

Jurassic News

Retrocomputer Magazine

Numero 2 — Marzo 2006

All'interno:

ZX80 CIRCUIT DIAGRAM

Introduzione all'analisi algoritmica
Prova su strada: il CP/M su Apple IIe
Costruiamoci un emulatore Z80



IC	TYPE NO	IC	TYPE NO
1	7805 -1	12	74LS00
2	8332	13	74LS04
3	2716	14	74LS05
4	2716 -1	15	74LS08
5	74LS279	16	74LS10
6	74LS157	17	74LS121
7	74LS157	18	74LS157
8	74LS157	19	74LS157
9	74LS157	20	74LS157
10	74LS157	21	74LS157
11	74LS157		



Jurassic News

Rivista aperiodica di
Retro-computing

Coordinatore editoriale

Tullio Nicolussi [tn]

Redazione

Sonicher [sc]

Hanno collaborato a questo numero:

Salvatore Macomer [sm]

Lorenzo 2 [L2]

Besdelsec [bs]

designer

Anna [an]

Sponsor

... per ora nessuno, ma se qualcuno si vuole fare avanti...

Diffusione

La pubblicazione viene distribuita in formato elettronico gratuitamente per i membri iscritti; i marchi citati sono di copyrights dei rispettivi proprietari.

Contatti

info@jurassicnews.com

Copyright

La riproduzione con qualsiasi mezzo di illustrazioni e di articoli pubblicati sulla rivista, nonché la loro traduzione, è riservata e non può avvenire senza espressa autorizzazione. E' consentita la diffusione a mezzo elettronico della rivista intera non modificata e in singolo file nel formato originale purché a titolo gratuito.

Sommario

Editoriale	3
Di qualche accadimento... [tn]	
Retro-Filosofia	4
Retro-computing, istruzioni per l'uso (parte 2). [tn]	
Le prove di J.N.	6
Sinclair ZX80 [sc]	
DIR	16
Il CP/M sui sistemi Apple II (parte 1). [sm]	
Emulazione	20
Extend2 per emulare lo ZX80 e ZX81 [L2]	
BBS	23
La posta dei lettori [sn]	
Emulazione	24
Costruiamoci un emulatore (parte 1) [sm]	
TAMC	30
Introduzione all'analisi algoritmica. [bs]	
Retro-Magazine	32
Re-Load Magazine. [sc]	
Retro-Magazine	33
Still-Alive. [sc]	
Retro-Code	34
AppleSoft Integer Basic (parte 1) [tn]	
Biblioteca	38
Digital Retro [sn]	

In copertina

Sinclair ZX80, uscito nel 1980 ha rappresentato una pietra miliare nella storia dell'home computing. Limitato nelle prestazioni dalla fobia del suo creatore per il contenimento dei costi ... a tutti i costi, è stato per molti il primo computer e per tutti la prima concreta speranza.

Jurassic News è una pubblicazione assolutamente "non schierata" da nessun punto di vista, né politico né tecnologico, né filosofico.

Si parlerà di tutti i computer, di tutti i programmi, senza preferenza alcuna.

Jurassic News è contraria a tutte le forme di copyright che limitano la libera circolazione delle idee in campo culturale ed informatico in particolare.

Di qualche accadimento...

I contatti si erano fatti frenetici, le mail non si contavano, qualcuno pure mi chiamava sul cellulare: -"Allora, si parte?"

"Certo!" la mia laconica risposta e infatti siamo partiti. Più o meno a mezzogiorno del 28 dicembre u.s. postavo uno stringato invito sul gruppo `it.comp.retrocomputer`: -"*C'è una nuova iniziativa editoriale dedicata al retrocomputing, se vi interessa...*". Non c'era molto altro da dire e poi io sono di poche parole...

Era il segnale convenuto, un quarto d'ora dopo il primo utente (che emozione...) si era registrato e fatto il download del primo numero. Il giorno dopo gli utenti erano 25 il 2 di gennaio siamo arrivati a 40. Insomma "è andata", ragazzi siamo nel giro dell'editoria specializzata! Così, tanto per esagerare :-)

Poi la lista delle cose da fare, mega riunione dopo Capodanno per gli auguri ma anche per rivedere il piano di lavoro.

L'intervento più urgente è apparso la messa a punto del sito Web. Il portale è abbozzato e le funzionalità principali ci sono, ma bisogna riordinare le idee, togliere i link non attivi, completare la presentazione delle rubriche, cominciare a metterci i contenuti. Senza contare che qualcuno si è accorto che con Internet Explorer i

CSS non vengono molto bene. IE? Ma qualcuno ancora lo usa? Bisognerà provvedere...

C'è sicuramente da lavorare molto sulla impaginazione; le prime segnalazioni arrivate riguardano proprio questo aspetto. Nessuno di noi è un grafico, bisognerà rimboccarsi le maniche ma senza pretendere di fare tutto e subito.

Poi ci sono le idee per nuovi articoli e nuove rubriche, l'idea di portare avanti il sito non come semplice supporto alla rivista ma come portale per servizi e per ospitare progetti "a latere".

Mi tocca frenare più di un entusiasmo; le nuove idee sono eccitanti ma poi bisogna avere la forza di portarle a compimento, quindi per favore calma e sangue freddo, facciamo un passettino alla volta per favore.

Poi c'è chi vorrebbe rivedere il piano di pubblicazione: -"... perché, sai, magari quelli del Commodore 64 potrebbero aversene a male...". E qualcuno punzecchia: -"Non vorrai mica mettere il 64 con lo Spectrum? Non c'è paragone!".

Mi sembra di essere ritornato a quei mitici giorni...

Vita da redazione, che bello!

[tn]

Jurassic News è una fanzine dedicata al retro-computing nella più ampia accezione del termine. Gli articoli trattano in generale dell'informatica a partire dai primi anni '80 e si spingono fino ...all'altro ieri.

La pubblicazione ha carattere puramente amatoriale e didattico, tutte le informazioni sono tratte da materiale originale dell'epoca o raccolte (e attentamente vagliate) da Internet.

Normalmente il materiale originale, anche se "jurassico" in termini informatici, non è privo di restrizioni di utilizzo, pertanto non sempre è possibile riportare per intero articoli, foto, schemi, listati, etc..., che non siano esplicitamente liberi da diritti.

La redazione e gli autori degli articoli non si assumono nessuna responsabilità in merito alla correttezza delle informazioni riportate o nei confronti di eventuali danni derivanti dall'applicazione di quanto appreso sulla rivista.

Retro-Filosofia

Retro-Filosofia è la rubrica che si occupa degli aspetti meno tecnici della rivista. Ospita pezzi di opinione e di storia.

Il fenomeno del Retro-Computing è partito circa dieci anni or sono in corrispondenza della diffusione dei PC di tipo "IBM" nelle case.

Retro-Computing, istruzioni per l'uso (parte 2)

Mettiamo che sono stato fortunato e che lo zio della mia ragazza mi abbia mollato quel vecchio home che stava in soffitta. Ok, mi sono dovuto sorbire tutte le menate su "...come era bello una volta che non esisteva il PC..." e anche "...allora si che era una sfida programmare, altro che adesso, che ti basta tirare col mouse quattro componenti preconfezionati su una maschera vuota e hai belle e fatto il programma...".

Tornando al momento fatale si tratta ora di poterla usare la "vecchia macchina dei ricordi". Sperando che sia ancora in ordine e non manchi qualche pezzo fondamentale (ad esempio l'alimentatore), conviene come primo passo aprirla e darci dentro di aria compressa per eliminare i quintali di polvere che inevitabilmente il tempo avrà accumulato. Questa operazione di sommaria pulizia ha anche lo scopo di dare un'occhiatina all'interno per assicurarsi che tutto sia in ordine prima di dare tensione e per verificare che i topi non ci abbiano magari fatto il nido dentro (succede, ve l'assicuro).

I casi in genere sono due: o si è già posseduto quel modello o uno simile, oppure è del tutto sconosciuto. As-

sodato che nel primo caso non ci saranno eccessivi problemi nel verificare il funzionamento del tutto, nella seconda ipotesi le cose non sono troppo semplici. Conviene frenare l'entusiasmo prima di collegare alla belle meglio cavi e cavetti e prendersi il tempo per raccogliere tutte le informazioni possibili prima di fare guai irreparabili.

Questo è maggiormente vero se la macchina è del tipo "raro" o magari non è nemmeno un home computer nel senso letterale del termine, ma magari un sistema Unix di classe mini, che richiede una certa conoscenza.

Internet è la fonte primaria di informazioni per il retro computerista. La semplice ricerca brutale con Google fornisce di norma qualche decina di siti da visitare. Il consiglio che sento di dare è di non limitarsi ai primi risultati, qualche volta le cose migliori si trovano nei link segnalati molto in basso. Quello che stiamo cercando infatti potrebbe essere una pagina marginale in un sito personale. Ci sono parecchi personaggi che pubblicano notizie utili su vecchi calcolatori posseduti, ma spesso enfatizzano altri interessi, magari più mo-

(Continua a pagina 5)

Da dove comincio?

Secondo me è sbagliato pensare che un buon collezionista sia colui che possiede una sterminata collezione o una grande varietà di pezzi. Il collezionismo è un fenomeno soprattutto privato, nel senso che ognuno privilegia delle proprie idee e passioni, anche senza che siano direttamente a confronto con altre. La strada che io suggerisco è quella della specializzazione: è più facile e può dare in poco tempo notevoli soddisfazioni. Si comincia magari con un sistema trovato in soffitta e si cerca di procurarsene un secondo o un modello leggermente diverso. Così ci si specializza su una particolare marca o su un particolare modello e ciò che ne risulta è una piccola collezione ma vicina all'obiettivo della completezza, che è quello a cui tendono in fondo tutti i collezionisti.

dermi, che in qualche modo ingannano i motori di ricerca che considerano poco rilevante la presenza dell'informazione retro-computeristica nel sito.

Una vera miniera di informazioni e prezioso aiuto ci viene dai newsgroup. In Italia ne sono disponibili due (almeno a quanto mi risulta) che sono: `it.comp.retrocomputer` e un forum disponibile su `www.retro-it.org`. Quest'ultimo accessibile dopo la consueta registrazione gratuita. Purtroppo la pecca del newsgroup `icr` (Italia Comp Retrocomputer) è il non essere moderato. Abbonda pertanto di flame anche abbastanza pesantini qualche volta e del consueto spam onnipresente sulla rete. Pazienza, sono dei prezzi da pagare per frequentare comunque l'unico punto informativo italiano dove più o meno bazzicano tutti gli appassionati di questo hobby.

Per chi non ha problemi con l'inglese sono altrettanto buoni i gruppi internazionali corrispondenti.

Anche i gruppi mono-marca sono da frequentare. Spesso chi li frequenta conosce un po' tutta la produzione di un certo sistema e può ricordarsi parecchi particolari utili. L'interazione con questa categoria di appassionati "verticali" è solitamente molto soddisfacente: chi ama una certa marca di computer, ad esempio Apple, si sente coinvolto in prima persona a promuoverne l'immagine in qualunque modo.

Insomma per concludere l'hobby del retrocomputing può essere affascinante, poco costoso e colmo di piacevoli sorprese ma richiede doti di pazienza sia per l'appassionato che dovrà perdere parecchio tempo a cercare documentazione, contatti o qualche pezzo di ricambio e sia della compagna nella vita che dovrà tollerare un po' di caos in garage o magari in salotto.

[tn]

Il cimitero degli elefanti.

Un giorno un mio amico mi ha fatto questo affascinante e divertente parallelo fra il mito famoso del mai scoperto cimitero dove i dentoluti preziosi pachidermi andrebbero ad ammassarsi spontaneamente sentendo avvicinarsi la fine. I calcoli sulle vendite di vecchio hardware suggerirebbero l'esistenza di detti depositi ma la realtà sappiamo che è purtroppo diversa. Le vecchie macchine sono state sparse in mille rivoli, oggetto di regali natalizi a bambini magari niente interessati al calcolatore di per se ma solo al gioco visto in casa di qualche compagnuccio. Qualche volta saltano improvvisamente fuori dei piccoli ammassi, sono le scuole che dimenttono le da sempre inusate aule informatiche cogliendo l'occasione di qualche docente volenteroso che propone l'attività integrativa (e ovviamente ci vuole tutto nuovo) o i magazzini delle ditte di riparazione che in qualche caso hanno conservato del materiale.

E' capitata anche a me l'occasione chiamata "cassonetto", cioè il trovare un vecchio hardware appoggiato a fianco del cassonetto o consegnato nelle isole ecologiche per la raccolta differenziata. Farsi amico uno degli addetti al riciclaggio assicura una buona rendita di sistemi anche se ormai si tratta per lo più di Piccioni male in arnese, brutti a vedersi e di nessun valore affettivo (come si fa ad amare un clone IBM?).

Altro giacimento sono le università dove l'hardware è sempre abbondato. Purtroppo per la maggior parte ce chi ci ha già messo gli occhi sopra e succede che ne faccia addirittura commercio. Attività molto al limite della legalità, trattandosi di beni dello stato che per definizione sono inalienabili.

Al limite si può arrivare all'acquisto, opzione non particolarmente gradita ai retrocomputeristi che preferirebbero "a gratis" o al limite con scambio di materiale. Ogni tanto su e-bay qualche pezzo viene battuto, ma generalmente con valore superiore a quanto personalmente sono disposto a spendere. Poi ci sono i mercatini. Fantastici quelli rionali spontanei dove qualche buon affare è possibile, raro ma possibile.

Le prove di J.N.

sinclair
ZX80

Sinclair ZX80

Le prove di Jurassic News ospita recensioni di hardware e software con l'intento di fornire le prime indispensabili informazioni per coloro che sono entrati in possesso di uno di questi sistemi e non hanno documentazione e magari nessuna idea di cosa fare per vederlo almeno in funzione

Nel gennaio del 1980 successe una cosa destinata a cambiare per sempre il giovane mondo dell'home computing. Stiamo parlando del lancio in Inghilterra dello ZX80 che, con le sue 99,99 sterline di prezzo rompeva un tabù fino a quel momento impensabile: un computer alla portata di tutte le tasche. Il signor Clive Sinclair si era battuto strenuamente per ottenere questo risultato, fino al punto di far uscire una macchina con parecchi problemi di funzionamento, ma che importava? Chi ha vissuto la magia di attaccare il piccolo giocattolino al televisore non lo dimenticherà in tutta la sua vita.

Il contesto storico in cui si

cala la prima macchina di Sinclair è un mondo dove i sistemi personali di calcolo sono dedicati ai soli

hobbisti e hanno un costo elevato. Questo brand deriva essenzialmente da due fattori: il calcolatore viene derivato dal mondo dell'elettronica e quindi viene avvicinato principalmente da una certa classe di persone abituate a considerare costoso il loro hobby e la scarsa diffusione iniziale delle macchine con conseguente scarsa automazione del processo produttivo.

Una cosa curiosa è che lo ZX80 viene venduto anche in kit al prezzo in Inghilterra di 79,99 sterline (circa 300 mila lire). L' -



autocostruzione è tutt'altro che complicata con i suoi 20

(Continua a pagina 7)

Come vengono scelti i sistemi per le recensioni.

E' una domanda che ci è stata fatta e alla quale rispondiamo volentieri: ebbene, non c'è un disegno definito e non sarà seguito nemmeno un criterio temporale. Verranno presi in considerazione i sistemi per i quali sia reperibile una sufficiente documentazione, meglio se posseduti direttamente dai recensori, senza particolari schemi precostituiti. Il criterio della sequenzialità temporale sarebbe una facile scelta ma condannerebbe la rivista a ripetere pari pari la storia delle macchine di calcolo personali, cosa che non si vuole fare anche per avere le mani completamente libere e poter presentare un sistema indipendentemente dalla data di uscita permettendosi di "dimenticare", o meglio "rimandare" la presentazione di un qualche altro pezzo, anche se per sua natura fondamentale per la corretta interpretazione della storia.

integrati e la cinquantina di componenti passivi. In Italia verrà importato dalla GBC (un gigante nella distribuzione dell'elettronica consumer, fallito dopo pochi anni, non si è mai capito per quale motivo, anche se il sottoscritto un'idea ce l'ha), ma solo in dicembre dell'anno 1980, al prezzo di 325.000 Lire IVA compresa, il kit a 275.000. Tutto sommato prezzi accettabili rispetto agli originali, considerando che il ricarico su queste apparecchiature provenienti dagli States è normalmente superiore al 30% e in qualche caso arriva a raddoppiare!

Impressioni iniziali.

Leggerissimo e "plasticoso", con quella tastiera disegnata sembra proprio un giocattolo e nient'altro!

Il modestissimo peso, raggiunto anche grazie all'adozione di un case di plastica dello spessore di appena 1 mm, è la caratteristica che colpisce al primo impatto.

Una vistosa scritta nero/oro "Sinclair ZX80" anima la parte superiore del contenitore, un altro inserto elenca le cosiddette in maniera pomposa "Integral Functions" che altro non sono che le poche funzioni cablate nell'interprete. Questa specie di memo non è presente nelle prime versioni della macchina. Coerentemente con il loro uso la funzione POKE è associata ad un tasto in modalità Key, mentre la corrispon-

dente PEEK, che preleva un byte dalla memoria è raggiungibile solo digitandola esplicitamente e fa parte di queste funzioni "integrali", qualsiasi cosa voglia dire questa definizione.

Il primo problema da superare è individuare il connettore di alimentazione. Nulla di difficile, ma per essere tranquilli si deve cercare la serigrafia stampata sotto la macchina. L'altro problema è sincronizzare correttamente la TV, problema peraltro comune a tutte le macchine dell'epoca.

L'uscita video in UHF è sul canale 36, un classico.

L'interfaccia del registratore a cassette non è proprio il massimo della tolleranza in fatto di segnali. A meno di non essere particolarmente fortunati, è necessario provvedere ad una lunga fase di taratura dei livelli di registrazione per essere sicuri di poter leggere poi senza difficoltà. La velocità (250 boud) non è particolarmente elevata ma comunque adeguata alla classe del sistema.

Accensione.

Inutile dire che non è presente alcun tasto di accensione (sarebbe costato

(Continua a pagina 8)

Il signor Clive Sinclair si era battuto strenuamente per ottenere un sistema commerciabile a meno di 100 sterline, fino al punto di far uscire una macchina con parecchi problemi di funzionamento.

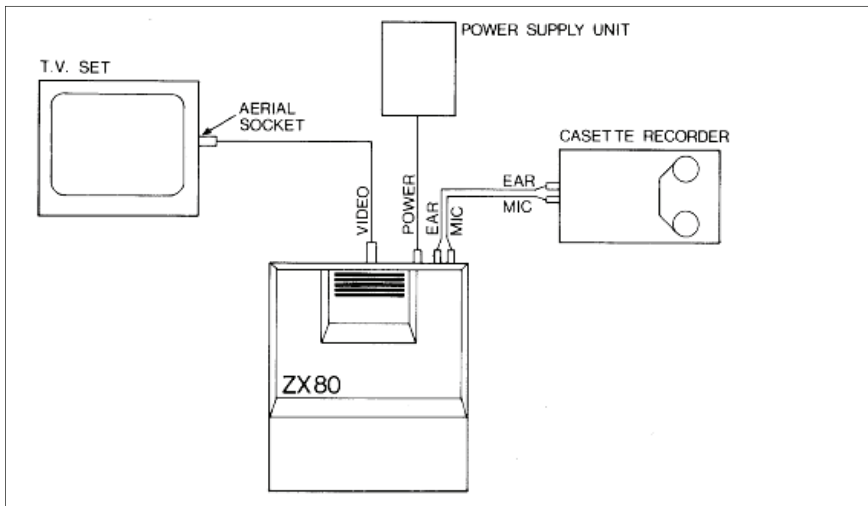
Il successo del computer dimostra che i tempi erano maturi e che l'intuizione del suo creatore era corretta.



Due versioni dello ZX80 a confronto

Lo ZX80 su Web.

Cominciando a raccogliere materiale per questo articolo ho fatto una ricerca sul Web senza aspettarmi troppo. Il sistema è infatti molto vecchio e la stessa diffusione (si parla di circa 300.000 esemplari venduti) non farebbe supporre grandi scoperte. Invece ancora una volta la rete si rivela la madre di tutte le informazioni: non siamo al livello di club come lo Spectrum o il Commodore 64 o l'Apple IIe, ma anche il piccolo ZX80 ha la sua schiera di appassionati e collezionisti. Il fatto che inoltre sia così semplice da riparare ed emulare lo rende un piccolo sistema ideale per chi si interessa in maniera attiva della tecnologia primordiale dei sistemi di calcolo personale. Quindi se volete saperne di più... Ask a Google...



Lo schema dei collegamenti è semplice e alla portata di tutti.

troppo). Appena alimentato il computer funziona facendo partire un monitor di boot che mette a disposizione l'ambiente Basic in ROM caratterizzato da un lettera "K" in campo inverso.

La "K" iniziale sta per "Key mode", il modo di funzionamento che predispone l'accettazione di un comando o di uno statement Basic pigiando un singolo tasto sulla tastiera. Ogni tasto infatti riporta tre modi di funzionamento: il modo "Key" che corrisponde alla scritta gialla appena sopra il tasto (ad esempio il tasto "A" fa

apparire il comando "LIST"), il simbolo normale del tasto (le lettere sono solo maiuscole) e un ulteriore simbolo "semigrafico" raggiungibile con lo SHIFT.

Lo ZX80 è fondamentalmente una calcolatrice programmabile (anzi, anche di meno) senza alcuna concessione ad aspetti grafici come ad esempio colore o animazioni. Lo stesso Basic è proprio minimale, senza capacità in virgola mobile. E' una macchina per imparare la programmazione e i fondamenti dell'informatica e come tale deve essere trattata e valutata.

Hardware.

Lo ZX80 è realizzato con un'unica piastra dove trova posto il circuito stampato e la matrice della tastiera. La tastiera è poi ricoperta da una speciale plastica conduttrice che permette, quando il tasto è pigiato, di chiudere il relativo contatto. Il cabinet è costituito da una base piatta e da una copertura sagomata in maniera tale da racchiudere sia gli integrati che il modulatore video. Una griglia di aerazione ricavata proprio sopra la copertura del modula-



Praticamente obbligatorio acquistare l'espansione di RAM dopo i primi semplici esperimenti in Basic.



tore, permette lo smaltimento di calore (che comunque non è rilevante, stante la scarsa presenza di componenti). L'unico "punto caldo" è l'integrato regolatore di tensione, un comune 7805, che forse è dotato di un dissipatore fin troppo minuscolo e appoggiato alla piastra cosa che ne provoca a lungo andare la "cottura"; e vai con il risparmio! Forse si è pensato che comunque non si tratta di un sistema che rimarrà acceso per molte ore consecutive.

Sul retro i connettori: alimentazione, registratore in/out, modulatore TV e pettine di espansione. Quest'ultimo è una semplice terminazione della piastrina madre che espone i segnali del processore senza particolari accorgimenti o protezioni. Inutile dire che questo rende particolarmente esposto proprio il cuore del sistema, cioè la CPU con la sola protezione di una chiave di inserimento del pettine che le periferiche di terze parti sono invitate a rispettare.

L'alimentatore esterno fornisce 9 V in cc che vengono stabilizzati a 5 Volt internamente dall'integrato 7805 appena menzionato.

Tutti gli osservatori specializzati, all'epoca dell'uscita sul mercato, hanno rilevato l'estrema economicità nella costruzione della macchina. E' ben vero che per una volta il risparmio viene ridistribuito al compratore che può finalmente permettersi un personal computer ad un prezzo più che abbordabile, però forse è stato superato un limite di decenza considerando che l'oggetto che

uno si porta a casa come computer vale francamente pochissimo.

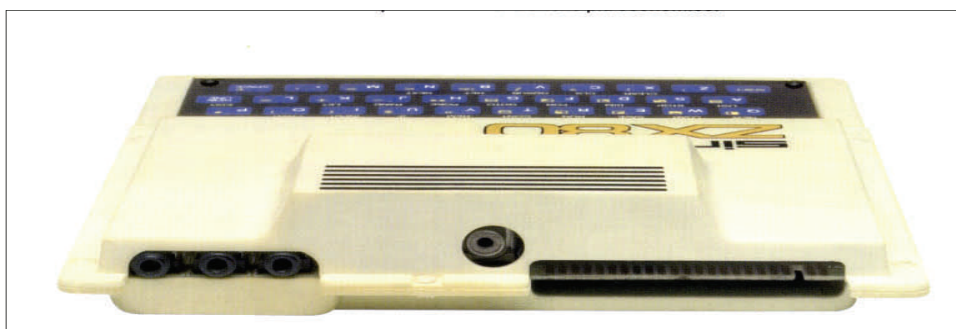
Una cosa è comunque certa: lo ZX80 ha dimostrato la potenza dei microprocessori nel senso che per la loro natura programmabile sono adatti a simulare qualsiasi schema di segnali che i progettisti intendono implementare. Uno dei difetti più fastidiosi dello ZX80 e cioè la perdita del sincronismo di quadro durante la digitazione e l'oscuramento del video al lancio del programma, derivano proprio dall'economicità di costruzione e dalla pretesa di adibire la CPU anche a generare i segnali video.

La CPU è la versione siglata "Z80A" del microprocessore della Zilog, che può arrivare a 4 MHz. Nello ZX80 è stata usata una versione della NEC siglata 780C-1 a 3,5 MHz. E' significativo il fatto che Sinclair abbia scelto una CPU con clock elevato per l'epoca, piuttosto che la versione base, nonostante

(Continua a pagina 10)



Sotto il vestito... quasi niente!



Il retro della macchina. Da sinistra: i due jack del registratore, l'alimentatore, l'uscita del modulatore video e il pettine di espansione.

```
100 FOR T=1 TO 1000
110 PRINT T
120 NEXT T
```

Un output di esempio. Si noti la buona definizione dei caratteri.

Ad onta delle clamorose limitazioni del sistema nel suo complesso, l'utilizzo può essere di molta soddisfazione.

la sua propensione al risparmio. Significa che un clock a 1 MHz non ce l'avrebbe fatta a gestire il video. La gestione della schermata video non è basata infatti sulla classica

mappatura in RAM della memoria video, ma è la CPU stessa che provvede a generare il segnale digitale da passare al modulatore per la conversione in analogico. Per fare un paragone si potrebbe dire che il video è trattato come una periferica seriale, ad esempio una stampante.

Il modulatore video è un classico ASTEC e qui forse è l'unico punto dove Sinclair non ha potuto risparmiare più di tanto essendo un componente standard e la ditta costruttrice praticamente monopolista. Ne deriva una ottima chiarezza e stabilità del video soprattutto per i caratteri in chiaro; per quelli inversi (bianco su nero) la qualità cala parecchio.

La RAM è limitata a 1 Kb (sì, avete letto bene) e la ROM a 4 Kb. Si può acquistare a parte (45.000 Lire in Italia) una espansione di memoria RAM da inserire nel pettine di espansione. Ogni Kb aggiuntivo costa poi sulle 20.000 Lire (il modulo di espansione inizialmente ne ospita solo 1 Kb).

Il clock del sistema non è stabilizzato da un quarzo (sarebbe costato troppo) ma è costituito da un oscillatore capacitivo che inevitabilmente è soggetto a qualche deriva durante il funzionamento. Nulla di grave, almeno finché rimane nei limiti di tolleranza,

non pensate però di poter programmare dei timer precisi al microsecondo!

La ROM contiene il monitor, la mappa dei caratteri e il Basic. Le possibilità di espansione del sistema, nonostante la ridotta dotazione iniziale è comunque intatta. Poco tempo dopo uscirà una ROM da 8 Kb con l'interprete in virgola mobile e sono commercializzate espansioni di RAM da collegare al pettine di espansione. Un Kb di RAM è comunque sufficiente per i primi esperimenti di programmazione anche grazie all'ottimizzazione del codice che viene in qualche misura "pre-compilato", cioè viene conservato un solo byte per individuare l'istruzione, idea che tutti i progettisti d'ora in poi adotteranno.

L'output prevede 24 linee da 32 caratteri ciascuna in bianco/nero. E' previsto l'uso dei soli simboli maiuscoli per le lettere, il campo inverso (bianco su nero) e una decina o poco più di simboli semigrafici.

Il connettore di espansione (gli inglesi lo chiamano "Edge Connector") a 44 poli porta i 37 segnali della CPU, le alimentazioni (0, 5 e 9 Volt), il clock, un segnale ad uso della memoria esterna e due pin di terra.

Utilizzo.

Ad onta delle clamorose limitazioni del sistema nel suo complesso, l'utilizzo può essere di molta soddisfazione. Bisogna pensare che chi lo acquista ha poca o nulla familiarità con l'informatica e di programmazione ne ha fatta sulle calcolatrici programmabili o, per i più fortunati, su qualche calcolatore all'università.

L'idea di associare il modo "Key" ad un solo tasto mitiga la scomodità della

(Continua a pagina 11)

tastiera ed ha anche un effetto educativo: tutte le funzioni sono immediatamente disponibili e individuabili sulla tastiera.

L'input primario è la digitazione diretta del programma che inizia scegliendo un tasto con la corrispondente parola chiave che apparirà sullo schermo e il cursore, diventato ora un "L" in campo inverso, pronto ad accettare un "literal", cioè qualsiasi lettera presente sulla tastiera.

Una caratteristica notevole per l'epoca è il controllo di sintassi al momento della digitazione. A parte l'associazione delle parole chiave ai tasti (modalità "K"), al momento della conferma il sistema controlla la sintassi dell'istruzione e, in caso di errore, posiziona un cursore ("S" in campo inverso) nel punto dell'errore. E' praticamente impossibile inserire una riga errata, un grosso aiuto per chi si avvicina alla programmazione!

L'uso della tastiera dello ZX80 è stato sempre oggetto di polemiche. La soluzione adottata dai progettisti per i contatti è poco affidabile. In pratica ci si accorge che il tasto "a preso" perché lo Z80 perde il sincronismo di quadro andando a servire la routine di controllo del tasto digitato con un evidente sfarfallio del video.

La disposizione dei tasti è una quasi standard QWERTY con una disposizione meno standard per quanto riguarda i simboli di interpunzione. La barra dello spazio non esiste proprio e bisogna arrangiarsi con un tastino di dimensioni normali all'estrema destra della riga bassa di tasti. Il return o invio qui si chiama "NEW LINE". C'è un solo tasto SHIFT sulla sinistra, ma è più che sufficiente, anche perché l'occasione di raggiungere il secondo simbolo dei tasti

(un carattere semigrafico) è abbastanza limitata.

Il tasto siglato "RUBOUT" permette la ripetizione del tasto digitato.

La disposizione delle parole chiave sui tasti segue una logica legata al "miglior utilizzo". La funzione "LIST" è associata al tasto "A", mentre il tasto "L" è rimasto libero e si è preferito ad esempio mettere la parola chiave "LET" sul tasto "K", strano.

La riga superiore di tasti ospita le dieci cifre decimali con associate alcune parole chiave e i tasti freccia (utili in modalità editor). Queste funzioni sono colorate in giallo, così come il tasto EDIT, nella versione arrivata in Italia, mentre originalmente erano tutte bianche.

Il Basic dello ZX80.

L'esecuzione del programma avviene con la classica RUN, il BREAK serve per interrompere e CONT per continuare.

Per caricare un programma da cassetta si usa LOAD e per salvare SAVE. La gestione del nastro è rudimentale: non si può associare un nome al programma salvato cosicché è necessario posizionare "a mano" il nastro.



Visto dall'alto la disposizione dei tasti è ben visibile.



Non manca la documentazione raccolta in un pratico blocco spiraleto.

(Continua a pagina 12)



Sir Clive Sinclair mostra orgogliosamente la sua "creatura".

stro prima di caricare. Il salvataggio e il caricamento da nastro è visto come un vero e proprio dump della memoria piuttosto che come salvataggio del solo codice del programma. Quindi un caricamento distrugge il contenuto iniziale della RAM e lo sostituisce con quello letto dal nastro.

I progettisti non hanno dimenticato gli "smanettoni" prevedendo le istruzioni **POKE** e **PEEK**, rispettivamente per il push e il pop di byte da e per locazioni di memoria che dovranno essere indicate in decimale. Altra funzione "avanzata" è la **USR** che permette l'esecuzione di una routine utente in linguaggio macchina.

Le variabili numeriche sono solo intere nel range degli 8 bit (da -32768 a 32767) mentre le alfanumeriche possono avere qualsiasi dimensione. È obbligatorio l'uso della funzione di assegnazione **LET** mentre alcune funzioni built-in sono state dedicate alle stringhe: **CHR\$** ad esempio converte un numero nel corrispondente carattere ASCII mentre l'inversa si chiama **CODE**; **TL\$** permette di estrarre il primo carattere di una stringa.

Inutile dire che mancano le funzioni trascendenti (seno, coseno, etc...) che sarebbero inutili nell'impossibilità di assegnarne il risultato ad una variabile di tipo real.

Una comodità dell'editor è il controllo di sintassi al momento dell'invio. A fronte di un errore il cursore diventa una "S" in campo inverso e si può agire per effettuare la correzione del caso.

Una curiosità dell'interprete è la possibilità di effettuare i salti **GOTO** con un valore di riga calcolato, tipo **GOTO A + B**.

Internamente le istruzioni sono conservate in forma "pre-compilata", rendendo il misero Kb di RAM almeno accettabile per i primi esperimenti. Il Basic non è privo di bachi, uno di questi è l'andare in crash quando si tenta di gestire stringhe superiori a 400 caratteri circa.

Considerando l'epoca, il costo, la novità e quant'altro possa venire in mente, questi difettucci sono ampiamente tollerabili.

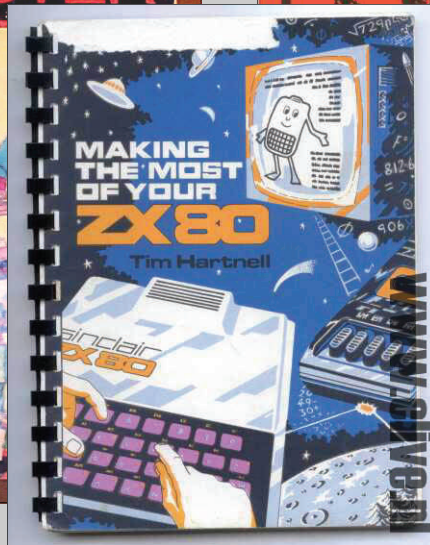
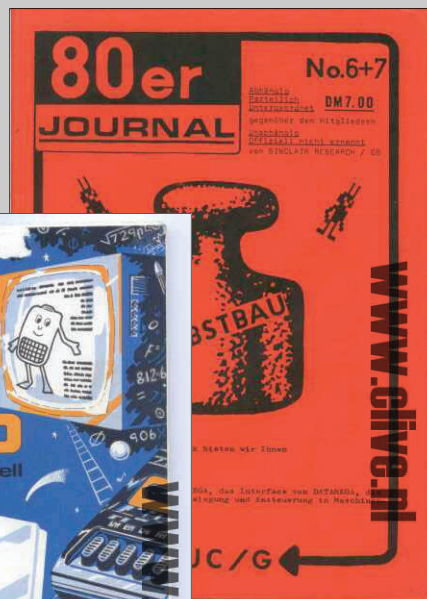
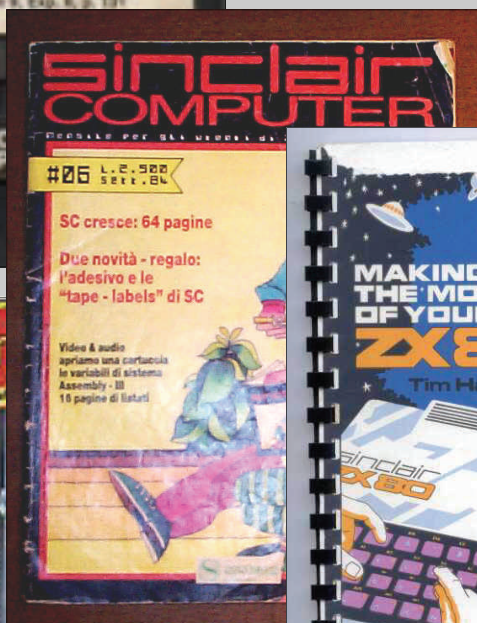
Conclusioni.

Lo ZX80, come si diceva all'inizio, ha segnato un punto di svolta nella politica di produzione dei calcolatori personali: il prezzo. Dal momento della sua introduzione chiunque si sia affacciato sul mercato ha dovuto tenerne conto per cercare di non far pagare troppo le funzionalità del proprio sistema. Da questa logica si è salvata solo la Apple Computer che proprio di guerra di prezzi non ne ha mai voluto saperne.

Le limitate possibilità di questo primo prodotto di quello che fra qualche anno sarà nominato baronetto e potrà fregiarsi del titolo di "Sir", Clive Sinclair sono state poi la base per le successive realizzazioni introdotte con sapiente maestria commerciale che raggiungeranno il l'apice con lo Spectrum.

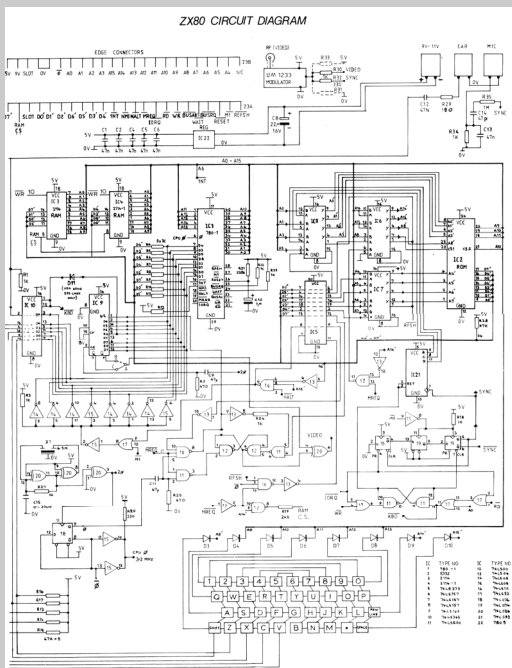
Poi l'inevitabile declino nonostante il lancio del QL, per molti versi anch'essa una macchina molto innovativa.

[sn]



Software, riviste e libri seguono a ruota.

L'Italia arriva in ritardo, ma arriva! Qui sotto la copertina di Bit numero 13 del dicembre 1980



Ma anche la documentazione tecnica è facilmente reperibile fin da subito. Una vera gioia per gli appassionati!



UK 10 — ekkor June 1980

advertisement

advertisement

ekkor June 1980 — UK 11

Britain's first complete computer kit.

A complete personal computer for a third of the price of a bare board.

Also available ready assembled for £99.95

The Sinclair ZX80.

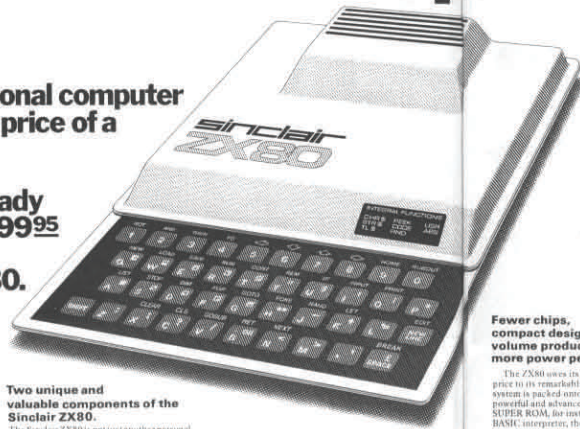
Until now, building your own computer could easily cost around £300... and still leave you with only a bare board for your trouble. The Sinclair ZX80 changes all that. For just £79.95 you get everything you need to build a personal computer at home. PCB, with IC sockets for all ICs, case, leads for direct connection to your own cassette recorder and television, everything!

And yet the ZX80 really is a complete, powerful, full-featured computer, matching or surpassing other personal computers on the market at several times the price. The ZX80 is programmed in BASIC, and you could use it to do quite literally anything from playing chess to running a power station.

The ZX80 is pleasantly straightforward to assemble, using a fine-tipped soldering iron. Once assembled, it immediately proves what a good job you've done. Connect it to your TV set... link it to an appropriate power source* and you're ready to go.

- Your ZX80 kit contains...**
- Printed circuit board, with IC sockets for all ICs.
 - Complete components set, including all ICs — all manufactured by selected world-leading suppliers.
 - New rugged Sinclair keyboard, touch-sensitive, wipe-clean.
 - Ready-moulded case.
 - Leads and plugs for connection to any portable cassette recorder (to store programs) and domestic TV (to act as VDU).
 - FREE course in BASIC programming and user manual.
- Optional extras**
- Mains adaptor of 600 mA at 9 V DC nominal unregulated (available separately — see coupon).
 - Additional memory expansion board (plugs in to take up to 1K bytes extra RAM chips. Chips also available — see coupon).

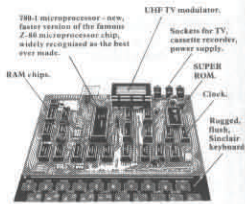
*Use a 600 mA at 9 V DC nominal unregulated mains adaptor. Available from New Line at £3.99 — see coupon.



Two unique and valuable components of the Sinclair ZX80.

- The Sinclair ZX80 is not just another personal computer. Quite apart from its exceptionally low price, the ZX80 has two uniquely advanced components: the Sinclair BASIC interpreter, and the Sinclair teach-yourself BASIC manual.
- The unique Sinclair BASIC interpreter... offers remarkable programming advantages.
- Don't use 'wordy' key word syntax: the ZX80 eliminates a great deal of tiresome typing. Key words RUN, PRINT, LIST, etc. have their own single-key syntax.
- Unique syntax check. Only lines with correct syntax are accepted into programs. A cursor identifies errors immediately. This prevents entry of long and complicated programs with faults only discovered when you run them.
- Excellent string-handling capability — takes up to 26 string variables of any length. All strings can undergo all relational tests (e.g. comparison). The ZX80 also has string inputs to request a line of text when necessary. Strings do not need to be dimensioned.
- Up to 26 single dimension arrays.
- POKE/NEXT loops nested up to 26.
- Integer names of any length.
- BASIC language also handles full Boolean arithmetic, conditional expressions, etc.
- Exceptionally powerful edit facilities, allows modification of existing program lines.
- Randomise function, useful for games and secret codes, as well as more serious applications.
- Timer under program control.
- PEEK and POKE enable entry of machine code instructions. ROM causes jump to a user's machine language sub-routine.

- High-resolution graphics with 22 standard graphic symbols.
 - All characters printable in reverse under program control.
 - ... and the Sinclair teach-yourself BASIC manual.
- If the features of the Sinclair interpreter listed alongside mean little to you — don't worry. They're all explained in the specially-written 96-page book (for with every kit). The book makes learning easy, exciting and enjoyable, and represents a complete course in BASIC programming — from first principles to complex programs. (Available separately — purchase price refunded if you buy a ZX80 later.)



£79.95
Including VAT
Including post and packing.
Including all leads and components

Fewer chips, compact design, volume production — more power per pound!

The ZX80 uses its remarkable low price to its remarkable design: the whole system is packed onto fewer, newer, more powerful and advanced LSI chips. A single SUPER ROM, for instance, contains the BASIC interpreter, the character set, operating system, and monitor. And the ZX80's 1K byte RAM is roughly equivalent to 4K bytes in a conventional computer, because the ZX80's brilliant design packs the RAM so much more tightly (key words, for instance, occupy just a single byte).

To all that, add volume production — and you've got a price breakthrough that really is a breakthrough.

The Sinclair ZX80. Kit: £79.95. Assembled: £99.95. Complete!

The ZX80 kit costs a mere £79.95. Can't wait to have a ZX80 up and running? No problem! It's also available, ready assembled, for only £99.95.

Whether you choose the kit or the ready-made, you can be sure of world-famous Sinclair technology — and years of satisfying use. Science of Cambridge Ltd is one of the Sinclair companies owned and run by Clive Sinclair.

To order, complete the coupon, and post to Science of Cambridge for delivery within 28 days. Return as received within 14 days for full money refund if not completely satisfied.

sinclair ZX80
Science of Cambridge Ltd
6 Kings Parade, Cambridge, Cambs., CB2 1SN.
Tel: 0223 314485.

Order Form

To: Science of Cambridge Ltd, 6 Kings Parade, Cambridge, Cambs., CB2 1SN.
Remember: all prices shown include VAT, postage and packing. No hidden extras.

Please send me:

Quantity	Item	Item price	Total
	Sinclair ZX80 Personal Computer kit's. Price includes ZX80 BASIC manual, excludes mains adaptor.	79.95	
	Ready assembled Sinclair ZX80 Personal Computer's. Price includes ZX80 BASIC manual, excludes mains adaptor.	99.95	
	Mains Adaptor — 600 mA at 9 V DC nominal unregulated.	8.95	
	Memory Expansion Board's — takes up to 1K bytes.	12.00	
	RAM Memory chips — standard 1K bytes capacity.	16.00	
	Sinclair ZX80 Manual's — manual free with every ZX80 kit or ready-made computer.	5.00	
	TOTAL		£

Note: Your Sinclair ZX80 may qualify as a business expense.

I've lost a cheque/postal order payable to Science of Cambridge Ltd for £

Please print Name: Mr/Ms/Miss

Address

ES/80

L'annuncio della disponibilità del computer ZX80 in Kit apparso sulle riviste specializzate inglesi nel gennaio del 1980.

Questa foto permette di apprezzare le dimensioni, davvero minuscole della macchina.



Come il Basic compila le istruzioni.

Nell'esempio sotto riportato si vede come vari statement occupino molta meno memoria di quanto sarebbe richiesto per una conservazione "piatta" del codice.

10 FOR A=16424 TO 17424	18 Bytes
20 PRINT PEEK(A);	12 Bytes
30 NEXT A	5 Bytes
40 STOP	4 Bytes

Questo accorgimento permette di scrivere programmi lunghi fino a circa 300 righe di codice con la dotazione iniziale di 1 Kb di RAM.

Emulare, che passione!

Esistono emulatori di ZX80 praticamente per qualsiasi piattaforma. Infatti lo Z80 è forse il processore a 8 bit più emulato in assoluto grazie anche alla sua sostanziale semplicità nel funzionamento.

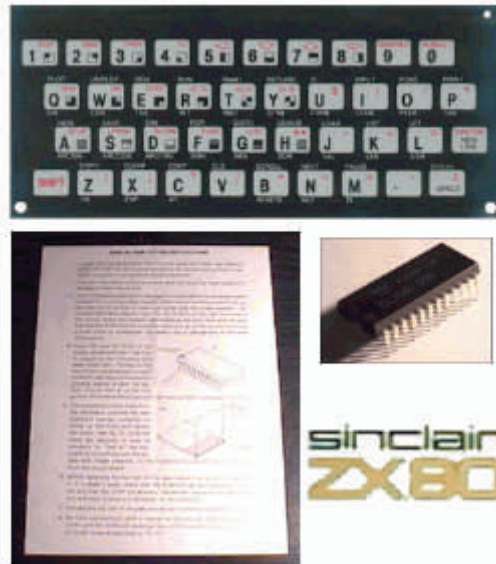
E' praticamente inutile elencarli in questa sede, visto che una semplice ricerca in Google ne restituirà più di quanti se ne possa visitare in una settimana. Rimandiamo quindi all'articolo sugli emulatori presente in questo volume che si occupa di EXTEND2, che per nostra opinione è una delle implementazioni meglio riuscite.

8K BASIC ROM



The 8K BASIC ROM used in the ZX81 is available to ZX80 owners as a drop-in replacement chip. With the exception of animated graphics, all the advanced features of the ZX81 are now available on a ZX80—including the ability to run much of the Sinclair ZX Software.

The ROM chip comes with a new keyboard template, which can be overlaid on the existing keyboard in minutes.



La ROM da 4K con l'interprete Basic in aritmetica intera fu presto sostituito da un upgrade che raddoppiando la capacità della ROM (8K) consentiva l'aritmetica floating point e in generale i miglioramenti resi disponibili dallo ZX81.

Bibliografia.

Personal Computer World, April 1980

Bit n. 13 , Dicembre 1980

Digital Retro, Mondadori, 2005

DIR

DIR è la rubrica
dedicata al sistema
operativo CP/M

Il CP/M sui sistemi Apple II (parte 1).

Una delle implementazioni più conosciute e diffuse del sistema operativo CP/M è quella realizzata dalla Microsoft per i sistemi della serie II di Apple Computer.

La soluzione prevedeva una scheda hardware (come poteva essere altrimenti, visto che l'Apple funzionava con CPU 6205 sostanzialmente molto distante dalla logica dello Z80), l'uso obbligatorio del floppy e il relativo porting del codice per adattarlo al sistema ospite.

Nella soluzione ibrida l'Apple II diventa una macchina bi-processore che in qualche maniera riescono, nelle soluzioni più ardite, a collaborare nell'esecuzione del codice, nel senso che è possibile richiamare dai programmi CP/M delle routine in assembler 6205 che saranno eseguite evidentemente dal processore nativo.

La scheda Z80 per Apple II è relativamente poco rara anche in Italia, merito anche di un costo non eccessivo e della disponibilità di software di qualità che ne rendono l'acquisto appetibile a molti. Adottando questa soluzione si apre all'utente Apple un mondo vasto di software: linguaggi di programmazione professionali come il COBOL e

software di produttività individuale come il mitico WordStar, il word processor che ha dettato legge per dieci anni nell'informatica individuale, e ancora DBII, il "papà" dei moderni RDBMS.

Questa serie di articoli è un viaggio in questo mondo "alternativo" rispetto al nativo 6502 e al classico DOS Apple per la gestione dei dischi.

Installazione della scheda.

Prima di tutto è necessario che le altre schede presenti nella macchina siano al posto giusto. Il CP/M, così come l'Apple Pascal funziona bene solo se si rispettano le regole di utilizzo delle linee di I/O che sono i veri vincoli nella comunicazione del sistema con l'esterno. Le schede di espansione, per quanto riguarda il comportamento di Input/Output sono raggruppabili nelle categorie elencate nella tabella 1.

C'è da dire che se il Pascal funziona già egregiamente significa che le schede sono tutte al loro posto e funzionerà altrettanto bene anche il CP/M, altrimenti conviene adattare il sistema alle esigenze del sistema operativo; fra l'altro questo non interferisce con il funzionamento dell'Applesoft.

Nella tabella 2 sono riportate i vari tipi di schede accettati nei vari slot del sistema. Supponendo un Apple IIe che possiede 8 slot di espansione fare

(Continua a pagina 17)

Tipo di scheda	Nomi delle schede
1	Apple disk II controller
2	Apple Communication Interface
3	Apple Super Serial Card Apple Silenotype printer
4	Apple Parallel Printer Card

Tabella 1

riferimento alla tabella per la disposizione ottimale delle schede.

Appurato che il posto giusto per la scheda Z80 è lo slot numero 7 e che lo slot 6 deve essere equipaggiato con un controller per le unità floppy che assumeranno nome A: e B: all'interno del sistema operativo, possiamo chiudere il coperchio e pensare al software vero e proprio. Un riassunto delle lettere per i drive disponibili è elencato nella tabella 3.

Boot.

La prima cosa da sapere prima di inserire il disco con il sistema operativo nel drive A: (cioè la prima unità floppy, quella che il Pascal chiama Apple #1), è verificare quale disco sia corretto. Esistono due versioni del CP/M per Apple, ospitate su due floppy diversi: uno formattato a 13 settori, adatto ai modelli che usano il DOS in versione 3.2 e l'al-

tro su floppy a 16 settori, per quei modelli di macchine che usano il DOS 3.3 o il Pascal UCSD.

Accendendo il sistema con il floppy del sistema operativo l'Apple "salta" direttamente sullo Z80 e si pone in modalità CP/M.

I sistemi più vecchi che non hanno in ROM il cosiddetto sistema di Auto-start, dovranno essere guidati a mano con una serie di CTRL-K fino ad arrivare allo slot giusto (il sesto) per avere il caricamento in memoria del contenuto del floppy.

L'avvenuto caricamento ci viene annunciato da un prompt che cita testualmente:

```
APPLE II CP/M
44K vers. 2.2X
(C) 1980 MICROSOFT
A>
```

La scheda Z80 per Apple II è relativamente poco rara anche in Italia, merito anche di un costo non eccessivo e della disponibilità di software di qualità che ne rendono l'acquisto appetibile a molti.

Tabella 2

Slot	Tipo di scheda accettata	Utilizzo
0	Non usato per I/O	In un IIe è mappato sulla Language card che contiene l'Applesoft.
1	Tipi 2, 3, 4	Line Printer Interface (il CP/M chiama questa periferica LST:)
2	2,3,4 per l'input 1,2,3,4 per l'output	Device generali di I/O (per il CP/M sono le periferiche PUN: e RDR:)
3	2,3,4	Device CRT: o TTY: per il CP/M. Se non è presente nessuna scheda di questo tipo viene usato il display dell'Apple.
4	1	Controller dischi per i drive E: e F:
5	1	Controller dischi per i drive C: e D:
6	1	Controller dischi per i drive A: e B: E' obbligatorio che sia presente
7	Tutti i tipi	Nessun assegnamento di I/O: è il posto ideale per installare la scheda Z80.


```

Softcard CP/M
60K Ver. 2.23
(c) 1980,1982 Microsoft

A>dir
A: CAT      COM : CONFIGIO  BAS : DDT      COM : BOOT      COM
A: MFT      COM : PATCH    COM : CPM60    COM : PIP       COM
A: STAT     COM : ASM      COM : AUTORUN  COM : LOAD      COM
A: COPY     COM : APDOS    COM : SUBMIT   COM : XSUB      COM
A: DUMP     ASM : DUMP     COM : DOWNLOAD COM : MBASIC    COM
A: GBASIC   COM : ED       COM
A>
    
```

Il CP/M usa la scheda 80 colonne su un Apple IIe.

Per inciso il prompt della shell è proprio la lettera del drive corrente (A) seguita dal simbolo di maggiore (>).

Il sistema è pronto ad accettare comandi; per ora quello che sappiamo è che si tratta di una versione 2,2X, che sono disponibili 44 Kb di RAM e che si tratta di un prodotto Microsoft.

Prima di tutto perché 44K se abbiamo un sistema da 64 Kb? Il sistema operativo CP/M si riserva di suo 4 Kb per il buffer di I/O e la mappatura della memoria video. Su un sistema di

arrivare a 56 Kb. Questo è consentito se è installata una Language Card su un IIe che ha di fatto la Language Card installata embedded nel sistema. Purtroppo non tutti i 16 Kb di RAM presenti sulla Language Card sono indirizzabili, così che il CP/M può aggiungere 12 Kb alla sua dotazione base di 44 portando il totale appunto a 56 Kb di RAM disponibile all'utente.

Primi passi con il CP/M.

Il comando DIR elenca il contenuto del floppy presente nel drive di boot (A:). Sono presenti numerosi file che costituiscono la dotazione del cosiddetto "Master Disk"; sono detti "comandi esterni" per distinguerli dai "Comandi interni" che sono disponibili all'interno della shell di sistema.

La shell del sistema operativo, cioè quella parte che permette l'interazione con l'utente accettando i comandi dal

(Continua a pagina 19)

Let-tera	Controlle r nello	Drive collegato al controller
A:	6	1
B:	6	2
C:	5	1
D:	5	2
E:	4	1
F:	4	2

tipo Apple][o Apple][Plus sono installati normalmente 48 Kb, in prima battuta il CP/M viene quindi configurato per indirizzare i 44 Kb rimanenti e sicuramente disponibili (con meno di 48 Kb non si carica).

Niente paura, una delle prime operazioni che faremo è proprio quella di generare una nuova immagine del sistema operativo che possa

Tabella 3

terminale, si chiama *COMMAND.COM* e viene caricata in memoria al momento del boot e tutte le volte che questo è necessario. Il *CP/M* è infatti in grado di scaricare il *Command* dalla RAM per fare posto a programmi eseguibili. Quando il programma finisce il *Command.com* si deve trovare nel driver corrente, altrimenti sono guai.

Chi ha familiarità con il *DOS* Microsoft o anche con le shell testuali dei sistemi *Windows*, troverà una somiglianza notevolissima fra i due sistemi. Ebbene è proprio così, *Bill Gates* e il suo socio *Paul Allen*, acquistarono un "simil *CP/M*" e lo adattarono al *PC IBM* conservandone le caratteristiche esteriori. Non è che non fossero in grado di cambiare nome ai comandi o inventarsi un diverso schema di gestione delle unità magnetiche, solo che volevano ereditare per così dire l'avviamento che il *CP/M* si era conquistato in anni di onorato servizio presso gli utenti.

La prima cosa da fare è, come buona norma detta, una copia del *Master Disk* da usarsi al posto del prezioso originale.

Chi possiede due drive floppy sarà avvantaggiato nelle operazioni e noi lavoreremo considerando questa configurazione. Le stesse cose si possono fare anche con un solo drive, è solo più scomodo ma sarà il sistema a chiederci di cambiare supporto quando necessario.

Il comando per formattare è:

FORMAT B:

Seguito ovviamente dal *Return* (Invio). Ci verrà chiesto di inserire il floppy da formattare e alla fine la scritta "*FORMAT COMPLETED*" ci avviserà dell'avvenuta operazione.

Per creare una copia di backup del *Master Disk* presente nel drive *A:* sul floppy appena formattato e ancora in-

serito nel drive *B:* si procede con il comando:

COPY A: = B:

Chi possiede un solo drive dovrà mettere due volte l'indicazione *A:* per innescare il dialogo di richiesta del floppy giusto al momento giusto.

Normalmente la copia del *Master Disk* è utile solo quando si effettuano lavori di tipo sistemistica e servono i comandi in essa contenuti. Dato che lo spazio sui floppy è scarsino (siamo nell'ordine di 100 Kb), conviene preparare un disco che faccia *Boot* e che contenga solo l'essenziale per lavorare, ad esempio un disco con il compilatore *Basic*, un disco con il *Word Processor*, etc...

Per fare questo è sufficiente formattare un floppy ed effettuare il comando di copia del solo sistema operativo:

COPY B: = A: /S

Il parametro */S* dice appunto al comando *COPY* di procedere al trasferimento del solo sistema operativo (in pratica il *boot sector* ed il *Command.com*).

Per questo numero ci fermiamo qui, la prossima puntata cominceremo a vedere come portare il sistema alla gestione dei 56 Kb di RAM e parleremo delle numerose utilities a corredo del sistema.

Nota: il comando *FORMAT*, così come altri numerosi comandi di uso non comune sono presenti sotto forma di file eseguibili sul floppy *Master Disk*.

[sm]

Emulatori

Gli emulatori sono
la più grande
invenzione
dell'informatica
dopo i
calcolatori!

Extend2—Emulatore per ZX80 e ZX81

Grazie alla presenza della CPU Z80, molto diffusa e quindi ricca di programmi di emulazione e alla semplicità dell'hardware del sistema, lo Z80 della Sinclair gode di una ricca scelta di emulatori. La maggior parte di essi per la verità sembra siano fermi nello sviluppo e supporto, forse proprio per il fatto che hanno esaurito la loro funzione, cioè sono perfetti così come sono. In realtà si sentirebbe il bisogno che fossero supportati dalle piattaforme moderne, mentre quasi tutti sono pensati per il DOS.

Oggi vogliamo parlare di Extend2, un emulatore di ZX80 e ZX81 disponibile per la piattaforma DOS. La versione che abbiamo testato è la 2.0 beta 13 rilasciata in aprile 2004, l'ultima al momento della stesura di questo articolo.

Installazione.

Scaricato e decompresso il tutto (poco più che 500k) in una directory dell'hard disk possiamo eseguire l'emulatore con il comando XT2 da una shell DOS. In questo modo il sistema prende di default lo ZX81 come macchina da emulare ma è sufficiente intervenire

sul file di configurazione XT2.CFG per lanciare l'emulazione di uno ZX80 all'avvio.

In alternativa l'emulatore accetta come parametro un file di configurazione precedentemente preparato:

```
XT2 = ZX80
```

Userà il file di configurazione Zx80.xt2.

L'autore, un olandese che risponde al nome di Carlo Delhez, ha preparato i file di configurazione per lo ZX80, il successivo ZX81 e qualche altra configurazione di test.

Sono presenti anche le definizioni di sistemi "alternativi" come ASZMIC, un sistema simile allo ZX80 del quale si trova qualche traccia sporadica sul Web, ma non sufficiente per presentarne le caratteristiche. Un'altra configurazione è presente per il TS1000 che da quanto ho dedotto dalla documentazione, peraltro non abbondante, trattasi di una versione dello ZX81 a 60 Hz, cioè con l'emulazione del segnale video TV in standard NTSC americano. Anche un clone dello ZX81 costruito in Cina, chiamato PC8300 è presente

(Continua a pagina 21)

Nome: Extend2 Versione 2.0 Beta 13

Autore: Carlo Delhez

Web: <http://www.delhez.demon.nl/>

Prezzo: Free

Host: DOS

Sistemi emulati: Sinclair ZX80 e ZX81

con tanto di ROM di font originali. Sia l'Aszmic che il PC8300 non sono riusciti a farli funzionare, forse colpa della troppa velocità del sistema sul quale ci ho provato. Credo che lavorando sui parametri di configurazione possa essere possibile risolvere il problema.

Un discorso merita la presenza nel pacchetto delle ROM, elemento indispensabile come sappiamo per il funzionamento di qualsiasi macchina emulata. Normalmente c'è una sorta di accanimento delle ditte proprietarie dei marchi al fine di una protezione ad oltranza dei loro diritti. Questa volta la Amstrad, attualmente detentrici del marchio Sinclair, fa una lodevole eccezione liberalizzando l'uso delle ROM se usate per emulare i sistemi originali. Grazie Amstrad!

Funzionamento.

L'autore dichiara nella documentazione che XT2 funziona in una finestra DOS di Windows. Personalmente su un sistema con Windows XP non sono riu-

scito ad attivare questa feature, cioè sono riuscito a farlo funzionare solo in full screen e di conseguenza non sono riuscito a catturare le videate per mostrare un esempio di funzionamento nell'articolo. Tutti i tool di mia conoscenza per la cattura dello schermo hanno fallito (vai a sapere il perché)!

L'emulatore da il meglio di se per il sistema ZX81 e questo è obiettivamente logico essendo il successore dello ZX80 meglio "carrozzato" e di conseguenza più "divertente" da emulare. In ogni caso l'emulazione dello ZX80 è perfetta, tanto che l'autore si è spinto fino a emulare la perdita di sincronismo quando la CPU è occupata a servire l'input da tastiera o nell'esecuzione del programma. E' possibile emulare anche una "cassetta virtuale" e quindi salvare o caricare programmi, sempre con le limitazioni imposte dall'estrema frugalità del sistema.

Una informazione importante: nell'emulazione la tastiera del PC è praticamente "morta" per cui per abbandonare la sessione è necessario chiudere l'emu-

L'emulazione dello ZX80 è perfetta, tanto che l'autore si è spinto fino a emulare la perdita di sincronismo quando la CPU è occupata a servire l'input da tastiera o nell'esecuzione del programma

(Continua a pagina 22)

```
//-----
//      XTender2 configuration file for ZX80 with 16k RAM, SAVE/LOAD patches
//
//      by Carlo Delhez, September 1998
//      copy and/or modify to suit your requirements
//
//      see DEMO.XT2 for information about XT2 configuration files
//
Name ZX80_File
Allocate 4k of ROM at 0k
Load ZX80_ORG.ROM at 0k          // load ZX80 ROM into ROM-area
NMIgenerator Absent             // exclude NMI generator
Allocate 16k of RAM at 16k      // give it 16k of RAM
ROMpatch ZX80save at 0x1B7      // save ZX80 program to disk
ROMpatch ZX80load at 0x207      // load ZX80 program from disk
Extension O                     // use *.O for ZX80 programs
// End of configuration file
```

```

*** ERROR: Aborted by F1 ***
-----
PC = 0038 : 0D C2 45 00 E1 05 C8 CB D9 ED 4F FB E9 D1 C8 18
SP = 7FF8 : 2C C0 A4 01 23 03 02 3F 21 FF 7F 3E 3F C3 61 02
HL = C02B : 76 76 76 76 76 76 76 76 76 76 76 76 76 76 76
AF = DD12 BC = 190F DE = C02C HL = C02B IX = 0000
AF' = 0000 BC' = 0000 DE' = D0F0 HL' = 402B IY = 4000
I = 0E R = 82 IM = 1 IPF1:2=0:0 NMI = 0 CSYNC = 0
executed 0x13B17CDB cycles in 0h:00'41" -> emulated clock is 7.874 MHz (242%)
-----

XTender2 BETA version 13 - Copyright 1998-2004 by Carlo Delhez

C:\EMULAT~1\XT2B13>_

```

Figura 1

latore con F1 o passare ad altri programmi con ALT-TAB o con ALT-INVIO. Quest'ultimo dovrebbe riportare il programma in una finestra DOS ma, come ho già detto, sembra che non funzioni su XP.

L'abort dell'esecuzione tramite F1 stampa a video un debug dei registri (vedi figura 1).

Conclusioni.

Emulare il sistema ZX80 e il successivo ZX81 della Sinclair fa veramente ritornare ai tempi pionieristici dell'informatica personale.

Grazie alla relativa semplicità costruttiva di questo hardware e alla presenza dello Z80 per il quale sono stati a sua volta prodotti moltissimi emulatori, si possono trovare un discreto numero di programmi di emulazione per i sistemi della Sinclair. La possibilità di ridistribuire le ROM è un altro punto che ha favorito chi ha voluto cimentarsi nella costruzione in proprio.

Pochi sembrano allo stato attuale i progetti ancora "vivi", cioè che hanno avuto una release recente o che sia

mantenuto aggiornato il loro sito. Questo potrebbe apparire come poco importante dal momento che le implementazioni che ci sono, parliamo naturalmente di quelle funzionanti e sufficientemente testate, svolgono soddisfacentemente il loro ruolo.

Non va dimenticato però che nell'informatica c'è l'insidioso tarlo dell'incompatibilità tecnologica. Avere oggi un programma che funziona su una certa piattaforma non assicura che fra 10 anni o anche meno si troverà l'hardware/software adatto per farlo girare! Pazienza per gli emulatori DOS, che in qualche maniera hanno mantenuto un funzionamento nel tempo grazie all'emulazione che Microsoft ha da sempre inserito nei suoi sistemi operativi (ma ancora per quanto?), ma gli altri? Prendiamo ad esempio gli emulatori che girano sotto MAC OS, non l'ultimo MAC OS X, intendo quello precedente non caricabile sulle macchine di ultima generazione prodotte da Apple. Queste realizzazioni fra qualche anno saranno appannaggio dei retrocomputeristi in possesso di un qualche hardware Apple compatibile.

Ecco che sarebbe importante che gli autori rilasciassero anche il codice sorgente di queste loro realizzazioni, e qualcuno, onore al merito, per la verità lo fa già.

[L2]

BBS

La posta dei lettori

Critiche e suggerimenti...

Da Devil Master

Per la redazione: considerate queste critiche come costruttive perché l'idea è veramente interessante. Vedo che ci sono diverse pecche nell'impaginazione.

(...Omissis...)

Da Gabriele

... (omissis)...Un'altro appunto va alla qualità delle immagini: i ritagli sono spesso molto approssimativi ed anche la risoluzione a volte lascia a desiderare. Considerate che, ad esempio, in Paint Shop Pro ci sono potenti funzioni di selezione che permettono anche una sfumatura dei bordi in modo da ottenere ritagli morbidi e che è sempre bene lavorare alla massima risoluzione possibile ed in 16 milioni di colori, downgradando e/o risizando semmai l'immagine finale dopo aver apporato tutte le modifiche del caso, vi garantisco che l'aspetto finale ne gioverà assai.

Risponde la redazione.

Le segnalazioni, critiche, suggerimenti ed impressioni d'uso sono essenziali per noi che abbiamo l'intento, pur nella nostra limitatezza di dilettanti, di dare vita ad una pubblicazione di qualità in tutti i suoi aspetti. Prontamente abbiamo pubblicato una nuova versione del file pdf (un service pack si direbbe oggi) con queste e anche altre piccole correzioni ed aggiustamenti.

Abbiamo molto da imparare, lo sapevamo, non stiamo forse nemmeno usando gli strumenti giusti e sicuramente non nelle loro piene potenzialità, ma cresceremo, questo è sicuro.

Una proposta da Fedora...

Le riviste on-line sono pratiche e poi... "a caval donato...", ma io preferisco la versione cartacea. D'accordo voi direte, stampatela! Ho calcolato che per avere una qualità un po' decente con copertina e rilegatura, dovrei spendere quasi 15 Euro! Non è che ci state pensando voi con relative economie di scala?

Risponde Tn

Caro/a Fedora, bello sarebbe arrivare alla stampa, ti dirò che per noi la facciamo anche se la tua stima dei costi mi sembra esagerata.

Da parte nostra un calcolo di massima indicherebbe la possibilità di offrire la versione cartacea al costo di 6/7 Euro spedizione compresa. Questo senza copertina patinata e rilegatura a patto di avere una tiratura minima di un centinaio di copie. Ma esistono cento appassionati di retrocomputer interessati? Meglio per ora stare con i piedi per terra e concentrarsi sui progetti realizzabili, poi

si vedrà.

Feedback negativi...

(letto su un forum)

La rivista non mi piace molto, certo vedere la copertina come si facevano una volta e leggere la recensione dell'Apple II fa impressione, ma per il mio gusto è troppo ancorata al passato. Avrei preferito una ZXNews che parlasse delle novità che continuano ad esserci per la piattaforma ZX come del resto anche per altre (vedi Amiga)...

Risponde Sonicher

Dire che la rivista sembra indistinguibile da una edita nei primi anni '80 è uno straordinario complimento! C'è comunque spazio e (speriamo) tempo per curare qualche pagina che guardi in avanti, in una specie di "ritorno al futuro". Il vero problema è la reperibilità di questo materiale "nuovo", confinato nel circolo dei soli "fondamentalisti" (in senso buono) di un certo modello-/marca.

J.N. è disponibile ad ospitare un contributo in questo senso, fatevi avanti.

Non di solo hardware...

Da vaxxten

... retrocomputer è anche software? Me lo chiedo spesso davanti alla mia collezione di programmi che ho raccolto con certissima pazienza (e forse altrettanta incoscienza) in oltre vent'anni di attività informatica. Mi ha confortato leggere sul primo numero che non considerate "di serie B" nessun tipo di passione legata al retro-computer. Secondo voi è lecito detenere queste raccolte?

Risponde Tn:

È proprio così, non esiste a mio modesto avviso un retro-computing di serie A e un altro di serie B. Del resto l'hardware senza il relativo software a cosa servirebbe? Azzardando un irriverente paragone (non me ne vogliano i bibliotecari) penso che la conservazione del software sia da considerarsi alla stregua della conservazione dei libri antichi e sia altrettanto delicato della conservazione degli incunaboli (i volumi editi prima del 1501). Fra l'altro so per esperienza che è tutt'altro che semplice assicurare la piena funzionalità di queste raccolte, sia per il decadimento dei media (sicuramente le avrai trasferite da floppy a cd-rom! Se non l'hai fatto ti consiglio di intervenire al più presto!) e sia per la progressiva scomparsa dell'hardware adeguato a farli "girare" (per fortuna ci sono gli emulatori!).

In merito se sia lecito o meno la detenzione di tali raccolte, non so rispondere. Recentemente un giudice di Bolzano ha assolto una persona che aveva un migliaio di cd-rom con software, film e mp3. La sentenza ha riconosciuto che trattasi di una collezione privata senza scopo di profitto. Mi sembra una interpretazione della legge sul

BBS
è la rubrica della posta dedicata al colloquio con chiunque voglia raggiungere la redazione con dei commenti, incoraggiamenti e, perché no, anche critiche, purché costruttive e non faziose.

Scrivete a:

redazione@jurassicnews.com

Info@jurassicnews.com

copyright basata sul buon senso e se questo è vero per i film mi parrebbe maggiormente vero per il software, sempre che si tratti di versioni prive di interesse economico. Diciamoci la verità: chi installerebbe un Windows 3.1 per lavorarci seriamente o userrebbe Quattro Pro come spreadsheet?

Emulatori

La costruzione "in casa" di un emulatore di CPU è una buona occasione per affinare le proprie competenze informatiche

Costruiamoci un emulatore

Abbiamo chiamato questo progetto **Zen80** ed è un progetto didattico che si propone lo scopo di costruire un emulatore partendo dai concetti di base e dall'emulazione di una CPU a 8 bit. L'idea è di proseguire poi con gli aspetti più complessi fino ad arrivare all'emulazione di un computer completo di prima generazione.

I concetti presentati vengono semplificati al massimo e sono tralasciate volutamente i particolari più complessi. Scriveremo del codice in C, quindi molto portabile, abbordabile per chi conosca anche in maniera non particolarmente approfondita il linguaggio, anzi potrebbe essere proprio questa l'occasione per impararlo!

Il contenuto del corso.

La serie di articoli (è inevitabile affrontare la cosa a puntate) sarà introdotta da una stringata lezione teorica sui concetti base dei microprocessori. Prenderemo come esempio la CPU Z80, che è stata una delle più diffuse e grazie alla sua sostanziale linearità, permette di affrontare il problema su diversi piani di complessità.

Seguirà l'impostazione delle strutture di base e dei primi obiettivi del progetto, quello che chiameremo Stage 0. Questo ci permetterà di costruire un emulatore di CPU completo.

Nello Stage 1 inseriremo la nostra CPU in un ideale hardware e impareremo a scrivere i primi semplici programmi per testare il progetto.

Aggiungeremo quindi un "monitor" (Stage 2) e la gestione di una tastiera esadecimale e sarà l'occasione di introdurre il concetto di programmi base (quello che oggi si chiama BIOS).

Il progetto prevede di fermarsi a questo punto anche se sviluppi successivi sono certamente possibili; non vorremmo però "allungare troppo la minestra"...

Lo scopo finale rimane comunque quello didattico anche perché se volessimo emulare un vero calcolatore, per quanto semplice possibile, dovremmo cominciare a dotarlo di un sistema operativo, di un interprete basic o peggio dovremmo affrontare il problema della simulazione di un PC reale con ROM e tutto il resto.

Concetti base.

Una CPU a 8 bit come il classico Z80 della Zilog, che è stato il cuore di moltissimi sistemi nell'epoca degli "home computer" (anni '80), è incredibilmente semplice rispetto alla complessità con la quale sono costruite le moderne CPU come i Pentium, Athlon e via dicendo.

Questi ultimi dispongono infatti di

(Continua a pagina 25)

Cosa serve:

- Un compilatore C visuale o a linea di comando GCC va benissimo, così come Turbo C o Visual C/C++.
- Qualsiasi sistema operativo che supporti la programmazione "console" (praticamente tutti).
- Una infarinatura di C e di programmazione.

"sezioni" altamente specializzate e di una "intelligenza" molto sofisticata del micro-codice. Anche a livello di gestione della memoria e interrupt le cose sono molto complicate oggi giorno, anche se non impossibili e lasciate all'appannaggio dei professionisti "veri", come quelli che costruiscono gli emulatori commerciali, VmWare e VirtualPC tanto per citare i due più famosi.

Una classica CPU a 8 bit è schematizzabile per gli scopi che ci prefiggiamo come illustrato in figura 1. Si distinguono essenzialmente tre blocchi:

1. l'unità aritmetico-logica (ALU) che fa materialmente tutti i calcoli;

2. il "decodificatore" di istruzioni e gestore degli indirizzi e dell'input/output dei dati da e verso memoria e periferiche;

3. i registri interni divisi in due sotto-sezioni: quelli che servono per i calcoli (registri generali) e quelli che servono per la gestione interna del processore (registri di controllo).

I registri sono gli oggetti direttamente interessati quando si scrive un programma per un microprocessore. Il programma sarà infatti una sequenza di istruzioni più o meno coinvolgenti i registri interni e l'Accumulatore in particolare. L'accumulatore è un registro molto speciale, destinato ad essere coinvolto nella maggioranza delle istruzioni e in generale ad accogliere il risultato delle elaborazioni; lo indicheremo con la lettera **A** maiuscola.

I registri appartenenti al gruppo che abbiamo definito "di controllo" servono per guidare il funzionamento del processore. Il Program Counter (**PC**) contiene l'indirizzo di memoria dal quale prelevare la prossima istruzione.

Normalmente viene incrementato di una unità ad ogni "ciclo di istruzio-

ne" (che non coincide con il ciclo di clock, visto che una istruzione normalmente impiega più di una unità di tempo), ma ovviamente esiste la possibilità di variarne a piacere il contenuto per effettuare i cosiddetti "salti" all'interno del programma.

Il Registro Flag (**F**) è una raccolta di 8 caselle in grado di esplicitare il risultato di certe operazioni. Ad esempio se una istruzione di somma provoca un overflow (questo concetto sarà spiegato più avanti), il flag corrispondente sarà "settato" al valore "1" e la successiva istruzione potrebbe essere proprio una istruzione che controlla se questa situazione si è verificata o meno.

Altri registri di controllo servono per contenere indirizzi di memoria da impiegare come operandi nelle istruzioni o per la gestione dell'hardware stesso. Ad esempio lo Z80 contiene un registro (**R**) che è utilizzato per il refresh della memoria RAM nel caso che il progettista abbia previsto l'impiego di chip di memoria RAM dinamica.

Alcune istruzioni possono utilizzare i registri a coppie, ad esempio **BC**, **DE**, **HL**, in questo caso le grandezze coinvolte sono a 16 bit.

Una CPU a 8 bit come il classico Z80 della **Z I L O G** è incredibilmente semplice rispetto alla complessità con la quale sono costruite le moderne .

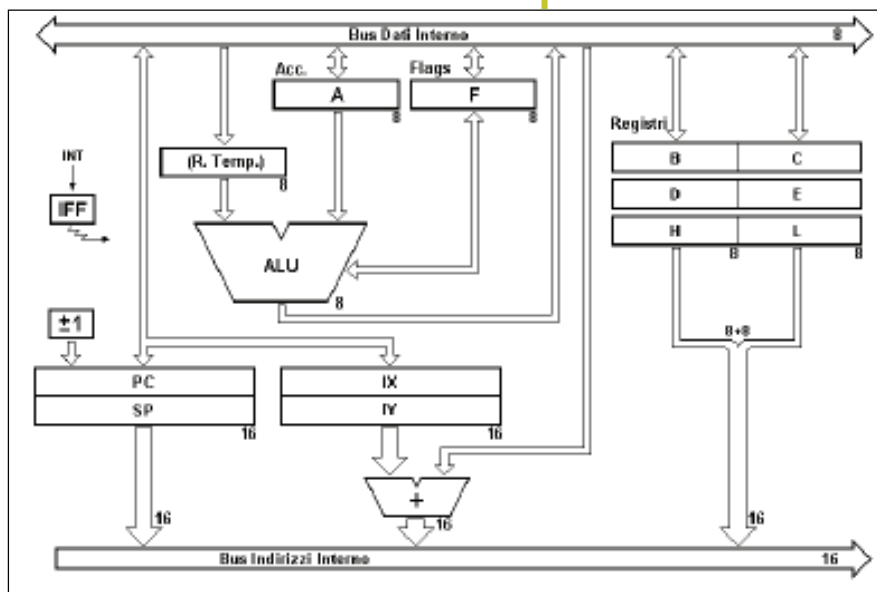


Figura 1—schema a blocchi della CPU Z80

(Continua a pagina 26)

Ram Dinamica, chi è costei?

La RAM dinamica conserva il suo stato intrappolando gli elettroni in una specie di condensatore che purtroppo tende a perdere la carica. L'operazione di refresh consiste nell'effettuare una lettura delle celle di memoria, cosa che come effetto collaterale ha il rinfresco delle informazioni contenute.

Lo Z80 ha poi la possibilità di disporre di coppie alternative di registri, chiamati **A'**, **B'**, etc..., caratteristiche che consente un "context switching" ad un livello estremamente efficiente.

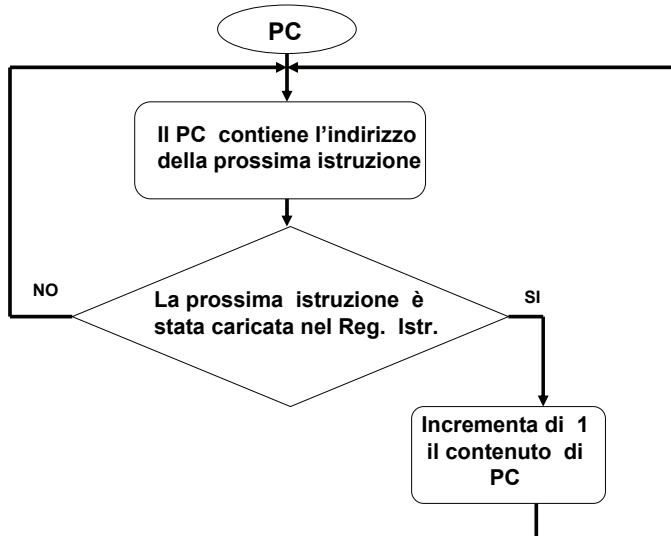
Per completare questa panoramica sui principi fondamentali dell'oggetto che andiamo ad emulare, parleremo brevemente di cosa ci sia alla base del funzionamento di un processore.

In figura 2 una rappresentazione a blocchi veramente minimale di cosa succeda all'interno del processore una volta data corrente. Il Program Counter viene impostato ad un certo valore dall'operazione di Reset. Di suo il registro PC conterrebbe l'indirizzo 0x0000 al momento del reset, spesso si preferisce forzare con un piccolo circuitino l'effettiva locazione di memoria nella quale sarà contenuta una istruzione di salto all'effettivo indirizzo iniziale.

Il funzionamento della CPU è tutto qui: un ciclo infinito (fino a che non si toglie corrente) di prelevamento istruzioni dalla memoria, incremento del program counter, esecuzione istruzione e via all'infinito...

Figura 2—funzionamento del Program Counter

LA FUNZIONALITA' DEL "PROGRAM COUNTER"



Pronti? Si parte.

Per incominciare quello che ci serve è la definizione della grandezza Byte e Word che otteniamo con le due righe di codice:

```
typedef char byte;
typedef unsigned short word;
```

Passiamo ora alla rappresentazione dei registri della CPU.

```
byte A, B, C, D, E, H, L;
byte F;
word AF, BC, DE, HL;
word IX, IY;
word SP;
word PC;
word I, R;
```

Abbiamo scelto di usare le convenzioni dei nomi per la CPU Z80. In questo modo si crea un parallelo fra il codice dell'emulatore e il microcodice della CPU. Inoltre sarà più immediata la lettura del codice binario che possiamo trovare in giro, su manuali o altro, relativo allo Z80.

L'immagine di figura 3 mostra i registri e la loro organizzazione logica.

C'è un doveroso approfondimento tecnico da introdurre a questo punto. In una CPU reale i registri "a coppie" come AF, BC, DE e HL non sono strutture a se stanti, ma l'unione dei due registri a 8 bit. Questo significa che una variazione del registro B, ad esempio, implica la corrispondente variazione del valore nella coppia BC. Si sarebbe potuto ricostruire lo stesso funzionamento nell'emulatore (ad esempio usando una struttura), ma dato il carattere educativo del nostro progetto abbiamo preferito definire variabili diverse per i registri e farci carico

(Continua a pagina 27)

dell'allineamento nella stesura delle istruzioni.

Un'altra considerazione riguarda i registri I e R che nella CPU sono usati per la funzione di refresh delle memorie dinamiche. Normalmente non sono utilizzati ma talvolta si ricorre al loro valore per simulare ad esempio un generatore di numeri casuali. Pur avendoli definiti per completezza non li useremo all'interno dell'emulatore.

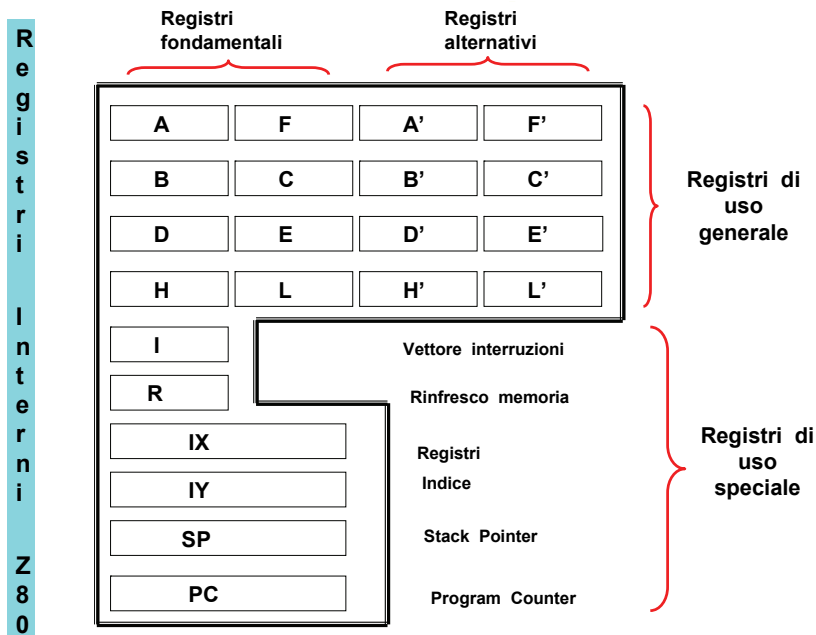
Il registro F (chiamato Flag Register) è un oggetto a 8 bit che viene manipolato dalle istruzioni e serve per la tracciatura di alcune condizioni come ad esempio l'overflow nelle operazioni di somma. Ne segue che è importante considerare i singoli bit del registro, piuttosto che il suo contenuto globale; per questo approntiamo una mascheratura di ogni singolo bit:

```
#define S_FLAG      0x80
#define Z_FLAG      0x40
#define H_FLAG      0x10
#define V_FLAG      0x04
#define N_FLAG      0x02
#define C_FLAG      0x01
```

Questo ci consentirà di testare i singoli elementi del byte applicando appunto una maschera in AND al valore contenuto nel registro.

Anche in questo caso si tratta di una scelta progettuale. Sarebbe stato più comodo forse definire ogni singolo bit come oggetto a se stante, qualcosa come:

```
struct F (
    char c;
    char n;
    char v;
    char h;
    char z;
    char s;
);
```



In modo da riferirsi ad ogni singolo elemento con istruzioni come:

```
if F.c = 0 then.....
```

La nostra scelta rende un po' più complicato la codifica del micro-codice ma più attinente la struttura interna della CPU emulata.

Una CPU senza memoria esterna (RAM e ROM) sappiamo che serve a poco. Definiamo la nostra memoria virtuale con le istruzioni:

```
byte ram[0xffff];
byte mem_map[0xffff];
```

Si tratta di due array da 64Kb. Quello chiamato ram è lo spazio di memoria simulato, la mem_map è una struttura di appoggio che ci servirà per inserire un attributo di sola lettura a certe aree di memoria che andremo ad occupare con memoria ROM.

Figura 3—I registri interni alla CPU

Il microcodice sono le istruzioni base che consentono alla CPU di effettuare le operazioni aritmetico-logiche che conosciamo. Si tratta di una programmazione a livello fisico, molto legata allo specifico progetto di CPU.

(Continua a pagina 28)

Big Endian or Little Endian

La filosofia costruttiva delle CPU decide il modo in cui i byte sono organizzati in memoria quanto si debbano rappresentare quantità superiori al singolo byte.

Non escluso che anche la scelta Big-endian vs. little-endian sia stata oggetto di "guerre di religione" fra le due fazioni sostenitrici. La verità è naturalmente che la scelta migliore dipende dall'architettura del processore, cioè se sia più efficiente una rappresentazione rispetto all'altra.

NOTA: è possibile che il vostro compilatore segnali un problema a questo punto. Gestire un array da 65.000 elementi, anche se di un byte ciascuno, può richiedere un parametro di compilazione specifico (tipo il modello huge per il TurboC). Se non riuscite a far supportare al compilatore questa variabile non vi resta che spezzare la memoria in "banchi" più piccoli gestendo manualmente la continuità degli indirizzi.

Non a caso la definizione della RAM viene fatta con la dimensione codificata in esadecimale: le CPU funzionano in binario e l'esadecimale è ciò che usano i programmatori assembly tutti i giorni, tanto vale abituarvici.

Vediamo un esempio di caricamento dei valori nella RAM. Le istruzioni seguenti predispongono il contenuto di appena cinque byte di memoria, ma sono sufficienti per capire il concetto.

```
ram[0x10]=0x01;// LD BC,<data16>
ram[0x11]=0x3f;// byte basso
ram[0x12]=0x21;// byte alto
ram[0x13]=0x03;// INC BC
ram[0x14]=0x76;// halt
```

Il primo byte è una istruzione di caricamento immediato di un valore nella coppia di registri BC. Il termine "caricamento immediato" si riferisce al fatto che il valore da caricare è nei byte immediatamente successivi di memoria.

La CPU Z80 lavora in big-endian, che è un termine informatico per dire che nella rappresentazione di quantità a 16 bit (e quindi che occupano due bite), il byte più "pesante" si incontra dopo il byte più "leggero".

A questo punto definiamo la struttura del microcodice della CPU. Questo per il nostro emulatore sarà un semplice array contenente gli indirizzi delle corrispondenti funzioni C che emulano le corrispondenti funzioni della CPU.

L'indirizzo dell'elemento è il codice della micro-istruzione, il contenuto è l'indirizzo della funzione C che effettua materialmente l'emulazione.

```
struct{
    int (*fun_addr)();
} microcode[128];
```

NOTA: in realtà avremo bisogno di altri array per il microcodice, per il semplice fatto che uno solo sarebbe sufficiente solo per le CPU che codificano tutte le istruzioni in un solo byte. La CPU Z80 possiede invece delle istruzioni a due byte che sono per la verità usate pochino, come si può ben immaginare.

Come sono scritte le funzioni di emulazione? Siamo giunti al core dell'emulatore: ora bisogna cominciare a scrivere il codice di emulazione vero e proprio.

L'istruzione più semplice in assoluto è naturalmente la NOP, quella che non fa assolutamente nulla:

```
// -- 0x00
int z80_nop(){
    return(0);
};
```

Come si vede codificheremo ogni istruzione della CPU con una funzione int del C. Il codice di ritorno 0 significa che tutto è andato bene, un valore diverso significherà qualche problema di esecuzione. Per la verità questa finenza potevamo anche risparmiarcela, visto che è impossibile che ci siano errori nelle istruzioni a livello assembler, ma tant'è per completezza... Se simulassimo una CPU con la funzione di divisione, allora si potrebbe verificare una divisione per zero e di conseguenza un

(Continua a pagina 29)

trap software, allora il codice di ritorno servirebbe a trattare l'anomalia.

Una istruzione leggermente (ma poco) più complicata è quella individuata dal codice 0x01, che è proprio la prima del nostro esempio. Vediamo di codificarla:

```
// -- 0x01
int z80_ld_bc_data16() {
    C = ram[PC];
    PC++;
    B = ram[PC];
    PC++;
    BC = B*256 + C;
    return(0);
};
```

Credo sia semplice capirne il funzionamento, comunque per andare avanti a piccoli passi la illustriamo nel dettaglio.

Il registro *C* viene caricato con il contenuto della memoria all'indirizzo indicato dal program counter (ricordate che il *PC* viene incrementato di uno ogni volta che l'istruzione viene prelevata?).

A questo punto bisogna re-incrementare il *PC* di uno per andare a puntare sul byte alto che verrà copiato nel registro *B*. Ancora un incremento del *PC* che sarà pronto ad indicare la prossima istruzione da prelevare.

Ultimo atto dobbiamo mettere in ordine la variabile *BC* (ricordiamo che per le scelte fatte non è sufficiente caricare *B* e *C* per ritrovarsi anche il registro *BC* inizializzato).

L'istruzione:

```
BC = B*256 + C;
```

È un classico della programmazione. Dati due numeri rappresentati in un byte (cioè al massimo 255), per ottenere una word a 16 bit si moltiplica il byte alto per 256 e si somma il valore del byte basso.

Ovviamente si poteva anche fare in altro modo. Qualcuno obietterà che sarebbe stato "più elegante" inserire il valore di *B* in *BC*, fare uno shift di 8 posizioni verso sinistra e una AND con il valore di *C*...

Beh, che ognuno faccia a suo modo, credo comunque che come impostato da noi sia più comprensibile oltre che traducibile in qualsiasi altro linguaggio che non sia il *C*.

E l'istruzione 0x03, che ci serve per il nostro mini-programma?

Eccola:

```
// -- 0x03
int z80_inc_bc() {
    BC = BC + 1;
    C = BC - B*256;
    B = BC>>8;
    return(0);
};
```

Questa volta la sottigliezza di usare l'operatore di shift del linguaggio *C* è stata utile. Rispetto all'istruzione precedente si vede come questa volta siano i due registri *B* e *C* a venire ricavati dal valore del registro *BC*.

Direi che come prima puntata ci accontentiamo. La prossima volta metteremo giù il codice dell'emulatore vero e proprio mostrando come la possibilità di usare puntatori a funzioni ci permetta di semplificare enormemente le fasi di fetch dell'istruzione.

[bs]

Introduzione all'analisi algoritmica

Teoria e
Applicazione delle
Macchine
Calcolatrici ospita
contributi rivolti in
maniera specifica
all'implementazione
di algoritmi
fondamentali a
alla teoria
informatica

Listato 1

Nell'articolo apparso sul numero precedente della rivista, abbiamo scritto l'algoritmo bubble-sort in Pascal cercando anche qualche ottimizzazione.

Quello che è stato subito evidente è il fatto che l'ottimizzazione del codice passa attraverso l'analisi e la modifica di quella parte di codice che è destinata a essere eseguita ripetutamente.

Sono quindi le parti di codice racchiuse nei loop che devono richiamare l'attenzione del programmatore.

Va precisato che stiamo parlando di ottimizzazione del tempo di esecuzione che è quella che richiama maggiore attenzione nei casi normali. Possono esistere problemi di ottimizzazione anche di grandezza dell'eseguibile, di occupazione di memoria, di I/O o altro.

Definizioni.

L'analisi degli algoritmi si propone lo scopo di "misurare" la grandezza dell'algoritmo stesso e quindi stabilire in quale "ordine" esso si collochi.

Ad esempio un algoritmo di ordine N [$O(N)$] significa che il tempo di esecuzione cresce linearmente con la dimensione del problema. Cioè se applico un certo algoritmo ad un problema di ordinamento di un vettore con 100 elementi, il tempo di esecuzione sarà un qualche valore, diciamo x secondi. Se eseguo lo stesso programma su un vettore di 200 elementi il tempo di esecuzione sarà 2^*x , cioè il doppio.

(Continua a pagina 31)

1	DATA: V(1..N) OF INTEGER	1
2	DATA: SCAMBI TYPE INTEGER	1
3	DATA: I TYPE INTEGER	1
4	SCAMBI = 1	1
5	WHILE SCAMBI > 0	N
6	SCAMBI = 0	N
7	FOR I = 1 TO N-1	$N*(N-1)$
8	IF V(I) > V(I+1)	$N*(N-1)$
9	SCAMBIA(V(I), V(I+1))	$N*((N-1)/2)$
10	INCREMENTA(SCAMBI)	$N*((N-1)/2)$
11	END IF	$N*(N-1)$
12	NEXT I	$N*(N-1)$
13	END WHILE	N

Esempio.

Osservando il listato 1 abbiamo numerato le righe dell'algoritmo Bubble-Sort visto nello scorso numero e a sinistra abbiamo cercato di stabilire quante volte una certa riga sarà eseguita se il vettore da ordinare è lungo N elementi.

Non occorre essere estremamente precisi e devono essere considerati dei casi medi. Si possono comunque anche analizzare le situazioni estreme e definire una complessità minima e massima per l'algoritmo considerato.

Si prende quindi in considerazione il massimo valore che compare nelle righe, nel nostro caso $N*(N-1)$, e si sviluppa il polinomio (cioè si fanno le operazioni algebriche indicate):

$$N*(N-1) = N^2 - N$$

L'ordine dell'algoritmo è il grado massimo del polinomio.

Il fatto che la formula risultante dal calcolo del massimo fattore di esecuzione sia un polinomio, ci fa dire che l'algoritmo in questione appartiene alla classe degli Algoritmi Polinomiali di grado 2.

Si indica con la notazione $O(N^2)$.

L'informazione, se pur approssimata è utile per valutare l'efficienza dell'algoritmo stesso. Infatti se in un esperimento ripetuto molte volte su un vettore di 100 elementi otteniamo ad esempio 15 secondi come tempo medio di esecuzione, allora possiamo essere certi che se il vettore da ordinare ha 200 elementi il suo tempo di esecuzione medio sarà:

$$15 * 2^2 = 15 * 4 = 60 \text{ secondi.}$$

Le cose si fanno più complicate nel caso di algoritmi complessi come ad esempio quello risultante dalla serie di

ottimizzazioni che abbiamo condotto nell'articolo sull'ultimo fascicolo.

Una domanda che sorge spontanea è la seguente: -"Quanto è possibile spingere l'ottimizzazione?".

Evidentemente quando si passa da un approccio generale come ad esempio l'ottimizzazione dei loop, all'approccio specifico legato alla natura del problema specifico, i risultati diventano mano a mano numericamente meno interessanti. Esiste anche un limite teorico, infatti nel caso dei problemi di ordinamento l'ordine minimo appartiene alla classe

$$N*LOG(N).$$

Questa performance è riservata agli approcci molto "studiati" nell'implementazione di algoritmi di sorting. C'è da dire peraltro che l'efficienza dei vari metodi di ordinamento dipende, oltre che dall'algoritmo utilizzato, da altri fattori che sono essenzialmente tre, qui sotto riportati.

La dimensione dell'insieme da ordinare, cioè certi algoritmi sono più efficienti di altri se il numero N è di un certo tipo.

La tipologia di oggetti costituenti l'insieme. E' molto diverso ordinare interi o ordinare record in una tabella di database.

Lo stato iniziale dell'insieme, come abbiamo già intuito parlando dei "casi estremi" nel primo articolo della serie.

[sm]

Nel caso di un algoritmo polinomiale il grado massimo del polinomio determina l'ordine dell'algoritmo

Retro-Riviste

Una rassegna dell'editoria specializzata dai primi anni '80 ad oggi

Scheda

Titolo: ReLoad

Editore:

Web:
www.reloadmagazine.net

ReLoad — n. 1 e n. 2

Anno 2004 — Euro 5,00

Riviste dedicate al retrocomputing esistono soprattutto all'estero. In Italia c'è qualcosa in edicola dedicato principalmente ai giochi e qualcosa accessibile da Web.

All'indirizzo web www.reloadmagazine.net troviamo questa iniziativa editoriale che ha più o meno la stessa mission di Jurassic News ma se ne differenzia per gli aspetti operativi. ReLoad Magazine è infatti a pagamento e stampata su carta. La distribuzione è per ora abbastanza poco organizzata: bisogna registrarsi (nel mio caso ci ho provato e riprovato ma l'account non ha mai funzionato) e quindi richiederne la spedizione.

Il costo di 5 Euro appare adeguato e sufficiente appena, come afferma la redazione, a coprire i costi di stampa e (credo) di spedizione. Ma non voglio indagare su questo aspetto anche perché purtroppo, per le cause che ho appena spiegato, non sono riuscito ad entrare in possesso di nessun numero.

Il sito Web di supporto non appare proprio aggiornatissimo mentre per ora si contano solo due fascicoli usciti in circa un anno di attività (potete vederne le copertine in questa

pagina). Non sta a noi decidere se sia un bene oppure un male, la redazione evidentemente avrà una propria strategia, un "business plan", come si dice oggi.

Dal sommario si possono dedurre i contenuti, peraltro interessanti, mentre un solo articolo di saggio è disponibile per il libero download. Si tratta di un breve saggio sulla tecnologia e l'evoluzione dell'uso del nastro magnetico come media di memorizzazione di dati.

Che dire, sembra un'iniziativa degna di nota ma che apparentemente soffre del male che affligge molte iniziative analoghe e cioè la disponibilità delle persone di dedicarci del tempo in maniera continuativa. So per esperienza che è molto difficile portare avanti qualsiasi forma di volontariato e di associazionismo: normalmente c'è un gruppo ristretto che "tira", mentre gli altri tendono a defilarsi dopo i primi apporti.

Per la verità non sono certo che le difficoltà di reload magazine siano proprio queste, ho fatto solo un'ipotesi ma anche tutti i miei tentativi (via email) per entrare in contatto con qualche redattore sono andati a vuoto...

Da parte della redazione di Jurassic News un augurio sincero di (speriamo) buon proseguimento.

[sc]



Retro-Riviste

Still-Alive—n.0...n.7

Anno 2003/2005 — Free

Ecco un'altra iniziativa amatoriale della quale piangiamo la preamara dipeartita.

Nel 2003 alcuni amici decisero che sarebbe stato bello rinverdire l'atmosfera dei tempi in cui i giochi elettronici occupavano buona parte della loro giornata di studenti liceali, realizzando una rivista specializzata. Nacque così Still-Alive, nome che era tutto un programma: i vecchi giochi vivono ancora!

La pubblicazione ha partorito sette numeri distribuiti gratuitamente sul Web, concludendosi nel 2005 con un laconico messaggio sul sito. Peccato!

Ed è un peccato veramente perché l'iniziativa era estremamente significativa. Che piacciono o no i giochi "prima generazione" è comunque una cultura che ci ha fatto crescere tutti e una palestra nella quale per una decina d'anni si sono allenate le squadre dei futuri programmatori o comunque degli utilizzatori "power" delle tecnologie informatiche.

Il contenuto è monotematico: recensioni di giochi per le più diffuse piattaforme ludiche degli anni '80: Commodore 64, Sinclair e console varie.

Lo stile delle recensioni è uno standard di fatto per questo tipo di idee: la collocazione del gioco in un contesto storico, le premesse fantasiose che ne giustificano l'esistenza e poi via per una partita quasi in tempo reale. Gli autori riescono quasi sempre a creare

l'atmosfera giusta e ad appassionare alla lettura fino all'ultima parola (non è poco, visto che non si dichiarano professionisti).

Lo speciale sulla storia dei giochi da bar, se anche non una esclusiva, che ha occupato ben quattro puntate è lo sforzo forse maggiore della rivista e un pezzo di inimmaginabile godimento per chi la storia dei giochi da bar l'ha vissuta in prima persona.

Il sottoscritto ad esempio ha iniziato la propria carriera lavorativa alle dipendenze di un noleggiatore. Si aggiustavano e si costruivano assemblando i vari pezzi: scheda, monitor, joystick, bottoni... per poi "metterlo giù" in qualche posto dove si sapeva che poteva incassare parecchio...

Ah tragico destino... Perché, perché sei morta Still-Alive?

[sc]

Una rassegna dell'editoria specializzata dai primi anni '80 ad oggi

Scheda

Titolo: Still-Alive

Editore:

Web:
www.stillalive.com

Retro-Code

cioè il software quando
era giovane

Essere un
programmatore non
significa conoscere
un linguaggio o anche
più linguaggi di
programmazione

Apple Integer Basic

Breve storia del BASIC

Il linguaggio BASIC ha avuto una fortuna per certi versi inaspettata nel campo della programmazione dei calcolatori. Al tempo dei sistemi di calcolo dipartimentali (mainframe o mini), l'idea che la programmazione potesse essere semplice faceva inorridire gli addetti ai lavori che, da che mondo è mondo, come ogni casta tende a proteggersi ammantandosi di non so che aurea di elezione.

Quando si è presentato il problema di allargare il numero di programmatori per effetto della domanda crescente da parte del mercato, le università si sono trovate davanti l'ostacolo dei linguaggi di programmazione pensati per specialisti e poco adatti alla divulgazione del "verbo".

Qualcuno ha cominciato così a pensare a qualcosa di molto più abbordabile dalle masse di un FORTRAN o un PL/M. La chiave di volta è stata capire che serviva un linguaggio per imparare e che tale linguaggio non doveva necessariamente arrivare a costituire un tool di produzione.

Essere un programmatore non significa conoscere un linguaggio o anche più linguaggi di programmazione, significa essere in possesso delle conoscenze fondamentali, dei principi. La sintassi è un "di cui...", un mezzo per dare forma alle idee, non un fine.

Personalmente trovo assurde e anche irritanti, lo confesso, le guerre di religione su quale sia il linguaggio migliore o magari "più moderno". I saggi

dicono semplicemente che: "Bisogna vedere caso per caso...", è vero ma è una considerazione pratica che si ferma alla superficie delle cose.

Buona parte dei programmatori "ad oggetti" hanno una vaga idea del polimorfismo che accomunano alla ridefinizione dei metodi nelle classi derivate mentre mi è capitato di sentire anche perle del tipo: "Un oggetto e' una specie di classe...". Pazienza la poca precisione ma quello che lascia sconcertati è che non ci sia un approccio alla programmazione ad oggetti con l'intento proprio di creare delle classi di uso generalizzato, ci si ferma al proprio piccolo problema.

Bando alle ciance torniamo al BASIC che come si diceva si è proposto come tool di istruzione sopportando le beffe dei "veri" programmatori che a colpi di definizioni di campi Hollerit se ne andavano per i corridoi della facoltà trasportando il loro bel pacchetto di schede perforate.

Contrariamente all'idea iniziale il linguaggio ha trovato una sua strada che l'ha portato prima sui piccoli calcolatori personali, gli home computer, costruiti a partire dai primi anni '80 e successivamente fino alla programmazione "nobile" delle interfacce grafiche. Il più famoso tool è naturalmente il Visual Basic di Microsoft ma ne esistono anche altri sia sotto Windows che sotto Linux o MAC che svolgono onestamente il loro lavoro.

Una delle prime implementazioni del BASIC su un microcomputer è questo Integer Basic che la Apple inserì nella

(Continua a pagina 35)

ROM del modello][. Si tratta di una sorta di ripiego: la maniera per cavar-sela onestamente mettendo a disposizione degli acquirenti uno strumento per programmare (altrimenti perché avrebbero dovuto acquistare il calcolatore? Mica c'era il software già pronto!), senza dover combattere con l'aritmetica reale. Ciò che noi diamo oggi per scontato e cioè la possibilità per un linguaggio di gestire valori numerici reali, pone dei grossi problemi di implementazione su una CPU che non ha istruzioni specifiche per farlo. Ci vuole tempo, idee e soprattutto spazio nella ROM, ma le ROM costano molto...

L'Integer Basic di Apple.

Quando i sistemi erano tutti proprietari, il sistema operativo e il linguaggio di programmazione (che era sempre un qualche dialetto Basic), andavano di pari passo. Quello che oggi chiamiamo BIOS (Basic Input Output System) veniva normalmente indicato come "ROM di sistema" e conteneva le routines di base, a volte un assembler, un monitor molto semplificato e quasi sempre un Basic.

La Apple, se voleva vendere il proprio Apple][, doveva per forza dotarlo di un linguaggio di programmazione, anche semplice, ma abbastanza potente da soddisfare la voglia dei primi smanettoni. Così Steve Wozniak si mise di buzzo buono e con l'aiuto di qualche amico/dipendente, mise su quello che venne chiamato Apple Integer Basic.

Come dice il nome si tratta di un interprete privo delle estensioni per virgola mobile e di conseguenza di tutte le funzioni che lavorano sui numeri Reali.

Primi passi e primi comandi.

Dopo il bootstrap l'Apple si presenta in modalità "monitor" con il prompt rappresentato da un asterisco ("*") e cursore lampeggiante. Premendo Control-B si entra in Integer Basic, rappresentato dal prompt ">". Per tornare al monitor si usa il tasto RESET.

Entrare in Integer Basic con il Control-B provoca la cancellazione del programma eventualmente presente e la predisposizione delle due costanti che determinano i limiti della memoria usabile dal Basic. Tali contatori sono chiamati HIMEM, che viene settato alla massima memoria disponibile, e LOMEM che ha valore iniziale 2048. Questi due valori si possono variare per riservare aree di memoria ad altri scopi, ad esempio a routine assembler.

Il comando Control-C permette di rientrare in Integer Basic se il sistema è in stato di monitor, senza intervento sul programma eventualmente presente; se ci si trova in Basic il comando provoca l'interruzione del programma in esecuzione con il display della linea dove l'interruzione è avvenuta (il run del programma si può continuare con il comando CON).

L'Integer Basic dispone di comandi e statement. I comandi sono le istruzioni che si possono digitare al prompt senza numero di riga e vengono eseguiti quando si preme RETURN. Gli statement sono le istruzioni che fanno parte dei programmi; sono organizzati in righe numerate e vengono eseguiti secondo la logica del programma quando si ordina l'esecuzione con il comando RUN. Alcuni comandi si possono inserire anche all'interno dei programmi per ottenere appunto un risultato "programmato".

(Continua a pagina 36)

Comandi.

Esaminiamo ora i comandi disponibili quando ci troviamo a livello Integer Basic (cursore ">"). Fra parentesi quadre i parametri opzionali.

AUTO *num1* [,*num2*]

Autonumerazione delle righe cominciando da *num1* e incrementando di *num2* (opzionale, altrimenti di 10).

MAN

Setta l'autonumerazione delle righe basic su "manuale"

CLR

Cancellazione delle variabili del programma corrente.

CON

Continua l'esecuzione dopo un break (che si ottiene con Control-C).

DEL *num1* [,*num2*]

Cancella la riga *num1* o le righe da *num1* a *num2*

DSP *var*

E' il watchdog della variabile *var*, ogni volta che essa cambia valore viene stampata con il numero di riga che ha provocato il cambiamento.

NO DSP *var*

Cancella il watchdog sulla variabile *var*.

HIMEM: *expr*

Determina il limite superiore della memoria disponibile per il programma. Deve essere usato prima di inserire statement in quanto il comando distrugge il programma Basic esistente. Il valore da inserire in *expr* deve essere espresso in decimale.

LOMEM: *expr*

E' il limite inferiore del programma Basic. Il valore deve essere espresso in decimale. Da usare prima di inserire il

programma.

GOTO *expr*

Salto alla riga definita da *expr*. Viene usato ad esempio dopo un Control-C per determinare la continuazione della esecuzione.

GR

Setta la modalità display definita "mixed" che è costituita da una parte grafica superiore e quattro righe di testo in basso sul display.

TEXT

Setta la modalità testo del display costituita da 24 righe da 40 caratteri ciascuna.

LIST [*num1* [,*num2*]]

Listing del programma, completamente se non sono indicati i valori di inizio (*num1*) e fine (*num2*).

LOAD *expr*

Carica un programma da cassetta. Viene emesso un primo beep quando l'inizio del programma è trovato sul nastro (devono passare pochi secondi, pena l'emissione di errore) e un secondo beep al termine del caricamento. Se qualcosa va storto si vedrà un messaggio di errore. Nota: il registratore va fatto partire a mano prima di premere RETURN.

SAVE

Salva il programma su cassetta.

NEW

Cancella il programma corrente. I valori di HIMEM e LOMEM rimangono inalterati.

RUN [*expr*]

Lancia il programma presente in memoria, eventualmente cominciando dalla riga *expr*.

Essere un programmatore non significa conoscere un linguaggio o anche più linguaggi di programmazione

(Continua a pagina 37)

TRACE

Stampa a video il numero di riga in esecuzione.

NO TRACE

Off del comando TRACE.

Controllo del cursore.

L'editing a video e in generale il controllo del cursore si ottengono con combinazioni di tasti piuttosto scomode, tanto è vero che sono proliferate le utilità che permettono l'editing delle righe Basic con una certa dose di maggiore agevolezza.

Alcuni comandi di controllo si ottengono con il CONTROL, altri pigiando e rilasciando il tasto ESC seguito poi dalla lettera del comando.

CONTROL-H

Cancella il carattere a sinistra del cursore. Si tratta di una concessione alla modalità Unix che usa questa combinazione per cancellare. Apple dispone in realtà del più comodo tasto Backspace che fa lo stesso lavoro.

CONTROL-U

Cancella il carattere a destra del cursore.

ESC A cursore a destra

ESC B cursore a sinistra

ESC C cursore in basso

ESC D cursore in alto

ESC E

Cancella dal cursore fino a fine riga

ESC F

Cancella dal cursore fino a fine pagina

ESC @

Muove il cursore in cima allo schermo, cancella tutta la pagina.

Operatori.

Le regole per la formulazione delle espressioni sono quelle ormai classiche: priorità e raggruppamento con parentesi sono disponibili per la formulazione corretta delle espressioni matematiche e per l'assegnazione dei valori alle variabili.

Esempi:

10 X = 4*(5 +1)

30 ALPHA = -(BETA * 2)

40 LET Y = 12 * A2 - 3

Nota: l'ultimo esempio significa:

Y = (12 * A2) - 3

Ovviamente esistono le quattro operazioni *, /, +, -. E' presente l'operatore MOD per calcolare il resto della divisione come in:

10 X = 12 MOD 7

Che restituisce 5 nella variabile X.

L'elevamento a potenza:

20 Z = X ^ 2

Che significa X al quadrato. Nota: il simbolo di elevamento a potenza si ottiene sulla tastiera con SHIFT+N.

Da questi esempi si deduce che l'assegnazione tramite l'operatore "=" è possibile senza usare lo statement LET (che sarebbe lo standard Basic per l'assegnazione).

Per questa puntata ci fermiamo qui. La prossima volta affronteremo l'esame delle istruzioni Basic e le regole delle espressioni e degli operatori aritmetici e logici.

[tn]

Biblioteca

Sugli scaffali della biblioteca vecchie e nuove monografie denunciano la nostra passione.

Scheda

Titolo:
Digital retro

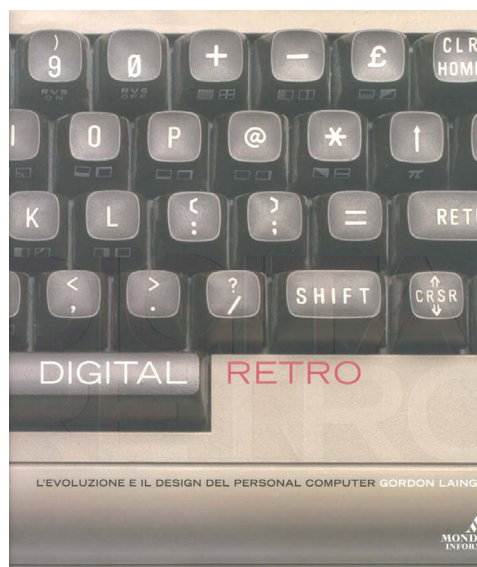
Sottotitolo:
L'evoluzione e il design del Personal Computer

Autore:
Gordon Laing

Editore:
Mondatori Informatica

Prezzo: 25 Euro
Pagine: 190
Lingua: Italiano
Anno: 2004

ISBN:
88-0453642-X

Digital Retro

Gordon Laing, autore di questo volume, è un giornalista e scrittore che si occupa di informatica personale dal momento (come afferma lui stesso nelle sue brevi note bibliografiche) in cui ricevette in regalo dai suoi genitori uno ZX80 della Sinclair). Come spesso accade a chi ha vissuto in prima persona i tempi pionieristici fra gli anni 75 e 90 del secolo scorso, anche il nostro ha continuato a conservare curiosità ed affetto per i sistemi di calcolo "prima maniera".

Il volume offre una esauriente, anche se non completa, panoramica sui sistemi di calcolo personali venduti fra il 1975 e 1988 a cominciare dall'Altair 8800 per finire con il Next Cube.

A parte una introduzione molto sintetica sulla storia dell'informatica e una conclusione che si focalizza sulla rete Internet piuttosto che sull'evoluzione hardware dei sistemi, il volume raccoglie in ordine cronologico le schede dei computer commercializzati con più o meno successo. In accordo con il titolo e con la dimensione del volume, cioè la classica dimensione dei volumi di

fotografie e architettura, la presenza di immagini è largamente predominante rispetto ai testi. I sistemi sono fotografati da ogni lato oltre che nei particolari, rendendo la lettura molto piacevole e rilassante.

Dobbiamo riconoscere all'autore una rara dote di sintesi che si manifesta nella presentazione della storia del singolo sistema e nelle vicende della società costruttrice. Una giusta dose di importanza è riservata agli uomini protagonisti della rivoluzione informatica, ai loro sogni e alle loro aspirazioni. Curiosamente sono quasi sempre storie di difficoltà economiche pesanti, risolte da colpi di genio con l'aiuto fondamentale della fortuna. Situazioni apparentemente "da manuale" nell'economia statunitense. Questo suggerisce che chissà quali e quanti altri sistemi sono stati progettati senza poi sfociare in una vera e propria distribuzione.

L'autore è un vero appassionato di retro-computing, tanto che probabilmente (anche se non viene detto esplicitamente) i sistemi fotografati sono di sua proprietà. Che si tratti di oggetti reali e non di trasposizioni di foto originali delle case costruttrici, lo dimostrano alcuni piccolissimi particolari che rivelano l'uso del sistema: piccoli graffi sul cabinet, etc... A parte queste minuzie tutte le macchine sono in splendida forma tanto da chiedersi quanto le foto non siano state ritoccate o quale magico prodotto il proprietario usi per conservare la brillantezza delle plastiche originali (quelle Apple tendono inesorabilmente ad ingiallire).

Il retro-computing, diffuso anche in Italia, trova terreno fertile oltre oceano dove le macchine sono state costruite e vendute in migliaia, se non

(Continua a pagina 39)

milioni, di esemplari. E' quindi relativamente più semplice per un appassionato statunitense costruirsi una significativa collezione anche se deve probabilmente lottare contro la tendenza ultra-consumistica degli americani che li porta a distruggere con molta facilità il materiale che non utilizzano più.

Oltre alla storia del sistema e della società costruttrice, il volume contiene piccole tips relativamente ai protagonisti e una opinione del perché la fortuna del sistema sia ad un certo punto terminata. Queste notizie sono raccolte direttamente dalla voce dei protagonisti nel corso della carriera di giornalista specializzato dell'autore Laing. L'autore peraltro non nasconde la sua predilezione per i sistemi dalla Apple Computer che viene presentata come la vera innovatrice del mercato; opinione che non può che trovarci d'accordo in pieno! Dall'Apple II del 1976 all'ultimo PowerMac G5, la società di Cupertino ha sempre stupito per la qualità dei prodotti e per il livello di innovazione.

L'appassionato italiano troverà schede di computer dei quali non ha mai visto un esemplare e forse non ha nemmeno sentito della loro esistenza, questo perché la diffusione dei sistemi e anche delle riviste specializzate è avvenuta in Italia con un certo ritardo e con molta minore penetrazione. Un'altra considerazione generale che si trae dalla lettura è la contrapposizione fra la filosofia statunitense, votata alla produzione standard di massa e la particolarità europea (limitata alla sola Inghilterra per la verità) che ha puntato molto più sulla fantasia, basta pensare ai sistemi Sinclair per convincersene.

Il volume è proprio perfetto così com'è per lo scopo che si prefigge. Come appassionato di tecnica avrei accolto favorevolmente una scheda più completa sulle caratteristiche hardware piuttosto che le quattro notizie in croce su CPU e memoria RAM disponibile.

Del tutto assente il software, sistemi operativi e linguaggi compresi, mentre buona parte del successo dei sistemi è stato dettato proprio dai programmi disponibili: non solo il VisiCalc citato dall'autore, ma il sistema operativo CP/M è stata una delle chiavi del successo della diffusione dei sistemi di calcolo personale. Anche l'importanza della CPU Z80 non è considerata nella giusta misura a mio giudizio. Lo stesso fenomeno degli MSX viene liquidato in un'unica scheda riassuntiva mentre la loro importanza avrebbe dovuto comportare una più ricca collezione di sistemi.

Le calcolatrici programmabili sono solo citate mentre trovano spazio le prime console di giochi: una scelta discutibile. L'unico altro dubbio che solleva è la fotografia dell'Apple II presentato con un monitor originale Apple che secondo me è stato commercializzato solo con la serie II; nulla di male, ma l'autore premette che tutti i sistemi sono fotografati nella prima versione originale apparsa sul mercato.

Conclusioni.

Bello, niente da dire. Il prezzo di 25 Euro non è proprio economico ma la qualità della pubblicazione lo giustifica, comunque già si trova nei remainder a prezzo dimezzato.

Senza ombra di dubbio è una monografia da possedere per chi ha la passione del retro computing o per chi, pur non coltivando l'hooby del retro-computer, percepisca una certa nostalgia per un'epoca che fu e volesse rinvivere qualche passata emozione sfogliandone le pagine.

[sn]

Dall'Apple II del 1976 all'ultimo PowerMac G5, la società di Cupertino ha sempre stupito per la qualità dei prodotti e per il livello di innovazione.

Jurassic News anteprima

nel prossimo numero...

In prova il Micro Professor II.

Viene dalla taiwanese Microtek un sistema con spiccate propensioni didattiche ma che non rinuncia a gettare uno sguardo verso l'impiego semi-professionale.



```
Softcard CP/M
60K Ver 2.23
(c) 1980,1982 Microsoft

A>dir b:
B: BASCOM COM : BASLIB REL : BAZIC COM : BASIC80 COM
A>dir b:
B: DBASEUPD OVR : DBASESRT OVR : DBASEBRO OVR : DBASEMAI OVR
B: DBASERPG OVR : DBASEMOD OVR : DBASEAPP OVR : DBASEII DOC
B: ZIP COM : VIDNI FRM : INSTALL COM : DBASE COM
B: DBASEMSG COM : DBASEMSC OVR : DBASEJOI OVR : DBASETTL OVR
B: LOADDB SUB : DU-V75 COM : DOT65 COM : AUTORUN COM
B: DBSET CMD : A DBF : SCR38:B DBF : CATALOG DBF
A>b:
B>dbase

ENTER TODAYS DATE AS MM/DD/YY OR RETURN FOR NONE :
*** dBASE II Ver 2.3B 22 FEB 82

EVALUATION VERSION
```

La seconda puntata della prova del sistema operativo CP/M e la seconda parte dell'articolo sulla realizzazione di un emulatore.

La seconda parte del corso dedicato all'Applesoft.

E inoltre: Una nuova rubrica dedicata alla recensione del software.

La recensione di una rivista "storica" e la rubrica Biblioteca con la presentazione delle monografie "da avere assolutamente"