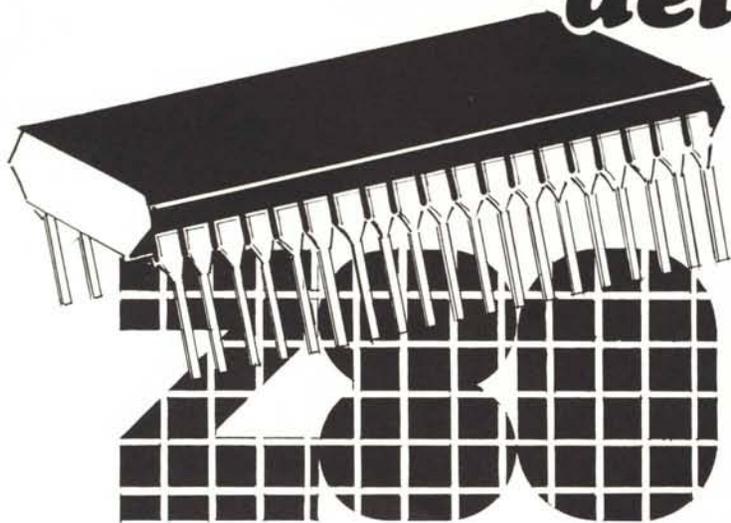


L'ASSEMBLER dello



di Pierluigi Panunzi

In questo numero tratteremo l'argomento dell'I/O (Input/Output), che viene realizzato tramite una serie di istruzioni che fondamentalmente, come meccanismo, ricalciano quelle viste nella precedente puntata.

La gestione dell'I/O

Vediamo innanzitutto brevemente cosa si intende per I/O e soprattutto in cosa si differenzia dalle usuali operazioni sulla memoria.

Per I/O si intende tutta una serie di operazioni software ed hardware che permettono al microprocessore, nel nostro caso lo Z80, di colloquiare con l'esterno attraverso dispositivi particolari, quali registri, porte, ecc.

Come dice il nome, infatti, l'I/O in genere è dedicato a tutto quanto occorra per permettere all'"uomo" di interagire con il microprocessore, nonché al microprocessore stesso di gestire i dati provenienti da altri dispositivi, ad esempio elementi di misura, convertitori analogico-digitali, trasduttori o altri microprocessori.

In generale il meccanismo di I/O si differenzia da quello di gestione della memoria a cominciare dal modo di "indirizzamento": in particolare nello Z80 abbiamo modo di indirizzare fino a 64K locazioni di memoria (siano esse RAM o EPROM), nonché fino a 256 dispositivi (ed in genere indirizzi) in Ingresso o in Uscita.

Dato che l'I/O è diretto verso vari dispo-

sitivi elettronici in generale dotati di registri di memoria, ad essi è intuitivo assegnare un indirizzo, che in nessun modo potrà essere in conflitto un indirizzo di memoria.

Facciamo un esempio: supponiamo di avere la possibilità di collegare 256 dispositivi (porte, convertitori, latch, ecc.). Ad ognuno di questi dispositivi è associato un indirizzo, compreso tra 0 e 255, al quale lo Z80 fa riferimento con istruzioni particolari, dette appunto di I/O, che permettono la "lettura del dispositivo" (Input - istruzione IN) e la "scrittura sul dispositivo" (Output - istruzione OUT).

Un esempio più concreto può essere fornito da un serializzatore di segnali, per intenderci quel particolare microprocessore dedicato all'invio su una linea (ad esempio telefonica tramite "modem") dei dati forniti dal microprocessore stesso: tale componente viene visto in genere come un insieme di registri di I/O consecutivi, aventi ad esempio indirizzi compresi tra 0 e 7.

Effettuando una scrittura in uno di questi registri, ad esempio, si può caricare un dato da inviare sulla linea, dato che verrà effettivamente inviato allorché si scriverà un opportuno dato in un altro dei registri.

Viceversa un dato proveniente dalla linea potrà essere "letto" dalla CPU, semplicemente andando a leggere un altro registro di quelli disponibili.

Le istruzioni di I/O

Cominciamo il discorso dell'I/O procedendo parallelamente per quanto concerne

l'Input e l'Output: concettualmente le due operazioni sono del tutto analoghe, dal punto di vista dell'Assembler, in quanto a questo livello ciò che cambia è semplicemente il "verso" in cui viaggiano le informazioni.

Supponiamo perciò di avere una certa porta dalla quale vogliamo leggere un dato e sulla quale inoltre vogliamo scrivere un altro: tale porta avrà un proprio indirizzo (come visto compreso tra 0 e 255), che indicheremo d'ora in poi con "n".

Detto che il meccanismo di lettura e/o scrittura coinvolge principalmente l'accumulatore (A), abbiamo le due prime istruzioni più semplici, elementari, che si trovavano anche nell'8080, il predecessore dello Z80.

Tali istruzioni sono:

IN A,(n)
e
OUT (n),A

e dovrebbe già essere chiaro il loro meccanismo: la prima trasferisce nell'accumulatore il contenuto della porta "n" e viceversa la seconda deposita il contenuto dell'accumulatore sulla porta di indirizzo "n".

Come si vede si tratta di istruzioni molto semplici e spesso usate, soprattutto a livello di "Sistemi Operativi": si può però vedere che ricordano molto le normali istruzioni LD di gestione della memoria.

Per vedere però ancora meglio la differenza tra le istruzioni di I/O e quelle relative alla memoria, andiamo a vedere più da vicino come si comporta nei due casi il microprocessore nei confronti dei dispositivi che lo circondano.

Iniziamo dalla memoria.

Abbiamo detto che lo Z80 può indirizzare fino a 65536 locazioni di memoria (celle in cui sono posti dati formati da 8 bit e cioè un byte): per indirizzare una di queste 65536 celle il microprocessore ha a disposizione 16 piedini ai quali fa capo il cosiddetto *Address Bus* (abbreviato con AB), mentre per gestire i dati ad 8 bit ha a disposizione altrettanti piedini, che prendono il nome di *Data Bus* (abbreviato con DB).

Ora, quando lo Z80 deve scrivere un certo dato in una locazione di memoria, nell'AB sarà presente proprio l'indirizzo della cella desiderata, mentre nel DB sarà presente il dato che vogliamo scrivere. Analogamente accade, ma non scenderemo nei particolari, nel caso che lo Z80 voglia leggere il dato contenuto in una locazione di memoria.

Nel caso di un'operazione di I/O, ed in particolare per le istruzioni IN ed OUT viste poco fa, si ha la situazione seguente:

— nelle prime 8 linee dell'AB (cioè quelle che in gergo vengono indicate con A₀, A₁, fino ad A₇) viene posto il valore "n" e cioè il numero (indirizzo) della porta a cui si fa riferimento

— nelle successive 8 linee dell'AB (da A₈ ad A₁₅) viene posto il contenuto dell'accumulatore

— nelle 8 linee che formano il DB (da

D₀ a D₇) viene posto di nuovo il contenuto dell'accumulatore.

Per tutte e due le istruzioni viste (e cioè IN ed OUT), vale la regola che i flag non vengono assolutamente alterati, come dire che anche se ad esempio si legge da una porta un dato nullo, il flag di Zero (Z) non verrà minimamente toccato.

Altre due istruzioni, che sono più generali di quelle viste e che mancavano totalmente nell'8080, sono

IN r,(C)

e

OUT (C),r

dove "r" può essere uno qualsiasi dei registri dello Z80 e cioè A, B, C stesso, D, E, H ed L.

La prima istruzione scrive nel registro "r" indicato il contenuto della porta, il cui indirizzo era posto nel registro C e viceversa la seconda scrive, nella porta il cui indirizzo è posto in C, il dato contenuto nel registro "r" specificato.

In questo caso, fermo restando che il DB contiene il dato da inviare oppure da ricevere, si ha un comportamento caratteristico per le linee dell'Address Bus:

— nelle linee A₀-A₇ è presente il dato contenuto del registro C e cioè l'indirizzo della porta

— nelle rimanenti linee dell'AB è presente il contenuto (attenzione!) del registro B, che però in questa istruzione non gioca un ruolo particolare, se non è esplicitamente indicato come registro "r".

Per quanto riguarda i flag si ha che l'istruzione IN r,(C) altera, in base al valore del dato, i flag Z, S, e P, che indica in questo caso la parità del dato.

Viceversa l'istruzione OUT (C),r, lascia inalterati i flag.

Le istruzioni di I/O per blocchi di dati

Analogamente al caso visto la scorsa puntata delle istruzioni di spostamento e di comparazione di blocchi di memoria, con lo Z80 è possibile inviare ad una porta, o ricevere da questa, un blocco di dati.

Questo blocco di dati sarà posto in memoria a partire dall'indirizzo contenuto in HL e sarà formato da un numero di byte contenuto nel registro B (perciò al massimo saranno 256 byte) ed inoltre l'indirizzo della porta in questione sarà contenuto nel registro C.

Sarà possibile perciò effettuare l'invio o la ricezione di un dato (uno solo alla volta) ottenendo automaticamente l'incremento o il decremento della coppia HL ed in ogni caso il decremento del contatore, identificato dal registro B: a sua volta sarà possibile ripetere automaticamente l'operazione di invio o ricezione fino a che tale contatore non si è annullato.

Come si vede, il tutto ricorda molto le varie LDIR, LDDR ecc. Anche i nomi sono mnemonici al punto giusto per suggerire quello che l'istruzione permette di eseguire.

Ecco perciò che abbiamo 8 istruzioni, 4

di input ed altrettante di output, che riassumiamo in una tabella sintetica:

Istruzione	HL	ripete?
INI	input incr.	no
INIR	input incr.	si
IND	input decr.	no
INDR	input decr.	si
OUTI	output incr.	no
OTIR	output incr.	si
OUTD	output decr.	no
OTDR	output decr.	si

Su questa tabella possiamo fare alcune considerazioni:

— ogni istruzione decrementa di un'unità il contenuto del registro B.

— le istruzioni il cui nome termina con la lettera "R" effettuano la ripetizione dell'operazione di I/O fino a che il contatore B non raggiunge il valore nullo.

— le istruzioni senza la "R" settano il flag di Zero (Z), se l'operazione di decremento fa arrivare il registro B a zero.

— i rimanenti flag non vengono alterati.

Nei casi di funzioni di input, il dato, proveniente dalla porta il cui indirizzo è posto in C, viene posto nella locazione di memoria il cui indirizzo è posto in HL.

Invece nei casi di istruzioni di output, il dato, contenuto nella cella il cui indirizzo è posto in HL, viene inviato alla porta il cui indirizzo è contenuto nel registro C.

In simbologia corrente si ha che

istruzioni	trasferimento
INPUT	(HL) ← (C)
OUTPUT	(C) ← (HL)

intendendo con "(C)" la porta il cui indirizzo è posto in C. Terminiamo questa puntata riguardante le istruzioni di I/O dicendo che, in tutti e 8 i casi visti or ora, sul DB si ha il dato che deve essere inviato o ricevuto ed inoltre si ha che:

— nelle linee A₀-A₇ è posto il contenuto del registro C e perciò, come in tutte le istruzioni di I/O, il valore dell'indirizzo della porta interessata: ciò è importante a livello hardware per poter indirizzare fisicamente la porta desiderata tra quelle effettivamente disponibili.

— nelle linee A₈-A₁₅ è infine posto il contenuto di B e cioè del contatore: anche in questo caso il fatto di avere disponibile "fisicamente" il contenuto del contatore può essere ancora una volta utile a livello hardware per la corretta progettazione dei dispositivi di I/O.

Un esempio di personal computer che sfrutta questo particolarissimo comportamento delle istruzioni di I/O per la gestione (geniale) della tastiera è il ben noto ZX81 della Sinclair: su questo argomento i lettori potranno trovare i dettagli nel n.5 di MC, nell'articolo riguardante la prova dello ZX81 ed in particolare in un riquadro posto a pagina 36.

Nella prossima puntata affronteremo un altro argomento molto delicato e per certi versi complesso: la gestione degli interrupt da parte dello Z80. **MC**

HAROTEK

by B&V

ACCESSORI PER COMPUTER

Viale Roma, 168

47100 FORLI'

☎ 0543/67078 - 52190

SPEDIZIONI IN TUTTA ITALIA

SINCLAIR QL

INTERFACCIA DRIVE PER QL SINCLAIR	L. 300.000
+ DRIVE DA 5" 1/4 - 1 M. BYTE	L. 780.000
INT. + DRIVE DA 5" 1/4 - 200 K BYTE	L. 690.000
CAVO RS 232 QL SINCLAIR	L. 45.000
INT. CENTRONICS PER QL SINCLAIR	L. 110.000
CAVO JOYSTICK PER QL SINCLAIR	L. 25.000

ZX SPECTRUM

INT. STAMPANTE SENZA SOFTWARE	
RS 232 COMPLETA DI CAVO	L. 110.000
INT. STAMPANTE SENZA SOFTWARE	
CENTRONICS COMPLETA DI CAVO	L. 110.000
ESPANSIONE DI MEMORIA 32 K	L. 65.000
MODEM CON INT. PER ZX SPECTRUM	L. 295.000
INT. JOYSTICK PROGRAMMAB. AL 100%	
CON SOFTWARE "TURBO"	
CARICA IN POCHI SECONDI	L. 78.000
INT. KEMPSTON CON 2 ATTACCHI	L. 30.000

CBM 64

MODEM CON INTERFACCIA PER CBM 64	L. 250.000
INTERFACCIA RS 232 PER CBM 64	L. 66.000
BACK UP PER 2 REGISTRATORI	L. 50.000
INT. PRINTER PER 64 CENTRONICS	L. 110.000
SPEECH PER CBM 64	L. 52.000
PROGRAMMATORE DI EPROM CON	
INTERFACCIA RS 232 PER TUTTI I TIPI	
DI COMPUTER, PROGRAMMA 6 TIPI	
DI EPROM	L. 290.000

e tanti altri accessori:

RICHIEDETE IL LISTINO "GRATUITO"

SCONTI PER RIVENDITORI

Distributore per Emilia Romagna e Toscana:

GIERRE INFORMATICA - Viale Umbria, 36
Reggio Emilia - Tel. 0522/51.23.45

Distributore per le Marche:

CIDI di Catozzi - Via Maierini, 10
Senigallia (An) - Tel. 071/63.343

SPEDIZIONI CONTRASSEGNO

**SPESE POSTALI
A CARICO DEL DESTINATARIO**