di Tommaso Pantuso

VIC da zero +64

I caratteri del computer

Nato traendo spunto da alcuni degli argomenti trattati nel software dei lettori, l'articolo di questo mese si integra bene nel filone che riguarda l'output video del C 64 e del VIC. Abbiamo infatti parlato della memoria di schermo e di come manipolarla a nostro piacimento per modificarla o conservarla su di un supporto magnetico. Oggi cominceremo invece a trattare alcuni degli aspetti più interessanti legati ai modi in cui i chip specializzati, contenuti nei due computer Commodore di cui ci stiamo interessando, processano un certo tipo di informazioni che concorrono a formare una determinata immagine sullo schermo.

Ad alcuni degli argomenti che tratteremo è già stato dato dello spazio su MC in altra epoca, essendo stati trattati nella loro generalità. Di essi, noi vogliamo approfondire alcuni aspetti trattandoli, come al solito, in maniera tale da renderne accessibili a tutti i contenuti e fornendo, anche al lettore meno smaliziato, le procedure necessarie che gli permettano di trarre tutti i vantaggi possibili dai concetti appresi.

Corrispondenza codice-carattere

Come sappiamo esistono, sia nella memoria del C 64 che del Vic 20, certe zone che contengono una parte delle informazioni sfruttate dal sistema per imprimere dei caratteri sullo schermo. Una di queste è la Memoria del colore dalla quale viene prelevato o nella quale si può depositare un numero cui corrisponderà un certo colore per il carattere sul video. Un'altra, la Memoria di schermo, rappresenta, in codice "l'immagine" di ciò che è presente sul teleschermo: modificando il contenuto di tale zona, viene modificato ciò che noi vediamo, cioè i caratteri che appaiono sullo schermo. Come visto, se ad esempio poniamo il numero "1" in una delle locazioni di questa zona di memoria, nel corrispondente punto dello schermo comparirà la lettera "A", che è appunto rappresentata, in codice, dal numero "1".

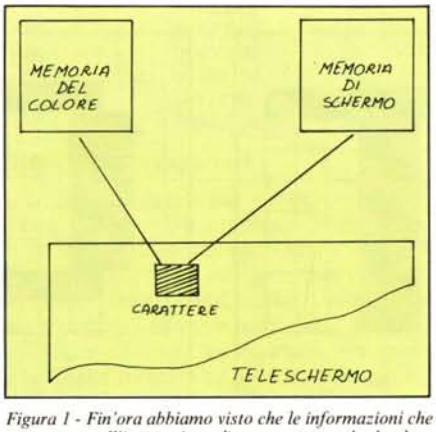

Figura 1 - Fin'ora abbiamo visto che le informazioni che concorrono all'impressione di un carattere sul teleschermo vengono prelevate dalla memoria del colore e dalla memoria di schermo. In effetti, per la formazione del carattere, devono essere prelevate ulteriori informazioni da un'altra sezione della memoria.

Ciò è quanto abbiamo appreso le scorse volte. Da questo modo "macroscopico" — se così si può dire — di vedere le cose, non traspare però il procedimento che permette di legare il numero che noi poniamo nella memoria dello schermo alla "forma" che praticamente poi vedremo visualizzata (figura 1). In altre parole, non si vede come il sistema operativo possa risalire al carattere partendo da un codice che si riferisce ad esso. Concettualmente la cosa è abbastanza semplice. Supponiamo di avere un insieme di cartellini, ad esempio tre, ciascuno dei quali riporti, sulla stessa faccia (quest'ultimo è un particolare poco rilevante), due informazioni: da una parte un numero, e dall'altra un disegno. Facendo riferimento alla figura 2, possiamo pensare che, ad esempio, al numero "1" corrisponda il disegno di un cubo, al "2" quello di un ombrello ed infine, al "3", una ciliegia.

Con un tipo di codifica del genere, non è difficile mettere in corrispondenza un numero con un oggetto. Supponiamo infatti — possedendo i nostri tre cartoncini contenuti (per il momento alla rinfusa) in una scatola — di essere incaricati di disegnare su una lavagna, uno dei disegni a nostra disposizione ogni volta che ci viene consegnato un foglio su cui è appuntato un numero da 1 a 3.

Allora, una volta in possesso del foglio, leggeremo il numero, cercheremo nella scatola "quel" numero e riprodurremo il disegno riportato di fianco ad esso (figura 3). Questo tipo di procedimento però può risultare rapido solo se abbiamo pochi cartellini su cui effettuare la ricerca in quanto, per un numero elevato di cartoncini il procedimento risulta abbastanza lento. Per velocizzarlo, possiamo allora pensare di cambiare la struttura del contenitore ed il metodo di accesso. Immaginiamo questa volta che esso sia formato da un tabellone luminoso su cui siano disposti tutti i disegni a nostra disposizione e al quale sia collegata una tastiera (figura 4): ogni volta che componiamo sulla tastiera in numero (sempre rilevato dal foglietto che ci viene consegnato), si accenderà la casella corrispondente al disegno che dobbiamo riprodurre. Come è semplice intuire, con questo metodo di accesso diretto, tutto il processo risulta più veloce.

Con questa descrizione, un po' pittoresca, abbiamo sostanzialmente descritto il modo in cui il nostro computer, a partire da un codice numerico, possa mettere in corrispondenza ad esso un determinato carattere la cui posizione, come sappiamo, viene stabilita da quella del codice nella memoria di schermo. In pratica, esiste nella memoria della macchina una zona, che può essere paragonata al precedente tabellone, in cui è contenuta la "forma" di ciascun carattere ed alla quale il sistema si riferisce sfruttando le informazioni conte-

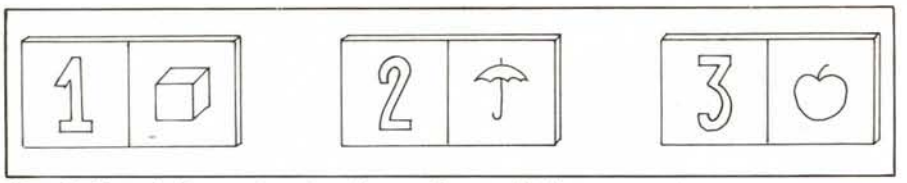

Figura 2 - Un modo di mettere in corrispondenza una forma qualsiasi con un numero.

nute nella memoria di schermo. In altre parole, quando noi digitiamo un carattere sulla tastiera, verrà letto il codice corrispondente che comparirà nella memoria di schermo (in una posizione corrispondente a quella del cursore sullo schermo) e, basandosi su questo codice, il sistema accederà all'esatta posizione da cui "cominciare" a prelevare il carattere. Abbiamo usato il termine "cominciare" per una precisa ragione. La "forma" completa del carattere è infatti rappresentata da più informazioni contenute in più locazioni per cui il sistema dovrà prelevarle tutte. Questo concetto sarà più chiaro se analizzeremo più in dettaglio la struttura di ciascun carattere e come essa venga codificata in quella zona chiamata Generatore di carattere.

Generatore e caratteri

Immaginate di avere una griglia quadrata composta da otto quadratini per lato: avremo a disposizione complessivamente sessantaquattro caselline. D'ora in poi chiameremo matrice quadrata 8x8 la griglia in questione. Supponiamo che in ciascuna casella sia contenuta una lampadina il cui stato (accesa o spenta) sia indipendente da quello delle lampadine circostanti. Se ad esempio supponiamo di accendere tutte le lampadine della quarta e quinta fila verticale e tutte quelle della prima orizzontale otterremo una configurazione simile a quella illustrata nella figura 5b) dove, per indicare le lampadine accese, abbiamo colorato di nero i quadratini corrispondenti. Se osservate bene la figura, noterete che, l'insieme delle lampadine accese (quadratini colorati), concorre a formare una figura simile alla lettera "T". Sostanzialmente, per la formazione di un carattere, succede la stessa cosa anche sul teleschermo. Ogni simbolo, può immaginarsi contenuto in una piccola matrice 8x8 della quale viene visualizzato solo un insieme di punti, che prendono il nome di "Pixel", assimilabili alle lampadine del nostro esempio precedente. In definitiva, quando un pixel è acceso (On), in corrispondenza ad esso vedremo sullo schermo un punto luminoso di colore differente da quello dello sfondo e, viceversa, se un pixel è spento (Off), esso non verrà visualizzato sullo schermo.

Vi chiederete a questo punto come faccia allora il sistema ad individuare quali siano i punti da tenere accesi o spenti partendo semplicemente dal codice contenuto nella memoria di schermo. Anche ciò non è concettualmente molto difficile.

Osservate, nella figura 6a), come la lettera "C" possa essere rappresentata in un modello a matrice come quello presentato poc'anzi. Immaginiamo ora di sostituire, al posto di ogni quadratino vuoto, che rappresenta un pixel spento, uno"0" e, al posto di un quadratino colorato, (pixel acceso), un "1". La rappresentazione che otteniamo è quella riportata nella sezione b) della figura 6. Quest'ultima configurazione ottenuta, si presta in maniera ideale ad essere decodificata dal sistema operativo in quanto, ciascuna riga, rappresenta una parola in codice binario, che è proprio quello secondo il quale "ragioniano" macchine come i computer. Vediamo allora come il nostro sistema sfrutta le informazioni racchiuse in questi gruppi di parole binarie per giungere alla rappresentazione di un carattere sullo schermo.

Il generatore di caratteri è un "magazzino" in cui ogni carattere è rappresentato da un gruppo di otto locazioni di memoria (figura 7), in ciascuna delle quali è conte-

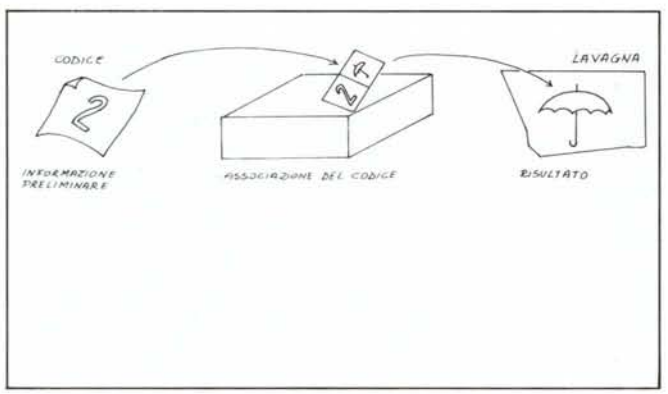

Figura 3 - Ecco come, partendo da un codice di riferimento fornitoci dall'esterno, si può risalire ad una determinata forma.

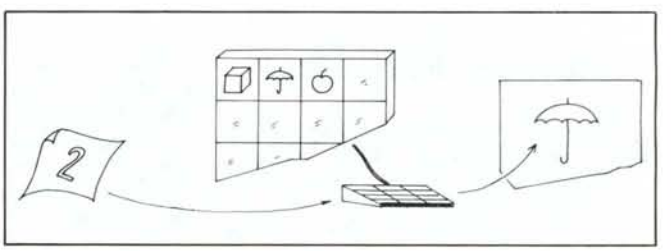

Figura 4 - Un metodo ad accesso diretto per la decodifica di un'informazione. Sul foglietto ci viene fornito un numero: componendo quel numero su una tastiera collegata ad un tabellone luminoso si illuminerà la casella riportante il disegno che dovremo riprodurre sulla lavagna.

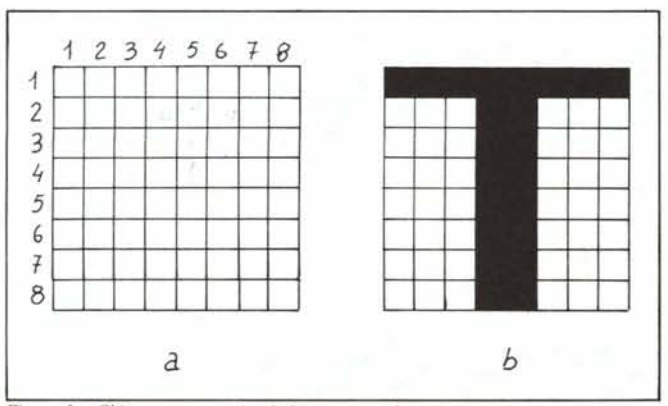

Figura 5 - Chiameremo matrice 8x8 questa griglia composta da 64 quadretti. Se immaginiamo che in ciascuna casella sia contenuta una lampadina e accendiamo tutte quelle della prima riga e della quarta e quinta colonna, otterremo un insieme che rappresenta la lettera T. In questa figura le lampadine accese sono indicate da un quadratino annerito.

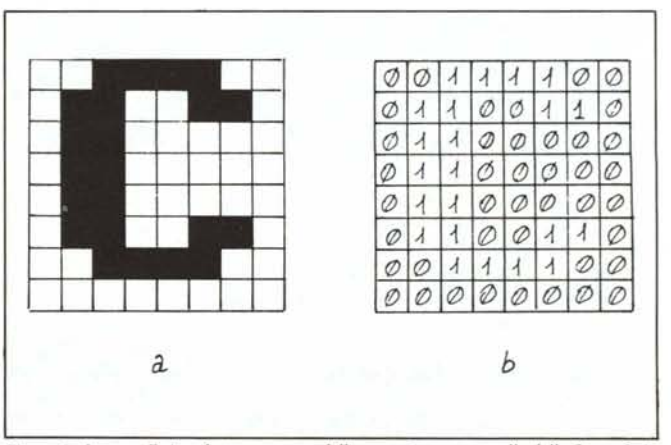

Figura 6 - Lettera C riprodotta con un modello a matrice come quello della figura 5a). In 6b) potete osservare una codifica numerica del contenuto della griglia riportata in 6a) in cui ad ogni pixel acceso è stato associato un "1" e ad ogni pixel spento uno "0".

nuta una configurazione come quella rappresentata da ciascuna riga della griglia della figura 6. Con il codice prelevato dalla memoria di schermo, il sistema operativo individua subito l'inizio di ciascun carattere posto nel generatore: legge il primo byte, corrispondente alla prima riga del carattere, e riproduce sul teleschermo un pixel acceso ogni volta che incontra un "1" ed un pixel spento ogni volta che trova uno "0". Ripete poi il procedimento con la seconda riga e così via fino al completamento del carattere. Se osservate lo schermo da vicino, potrete facilmente individuare "ad occhio" la posizione dei pixel "On" e di quelli "Off".

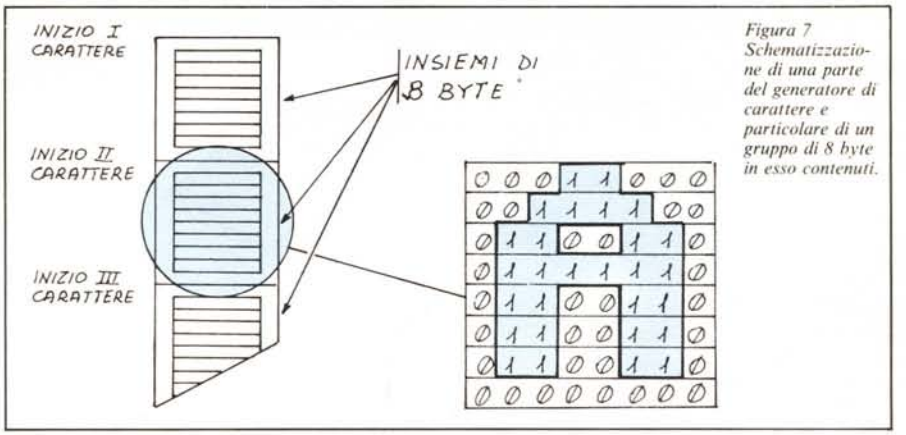

Figura 7 Schematizzazione di una parte del generatore di carattere e particolare di un gruppo di 8 byte in esso contenuti.

Come noi vediamo il contenuto del generatore di carattere

Supponiamo di aver individuato nel generatore di carattere la posizione esatta da cui parte l'insieme dei byte che racchiudono la lettera "C" e che essa inizi dalla locazione di memoria 53272. Se proviamo a leggere da Basic con "Peek" (ammesso che lo si possa fare in maniera semplice) il contenuto degli otto byte racchiusi tra 53272 e 53279 troveremo i seguenti numeri:

60,102,96,96,96,102,60,0.

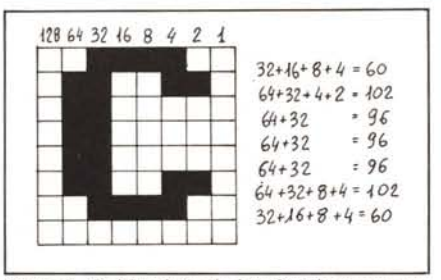

Figura 8 - Codifica decimale dei valori che compongono il carattere.

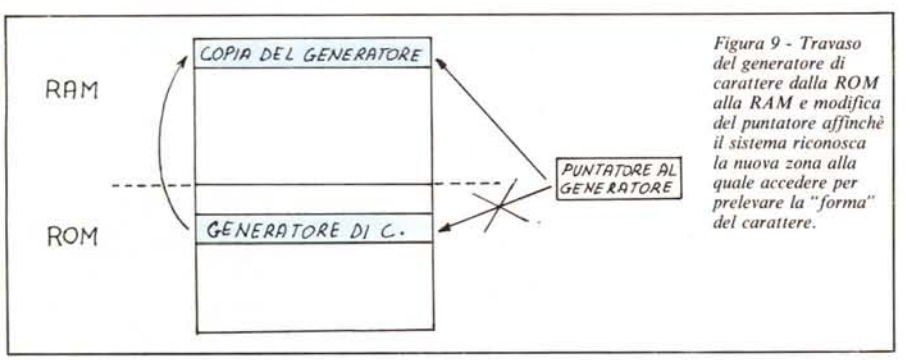

Figura 9 - Travaso del generatore di carattere dalla ROM alla RAM e modifica del puntatore affinchè il sistema riconosca la nuova zona alla quale accedere per prelevare la "forma" del carattere.

Sembrerebbe allora che, tutto quanto abbiamo detto, non corrisponda alla realtà. In effetti i conti tornano ancora. Guardate la figura 8, nella quale riportiamo ancora la matrice in cui è contenuta la lettera "C". In essa, ogni colonna è contrassegnata da un numero che d'ora in poi chiameremo "peso". Prendete ora ciascuna riga della matrice e sommate, per ciascuna di esse, tutti i pesi corrispondenti ad un pixel acceso. Per la prima fila allora, essendo accesi i pixel corrispondenti terza, quarta, quinta e sesta colonna, otterremo il numero 32+16+8+4=60, per la seconda otterremo 64+32+4+2=102 e così via fino all'ultima riga per la quale otterremo come risultato 0. I numeri trovati sono esattamente quelli che abbiamo letto nel generatore di caratteri: tali numeri altro non sono che la rappresentazione decimale, che il computer fornisce a noi, dei pattern binari che si trovano nella sua memoria. Il riepilogo di queste ultime evoluzioni può essere sintetizzato con la situazione seguente:

CARATTERE	BINARIO	DEC
++0000++	00111100	60
+00++00+	01100110	102
+00+++++	01100000	96
+00+++++	01100000	96
+00+++++	01100000	96
+00++00+	01100110	102
++0000++	00111100	60
++++++++	00000000	0

Modifica dei caratteri

Una volta imparato com'è codificato ciascun carattere e supponendo di conoscere il punto da cui essi cominciano ad essere memorizzati nel generatore, viene da pensare che, modificando il contenuto dei gruppi di locazioni interessate, non sia difficile modificare un carattere per cambiare la sua forma. Ciò in teoria è possibile, ma in pratica bisogna prima risolvere alcuni problemi. Se stiamo operando ad esempio su di un Vic 20, il generatore è posto a partire dalla posizione 32768 dalla quale si estende in su nella memoria per 4K fino alla locazione 36863. Proviamo allora a scrivere in alcune di queste locazioni con poke ma, dopo vari tentativi, ci accorgiamo di non riuscire a modificarne nessuna. La ragione è molto semplice: il generatore è situato su una memoria Rom sulla quale, per definizione, non possiamo scrivere e non possiamo quindi modificare con il comando poke. Deduciamo allora che è impossibile modificare i caratteri. Questa deduzione è però errata in quanto il sistema ci mette a disposizione i mezzi necessari per la creazione di caratteri personalizzati.

Supponiamo per un attimo di poter copiare il contenuto del generatore di caratteri in una zona libera della memoria Ram e di poter dire al sistema operativo di andare a prelevare i vari caratteri da questa nuova zona: se questo fosse possibile, i nostri problemi sarebbero risolti in quanto, trovandosi ora il generatore in Ram, sarebbe possibile modificarne il contenuto a nostro piacimento (figura 9).

In effetti, con minima difficoltà questo procedimento può essere attuato poichè esistono, sia nel Vic che nel C 64 dei registri che permettono al sistema di puntare ad un nuovo punto della memoria per andarvi a leggere i caratteri. Per continuare a fare un discorso qualitativo (le operazioni reali da compiere le studieremo la prossima volta), pensando di aver individuato la posizione dei puntatori al generatore di caratteri, non dovremo fare altro che far eseguire alla macchina per prima cosa un programma del genere:

```
10 FOR I=OTO4095
20 POKE I+A,PEEK(I+B)
30 NEXT I
```

dove A rappresenta la zona di Ram da cui vogliamo incominciare a porre la copia del generatore e B il punto d'inizio del generatore in Rom. Fatto ciò, se N è il registro (posto in Ram) da modificare, per permettergli di puntare alla nuova zona in cui abbiamo posto il generatore, aggiungeremo: 40 POKE N,n

dove n è l'adeguato numero che porremo in esso. A questo punto siamo pronti per modificare il generatore in Ram e creare i nuovi caratteri. Queste operazioni sono sufficienti per un Vic 20 mentre sono incomplete se agiamo su di un C 64 è l'argomento che tratteremo la prossima volta.

MC