

Eliminazione dei bordi sul 64

di Massimo Morlupi - Frascati (RM)

L'argomento presentato nell'articolo riguarda la possibilità, sul Commodore 64, di eliminare il bordo superiore ed inferiore dello schermo, onde poter visualizzare degli sprite in tali zone normalmente inutilizzate.

L'articolo è diviso principalmente in due parti: nella prima viene illustrato il funzionamento del raster-register e quindi il metodo che permette un corretto utilizzo delle interruzioni video; nella seconda parte, sulle basi della prima, viene descritto il metodo tramite il quale può essere eliminato il bordo, e di conseguenza tutte le notizie che permettono un miglior utilizzo della suddetta tecnica.

Un po' di raster

Chi ritiene di conoscere il funzionamento del raster-register e saper utilizzare di conseguenza le interruzioni video, può saltare questa parte preliminare. In caso contrario consiglio di leggere quanto segue affinché risulti di più facile comprensione la parte successiva

È disponibile, presso la redazione, il disco con i programmi pubblicati in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 265.

nella quale verrà affrontato il tema principale dell'articolo.

Il VIC-II è l'interfaccia che nel 64 si occupa della gestione dell'output video. La particolarità di tale chip è quella di offrire, accanto a prestazioni direttamente constatabili (modalità grafiche, modalità testo, possibilità di visualizzazione di sprite), anche una serie di caratteristiche che permettono una gestione più complessa delle stesse.

Tali possibilità non sono direttamente sfruttabili dall'utente, in quanto spesso richiedono tecniche di programmazione realizzabili solo tramite il linguaggio Assembler.

Una peculiarità del VIC-II, di fondamentale importanza, è quella di permettere il controllo della posizione del fascio elettronico sullo schermo.

L'immagine visualizzata sul video viene infatti composta da un fascio di elettroni, il quale partendo dalla posizione in alto a sinistra dello schermo disegna ogni linea di esso compiendo un percorso da sinistra verso destra (in riferimento a chi osserva).

Ogni qualvolta una linea è terminata, il fascio viene riportato all'inizio della successiva. Giunto alla fine dell'ultima linea il fascio di elettroni viene riportato nella posizione iniziale (vedi fig. 1a).

Tale processo viene ripetuto (su video PRL) 50 volte al secondo.

Vediamo ora [fig. 1a-1b] come sia possibile controllare la posizione del fascio elettronico sullo schermo.

I registri di comunicazione del microprocessore 6502 con l'interfaccia VIC-II sono mappati nella porzione di RAM che si estende dall'indirizzo \$d000 (53248), all'indirizzo \$d02e (53294). Ivi

troviamo tutte le locazioni di memoria dedicate al controllo degli sprite e attributi (priorità, colore, modalità, ecc.).

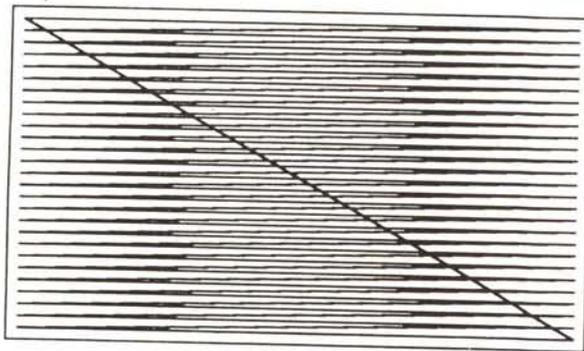
La locazione \$d012 (53266) contiene il numero di linea corrente disegnata dal fascio elettronico. Essendo una qualsiasi immagine composta da 276 linee, avremo bisogno di 9 bit perché queste possano venir effettivamente numerate tutte. A questo scopo oltre al registro \$d012 viene utilizzato anche il bit 7 del registro \$d011 (53265) considerato come bit più significativo (vedi fig. 2).

Scrivendo un valore di linea (sempre nel formato a 9 bit) all'interno dei due registri, tale valore verrà salvato automaticamente in un registro di comparazione di quadro, mentre i registri \$d011 e \$d012 continueranno a riportare l'attuale posizione del fascio elettronico. Non appena però verrà disegnata la linea specificata nel registro di comparazione (precedentemente impostato) il VIC genererà un IRQ (interruzione mascherabile), controllabile dall'utente a patto che siano state rispettate alcune condizioni.

La locazione \$d01a (53274) rappresenta il registro di abilitazione delle IRQ del circuito video (VIC); il registro di controllo di tali interruzioni è posto alla locazione \$d019 (53273) (vedi tabella di fig. 3).

Affinché il VIC-II generi realmente l'interruzione desiderata, dovremo impostare ad 1 il bit 7 e il bit 0 del registro \$d01a. Il primo indicherà al VIC che vogliamo avvalerci di una qualsiasi delle interruzioni disponibili, mentre il secondo specificherà il tipo di interruzione attivata (interruzione di comparazione di quadro nel nostro caso).

In tal modo modificando opportuna-



Movimento del fascio elettronico sullo schermo

Figura 1a

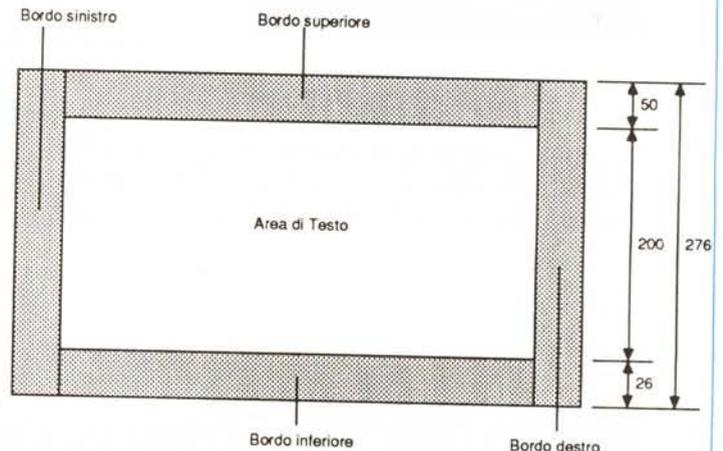


Figura 1b ▶

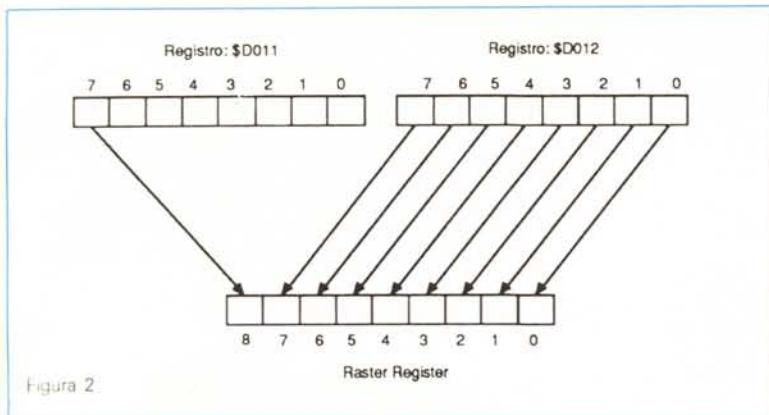


Figura 2

mente il vettore di IRQ (posto alle locazioni \$0314-\$0315) in modo tale che punti ad una nostra routine di controllo in LM potremo gestire a nostro piacimento le interruzioni generate dal VIC.

Naturalmente bisognerà controllare che l'interruzione sia provenuta proprio dal circuito video e non da altri dispositivi, testando il bit 7 del registro \$d019 (che nel caso di interruzione video verrà impostato a 1). Le routine di interrupt da noi create dovranno terminare con un salto incondizionato alla locazione \$fbc (jmp \$fbc).

Gli sprite sul bordo ...

Veniamo finalmente al nostro problema: come possono essere sovrapposti gli sprite al bordo?

Come molti avranno già avuto modo di notare, uno sprite ha, rispetto al bordo, una priorità più bassa e di conseguenza tende ad essere nascosto da esso. Per tale motivo era logico pensare che non fosse in alcun modo possibile sovrapporre scritte di nessun genere al bordo. Tale teoria è stata però smentita all'uscita dei primi giorni che presentavano la particolarità di far apparire punteggi ed altro proprio dove normalmente è presente il bordo.

Preso dalla curiosità ho cercato di comprendere quali registri del VIC fossero implicati in tale «magia». Dopo circa un mese di inutili tentativi ho deciso di lasciar stare manuali e mappe di memoria nel tentativo di capire il funzionamento, almeno a grandi linee, del VIC.

Lo schermo è composto da 276 linee; 200 di queste rappresentano l'area di visualizzazione vera e propria, mentre le restanti vanno a comporre il bordo inferiore e superiore (fig. 1b). Il diagramma di flusso di figura 4 rappresenta il modello di funzionamento del VIC-II, da me ottenuto.

Il testo contrassegnato con la lettera 'A' è quello che, se verificato, dà inizio alla visualizzazione del bordo; il test successivo (contrassegnato con la lettera 'B') è quello che, terminato il bordo, dà inizio ad una nuova visualizzazione dell'area di testo.

Se fosse possibile far «saltare» il controllo 'A', ovvero fare in modo che non risulti mai verificata la condizione in esso specificata, otterremmo che il VIC, ignaro delle nostre macchinazioni, continuerebbe a disegnare area di testo anche dove normalmente è presente il bordo.

Far saltare il suddetto controllo è tut-

Registro \$d019 (indicatore IRQ)	Registro \$d01a (abilitatore IRQ)
Bit 7: Indicatore: si è verificata una delle seguenti IRQ	Abilita una qualunque condizione di IRQ generabile dal VIC
Bit 6: Non usato	Non usato
Bit 5: Non usato	Non usato
Bit 4: Non usato	Non usato
Bit 3: Indicatore IRQ di penna ottica	Abilitatore IRQ di penna ottica
Bit 2: Indicatore IRQ di contatto fra due animazioni	Abilitatore IRQ di contatto fra due animazioni
Bit 1: Indicatore IRQ di contatto animazione-fondo	Abilitatore IRQ di contatto animazione-fondo
Bit 0: Indicatore IRQ di comparazione di quadro	Abilitatore IRQ di comparazione di quadro

Figura 3

t'altro che difficile: il bit 3 del registro \$d011, indica il numero di righe di testo; impostando a 0 tale bit avremo un restringimento dell'area di testo da 25 a 24 righe. Più precisamente il bordo superiore si sarà abbassato di 4 linee, mentre quello inferiore si sarà alzato di 4 linee (per un totale di 8 linee = una riga di testo). Di conseguenza il bordo inferiore avrà ora inizio alla linea 246 anziché alla linea 250.

Supponiamo di impostare la modalità a 24 righe mentre il VIC sta disegnando una qualsiasi delle linee comprese fra la 246-esima e la 250-esima.

Immaginate cosa può succedere? Se non lo avete ancora capito ve lo dico io: succede che il test 'A' non risulta mai verificato, in quanto esso sarà indirizzato ad una linea (la 246-esima) che il VIC ha già disegnato in precedenza.

Infatti quando il VIC aveva disegnato la 246-esima linea era impostata la modalità a 25 righe testo e il bordo avrebbe dovuto iniziare alla 250-esima. Quando poi il VIC giunge a disegnare la 250-esima linea è stata già impostata la modalità a 24 righe testo e il bordo, secondo tale indicazione, avrebbe dovuto iniziare alla 246-esima.

Risultato: impostando alternativa-

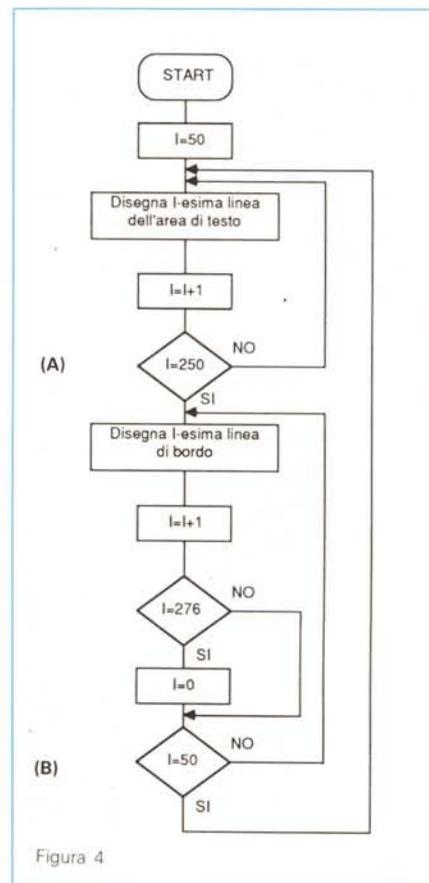


Figura 4

```

Programma 1
5 POKE$3281,6:POKE$3280,14
7 PRINT"(CLR)(WHT)"
10 FOR I=8192 TO 8292
20 READ A:POKE I,A:C=C+A
30 NEXT
35 IF C<>12093 THEN PRINT"ERRORE NEI DATA !!!":END
40 SYS 8192
50 PRINT"BORDO DISATTIVATO !!!"
100 DATA120,169,46,141,20,3,169,32,141,21,3,169,0,141,255,63
105 DATA173,26,208,9,129,141,26,208,173,17,208,41,127,141,17,208
110 DATA169,247,141,18,208,169,180,205,18,208,208,251,88,96,173,25
115 DATA208,141,25,208,48,7,173,13,220,88,76,49,234,230,2,165
120 DATA2,41,1,208,16,169,248,141,18,208,169,8,13,17,208,141
125 DATA17,208,76,188,254,169,0,141,18,208,169,247,45,17,208,141
130 DATA17,208,76,188,254

```

Figura 5

mente la modalità a 24 e 25 righe di testo (utilizzando le interruzioni raster) il bordo non verrà più disegnato. Ma cosa disegna il VIC in alternativa ad esso? Semplice (ma badate bene questo l'ho scoperto solo per caso), la zona destinata al bordo viene riempita con il dato costante contenuto nell'ultima locazione di memoria del banco «visto» dal VIC:

Ultima locazione del banco		
Banco		
1	\$3fff (16383)	(banco di default)
2	\$7fff (32767)	
3	\$bfff (49151)	
4	\$ffff (65535)	

Per quanto riguarda il colore, la zona suddetta assume la stessa tonalità del fondo.

Se proviamo ora a posizionare uno o più sprite, ad esempio nella zona ove prima avevamo il bordo inferiore, esso verrà correttamente visualizzato. L'unico inconveniente è che la parte di sprite sottostante la 255-esima linea di schermo verrà ugualmente visualizzata a partire dalla prima linea dello stesso. In altre parole ciò che verrà visualizzato

nelle ultime 21 linee dello schermo verrà anche visualizzato nelle prime 21 linee di esso.

Per ovviare all'inconveniente bisognerà ad esempio generare un'interruzione raster alla linea 0, dove magari gli sprite verranno disabilitati (o spostati a seconda della necessità) per poi essere ristabiliti all'interruzione successiva.

Più facile di così ...

... un ultimo avvertimento: la tecnica citata non consente di eliminare il bordo laterale. Per la maggior parte delle applicazioni comunque l'assenza del bordo inferiore e superiore si rivela più che sufficiente.

Alcuni esempi

Nelle figure 5 e 6 vengono riportati due programmi dimostrativi, scritti in linguaggio Assembler. Il primo si limita ad eliminare il bordo. Il secondo invece, oltre ad eliminare il bordo, vi posiziona anche 16 sprite pieni, distribuiti 8 nella zona del bordo inferiore e 8 nella zona del bordo superiore.

Entrambi i programmi utilizzano due interruzioni raster. La prima viene attivata alla linea 248 e qui viene impostata la

modalità a 24 righe di testo; la seconda viene attivata alla linea 0 dove viene impostata la modalità a 25 righe di testo. Il secondo programma prevede lo spostamento delle coordinate di 8 sprite ogni volta che una delle due interruzioni si è verificata.

Buon divertimento!!

Portfolio 64

di Andy Cavallini - Guanzate (CO)

Sappiamo tutti cos'è Portfolio, l'argomento del programma? È un concorso indetto da «La Repubblica» e da altri quotidiani in cui i possessori delle tessere devono calcolare la somma delle variazioni in Borsa degli 8 titoli per ogni tessera e confrontarla con un dividendo giornaliero: in caso di congruenza si vincono dei milioni.

Questi calcoli sono semplici e rapidi per 2 o 3 tessere, ma stressanti se le tessere sono 20 o 30.

Esistono già dei PRG per questo scopo, che io non ho mai visto, comunque ho deciso di farne uno ad hoc per me, ma che mi piacerebbe fosse usato da più persone.

Inserimento

Per prima cosa il PRG legge l'indice, cioè il numero di tessere memorizzate, che permette al 64 di fare molte cose, la più importante delle quali è la gestione dei file.

Ho scelto liberamente di usare sempre e solo file sequenziali per mantenere una portabilità del PRG tra varie macchine, dato che tutti i computer (che io sappia) supportano questo tipo di file.

Saputo questo, il fido 64 dimensiona vettori e matrici, dopodiché passa all'input da tastiera.

Vengono richieste le 8 società per ogni tessera che si vuole memorizzare, il suo proprietario con relativo numero di telefono e infine, se si vogliono apportare correzioni. Detto fatto, ogni tessera viene salvata in un piccolo file seq di 10 elementi, il cui nome sarà TESSN x.

Viene letto il file society, che contiene in ordine numerico tutte le società di tutte le tessere.

Saranno aggiunte in coda al vettore derivante, le ultime società appena immesse.

Il vettore ottenuto sarà riordinato con un modesto quanto efficiente BUBBLE SORT e risalvato. Anche l'indice viene aggiornato.

```

10 REM *****
20 REM ***** PROGRAMMA 2 *****
30 REM *****
40 REM
50 POKE$3280,0:PRINT"(WHT)(CLR)"
60 FOR I=8192 TO 8384
70 READ A:POKE I,A:C=C+A
80 NEXT
90 IF C<>22559 THEN PRINT"ERRORE NEI DATA !!!":END
100 PRINT"OK":SYS8192:END
110 DATA120,169,108,141,20,3,169,32,141,21,3,169,0,141,255,63
120 DATA141,28,208,141,23,208,141,29,208,169,14,141,33,208,169,3
130 DATA141,16,208,162,7,169,13,157,248,7,189,185,32,157,39,208
140 DATA202,16,242,169,255,141,21,208,162,63,157,64,3,202,16,250
150 DATA169,33,162,14,157,0,208,24,105,39,202,302,16,246,173,26
160 DATA208,9,129,141,26,208,173,17,208,41,127,141,17,208,169,247
170 DATA141,18,208,169,180,205,18,208,208,251,88,96,173,25,208,141
180 DATA25,208,48,7,173,13,220,88,76,49,234,230,2,165,2,41
190 DATA1,208,27,169,248,141,18,208,169,8,13,17,208,141,17,208
200 DATA162,15,169,27,157,0,208,202,202,16,249,76,188,254,169,0
210 DATA141,18,208,169,247,45,17,208,141,17,208,162,15,169,8,157
220 DATA0,208,202,202,16,249,76,188,254,1,13,2,0,11,6,7
230 DATA8

```

Figura 6

ZORRO BIG BLUE



IL POTENTE SISTEMA MULTIMODULARE PER AMIGA 500 E 1000 CHE VI OFFRE:

- 3 SLOT A 100 PIN ZORRO 2 AMIGA 2000 COMPATIBILE
- 3 SLOT IBM XT COMPATIBILE
- 3 SLOT IBM AT COMPATIBILE
- 2 POSTI PER 2 DRIVE DA 3,5"
- 1 POSTO PER 1 DRIVE DA 5 1/4"
- 1 POSTO PER HARD DISK
- ALIMENTATORE SWITCHING

...SE HAI L'AMIGA NON LASCIARLA DA SOLA...

GLI HARD DISK

AMEGADRIVE SCSI CONTROLLER

Hard Disk controller per A500/1000 in standard SCSI con orologio e batteria tampone. L.240000

AMEGADRIVE

HardDisk da 20.40.80 Mb 3,5" SCSI per controller SCSI. Viene fornito in contenitore per A500/1000 o su scheda per A2000. Prezzi a partire da L.790000

IMPACT A2000 GVP

HD controller SCSI piu' Espansione RAM da 1 o 2Mb per A2000 con autoboot. 0 Kb 1Mb L.490000 Per altre configurazioni CHIEDERE

IMPACT A500 20 HD GVP

come sopra ma per A500 piu' Hard Disk da 20 Mb. L.1340000 per altre configurazioni CHIEDERE

A2090 Commodore

HD controller piu' Harddisk da 20 Mb per A2000. L.1090000

HD2000card

Controller ed HardDisk da 30 Mb su scheda per IBM/XT o A2000 con Janus. L.790000

JANUS XT

Emulatore IBM/XT per A2000. L.949000
JANUS XT + HardDisk da 30Mb HD2000card L.1490000

LE ESPANSIONI DI MEMORIA

AMEGABOARD

Espansione di memoria per A500/1000 da 2 a 8Mb Ester. n. Autocronifg 9Kb L.290000

RAMINT 1 4

Espansione di memoria da 1 a 4Mb per A1000. Interna. Con orologio tampone. 0 Kb L.190000

AMINTERAM

Espansione di memoria per A500 da 512 Kb. Con orologio tampone. 0 Kb L.59000

PRORAM 8Mb PROGRESSIVE PER SYS

Espansione da 0-2-4-6-8Mb sulla stessa scheda per A2000. 0 Kb L.460000

A2052 Commodore

Espansione da 2 Mb per A2000. CHIEDERE

KICKROM

Kickstart su Eprom senza saldature per A1000 con orologio tampone. L.148000

ATTENZIONE!! ABBIAMO BUONE SCORTE DI RAM DA 256 Kb E 1Mb I PREZZI DELLE SOPRADESCRITTE SCHEDE GIA' ESPANSE VERRANNO FATTI AL MOMENTO DELL'ORDINAZIONE

I DRIVE

ADRIVE

Drive da 3,5" esterno per A500/1000/2000 con passante. L.270000

ADRIVE TOWER

Come sopra ma tipo torre nello stesso contenitore. L.590000

ADRIVE2000

Drive interno da 3,5" per A2000. L.218000

ACCELERATORI-PROCESSORI-COPROCESSORI

ATTENZIONE!!! I PREZZI SOTTOINDICATI COMPREDONO LE SCHEDE ACCELERATRICI SENZA PROCESSORI E COPROCESSORI CHE SONO INDICATI A PARTE. LA SCHEDE BANG PUO' MONTARE LE COPPIE 68010-68881/68882 O 88020/88030-68881/68882 MENTRE LA HURRICANE 88020/88030-68881/68882 QUESTO PER LASCIARE IL MASSIMO GRADO DI SCELTA ALL'UTENTE

HURRICANE

Scheda acceleratrice per A1000/A2000. Hurricane A1000 L.599000- Hurricane A2000 L.899000.

BANG

Scheda acceleratrice per A1000 e ATARI ST. BANG A1000 L.340000. BANG ST CHIEDERE

HURRICANE MEMORY 1 4Mb

Espansione di memoria a 52 bit per Hurricane. Hurricane Mem 1Mb L.990000

BANG MEM

Espansione di memoria con Ram statiche a 32 bit per BANG da 128 a 512 Kb BANGMEM 128Kb L.240000

ADAPTER 030

Adattatore per 68030 per Hurricane e Bang. L.390000

PROCESSORI: 68010-L.490000 68020-L.340000 68030- L.890000
COPROCESSORI: 68881 12MHz L.270000 16MHz L.340000
25MHz L.840000 68882 16MHz L.570000 25MHz L.1100000

I DIGITALIZZATORI AUDIO VIDEO

DIGIBOARD

Digitalizzatore audio stereo piu' interfaccia MIDI per A500/1000/2000. L.99000

LIVE! ASQUARED

Digitalizzatore a colori video in tempo reale con effetti video per A500/1000 o A2000 (su scheda). Live500 L.549000-LIVE1000 L.440000-LIVE2000 L.630000

PROGEN PROGRESSIVE PER SYS.

Genlock in standard RG 170A per A500/1000/2000. L.629000

FLICKER FIXER MICROWAY

Scheda da inserire nello slot video dell'A2000 ed elimina il flicker. L.890000

ZORRO BIG BLUE

UNITA' CENTRALE

Chassis metallico per A500/1000 per contenere tutte le periferiche dello Zorro Big Blue. L.120000

ZORRO BIG BLUE BUS

Scheda contenente 3 slot 100 pin A2000 piu' 3 slot XT e 3 slot AT compatibili. L.150000

ALIMENTATORE SWITCHING

Alimentatore da 40W e necessario solo in caso di montaggio di hard disk o piu' schede. L.99000

MODULO DRIVE

1 o 2 drive da 3,5"-680Kb. L.219000

MODULO MIDI+ DIGI STEREO

scheda contenente digitalizzatore stereo piu' interfaccia MIDI. L.79000

ATTENZIONE!!! NELLO ZORRO BIG BLUE SI POSSONO MONTARE TUTTE LE SCHEDE PER L'AMIGA 2000 (JANUS XT/AT, PROGRAM, IM, PACT, A2052, A2090, ecc.)

I MONITORS

COMMODORE 1084

Monitor HiRes per A500, A1000, A2000. L.540000

COMMODORE 2080

Monitor HiRes ad alta persistenza per A500, 1000, 2000. L.620000

PHILIPS 8833

Monitor HiRes Stereo per tutti gli Amiga. L.480000

MULTISYNC

Monitor multisync per A2000 con flicker fixer o IBM. L.960000

STAMPANTI

STAR LC 10

Stampante 120cps NLQ. L.440000

STAR LC 10 color

Come sopra ma a colori. L.529000

EPSON LQ 500

Stampante a 24 aghi 150cps NLQ. CHIEDERE

HP LASERJET IIPUS

Stampante laser. CHIEDERE

I COMPUTERS

SONO IN CORSO DELLE OFFERTE PROMOZIONALI PER I COMPUTER AMIGA 500E 2000. CHIEDERE

SONO INOLTRE DISPONIBILI TUTTI I COMPUTER E LE PERIFERICHE AMSTRAD. CHIEDERE

DISPONIBILI ANCHE TUTTI I COMPATIBILI XT, AT E PS2. CHIEDERE

SOFTWARE

POSSIAMO PROCURARE TUTTO IL SOFTWARE ORIGINALE PUBBLICATO IN AMERICA. MOLTIPLICARE IL PREZZO IN DOLLARI PER IL CAMBIO UFFICIALE IN LIRE, AGGIUNGERE L'IVA E IL 6% DI SPESE DI SPEDIZIONE E VARIE.

PER INFORMAZIONI E/O ORDINAZIONI

COMPUTER CENTER

VIA FORZE ARMATE 260 20152 MILANO

TELEFONO 02-4890213

RICHIEDETE IL NUOVO E GRATUITO CATALOGO CON LE SCHEDE

TECNICHE DI TUTTI I PRODOTTI

VENDITA SOLO PER CORRISPONDENZA

TUTTI I PREZZI SONO IVA COMPRESA



1981/1/12

21-25 OTTOBRE 1988

FIERA DI TRIESTE

QUINTA
RASSEGNA
DELLA RICERCA
E DELLA
TECNOLOGIA

RICERCA, ARTE
DIDATTICA
INFORMATICA

PIAZZALE DE GASPERI, 1
TRIESTE - TEL. (040) 39.29.61/2/3
CAS. POST. 1509 - TELEX 460440 FAIRTS I