

PPI 8255

L'8255 è un circuito integrato di I/O molto tipico, usato in molte applicazioni. Mette a disposizione un massimo di 24 linee di I/O, programmabili in ingresso, in uscita o anche in Modo bidirezionale. Viene chiamato PPI (**P**rogrammable **P**eripheral **I**nterface) in quanto è un dispositivo di I/O "programmabile".

Come tale esso si comporta in Modo diverso a seconda di come viene "programmato". Per programmarlo basta scrivere un Byte, opportunamente codificato, in uno dei suoi registri.

Il circuito dispone di quattro registri interni, visibili a quattro indirizzi consecutivi. Attraverso uno di questi registri è possibile programmare il circuito per farlo funzionare in modi diversi. Utilizzando gli altri tre registri si possono leggere o scrivere i valori degli I/O digitali collegati con i piedini del circuito.

L'indirizzo più basso del circuito (indirizzo BASE) viene detto PortA (PA) e fornisce 8 I/O digitali, che possono funzionare in tre modi diversi. Il secondo indirizzo (BASE + 1) è il PortB (PB), che può funzionare in due modalità. Al terzo (BASE + 2) indirizzo si trova il PortC (PC) che è un Port particolare, i bit del quale hanno funzioni diverse a seconda della modalità di funzionamento degli altri port. Il PortC è diviso in due metà programmabili separatamente.

Il quarto registro del dispositivo (BASE + 3) viene detto "Control Word" (anche se è un Byte!) ^{(*)1}. Tramite la CW è possibile programmare il dispositivo, stabilendo le modalità di funzionamento degli altri port.

Come accennato le modalità di I/O programmabili sul circuito sono al massimo tre e vengono dette: Modo 0, Modo 1 e Modo 2.

In Modo 0 l'8255 fornisce un massimo di 24 I/O digitali semplici, "senza controllo"; in Modo 1 si hanno un massimo di 16 I/O controllati, con linee di "strobe, acknowledge e interrupt"; in Modo 2 si dispone di un singolo Port di I/O bidirezionale da 8 bit.

I Port sono programmabili indipendentemente uno dall'altro, ciò significa per esempio che se il PortA è programmato in Modo 0 il PortB può essere contemporaneamente programmato in Modo 1.

Per programmare il dispositivo si deve scrivere il Control Register, che si trova al quarto ed ultimo indirizzo del chip. Ogni bit del Control Register ha un particolare significato per il circuito, la seguente tabella mostra il significato di ogni bit:

7	6	5	4	3	2	1	0
set/reset singolo bit del PortC	Modo PortA Bit alto	Modo PortA Bit basso	Direzione PortA	Direzione PortC High nibble	Modo PortB	Direzione PortB	Direzione PortC Low nibble
Mettere 1	00 = Modo 0 (I/O "libero") 01 = Modo 1 (handshake) 1X = Modo 2 (bidirezionale)		1 = Input 0 = Output	1 = Input 0 = Output	0 = Modo 0 1 = Modo 1	1 = Input 0 = Output	1 = Input 0 = Output

Figura 1: modalità di programmazione dell'8255

Come si vede dalla Figura 1: modalità di programmazione dell'8255 il PortA può essere programmato in tre modi tramite i bit 6 e 5 (1X significa che qualunque sia il valore di X il Port è programmato in Modo 2). Il PortB invece ammette solo il Modo 0 ed il Modo 1, per cui basta il solo bit 2 per programmarlo. In Modo 0 e 1 la direzione di ogni Port può essere programmata in ingresso o in uscita.

Per i PortA e B tutti gli 8 bit possono essere programmati o in ingresso oppure in uscita (bit 1 e 4), mentre il PortC è diviso in due metà e può essere programmato in ingresso o uscita nibble per nibble (bit 0 e 3).

Perché il Control Register funzioni come scritto in tabella il bit più significativo deve essere a 1. Infatti quando il bit "set/reset" è a zero CW serve per cambiare individualmente lo stato di uno qualsiasi dei bit del PortC, lasciando gli altri così come sono. In questo caso il significato di ciò che si scrive è del tutto diverso da quello di tabella ed indica quale bit del PortC modificare e quale valore attribuirgli. Non si indicheranno i dettagli, per i quali si rimanda alla documentazione del produttore, che può essere trovata in <http://developer.intel.com/> come application note AP-15 e nel CDROM di questo libro. Dunque, per quanto si è detto, quando si programma un 8255 il bit più significativo del CW sarà sempre tenuto a 1.

0.1. Modo 0

Nel Modo 0 i dati sono semplicemente scritti o letti dai port. I Port sono buffer elettronici che alzano od abbassano le linee elettriche ad essi collegate a seconda dei numeri binari scritti al loro interno. Nessun altro segnale è coinvolto.

Se il PortA è programmato in Modo 0 anche i quattro bit alti del PortC possono essere programmati come normali I/O digitali. La direzione dei trasferimenti viene indicata con i bit 0,1, 3 e 4.

Una volta che si è scritto nella control word il dispositivo funziona come è stato programmato, per cui basta scriverlo una volta sola. Sui Port programmati in ingresso sarà possibile fare le IN; su quelli di uscita le OUT. Sui Port program-

¹ Questa definizione ci fa capire che questo circuito è stato sviluppato ai tempi delle CPU ad otto bit, che avevano "word" da un Byte

mati in ingresso le OUT non funzioneranno. Per cambiare la direzione dei trasferimenti con Port programmati in Modo 0 e 1 è necessario riprogrammarli, cambiando il contenuto della Control Word².

Il PortC può essere programmato come normale I/O solo quando la porta che lo controlla è programmata in Modo 0. Negli altri modi esso contiene linee accessorie relative al PortA ed al PortB.

Vediamo un esempio in cui tutti i Port di un 8255 sono programmati in Modo 0. Supponiamo di volere: PortA in ingresso, PortC High in uscita, PortB in ingresso, PortC Low in 1 uscita (si noti come le parti alta e bassa del PortC siano una in uscita, l'altra in ingresso; essi sono perciò indipendenti!). Per ottenere questo funzionamento bisogna che il CW sia programmato così:

sempre 1	Modo 0 PortA		PortA In	PortCH Out	Modo 0 PB	PortA Out	PortCL In
1	0	0	1	0	0	1	1

Il numero da scrivere in CW è quindi 10010011b.

Ecco il codice corrispondente:

```
Base EQU 80 ; questo è un indirizzo fittizio, minore di 256
; (un indirizzo usato effettivamente in schede 8255 in commercio è XXXX !!!!metterlo!!!!)

PortA EQU Base
PortB EQU Base + 1
PortC EQU Base + 2
Control EQU Base + 3
..

MOV DX, Control
MOV AL, 10010011b
; programmazione dell'8255: tutti i Port in Modo 0, PortA in Input,
; PortC High in Output, PortB in Input, PortC Low in 1 Input:
OUT DX, AL
..
; nelle parte successiva del codice potrò fare:
IN AL, PortA ; !! funziona solo se PortA è un numero < 256 (8 bit)!!
OUT PortB, AL
OUT PortC, AL ; si usano solo i bit della parte alta, gli altri sono in ingresso
IN AL, PortC ; sono significativi solo i bit della parte bassa,
; gli altri vanno messi a zero:
AND AL, 00001111b
..
```

0.2. Modo 1

Il funzionamento del Modo 0 è semplice ma non è sufficiente per molti dispositivi elettronici. Molte periferiche infatti necessitano di sapere quando il numero che viene scritto sulle linee dei dati può essere letto. Per avere questa informazione è necessario aggiungere delle linee dedicate, facendo il trasferimento con una tecnica di "handshake" ("stretta di mano"). Per il trasferimento con handshake con l'8255 si usa il Modo 1.

Sia il PortA che il PortB possono essere programmati per funzionare in Modo 1. Quando un Port funziona in Modo 1 le linee del PortC non sono più indipendenti, ma sono legate a ciò che succede nel Port corrispondente. Il Modo 1, detto "strobed I/O" o "con handshake", prevede che lo stato di alcune linee cambi automaticamente a seconda di quello che succede dentro il chip ed in altre linee.

In Modo 1 l'8255 e l'altro circuito con il quale si devono scambiare informazioni sono in grado di identificare il momento in cui quelle informazioni sono valide. Utilizzando l'8255 in Modo 0 questo non è possibile e non c'è nessun Modo per comunicare all'altro dispositivo che il dato è pronto. Per dire che il dato è valido si usano linee elettriche apposite, utilizzando i piedini dell'8255 che corrispondono al PortC. In Modo 1 il PortC non è più un Port di I/O generale ma contiene linee il cui valore è legato al funzionamento del chip. Il PortC assume la funzione di "status word", tramite la quale è possibile vedere come sta evolvendo la comunicazione fra i dispositivi.

Vediamo che significato hanno le linee del PortC in Modo 1, considerando prima il funzionamento in uscita, poi in ingresso.

0.2.0.0.0.1. Output con handshake

Supponiamo che sia il PortA che il B di un 8255 siano programmati in Modo 1 ed in uscita. In questo caso il Control Register ha questa forma:

sempre 1	Modo 1 PortA		PortA Out	Direz. PCH	Modo 1 PB	PortB Out	Direz. PCL
1	0	1	0	1 = uscita 0 = ingresso	1	0	Non importa

² si deve notare che la documentazione del produttore sconsiglia di riprogrammare "al volo" le porte perché la programmazione di una porta influenza il funzionamento delle altre, il cui valore può cambiare in seguito alla programmazione

Se si programma il CW con il numero 10100100b il significato dei bit del PortC (indirizzo Base + 1) è il seguente:

7	6	5	4	3	2	1	0
$\overline{\text{OBF}}_A$	$\overline{\text{ACK}}_A$	bit di I/O "normale"	bit di I/O "normale"	INTR_A	$\overline{\text{ACK}}_B$	$\overline{\text{OBF}}_B$	INTR_B
Strobe PA	acknowledge PA	Bit di I/O normale	Bit di I/O normale	Interrupt PA	acknowledge PB	Strobe PB	Interrupt PB
Relativi al PortA				Relativi al PortB			

Il pedice A o B significa che la linea di controllo è relativa al PortA od al PortB.

$\overline{\text{OBF}}$ = (**O**utput **B**uffer **F**ull) significa che un nuovo dato è pronto sul Port corrispondente. Normalmente è alto; va basso AUTOMATICAMENTE non appena il dato è stato scritto sulle linee del port.

$\overline{\text{OBF}}$ viene collegato al filo di strobe che va dall'8255 all'altro dispositivo: comunica all'altro dispositivo che il dato è pronto.

I piedini del chip che corrispondono al PortC1 e PC6 sono collegati al segnale di acknowledge dell'altro dispositivo. Il segnale è in logica negata: normalmente è alto e diviene basso quando l'altro dispositivo comunica l'acknowledge.

INTR = Segnale che può essere usato come Interrupt.

Si noti che i bit del PortC "usati" dal PortA sono cinque, mentre quelli usati dal PortB sono 3. La suddivisione dei bit in Modo 0 era invece 4 – 4. Il nibble alto faceva parte del gruppo A, mentre quella basso del gruppo B:

Se il PortA è in Modo 1 sono 5, invece di quattro, i bit sotto il suo controllo, perché il bit di peso 3 del PortC ha la funzione di INTR_A .

Si noti anche che ci sono due bit (PC4 e PC5) che possono essere usati come normali I/O, che funzionano come funzionerebbero in Modo 0. Se queste linee sono programmate in lettura non c'è nessun problema a leggerle: basta fare una IN sul PortC. Se invece sono di output bisogna scrivere in quei bit con il Modo a singolo bit. Consultare il data sheet del produttore se la cosa è necessaria.

La sequenza di ciò che accade quando un Port dell'8255 è programmato per l'output in Modo 1 è la seguente (supponiamo che il PortA sia programmato in uscita e facciamo riferimento ad esso, naturalmente il discorso vale identicamente anche per il PortB a patto che anch'esso sia programmato in uscita).

1. La CPU scrive sul PortA, con una OUT, il valore che vuole mettere in uscita.
2. Automaticamente, senza nessun'altra OUT da parte del programma, all'atto della scrittura nel Port del dato da emettere in uscita, l'8255 abbassa il segnale $\overline{\text{OBF}}$ (**O**utput **B**uffer **F**ull), che in condizioni normali se ne sta a livello alto.
3. Quando il dispositivo vede che la linea $\overline{\text{OBF}}$ va bassa, capisce che c'è un segnale buono da campionare sui fili che collegano i due sistemi. Perciò provvede a leggere le linee ed a utilizzare il numero che contengono.
4. Dopo aver felicemente effettuato la lettura, il dispositivo abbassa il suo segnale di acknowledge, che dà comunicazione all'8255 che il trasferimento è finito.

Vediamo un esempio in codice, visto dalla parte dell'8255:

```

..
    OUT PortA, AL    ; scrive il dato
; strobe già basso, ora l'altro legge il dato, poi abbasserà ACK
AttendoACK:
    ; leggo lo stato di ACK:
    IN AL, PortC
    AND AL, 100000b
    JNZ AttendoACK
; qui posso scrivere un altro dato perché il dispositivo ha finito di leggere
..

```

!!!!

Disegno: "Scrittura in Modo 1 con il PortA"

0.2.0.0.1.1. L'interrupt

Il segnale INTR può essere collegato al sistema d'interruzione della CPU. Inoltre può essere abilitato scrivendo un flag detto INTE (**I**nterrupt **E**nable).

La linea INTR viene alzata se i tre segnali OBF, ACK e INTE sono a uno. Ciò significa che l'interrupt viene lanciato non appena la transazione di scrittura è conclusa.

Visto come funziona INTR il flag INTE assume la funzione di abilitazione dell'interrupt (Interrupt enable, appunto): se INTE è 1 l'interrupt può essere lanciato, altrimenti ciò non succede. L'unico Modo per poter scrivere uno 0 o un 1 nel flag INTE è usare il Modo di scrittura del singolo bit, scrivendo nel bit PC6 per INTE_A e PC2 per INTE_B. Dato che abbiamo evitato di descrivere in dettaglio come si può scrivere nel singolo bit del Control Register, di più non possiamo dire.

Utilizzando l'interrupt il loop AttendoACK del programma precedente non c'è, dato che non appena il trasferimento è finito si salta direttamente alla ISR.

0.2.0.0.2. Input con handshake

L'input in Modo 1 è analogo all'output, naturalmente cambiano in Modo duale i significati dei segnali e le loro direzioni.

sempre 1	Modo 1 PortA		PortA In	Direz. PCH	Modo 1 PB	PortB In	Direz. PCL
1	0	1	1	1 = uscita 0 = ingresso	1	1	Non importa

Se si programma il CW con il numero 10110110b il significato dei bit del PortC (indirizzo Base + 1) è il seguente:

7	6	5	4	3	2	1	0
bit di I/O "normale"	bit di I/O "normale"	IBF _A	$\overline{\text{STB}}_A$	INTR _A	$\overline{\text{STB}}_B$	IBF _B	INTR _B
7	6	5	4	3	2	1	0
Relativi al PortA				Relativi al PortB			

Il significato delle sigle spiega in parte a cosa servono questi segnali:

$\overline{\text{STB}}$ significa "strobe", è negato perché la sua logica è negativa: è normalmente alto.

IBF significa **I**nput **B**uffer **F**ull e costituisce l'acknowledge che l'8255 manda all'altro dispositivo. Si alza automaticamente, senza fare nessuna OUT, quando STB va bassa e si alza quando il programma effettua una IN sul Port corrispondente, cioè quando è conclusa l'operazione di lettura. Si noti che non è negato, quindi la sua logica è positiva.

Anche in questo caso si hanno a disposizione due linee di I/O ordinarie, nella parte alta del PortC, ai bit PC6 e PC7. Anche in questo caso valgono le avvertenze per la scrittura indicate nel paragrafo precedente.

La sequenza di lettura si può perciò sintetizzare così:

1. Il dispositivo che fornisce il dato lo scrive sulle linee collegate con il Port poi
2. abbassa la sua linea di strobe
3. non appena lo strobe viene ricevuto l'8255 campiona le linee e ne memorizza i valori nei suoi latch. Contemporaneamente inizia una sequenza di acknowledge, alzando IBF
4. il programma esegue una IN sul port, leggendo il valore memorizzato nei latch
5. non appena la lettura finisce l'8255 conclude la sequenza di acknowledge abbassando IBF.

Un programma in polling che legga in Modo 1 sul PortB potrebbe essere così:

```

..
; loop di polling in attesa dello strobe:
AttendoSTB:
    IN AL, PortC      ; leggo per attendere lo strobe
    AND AL, 100b     ; maschero il bit di strobe per il PortB
    JNZ AttendoSTB
; strobe basso, ora leggo il dato da port, intanto inizia automaticamente l'ACK
    IN AL, PortB
    ; in AL ho il Byte che cercavo
    ; posso proseguire perché l'8255 ci pensa da solo a gestire
    ; la sequenza di acknowledge.
..

```

Per quanto riguarda l'interrupt in lettura il discorso è del tutto analogo a quello già fatto per la scrittura. L'interrupt, se è abilitato dal flag INTE, viene lanciato non appena giunge lo strobe. Questo fa capire alla CPU che il dispositivo cui siamo collegati ha pronto un dato che deve essere letto. La ISR scatterà ed andrà a leggere il dato.

Si noti che il produttore sconsiglia il collegamento "diretto" di due 8255 che funzionino in Modo 1 o anche in Modo 2, perché le temporizzazioni dei segnali in lettura e scrittura non sono del tutto compatibili. Il collegamento diretto fra due 8255 può funzionare per cavi corti o in condizioni fortunate, ma se si vuole un sistema affidabile bisogna invertire il segnale IBF prima di farlo andare nel piedino ACK dell'altro 8255 ed aggiungere almeno un ritardo per rendere compatibile i tempi dei due dispositivi. Nel sito citato si possono trovare indicazioni al riguardo.

0.2.1. Modo 2

0.2.2. Programmazione del dispositivo

0.2.3. Schede che usano l'8255

0.2.4. Uso dell'8255 nella motherboard del PC

In Modo 2 la direzione del trasferimento è stabilita dall'hardware. Per questo si dice che questo Modo di funzionare è "bidirezionale".

Anche in Modo 2 la sincronizzazione avviene per mezzo di un handshake di tipo strobe – acknowledge ed esistono segnali di interrupt che possono essere disabilitati con due flag INTE, uno per l'interrupt in ricezione e l'altro per l'interrupt in trasmissione.

Solo il PortA può essere programmato in Modo 2. In Modo 2 cinque dei bit del PortC contengono le linee per il controllo e la lettura dello stato del PortA. Gli altri tre bit sono ancora usati dal PortB, se esso funziona in Modo 1 oppure sono normali I/O digitali, se il PortB funziona in Modo 0.

Il verso del trasferimento è automaticamente comandato in hardware a seconda se si scrive o si legge nel PortA. In questo caso il circuito funziona come un bus, in Modo simile al collegamento fra CPU e memoria. La differenza è che ci si collega ad un dispositivo che potrebbe essere molto più lento di una memoria.

Dato che l'8255 si occupa di impostare in hardware la direzione del trasferimento basterà scrivere nel PortA con una OUT per effettuare un trasferimento dall'8255 al dispositivo esterno, mentre basterà leggere con una IN per effettuare il trasferimento contrario.

Il software ha la vita facile in Modo 2. Per effettuare una lettura, dopo aver programmato il PortA in Modo 2, basterà fare una IN, una OUT per la scrittura.

```

..
; spedisco il valore di AL all'altro dispositivo:
OUT PortA, AL
; non devo fare nient'altro, aspettare alquanto,
; l'8255 si arrangia da solo ..

; leggo ciò che il dispositivo ha memorizzato nell'8255:
IN AL, PortA
; niente polling, niente di niente
..

```

Molte schede di I/O per PC fanno uso di 8255. Spesso una scheda da aggiungere nel bus di espansione del PC comprende due 8255, dando luogo ad una capacità massima di 48 I/O digitali (in Modo 0).

