MAGGIO 1984 NUMERO 30 LIRE 3500

MC microcomputer®

HARDWARE & SOFTWARE
DEI SISTEMI PERSONALI

VIC-20: la Super Expander
64: caratteri utente & sprite
SPECTRUM: i caratteri grafici
Trasmissione RTTY con il **64**

PROVE

Guida computer: tutti i prezzi

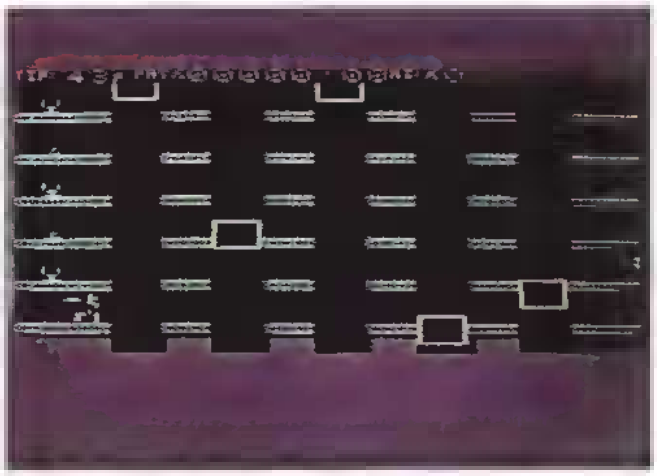
MC MICROCOMPUTER ANNO IV N. 30 - MAGGIO 1984 SPED. ABB. POST. GRUPPO III - 70% - MENSILE

IMAGINE

Wacky Waiters

VIC-20

È un gioco che non presenta notevoli difficoltà e risulta adatto anche ai più piccini. La grafica e la velocità non sono sorprendenti ed il formato dello schermo sfruttato non è il massimo ottenibile con il VIC-20 che può offrire più ampie possibilità. Può essere giocato o utilizzando il joystick o usando la tastiera del computer. Il gioco è presentato su una cassetta di discreta qualità contenente un cartoncino sul quale sono ampiamente illustrate le istruzioni per l'uso. Il modo di caricamento è quello usuale dei giochi della Imagine Software. Connesso il registratore e premuto il tasto play dopo aver dato il comando LOAD, viene caricato un programma "caricatore" che dopo il RUN penserà a trasportare nella memoria del computer il contenuto della cassetta. Durante il caricamento vedrete sostituirsi alle scritte sullo schermo degli strani caratteri, ma ciò è del tutto normale. Diamo uno sguardo all'andamento del gioco. Esso narra le vicende di un cameriere che presta servizio presso un albergo. Egli non è molto pratico del mestiere poiché è la prima volta che svolge un tale lavoro ed è quindi strettamente sorvegliato dal direttore per evitare che compia danni.



Dopo il RUN compare sullo schermo, che come abbiamo detto non è sfruttato in pieno nelle sue possibilità, lo spaccato di un albergo formato da sei piani, compreso il pianterreno, e da cinque scale. Per passare da una scala all'altra bisogna necessariamente prendere uno degli ascensori che fanno su e giù per lo schermo come bolle di sapone. Il nostro amico novello cameriere, si trova all'inizio nella hall al pianterreno (in basso a sinistra) ed è pronto a ricevere le chiamate. Ecco che dall'altra parte dell'albergo compare un cliente che chiama con il braccio ed allora il baldo lavoratore salta da un ascensore all'altro per andare a prendere l'ordinazione, recandosi al piano da cui è stato chiamato per poi ritornare indietro. Egli deve stare inoltre molto attento perché può saltare sull'ascensore nell'istante sbagliato mancandolo e cadendo giù. Si riprende subito e riparte, ma paga di tasca sua la roba che rovescia. I suoi viaggi devono svolgersi rapidamente perché il pagamento avviene per ogni vassoio consegnato. Se perde tempo a passeggiare negli ascensori riceve dei richiami scritti sullo schermo. Come abbiamo detto, si può giocare sia da tastiera che da joystick ed è forse più comodo giocare nel primo modo perché i movimenti avvengono solo in orizzontale e possono essere effettuati utilizzando due tasti (per esempio i tasti SHIFT, X, V o N per effettuare i movimenti verso sinistra ed i tasti Z, C, B o M per i movimenti verso destra). Precisiamo che ad ogni pressione del tasto l'omino raffigurante il cameriere assume una diversa posizione ed è pronto ad imboccare l'ascensore solo in una di queste (quella che lo rappresenta con la gamba tesa). Come al solito sullo schermo, nella parte alta, compare la linea di stato indicante il punteggio della partita in corso e quello più alto ottenuto nelle altre partite. Nella stessa linea, dopo lo start, viene indicato un numero che, partendo da 99 decresce nel tempo e rappresenta la cifra che si riceverà ad ordinazione assolta: minor tempo si impiega, quindi, maggiore sarà la paga.

Questo gioco è dotato di rumori di fondo che lo rendono più gradevole e psicologicamente più efficace: essi non sono però molto ricchi. Si carica sul VIC 20 in qualunque configurazione di memoria.

T P

Produttore:
Imagine
Maxons Buildings
Exchange Street East
Liverpool L2 3 PN
Prezzo:
5 50 sterline

IMAGINE

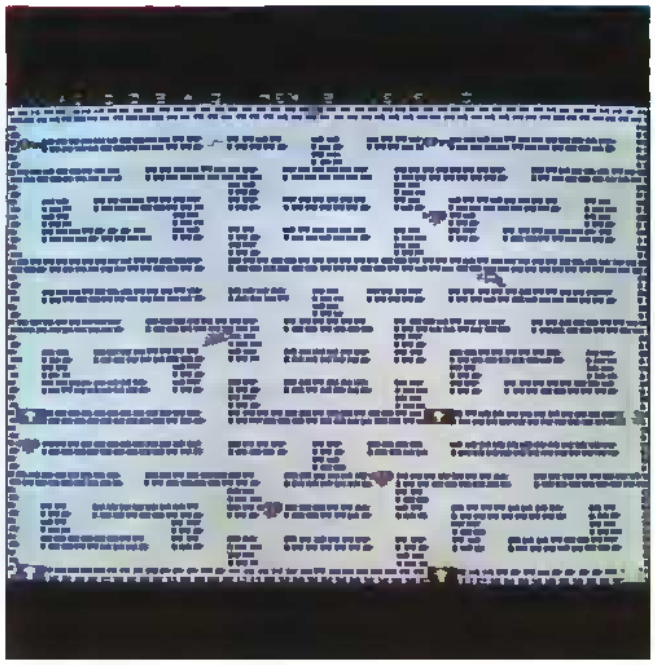
Bewitched

VIC-20

Un nuovo gioco si aggiunge a quelli offerti dalla Imagine Software per il VIC 20 Se esso è nuovo nel nome, non si può dire che lo sia altrettanto come concezione Infatti non bisogna essere molto esperti per accorgersi che appartiene alla serie dei giochi con fantasmi e labirinti in voga già da parecchio tempo Il gioco è comunque simpatico e non dovrebbe annoiare troppo presto Esaminiamone per sommi capi lo svolgimento

Siamo stati catturati da un malvagio mago e gettati nei più profondi meandri di un labirinto sotterraneo dove siamo costretti a vagare per l'eternità Ma con un po' di abilità potremo uscire riguadagnando la libertà perduta

Il disegno del labirinto appare naturalmente sullo schermo Esso si può osservare visto dall'alto (in pianta) ed è suddiviso in più strati separati da quattro porte chiuse, di quattro differenti colori, di cui vengono disegnate le toppe All'ingresso del labirinto si trovano quattro chiavi, ognuna con un colore corrispondente a quello di una porta Il gioco consiste nel prendere una di queste chiavi e dirigersi con essa verso la porta di colore corrispondente, sempre inseguiti da nugoli di fantasmi, aprendola per passare al settore successivo del labirinto Giunti in prossimità della porta non dobbiamo subito cantare vittoria poiché al di là di questa può esserci un'amara sorpresa Infatti, una volta aperta, al posto del varco comunicante con lo strato successivo del labirinto, possiamo trovare un muro che impedisce il passaggio Dobbiamo allora ripercorrere i nostri passi portandoci alla sommità del labirinto, sempre sotto scorta dei fantasmi pronti a mangiarci, e cambiare il colore della nostra chiave (passando con essa attraverso la chiave del colore desiderato) andando poi ad attaccare un' altra porta colorata



Man mano che si corre verso il basso lo schermo scorre verso l'alto rivelando i vari strati del sotterraneo, mentre i fantasmi si fanno sempre più fastidiosi (ma non eccessivamente) cercando di ostacolare in tutti i modi i nostri progressi La via della libertà è molto lunga e se è molto emozionante per l'audace prigioniero del mago forse lo è un po' meno per noi che siamo costretti a percorrere lunghi cammini all'indietro se abbiamo sbagliato chiave e dobbiamo andare a sostituire quella in nostro possesso

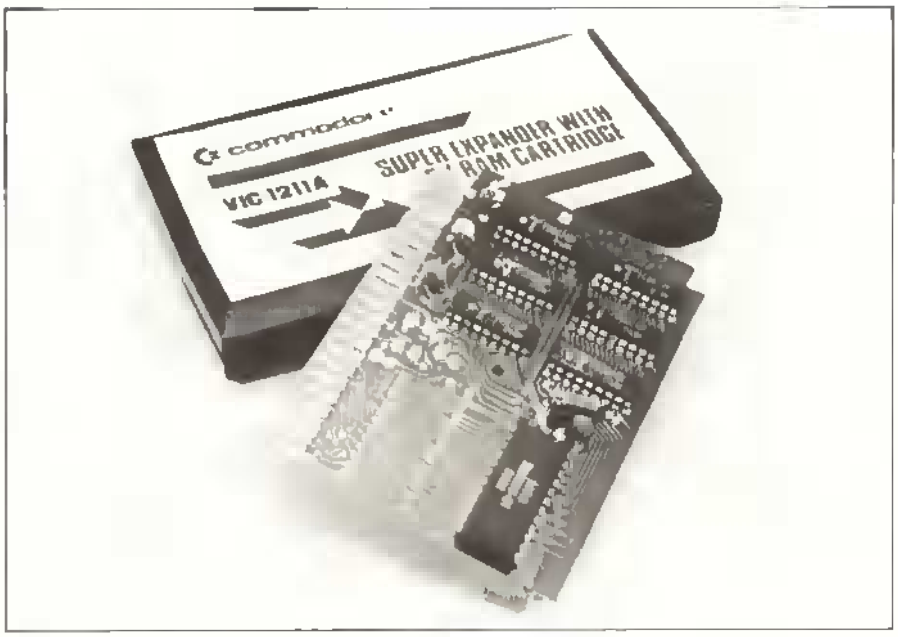
La grafica è di media fantasia e la velocità non è sconvolgente I suoni sono quelli consueti, anch'essi non eccessivamente fantasiosi Le chiavi possono essere distinte solo in base al loro colore e questa è una limitazione poiché non tutti hanno sempre a disposizione un televisore a colori per giocare Naturalmente nulla vieta di giocare in bianco e nero se si sopporta di rendere il gioco abbastanza difficoltoso potendoci basare in tal caso, per l'identificazione delle chiavi, su due livelli di grigio

Il disegno del labirinto copre tutto lo schermo del televisore offrendo un'immagine molto grande e gradevole Il gioco può essere comandato da tastiera o usando un joystick Naturalmente la seconda soluzione rende più efficaci le manovre Durante lo svolgimento in alto sullo schermo appaiono le informazioni necessarie al giocatore e cioè il numero di "vite" ancora a disposizione prima di essere completamente sgominati dai fantasmi, il livello a cui si è giunti ed i punti ricevuti per ogni apertura di porta

T P

Produttore:
Imagine
Marons Buildings
Exchange Street East
Liverpool L2 3 PN
Prezzo:
5.50 sterline

Tra il discreto numero di ROM contenute in pacchetti tipo cartridge per il VIC 20, questa non è certo una delle più recenti essendo già da molto tempo a disposizione degli utenti ma forse essa è stata fino ad ora un po' trascurata. Si tratta di una scheda che potrebbe semplificare notevolmente i problemi di programmazione quando si vuole entrare nel mondo della grafica o della musica con il computer, poiché essa sintetizza in ciascuno dei suoi comandi un notevole numero di istruzioni e di routine che produrrebbero un massivo impiego di tempo per chi volesse ottenere gli stessi risultati usando dei singoli programmi. Oltre alla semplicità d'uso di alcune specifiche caratteristiche del computer, ne deriva spesso una notevole compattazione dei programmi con conseguente leggibilità, particolarità non trascurabile.



di Tommaso Pantuso

Cartridge SUPER EXPANDER per VIC-20

Messa in funzione e caratteristiche generali

La ROM è alloggiata, insieme alla scheda che la ospita nel solito contenitore di plastica formato 14×8 cm circa da cui fuoriescono i contatti dorati della scheda vera e propria, che andranno inseriti nell'apposito connettore del bus di espansione della memoria, posto nella parte più a sinistra del computer guardandolo da dietro. La prima cosa di cui dobbiamo assicurarci prima di inserire la cartuccia è che l'alimentazione della macchina sia esclusa, cioè l'interruttore posto sul lato destro deve essere in posizione OFF, altrimenti si rischia di danneggiare i componenti elettronici interni alla scheda. Fatto questo possiamo inserire la cartuccia facendo attenzione che l'etichetta con la scritta Super Expander sia rivolta verso l'alto. Se l'ingresso oppone eccessiva resistenza consigliamo di sfilare il cartridge e di riprovare fino al raggiungimento di esiti positivi. A questo punto possiamo accendere la macchina. La cartuccia contiene una ROM da 4K ed una RAM da 3K.

La ROM va ad allocarsi all'indirizzo esadecimale A000 (40960 decimale) che è la cosiddetta zona di autostart della memoria del VIC. All'accensione il calcolatore provvede a mandare in esecuzione una routine che parte dall'indirizzo FD22 (64802 decimale) la quale per prima cosa verifica se nella zona di autostart sia inserita una ROM. Essa controlla se dalla locazione 40964 alla locazione 40968 è memorizzata la stringa A0CBM ed in caso affermativo manda in esecuzione il programma puntato dal contenuto delle locazioni A000 ed A001 (40960 e 40961 decimali) se no il controllo passa al Basic della macchina.

NOTA	N		NOTA	N
DO#	130		DO 4	222
RE	137		DO#	224
RE#	144		RE	226
MI	150		RE#	227
FA	156		MI	229
FA#	161		FA	230
SOL	167		FA#	232
SOL#	172		SOL	233
LA	176		SOL#	234
LA#	181		LA	236
SI	185		LA#	237
DO 3	189		SI	238
DO#	193		DO 5	239
RE	196		DO#	240
RE#	199	
MI	203	
FA	206	
FA#	208	
SOL	211	
SOL#	213	
LA	216	
LA#	218	

Figura 1 - Nota ottenibile in funzione del valore di N

La memoria RAM va a porsi a partire dalla posizione 0400 (1024 decimale) e per una lunghezza di 3K. A questo proposito precisiamo che la RAM della Super Expander è compatibile con l'espansione da 8 e 16K in quanto gli indirizzi di partenza delle rispettive zone di utilizzo della memoria sono tali da non produrre sovrapposizioni nel sistema, mentre è incompatibile con l'espansione da 3K.

Con la RAM contenuta nel cartridge lo start del Basic inizia all'indirizzo 0400, la mappa video a 1E00 (7680 decimale) e la mappa colori a 9600 (38400 decimale).

Inserita la cartuccia e data tensione alla macchina, sullo schermo comparirà la scritta (con VIC inespanso)

```
**** CBM BASIC V2 ****
6519 BYTES FREE
```

con l'aggiunta dell'indicazione READY ed il cursore che lampeggia. Da questo momento in poi sono abilitati:

a) i tasti funzione,
b) la grafica,
c) i comandi musicali.

I tasti funzione

Come detto, quando il computer è in funzione con la Super Expander, diventano operanti i quattro tasti gialli posti alla destra della tastiera della macchina. Questo significa che con la pressione di ciascuno di essi otterremo un effetto diverso eventualmente programmabile. Il numero dei tasti viene inoltre virtualmente raddoppiato con la pressione contemporanea di ognuno di essi insieme allo SHIFT. All'accensione, pigiando ciascun tasto, vengono impresse le seguenti parole chiave:

f1 GRAPHIC
f2 COLOR
f3 : DRAW
f4 SOUND
f5 · CIRCLE
f6 POINT
f7 PAINT
f8 LIST + CHR$ (13)

f1, f2, f3, f5, f6 imprimono comandi relativi alla gestione della grafica, SOUND è uno dei comandi di gestione del suono e LIST ottiene come risultato quello di provocare il listing del programma composto dell'utente. CHR$(13) come è noto equivale alla pressione del RETURN.

I comandi riprodotti dai tasti funzione possono essere modificati dopo l'accensione. Il metodo è molto semplice: basta scrivere la parola chiave KEY e premere RETURN per aver visualizzate contemporaneamente tutte le parole chiave relative ai "tastoni". A questo punto dobbiamo solo

portarci col cursore in corrispondenza della parola da modificare ed agire di conseguenza La lunghezza della stringa assegnabile è quella consueta

I tasti funzione possono essere disattivati con SYS 64850 e riattivati con SYS 41031

La musica

Questa cartuccia offre ampie possibilità a chi vuole dilettarsi con esperimenti musicali permettendo melodie, accordi ed una soddisfacente scelta delle pause

Le melodie possono essere riprodotte sia in modo diretto che naturalmente, da programma Il suono viene prodotto dall'altoparlante del TV Premendo il tasto CTRL insieme alla freccetta laterale si entra nel modo musicale Se stiamo operando in maniera diretta allora i tasti

C D, E, F, G, A, B

diventano la tastiera di uno strumento e corrispondono, secondo la rappresentazione inglese delle note, a

DO, RE, MI, FA, SOL, LA, SI

Se mentre pigiamo questi tasti vogliamo vedere la lettera corrispondente al tasto musicale premuto visualizzata sullo schermo, possiamo farlo tramite la pressione, all'inizio dell'esecuzione, del tasto P Tale visualizzazione può essere disattivata premendo il tasto Q Il volume possiamo invece regolarlo premendo V seguito da un numero da 0 a 9 che attiva nove differenti livelli sonori

Le tre voci messe a disposizione dal 6561 possono invece essere selezionate tramite S seguito da un numero che può variare da 1 a 3 (S4 seleziona il *rumore bianco*) mentre le ottave coperte da ciascuna voce vengono scelte mediante il comando O seguito sempre da un numero da 1 a 3

La durata di ogni nota è stabilita da Tn con n che varia da 0 a 9

Riportiamo di seguito la durata della pausa in funzione di n espressa in sessantesimi di secondo

n	durata
0	4
1	6
2	8
3	12
4	16
5	24
6	32
7	64
8	128
9	255

SOUND è il comando che permette di ottenere degli accordi e la sua forma è la seguente

SOUND s1, s2, s3, s4, s5

dove s1, s2 ed s3 definiscono la nota da associare nella gamma dei numeri compresi tra 128 e 255 A tale proposito riportiamo in figura 1 una tabella corrispondente approssimativamente alla nota ottenuta nella rispettiva ottava a seconda del numero scelto

Per includere questi comandi in un programma basta semplicemente introdurli in un espressione del tipo

PRINT CTRL + freccia laterale

al posto dei puntini

I diesis ed i bemolle sono ottenibili apponendo prima della nota i caratteri $ e #

Ulteriori comandi sono descritti in figura 2 (pag 74) e comunque nel manuale allegato al prodotto

La grafica

E veniamo alla parte più ambita dai programmatori, che riguarda la grafica forni-

```
10 REM ----------
20 REM --          --
30 REM -- AMPL. AC --
40 REM --          --
50 REM -- (C) TP 84 --
60 REM --          --
70 REM ----------
80 POKE36879,25 REM COLORE SFONDO - BORDO
90 PRINT"□"
100 PRINT"□AMPLI INVERTENTE   ■■■"
110 PRINT"□ SCHEMA ELETTRICO  ■■"
120 PRINT"□  CALCOLO COMPONENTI■■"
130 PRINT"□ CARATTERISTICHE   ■■■"
140 PRINT"   PREMI A B O C    ■■■"
150 PRINT"□          C  TP 1984   ■■■"
160 GETA$:IFA$=""THEN160
170 IFA$="A"THENPRINT"□":GOSUB340:GOTO60
180 IFA$="B"THENPRINT"□":GOSUB640:GOSUB2:GOTO60
190 IFA$="C"THENPRINT"□":GOSUB2:GOTO160
200 GOTO160
210 PRINT"□CARATTERISTICHE■■■
220 PRINT"R1  OHM   R1
230 PRINT R2  OHM   R2
240 PRINT R   OHM   R
250 PRINT C   MF    C1
260 PRINT C2  MF    C2
270 PRINT"C3 (MF
280 PRINT FH  HZ    F
290 PRINT C         G
300 PRINT   MF THM  R1
310 PRINT■■■
```

ta dalla sezione alta risoluzione della cartuccia, la quale permette di disegnare sullo schermo grafici di funzioni e figure in multicolor

Il disegno viene composto in un rettangolo di 176 × 184 punti che vengono rappresentati su un piano cartesiano x,y ma il sistma accetta valori di x ed y compresi tra 0 e 1023 poiche provvede automaticamente alla riduzione in scala compatibile con le dimensioni dello schermo L origine degli assi e posta in alto a sinistra La grafica presenta quattro fondamentali possibilita

GRAPHIC 0 che seleziona il modo normale di operare del VIC ed e generalmente posto alla fine di un programma di grafica per ritornare ad agire in modo testo

GRAPHIC 1 seleziona il modo multicolor ed ogni carattere (punto) che appare sullo schermo e formato da 16 × 8 pixel, in altre parole ogni area elementare impegnata sullo schermo e lunga 16 e larga 8 in una matrice di schermo formata da 1024 × 1024 punti

GRAPHIC 2 pone il calcolatore in modo alta risoluzione dei colori ed il disegno e piu accurato che nel caso precedente, il formato dei caratteri e in questo modo 8 × 8 In tale stato l'uso del numero di colori e ridotto

GRAPHIC 3 attiva sia il modo multicolor che l'alta risoluzione Il colore viene controllato mediante l'uso di quattro registri inseriti nel comando COLOR s, b, c a che controllano il colore dello sfondo, del bordo, del carattere ed e introdotto il colore per un carattere ausiliario usato per dipingere ad esempio grafici in piu colori nel modo multicolor Il colore assegnato al

```
COMANDI DELLA SUPER EXPANDER

KEY        LEGGE TASTI FUNZIONE
GRAPHIC    SELEZIONA SET GRAFICO
COLOR      SCELTA DEL COLORE
POINT      DISEGNA PUNTI
REGION     CAMBIA COLORE AL CARATTERE
DRAW       TRACCIA LINEE
CIRCLE     TRACCIA CERCHI ED ARCHI
PAINT      COLORA AREE
CHAR       SCRIVE SUI DISEGNI
SCNCLR     CANCELLA SCHERMO
SOUND      COMPONE ACCORDI
CTRL + +   ATTIVA MODO SONORO
P          VISUALIZZA NOTE SUONATE
Q          ANNULLA IL PRECEDENTE
V          REGOLA IL VOLUME
S          SELEZIONA LA VOCE
O          SCEGLIE L OTTAVA
T          DURATA TEMPO NOTA
R          DURATA PAUSA TRA NOTE
#          DIESIS
*          BEMOLLE
RGR        LEGGE MODO GRAFICO
RCOLR      LEGGE VAL  NEL REG  COLORE
RDOT       LEGGE COL. DI UN PUNTO SULLO SCHERMO
RSND       LEGGE VAL  NEL REG  SUONO
RPD*       LEGGE VAL  NEL REG. PADDLE
RPEN       LEGGE VAL. NEL REG  LIGHT PEN
RJOY       LEGGE VAL  NEL REG. JOYSTICK
```

Figura 2 Riepilogo dei comandi della Super Expander

carattere tramite color può essere modificato nel corso del programma con il comando REGION n dove n corrisponde ad un colore assegnabile

Per il disegno vero e proprio sono disponibili i comandi POINT cr x y che imprime sullo schermo un carattere di colore cr in posizione x y DRAW cr,x,y TO xl, yl che traccia una linea di colore cr tra i punti x,y e xl,yl CIRCLE che disegna cerchi od archi di cerchio di raggio assegnato

Per colorare delle aree dentro una figura viene usato il comando PAINT cr,x,y che riempie con il colore cr un'area di cui e assegnato un punto interno x,y La variabile cr corrisponde al codice colore Per pulire lo schermo grafico e usato il comando SCNRL

Peculiarità notevole è che sullo schermo che riproduce il grafico possono contemporaneamente essere impresse delle stringhe per mezzo del comando CHAR riga, colonna, "stringa"

Un programma

A conclusione dell'articolo vi presentiamo un programma per il calcolo di un amplificatore invertente BF con un operazionale Nel listato completo il disegno dello schema elettrico e stato ricavato utilizzando le normali capacità grafiche del VIC

Non sono necessari molti commenti poiche il programma stesso richiede le variabili di cui ha bisogno per i calcoli, comprese le unita di misura, e cioè guadagno, frequenza minima amplificabile ed impedenza d'ingresso Anche l'output e molto chiaro

Nel listato che segue è riportato un programma che disegna solo lo schema elettrico utilizzando la Super Expander e chi volesse farlo potrebbe sostituire questo segmento nel programma precedente

Da tale schema sono stati omessi i due condensatori C2 e C3 che risultano in effetti di valore fisso

Salta subito agli occhi che la semplicita di formattazione del grafico nel primo caso va a discapito delle rifiniture

Si tenga infine presente che schemi piu complicati sono difficilmente eseguibili solo con i caratteri grafici del VIC

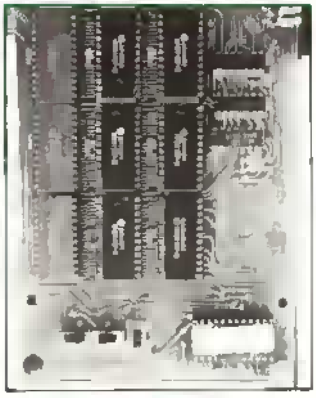
Una alternativa

La Super Expander non e l unica scheda che produce grafica introdotta sul mercato negli ultimi anni ad uso e consumo del VIC-20

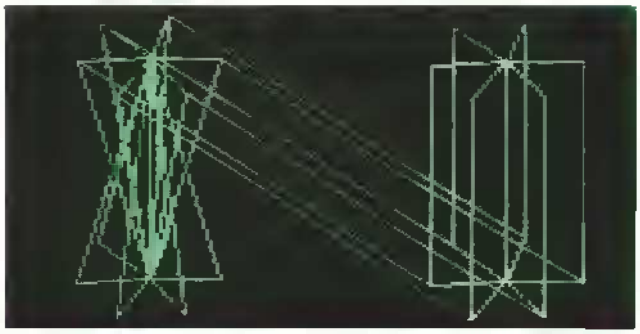
Esiste una EPROM della STACK denominata VIC KIT II (esisteva anche la VIC KIT I) che aggiunge alle capacita grafiche anche una serie di utility di aiuto alla programmazione che risultano di una certa comodita quando si lavora con il computer La EPROM in questione occupa 4K tra gli indirizzi decimali 45056 e 49152 e viene collegata alla macchina ponendola in uno zoccolo previsto appositamente su una MEMORY EXPANSION BOARD sempre della STACK, che prevede un espansione di memoria RAM fino a 24K divisi a blocchi di 8K

Le funzioni del VIC KIT II relative alla parte di aiuto alla programmazione sono AUTO che offre molteplici possibilita di numerazione delle linee di programma in modo automatico durante la stesura di un programma, DELETE per cancellare linee DUMP e FIND per ricercare parole o variabili in memoria HELP STEP, TRACE per il debug RENUMBER per la renumerazione delle linee di un programma I comandi grafici sono abbastanza simili a quelli della S E e sono GRAPHICS che serve per entrare nel modo grafico CLEAR per la selezione del colore, TEXT per ritornare in modo testo SET RESET ed INVERT per disegnare e cancellare punti dal quadro grafico, POINT per controllare se in una determinata posizione un punto è impresso LINE per tracciare linee DRAW per disegnare qualunque forma (ed in modo molto piu semplice del draw della S E) FILL per colorare PUT per manipolare aree complete sullo schermo CIRCLE per tracciare cerchi (in verita questa routine va caricata da cassetta) Naturalmente anche questa EPROM prevede il modo HI RES ed il MULTICOLOR

L unica pecca e che risulta un po complicato aggiungere del testo sui disegni ma vista la comodita d uso del DRAW i caratteri possono essere prodotti graficamente in qualunque formato



Questa foto riproduce la porzione della scheda di espansione RAM della STACK su cui è alloggiata la EPROM denominata VIC KIT II Sotto un esempio di grafica con il VIC KIT II



VIC da zero

di Tommaso Pantuso

Sul numero 28 di MC abbiamo introdotto la nozione d'interrupt ed abbiamo discusso sull'importanza di una tale tecnica per il colloquio di un microprocessore con le sue periferiche. Abbiamo anche visto sempre in linea teorica come essa viene gestito dal 6502, il processore del VIC-20. Vogliamo oggi ritornare su tale argomento per analizzarlo più a fondo e vedere come sia possibile da parte dell'utente manipolare la routine d'interrupt di tale computer per asservirla ai propri scopi.

L'interrupt del VIC

Pensiamo sia il caso di fare un breve riepilogo sul modo in cui viene gestito l'interrupt nel VIC per dar modo a tutti di capire gli argomenti che seguiranno, anche a quelli che hanno avuto la "sfortuna" di non aver letto l'articolo contenuto nel numero cui abbiamo accennato.

La funzione dell'interrupt è sintetizzata brevemente nella parola stessa che significa interruzione. In pratica quando un microprocessore riceve una richiesta d'interruzione, esso smette di eseguire ciò che stava eseguendo, va a controllare chi ha inviato la richiesta, effettua le operazioni specifiche comandate per il tipo di periferica che deve servire, eseguendo alcune routine dette di *manipolazione dell'interrupt*, e ritorna a fare ciò che stava facendo prima dell'interruzione. Naturalmente prima di diramare verso la routine di manipolazione, un processore deve conservare tutte le informazioni che gli saranno utili al rientro per riprendere il lavoro da dove lo aveva interrotto, come se nulla fosse successo, queste informazioni sono conservate in un posto chiamato *area di stack* ma per ora non ci occuperemo di questa. Aggiungiamo che ci sono due tipi fondamentali di interrupt: uno di tipo mascherabile, IRQ ed uno di tipo non mascherabile, NMI.

```
ROUTINE DI MANIPOLAZIONE
DELL'IRQ

FF72   48          PHA
FF73   8A          TXA
FF74   48          PHA
FF75   98          TYA
FF76   48          PHA
FF77   BA          TSX
FF78   BD 04 01    LDA  $0104,X
FF7B   29 10       AND  #$ 10
FF7D   F0 03       BEQ  $FF82
FF7F   6C 16 03    JMP  ($0316)
FF82   6C 14 03    JMP  ($0314)
```

Figura 1. Questa routine è la prima che viene eseguita dopo una diramazione, a causa di una richiesta di interruzione e provvede a salvare il contenuto dell'accumulatore, dei registri X e Y sullo stack. Le ultime due istruzioni sono dei salti a due diverse locazioni che contengono un puntatore a 16 bit: il primo viene effettuato se si tratta di un interrupt di tipo Break, il secondo se si tratta di un IRQ.

Una richiesta di attenzione da parte di una periferica tramite una interruzione del primo tipo può essere nascosta al microprocessore che quindi la ignorerà e proseguirà il suo lavoro, mentre una richiesta del secondo tipo giungerà sempre al microprocessore e non c'è modo di nasconderglielo.

Fatto questo breve riepilogo, occupiamoci più da vicino del VIC-20 e dei suoi interrupt sulla linea IRQ (Interrupt ReQuest).

Su tale linea il processore del VIC riceve sessanta richieste di interruzione al secondo. Ciò significa che ogni sessantesimo di secondo esso interrompe il suo lavoro, ad esempio l'esecuzione di una linea di programma, al fine di eseguire le routine di manipolazione. Egli infatti dopo ogni richiesta, completa per prima cosa l'operazione che sta compiendo e poi salta alle locazioni, di memoria ROM FFFE-FFFF (65534 e 65535 in decimale) che "puntano" ad una routine che manovra l'IRQ.

Molti di voi si chiederanno a questo punto cosa significa dire che due locazioni di memoria "puntano l'indirizzo di partenza" di un sottoprogramma e riteniamo quindi doveroso un chiarimento.

Spesso in un programma si trovano delle istruzioni di salto non diretto tramite le quali il programma va ad esaminare il contenuto di una locazione che contiene l'indirizzo effettivo a cui esso si deve spostare. Tale locazione punta quindi (indica) lo spostamento finale. L'utilità di un tale modo di agire sarà compresa nel corso della lettura di questo articolo.

Per fare un esempio umano: il sistema si comporta come un fattorino che debba fare delle consegne ma non conosca a priori l'indirizzo dei destinatari perché esso gli viene comunicato di volta in volta scrivendolo su di un foglietto e depositandolo in una cassetta. Egli deve perciò, ogni volta che vuole consegnare un pacco (il cui contenuto supponiamo sia sempre lo stesso), recarsi nel posto in cui si trova la cassetta, prelevare l'indirizzo ed effettuare la consegna. L'indicazione del foglio di carta equivale al puntatore e la cassetta alla locazione di memoria in cui esso è depositato.

Leggere l'indirizzo che viene indicato da un puntatore in memoria è altrettanto semplice che leggere quello scritto sul foglietto. Come certo saprete (e chi non lo sa lo saprà ora), la memoria del VIC è paragonabile per la sua conformazione ad un libro di 256 pagine su ciascuna delle quali sono scritte (o si possono scrivere) 256 parole. Per completare l'analogia diremo quindi che ogni parola del libro equivale ad un byte nel computer e che ogni pagina equivale a 256 byte. Se torniamo alle locazioni da cui siamo partiti per questa breve digressione, cioè FFFE ed FFFF, se an-

diamo a leggere il loro contenuto tramite il comando *peek* troveremo

PEEK (65354) = 114
PEEK (65535) = 255

A questo punto voi direte "e allora" Allora il contenuto del registro di valore più elevato indica la *pagina* in cui è contenuto l'indirizzo e quello di valore inferiore indica a quale parola bisogna fermarsi scorrendo la pagina dal basso verso l'alto In altre parole l'indirizzo effettivo è fornito dall'operazione

255 × 256 + 114 = 65394

e questa una regola generale per calcolare dove puntano due registri tramite il loro contenuto

Tornando a noi, il 65394 (FF72 esadecimale) inizia una routine indicata in figura 1 che conserva nello stack che come detto è un'area appositamente studiata per operazioni di immagazzinamento dati da recuperare in un secondo momento il contenuto dell'accumulatore del microprocessore (PHA PUSH A = spingi il contenuto di A nello stack), del registro X (TXA Transfer X A PHA = trasferisci il contenuto di X in A e spingi poi A nello stack) del registro Y (TYA, PHA analoga alla precedente) Il *contatore di programma* e il *registro di stato* vengono preservati automaticamente nello stack del 6502

Dopo tale operazione, il sistema va ad esaminare il contenuto di altre due locazioni di memoria e cioè 0314 - 0315 (788 e 789 decimale) se si è verificato un IRQ e 0316 - 0317 se è avvenuto un altro tipo di interruzione il BRK (BReaK) di cui noi non parleremo

Tale contenuto punta con il solito sistema, ad una locazione di memoria da cui parte un insieme di sottoprogrammi di gestione dell'interruzione Ricapitoliamo le ultime cose

Dopo una richiesta di IRQ il processore legge il contenuto dei registri FFFE ed FFFF il quale fornisce l'indirizzo d'inizio di una routine che salva nello stack dei dati importanti e che verranno ripresi al ritorno Terminata l'esecuzione di tale routine il processore va ad esaminare il contenuto di altri due registri di memoria e precisamente 0314 e 0315 il cui contenuto punta ad una locazione da cui inizia una serie di routine che per il VIC provvedono ad aggiornare l'orologio TI (e quindi TI$), producono il lampeggiamento del cursore controllano i tasti del registratore e verificano se è stato premuto un tasto sulla tastiera per agire di conseguenza Questi passaggi sono illustrati nella figura 2

I lettori più acuti si chiederanno perché l'indirizzo della seconda routine non venga fornito direttamente alla fine della prima senza passare attraverso il puntatore memorizzato in 0314 e 0315 risparmiando così un passaggio

Il fatto importante è che questi ultimi registri sono due locazioni della memoria RAM a cui noi possiamo accedere da programma per memorizzare in esse un nuovo indirizzo che invii ad una serie di routine da noi progettate per gli scopi più svariati

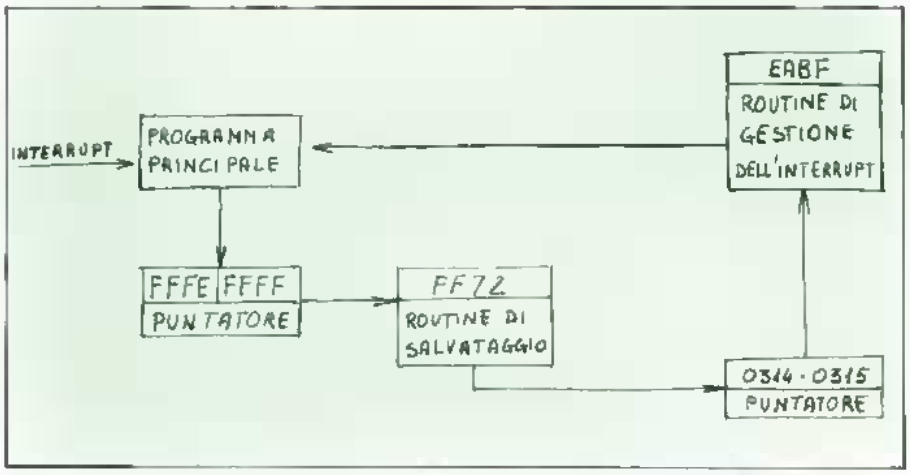


Figura 2 Quando il 6502 riceve una richiesta di interruzione tramite un segnale inviato sulla linea IRQ [illegible] all'indirizzo memorizzato con una coppia di registri FFFE ed FFFF per un IRQ Dopo un [illegible] di parametri importanti nell'area di Stack che [illegible] alle locazioni 0314 - 0315 [illegible] l'indirizzo di partenza di una serie di routine [illegible] Normalmente per il VIC tale [illegible] quando è terminata la loro esecuzione il [illegible] al programma principale [illegible]

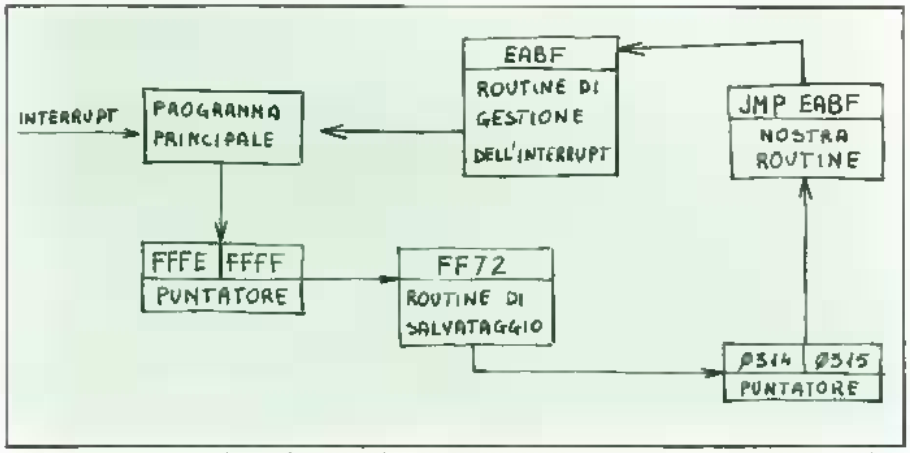


Figura 3 Noi possiamo modificare il contenuto dei registri 0314 e 0315 per farli puntare ad una routine scritta da noi alla fine della quale inviamo il controllo al [illegible] dall'IRQ

```
10 REM ******************************
20 REM ****** MINIASSEMBLER ESA ******
30 REM ******                  ******
40 REM ******     (C)TP 1984   ******
50 REM ******************************
60 POKE36879,25 :REM CAMBIA COLORE SFONDO
70 PRINT"[illegible] MINIASSEMBLER ESA [illegible] PRINT
80 PRINT" [illegible] PER [illegible]"
90 PRINT" [illegible] PER STOP[illegible]"
100 PRINT"
110 PRINT"    PREMI UN TASTO
120 PRINT"
130 PRINT"[illegible](110)"[illegible]      (C) TP 1984      [illegible]
140 GET[illegible] IF[illegible]="[illegible]"THEN140
150 INPUT"[illegible]IND. DI START[illegible]" I PRINT
160 PRINT" IND      OPCOD   DCODM"
170 PRINTI
180 INPUT [illegible] A$
190 IF A$="[illegible]"THEN I=I-1 GOTO170
200 IFA$="[illegible]"THENEND
210 A=ASC(A$)-48 B=ASC(RIGHT$(A$ 1))-48
220 N=B+[illegible](B>9)+16[illegible](A>7[illegible])
230 PRINT"[illegible]"[illegible]
240 POKE[illegible]N I=I+1 GOTO170
250 PRINTPEEK(283) GOTO250
260 REM "[illegible]" = SHIFT+CLR HOME
270 REM "[illegible]" = CURSORE VERTICALE
280 REM "[illegible]" = CTRL+RVS ON
290 REM "[illegible]" = CTRL+RVS OFF
300 REM "[illegible]" = SHIFT+CURSORE VERTICALE
310 REM "[illegible]" = CURSORE ORIZZONTALE
```

Listato 1 Programmino per riscrivere in maniera [illegible] i codici operativi delle istruzioni del 6502

Questo nuovo sistema di operazioni è indicato in figura 3

Il contenuto di tali locazioni è

PEEK(788) = 191
PEEK(789) = 234

quindi normalmente esse indirizzano alle routine che iniziano da

234 × 256 + 191 = 60095

cioè EABF esadecimale

Precisiamo che, alla fine di una nostra eventuale routine, dobbiamo inviare il programma a questo indirizzo se vogliamo che vengano eseguite anche le manipolazioni tradizionali

Capite ora l'importanza che assume il fatto di effettuare dei salti indiretti quando l'indirizzo a cui sono destinati può essere modificato perché depositato in RAM

Con la speranza di essere stati abbastanza chiari sull'argomento abbandoniamo la teoria ed andiamo a vedere come si possono sfruttare in pratica le cose descritte

Modifica del puntatore: il linguaggio macchina

Visto che si tratta di modificare le locazioni di memoria 0314 e 0315 (788 e 789

decimale), potreste pensare che basti scrivere in esse i valori desiderati tramite il solito comando di poke per risolvere il problema. Verrebbe infatti naturale credere che per inviare il controllo ad una routine che si trova a partire dall'indirizzo esadecimale 6000 (24576 decimale) basti scrivere

POKE 788 96 POKE 789 0

(infatti $96 \times 256 + 0 = 24576$) per ottenere l'effetto desiderato, ma le cose non vanno così. Infatti il contenuto delle suddette locazioni può essere aggiornato solo "turbando" momentaneamente il normale svolgimento delle operazioni del VIC, facendo in modo che, per un attimo, esso non risponda più alle richieste di interruzione permettendoci di andare a scrivere nei registri che ci interessano. Si tratta di un operazione molto semplice che però richiede di scrivere una brevissima routine in LM che sortisca l'effetto desiderato. Tale routine è riportata in figura 4 ampiamente commentata, ma noi daremo ulteriori chiarimenti per i meno esperti. Questi ultimi potranno o scegliere di approfondire un argomento fondamentale come la programmazione in linguaggio macchina (LM) che non è poi così difficile come sembra, (almeno ai primi livelli), o seguire semplicemente le metodologie che andiamo a descrivere. Noi cercheremo comunque di essere sufficientemente chiari descrivendo, istruzione per istruzione i programmi in LM che riportiamo.

Sempre ai meno esperti diciamo che il computer comprende solo il linguaggio dei numeri e scrivere in LM equivale ad inserire in macchina un programma formato da numeri, codificando numericamente le istruzioni che vogliamo impartire al microprocessore.

E come se un ricco signore che voglia parlare poco dica al suo maggiordomo "uno" se vuole mangiare, "due" se vuole bere e "tre" se vuole uscire in automobile a fare una passeggiata (per il momento fermiamoci qui). Allora se egli pronuncia, rivolto al maggiordomo, "uno, tre", significa che vuole mangiare e *poi* uscire a fare una passeggiata in automobile, mentre se dice "tre due" significa che vuole prima uscire in automobile e *poi* (probabilmente al ritorno dalla passeggiata) bere. Le istruzioni numeriche prendono il nome di *codici operativi* e sono rappresentate per il 6502, da numeri in notazione esadecimale. Essendo difficile ragionare con dei simboli sintetici come quelli numerici, qualcuno ha pensato bene di semplificare la vita ai programmatori, facendo in modo che essi potessero inserire in macchina delle parole che ricordassero l'operazione da compiere lasciando poi alla macchina il compito di trasformarle in un numero. Sono nati così i programmi assemblatori tramite i quali è possibile comunicare alla macchina i codici operativi non direttamente ma tramite un'istruzione detta *mnemonica*. Ad esempio se vogliamo dire alla macchina di caricare un dato nell'accumulatore del microprocessore (un suo particolare registro) per esempio il numero 10 (0A in esadecimale),

```
                    ABILITAZIONE DELLA NOSTRA ROUTINE

SEI             DISABILITA GLI INTERRUPT
LDA #$ LB       METTE NELL'ACCUMULATORE IL BYTE BASSO
                DELL'INDIRIZZO D INIZIO DELLA NOSTRA ROUTINE
STA $0314       LO COPIA NELLA LOCAZIONE DECIMALE 788
LDA #$ HB       METTE NELL'ACCUMULATORE IL BYTE ALTO
                DELL'INDIRIZZO D'INIZIO DELLA NOSTRA ROUTINE
STA $0315       LO COPIA NELLA LOCAZIONE DECIMALE 789
CLI             RIABILITA GLI INTERRUPT
RTS             RITORNA AL PROGRAMMA PRINCIPALE
```

Figura 4 - Questa routine disabilita gli interrupt diretti alla CPU per il tempo necessario alla modifica dei due indirizzi registri 0314 e 0315

```
                        ABILITAZIONE

SEI             DISABILITA GLI INTERRUPT   78          120            02A2
LDA #$ B9       BYTE BASSO LOCAZIONE DI    A9 B9       169 185
                INIZIO DELLA ROUTINE BEEP
STA $0314       LO PONE NELLA LOCAZIONE    0D 14 03    141  20   3
                DECIMALE 789
LDA #$ 02       BYTE ALTO LOCAZIONE DI     A9 02       169   2
                INIZIO DELLA ROUTINE BEEP
STA $0315       LO PONE NELLA LOCAZIONE    8D 15 03    141  21   3
                DECIMALE 789
LDA #$ F4       CARICA IN A 244 DECIMALE   A9 F4       169 244
STA $900A       LO PONE NELLA LOCAZIONE    8D 0A 90    141  10 144
                36875 E ATTIVA IL TONO
NOP             SPAZIO LIBERO              EA          234
NOP             SPAZIO LIBERO              EA          234
NOP             SPAZIO LIBERO              EA          234
NOP             SPAZIO LIBERO              EA          234
NOP             SPAZIO LIBERO              EA          234
CLI             ABILITA GLI INTERRUPT      58           88
RTS             RETURN                     60           96

                        DISCRIMINAZIONE

LDA #$ 40       64 DECIMALE                A9 40       169  64            02B9
BIT $00C5       ESEGUE L'AND DI A CON IL   2C C5 00     44 197 0
                CONTENUTO DEL REG. 197
BNE 02E3        SE NON E' SODDISFATTO      D0 23       208  35
                SALTA ALL'ULTIMA ISTRUZ.

                        BEEP SOFT

LDA #$ 0F       15 DECIMALE IN A           A9 0F       169  15
STA $900E       LO PONE NEL REGISTRO DEC.  0D 0E 90    141  14 144
                36070 E ATTIVA IL VOLUME
LDA #$ 00       0 DECIMALE IN A            A9 00       169   0
STA $911B       LO PONE NEL REGISTRO DEC.  8D 1B 91    141  27 145
                37147 POSIZIONANDO ACR
LDA #$ 00       0 DECIMALE IN A            A9 00       169   0
STA $9118       CARICA LATCH BASSO DI T2   8D 18 91    141  24 145
LDA #$ FF       255 DECIMALE IN A          A9 FF       169 255
STA $9119       CARICA LATCH ALTO  DI T2   8D 19 91    141  25 145
                E AVVIA IL CONTEGGIO
LDA #$ 20       32 DECIMALE IN A           A9 20       169  32
BIT $911D       ESEGUE L'AND DI A CON IL   2C 1D 91    44   29 145   02D6
                TENUTO DEL REGISTRO IFR
BEQ 02D6        SE NON E' SODDISFATTO      F0 FB       240 251
                VA  ALL'ISTRUZ. PRECED.
LDA $9118       CARICA IN A IL CONT.DI T2  AD 18 91    173  24 145
LDA #$ 00       0 DECIMALE IN A            A9 00       169   0
STA $900E       LO PONE NEL REGISTRO DEC.  8D 0E 90    141  14 144
                36878 E DISATTIVA VOLUME
JMP $EABF       SALTA ALL'INTERRUPT ORIG.  4C BF EA    76  191 234   02E3
```

Figura 5 - Questa routine produce un beep udibile dall'altoparlante del tv/monitor ogni volta che viene premuto un tasto

dobbiamo comunicare avendo a disposizione un tastierino numerico, l'istruzione A9 0A che significa proprio carica nell'accumulatore (A9) il numero 10 (0A)

Avendo invece a disposizione una tastiera alfanumerica ed un programma assemblatore, potremo comunicare alla macchina l'istruzione

LDA 0A

dove LDA deriva dalla parola LoaD A cioe *carica l'accumulatore* che viene convertita in termini numerici dall'assemblatore (assembler) Tutto qui Per fare i primi programmi non rimane quindi che imparare i codici operativi od i comandi mnemonici delle istruzioni del microprocessore che si ha a disposizione (a seconda che si possegga o no un assembler), e cominciare a fare un po di pratica

Nel listato I (pag 87) vi presentiamo un programmino, impropriamente chiamato MINIASSEMBLER ESA, che vi permette di inserire in macchina direttamente i codici operativi delle istruzioni in esadecimale, in quanto pensera esso a tradurli in notazione valida per il Basic del VIC (decimale) Non pensiamo sia il caso di ulteriori commenti in proposito in quanto il programma stesso inviera i messaggi di richiesta delle opportune informazioni Aggiungiamo che

IND DI START >

richiede l'indirizzo da cui si vuole incominciare a memorizzare il programma

Primo passo: la routine di abilitazione

In figura 4 e riportata la routine in LM che permette di modificare il contenuto dei registri 0314 e 0315 per fare puntare il loro contenuto a quella costruita da noi descriviamola

SEI

disabilita gli interrupt forzando ad "uno" il bit di disabilitazione del *registro di stato* del microprocessore Dopo questa istruzione la CPU non risponde più alle richieste di interruzione Possiamo a questo punto andare a modificare il valore del puntatore Questo si fa ponendo il byte basso dell'indirizzo a cui si punta, LB, in 0314 e quello alto, HB, in 0315 (ricordate che, come detto, deve sempre essere HB × 256 + LB = indirizzo puntato)

Cio puo essere ottenuto con la sequenza

LDA #$ LB
STA $0314

cioe carica nell'accumulatore, A, il numero esadecimale LB e copialo nel registro 0314,

LDA #$ HB
STA $0315

stessa procedura per HB

Tenete presente che il trasferimento di un numero in memoria con il 6502 non puo essere fatto direttamente, ma bisogna prima depositare il numero in oggetto in un registro di passaggio, chiamato accumulatore, e poi da questo trasferirlo all'indirizzo desiderato

CLI CLear I

rimette a zero il bit di disabilitazione degli interrupt contenuto nel *registro di stato* e riabilita la macchina a ricevere di nuovo le interruzioni che seguiranno Ricordatevi di inserire sempre dopo una routine di manipolazione degli interrupt, questa istruzione per evitare che vadano perse delle richieste d'interruzione

RTS RaTurn to Subroutine

rimanda il controllo al programma principale e chiude la routine

A questo punto la macchina, 60 volte al secondo, andra ad eseguire il programma che inizia all'indirizzo esadecimale HBLB (ricordate che gli indirizzi devono essere memorizzati in due byte, cioè essere a 16 bit, per poter coprire tutta la memoria da 0 a 65535).

Una volta inizializzato il sistema, bisogna necessariamente che esso trovi ad HBLB una routine "eseguibile" (vedremo poi in che senso), altrimenti puo succedere di tutto, ad esempio che la macchina si blocchi.

Bisogna quindi che il programma di abilitazione descritto sia seguito subito dalla routine da noi progettata

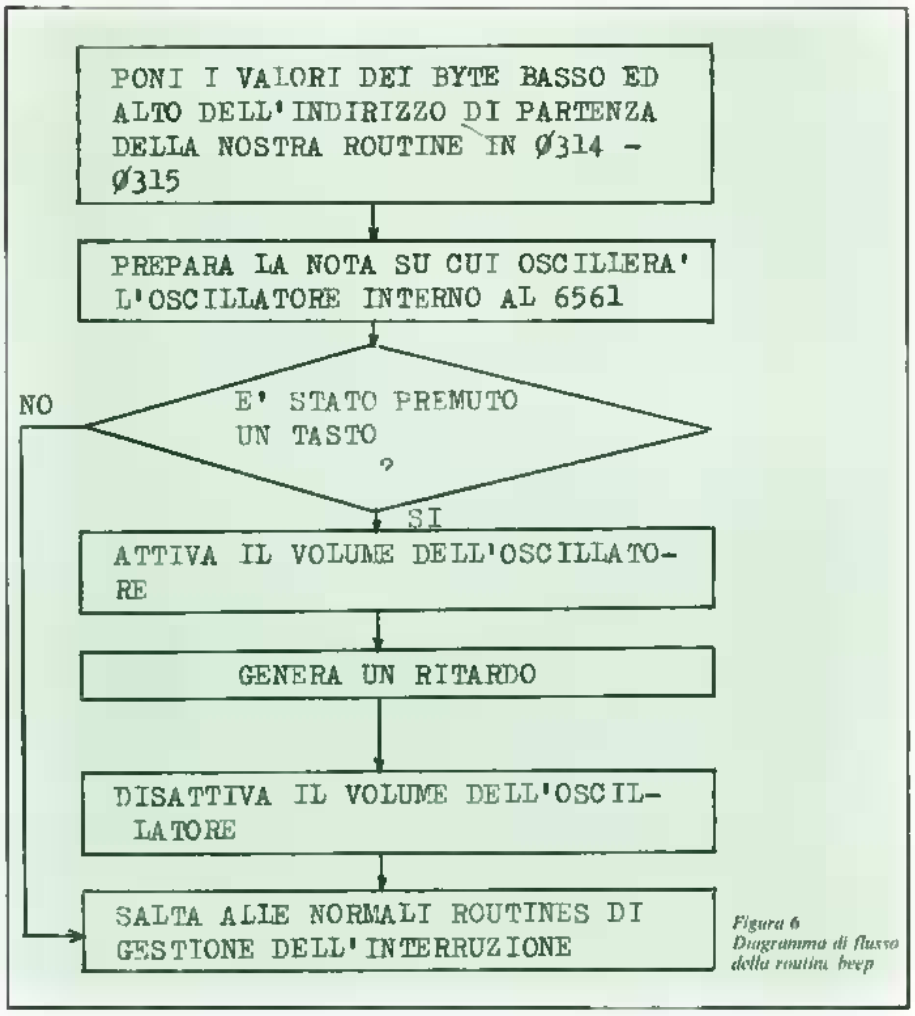


Figura 6
Diagramma di flusso della routine beep

La nostra routine: il beep

La routine che andremo ad inserire serve a produrre un suono, ogni volta che si preme un tasto sulla tastiera, che verra emesso dall'altoparlante del televisore collegato al computer (chi possiede un monitor non provvisto di altoparlante non si senta escluso dal discorso perche abbiamo pensato anche a lui!)

Di questo programma viene riportato il listato in figura 5

Si osservi che, come specificato all'inizio dell'articolo ed illustrato nella figura 3 (pag 87), esso termina con un'istruzione di salto all'indirizzo EABF da dove iniziano le *routine di gestione dell'interrupt*, alla fine delle quali si ritorna al programma principale (JMP EABF = salta all'indirizzo esadecimale EABF, JMP deriva da JuMP)

Passiamo alla descrizione del programma Esso e diviso in tre sezioni quella di abilitazione, quella di discriminazione che serve a verificare se e stato premuto un tasto e la routine vera e propria denominata beep soft

I vari passaggi saranno piu chiari se si osserva il diagramma logico che descrive l'algoritmo impiegato, illustrato in figura 6

La routine inizia dall'indirizzo esadecimale 02B9 (697 decimale) poichè da 02A1 a 02FF (673 e 767 decimali) e situato un "pezzo" di RAM non utilizzata dal sistema e sufficiente ad ospitare programmi in LM lunghi fino a 94 byte Questo indirizzo di partenza è posto nelle locazioni 0314 e

0315, rispettivamente B9 (185 decimale) in 0314 e 12 (2 decimale) in 0315. Infatti, se effettuate la semplice verifica che vi abbiamo insegnato, otterrete

2 × 256 + 185 = 697

che è proprio l'indirizzo decimale da cui inizia il nostro sottoprogramma.

Il *beep* è ottenuto tramite l'oscillatore audio contenuto all'interno del chip 6561. Come certo saprete, esso può oscillare su tre ottave differenti (cioè su tre gamme di frequenza) ed ognuna di queste può essere attivata ponendo il valore della nota da generare nel registro 36874 per la prima ottava, nel registro 36875 per la seconda ed in 36876 per la terza. I valori da introdurre per ottenere le differenti note variano da 128 a 254. Per udire effettivamente un suono, che viene emesso dall'altoparlante, bisogna abilitare il volume dello stadio finale di amplificazione del chip che varia tra quindici livelli possibili, regolabili tramite un valore compreso tra 0 e 15 posto nel registro 36878. Il valore '0' inibisce l'audio.

Durante la routine di abilitazione viene introdotto nel registro 900A (36874 decimale) il valore F4 (244 decimale), che abilita l'oscillatore ad oscillare sulla frequenza di circa 395 hertz. Gli spazi liberi lasciati in tale sezione del programma serviranno ad inserire in seguito delle altre istruzioni che esplicheranno un'importante funzione, ma per ora non ci occuperemo di esse.

Passiamo a descrivere il comportamento della parte denominata DISCRIMINAZIONE.

Nella locazione C5 (197 decimale) è contenuto un valore che varia a seconda del tasto premuto e ne rappresenta il codice di tastiera. Se non viene pigiato alcun tasto, tale registro contiene il valore 40 (64 decimale). La nostra routine comincia col caricare nell'accumulatore il valore esadecimale 40 e ne esegue l'AND con il contenuto del registro suddetto (BIT 00C5). Se il contenuto di 00C5 e dell'accumulatore (che in questo momento contiene 40) sono uguali, significa che non è stato premuto nessun tasto, quindi l'AND è soddisfatto ed un flag del registro di stato (il flag Z) viene posto a zero: in tali condizioni il sistema non deve continuare ad eseguire la nostra routine e salta all'ultima istruzione (BNE 02E3 = Branch on Not Equal to zero = dirama alla linea 02E3 se Z = 0).

Se invece un tasto è stato premuto, il contenuto di A e di C5 sono diversi: l'AND tra essi non è soddisfatto ed il flag Z viene posto ad 1. In tali condizioni si passa ad eseguire il resto della nostra routine e cioè:

a) si pone 0F (15 decimale) in 900E (36878 decimale) ed in tal caso il volume viene attivato;

b) si predispone uno dei timer del VIA 6522 a funzionare come generatore di ritardo hardware;

c) si caricano i latch ed i contatori con il valore del ritardo e si avvia il conteggio.

Quando il conteggio è terminato, viene posto ad *uno* il bit corrispondente a T2 (che è il timer che stiamo usando) nel registro dei flag del 6522 posto all'indirizzo 911D (37149 decimale).

Quando tale flag è ad *uno* nel registro 911D si trova il valore 20 (32 decimale): noi carichiamo allora 20 nell'accumulatore ed eseguiamo l'AND con il contenuto di tale registro. Se esso non è soddisfatto, significa che il timer non ha finito il conteggio e si riesegue l'AND; altrimenti si riazzera il registro dei flag del VIA, grazie all'artificio di una lettura nel registro contenente il latch basso di T2 (9118), e si salta all'indirizzo EABF da cui partono le consuete routine di gestione delle interruzioni.

Non ci siamo soffermati molto sull'uso dei registri del 6522 poiché lo abbiamo fatto già ampiamente nei numeri 28 e 29 della rivista e dovreste perciò aver acquisito una certa dimestichezza con essi.

Ed ora?

Se osservate il listato della routine descritta, esso riporta nella parte destra la codifica delle istruzioni mnemoniche in esadecimale ed in decimale. Le prime possono essere introdotte tramite il programma MINIASSEMBLER ESA riportato in questo articolo, oppure, come ormai ben sapete, memorizzate tramite il comando POKE.

Non dovete stancarvi a comporre il programmino che esplichi l'ultima funzione descritta perché ve lo forniamo noi in figura 7.

Copiate tale programma ed avviatelo. Naturalmente dopo il RUN non osserverete nulla di diverso ma non preoccupatevi perché il programma è stato memorizzato e quindi potete anche dare il NEW che cancellerà le istruzioni in Basic (se non lo fate è lo stesso).

Ponete al giusto volume il televisore e digitate

SYS 674

che renderà operativa la routine di preparazione dell'interrupt e ritornerà poi al Basic. Da questo momento in poi potrete usare normalmente il vostro VIC, solo che ora, ogni volta che voi premerete un tasto, udirete un *beep* prodotto dall'altoparlante del TV.

Simpatico, no?

```
10 REM * BEEP SOFT *
20 REM
30 FORI=674TO741
40 READA POKEI,A NEXT
50 DATA 120,169,185,141,20,3,169,2,141,21,3,169
60 DATA 244,141 10,144,234,234,234,234,234,88 96
70 DATA 169,64,44,197,0,208,25,169,15,141,14,144
80 DATA 169,0 141,27,145,169,0,141,24,145,169,255
90 DATA 141,25,145,169,32 44,29,145,240,251,173,24
99 DATA 145,169,0,141,14,144,76,191,234
```

Figura 7. Programma per introdurre in memoria la routine in LM che produce il beep.

I commenti

Benché la routine funzioni bene in pratica e sia sufficientemente valida in linea teorica, essa può spesso risultare scomoda per determinate operazioni. Infatti, se bisogna semplicemente scrivere o listare dei programmi o cose del genere, tutto va bene, mentre se si deve fare un uso molto frequente del cursore per inserzioni o correzioni, la velocità di ripetizione viene turbata dall'eccessivo ritardo.

Noi abbiamo introdotto il ritardo massimo ponendo in 9118 il valore FF, cioè 255 decimale. Voi potrete modificarlo con il comando

POKE 720 N

dove N può variare da 1 a 255, che modifica una linea del programma in LM, oppure sostituendo lo stesso valore nell'ultimo dato della linea 80 del listato della figura 7.

Vi accorgerete che sotto un certo valore, normalmente sotto N = 100, il suono udito sarà sempre più distorto e simile più ad UN PRR che ad un beep. Decidete comunque voi cosa volete fare: se cambiare tale valore o no.

Questo problema sarà risolto nel prossimo numero dove presenteremo lo stesso programma modificato in modo che possa fornire un segnale di controllo ad un circuitino esterno appositamente studiato, il quale provoca un *beep* ogni volta che è sollecitato e genera un proprio ritardo indipendentemente dalla durata del segnale di controllo.

Ci preoccuperemo di evitare che la nostra routine venga disabilitata premendo insieme i tasti STOP e RESTORE, descrivendo il modo in cui disabilitare il tasto RESTORE.

Vi forniremo inoltre altre routine di interesse ed utilità.

Appuntamento al prossimo mese. MC

DÀ PIÙ GRINTA AL TUO VIC 20

CON I VIDEOGIOCHI IMAGIC E I CONTROLLI POINTMASTER

Finalmente anche i passessari del computer Commadare Vic 20 possono videogiocare da esperti.

Imagic infatti, ho reso compatibili con il loro sistemo alcuni tra i successi mondioli dell'ultimo minuto come Demon Attack, proclamoto gioco dell'onno, Atlantis e Dragon Fire Sono quindi per lo prima volta disponibili i videogames **imagic** ai primi posti nelle clossifiche

Paintmaster Competitian Joystick, il controllo di gioco più perfezionoto e funzionole in commercio è oro odottobile ol tuo Vic 20. Pointmoster consente di roggiungere punteggi non ottenibili con i controlli standord; il vero Joystick per professionisti di videogomes.