



Non si può certo negare che per molte applicazioni specialistiche, dall'ingegneria civile all'elettronica, dalla matematica teorica alla statistica, un personal computer sia veramente un ottimo aiuto. Praticamente tutte le macchine sono ormai dotate di interpreti Basic sufficientemente potenti per svolgere tutte le operazioni necessarie. La gestione e il trattamento di matrici non rientra, però, nelle prestazioni "standard" neppure degli interpreti più evoluti: quando si deve invertire una matrice o calcolarne il determinante (p.es. risoluzione di sistemi di equazioni) è perciò necessario realizzare routine in Basic di complessità spesso notevole, che impegnano non poco sia il programmatore, sia successivamente la macchina che deve eseguire le numerose operazioni che i calcoli matriciali richiedono. Per alcune macchine, molto poche per la verità, sono disponibili ROM opzionali che, una volta installate, aggiungono la possibilità di trattare le matrici. Una di queste macchine è il Commodore 8032, per il quale la HSH di Padova ha realizzato una ROM matrici. La HSH è uno dei più qualificati produttori di software per Commodore, in tutti i campi e in particolare

HSH MATRIXPET

ROM MATRICI PER COMMODORE 8032

di Pierluigi Panunzi

in quello dell'ingegneria edile, in cui ha sviluppato numerosi package anche con l'uso del plotter Digiplot della Watanabe. I calcoli matriciali sono molto frequenti nell'ingegneria civile e forse questo ha spinto la HSH alla realizzazione della ROM. Non è questo, tra l'altro, tutto l'hardware di produzione HSH: c'è una ROM con il Basic Plus, una per la gestione dell'espansione a 96K (vedi riquadro nel testo), oltre ad alcune interfacce per collegamenti vari.

In queste pagine presentiamo il "Matrixpet", ossia la ROM che contiene un insieme di routine che permettono di effettuare un considerevole numero di operazioni matriciali, dalla semplice inizializzazione al "non plus ultra", l'inversione.

Prima però di parlare della ROM, diamo uno sguardo al computer che può ospitarla con particolare riguardo per le sue caratteristiche hardware.

Il PET Commodore CBM 8032

Ormai da parecchio tempo sul mercato, il CBM 8032 è il risultato dell'affermata tendenza dei costruttori a migliorare i propri modelli nel corso degli anni. All'inizio apparve sul mercato il PET (appellativo inglese per un animaletto domestico) 2001, con la sua strana tastiera rettangolare sulla destra ed il registratore a cassette incorporato sulla sinistra, con lo schermo video da 1000 caratteri. Era per l'appunto un "cucciolo" di una famiglia in crescita.

Poi venne il PET 3032 con una tastiera di dimensioni "regolamentari", ma ancora con i simboli grafici sui tasti, senza il registratore a cassette, ma con possibilità di collegamento ad un floppy disk driver e ad una stampante.

Infine è arrivato il PET 8032 (soprannominato "il PET grande"), con il suo schermo da 2000 caratteri (25 linee da 80 caratteri), con un esteso set comprendente sia le minuscole (all'accensione) che le maiuscole, con una tastiera ASCII di caratteristiche standard e con una memoria da 32K espandibile. È inoltre dotato di una serie di connettori per il collegamento con altre apparecchiature: uno slot (non standard) per la connessione secondo il protocollo IEEE-488 (HPIB), una porta di ingresso-uscita e due porte di collegamento con altrettanti registratori a cassette.

L'hardware

Il "cuore" dell'8032, come pure dei predecessori, è il microprocessore 6502, che risulterà ben noto ai lettori in quanto presente in altri personal computer (Apple, Atari per citarne un paio) e in piastre a microprocessore (AIM 65).

Pur non possedendo un vasto set di

Realizzazione e distribuzione:
HSH Hardware Software House
Via Falloppio 39, 35100 Padova

Prezzi:

ROM Matrixpet: L. 400.000 + IVA
ROM Matrixpet + software di base per la gestione dell'espansione a 96K: L. 600.000 + IVA

istruzioni (a paragone, ad esempio, con lo Z80), il 6502 presenta notevoli caratteristiche di praticità d'uso dovuta alla semplice architettura del microprocessore e dei circuiti ad esso collegati: per quanto riguarda la programmazione in linguaggio assemblativo si hanno svariate possibilità di indirizzamento, tra le quali ad esempio quella "in Pagina Zero", che si riferisce a locazioni di memoria aventi indirizzo compreso tra \$0000 e \$00FF.

Passando alla gestione della memoria nel "grande PET", si hanno a disposizione 32K di memoria RAM (Random Access Memory), compresi tra gli indirizzi \$0000 e \$7FFF, così suddivisi:

- da \$0000 a \$00FF, come visto, la "Pagina Zero" (Zero Page)
- da \$0100 a \$0278 la zona comprendente lo stack, il buffer per il BASIC e per la tastiera
- da \$027A a \$0339 il buffer per il primo registratore a cassette
- da \$033A a \$03F9 il buffer per il secondo registratore.



Con la ROM, la HSH fornisce una piastrina adesiva da applicare sul frontale del Pet.

Invece da \$0400 a \$7FFF è presente la RAM riservata ai programmi redatti in BASIC dall'utente e comprende vari settori contigui di ampiezza variabile dinamicamente, rispettivamente per le istruzioni costituenti il programma, le variabili, le matrici ed i vettori, e le stringhe.

Normalmente questa parte di memoria viene gestita automaticamente dal Sistema Operativo, soprattutto per quanto riguarda gli indirizzi "di frontiera" tra le varie

zone: istruzioni - variabili - matrici - stringhe.

Variando però opportunamente i contenuti di alcune locazioni di memoria in pagina zero (riempite dal Sistema Operativo all'accensione della macchina), si può ad esempio fissare un limite per la memoria dedicata al programma lasciando tutta la rimanente per altri scopi.

Proseguendo verso indirizzi più alti, troviamo, a partire da \$8000, 4K di RAM per la gestione del video, delle quali le prime 2000 locazioni contengono istante per istante i codici dei caratteri visualizzati sullo schermo.

Quindi dall'indirizzo \$9000 a \$BFFF vi è un "buco" di 12K, previsto come area per le ROM con cui si espandono le caratteristiche del sistema (è appunto il caso della EPROM che stiamo presentando).

Infine lo spazio di 16K compreso tra gli indirizzi \$C000 e \$FFFF contiene l'interprete BASIC, il Sistema Operativo, la gestione della tastiera e del video, il controllo degli interrupt e dell'I/O.

La EPROM "Matrixpet"

Eccoci dunque alla EPROM, realizzata dalla HSH di Padova, che, come vedremo, presenta delle caratteristiche decisamente interessanti.

Si tratta di una EPROM da 4K da inserire nello zoccolo UD11 del circuito del PET, cioè nel primo dei due zoccoli liberi presenti nella piastra del computer; in alternativa si può inserire la EPROM in una basetta che va collegata allo zoccolo di cui sopra tramite un cavetto piatto: questo dà la possibilità a chi possiede più di una ROM (Basic Plus, ad esempio) di eseguirne la sostituzione senza aprire ogni volta la macchina.

Perché la ROM possa funzionare, è necessario connettere un piccolo "scatolino" allo slot posto sul lato destro del PET, riservato al secondo registratore. Questo "scatolino", serve per evitare copie della ROM ed è, in pratica, una "chiave di accesso", senza la quale ad ogni impostazione di "nuove" funzioni la macchina risponde con "?syntax error". Per inciso, anche se di dimensioni ridotte, la chiave sporge di qualche centimetro dal "profilo" del computer e perciò è richiesta una certa attenzione per non urtarla inavvertitamente specie quando il computer è acceso.

Insieme a questi due componenti viene fornito un esauriente manuale d'uso (in italiano, il che di certo non guasta!) che ci rende prontamente edotti sulle caratteristiche dell'estensione BASIC, fornendo per ogni nuovo operatore informazioni dettagliate sulla sintassi (corretta impostazione della funzione e degli operandi) e sulla semantica (la spiegazione di ciò che viene

eseguito), il tutto completato da alcune note ed esempi d'uso.

Prima di passare alla descrizione dei vari operatori disponibili, segnaliamo che per inizializzare i calcoli matriciali (cioè per selezionare la EPROM) si deve impostare il comando BASIC "sys (40960)", da programma oppure direttamente: infatti sys(n) è un comando che permette di eseguire, dal BASIC, una routine in linguaggio macchina posta in memoria a partire dall'indirizzo decimale "n".

Gli operatori

Sono in tutto 33 e si possono utilizzare come normali istruzioni BASIC espandendo così il set originario. Data la relativa novità e la ricchezza di possibilità, andiamo ad analizzarli singolarmente.

In pratica queste nuove istruzioni sono suddivise in due blocchi logici: istruzioni per matrici multidimensionali e per matrici bidimensionali.

Occupiamoci dunque degli operatori del primo tipo ed indichiamo con lettere maiuscole le matrici, le cui dimensioni dovranno essere impostate con un'istruzione "dim".

- aprt A: stampa o visualizzazione della

279	632	593	511	32	613	424	887
711	668	68	852	746	766	813	685
742	444	934	197	722	997	966	881
569	373	515	558	616	542	485	181
816	881	154	885	781	216	48	481
161	83	558	28	412	358	681	828
279	587	486	252	91	342	488	183
844	681	683	95	281	981	79	597

il determinante e' 1.78829395e+21

ready.

Un esempio del calcolo del determinante di una matrice 8x8.

matrice A per colonne

- ara A = B*C: calcolo degli elementi di A come prodotto degli elementi corrispondenti di B e di C

- ara A = B+C: calcolo degli elementi di A come somma degli elementi corrispondenti di B e di C

- erase A: cancella la matrice A

- do <stringa>: per questa istruzione vedere il paragrafo seguente

- dup A = B: pone la matrice A uguale alla matrice B

- ina A = <espr.>: inizializza al valore di

<espr.> tutti gli elementi di A

- ini A: azzerata la matrice A

- e = max A: la variabile e assume il valore massimo degli elementi di A

- e = min A: come sopra, per il valore minimo

- mrea A: lettura di A per colonne da istruzioni "data"

- mrea#n,A: lettura di A per colonne dal disco (canale#n)

- msa#n,A: memorizzazione di A per colonne su disco (canale#n)

- mstd A = B: gli elementi del vettore B(n) vengono posti sulla diagonale principale di A(n,n)

- rmvd B = A: gli elementi della diagonale principale di A(n,n) vengono posti nel vettore B(n)

- rdm A: ridimensiona A, però nello stesso spazio di memoria precedentemente occupato da A

- msca A* <espr.>: moltiplica ogni elemento di A per il risultato di <espr.>

- msum A* <espr.>: somma ad ogni elemento di A il valore di <espr.>

Inoltre vi è una serie di istruzioni che opera nel seguente modo: ogni elemento $a_{i,j}$ della matrice A viene sostituito da $f(a_{i,j})$ dove le possibili funzioni $f(x)$ sono:

istruzione	$f(x)$
invl A	$1/x$
mab A	$ x $
mco A	$\cos(x)$
mcs A	$-x$
mex A	e^x
mlo A	$\ln(x)$
mnt A	$\text{Int}(x)$
msg A	$\text{sign}(x)$
msi A	$\sin(x)$
msq A	x
mta A	$\text{tg}(x)$
mtn A	$\text{arctg}(x)$

Il secondo set di operatori invece si riferisce a matrici a due dimensioni e comprende le istruzioni:

- idn A: pone la matrice quadrata A come matrice unità (cioè pone la diagonale principale ad 1)

- d = inv A: calcola l'inversa della matrice quadrata A ed il suo determinante; su questa istruzione ritorneremo più oltre.

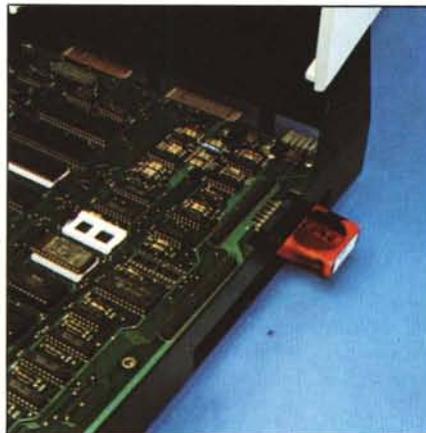
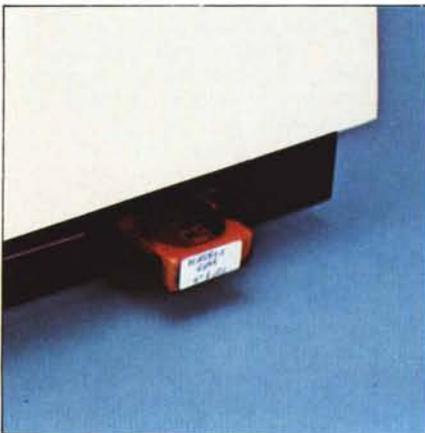
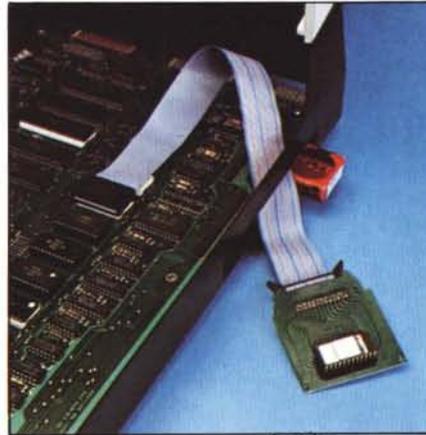
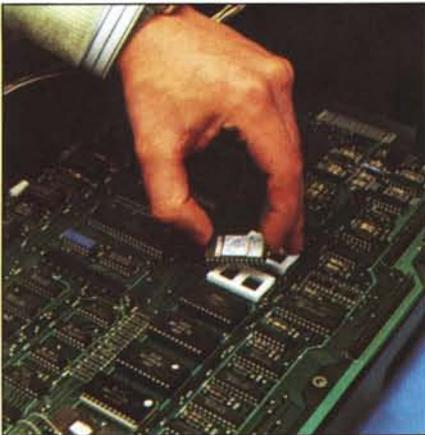
- mat A = B*C: effettua il ben noto prodotto "righe per colonne" tra le due matrici B e C, ponendo il risultato in A, previa verifica di correttezza delle dimensioni delle tre matrici

- trn A = B: calcola la matrice A come trasposta della matrice B.

La funzione "do <stringa>"

Questa funzione consente, da BASIC, di effettuare una qualsiasi istruzione o serie di istruzioni contenute in "stringa": cioè, supponendo di aver eseguito l'istruzione a\$ = "goto 500" con "do a\$" viene appunto eseguita l'istruzione "goto 500".

È notevole il fatto che a\$ può essere manipolata come si vuole da programma



La EPROM può essere inserita direttamente nello zoccolo UD11; per chi possiede più di una ROM (p.es. Basic Plus) la HSH fornisce una basetta con un cavo piatto, in modo che si possa sostituire la ROM senza aprire la macchina. Notare, nella foto, la "chiave di accesso" inserita nello slot del secondo registratore a cassette, sul lato destro del computer.

(con funzioni del tipo "left\$", "right\$", ecc.) ed è proprio il suo "contenuto" che viene eseguito dalla "do".

Altra possibilità è di crearsi una matrice di stringhe, ad esempio a\$(9,9), contenente in ogni elemento una o più istruzioni BASIC, eseguibili poi in seguito da programma, con "do a\$(i,j)".

Inutile dire che in questo modo si possono realizzare programmi autogenerantisi, fatto che, normalmente, richiede la conoscenza di come il sistema operativo del PET gestisca le istruzioni BASIC.

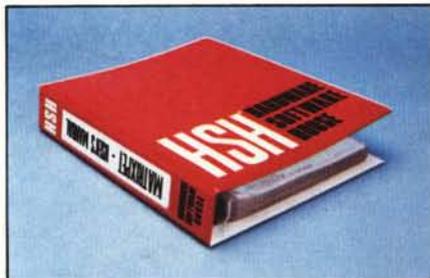
L'istruzione d=inv A

Abbiamo lasciato per ultimo, ma non certo per importanza, il "fiore all'occhiello" della EPROM: la possibilità di calcolare la matrice inversa di A(n,n).

È ben noto che la determinazione dell'inversa di una matrice è il problema di gran lunga più complicato tra quelli riscontrabili in matematica e soprattutto nelle sue applicazioni in campo scientifico.

È anche ben noto che su questo problema sono stati scritti fiumi di trattati, inventati metodi più o meno disparati, che costringono il povero programmatore a districarsi tra indici, pivot, scambi di righe o di colonne ecc.

Come riporta il manuale d'uso, una volta calcolati o impostati gli elementi della matrice A, la routine chiamata dall'istruzione "inv" effettua l'inversione impiegando il "metodo di Gauss-Jordan con pivoting



La ROM è corredata da un voluminoso ed esauriente manuale, costituito da un raccogliatore ad anelli con copertina di plastica rossa.

totalmente modificato, con algoritmo stabile in senso forte".

Una volta eseguita l'operazione (in tempi di cui parleremo più avanti), la matrice inversa occuperà la zona di memoria occupata originariamente dalla matrice A. Inoltre viene effettuato il calcolo del determinante di A, valore che viene posto, se si desidera, nella variabile d.

Ora se la matrice A è singolare oppure se durante i calcoli con l'operazione di pivoting si arriva (per la propagazione degli errori a causa di successive approssimazioni) ad un valore nullo del determinante, fatto che può capitare se A è di tipo particolare e che senza dubbio è uno "spauracchio" per i matematici, allora il calcolo viene interrotto e la matrice A viene distrutta: infatti in parte conterrà elementi "vecchi" ed in parte elementi "nuovi". Buona strategia è di duplicare la matrice A in un'altra matrice prima di effettuare l'inversione.

Inoltre c'è da dire che la matrice A(n,n), contenente in realtà (n+1)x(n+1) elementi in quanto per il BASIC gli indici variano tra 0 ed n, viene considerata invece senza la riga "0" e la colonna "0", riducendola perciò ad una matrice nxn. Questa riga e questa colonna vengono utilizzate per contenere via via i valori successivi del determinante e gli indici di scambio di righe o di colonne, propri dell'algoritmo usato.

Infine vengono usati i due buffer per i registratori a cassette per la memorizzazione dei dati intermedi.

Veniamo ora al problema dei tempi di elaborazione, fortemente dipendenti dalle dimensioni della matrice A: abbiamo effettuato alcune prove con un piccolo programma che, dimensionata la matrice ad un valore n impostato, ne generava casualmente tutti gli elementi e poi effettuava il calcolo dell'inversa.

Grazie all'orologio interno, accessibile con la variabile ti\$, abbiamo misurato i tempi di calcolo.

Per n fino a 6-7 la risposta è quasi istantanea, mentre successivamente si ha: per n=10, 5 secondi; per n=20, 47 secondi; per n=30, 3 minuti; per n=40, 8 minuti; per n=50, 17 minuti e 24 secondi.

Abbiamo infine "girato" il programma (che non a caso era stato denominato "tortura") con il valore massimo possibile per n: 78, valore oltre il quale il computer segnala "Out of memory error".

Ebbene per calcolare il determinante e l'inversa di una matrice 77 x 77 (e cioè formata da ben 5929 elementi generati casualmente e compresi tra 0 ed 1), il PET ha impiegato qualcosa come 1 ora 24 minuti e 46 secondi.

È alquanto insolito dover affrontare problemi con matrici di tali proporzioni, ma ricordiamo che l'inversione di matrici anche di dimensioni più consuete richiede, in Basic, tempi di elaborazione enormemente più elevati oltre naturalmente, alla necessità di realizzare il programma adeguato.

Conclusioni

Le possibilità offerte da questa EPROM sono senza dubbio molto interessanti: una volta inserita nello zoccolo ci si può dimenticare di lei, ma certo non ci dimenticheremo che possediamo un computer con "una marcia in più", da innestare ogni volta che ci si presenteranno complicati calcoli matriciali, i cui risultati ci verranno forniti in breve tempo, a partire da istruzioni particolarmente semplici.

Tanto, se sbagliamo qualcosa o ci dimentichiamo il nome di una funzione, ci sarà di grande aiuto il manuale "made in Italy".

Il prezzo non può essere considerato contenutissimo ma, obiettivamente, neppure troppo elevato tenendo presenti le prestazioni della ROM.

Espansione a 96K per il PET Commodore CBM 8032

Oltre alla EPROM per calcoli matriciali, la HSH ne fornisce un'altra dedicata alla gestione della scheda di espansione a 96K di RAM del CBM 8032, distribuita dalla Harden. Tale gestione avviene considerando l'espansione di memoria come un vettore, chiamato "vrc", avente un numero di componenti variabile a seconda del "tipo" di dati introdotti: caratteri, numeri interi o numeri reali.

L'elemento n-esimo di tale vettore potrà essere identificato semplicemente con vrc(n), con n variabile tra 0 ed un valore massimo secondo il seguente criterio: se il tipo è "carattere" ogni elemento occuperà un solo byte del vettore e si avrà $n_{max} = 65532$; se il tipo è "intero" ogni numero impostato occuperà due byte (valore compreso tra -32768 e 32767) ed $n_{max} = 32705$; se il tipo è invece "reale" ogni numero occupa 5 byte e si avrà $n_{max} = 13105$. In tutti e tre i casi la codifica avviene secondo le regole del CBM 8032 e quindi non si hanno problemi di compatibilità.

Per selezionare la EPROM si deve introdurre all'inizio del programma BASIC l'istruzione sys (40960): successiva-



mente con l'istruzione "declare n" si stabilisce il contenuto del vettore vrc, a seconda del valore di n: n=1 per il tipo "carattere", n=2 per i numeri interi, ed n=5 per i reali. Altri valori di n comporteranno la segnalazione di un errore.

Per quanto riguarda infine le componenti del vettore, queste possono essere indirizzate, come detto, con "vrc (<expr.>)", dove <expr.> può essere un'espressione qualsiasi, ma il cui valore numerico deve cadere nei campi permessi, altrimenti si avrà segnalazione di errore (Insufficient Memory").

Infine, per disabilitare la EPROM bisognerà impostare il comando "exit".