

PROGETTO DI INTERFACCIA DI I/O PER IL BUS DEL PC.

Scopo del progetto è la realizzazione di una scheda di interfaccia di I/O che, inserita in uno degli slot liberi di un computer XT o AT, consente di effettuare trasferimenti di dati a 8 bit in ingresso o in uscita, da o verso un dispositivo periferico esterno.

IL BUS DEL PC IBM.

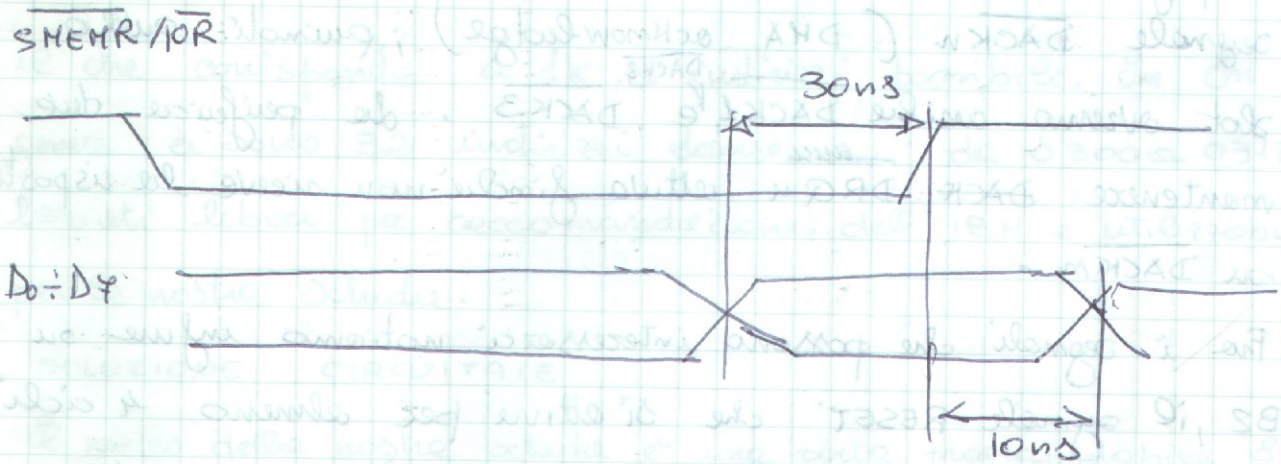
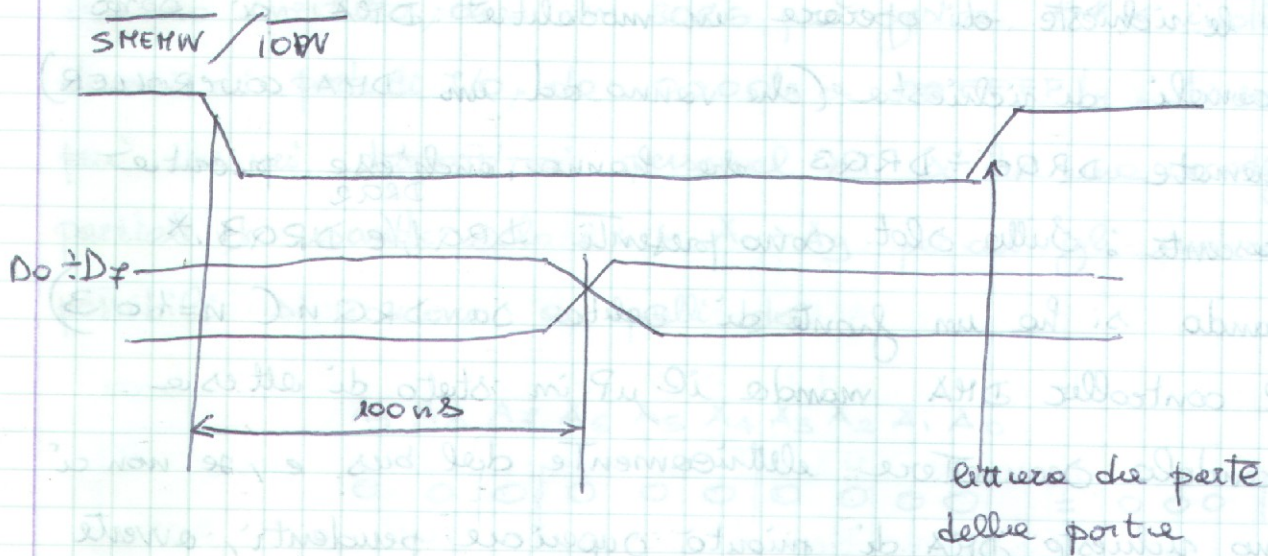
Un personal presente sulla scheda madre connette lo slot sui quali esistono i segnali del bus di sistema. In fig 1 è rappresentato uno slot con i vari segnali. \times

Notiamo sulla destra i piedini $A_{12} \div A_{31}$ ai quali esistono i 20 bit del bus indirizzi $A_0 \div A_9$ dell'8086 (il quale può fornire indirizzi a 20 bit e indirizza quindi $2^{20} = 1$ Megabyte di memoria); sui piedini $A_{2} \div A_9$ troviamo invece, gli 8 bit del bus dati $D_0 \div D_7$ (in realtà il bus dati dell'8086 è a 16 bit per cui eventuali trasferimenti di parole a 16 bit avvengono inviando prima il byte basso e poi il byte alto). Sui piedini B_{11} e B_{12} (a sinistra) sono presenti i segnali \overline{MEMW} e \overline{MEMR} che sono i segnali di scrittura e lettura in memoria. Sui piedini B_{14} e B_{13} sono presenti invece \overline{IOW} (che serve per scrivere su una porta di I/O) e \overline{IOR} (che serve per leggere da una porta). Per una operazione di scrittura (\overline{MEMW} o \overline{IOW}) si abbassa il segnale e dopo circa 100 ns i dati si stabilizzano su $D_0 \div D_7$; il dispositivo esterno dovrebbe leggerli sul fronte di risalita. Per la lettura invece

FIG. 56

Ground	B1		A1	I/O CH CK I/O Channel C
Reset (To Card) RESET DRV	B2		A2	SD7 Data 7
+5 Volts DC	B3		A3	SD6 Data 6
Interrupt Request IRQ 2	B4		A4	SD5 Data 5
-5 Volts DC	B5		A5	SD4 Data 4
DMA Request From Card DRQ2	B6		A6	SD3 Data 3
-12 Volts DC	B7		A7	SD2 Data 2
Zero Wait State (From Card) OWS	B8		A8	SD1 Data 1
+12 Volts DC	B9		A9	SD0 Data 0
Ground	B10		A10	I/O CH RDY I/O Channel R
Card Memory Write <u>SMEMW</u>	B11		A11	AEN Address Enable (DMA)
Card Memory Read <u>SMEMR</u>	B12		A12	SA19 Address 19
Card I/O Write <u>IOR</u>	B13		A13	SA18 Address 18
Card I/O Read <u>IOW</u>	B14		A14	SA17 Address 17
DMA Request Acknowledge <u>DACK3</u>	B15		A15	SA16 Address 16
DMA Request From Card <u>DRQ3</u>	B16		A16	SA15 Address 15
DMA Request Acknowledge <u>DACK1</u>	B17		A17	SA14 Address 14
DMA Request From Card <u>DRQ1</u>	B18		A18	SA13 Address 13
Memory Refresh Occurring <u>REFRESH</u>	B19		A19	SA12 Address 12
System Clock From Motherboard CLK	B20		A20	SA11 Address 11
Interrupt Request IRQ 7	B21		A21	SA10 Address 10
Interrupt Request IRQ 6	B22		A22	SA9 Address 9
Interrupt Request IRQ 5	B23		A23	SA8 Address 8
Interrupt Request IRQ 4	B24		A24	SA7 Address 7
Interrupt Request IRQ 3	B25		A25	SA6 Address 6
DMA Request Acknowledge <u>DACK 2</u>	B26		A26	SA5 Address 5
DMA Terminal Count Pulse T/C	B27		A27	SA4 Address 4
Buffered Address Latch Enable BALE	B28		A28	SA3 Address 3
+ 5 Volts DC	B29		A29	SA2 Address 2
14.31818 Mhz Clock From Motherboard OSC	B30		A30	SA1 Address 1
Ground	B31		A31	SA0 Address 0

si ha che prima delle uscite di \overline{SHEWR} o \overline{IOR} i dati



devono essere stabilizzati da almeno 30ns e persistere dopo la risalita per almeno altri 10ns.

Il μP 8086 gestisce le richieste di interruzione hardware mediante un dispositivo esterno detto INTERRUPT CONTROLLER che mette a disposizione dei dispositivi esterni 8 linee a priorità decrescente da $IRQ0$ a $IRQ7$. $IRQ0$ e $IRQ1$ non sono presenti sullo slot. Di quelle presenti sullo slot restano libere e quindi disponibili per l'utente solo $IRQ3$ ed $IRQ4$; però, se c'è il mouse, questo utilizza

una delle due. Per le richieste di operare in modalità DMA ci sono 4 canali di richiesta (che vanno ad un DMA CONTROLLER) chiamati DRQ0 ÷ DRQ3 che hanno, anch'esse, priorità decrescente. Sullo slot sono presenti DRQ1, e DRQ3.*

Quando si ha un fronte di salita su DRQn (n=1 o 3) il controller DMA manda il μP in stato di attesa facendolo sconnettere elettricamente dal bus e, se non ci sono richieste DMA di priorità superiore pendenti, avverte la periferica che la richiesta è accettata mediante un segnale \overline{DACKn} (DMA acknowledge); quindi sullo slot avremo anche $\overline{DACK1}$, e $\overline{DACK3}$. La periferica deve mantenere \overline{DACKn} attivo finché non riceve la risposta su \overline{DACKm} .

Fra i segnali che possono interessarci notiamo infine su B2, il segnale RESET che si attiva per almeno 4 cicli di clock all'accensione o in un reset software e può essere utilizzato per resettare il dispositivo esterno collegato allo slot.

Infine c'è il segnale AEN che è ad 1 se è il controller DMA a gestire i trasferimenti di dati mentre è a 0 se è il μP a gestire i trasferimenti.

* però DRQ2 è usato dal sistema, per cui l'utente ha a disposizione solo DRQ1 e DRQ3

INDIRIZZI DI I/O POSSIBILI

Diciamo anzitutto, che un 8086 può gestire 64 K indirizzi diversi di porte di I/O, da $0000H$ a $FFFFH$. Avvane, però, che nei dispositivi personal XT si ha una decodifica parziale di usando solo i bit $A_0 \div A_9$ per cui gli indirizzi possibili si riducono a quelli da

$A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$

$0 0 | 0 0 | 0 0 0 0 = 000 | H$

$1 1 | 1 1 | 1 1 1 1 = 3FF | H$

ie che corrisponde a 1K di indirizzi possibili. In In genere ci sono 32 indirizzi consecutivi da 0300 a $031FH$ lasciati liberi per raccomandazione dell'IBM e utilizzabili per le nostre schede.

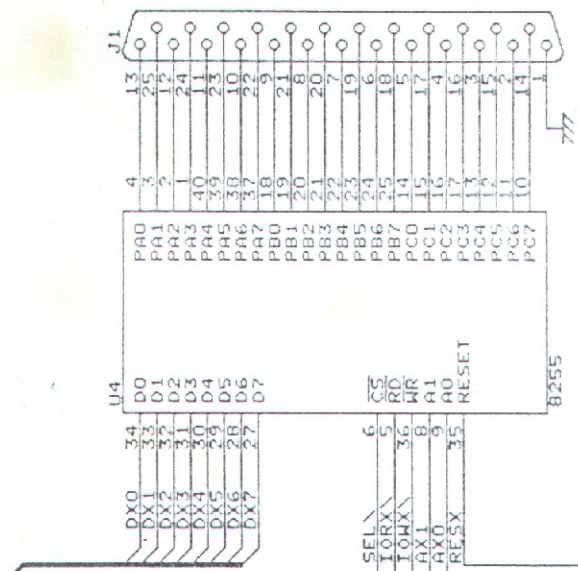
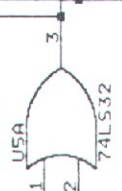
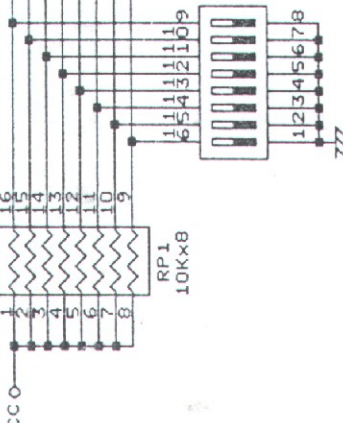
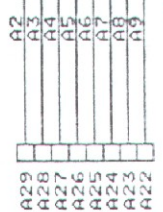
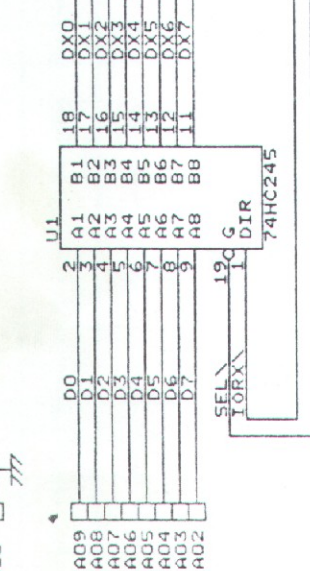
SOLUZIONE CIRCUITALE

Il nucleo delle nostre schede è una porta programmabile 8255. I bit di selezione delle porte sono ricevuti dai bit A_1 e A_0 del bus, in modo che se l'indirizzo delle porte A (IPA) è IPA , quello delle B sarà $IPA+1$, quello delle C $IPA+2$ e quello dell'registro di controllo $IPA+3$. Gli altri bit da A_2 ad A_9 del bus indirizzi vanno all'integrato 74LS688 il quale è un comparatore di dati a 8 bit ed è in grado di segnalare, ponendo a zero il suo piedino 18, se i due dati sono uguali. L'integrato ha un \overline{sel} ingresso di abilitazione che viene posto a meno per tenerlo sempre attivato

L'ingresso a 8 bit da comparatore con i bit $A_2 \div A_9$ è ricevuto mediante 8 interruttori secondo la solita tecnica per cui l'utente può posizionare a piacere l'indirizzo delle schede (l'indirizzo base IPA dell'8255). L'uscita del comparatore va in ingresso ad una OR, il cui altro ingresso ha il segnale AEN per cui l'8255 è effettivamente abilitato soltanto quando è il μP ad operare, cioè non siamo in modalità DMA (cioè AEN=0). Gli ingressi RD e WR dell'8255 sono ricevuti da \overline{IOR} e \overline{IOW} e il RESET del RESET dello slot. Un 742524 bufferizza AEN, \overline{IOR} , \overline{IOW} , A1, A0, e RESET. I suoi ingressi di abilitazione $\overline{1A_0}$ e $\overline{2A_0}$. Gli ingressi 2A3 e 2A4 non utilizzati vanno posti e messi per il solito problema dei disturbi generati dalle porte i cui ingressi siano lasciati liberi.

I bit dati: $D_0 \div D_7$ sono bufferizzati da un 74245 bidirezionale (poiché i dati devono poter viaggiare in entrambe le direzioni). Il buffer è attivato mediante l'uscita della OR, mentre il segnale DIR è ricevuto da \overline{IOR} , così se $\overline{IOR}=0$, cioè il μP sta leggendo, DIR=0 e quindi i dati vanno dall'8255 al μP e viceversa.

B03-B29
B01, B10
B31



Connettore esterno

VCC	GND	Integrati
10	20	U1, U2, U3
7	26	U4
7	14	U5

Lab. IDP

Title		Interfaccia di I/O programmabile
Size	Document Number	TAV.5.1
REV		01
Date:	Sheet	1 of 1

COLLAUDO

LE

ricalca lo schema a blocchi
elencati in tab. 2;

ressi P di un comparatore
vece collegato un dip-switch
ccia. Fissando ad esempio
aratore si attiverà in corri-
lo l'uscita del comparatore
a segnale di selezione (SEL)

uffer bidirezionali contenuti
nale SEL , solo quando è in
indirizzi dell'interfaccia. La
 TOR della CPU. Si osservi
e dal bus di sistema (es. $D0$)

da inviare al PPI 8255, è
ite tri-state; questi possono

ccia, viene inizializzato al
esso le linee indirizzo AX_0 e
base a 340_H , al variare delle
istro di controllo. I segnali
nsentono rispettivamente di

25 pin

bidirezionale

t
erfaccia

otata di connettore a pettine
i pin. Successivamente si è
ori con tecniche wire-wrap.
2) e i bit delle tre porte sono

Il collaudo è stato eseguito su un sistema 8086 disponibile in laboratorio e dedicato ad esercitazioni che coinvolgano direttamente l'hardware del computer. I dip-switch sono stati posizionati in modo da fissare l'indirizzo base a 300_H .

Inserita la scheda in uno degli slot liberi, si è collegato il connettore di uscita ad una bread-board esterna, che ha consentito di prelevare i dati e i segnali di controllo e di visualizzarli con l'oscilloscopio. Utilizzando un alimentatore esterno, si è potuto verificare il funzionamento in input portando ciascun bit a $+5 V$ e a massa.

In fase di collaudo si è preferito lavorare con software scritto in linguaggio Basic, che offre indubbi vantaggi di praticità.

Modo 0 - Output. In fig. 6 è riportata una routine che consente di verificare il funzionamento in modo 0 delle tre porte configurate come output. Prelevando i segnali presenti sui terminali della porta A si visualizzano onde quadre di frequenza decrescente da PA_0 a PA_7 ; le stesse forme d'onda si rilevano su PB o PC .

Modo 0 - Input. La verifica del funzionamento in input è stata eseguita configurando le tre porte, in modo 0, tutte in input. Collegate a massa tutte le linee PA , PB e PC , si sono eseguite letture cicliche dei dati presenti sulle porte scrivendo il dato su video. In fig. 7 è riportata la routine utilizzata.

Dopo aver constatato che effettivamente sul video comparivano tre caratteri 0 per ciascuna riga, si è ripetuta la prova collegando le linee PA a $+5 V$. Sul video si susseguivano correttamente righe composte dal numero FF seguito da due caratteri 0.

```

10 'Configurazione interfaccia
20 IPA = &H300           'Indirizzo base = Porta A
30 IPB = IPA + 1        'Porta B
40 IPC = IPA + 2        'Porta C
50 ICR = IPA + 3        'Registro di controllo
60 OUT ICR,&H80         'Modo 0; PA, PB, PC in output
100 'Emissione ciclica sulle tre porte di un dato variabile da 0 a 255
110 FOR I = 0 TO 255
120 OUT IPA,I: OUT IPB,I: OUT IPC,I
130 NEXT
140 GOTO 110

```

FIG. 6

Collaudo out

```

10 'Configurazione interfaccia
20 IPA = &H300           'Indirizzo base = Porta A
30 IPB = IPA + 1        'Porta B
40 IPC = IPA + 2        'Porta C
50 ICR = IPA + 3        'Registro di controllo
60 OUT ICR,&H9A         'Modo 0; PA, PB, PC in input
100 'Lettura ciclica delle tre porte e scrittura dei dati su video
110 DA = INP (IPA): DB = INP (IPB): DC = INP (IPC) 'Lettura delle porte
120 DAS = HEX$(DA): DBS = HEX$(DB): DCS = HEX$(DC) 'Conversione in esadecimale
130 PRINT DAS;DBS;DCS 'Scriva su video
140 GOTO 110           'Ricomincia il ciclo

```

FIG. 7

Collaudo inp

Modo 1 - Input. È stata poi effettuata una prova con modo di funzionamento 1 sulla porta A configurata in input e modo 0 sulla porta B configurata in output; in output sono stati definiti anche i bit disponibili di PC ($PC_0 \div PC_2, PC_6, PC_7$).

Poiché in questo caso il trasferimento in input è attivato da un fronte di discesa applicato esternamente sul pin PC_4 (STBA), è stato predisposto sulla bread-board un generatore di impulso negativo, comandato da un pulsante, la cui uscita è stata connessa appunto a PC_4 . Come si vede nel diagramma di fig. 8, dopo un tempo t_{SI} (max 600 ns) durante il quale il dato viene caricato nel registro interno della porta A, il segnale ACKA (pin PC_5) diventa alto. A questo punto la CPU può leggere il registro della porta A mentre i dati in ingresso diventano irrilevanti. Si noti che dopo un tempo t_{RI} (max 300 ns), ACKA ritorna basso.

In fig. 9 è riportata la routine utilizzata.

FIG. 8
Diagramma temporale input modo 1.

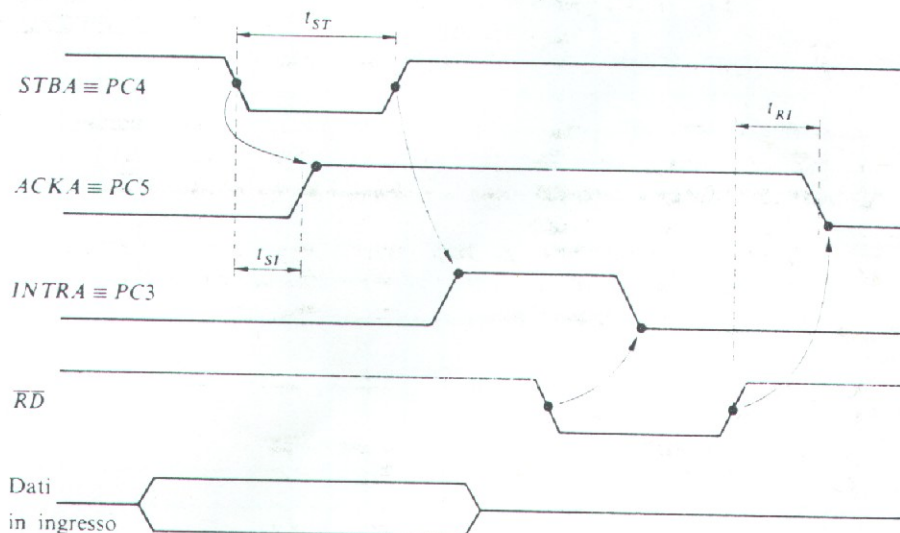


FIG. 9
Collaudo input modo 1.

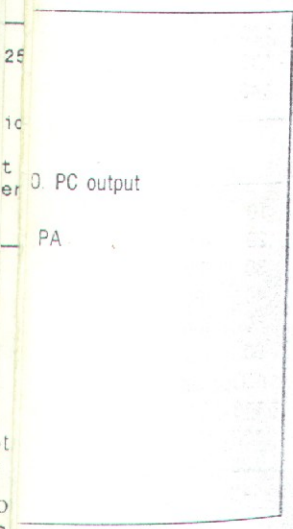
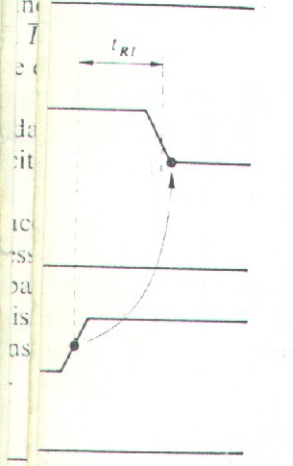
```

10 'Configurazione interfaccia
20 IPA = &H300 'Indirizzo base = Porta A
30 IPB = IPA + 1 'Porta B
40 IPC = IPA + 2 'Porta C
50 ICR = IPA + 3 'Registro di controllo
60 OUT ICR,&HBD 'PA input Modo 1. PB output Modo 0. PC output
100 'Input modo 1
110 'Lettura porta PC e test di PC5: quando diventerà alto si leggerà il dato su PA
120 STATOC = INP(IPC) 'Lettura porta C
130 ACK = STATOC AND &H20 'Maschera bit PC5
140 IF ACK = 0 THEN 120 'Aspetta PC5 alto
150 DA = INP(IPA) 'Lettura dato PA
160 DAS = HEX(DA) 'Conversione in esadecimale
170 PRINT DAS 'Scrive su video il dato letto
180 'Output modo 0
190 OUT IPB,DA 'Emette su PB il dato letto su PA
200 GOTO 110 'Ricomincia il ciclo
    
```

00100000

onamento I sulla porta
iti in output sono stati

un fronte di discesa
sulla bread-board un
uscita è stata connessa
tempo t_{SI} (max 600 ns)
porta A, il segnale ACK_A
o della porta A mentre
 t_{RI} (max 300 ns), ACK_A



Prove finali. Verificata la funzionalità generale, conviene effettuare qualche semplice prova, ad esempio output in modo 0, utilizzando routine in linguaggio Assembly. La routine di fig. 6, ad esempio, può essere facilmente tradotta in Assembly ed eseguita in ambiente DEBUG. Si ha modo così di valutare le prestazioni del sistema anche in termini di velocità di trasferimento dei dati.

Le prove su questa interfaccia possono articolarsi nei modi più svariati e può essere anche interessante approfondire tutte le particolarità di funzionamento del PPI 8255. A questo proposito occorre segnalare che la nomenclatura usata per identificare i segnali non è sempre uniforme nella letteratura esistente (vedi ad esempio i diagrammi dei fogli tecnici riportati in Appendice).

In particolare il segnale qui denominato STB viene talvolta chiamato OBF (output buffer full) per le operazioni di output. Analogamente, ACK viene chiamato IBF (input buffer full) nelle operazioni di input. Il diagramma riassuntivo riportato in fig. 10, indicando il verso dei segnali e la numerazione dei bit della porta C, può chiarire alcune ambiguità.

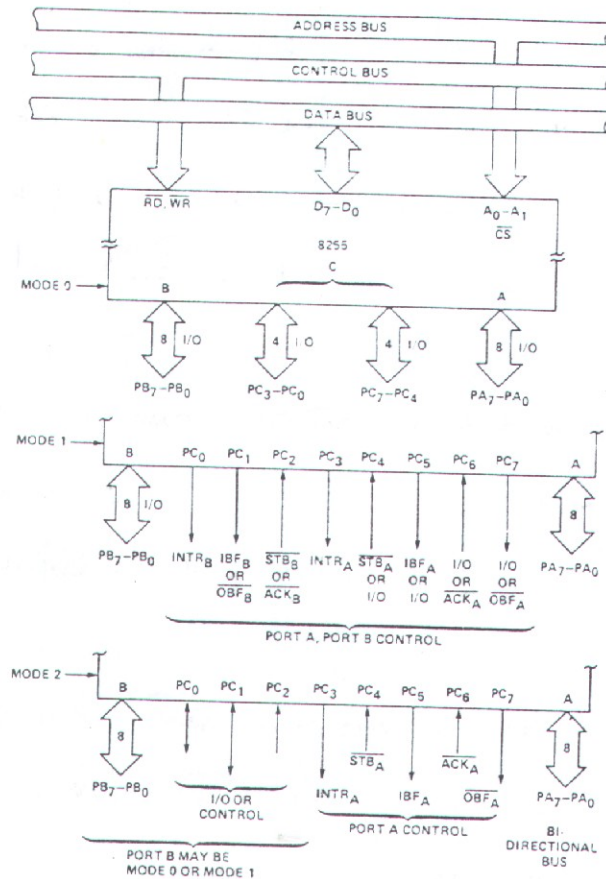


FIG. 10
Diagramma
riassuntivo.