

128 da zero

di Andrea de Prisco

Video Display Controller 8563

A partire da questo mese, 128 da zero si occuperà dell'integrato 8563 che provvede alla visualizzazione del modo testo a 80 colonne o alla fatidica grafica 640 × 200. Fatidica in quanto, come altre cose di questo benedetto computer, «c'è ma non si vede».

Così come non si vedono i 16 K di memoria di ram necessari per visualizzare tale pagina grafica, i 37 registri interni a questo processore, i modi di visualizzazione a un numero variabile di colonne, i caratteri a doppia altezza, i sipari e tant'altro. Come dire... procediamo con ordine.

Caro vecchio indimenticabile VIC

Ricordo i bei tempi del mio VIC-20 come un periodo in cui il computer era essenzialmente un nemico da sconfiggere. Inoltre l'oggetto in questione sembrava fatto apposta per tale scopo: con la macchina veniva dato uno stupido manualletto con su scritto «Vic 20, come giocare con un vero computer» e dal canto mio il gioco consisteva per l'appunto nello scoprire tutto (sottolineo tutto) quanto non fosse scritto su quelle pagine. E giù con peek e poke nei posti più assurdi, monitor di linguaggio di macchina per capire il «dentro», implementazione di grafica a risoluzione più alta di quanto era riuscita a fare mamma Commodore (vedi MC n. 16 pagg. 64 e 65) e altre, come direbbero a Napoli, «paz-zielle» simili.

Centoventotto alla mano sembra proprio lo stesso andazzo, con l'unica differenza che ci sono rimasto molto male quando, pochi giorni dopo l'acquisizione di una di queste bestioline,

sempre col solo (inutile) manuale fornito con la macchina, impazzivo non riuscendo a trovare la pagina video 80 colonne nei due banchi di memoria di cui la macchina dispone. La risposta era semplice: da qualche parte ci doveva stare dell'altra ram. Punto e basta.

Questo fino a quando non è venuta alla luce una bella guida *quasi-di-riferimento* sulla quale si leggono un po' di cose carine: state calmi, non è in vendita in Italia; e poi è anche piena zeppa di imprecisioni. Fatto sta però che i kappa in più saltano fuori e in numero di 16, quanti effettivamente ne servono per gestire la grafica 640 × 200. Inoltre si scopre che 'sto benedetto processore video dispone di 37 registri interni, ai quali come per il 16 K di cui sopra, si accede solo in maniera indiretta, ovvero tramite il registro 0 e il registro dati di quest'ultimo.

16 kappa nascosti

Rapido calcolo: la grafica è 640 × 200 pari a un totale di 128 mila punti direttamente indirizzabili. Ovvero è

necessaria una pari quantità di bit per memorizzare una pagina grafica di tali dimensioni. 128 mila bit diviso 8 fa 16000 byte. Per accedere a questa memoria che, non per essere noiosi, non ha nulla a che vedere con i 128 kappa del centoventotto, si usano dei registri dell'8563 dato che questa non è direttamente accessibile con normali operazioni di peek e poke. Il divertimento però inizia quando si rileva che i registri dell'8563 sono anche loro non direttamente indirizzabili eccezion fatta del primo. E a questo punto se qualcuno vuole inventare qualcosa di più contorto è nei guai: non credo che ci riuscirà.

Come dire che per scrivere il numero 3 nel byte 1000 della ram video occorre usare i registri 18 e 19 per parcheggiare il 1000 spaccato in parte alta e parte bassa. Fatto questo accediamo al registro 31 per inserire il numero 3 da scrivere nella locazione 1000. L'accesso ai registri, banalmente avviene accedendo al registro 0 dove mettiamo il numero del registro desiderato e al

registro dati dove immettiamo il valore da pok-are nel registro. Tutte queste parole (e non credo che sia ancora chiaro) per descrivere una cosa che in milioni di computer più puliti si indica con:

POKE 1000,3
 sì, abbiamo fatto proprio questo.

Il registro Read/Write

L'unico oggetto dell'8563 accessibile dal 128 (giusto per sottolineare il fatto che è davvero lontano) è il registro 0 tramite il quale si dovrà fare «tutto». Esso è mappato nella memoria (normale) all'indirizzo esadecimale \$D600 del banco 15. Ad esso è associato anche un registro dati o canale, mappato \$D601, tramite il quale possiamo leggere contenuti di altri registri o immettere un valore da «schiaffare» in uno qualsiasi dei 36 registri del VDC.

Tanto per cominciare potremmo modificare il numero di colonne da visualizzare, minore o uguale a 80. Per fare ciò è sufficiente inserire il numero di colonne nel registro 1 di tale integrato. La sequenza di poke da effettuare è la seguente:

```
POKE DEC («D600»),1:
POKE DEK («D601»),N
```

dove N sono le colonne volute. Si badi bene che le due poke devono essere date insieme, separate da due punti e non prima l'una e dopo l'altra, con due return separati. Provate a scrivere ora qualcosa sul vostro schermo a 80 colonne: noterete che arrivati alla ennesima colonna non sarà possibile proseguire sulla stessa linea perché il cursore andrà a capo.

Le due POKE di cui sopra servono la prima per indicare a quale registro fare riferimento, la seconda per specificare il dato da inserire nel registro. Per leggere il contenuto di un registro si procede in modo analogo: si indica il registro desiderato e leggiamo il suo contenuto in D601. Nella fattispecie, per leggere ad esempio il registro 1, scriveremo:

```
POKE DEC («D600»),1:
PRINT PEEK («D601»)
```

come sempre, sulla stessa linea.

```
10 INPUT "LOCAZIONE ";X
20 HI=X/256:LO=X AND 255
30 INPUT "VALORE ";V
40 POKE 54784,18:POKE54785,HI
50 POKE 54784,19:POKE54785,LO
60 SYS 52682,V
70 END
```

Figura 1 - Programma per scrivere in una cella della video ram.

Accediamo alla Video Ram

Per accedere a una qualunque delle 16384 celle di memoria del VDC 8563, è «sufficiente» ricopiare i listati di figura 1 e 2. Il primo serve per operazioni di scrittura, il secondo in lettura. I registri interessati a tali operazioni sono i 18, 19 e 31. Come sempre per accedere a questi ci serviremo del registro 0 mappato a \$D600, in decimale 54784. Oltre a ciò useremo una routine del sistema operativo, locata in pagina 15 all'indirizzo 52682 che ha il compito di sincronizzare l'invio del dato con il Ready del processore video (traduzione: glielo manda solo quando questo è disposto ad accettarlo). Passiamo al commento dei due listati. La linea 10 tramite l'input accetta nella variabile X la locazione della video ram che intendiamo adoperare. Segue alla linea 20 la suddivisione di X in parte alta (HI) e parte bassa (LO). Nel listato 1, dato che si riferisce ad una operazione di scrittura alla linea 30 è chiesto il valore da pokare. Alle linee 40 e 50 sono spediti nei registri 18 e 19 rispettivamente la parte alta e la parte bassa dell'indirizzo. A questo punto, se si tratta di scrittura (figura 1) sarà sufficiente un SYS 52682,V che invia il dato al chip. Se invece si tratta di lettura (figura 2), non faremo altro che leggere il nostro byte nel registro 31 del VDC. Semplice, no?

La mappa della memoria

Come se non bastassero le 16 mappe già discusse nella prima puntata di

```
10 INPUT "LOCAZIONE ";X
20 HI=X/256:LO=X AND 255
40 POKE 54784,18:POKE54785,HI
50 POKE 54784,19:POKE54785,LO
60 POKE 54784,31:PRINT PEEK (54785)
70 END
```

Figura 2 - Programma per leggere una cella della video ram.

128 da zero, ci accingiamo ora a discutere la 17-esima, di proprietà del signor 8563.

La dimensione di tale memoria, come già detto, assomma a 16 k byte, dovendo memorizzare una pagina grafica di 640x200 pixel. Di tale grafica ad alta risoluzione, avremo modo di riparlare con calma sui prossimi numeri: questo mese tratteremo l'argomento solo dal punto di vista 80 colonne di testo.

Detto questo, la mappa di memoria dell'8563, quando non è usato in modo grafico, si presenta come mostrato in figura 3. Toviamo 2k di mappa video (80x25=2000 caratteri), di seguito a questa altri due k per le informazioni riguardanti il colore e il modo di visualizzazione dei caratteri, 4k di memoria libera e 8k per il generatore dei caratteri.

Per quanto riguarda la pagina video, non c'è molto da dire in quanto il funzionamento è «compatibile» con l'arci frita e rifrita pagina video del modo 40 colonne identico a quello del 64 o del VIC-20. La Commodore «si è inventata» una corrispondenza codice-di-schermo/carattere, lungi dall'essere ascii-like, in cui al carattere 0 corrisponde il primo carattere descritto nel generatore dei caratteri, al secondo il secondo e così via. La distinta dei codici, la trovate sul manuale fornito con la macchina.

I 2k seguenti, servono per immagazzinare, carattere per carattere presente sullo schermo, il colore e il modo di visualizzazione di ognuno di questi.

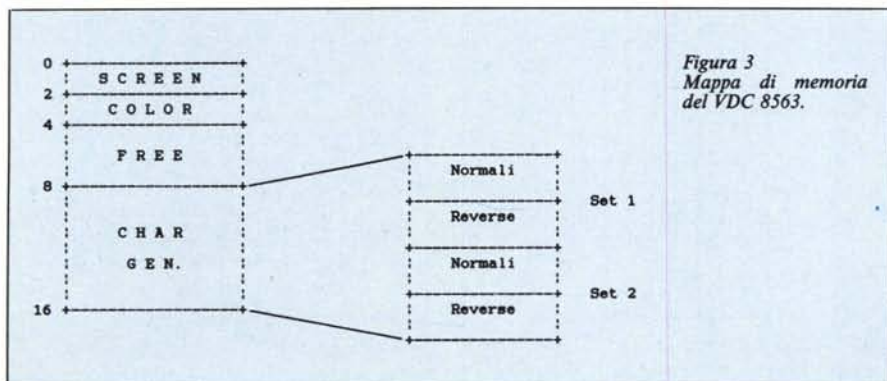


Figura 3
 Mappa di memoria
 del VDC 8563.

```
BIT 0 Luminosità
BIT 1 Blu
BIT 2 Verde
BIT 3 Rosso
BIT 4 Flash
BIT 5 Sottolineato
BIT 6 Reverse
BIT 7 Secondo set
```

Figura 4 - Contenuto di una cella della pagina colore del VDC.



128 da zero

Quindi un tantino in più di una normale pagina colore dei nobili predecessori.

Come mostrato in figura 4, tutti i bit di ogni posizione di schermo sono utilizzati: i primi 4 per il controllo colore e luminosità (in tutto 16 colori), segue il bit di flash che se è settato fa lampeggiare il carattere cui stiamo riferen-

do, i bit 5, 6, 7 servono rispettivamente per la sottolineatura, il modo reverse o il set alternativo maiuscolo/minuscolo. Tutto ciò implica che per ogni carattere visualizzato sullo schermo possiamo selezionare qualsiasi combinazione dei bit di cui sopra, ponendo l'accento sul fatto che sono contemporaneamente visualizzabili sia i caratte-

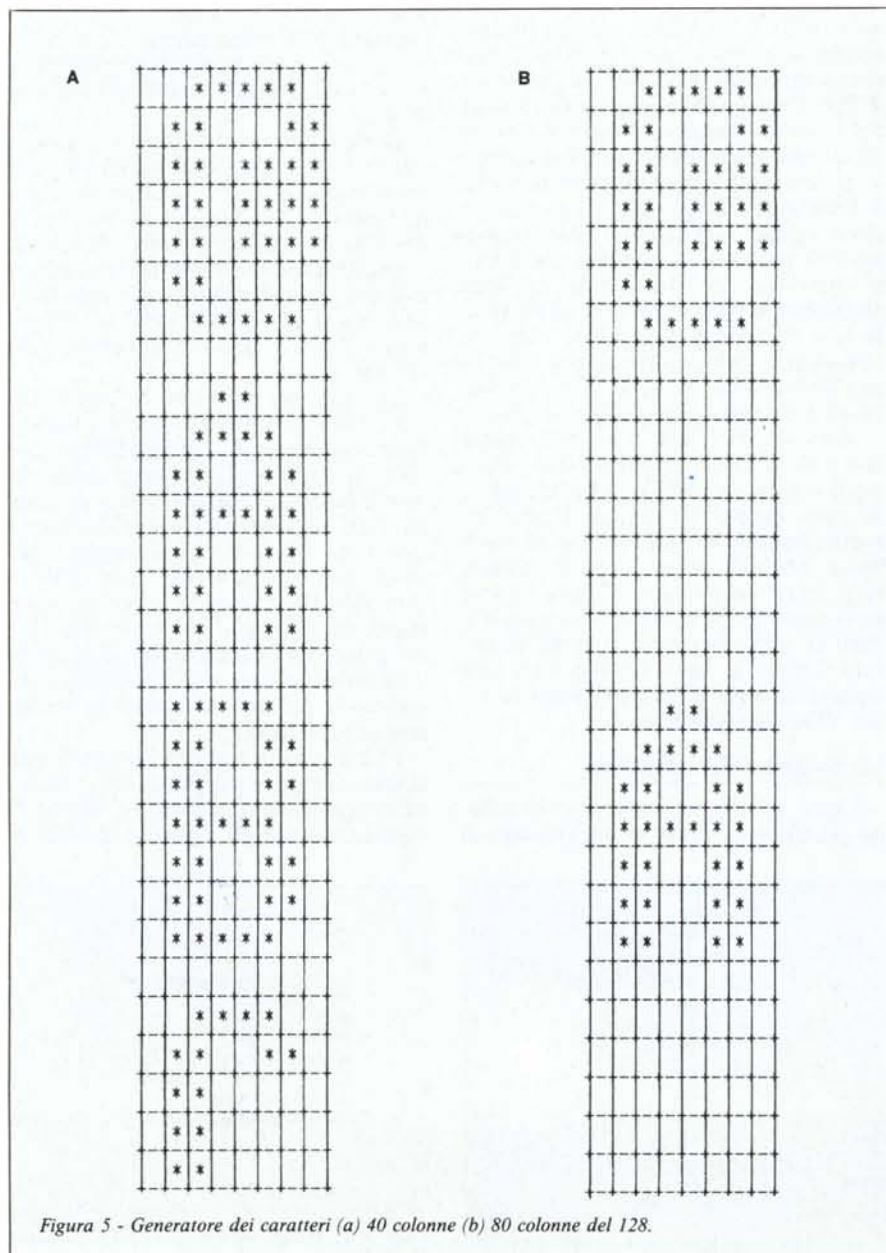


Figura 5 - Generatore dei caratteri (a) 40 colonne (b) 80 colonne del 128.

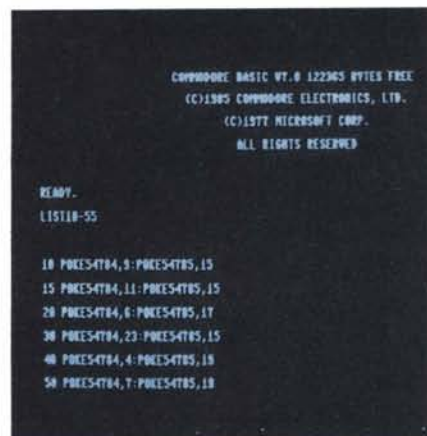


Foto 1



Foto 2

Figura 6

```

1 REM *****
2 REM *
3 REM *   C A R A T T E R I   8 X 16
4 REM *
5 REM *
6 REM *   (C) 1986 ADP SOFTWARE
7 REM *
8 REM *
9 REM *****
10 POKE54784,9:POKE54785,15
15 POKE54784,11:POKE54785,15
20 POKE54784,6:POKE54785,17
30 POKE54784,23:POKE54785,15
40 POKE54784,4:POKE54785,19
50 POKE54784,7:POKE54785,19
60 FAST
70 FORI=0TO511:FORJ=0TO7:X=I*16+J+8192
80 GOSUB200:A%(J)=A:NEXT
90 FORJ=0TO15
100 X=I*16+J:T=INT(J/2)
110 V=A%(T):X=X+8192:GOSUB300
120 NEXT:END
200 HI=X/256:LO=X AND 255
210 POKE 54784,18:POKE54785,HI
220 POKE 54784,19:POKE54785,LO
230 POKE 54784,31:A=PEEK(54785)
240 RETURN
300 HI=X/256:LO=X AND 255
310 POKE 54784,18:POKE54785,HI
320 POKE 54784,19:POKE54785,LO
330 SYS2682.V
340 RETURN

```


In figura 5 è mostrato come sono immagazzinate le varie descrizioni di caratteri in un generatore standard (a esempio quello in rom del processore video 40 colonne) e nel generatore in ram dell'8563. Nel primo caso ogni 8 byte abbiamo un carattere, nel secondo caso, abbiamo un carattere 8×8 e poi 8 byte blank poi un carattere e così via.

Caratteri 8 x 16

Come prima applicazione 8563-ec-
cia proviamo a far saltare fuori i carat-
teri 8×16 dal nostro 128. Il pro-
gramma, mostrato in figura 6, si compone
di due parti: nella prima (linee 5-60) è
impostato il modo caratteri doppia al-
tezza, nella seconda parte, tali carat-
teri sono effettivamente costituiti raddop-
piando i byte che li descrivono. In al-
tre parole, occorre sfruttare tutta la
matrice di punti 8×16 di cui i nuovi
caratteri disporranno. Ciò avviene
semplicemente prendendo gli otto byte
che descrivono un carattere e sostitui-

Nota

I codici di controllo nei listati sono riportati in forma «esplicita», in conseguenza dell'impiego della stampante Star NL-10 e relativa interfaccia per Commodore. Ovviamente, nella digitazione del programma è necessario usare i consueti tasti che corrispondono alle indicazioni fra parentesi: ad esempio cursore destro per (RGHT), CTRL-3 per (RED) eccetera.

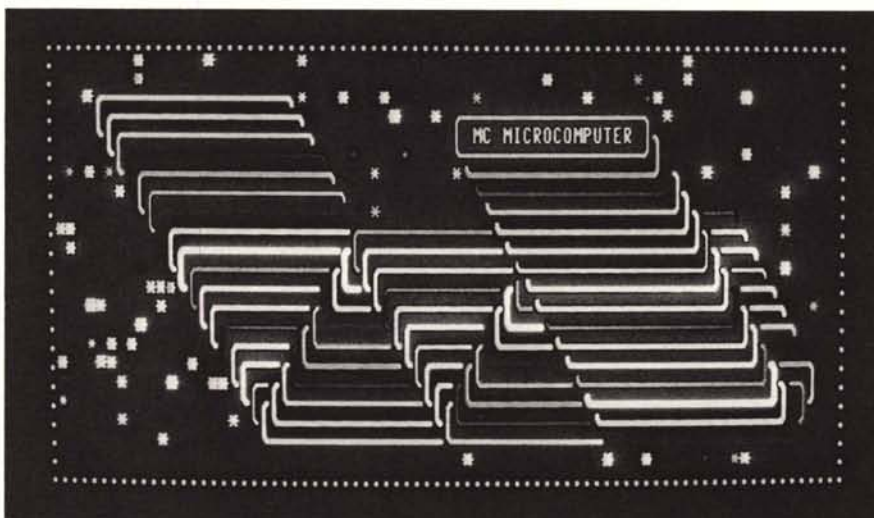


Foto 3

Figura 7

```

2 REM *
3 REM *   D E M O       S I P A R I O   *
4 REM *
5 REM *
6 REM *           (C) 1986 ADP SOFTWARE
7 REM *
8 REM *
9 REM *****
10 PRINT"(CLR)":FAST
15 Z1=31;Z2=11;U1=1;U2=1
20 PRINT"....."
.....
30 FOR I=1TO22
40 PRINT"";
50 FOR J=1TO78
60 PRINT"(RGHT)":NEXT
70 PRINT"";
80 NEXT
90 PRINT"....."
.....(HOME)"
100 AS="(HOME)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(R
)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(R
GHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGH
T)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(R
GHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(R
GHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGH
T)(RGHT)"
110 BS="(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)"
120 FORI=1TO150+RND(1)*50:PRINTLEFTS(AS,RND(1)*77+2);RIGHTS(BS,24-RND(1)*22)
130 K=(K+1)AND15:PRINTMIDS(" (BLU) (WHT) (RED) (CYN) (PUR) (GRN) (BLU) (YEL) (ORNG) (BRN) (
LRED) (GRY1) (GRY2) (LGRN) (LBLU) (GRY3)".K+1,1):NEXT
140 FORI=1TO100:GOSUB4000:NEXT
150 BANK15
160 FORX=2TO80STEP2:GOSUB1000:NEXT:X=1:Y=1:GOSUB3000
170 FORY=2TO80STEP2:GOSUB2000:NEXT
180 FORX=2TO70STEP2
190 Y=X+10:GOSUB3000:NEXT
200 FORX=69TO1STEP-2
210 Y=X+10:GOSUB3000:NEXT
220 FORY=12TO80STEP2:GOSUB2000:NEXT
230 FORX=16TO80STEP2
240 Y=X-15:GOSUB3000:NEXT
250 FORX=78TO16STEP-2
260 Y=X-15:GOSUB3000:NEXT:X=40:Y=41:GOSUB3000
270 FORI=1TO39STEP2:X=40-I:Y=41-I:GOSUB3000:NEXT
280 FORI=1TO20:GOSUB4000:NEXT
290 FORI=38TO0STEP-2:X=40-I:Y=41-I:GOSUB3000:NEXT
300 X=0:Y=0:GOSUB3000
305 FORI=1TO10:GOSUB4000:NEXT
310 FORY=2TO80STEP2:GOSUB2000:NEXT
999 PRINT"(HOME)(HOME)":GOTO 120
1000 POKES4784,34:POKES4785,X+5
1010 H=H+1:IF(H AND 1) THEN RETURN
1020 GOTO4000
2000 POKES4784,35:POKES4785,Y+6
2010 H=H+1:IF(H AND 1) THEN RETURN
2020 GOTO4000
3000 POKES4784,35:POKES4785,Y+6
3010 POKES4784,34:POKES4785,X+5
3020 H=H+1:IF(H AND 1) THEN RETURN
4000 K=(K+1)AND15:PRINTMIDS(" (BLU) (WHT) (RED) (CYN) (PUR) (GRN) (BLU) (YEL) (ORNG) (BRN) (
LRED) (GRY1) (GRY2) (LGRN) (LBLU) (GRY3)".K+1,1);
4010 Z1-Z1+U1:IFZ1>56ORZ1<4:THENU1=-U1
4020 Z2-Z2+U2:IFZ2>18ORZ2<3:THENU2=-U2
4030 IFZ2<13ANDZ2>8 THEN IF RND(0)<.05THENU2=-U2
4100 WINDOWZ1,Z2,Z1+20,Z2+2
4105 PRINT" "
4110 PRINT" MC MICROCOMPUTER "
4120 PRINT" "
4130 RETURN

```




128 da zero

tuendo per ognuno di essi due byte uguali in modo da raddoppiare la dimensione del carattere.

La serie di poke presenti nelle prime linee, servono per inizializzare l'8563 al nuovo modo di funzionamento. Ci serviremo quindi dei registri 4, 6, 7, 9, 11, 23.

Il primo serve per regolare il sincronismo verticale di video in modo che non scrolli continuamente a causa delle modifiche agli altri registri. Col registro 6 posto al valore 17 intendiamo indicare il numero di linee da visualizzare sullo schermo. Nei registri 9 e 23 poniamo il valore 15 che corrisponde al numero di righe - 1 di cui è composto il carattere (nel nostro caso $16 - 1 = 15$). Il registro 7 è usato per centrare lo schermo, infine il registro 11 contiene il numero di righe di cui è composto il cursore.

Al termine di questa inizializzazio-

ne, il nostro processore video visualizzerà l'intera matrice 8×16 per ciascun carattere. Dato che le matrici dei caratteri all'accensione sono come mostrate in figura 5B avremo l'effetto di vedere i caratteri ancora 8×8 concentrati nella parte alta di ogni marice 8×16 (vedi foto 1).

Le linee 70-340 descrivono l'espansione dei caratteri nel modo descritto prima (foto 2). Come facilmente immaginabile è fatto uso di due subroutine (200 e 300) che implementano le operazioni di lettura e scrittura nella video ram come mostrato in figure 1 e 2. Dando run al programma, l'espasante lentezza che avrete modo di notare (e odiare) è tutta dovuta alla macchinosa delle operazioni necessarie per arrivare a scrivere qualcosa lì dentro. Come alla Commodore gli sia venuto di inventare una cosa simile, resterà un mistero. Contenti loro...

Cala il sipario

Prima di chiudere questo primo appuntamento con l'8563, vedremo come sia possibile implementare una sorta di sipario sullo schermo video a 80 colonne. I registri interessati sono il 34 e il 35 denominati per l'appunto «inizio rappresentazione schermo» e «fine rappresentazione schermo». Ciò vuol dire che se vogliamo rappresentare da colonna 10 a colonna 60 basterà inviare il numero 10 al registro 34 e il numero 60 al registro 35. Inoltre, è anche ammesso che il valore nel registro 35 sia inferiore a quello nel registro 34. In questo modo è possibile creare vuoti centrali invece che laterali. Un rapido susseguirsi di cambiamenti in questi due registri può creare effetti assai invitanti come quelli provocati dal programma di figura 7 e mostrati nella foto 3. Buon divertimento.

Ci sono
i compatibili.

E ci sono i

PCbit



A Napoli
Vi aspettano da



GENERAL
COMPUTERS

PCbit: totalmente compatibile PC/XT IBM PCbit at: totalmente compatibile PC/AT IBM

Fino al 31/12/86 - Sconto del 20% su tutti i prodotti

RIVENDITORE AUTORIZZATO: PC BIT - ESA COMPUTER / RIVENDITORE OLIVETTI

IBM, PC/XT e PC/AT sono marchi registrati della International Business Machines Napoli, calata S. Marco 13 - tel. 081/310114-5510114 ≈ affiliato PCbit computers