

PIO Z80	1
Piedinatura	3
Struttura interna	4
MODI DI FUNZIONAMENTO	6
Modo 0: byte in uscita	7
Modo 1: Byte in ingresso.	9
Modo 2: byte bidirezionale	11
Modo 3 bit a bit	12
Interfacciamento dello Z80	14
Programmazione del PIO	16
Modo 0, 1, 2	16
Modo 3	17

Pio Z80

IL PIO Z80 è un'interfaccia programmabile parallela d'ingresso uscita. A cosa serve un'interfaccia parallela? L'interfacciamento fra microprocessore e periferiche comporta la risoluzione di diversi problemi.

- Interfacciare una periferica al microprocessore comporta il fatto che la periferica sia collegata al bus. Ciò significa che la periferica deve essere dotata

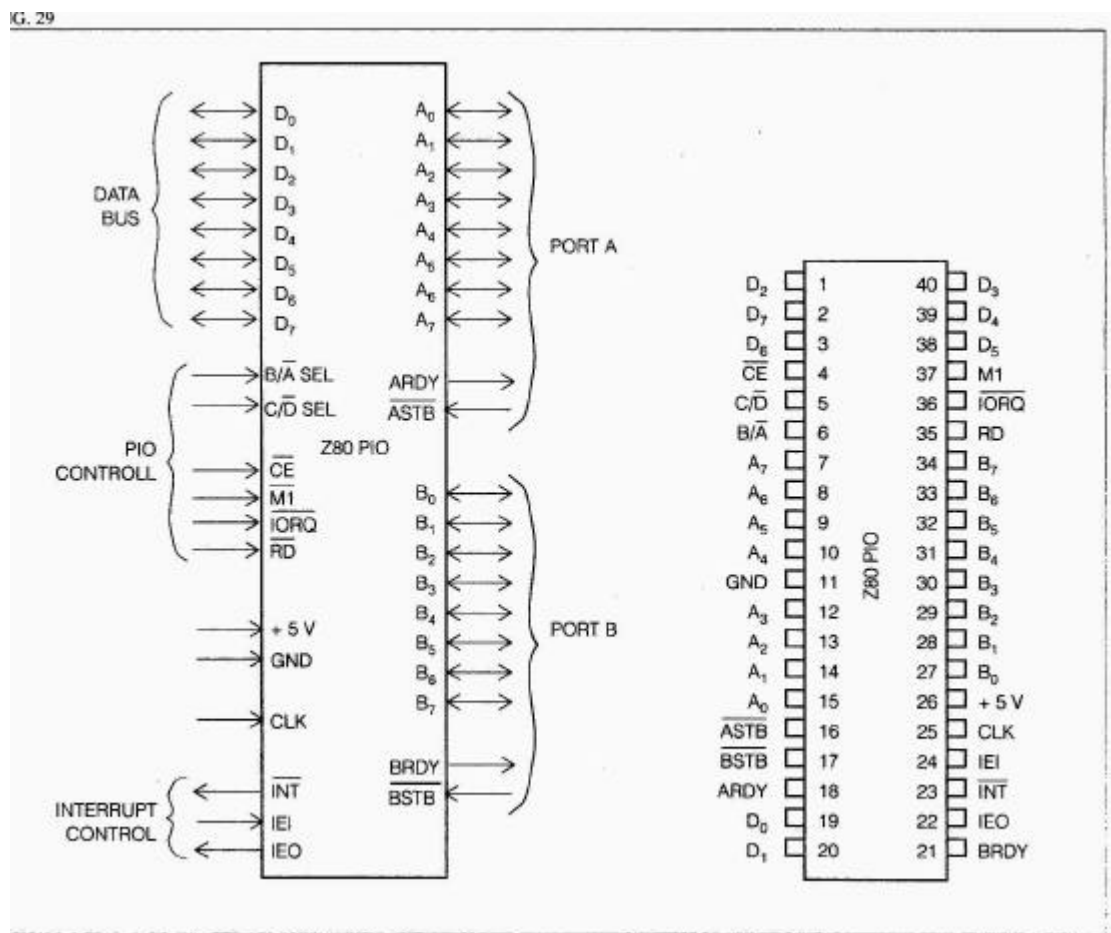
di uscite di three-state per evitare conflitti sul bus. Se la periferica non è dotata di uscite di questo tipo si deve interporre un dispositivo apposito fra periferica e bus.

- La periferica è in genere più lenta del microprocessore nelle operazioni di IO, per cui occorre che essa sia dotata di un buffer o zona di memorizzazione temporanea, in cui il microprocessore possa parcheggiare temporaneamente i dati.
- Anche in questo caso il microprocessore è inutilmente rallentato dalla periferica. Infatti, se la periferica non ha ancora letto il dato nel buffer, il microprocessore non può inviarle altri dati e deve rimanere in attesa
- In questo caso si rende necessario un protocollo di handshake fra periferica e microprocessore costituito da alcuni segnali che consentano al microprocessore di avvertire la periferica che è disponibile un dato per essa, e alla periferica di avvertire il microprocessore quando essa ha terminato di acquisire il dato.
- Per evitare che il microprocessore debba subire i tempi morti dovuti alla lentezza della periferica sarebbe utile usare le tecniche di interrupt. Ad esempio, il microprocessore potrebbe inviare un dato alla periferica e poi svolgere altre operazioni. Quando la periferica termina di acquisire il dato può generare un'interruzione per costringere il microprocessore a dedicarle nuovamente la sua attenzione ed inviarle un nuovo dato.

Tutte queste problematiche vengono risolte automaticamente interponendo un PIO Z80 fra periferica e microprocessore.

Piedinatura

La piedinatura del PIO è illustrata nella figura seguente



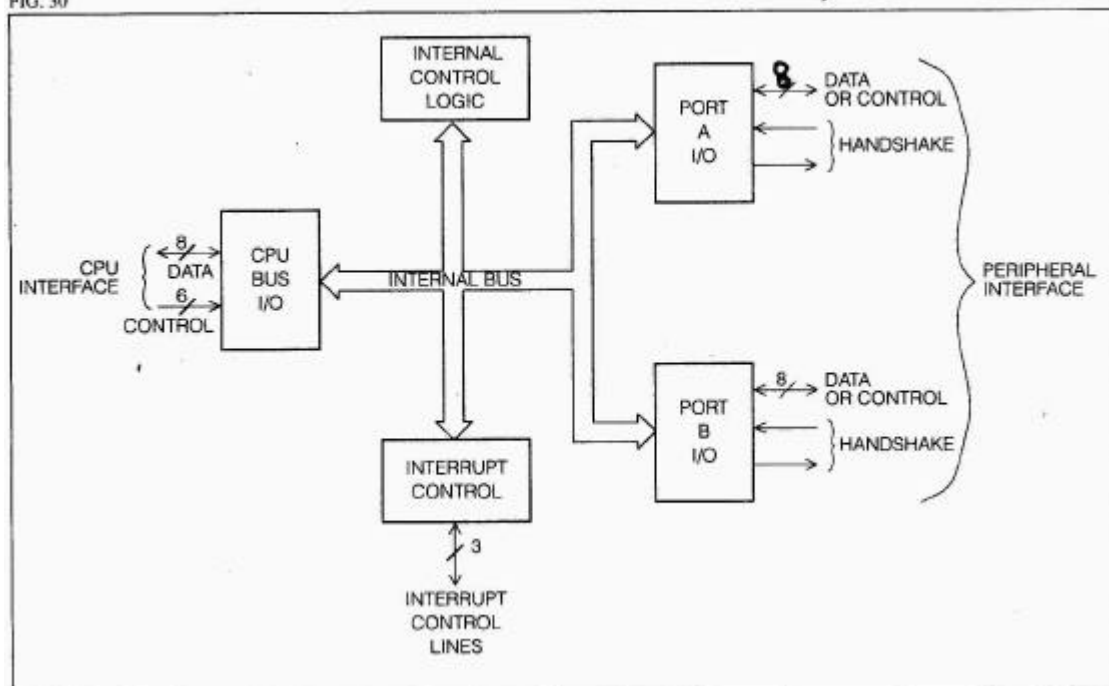
IL PIO presenta 8 pin da D₀ a D₇ per poter essere collegato al bus dati. Ha a disposizione poi due porte denominate A e B per poter essere collegato a due periferiche distinte. Ogni porta presenta otto piedini sui quali potranno transitare i dati e due piedini RDY (Ready) e STB (Strobe) per poter effettuare un dialogo di handshake con la periferica. Poiché il Pio è in grado di generare interruzioni presenta un piedino INT. Esso può essere messo anche in daisy chain con l'altri dispositivi per cui presenta l'ingresso IEI e l'uscita IEO. Il PIO è progettato per poter utilizzare la modalità di gestione 2 delle interruzioni dello Z80. Ricordiamo brevemente che in tale modalità, quando la periferica genera un'interruzione, il

microprocessore innesca un ciclo di riconoscimento delle interruzioni caratterizzato dal fatto che le linee M1 e IORQ vanno contemporaneamente a zero. La periferica, vedendo le due linee contemporaneamente a zero, pone sul bus dati la parte bassa del vettore delle interruzioni. Il PIO per poter sapere che il microprocessore ha intrapreso il ciclo di riconoscimento delle interruzioni, ha bisogno di essere collegata alle due linee M1 e IORQ. E' presente inoltre il piedino CE di abilitazione in modo che il PIO si possa collegare al bus quando lo decide il microprocessore. Per poter selezionare la porta A o la porta B, il microprocessore può utilizzare il piedino B/A. Quando tale piedino è a zero, viene selezionata la porta A, quando è ad 1 viene selezionata la porta B. infine il microprocessore deve poter indicare al Pio se gli sta inviando un dato che va poi passato alla periferica collegata al PIO oppure gli sta inviando una parola di programmazione. Per fare questo utilizza il piedino C/D. se questo pedino è a zero vuol dire che si tratta di un dato, se è ad uno vuol dire che è una parola di programmazione. Si noti che al PIO arriva solo il segnale RD.

Struttura interna

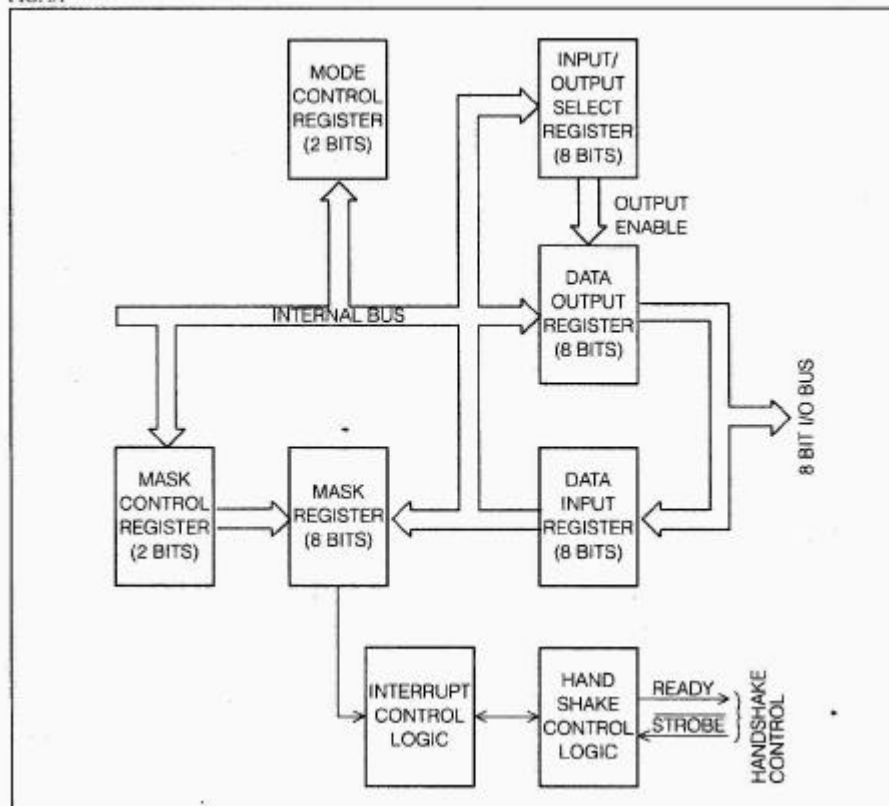
La struttura interna del PIO è rappresentata nella figura seguente

FIG. 30



La struttura di ogni singola porta è invece la seguente

FIG. 31



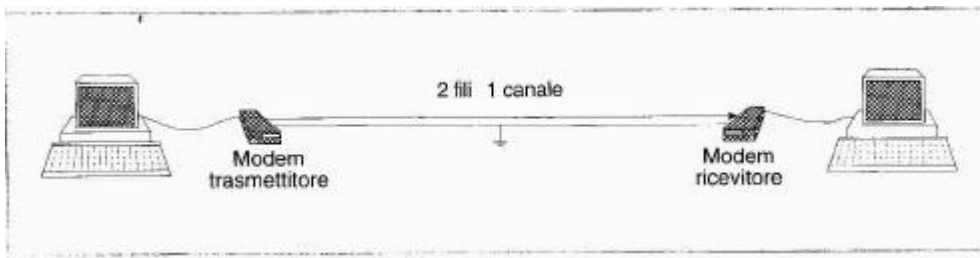
- Il Mode Control Register è il registro il cui contenuto indica il quale modalità opera la porta
- L'interrupt control logic controlla la generazione delle interruzioni
- L'handshake control logic controlla i segnali di handshake ed il dialogo fra porta e periferica
- Il data output register è un buffer per i dati che devono andare verso la periferica
- Il data input register è un buffer per i dati che provengono dalla periferica
- L'input/output select register, il mask control register e il mask register sono registri che vengono coinvolti nel modo tre di funzionamento della porta.

MODI DI FUNZIONAMENTO

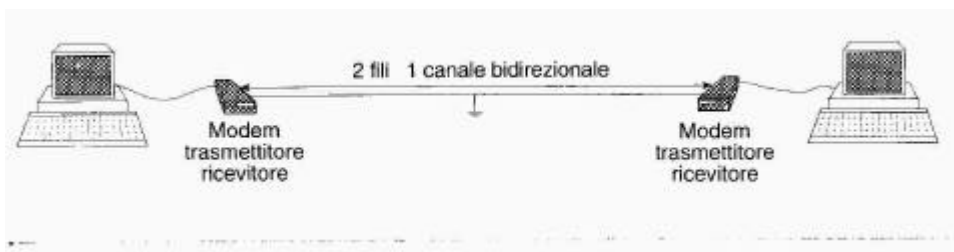
Si possono definire tre modalità diverse di trasmissione:

- Simplex
- Half-duplex
- Full-duplex

Nel collegamento simplex i dati possono viaggiare soltanto in un senso. Il ruolo di trasmettitore e di ricevitore sono assegnati una volta per tutte. Il dispositivo ricevitore non può inviare a sua volta dati al trasmettitore.



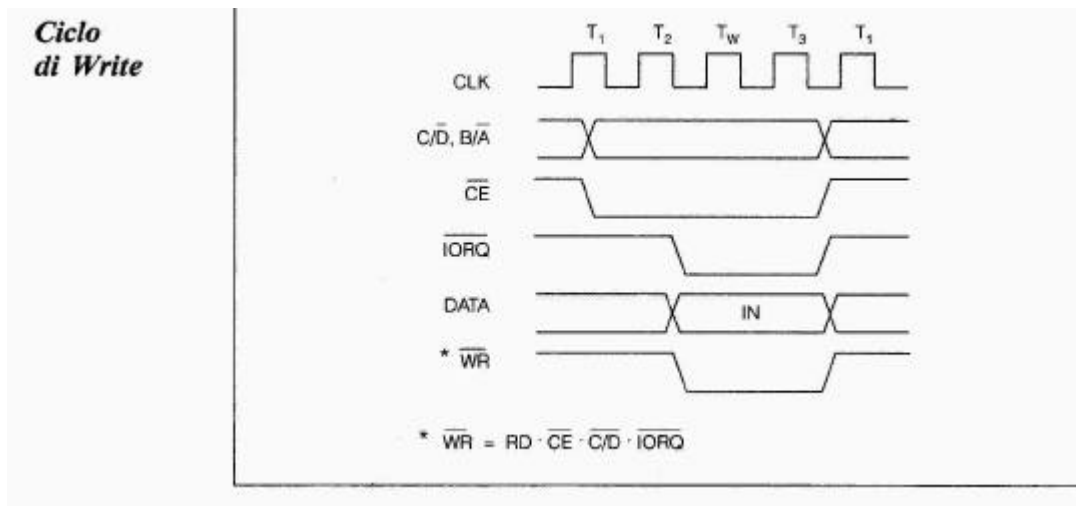
Nel collegamento half-duplex esiste un'unica linea di connessione come nel caso del collegamento simplex. Stavolta, però, i ruoli di trasmettitore e ricevitore non sono fissati in modo rigido ed i dati possono viaggiare in entrambe le direzioni. Essendo, però, la linea di comunicazione unica, di volta in volta uno dei due terminali è in ricezione e l'altro in trasmissione. Perché si scambino i ruoli occorre un certo tempo di turnaround.



Nel collegamento full-duplex i due sistemi sono collegati mediante una doppia linea trasmissiva che consente di effettuare contemporaneamente le operazioni di trasmissione e ricezione. Raddoppia la quantità di dati che si possono scambiare nell'unità di tempo e si eliminano i tempi morti dovuti al turnaround.

Modo 0: byte in uscita

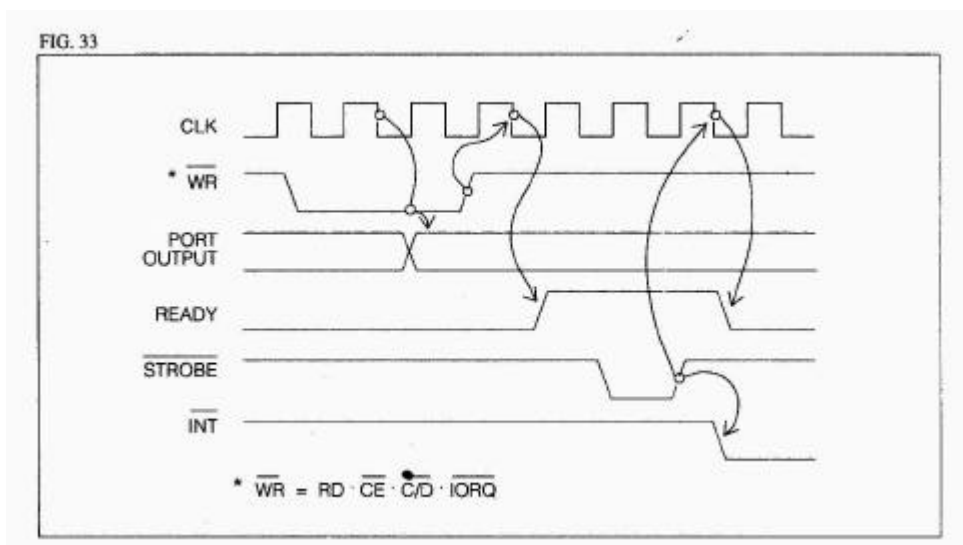
È una modalità simplex che prevede che i dati vadano sempre dal microprocessore alla periferica. Il microprocessore scrive il dato all'indirizzo della porta del PIO (vedi figura)



Come si vede dalla figura, all'interno della porta si genera un segnale *WR che va a zero quando il microprocessore scrive il dato sulla porta per cui si ha contemporaneamente che

- $\overline{RD}$ è ad uno quindi RD è a zero (si sta scrivendo)
- $\overline{CE}$ è a zero (il microprocessore ha abilitato il PIO)
- $\overline{C/D}$ è a zero (quindi si sta inviando un dato alla periferica)
- $\overline{IORQ}$ è a zero (si sta effettuando un'operazione di ingresso uscita)

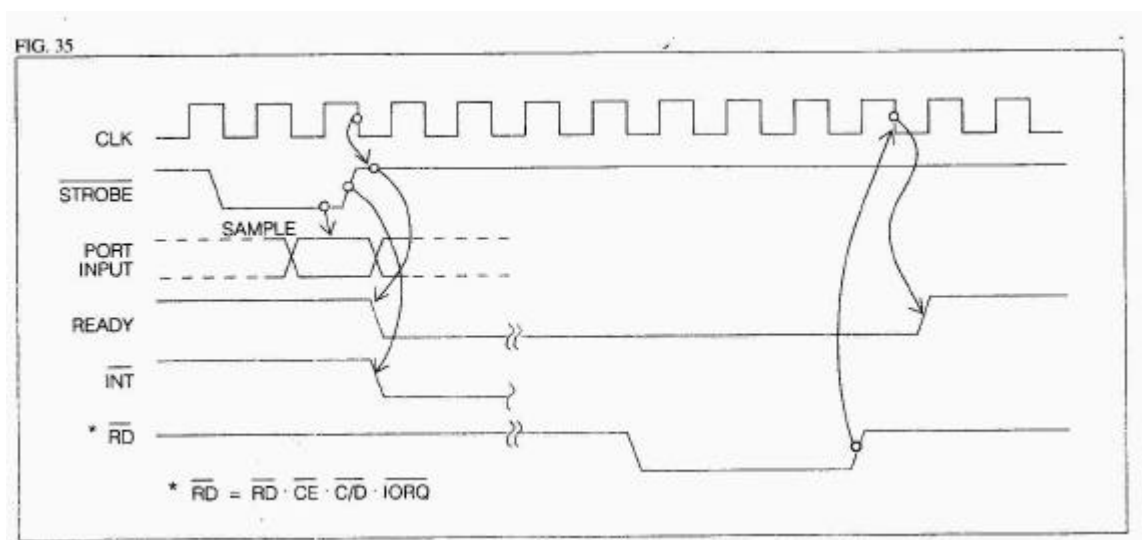
Dal lato della porta si ha la seguente temporizzazione



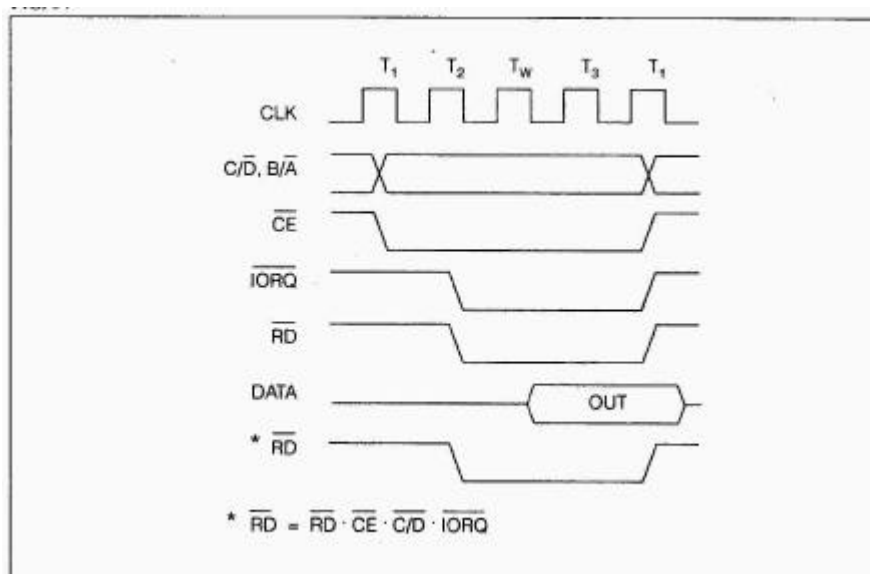
Mentre $\overline{*WR}$ è attivo, il dato viene trasferito sui piedini che collegano la porta alla periferica (Port Output), quando il segnale $\overline{*WR}$ si rialza vuol dire che l'operazione di scrittura da parte del microprocessore sulla porta è terminata. Allora la porta alza il segnale READY per avvertire la periferica che un dato è disponibile per essa. La periferica risponde mandando basso il segnale STROBE. Finché questo segnale rimane basso vuol dire che la periferica è ancora impegnata nell'acquisizione del dato stesso impedendo alla porta del PIO di inviarle altri dati. Quando STROBE si rialza indica alla porta del P_{io} che il trasferimento del dato è stato completato e si può inviare alla periferica un altro dato. Se la porta è abilitata ad utilizzare le interruzioni genera a questo punto un'interruzione per il microprocessore per costringerlo ad inviare un nuovo dato.

Modo 1: Byte in ingresso.

Anche in questo caso si tratta di un collegamento simplex. Stavolta, però, i dati provengono dalla periferica e vanno verso il microprocessore.



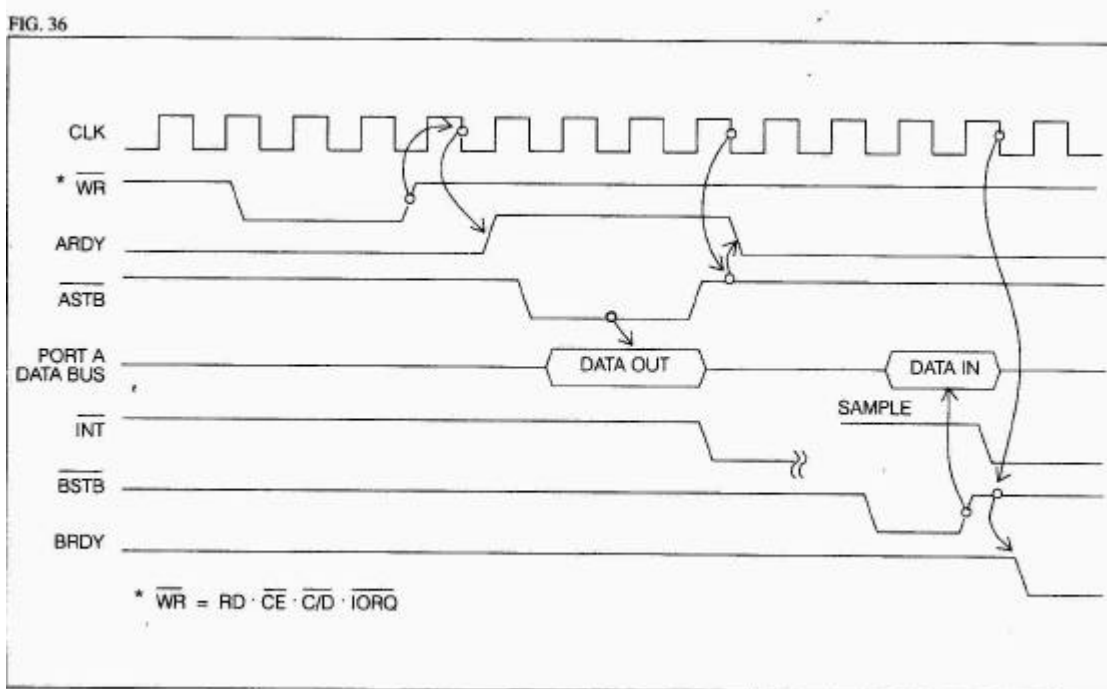
la periferica abbassa il segnale di strobe per avvertire la porta che è disponibile un dato da mandare al microprocessore. La porta provvede poi a scaricare il dato sul bus che la collega alla porta. Quando si rialza il segnale di strobe, la porta del PIO abbassa il segnale READY. Finché questo segnale rimane basso la periferica sa che il dato non è stato ancora acquisito dal microprocessore e quindi essa non può inviare un altro dato. Quando abbassa il segnale READY, la porta, se è abilitata, può generare un'interruzione per indicare al microprocessore che deve leggere il dato depositato nel buffer della porta. Quando il microprocessore inizia il ciclo di lettura



nella porta si genera un segnale $*RD$ che resta a zero finché è in corso l'acquisizione del dato da parte del microprocessore. Se questo segnale si rialza, vuol dire che la lettura del dato da parte del microprocessore è terminata per cui READY si rialza indicando alla periferica che può inviare un nuovo dato.

Modo 2: byte bidirezionale

In questa modalità il PIO consente un dialogo di tipo half-duplex permettendo ai dati di viaggiare in entrambi i sensi alternativamente. In questo caso occorre, però, duplicare le linee di handshake. Occorrono due segnali per garantire il dialogo nel caso che il dato vada verso la periferica e due linee per il dialogo nel caso che il dato vada dalla periferica alla porta. La soluzione consiste nel collegare la periferica alla porta A e di utilizzare per l'handshake le linee di controllo sia della porta A che della porta B. In tal caso la porta B non può essere utilizzata per collegare una periferica, poiché non sarebbe in grado di gestire protocolli di handshake. La porta B può essere utilizzata soltanto nel modo 3 che vedremo più avanti. Le linee della porta A vengono utilizzate per controllare l'invio dei dati dalla porta alla periferica e le linee della porta B vengono utilizzate per controllare l'invio dei dati dalla periferica alla porta A.



Come si può notare dal diagramma temporale si tratta, insomma, della fusione dei modi 0 ed 1. C'è però una piccola differenza. Nel modo zero la porta scarica il dato sul bus che la collega alla periferica prima di attivare il segnale READY. Nel modo 2, invece, la porta attiva prima READY, aspetta di avere una conferma dalla periferica mediante il segnale STROBE e poi invia il dato in uscita. Il motivo di questa differenza è il seguente. Immaginiamo che la periferica abbia inviato un dato verso la porta. Ora vi è un turnaround, cioè è la porta a voler inviare un dato alla periferica. Quest'ultima potrebbe aver ritardato nel rilasciare il bus, per cui si potrebbe avere che contemporaneamente sia la porta che la periferica tentano di inviare un dato sul bus creando un conflitto. E' per questo che, nel modo 2, la porta aspetta l'OK da parte della periferica prima di inviare il dato sul bus.

Modo 3 bit a bit

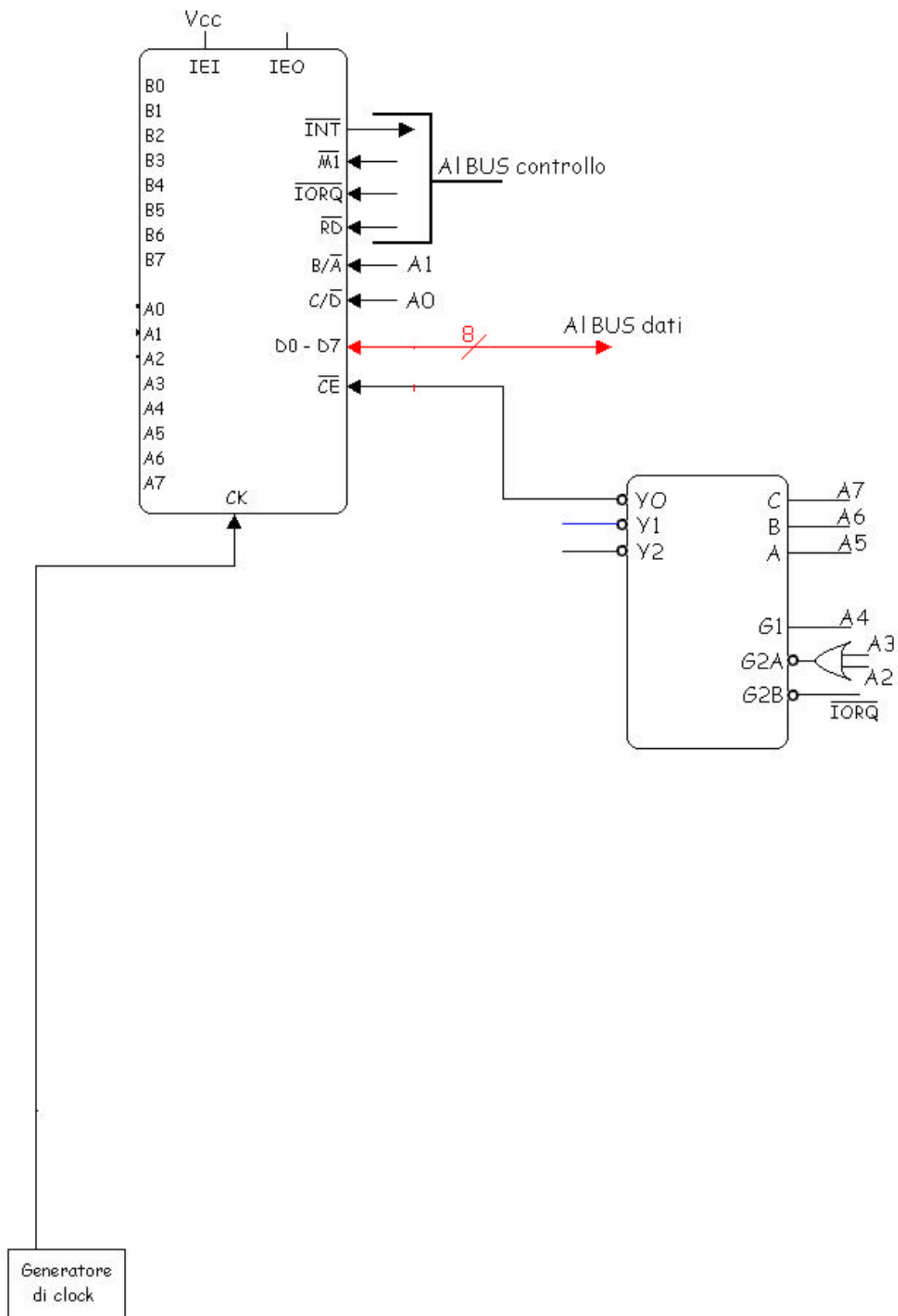
Questa modalità viene utilizzata quando si vuole interfacciare il microprocessore con sensori o attuatori da comandare. In questo caso non si tratta di far transitare dati con parallelismo a 8 bit, ma ogni bit della porta ha una sua identità indipendente. In questa modalità ogni singola linea della porta può essere programmata come ingresso o come uscita indipendentemente dalle altre. In questa modalità, i segnali di handshake non vengono utilizzati. Quando il microprocessore tenta di leggere la porta del PIO, le linee di ingresso vengono latchate, cioè vengono congelate al valore che avevano quando si è abbassato il segnale di RD $\bar{}$. In tal modo eventuali variazioni che subiscono le linee di ingresso

durante la lettura non vengono prese in considerazione dal microprocessore. Per le linee impostate come uscita i bit corrispondenti che vengono letti contengono l'ultimo valore che era stato inserito nell'ultima operazione di scrittura. Molto utili sono le caratteristiche di tale modo per quanto riguarda la generazione di interruzioni.

- Innanzitutto tra tutti i sensori collegati ai pin di ingresso della porta, può accadere che solo alcuni abbiano un'importanza tale da volere che generino un'interruzione. In tal caso il PIO consente di mascherare i bit di ingresso che non vogliamo che possano causare un'interruzione.
- Inoltre può accadere che noi vogliamo che sia generata un'interruzione quando almeno uno dei sensori d'ingresso sia attivo (si parla di condizione di OR logico) oppure che sia generata l'interruzione quando tutti i sensori non mascherati siano attivi (condizione di AND logico). Il Pio consente, tramite opportuna programmazione di impostare, per la generazione delle interruzioni, l'AND o l'OR logico
- Si possono collegare alla porta sensori attivi alti o attivi bassi. Il Pio consente di stabilire se i sensori collegati vanno considerati attivi quando danno un segnale alto o danno un segnale basso. In tal caso non si possono, però, utilizzare sensori promiscui. Si possono utilizzare cioè, sensori di un solo tipo in una porta.

Interfacciamento dello Z80

Nella figura seguente vi è un esempio di interfacciamento fra microprocessore Z80 e PIO



come si può notare, i bit A_1 e A_0 sono stati utilizzati per realizzare i segnali C/D e B/A, rispettivamente. In tal modo il PIO avrà quattro indirizzi di IO, secondo la seguente tabella

A_1	A_0	
0	0	Dato alla porta A (C/D=0, B/A=0)
0	1	Dato alla porta B (C/D=0, B/A=1)
1	0	Comando alla porta A (C/D=1, B/A=0)
1	1	Comando alla porta B (C/D=1, B/A=1)

Programmazione del PIO

Modo 0, 1, 2

Per programmare una porta del PIO in questi modi occorrono

- Una parola di controllo di modo (mode control word) che ha la seguente struttura

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
		X	X	1	1	1	1

I bit da D_0 a D_3 hanno sempre lo stesso contenuto che identifica la mode control word, i bit D_4 e D_5 possono avere un valore qualunque mentre i bit D_6 D_7 identificano il modo secondo la seguente tabella

D_7	D_6	
0	0	MODO 0
0	1	MODO 1
1	0	MODO 2
1	1	MODO 3

- Segue la parte bassa del vettore delle interruzioni (interrupt vector word), quel byte che la porta restituirà al microprocessore quando avrà generato un'interruzione
- Interrupt disable word o parola di disabilitazione delle interruzioni

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
	X	X	X	0	0	1	1

I bit da D_0 a D_3 hanno sempre lo stesso contenuto che identifica la parola, i bit D_4 e D_5 e D_6 possono avere un valore qualunque mentre il bit D_7 specifica se le interruzioni vanno abilitate (bit a 1) o disabilite (bit a zero).

Modo 3

La programmazione di una porta in modo 3 è più complessa

- Mode control word

- Interrupt vector word
- I/O register control word o parola di controllo registro di IO. Questa parola viene registrata nel Input/Output select register che abbiamo incontrato nella struttura della porta ed indica quali bit sono di ingresso e quali sono di uscita. Se un bit è posto a zero, la linea corrispondente del bus della porta è settata come uscita, se il bit è ad 1 la linea corrispondente è settata come ingresso.

Ad esempio, se inviamo alla porta A il seguente byte

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

I bit A₅ e A₄ sono uscite , tutti gli altri sono ingressi.

- Interrupt control word o parola di controllo delle interruzioni. Questa parola setta tutte le caratteristiche che abbiamo visto si possono selezionare per la generazione delle interruzioni quando la porta lavora in modo 3.

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

I bit da D₀ a D₃ hanno sempre lo stesso contenuto che identifica la interrupt control word, il bit D₄ specifica se gli ingressi sono mascherati o meno, se esso è a zero gli ingressi sono tutti coinvolti nella generazione delle interruzioni, se il bit è ad 1 alcuni bit d'ingresso non contribuiranno alla generazione delle interruzioni per cui dovrà seguire un byte di maschera che indicherà quali bit di ingresso dovranno essere mascherati cioè ignorati per la generazione delle

interruzioni. Il bit D_5 indica se i sensori di ingresso sono attivi bassi (bit a zero) o alti (bit ad 1). Il bit D_6 indica se la interruzione verrà generata su condizione di OR logico (basta che un sensore sia attivo) oppure di AND logico (devono essere attivi tutti i sensori contemporaneamente). Il bit D_7 indica se la generazione delle interruzioni è disabilitata (bit a 0) oppure abilitata (bit ad 1).

- Mask control word o parola di controllo maschera. Questa parola di controllo indica se un bit di ingresso va preso in considerazione per la generazione delle interruzioni (bit corrispondente della parola a zero) oppure va ignorato (bit ad 1)
- Interrupt disable word. Nel modo tre questa parola può essere usata per abilitare o disabilitare le interruzioni senza modificare le altre impostazioni definite con la interrupt control word