

# SINCLAIR ZX 81

di Mauro di Lazzaro



**Costruttore:**

Sinclair Research Ltd,  
6 Kings Parade, Cambridge, Cambs.,  
CB21SN (GB)

**Distributore per l'Italia:**

Rebit Computer - GBC Italiana SpA -  
C.so Matteotti 66, 20092 Cinisello Balsamo (MI)

**Prezzi:**

ZX81	260.000 + IVA
Espansione 16 K RAM	191.500 + IVA
Segnalatore acustico per tastiera	44.500 + IVA
Interfaccia opzionale amplificata per registratore	41.000 + IVA
Interfaccia per monitor	41.000 + IVA
Stampante	Annunciata

Per molti la presentazione sarà inutile, ma non vogliamo dare nulla per scontato. Lo ZX81 è il nuovo modello della Sinclair Research, una ditta inglese che ormai da lungo tempo presenta tempestivamente sul mercato prodotti a basso costo.

Poco più di cinque anni fa la casa pubblicizzava in Italia (e quindi ancor prima nel paese di origine) una delle prime calcolatrici scientifiche. Anche allora il nuovo prodotto veniva fornito sia montato che in kit, fattore autoesplicativo, per la comprensione della loro politica di vendita.

È ormai iniziata la distribuzione del nuovo modello, progettato da Clive Sinclair. "Nuovo" non lo è poi molto, e capirete presto perché.

È affascinante, perché è un microcomputer completo, che misura una spanna per una spanna ed è spesso pochi centimetri. Ed ha un prezzo accessibile a tutti. Dalla calcolatrice scientifica sono passati solo cinque anni e i salti tecnologici si misurano in modo esponenziale. Aspettiamo con ansia, di vedere cosa ci proporrà la casa inglese fra un altro lustro, chissà...

**Descrizione**

Non possiamo fare a meno di ricordarvi alcune caratteristiche dello ZX80, così vi sarà più facile apprezzare (o, perché no, criticare) le novità della nuova macchina.

Il predecessore dello ZX81 aveva tutte le sue parti fisicamente raccolte su un unico circuito stampato. Gli integrati logici erano 21, fra i quali il microprocessore Z80A (o il D780C-I di produzione NEC), una ROM da 4K, che comprendeva un Basic con variabili intere e pochissime funzioni matematiche, due RAM da 1024 x 4 bit, che venivano usate per il video, per le variabili di sistema e per i programmi. Oltre a questi circuiti essenziali, erano quindi 17 gli integrati che si occupavano del controllo del sistema (interfacciamenti, decodifiche, bufferizzazioni, ecc.) e, se li mettiamo in rapporto alle prestazioni, non ci rimane che riconoscere che erano in numero decisamente limitato.

Nello ZX81 è stato fatto un passo decisivo: tutti i 17 integrati sono stati sostituiti da un unico chip da 40 piedini (il Sinclair Computer Logic). Anche la RAM da 1K è stata ridotta a un solo chip, mentre il resto dei componenti ha subito solo lievi cambiamenti.

La tastiera è separata dallo stampato principale ed è costruita con gli stessi criteri adottati nello ZX80. È formata da un secondo stampato su cui poggia una maschera con fori circolari, e da uno strato di materiale plastico flessibile, conduttivo nella parte inferiore e con il disegno e le diciture dei tasti nella parte superiore.

Premendo il tasto con un dito (ma, attenti, bene in centro), si realizza il contatto fra due piste adiacenti del circuito stampato. Anche su vecchi ZX80 questo sistema non ha presentato problemi di affidabilità, pertanto accettiamo ancora una volta di buon grado la novità, anche

se con il rimpianto di una vera tastiera, e continuiamo a ripeterci che dopotutto è una soluzione molto economica e poco ingombrante.

I tasti sono 40, organizzati in una matrice di 8 righe e 5 colonne. Per questo motivo il collegamento (flessibile) con lo stampato principale è realizzato con 13 conduttori.

Il contenitore in plastica nera conferisce alla macchina un aspetto più serio ed elegante rispetto al suo predecessore. Dal punto di vista funzionale non ci sono stati cambiamenti: ci sembra



abbiamo visto, ha anche raggiunto quello che ci sembra il più alto grado di integrazione possibile per una macchina di questo tipo.

Attorno al microprocessore sono necessarie le RAM, almeno una ROM di sistema e le interfacce per comunicare con le periferiche. Una tastiera alfanumerica completa, ad esempio, ha sempre il suo encoder. Ciò le permette di presentare il codice del tasto premuto indipendentemente dal lavoro della CPU, alla quale rimane il compito di decidere se accettare o no il dato. In



Lo ZX81 a confronto con lo ZX80. Le dimensioni della nuova macchina sono, come si vede, abbastanza sensibilmente più contenute nel nuovo modello.

che sia sempre della necessaria robustezza per proteggere il contenuto da forti urti o cadute accidentali.

Sul retro c'è lo stesso connettore da 44 contatti dello ZX80, un connettore passo 2,54 mm con slot di centraggio che mantiene invariate le caratteristiche di espandibilità del modello precedente. Sono state invece trasportate sul lato sinistro le connessioni per il registratore, il televisore e l'alimentazione.

**Hardware**

Lo ZX81 ha una struttura hardware abbastanza particolare ed è questo, a nostro parere, il punto più innovativo della macchina. Non solo ha delle soluzioni circuitali adottate per la prima volta in un microcomputer in Basic ma, come

alcuni business-computer vi è un microprocessore dedicato al solo controllo dei dati acquisiti dalla tastiera. Non possiamo certo affermare che Clive Sinclair si sia prodigato in questo senso, ma il risultato è stato estremamente interessante ed è per questo motivo che lo trattiamo in un riquadro a parte.

Nel nostro caso la tastiera è una semplice matrice di tasti che vengono periodicamente interrogati dalla CPU. Con delle routines in linguaggio macchina presenti nella ROM viene rilevata la condizione di tasto premuto, viene simulato un circuito antirimbato e, dove necessaria, viene fatta la conversione da codice tasto al carattere corrispondente. Un encoder di tastiera avrebbe pesato sensibilmente sul costo finale della macchina, avrebbe permesso un risparmio di spazio in ROM quasi irrisorio e non



La tastiera è riprodotta, in questa foto, a grandezza pressoché naturale. È evidente che non si può pretendere, con il Sinclair, di digitare in modo particolarmente veloce. Si può notare, nella stessa foto, come ad ogni tasto corrisponda più di una funzione.

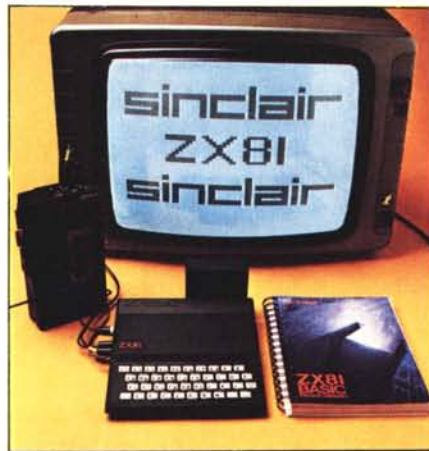
ne sarebbe stata sfruttata la maggior velocità. Coi tasti di cui dispone lo ZX81, infatti, digitare in fretta e con sicurezza è praticamente impossibile.

L'interfaccia video presenta delle soluzioni circuitali ancor più originali, che però sono la fonte delle maggiori limitazioni di questa macchina.

Normalmente per mostrare i dati sullo schermo viene usato un CRT controller, un integrato LSI che svolge autonomamente tutte le funzioni necessarie alla generazione dei sincronismi e del segnale video (ultimamente spesso a colori). Anche nei casi in cui non è usato un solo integrato, c'è un circuito incaricato di fornire costantemente questi segnali. È poi necessaria una ROM generatrice di caratteri e un'area di RAM con tante locazioni quanti sono i caratteri che si vogliono mostrare. Qualche volta la memoria di schermo è gestita come una periferica non accessibile direttamente dal microprocessore, ma molto più spesso è memory mapped, cioè costituisce una parte della memoria del microprocessore. In quest'ultimo caso è il controller del video ad accedere direttamente a quell'area di memoria (DMA), dove trova il codice dei caratteri da mostrare. Non appena cambia il contenuto di una locazione di memoria video, cambia immediatamente il corrispondente carattere sullo schermo, che presenta incessantemente il contenuto della pagina video, indipendentemente dal lavoro svolto dalla CPU.

Nello ZX81 l'elettronica a cui è affidato questo compito è veramente ridotta al minimo. I sincronismi vengono ricavati dalla stessa porta seriale di uscita per il registratore, facendo ampio uso di ritardi introdotti da software. Il generatore di caratteri è residente nella ROM (da 8K) a partire dall'indirizzo decimale 7680. Il bus degli indirizzi della ROM viene commutato con dei multiplexer in due posizioni diverse, a seconda che si trovi in stato di lettura di un'istruzione o che venga usata per generare un carattere. Nel primo caso riceve un indirizzo direttamente dal

bus del microprocessore, mentre nel secondo caso la sequenza di operazioni è decisamente complicata. La RAM viene indirizzata dalla CPU nell'area di schermo: il contenuto di ogni locazione viene posto sul bus dati che in questa fase è collegato con parte degli indirizzi della ROM. Per disegnare una riga di caratteri sullo schermo, questa sequenza di operazioni viene ripetuta otto volte per ogni carattere della riga; ad ogni nuova scansione del pennello sul video, viene incrementato di uno un conteggio a tre bit realizzato da hardware. Questi tre bit vengono anch'essi destinati ad indirizzare la ROM, che fornisce otto punti alla volta, di ogni singolo carattere; il circuito dell'oscillatore (che usa un filtro ceramico, non un quarzo) fornisce alla CPU un clock di 3,25 MHz e di 6,5 MHz a un convertitore parallelo-serie. Questo convertitore genera il segnale video partendo dal byte di



Collegato ad un registratore Sony "Walkman", lo ZX81 ha funzionato senza perdere un colpo. La foto mostra il manuale in inglese, ma le macchine sono normalmente fornite con istruzioni in italiano.

punti fornito dal generatore di caratteri e trasformandolo in una sequenza di otto punti da circa 154 nanosecondi ciascuno.

Dover disegnare otto punti di ogni carattere in 1,2 microsecondi è veramente un problema; per questo motivo l'aver limitato drasticamente il numero di integrati per lo svolgimento di questo compito ha implicato la necessità di un utilizzo complicato del poco hardware disponibile e, cosa più importante, l'impegno costante della CPU.

Ora avete tutte le informazioni necessarie per capire che quando la CPU è impegnata nella visualizzazione dei caratteri, non può fare altro ed è quindi costretta a fermare l'esecuzione del programma.

Nello ZX80 il video veniva mostrato solamente alla fine del programma, oppure in attesa di un input, poiché in questa occasione l'elaborazione è sospesa per attendere un dato dall'esterno. Anche durante la digitazione dei caratteri da tastiera la CPU è impegnata nella gestione del dato introdotto e pertanto non si può occupare della visualizzazione. Poiché il tempo richiesto è estremamente breve, si ha come risultato lo sgancio momentaneo dei sincronismi del televisore. Non possiamo negare che sia un effetto molto fastidioso, ma ancor più grave è la sparizione completa del video durante l'esecuzione del programma.

Nello ZX80 il difetto veniva parzialmente risolto con l'adozione della stessa ROM da 8K che ora viene montata sullo ZX81. Un sistema di gestione del video più completo del precedente permette di interrompere l'elaborazione per mostrare un numero di quadri video fino a 32767, specificati come argomento dell'istruzione PAUSE. Anche con questo sistema continua l'impossibilità di fare delle animazioni, sia pur molto primitive, per due motivi: il continuo sgancio dei sincronismi nel passaggio da una pausa all'altra, e la mancanza di elaborazione nel momento in cui viene mostrata l'immagine.

Lo ZX81 ci porta, oltre alla ROM da 8K

montata di serie, anche una felice aggiunta hardware. Nel modello precedente il comando SLOW non aveva nessun effetto, così lo ZX80 poteva lavorare solamente in modo FAST (stiamo sempre parlando della ROM da 8K), interrompendo l'elaborazione per mostrare il video, con l'istruzione PAUSE vista prima.

Nello ZX81 il comando SLOW ha il seguente effetto. Lo schermo è organizzato in 24 righe da 32 caratteri e, poiché ogni carattere occupa otto righe orizzontali di scansione, sono necessarie 192 righe di scansione. Il nostro standard televisivo conta 625 righe, formate però da due semiquadri interallacciati di 312 righe (e mezza, naturalmente). Siccome nel nostro caso l'interallacciamento non è utilizzato, quelli che per noi sono i "quadri" corrispondono ai semiquadri dello standard. Facendo un conto semplicissimo, troviamo che nelle rimanenti 120 righe di scansione la CPU è poco proficuamente lasciata senza lavoro.

Tutto questo ci permette di risolvere definitivamente il problema della visualizzazione; considerando quest'ultimo come compito principale della CPU, viene dedicato all'elaborazione tutto il tempo di cui è possibile disporre nella parte alta e nella parte bassa dello schermo, dove non devono comparire i caratteri.

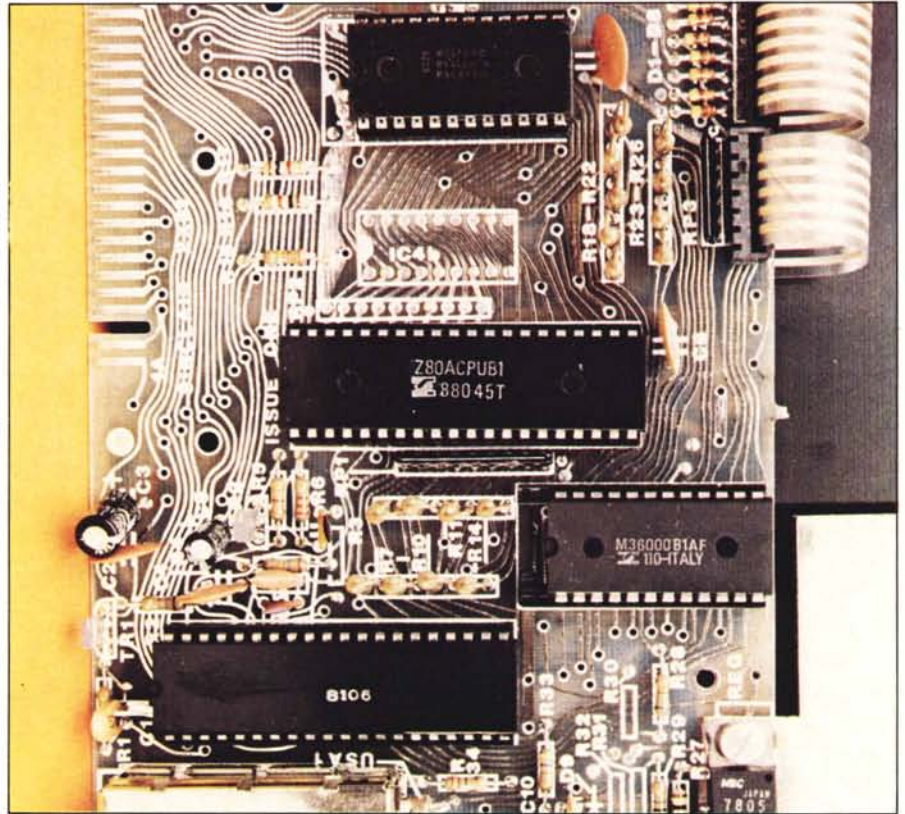
Anche se il meccanismo è molto complicato (si tratta di lavorare in divisione di tempo fra due compiti), la gestione del video è finalmente trasparente all'utente; questo vuol dire che sullo schermo compare sempre un'immagine stabile e che, modificando il contenuto di una locazione di memoria video, viene immediatamente aggiornato il corrispondente carattere.

Diventa finalmente possibile fare delle semplici animazioni anche tramite il Basic, ma rimane una discreta limitazione di velocità. Il rapporto fra il tempo dedicato alla visualizzazione e il tempo riservato all'elaborazione è circa di 4:1. Ciò significa che in modo SLOW (che è il modo in cui si trova lo ZX81 all'accensione) la velocità diminuisce di quattro volte.

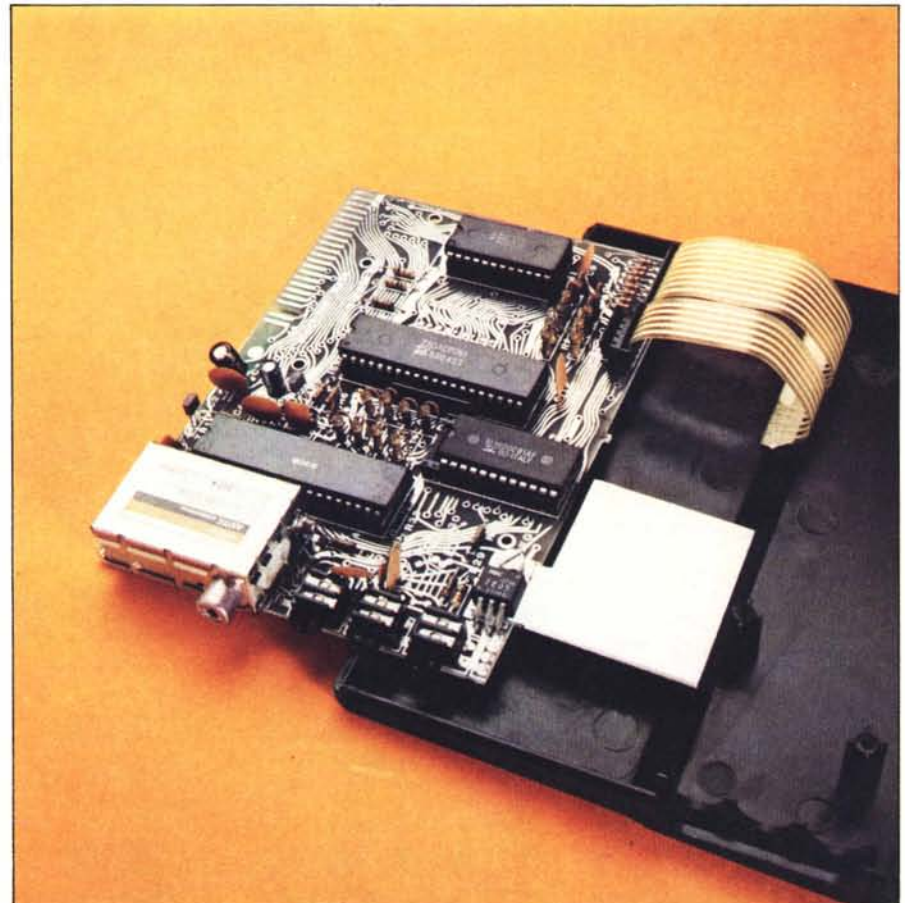
Solo così riteniamo che sia veramente riscattata la scelta di limitare al massimo la parte elettronica. Rimane sempre un sensibile decremento di velocità, è vero, ma secondo noi è da tenere in continuo rapporto con il basso costo di questa macchina.

Ci resta da fare qualche appunto sul sistema di interfacciamento con il registratore. Avevamo già accennato al fatto che i sincronismi del segnale video sono presenti sull'uscita registratore, per cui sia all'inizio che alla fine di una registrazione sono presenti sul nastro (in pratica è udibile solamente il sincronismo di quadro a 50 Hz, mentre il sincronismo di riga è a 16525 Hz, troppo elevato per la maggior parte dei registratori).

L'uscita del segnale viene portata dal livello TTL a pochi millivolt attraverso una rete di resistenze e condensatori. In moltissimi casi (registratori che non dispongono di un ingresso per microfono) il livello di uscita è insufficiente, perciò consigliamo di prelevare il segnale prima che venga attenuato e di dosare l'uscita con un potenziometro. Lo ZX81, assieme al suo predecessore, è forse l'unico microcomputer a non presentare in uscita una nota fissa prima di dare inizio alla registrazione; dopo aver dato il SAVE resta alcuni secondi in religioso silenzio. Questa sua particolarità è spesso fonte di seri problemi con quei registratori provvisti di controllo automatico del guadagno. Durante i pochi secondi di silenzio viene aumentata la sensibilità d'ingresso; il primo treno di impulsi che esce dallo ZX81 è molto importante, perché contiene i codici di start e il nome del programma. Se questa serie di impulsi ha un livello troppo ele-



Un particolare del circuito con, al centro, il microprocessore Z80A.



Una vista dello ZX81 aperto. I componenti sono su un'unica piastra (in effetti bisognerebbe definirla piastrina...) posta verso il retro, cioè non al di sotto della tastiera.

vato nei confronti della sensibilità di ingresso, viene irrimediabilmente distorta prima che intervenga il controllo automatico di volume. Inutile dire che in questa situazione il programma non può essere più riletto, nonostante si siano persi solo i primi millisecondi della registrazione, che, come vedete, non sono meno importanti. Anche in questo caso consigliamo la modifica descritta prima, che permette di regolare il volume di uscita al minimo necessario, evitando l'intervento del controllo automatico di guadagno.

Per quanto riguarda invece l'ingresso del segnale nello ZX81 è necessario un livello di un paio di volt, poiché si entra direttamente in una porta logica, senza passare da uno stadio ampli-

ficatore. Per questo motivo non è sufficiente un'uscita ausiliaria, ma bisogna prelevare il segnale da un'uscita per altoparlante o per auricolare. Con poche prove di regolazione del volume di ascolto si trova la posizione ottimale che non provoca errori di lettura.

### Utilizzazione

Dopo aver dato l'alimentazione (che può variare da 7 a 12 volt, a seconda della temperatura che volete far raggiungere allo stabilizzatore interno), vedrete comparire una K in campo inverso nell'angolo in basso a sinistra dello schermo. Non è una presentazione del tutto conven-

zionale, ma si tratta del cursore e vi diciamo subito che potrete vedere, sempre in campo inverso, anche una L, una F, una G, o una S che si somma ad una delle precedenti. Ognuno dei quattro stadi del cursore vi indica quale genere di messaggio dovete introdurre. K è l'iniziale di keyword, e vi indica che potete inserire un comando preceduto, se intendete inserire una linea di programma, da un numero di linea. La K in campo inverso si trova quasi esclusivamente all'inizio della riga, ma ad esempio compare anche dopo un THEN, per permettervi di specificare quale comando va eseguito se la relazione è vera. Da queste poche righe potete vedere che l'istruzione LET non può essere omessa, poiché sia

## Sinclair Research: qualche notizia

Ecco alcune notizie riguardanti l'attività della Sinclair, conosciuta in Italia fino a poco tempo fa solo per le sue calcolatrici ed i suoi strumenti di misura.

La Sinclair Radionics fu fondata nel 1962 da Clive Sinclair, conosciuto in Italia fino a poco tempo fa solo per le sue calcolatrici ed i suoi strumenti di misura. Nel 1976, in seguito a difficoltà finanziarie, si era inserita come una delle maggiori azioniste la National Enterprise Board (NEB). In seguito a divergenze circa il campo in cui doveva agire la compagnia, nel 1979 vi fu una scissione interna. Alla NEB rimase la parte di mercato relativa alla strumentazione mentre Clive Sinclair, che credeva nelle prospettive dell'elettronica 'consumer', creò nello stesso anno la Sinclair Research Ltd. Lo scopo di questa nuova società, nata nel luglio di quell'anno, era di ideare e di sviluppare nuovi prodotti che fossero in grado di soddisfare le richieste del mercato nel campo dell'elettronica 'consumer'.

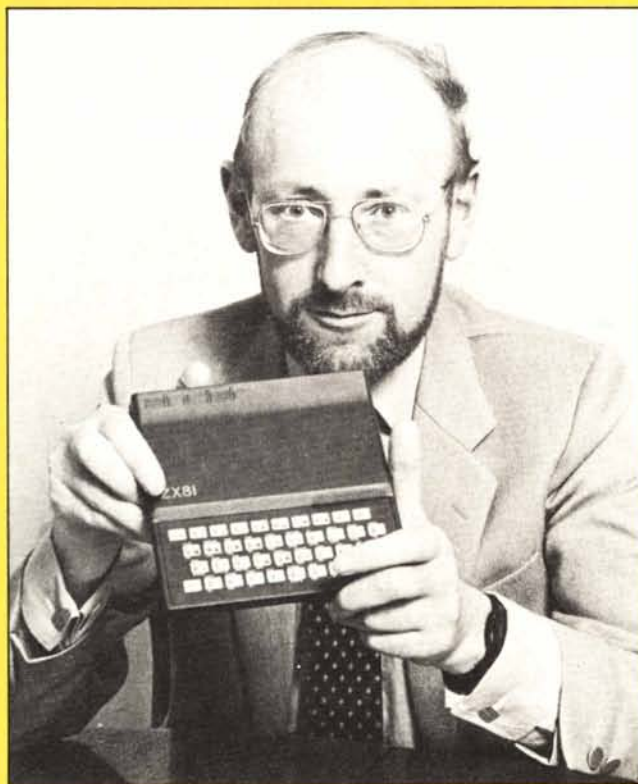
La Sinclair Research è, in pratica, un centro di ricerca dove un prodotto viene 'pensato' e progettato; per quanto riguarda le realizzazioni, ci si affida, poi, a ditte specializzate. Non esiste, infatti, una vera e propria fabbrica Sinclair ma vari fornitori esterni che costruiscono per suo conto. La prima iniziativa fu quella di realizzare lo ZX80, lanciato nel febbraio del 1980. Nello stesso mese l'apparecchio entrò in produzione presso gli stabilimenti della Timex Corporation a Dundee in Scozia (una multinazionale a capitale americano famosa ormai in tutto il mondo per gli orologi), su commissione della Sinclair. La costruzione dell'alimentatore veniva invece affidata alla Adaptors and Eliminators Ltd. (UK), una ditta specializzata. Lo ZX80 rimase così in produzione fino allo scorso agosto totalizzando oltre 100.000 pezzi venduti. La costruzione dello ZX81 cominciò invece ai primi del marzo 1981, con un ritmo di 10.000 unità al

mese. Questo ritmo raggiunse poi valori quattro volte superiori, tanto che fino a questo ottobre sono stati venduti più di 100.000 pezzi.

Per lo ZX81 ci si avvale di un nuovo aiuto tecnologico: la Ferranti, ditta inglese produttrice di integrati. Questa disponeva infatti di un integrato in grado di essere adattato facilmente a diverse esigenze; fu così progettata

to ad un prezzo sensibilmente inferiore a quello del suo predecessore.

Sebbene entrati sul mercato in tempi differenti, lo ZX80 e lo ZX81 hanno vissuto in comune un periodo di gestazione, precedente il 1979, che durò circa due anni. Ciò spiega come la Sinclair abbia potuto, avvalendosi di un progetto già pronto nel cassetto, immettere sul mercato un nuovo



e prodotta la versione Sinclair, chiamata SCL (Sinclair Computer Logic) nella quale furono racchiusi, grazie alla tecnica di integrazione a larga scala (LSI), la maggior parte degli integrati che erano già presenti nello ZX80: ben 18. Questo fatto rappresentò un passo in avanti verso la miniaturizzazione, e portò inoltre una notevole diminuzione dei costi effettivi, tanto che lo ZX81 venne immesso sul merca-

to dopo appena 16 mesi.

Qualche notizia sulle periferiche. Lo sviluppo di memorie di massa più veloci non è stato ancora preso in esame dalla Sinclair, ma la Micro Ace Ltd. (una ditta californiana a cui la Sinclair ha venduto alcuni diritti) da noi interpellata, afferma che in futuro verranno messe a disposizione degli utenti ZX80/81 delle periferiche come: uno Stringy Floppy (cioè un drive che, invece

di usare un disco magnetico, usa nastri in cassette a ciclo continuo; la velocità è di circa 7200 Baud, contro i 250 dell'interfaccia registratore!) o addirittura un Floppy Disk Drive. Lo ZX81 è stato disegnato per essere collegato con la sua Printer (immessa sul mercato inglese nel settembre 1981); la Sinclair ha comunque sviluppato un modulo hardware per collegare lo ZX81 con altri tipi di stampante. Purtroppo ci è stato assicurato che per il momento non è previsto l'uso di una tastiera standard.

Il lancio dello ZX80 e dello ZX81 aveva lo scopo di procurare i fondi necessari allo sviluppo di progetti ben più impegnativi e al raggiungimento di una certa autonomia.

Infatti la Sinclair ha ora annunciato un investimento distribuito in quattro anni, per la produzione del tubo Flat-Screen per il televisore tascabile 'Microvision', e di un ricevitore FM-TV capace di captare qualsiasi trasmissione commerciale che sarà disponibile in estate ad un prezzo circa pari a quello dello ZX81. Qualche dato sul Flat-Screen, lo schermo piatto che sarà montato sul Microvision, basterà a definire il valore di questo prodotto e a suscitare l'interesse degli addetti ai lavori.

Il tubo Sinclair misura 10 x 5 x 2 cm, è tre volte più luminoso, richiede tra 1/4 e 1/5 del normale assorbimento, e ha un volume pari alla metà di un convenzionale CRT con le stesse dimensioni dello schermo.

Sebbene giovane, la Sinclair Research vanta già oggi, una notevole rete commerciale. La sua sede principale è a Cambridge, ed ha uffici sussidiari a Monaco di Baviera, a Boston (USA) e a Parigi, che rappresentano la Sinclair in tutto il mondo. Infatti lo ZX81 viene venduto attualmente in più di 20 paesi, ai quali se ne aggiungeranno altrettanti in futuro. È recente l'accordo con il gigante Giapponese Mitsui e Co. che ha permesso anche ai giapponesi di avere sotto l'albero lo ZX81.

Giovanni Cozza



vorrebbe dire che per il Basic non ci siete tagliati...

Vogliamo farvi notare che con 1K di RAM le limitazioni sono ancor più evidenti che con lo ZX80. I byte occupati dalle variabili di sistema sono passati da 40 a 125, 22 righe complete di schermo occupano 726 byte, le variabili numeriche sono rappresentate da 5 byte anziché da 2 e lo stack, che parte dal fondo della memoria, vuole generalmente qualche decina di byte; tutto questo vuol dire che con 1024 byte in totale ci stanno ben poche righe di programma (con lo schermo pieno circa dieci). In realtà le cose sono meno gravi, perché, fino a quando non è presente l'espansione di memoria, lo schermo occupa soltanto lo spazio dei caratteri stampati, aumentato di uno per ogni riga.

In fase di scrittura del programma l'effetto è quello di veder diminuire il numero delle righe mostrate man mano che se ne inseriscono di nuove. Sembra un assurdo, ma non vi dovete preoccupare; finché ne vedete almeno una, vuol dire che ci sono ancora tutte quelle che avete battuto. Ricordatevi però che più è lungo il programma, minore è lo spazio dedicato allo schermo.

Naturalmente inserendo nel connettore posteriore l'espansione da 16K tutti questi problemi spariscono, anzi si presenta il problema di come riempire la memoria. Dobbiamo però farvi notare che un programma di grandi dimensioni arriva facilmente a richiedere cinque minuti per essere trasferito o letto dal registratore. Il sistema di registrazione usa serie di quattro o otto impulsi intercalate da una pausa; non è un sistema molto veloce, ma per contro presenta un'ottima immunità anche ad ampie variazioni di velocità del registratore.

Come potete vedere dalle fotografie, funzioni, comandi e parole chiave vengono inseriti con la pressione di un solo tasto, non battendo tutta la parola. Questa ci sembra una grossa novità in grado di fare risparmiare tempo a chi intenda usare solo lo ZX81. Per chi invece ha la sfortuna (o la fortuna?) di usare anche altri microcomputer, ci sono dei grossi problemi; assai spesso viene istintivo scrivere per intero un'istruzione, ma non appena alziamo gli occhi scopriamo di non essere capiti e ce lo conferma il syntax error marker, che segnala immediatamente il suo stupore. Riteniamo sia molto più facile passare dallo ZX81 ad un microcomputer con la tastiera tradizionale e ci rendiamo anche conto che è effettivamente questo il passaggio più logico.

Quando si imposta il programma, le linee vengono create e corrette nelle due righe in basso dello schermo, che si estendono verso l'alto per linee di lunghezza maggiore. È da notare che è difficile scrivere linee di programma più lunghe di due righe, soprattutto quando si muovono i primi passi. Non è possibile concatenare più istruzioni sulla stessa riga, come invece succede di solito sulle altre macchine e questo indica che una linea di programma così lunga non capita di frequente.

Il sistema di correzione degli errori è sovente un punto debole. Al contrario sullo ZX81 si è dimostrato sufficientemente pratico e veloce. Per correggere una linea si sposta l'indicatore, che compare a destra del numero di linea, sulla linea desiderata; questa operazione si può effettuare o con due tasti che muovono l'indicatore in alto e in basso, o con LIST e il numero di linea. Quando c'è in memoria un programma molto lungo, è molto utile il controllo alto e basso dell'indicatore, perché, se si tenta di portarlo oltre i limiti dello schermo, si ha lo scrolling automatico del listato nel senso opposto.

Dopo che si è indicata la linea da correggere, si preme EDIT e la linea viene duplicata nella

## La tastiera

Vi diamo qualche ulteriore notizia sul funzionamento della tastiera, sia per aiutare coloro che intendono programmare in linguaggio macchina, sia per chi si diletta nella costruzione di piccoli sistemi con lo Z80.

I tasti (vedi fig. 1) sono organizzati in una matrice di 8 righe e 5 colonne. Le righe sono collegate alla metà superiore del bus degli indirizzi con dei diodi che hanno il catodo rivolto verso gli indirizzi. Le colonne sono collegate al positivo con delle resistenze di pull-up e alle entrate di un sestuplo buffer three-state (74LS367). Il buffer è abilitato a trasferire la configurazione presente ai suoi ingressi sul bus dati tramite una istruzione IN A, port oppure IN reg. (C) dove il codice dispositivo (il port, o il contenuto di C) può essere qualsiasi numero pari minore di 255, cioè è necessario che sia a zero il bit più basso. L'istruzione IN A, port mette sulla metà alta del bus dati il contenuto presente nell'accumulatore fino all'istruzione precedente. IN reg. (C) carica invece in un registro il dato trasferito dal port indirizzato dal contenuto del registro C. Questa istruzione pone sulla metà alta del bus degli indirizzi il contenuto del registro B. Facendo uso di queste osservazioni, che sono seminascolte anche su un buon manuale sullo Z80, è possibile interfacciare una tastiera con un numero veramente esiguo di componenti.

Ponendo a zero uno degli indirizzi da A8 a A15, e andando a esaminare lo stato delle colonne, è possibile rilevare quale tasto è stato premuto.

Vi proponiamo (fig. 2) due brevi programmini in linguaggio macchina che testano il tasto BREAK (SPACE). Il secondo ci sembra decisamente più versatile nel caso che si voglia fare una scansione completa della tastiera, mentre con il primo si risparmiano due byte nel caso che si voglia testare un solo tasto. Facendo partire le subroutine con USR, che abbia per argomento l'indirizzo di partenza delle routines in memoria, quando viene premuto il tasto si ritorna al BASIC. Provatelo e studiatele. Ricordatevi che l'esperienza insegna!

	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>
A <sub>8</sub>	SHIFT	Z	X	C	V
A <sub>9</sub>	A	S	D	F	G
A <sub>10</sub>	Q	W	E	R	T
A <sub>11</sub>	1	2	3	4	5
A <sub>12</sub>	0	9	8	7	6
A <sub>13</sub>	P	O	I	U	Y
A <sub>14</sub>	N/LINE	L	K	J	H
A <sub>15</sub>	SPACE	M	N	B	

figura 1

3E	LD A, N	OE	ld, C, N
7F	7F	00	0
DB	in A, port	06	ld, B, N
00	0	7F	7F
F6	or N	ED	
FE	FE	78	in A, (C)
FE	cp N	F6	or N
FE	FE	FE	FE
20	jr nz, DIS	FE	cp N
F6	F6	FE	FE
C9	ret	20	jr nz, DIS
		F8	F8
		C9	ret

figura 2

parte bassa dello schermo, dove è possibile spostare il cursore a sinistra e a destra, cancellare e inserire caratteri. Chiudendo la riga con NEW LINE, la linea viene inserita nel programma e, se non si è alterato il numero di linea, sostituisce quella con il medesimo numero.

## Il Basic

Una delle prime cose che si notano con piacere nel Basic dello ZX81 sono i nomi delle variabili numeriche. Contrariamente alle stringhe, che vengono chiamate con una sola lettera seguita da \$, le variabili numeriche non hanno limiti nella lunghezza del nome e tutti i caratteri che lo formano sono significativi.

Tutte le funzioni, tranne PI e RND che non hanno argomento, possono venire scritte senza racchiudere l'argomento fra parentesi, nel caso che si tratti di una variabile o di una costante (o di un'altra funzione). Se invece si vuole usare come argomento un'intera espressione, bisogna racchiuderla fra parentesi.

Per quanto riguarda le funzioni trigonometriche è da notare la presenza di tutte le funzioni inverse, mentre assai spesso si è costretti a ricavare arcoseno e arccoseno dall'arcotangente.

Sulla tastiera è presente un carattere che si presenta come una coppia di apici; viene utilizzato nell'istruzione PRINT per inserire gli apici nella frase da stampare, senza essere costretti a utilizzare la funzione CHR\$.

Con PRINT si possono riempire direttamente 22 righe di schermo; nelle rimanenti due righe è possibile scrivere con delle POKE, dopo aver trovato le locazioni di memoria corrispondenti. Con il PRINT seguito da una virgola il cursore

si sposta al prossimo campo di stampa (all'inizio della riga, oppure al centro dello schermo). C'è anche l'istruzione TAB seguita da un numero positivo minore di 255, o uguale, che viene ridotto a modulo 32 e sposta il cursore nella posizione corrispondente al risultato.

Un'altra particolarità interessante è che, se si indirizza un numero di linea che non esiste, il programma prosegue dalla prima linea che ha numero maggiore, senza incorrere in condizione di errore.

Per l'impossibilità di concatenare le istruzioni su una stessa riga, si incontrano spesso delle difficoltà nel caso che in una frase condizionale si vogliono eseguire più istruzioni. Forse il modo più sbrigativo è quello di effettuare un salto a subroutine, se la relazione è verificata; purtroppo si nota un rallentamento nella velocità di esecuzione, se i test condizionali sono molto numerosi.

Un'altra novità che abbiamo accettato con vero piacere è la possibilità di far seguire a GOTO e GOSUB una variabile o un'intera espressione. È una novità che si fa apprezzare spesso e che manca sulla maggior parte delle altre macchine. Nello ZX81 questa particolarità supplisce egregiamente alla mancanza dell'istruzione ON ... GOTO/GOSUB.

Nella registrazione dei programmi è necessario far seguire il SAVE da un nome racchiuso fra apici, non più lungo di 127 caratteri. Quando il SAVE viene eseguito da programma, fa sì che caricando lo stesso programma con LOAD, questo parta da solo a caricamento avvenuto. Se si vuole caricare il primo programma presente sul nastro bisogna battere LOAD''''.

Vogliamo ricordarvi che in fase di stampa è possibile indirizzare il cursore in qualsiasi punto

dello schermo, eccetto le ultime due righe in basso, con AT Y, X, indicando nell'argomento prima la linea e successivamente la colonna. Un'altra istruzione particolare è SCROLL, che muove il contenuto dello schermo di una posizione verso l'alto e posiziona il cursore all'inizio della ventiduesima riga.

Le istruzioni PLOT e UNPLOT permettono di settare e di resettare ogni singolo quadratino di una matrice di 64X44. È una grafica a risoluzi-

zione molto bassa, ma sufficiente per il tracciamento di istogrammi o per i giochi.

INKEY è l'equivalente del più comune GET; legge il carattere premuto sulla tastiera nel momento in cui viene incontrata l'istruzione (e non dopo che è stato premuto perché manca un buffer di tastiera).

La ROM da 8K contiene anche le routine per la gestione della ZX81 printer, una stampantina termica dal costo limitato, in grado di stampare

32 colonne a 50 cps e di fare l'hard copy del video.

La gestione delle stringhe non è tradizionale. Non sono presenti le classiche LEFTS, MIDS, RIGHTS, ma è possibile effettuare lo slicing di una stringa indicando il carattere di partenza e quello di arrivo. Quando vengono omessi, viene considerato implicitamente l'estremo corrispondente.

Ancora una sorpresa ci viene dagli array, che possono avere un numero di dimensioni a piacere, limitatamente allo spazio disponibile in memoria. Gli array di stringhe vengono considerati come se ogni elemento contenesse un solo carattere. Pertanto l'ultima dimensione in una frase di dimensionamento di un array definisce la lunghezza delle stringhe considerate. Quando si indica un elemento dell'array e si omette l'ultima dimensione, viene presa l'intera stringa, e se si specifica anche l'ultima dimensione viene preso il carattere corrispondente. Come ultima dimensione si può anche indicare una sottostringa.

Nel Basic dello ZX81 non sono presenti le istruzioni DATA, READ e RESTORE. Quando è necessario introdurre nel programma delle tabelle di dati, è possibile farlo, ma bisogna ricorrere a una procedura abbastanza complicata. Si introducono i dati sotto la forma di arrays e si registra il programma con il solito metodo. Quando si dà l'esecuzione al programma bisogna battere GOTO seguito da un numero di linea, poiché questa istruzione (a differenza del RUN) non cancella le variabili. Si ottiene grosso modo, lo stesso risultato, ma in maniera indubbiamente più macchinosa e viene persa la compatibilità con il Basic tradizionale.

Questo procedimento è possibile grazie al fatto che il comando SAVE e LOAD agiscono non solo nella memoria di programma, ma anche sui dati.

Il manuale contiene anche una parte sufficientemente esplicita sulla ripartizione delle aree di memoria, indicando il nome di tutti i puntatori delle varie aree. Viene descritto il modo di rappresentazione in memoria delle linee di programma e delle variabili. In una tabella sono indicate le locazioni e il significato delle variabili di sistema, compresi i puntatori.

## Conclusioni

Vorremmo sottolineare quello che secondo noi è il valore di una macchina progettata con i criteri descritti. Può essere considerata una validissima alternativa ai sistemi didattici, poiché permette di avere una scheda con microprocessore a un costo molto ridotto. La programmazione in Basic conferisce al sistema un'enorme versatilità; permette di programmare il microprocessore tramite una tastiera alfanumerica, di scrivere i propri comandi di editing che visualizzano, modifichino, spostino aree di memoria. Si possono listare i programmi in linguaggio macchina sul video, anziché su un semplice display alfanumerico. Con programmi molto semplici possono essere usati tutti i sistemi di numerazione possibili, per trattare i dati in ingresso e in uscita.

Il connettore posteriore riporta tutte le terminazioni necessarie per qualsiasi espansione. Questo significa che, per chi ha aspirazioni di progettista hardware, lo ZX81 può essere un piccolo ed economico sistema di sviluppo per i suoi progetti.

Ci sembra insomma una macchina adatta a molte persone, sia a chi si avvicina alla programmazione per gioco, sia a chi vuole lavorare senza troppi sforzi con un microprocessore e ha limitate disponibilità economiche.



## Due programmini di utility

Il primo programma vi permette di esaminare il contenuto della memoria a partire da una locazione a piacere, presentando indirizzi e dati sia in decimale che in esadecimale. Purtroppo presenta sullo schermo soltanto sei righe alla volta ed è una limitazione necessaria per fare girare il programma con 1K di RAM. Se avete l'espansione potete cambiare le linee 10 e 70.

Il secondo programma è assai interessante. Fino alla linea 100 è un ottimo programma di utilizzo generale per caricare in memoria una routine in linguaggio macchina scritta in esadecimale (ad esempio una di quelle che vi abbiamo suggerito per prelevare i dati dalla tastiera). Il programma presentato genera, invece, una sequenza di note casuali che escono dall'uscita registratore. Anche qui potete esaminare la routine in linguaggio macchina e scoprire tante cose interessanti!

```

10 REM 160106050E800D20FDDB000E800D20FDD3000520EF1520EAC9
15 FAST
20 LET A = 16514
30 IF PEEK A = 40 AND PEEK (A+1) = 37 THEN GOTO 60
40 LET A = A+2
50 GOTO 30
60 LET A = INT ((A-16512)/2)
70 DIM S (INT ((A+3)/5))
75 LET P = PEEK 16400 + PEEK 16401 * 256 + 12
80 FOR T = 0 TO A-1
90 POKE P+T, (PEEK (16514 + T * 2) - 28) * 16 + PEEK (16515 + T * 2) - 28
100 NEXT T
110 LET G = INT (RND * 255) + 1
112 POKE P+1, INT ((G-1)/INT ((256-G)/16)+1) + 1
120 POKE P+5, 256-G
130 POKE P+12, 256-G
140 LET R =USR P
150 GOTO 110

1 FAST
2 CLS
3 PRINT"LOC FROM? (C TO CONTINUE)";TAB 10;"L TO RE-ENTER LOC"
4 INPUT L
8 CLS
10 FOR I=0 TO 5
15 LET K=L+I
25 LET A=INT(K/256)
35 LET B=K-A*256
40 LET V=PEEK K
50 PRINT"LOC:";K;" ";CHR$(INT(A/16)+28);CHR$(A-INT(A/16)*16+28);" ";CHR$(INT(B/16)+28);CHR$(B-INT(B/16)*16+28);TAB 16;"DEC:";V;TAB 24;"HEX:";CHR$(INT(V/16)+28);CHR$(V-INT(V/16)*16+28)
60 NEXT I
70 LET L=L+6
75 PAUSE 40000
80 LET A$=INKEY$
90 IF A$="C" THEN GOTO 8
100 IF A$="L" THEN GOTO 2
110 GOTO 75

```