

8255

(interfaccia parallela)

M. Rebaudengo - M. Sonza Reorda

Politecnico di Torino
Dip. di Automatica e Informatica

1

M. Rebaudengo, M. Sonza Reorda

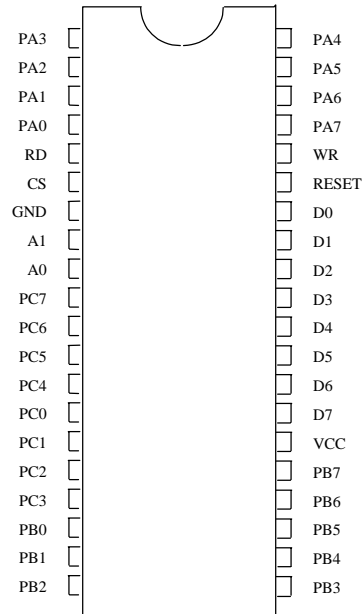
Generalità

- L'Intel 8255 implementa un'interfaccia di I/O parallela per sistemi delle famiglie 8085 e 8086
- È realizzato tramite un chip LSI di tipo DIP a 40 pin
- 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
- Il data sheet è disponibile all'indirizzo <http://developer.intel.com/design/periphrl/datashts/231256.htm>.

2

M. Rebaudengo, M. Sonza Reorda

Il Chip



3

M. Rebaudengo, M. Sonza Reorda

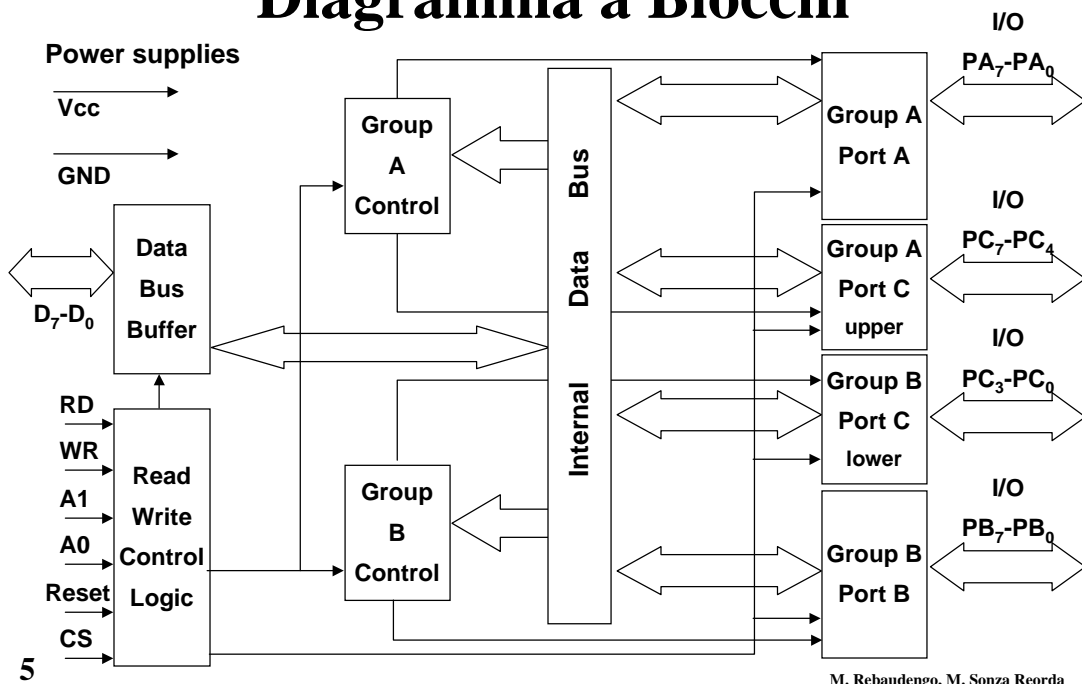
Descrizione dei Pin

- 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

4

M. Rebaudengo, M. Sonza Reorda

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 ed 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.

7

M. Rebaudengo, M. Sonza Reorda

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	Registro
0	0	Porta A
0	1	Porta B
1	0	Porta C
1	1	Registro di Controllo

8

M. Rebaudengo, M. Sonza Reorda

8255 Input/Output Operation

A1	A0	RD	WR	CS	
0	0	0	1	0	Porta A ⇒ Data Bus
0	1	0	1	0	Porta B ⇒ Data Bus
1	0	0	1	0	Porta C ⇒ Data Bus
0	0	1	0	0	Data Bus ⇒ Porta A
0	1	1	0	0	Data Bus ⇒ Porta B
1	0	1	0	0	Data Bus ⇒ Porta 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	Non ammessa
X	X	0	0	X	Non ammessa

9

M. Rebaudengo, M. Sonza Reorda

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.

10

M. Rebaudengo, M. Sonza Reorda

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).

11

M. Rebaudengo, M. Senza Reorda

Parola di Controllo

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

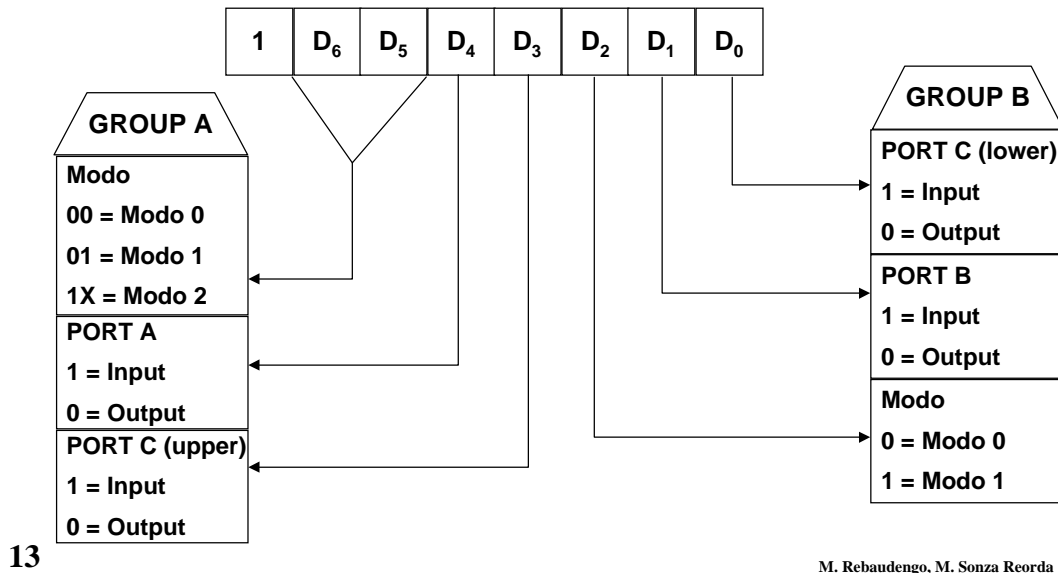
Può avere due funzioni:

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

12

M. Rebaudengo, M. Senza Reorda

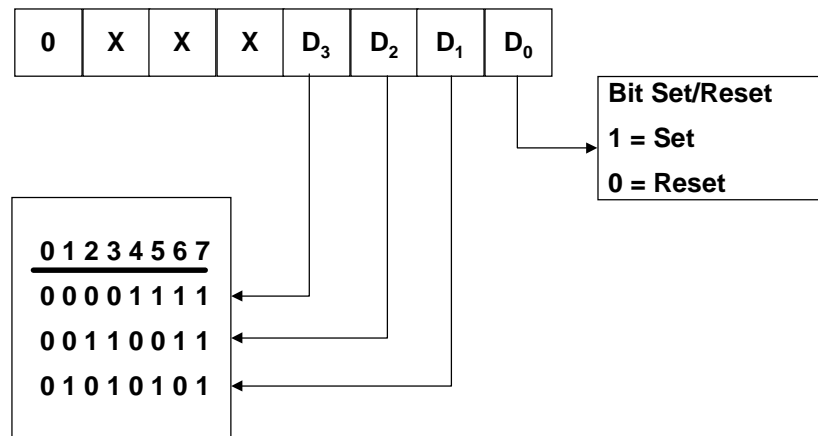
Programmazione 8255



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



15

M. Rebaudengo, M. Sonza Reorda

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.

16

M. Rebaudengo, M. Sonza Reorda

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*.

17

M. Rebaudengo, M. Sonza Reorda

Modo 0

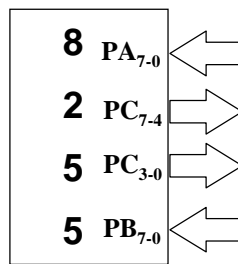
- 2 Porte da 8 bit e 2 porte da 4 bit.
- Ciascuna porta può funzionare in input oppure in output.
- Gli output sono *latched*.
- Gli input non sono *latched*.
- 16 possibili diverse configurazioni di I/O.

18

M. Rebaudengo, M. Sonza Reorda

Esempio

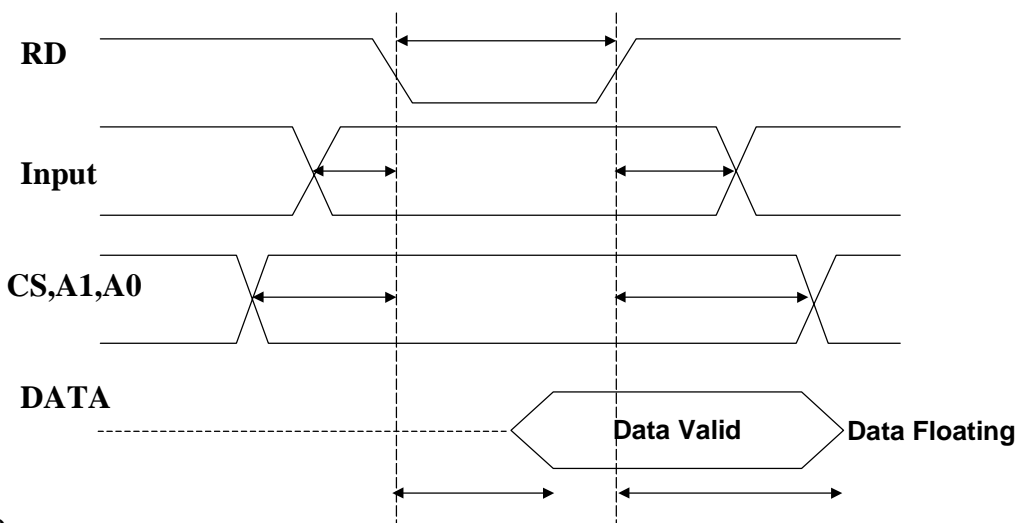
D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	0



19

M. Rebaudengo, M. Sonza Reorda

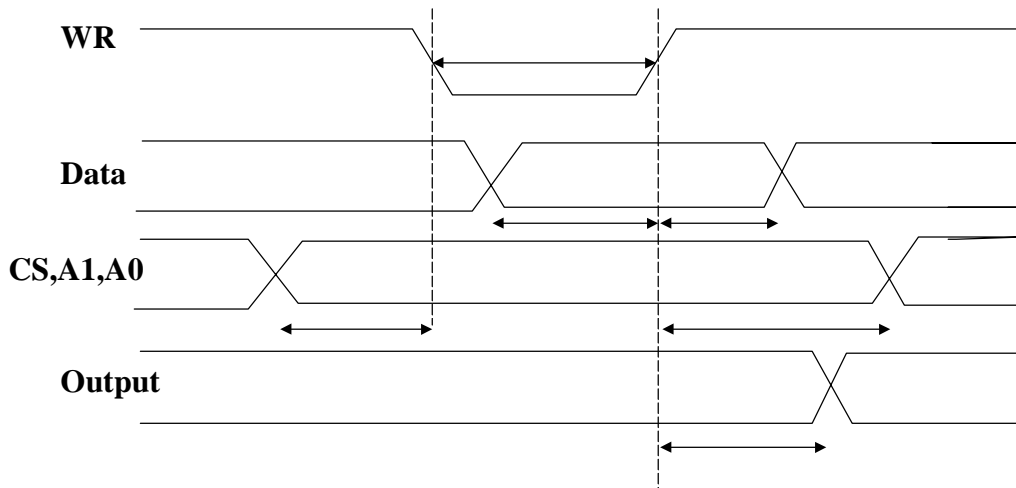
Modo 0 - Input



20

M. Rebaudengo, M. Sonza Reorda

Modo 0 - Output



21

M. Rebaudengo, M. Sonza Reorda

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*.

22

M. Rebaudengo, M. Sonza Reorda

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.

23

M. Rebaudengo, M. Sonza Reorda

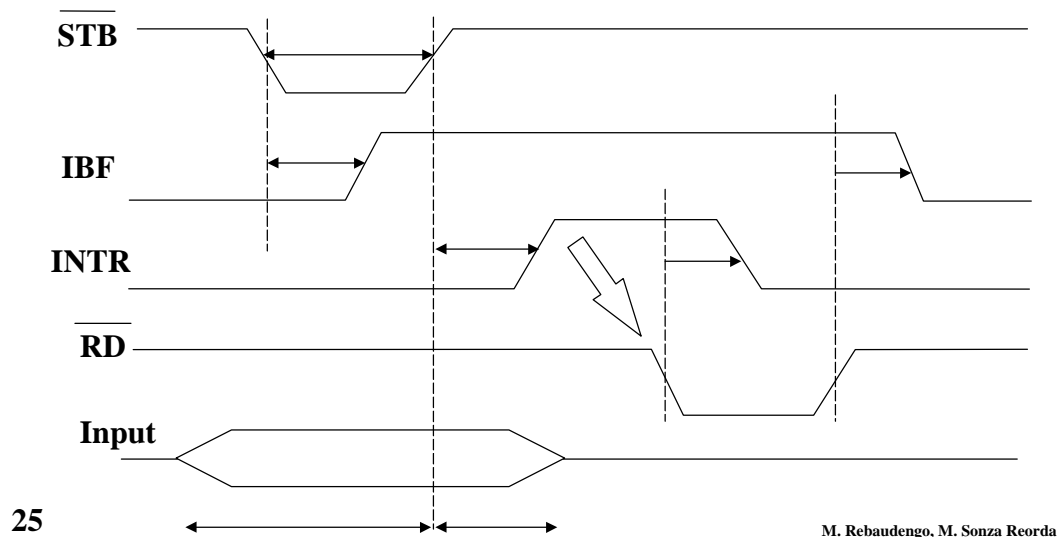
Segnali di Controllo in Input

- **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_A** (*Interrupt Enable per il gruppo A*): controllato dal bit set/reset di PC4.
- **INTE_B** (*Interrupt Enable per il gruppo B*): controllato dal bit set/reset di PC2.

24

M. Rebaudengo, M. Sonza Reorda

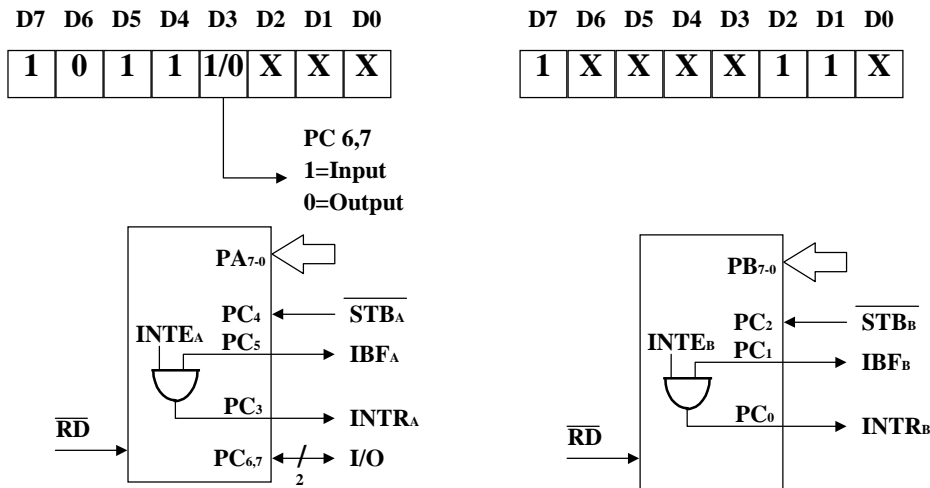
Temporizzazioni



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



27

M. Rebaudengo, M. Sonza Reorda

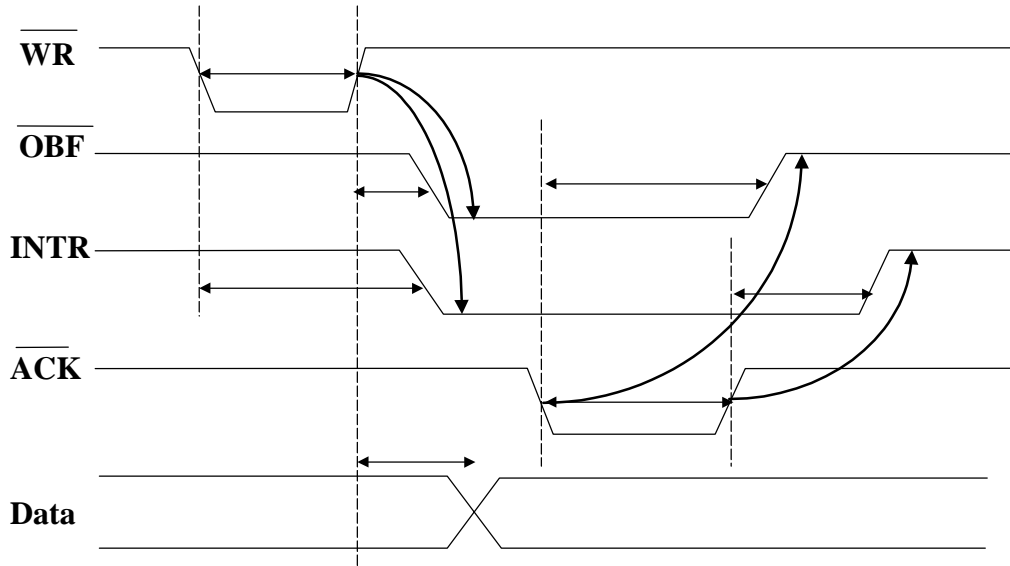
Segnali di Controllo di Output

- **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_A (Interrupt Enable per il gruppo A):** controllato dal bit set/reset di PC6.
- **INTE_B (Interrupt Enable per il gruppo B):** controllato dal bit set/reset di PC2.

28

M. Rebaudengo, M. Sonza Reorda

Temporizzazioni



29

M. Rebaudengo, M. Sonza Reorda

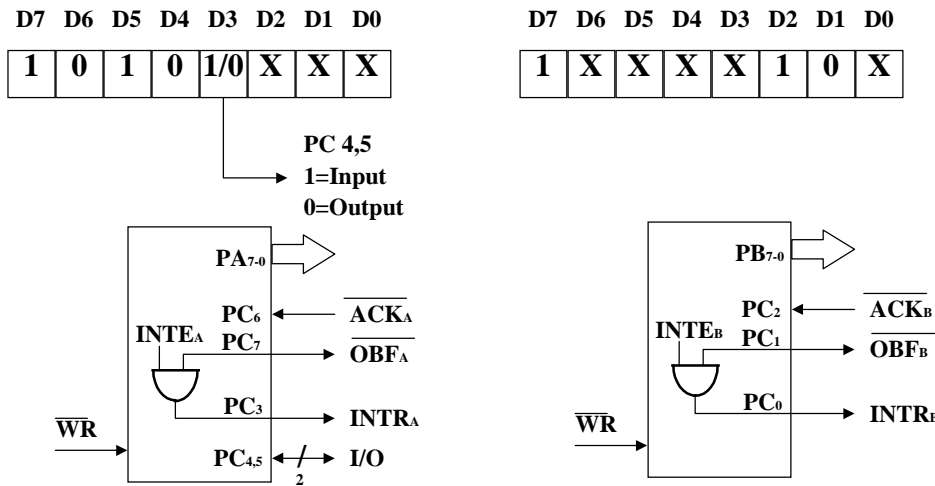
Funzionamento

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

30

M. Rebaudengo, M. Sonza Reorda

Modo 1: Output



31

M. Rebaudengo, M. Sonza Reorda

Modo 2 (Bidirectional I/O)

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

32

M. Rebaudengo, M. Sonza Reorda

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.

33

M. Rebaudengo, M. Sonza Reorda

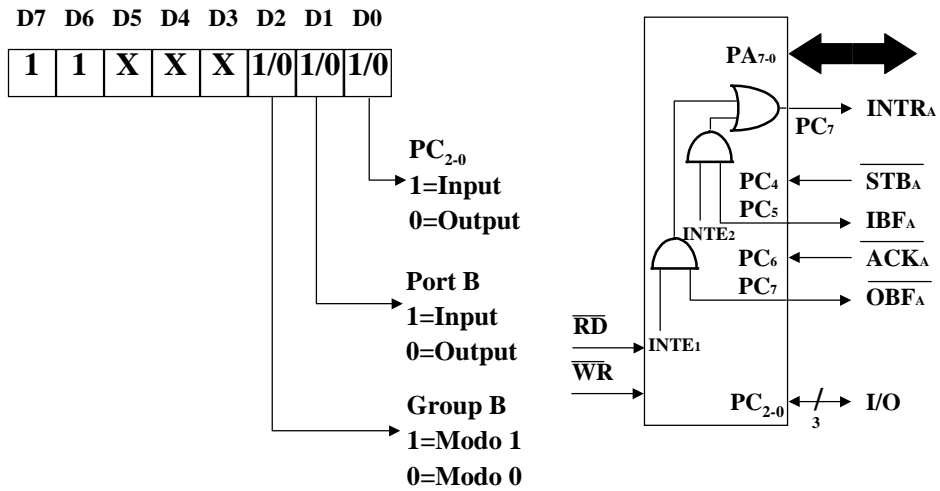
Segnali di controllo

- **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.
- **INTE₁ (Interrupt Enable):** controllato dal bit set/reset di PC₆.
- **INTE₂ (Interrupt Enable):** controllato dal bit set/reset di PC₄.

34

M. Rebaudengo, M. Sonza Reorda

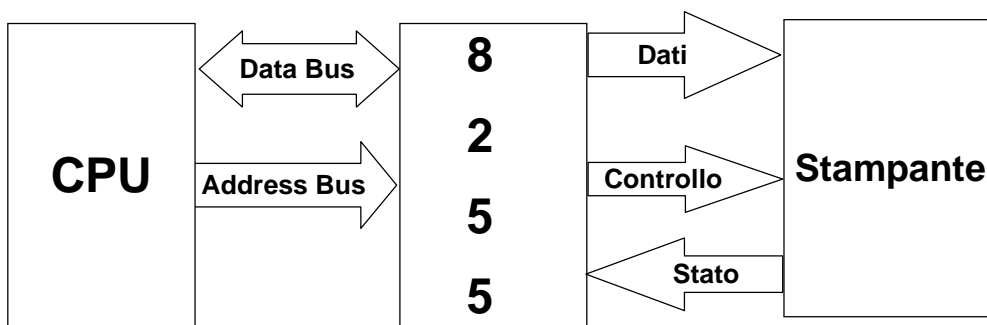
Modo 2



35

M. Rebaudengo, M. Sonza Reorda

Esempio: interfaccia PC-Stampante



36

M. Rebaudengo, M. Sonza Reorda

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;

37

M. Rebaudengo, M. Sonza Reorda

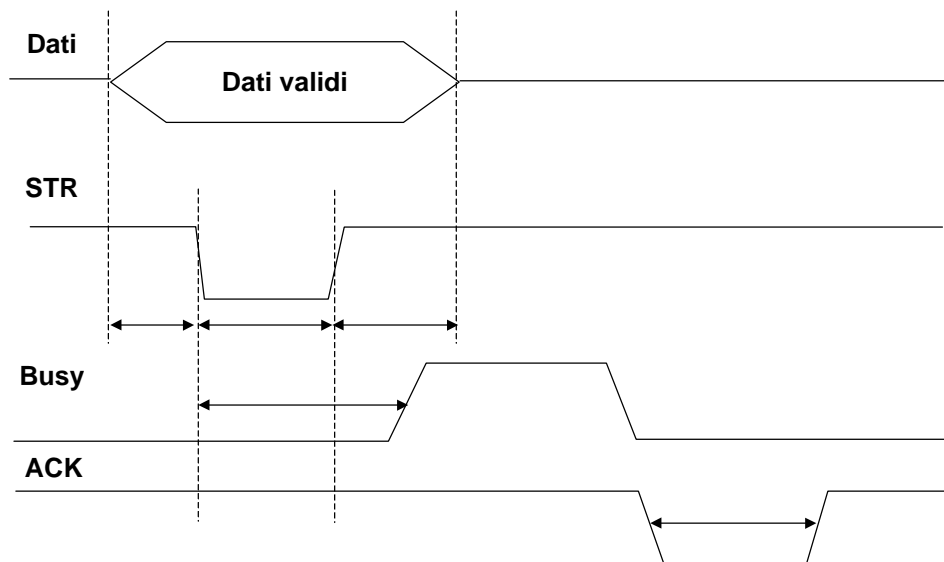
Cavo Parallelo *Centronics* (II)

- **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.

38

M. Rebaudengo, M. Sonza Reorda

Protocollo di comunicazione



39

M. Rebaudengo, M. Sonza Reorda

Interfaccia 8255-Centronics

Porta A: Parola di Dato

Porta B: Stato della stampante

Porta C: Controllo della stampante.

40

M. Rebaudengo, M. Sonza Reorda

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).

41

M. Rebaudengo, M. Senza Reorda

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).

42

M. Rebaudengo, M. Senza Reorda

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

```

M. Rebaudengo, M. Sonza Reorda

```

.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

```

44

M. Rebaudengo, M. Sonza Reorda

```

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

```

45

M. Rebaudengo, M. Sonza Reorda

```

               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

```

46

M. Rebaudengo, M. Sonza Reorda

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