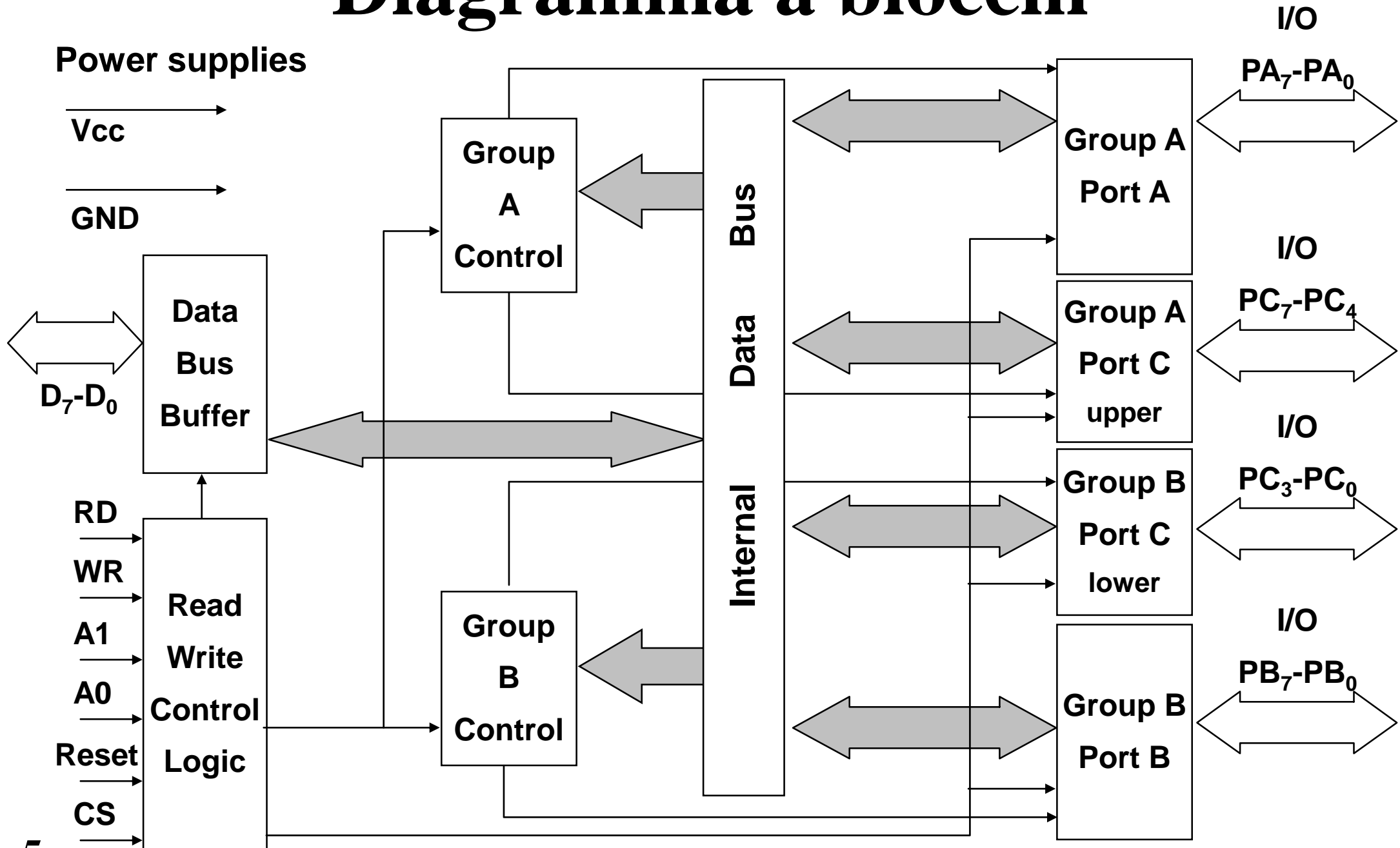
# Il chip



# Descrizione dei pin

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

# Diagramma a blocchi



# Modello logico

**Dal punto di vista del programmatore l'8255 si presenta come un insieme di 4 registri da 8 bit, corrispondenti alle 3 porte e al Registro di Controllo.**

**Accedendo ai registri associati alle 3 porte si esegue il trasferimento dati.**

**Accedendo al Registro di Controllo si definisce il modo di funzionamento per ciascuna porta.**

**Il Registro di Controllo può essere solo scritto.**

**I 4 registri sono accessibili tramite i pin  $D_{0-7}$ , selezionando quello desiderato tramite i pin  $A_0$  e  $A_1$ .**

# Segnali di controllo

- **CS:** un segnale basso sul pin CS abilita le comunicazioni tra CPU e 8255
- **RD:** un segnale basso sul pin RD abilita l'8255 ad inviare il dato o l'informazione di stato sul data bus per la CPU
- **WR:** un segnale basso sul pin WR abilita la CPU a scrivere il dato o il registro di controllo in un registro interno all'8255
- **RESET:** un valore logico alto sul segnale di RESET re-inizializza il dispositivo. Tutte le porte sono riprogrammate in modo 0 in direzione Input.

# Selezione delle porte

I segnali di input  $A_0$  e  $A_1$  controllano la selezione delle 3 porte di I/O o del *Registro di Controllo* (CW).

Sono normalmente connessi ai bit meno significativi dell'Address Bus.

$A_1$	$A_0$	<i>Registro</i>
0	0	Porta A
0	1	Porta B
1	0	Porta C
1	1	Registro di Controllo



# 8255 Input/Output Operation

A1	A0	RD	WR	CS	
0	0	0	1	0	Porta A $\Rightarrow$ Data Bus
0	1	0	1	0	Porta B $\Rightarrow$ Data Bus
1	0	0	1	0	Porta C $\Rightarrow$ Data Bus
0	0	1	0	0	Data Bus $\Rightarrow$ Porta A
0	1	1	0	0	Data Bus $\Rightarrow$ Porta B
1	0	1	0	0	Data Bus $\Rightarrow$ Porta C
1	1	1	0	0	Data Bus $\Rightarrow$ Control
X	X	X	X	1	Data Bus $\Rightarrow$ 3-State
X	X	1	1	0	Data Bus $\Rightarrow$ 3-State
1	1	0	1	0	Non ammessa
X	X	0	0	X	Non ammessa

# Modi di funzionamento

**Le porte dell'8255 possono essere programmate in 3 modi operativi**

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

**Al reset l'8255 è inizializzato con tutte le porte programmate in modo 0 in input.**

# Gruppi

**I 24 pin di I/O sono suddivisi in 2 gruppi di 12 pin:**

- **Gruppo A: porta A e porta C (parte alta)**
- **Gruppo B: porta B e porta C (parte bassa).**

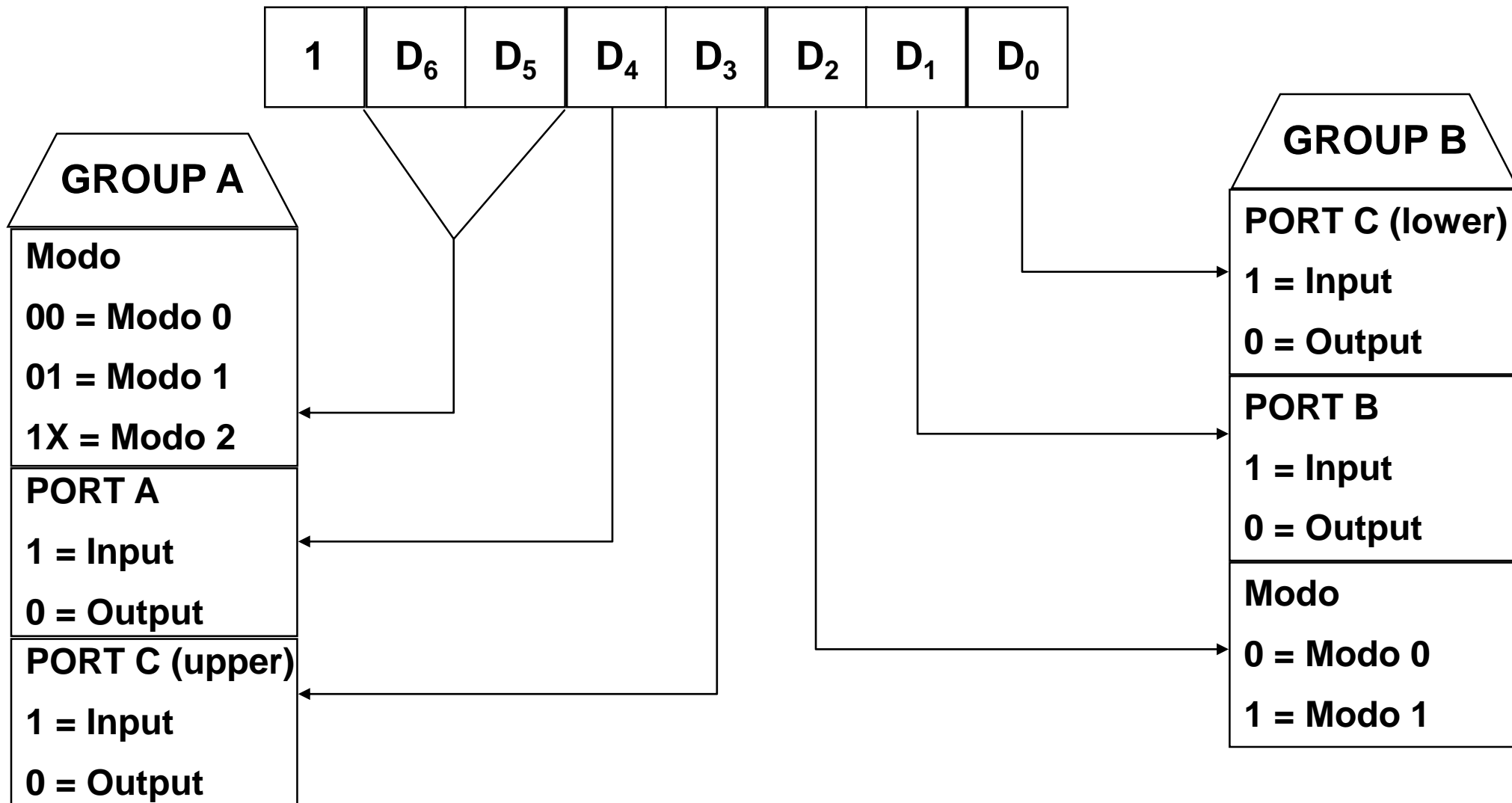
# **Parola di controllo**

**La parola di controllo viene scritta dalla CPU nel registro di controllo dell'8255.**

**Può avere due funzioni:**

- **programmazione del modo di funzionamento delle porte dell'8255**
- **scrittura di un valore logico in un singolo bit della porta C.**

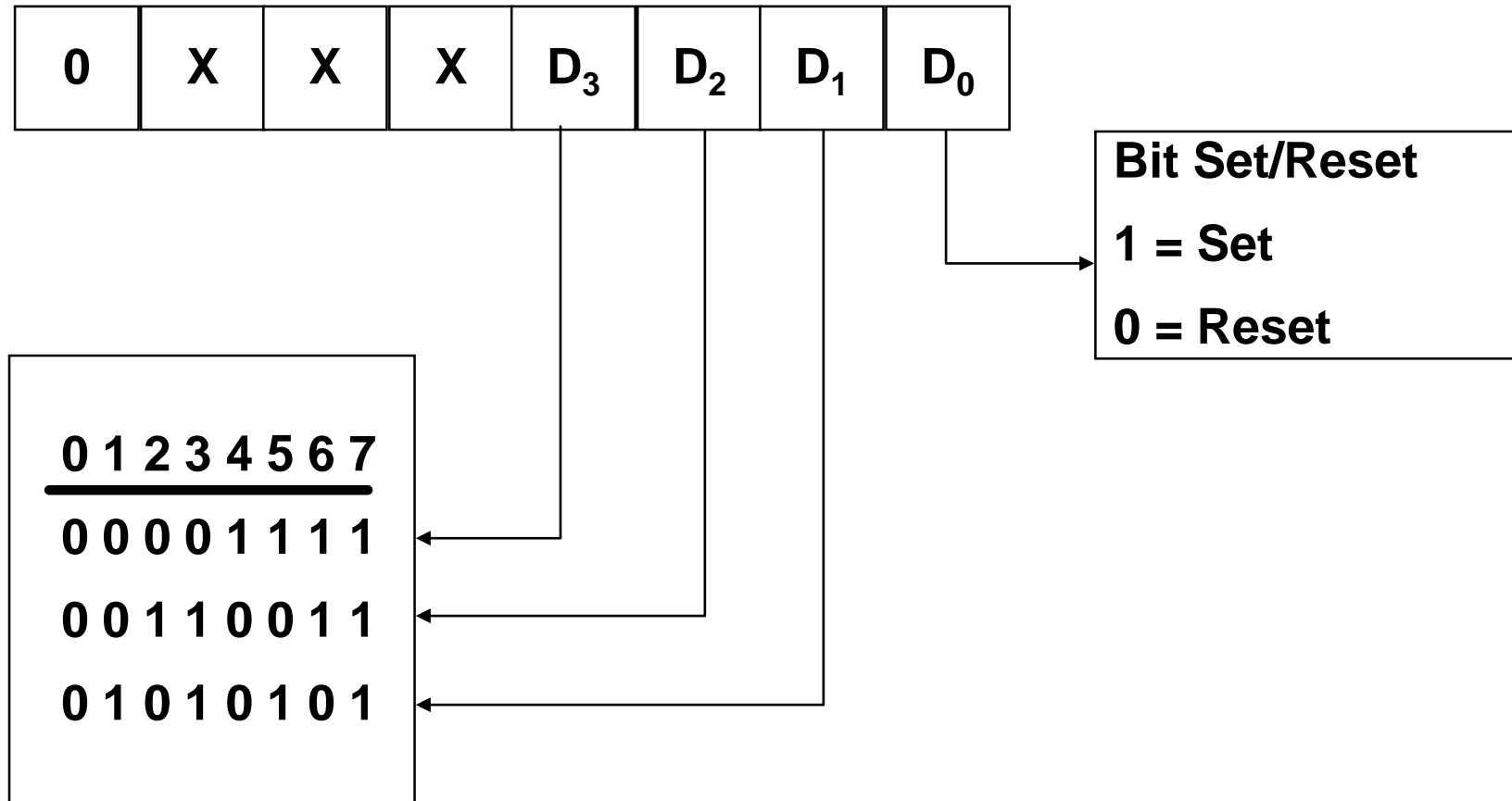
# Programmazione 8255



# Single Bit Set/Reset

**Attraverso un'operazione di scrittura sul Registro di Controllo si può forzare il valore di un singolo bit della porta C.**

# Single Bit Set/Reset



# Interrupt Control

**In modo 1 e 2 alcuni segnali di controllo, provenienti dalla porta C, possono essere utilizzati per inviare una richiesta di interrupt alla CPU.**

**Tali segnali possono essere disabilitati o abilitati settando o resettando il flip-flop interno di *interrupt enable* (INTE) attraverso *l'operazione di bit set/reset* della porta C.**

**INTE abilita l'interrupt quando l'opportuno bit della porta C è forzato ad 1.**



# **Modo 0**

## ***(Basic Input/Output)***

**Questo modo di funzionamento permette di far eseguire operazioni di Input e Output a ciascuna delle singole Porte.**

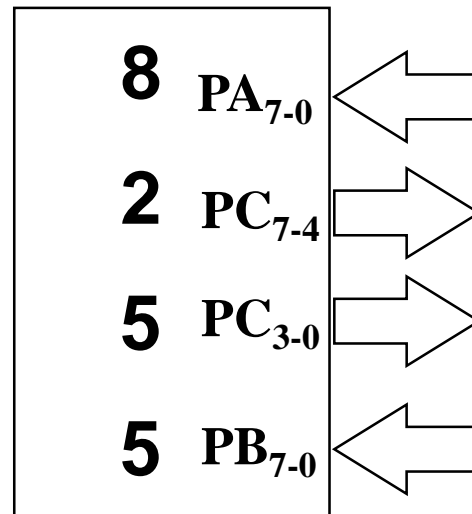
**Non sono richiesti segnali di *handshaking*.**

# Modo 0

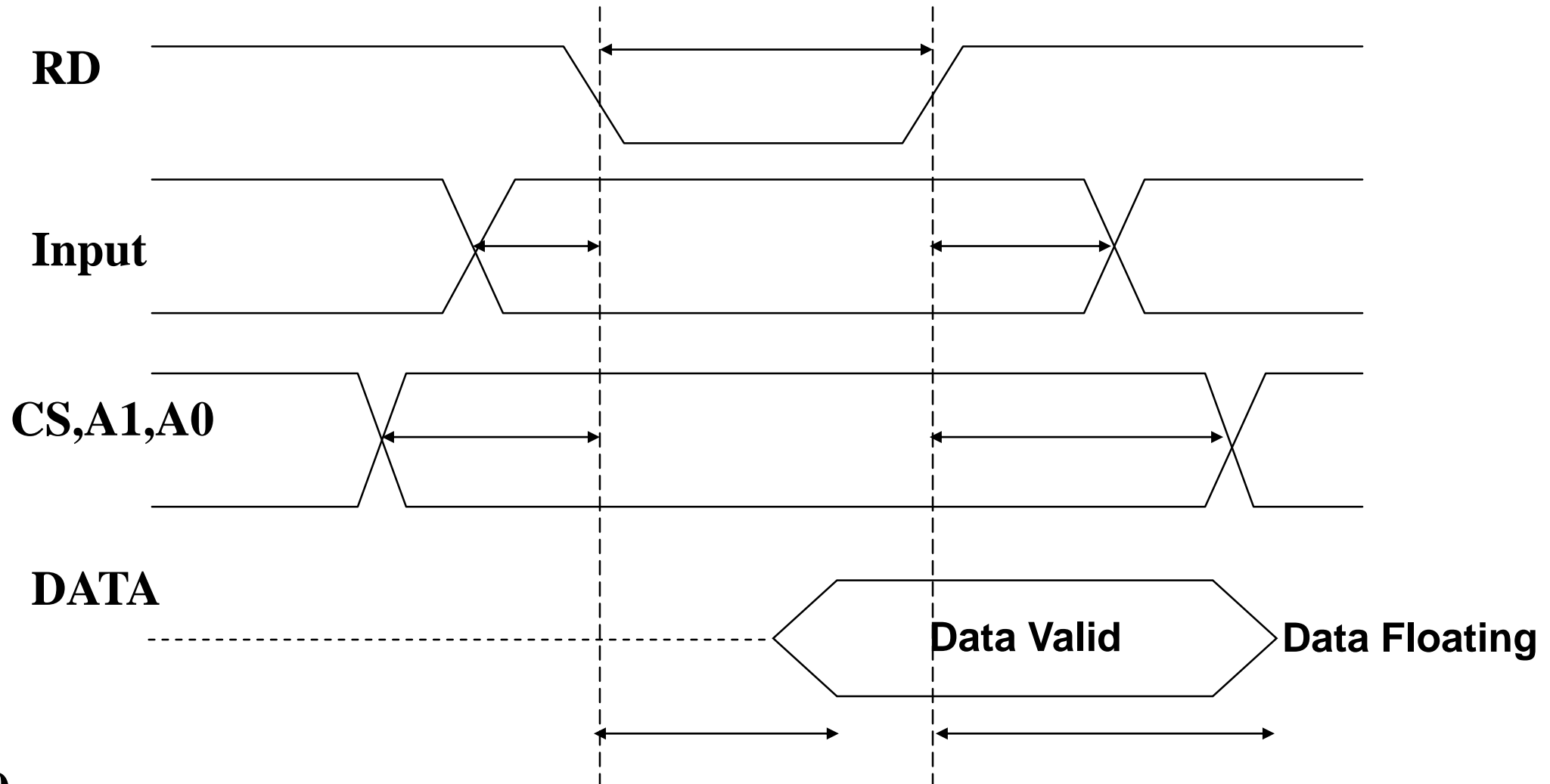
- 2 Porte da 8 bit e 2 porte da 4 bit
- Ciascuna porta può funzionare in input oppure in output
- Gli output sono *latched*
- Gli input non sono *latched*
- 16 possibili diverse configurazioni di I/O.

# Esempio

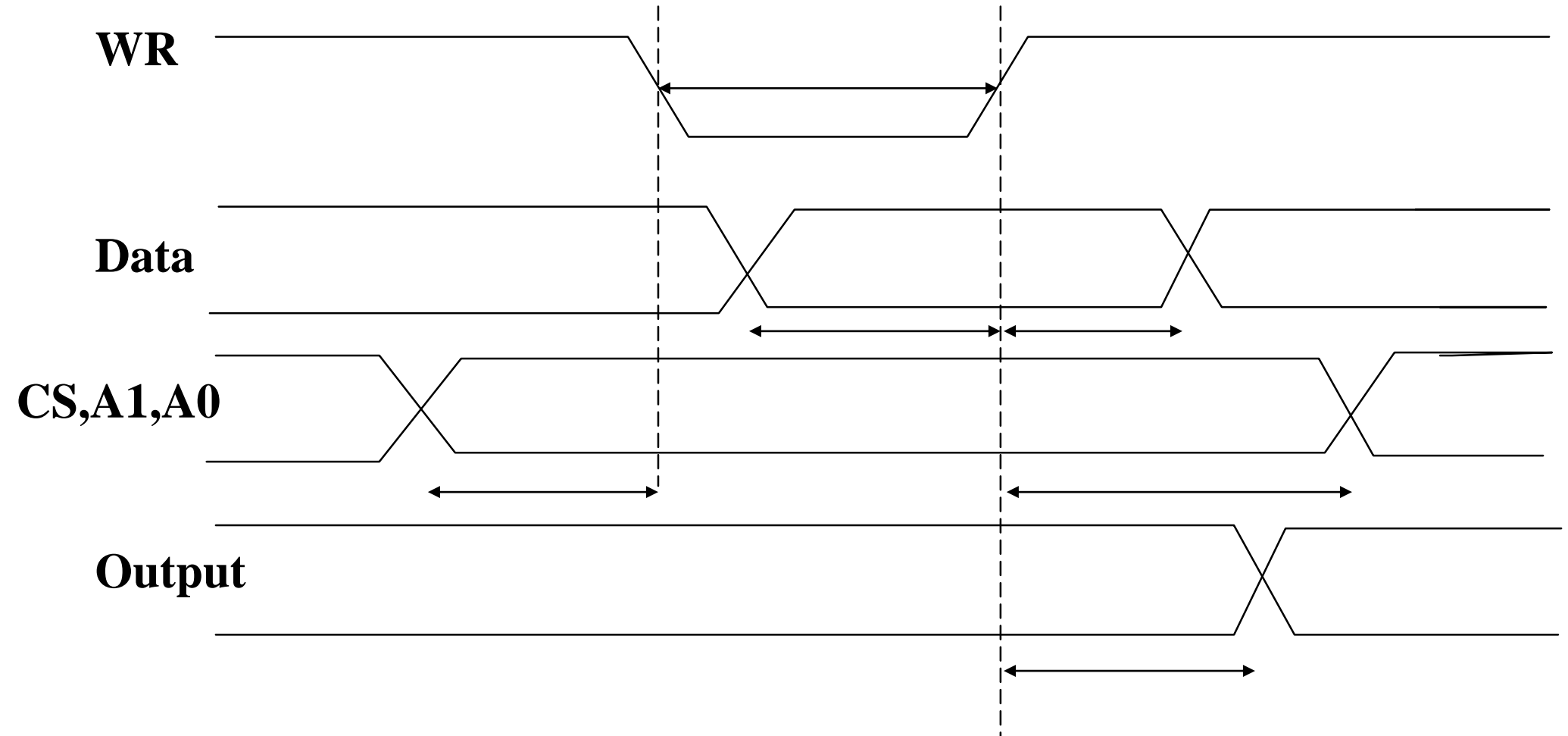
D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	0



# Modo 0 - Input



# Modo 0 - Output



# **Modo 1**

## ***(Strobed Input/Output)***

**Il trasferimento parallelo di dati è supportato da una serie di segnali di handshake che permettono di gestire le comunicazioni in *interrupt*.**

# Modo 1

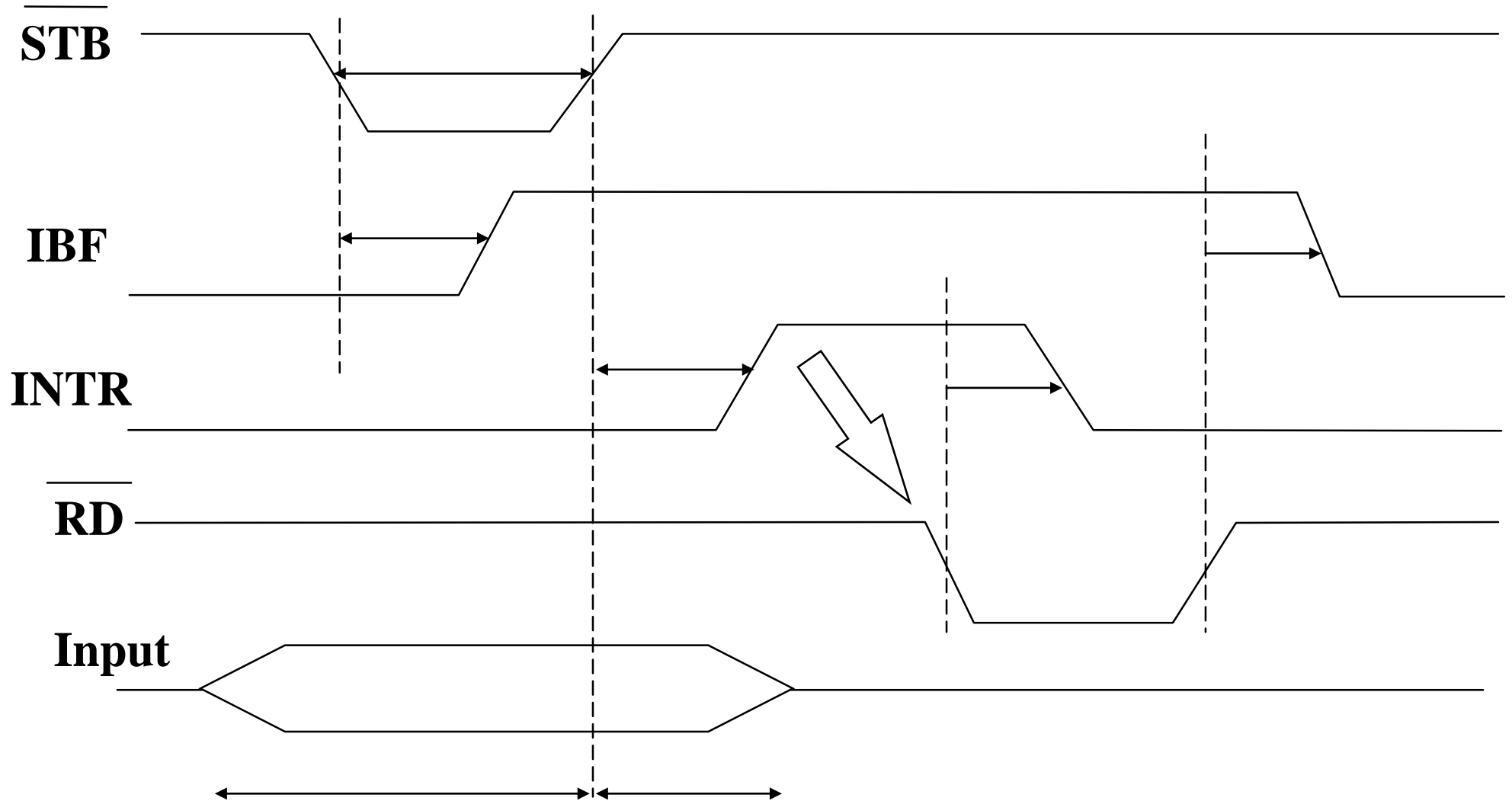
- **2 gruppi**
- **Ogni gruppo è composto da una porta di dato di 8 bit e da una porta di controllo di 4 bit**
- **I bit di dato possono essere di Input o di Output**
- **Input ed Output sono latched.**

# Segnali di controllo in input

- **STB (*Strobe Input*)**: un valore basso carica il dato nell'input latch
- **IBF (*Input Buffer Full*)**: un valore alto indica che il dato è stato caricato nell'input latch (funziona da acknowledge)
- **INTR (*Interrupt Request*)**: un valore alto può essere usato come richiesta di interrupt per la CPU
- **INTE<sub>A</sub> (*Interrupt Enable* per il gruppo A)**: controllato dal bit set/reset di PC4
- **INTE<sub>B</sub> (*Interrupt Enable* per il gruppo B)**: controllato dal bit set/reset di PC2.



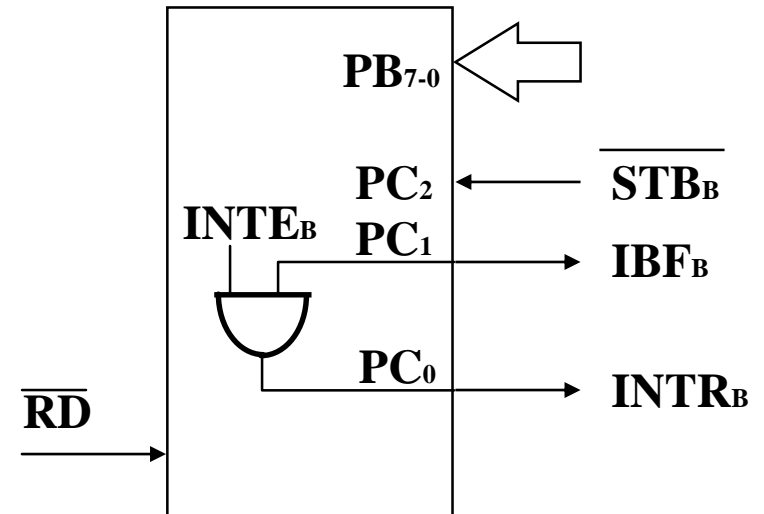
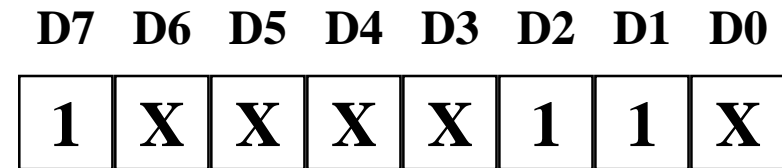
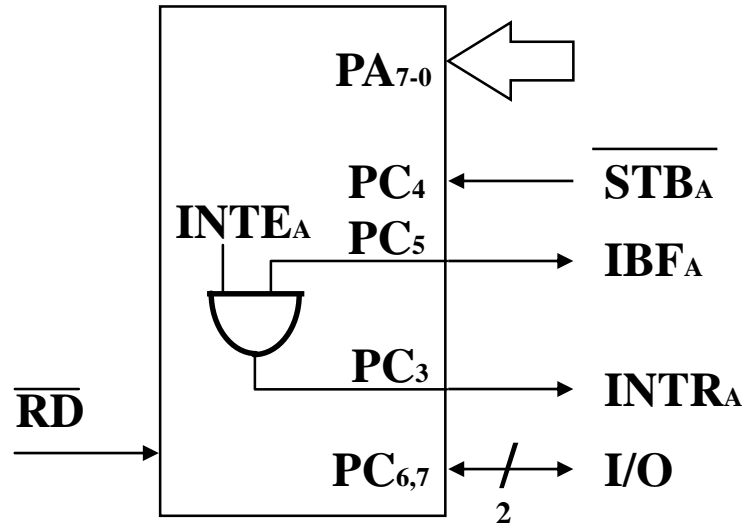
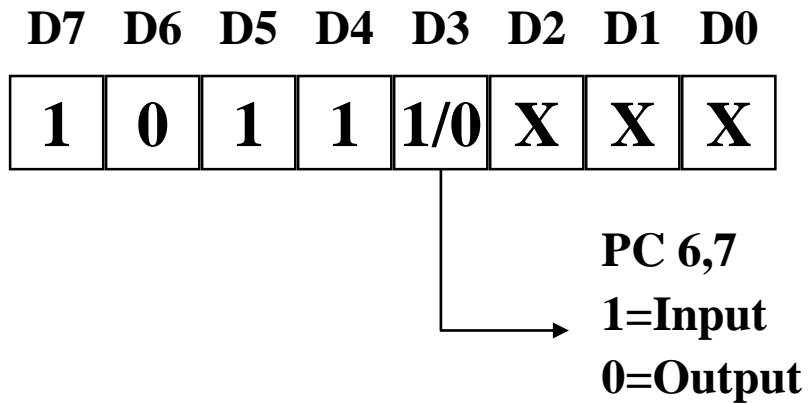
# Temporizzazioni



# Funzionamento

- **IBF è settato quando STB va basso, ed è resettato dal fronte di salita di RD**
- **INTR è settato quando STB va alto, IBF è alto, e INTE è settato; è resettato sul fronte di discesa di RD.**

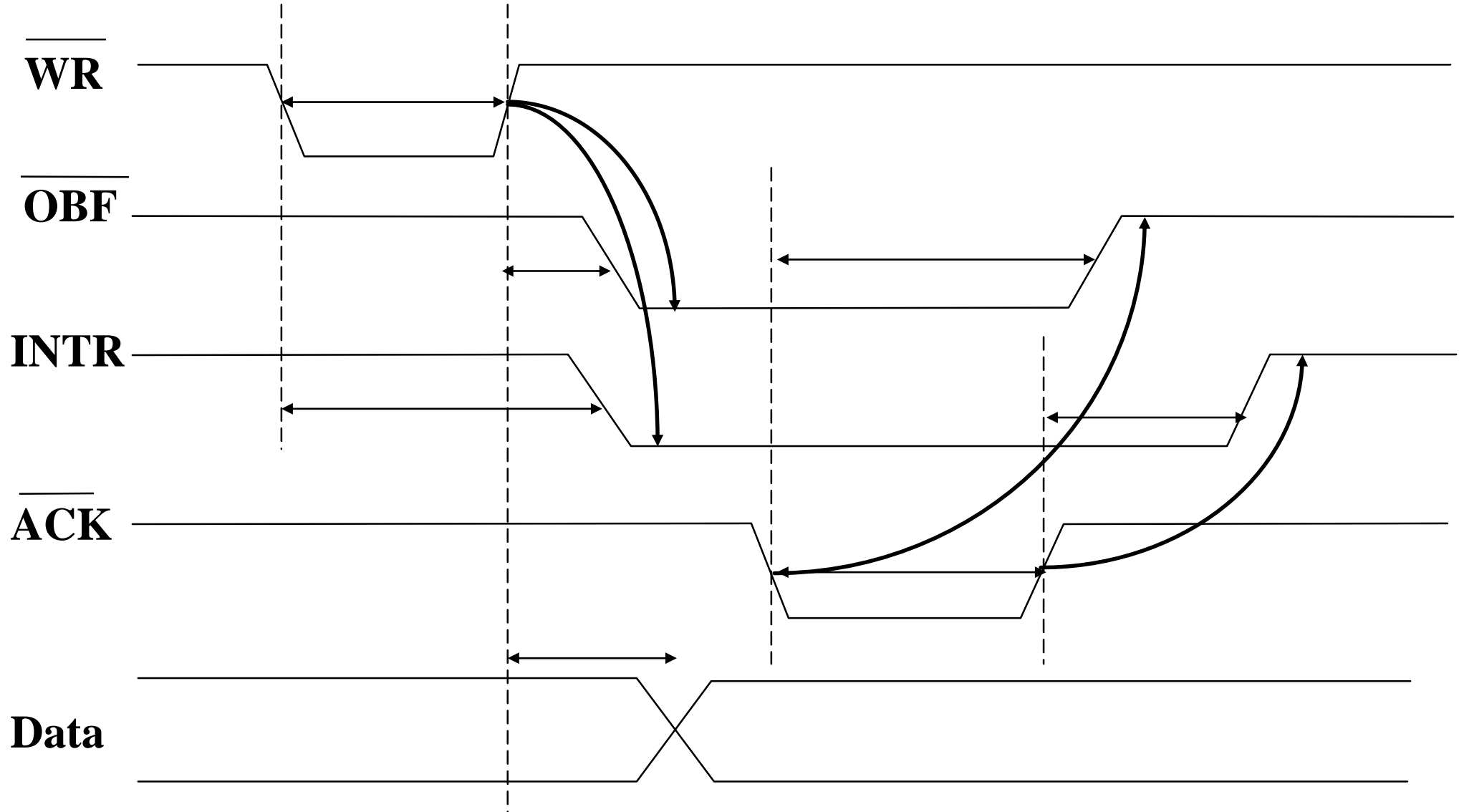
# Modo 1: Input



# Segnali di controllo di output

- **OBF (*Output Buffer Full*)**: un valore basso indica che la CPU ha scritto il dato sulla porta
- **ACK (*Acknowledge Input*)**: un valore basso informa l'8255 che il dato è stato ricevuto dalla periferica
- **INTR (*Interrupt Request*)**: un valore alto può essere usato come richiesta di interrupt per la CPU
- **INTE<sub>A</sub> (*Interrupt Enable* per il gruppo A)**: controllato dal bit set/reset di PC<sub>6</sub>
- **INTE<sub>B</sub> (*Interrupt Enable* per il gruppo B)**: controllato dal bit set/reset di PC<sub>2</sub>.

# Temporizzazioni



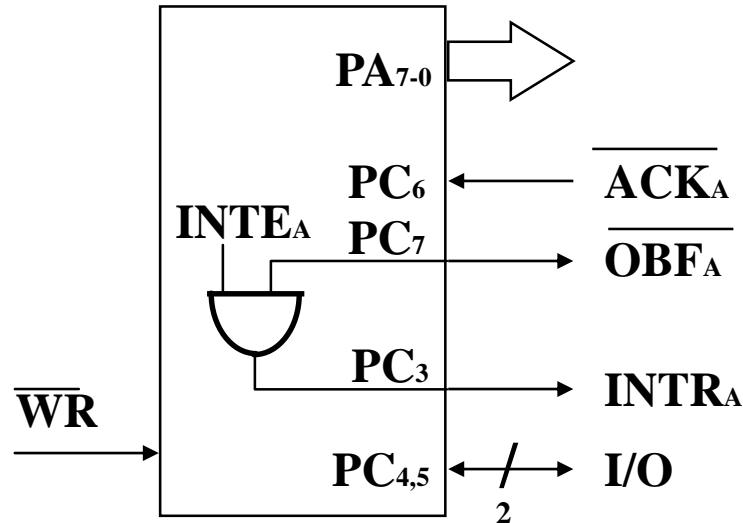
# Funzionamento

- **OBF** è settato sul fronte di salita di **WR** ed è resettato quando **ACK** diventa basso
- **INTR** è resettato sul fronte di discesa di **WR**, ed è settato quando **ACK** è alto, **OBF** è alto e **INTE** è settato.

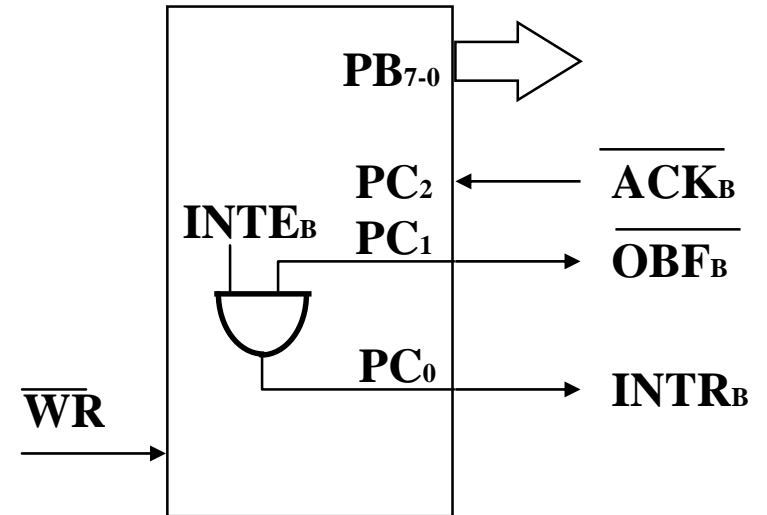
# Modo 1: Output

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



# **Modo 2**

## ***(Bidirectional I/O)***

**Questo modo di funzionamento permette di gestire una porta bidirezionale di I/O supportata da segnali di handshake.**



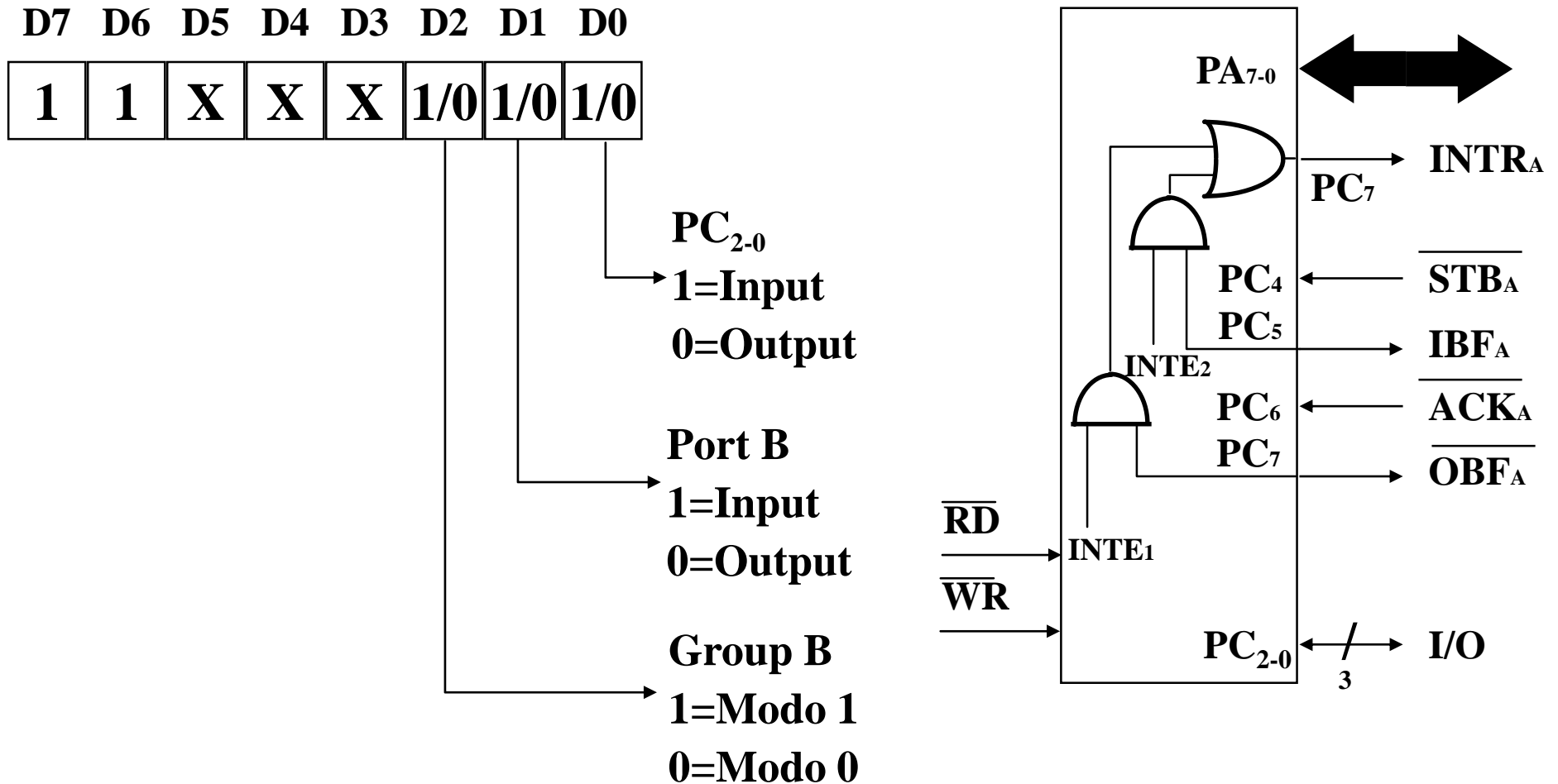
# Modo 2

- **Disponibile solo per il Gruppo A**
- **1 porta bidirezionale ad 8 bit (port A) ed 1 porta di controllo a 5 bit (port C)**
- **Input ed Output sono latched.**

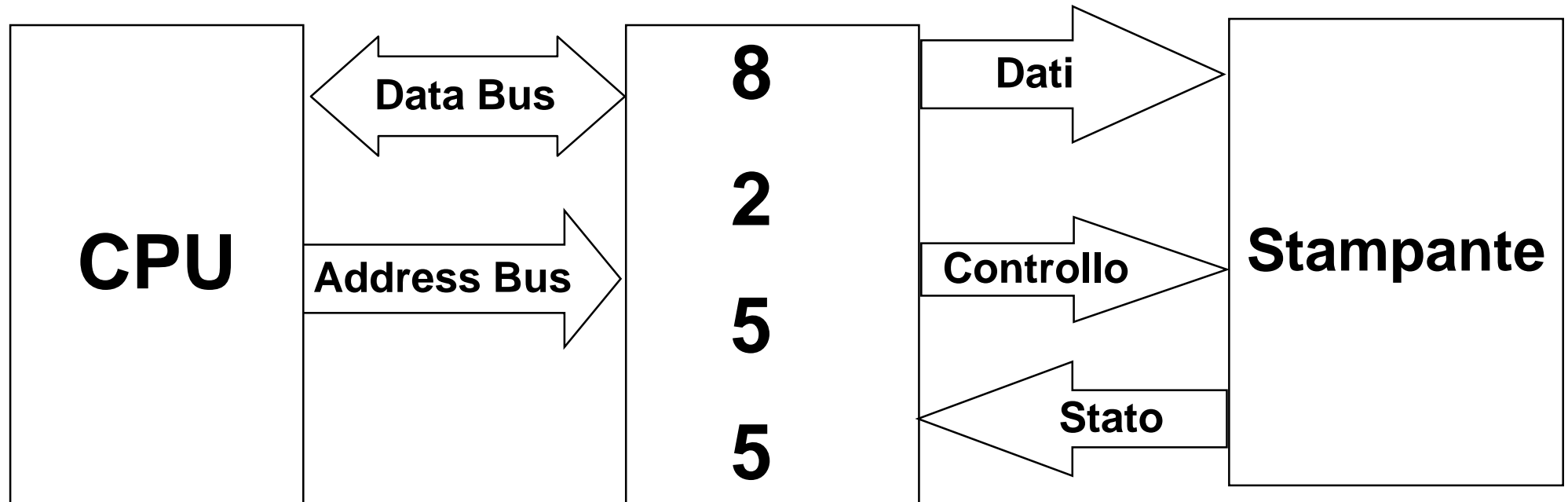
# Segnali di controllo

- **INTR:** un valore alto può essere usato per una richiesta di interrupt per la CPU
- **OBF:** un valore basso indica che la CPU ha scritto un dato sulla porta A
- **ACK:** un valore basso abilita ad inviare il dato
- **STB:** un valore basso carica il dato nell'input latch
- **IBF:** un valore alto indica che il dato è stato caricato sull'input latch
- **INTE<sub>1</sub>** (Interrupt Enable): controllato dal bit set/reset di PC<sub>6</sub>
- **INTE<sub>2</sub>** (Interrupt Enable): controllato dal bit set/reset di PC<sub>4</sub>.

# Modo 2



# Esempio: interfaccia PC-stampante



# Cavo parallelo *Centronics*

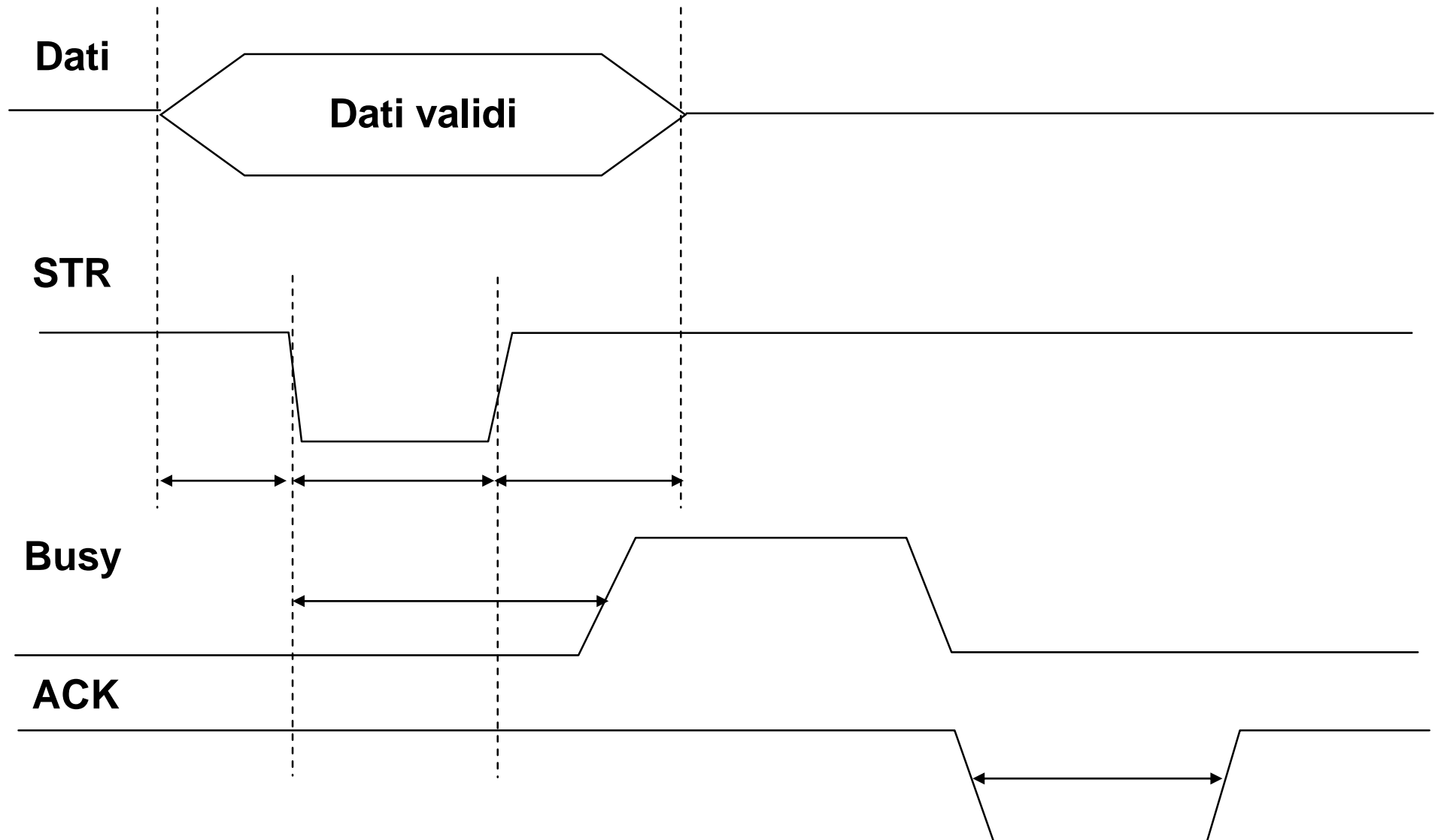
Il cavo parallelo Centronics è composto dai seguenti piedini:

- **8 bit di Dato**
- **STROBE:** un impulso basso di durata superiore ad 1 microsecondo scrive sulla stampante un byte di dati
- **ACK:** la stampante invia un impulso basso per segnalare l'avvenuta ricezione dei dati
- **BUSY:** un livello logico alto indica che la stampante non può ricevere dati
- **AUTOFDX:** comanda alla stampante di andare alla pagina successiva

# Cavo parallelo *Centronics* (II)

- **ERRORE CARTA:** un livello logico alto indica che la carta è esaurita
- **ON LINE:** un livello alto indica che la stampante è attiva
- **HI:** la stampante emette un livello logico alto mentre viene accesa
- **REIMPOSTA:** un impulso basso reimposta la stampante e svuota il buffer di stampa
- **ERRORE:** un livello logico basso indica che la stampante si trova in stato di errore.

# Protocollo di comunicazione



# **Interfaccia 8255-Centronics**

**Porta A: parola di dato**

**Porta B: stato della stampante**

**Porta C: controllo della stampante.**



# **Porta B:**

## **stato della stampante**

- **PB7: BUSY** (la stampante non può accettare nuovi dati)
- **PB6: ACK** (ricezione dei dati, stampante pronta)
- **PB5: PE** (fine carta)
- **PB4: SLCT** (On line)
- **PB3: ERR** (condizione di errore).

# Porta C:

## controllo della stampante

- **PC0: STB** (impulso per trasferire i dati)
- **PC1: AUTOFDX** (*autofeed*, avanzamento carta di una riga)
- **PC2: INIT** (inizializza stampante)
- **PC3: SLCTIN** (abilita la stampante ad accettare nuovi dati).

# Programma

prDATA	EQU	0378h	; indirizzo LPT1
prSTAT	EQU	prDATA+1	; porta B
prCTRL	EQU	prDATA+2	; porta C
DELAY	EQU	100	; ritardo
sERR	EQU	08h	; attivo BASSO
sSEL	EQU	10h	; attivo ALTO
sPE	EQU	20h	; attivo ALTO
sACK	EQU	40h	; attivo BASSO
sBUSY	EQU	80h	; attivo BASSO
cSTB	EQU	01h	; attivo ALTO
cAUTO	EQU	02h	; attivo ALTO
cINIT	EQU	04h	; attivo BASSO
43 cSEL	EQU	08h	; attivo ALTO

```

.MODEL      small

.STACK

.DATA

msg         DB      'Ciao a tutti',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      ;stampa un carattere
          INC  SI
          JMP  next

done:      .EXIT

```

```

pr_al      PROC
            PUSH DX
            PUSH AX
            MOV  AH, AL      ; salva AL in AH
            MOV  DX, prSTAT
pr_n_ready: IN   AL, DX      ; legge lo stato
                                   ; della stampante
            TEST AL, sERR
            JZ   pr_error    ; errore
            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      ; invia il dato
        MOV    CX, DELAY
ciclo1:  LOOP   ciclo1
        MOV    DX, prCTRL
        IN     AL, DX
        OR     AL, cSTB    ; setta il bit di strobe
        OUT    DX, AL
        MOV    CX, 2*DELAY
ciclo2:  LOOP   ciclo2
        AND    AL, not cSTB ; resetta lo strobe
        OUT    DX, AL
        MOV    CX, DELAY
ciclo3:  LOOP   ciclo3

```

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