Riparazione floppy Sinclair ZX Spectrum+3

by Davide (www.zxspectrum.it)

Se avete uno Spectrum +3 ed avete dei problemi di funzionamento con il floppy disk al 99% lo potete riparare senza grossi problemi: Questo "stupendo" lettore da 3 pollici by Amstrad ha un difetto molto comune, il disco viene ruotato da una puleggia azionata da un motorino elettrico tramite una cinghia, il problema nasce proprio da questa cinghia, e' di un materiale poco elastico e specialmente dopo tutti questi hanno puo' cedere. I drive che ho aperto io infatti due avevano la cinghia ormai lente e quindi il morino girava ma non la puleggia invece uno aveva la cinghia rotta. Ma adesso iniziamo l'intervento: primo leviamo il drive dal computer, niente di difficile..., una volta tolto il drive occorre rimuovere: frontalino del drive tramite due viti che lo fissano; schedina elettronica ci sono tre viti, poi occorre staccare tramite gli appositi connettori ad incastro due collegamenti di fili: uno per il motorino l'altro della testina; a questo punto occorre svitare altre due viti che fissano i due pezzi di un sensore del disco. A questo punto la schedina elettronica puo essere rimossa.

Occhio alle viti che hanno dei passi diversi quindi non le mischiate fra di loro.

Fate attenzione quando levate la schedina dalla sua sede a non perdere un piccolo perno di metallo che aziona un piccolo microswitch presente sulla schedina.

A questo punto potete vedere la cinghia in tutto il suo splendore: noterete che e' a sezione rettangolare e purtroppo poco spessa, Io l'ho sostituita con una guarnizione o-Ring a sezione tonda di medesima lunghezza dal costo di 1 euro.

Riassemblato il tutto il drive adesso funziona alla perfezione!!

L'intervento è applicabile pari pari al floppy degli Amstrad (ma và??) alle piastre degli HI-FI e ai Walkman più vecchi.

Per i floppy esterni degli Atari la cinghietta è piatta e va ordinata a lunghezza, quindi non buttatela subito, ma cercate di misurarla per poi andare nel negozio di riparazione più vicino a prenderla. Prezzo intorno alle 3K.

Se volete essere strasicuri della durata eterna dell'O-ring o se dovete usare molto spesso la piastra dell'HI-FI, ad esempio, chiedete al negoziante se vi da quelli al silicone. In campo automobilistico sono rossi, ma comunque sono sempre colorati diversamente dal nero di quelli comuni. Costeranno anche di più, ma non dovrete mai più rifare l'intervento.

ALCUNE DOMANDE E RISPOSTE:

Finalmente mi sono arrivati 5 floppy HD con frontalino nero, attesi pazientemente per mesi, per attrezzare decentemente Disciple, Plus D e Spectrum +3. Ricordo vagamente di aver letto qualcosa contro l' utilizzo dei cavi floppy per picci' per indirizzare correttamente i due drive. Qualcuno si ricorda al volo se ci sono controindicazioni o no? Potrei andare a cercare il pinout dei floppy per verificare da me ma sono diventato pigro e sono a corto di tempo libero...

Risposta

Effettivamente il collegamento dei floppy disk nei computer anni 80 rispetto ai PC odierni e' cambiato: prendendo come standard i drive esterni dei BBC Shugart veniva utilizzato un cavo dritto a 34pin fra il controller ed il/i disk drive. Si potevano collegare fino a 4 drive e la lettera al drive veniva dato appunto da un jumper (pin 10 Drive A, pin 12 Drive B, pin 14 Drive C, pin 6 DriveC) , notare che quando il controller doveva accedere ad un drive "tutti" i drive giravano il motore, questo perche' il segnale era comune per tutti (pin 16 Load head[Motor On]). Quindi i segnali interessati sono cosi mappati:

PIN 06Drive Select D

PIN 07Ground
PIN 08Index
PIN 09Ground
PIN 10Drive Select A
PIN 11Ground
PIN 12Drive Select B
PIN 13Ground
PIN 14Drive Select C
PIN 15Ground
PIN 16Motor Enable

Con i PC "moderni" e' invece diventato standard usare "solo" due drive (faccio notare che il BIOS di alcuni primi PC permetteva l'uso di 4 drive) lasciando impostato sul disk drive il jumper dell' unita' sul drive B (DS1), e facendo selezionare il tipo drive " rigirando" il flat del cavo dal pin 10 sul 16 e viceversa dal 16 al 10 per poter selezionare il drive A. Inoltre viene separato anche il segnale del Motor On fra i due drive.

PIN 10Motor Enable A
PIN 11Ground
PIN 12Drive Select B
PIN 13Ground
PIN 14Drive Select A
PIN 15Ground
PIN 16Motor Enable B

Personalmente ho usato per collegare all interfaccia PlusD e Betadisk dei moderni drive da 1.44 della ALPS utilizzando un cavo a 34 pin dritto, poi ho aperto il case del floppy disk drive ed ho visto che c'era la possibilta' di selzionare l'ID del drive tramite un ponticello a "stagno" tra DS0 DS1, se i tuoi drive non hanno questa possibilita' devi impostare il drive rigirando il cavo in modo da invertire il segnale del drive A/B ma NON invertire il segnale del pin 16 (Motor On)

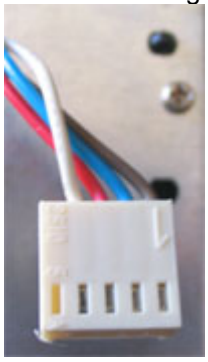
COME COLLEGARE UNA PRESA EAR E MIC ALLO SPECTRUM 128K +2A

by ZX Magazine n.7 2004 (http://www.speccy.org/magazinezx)

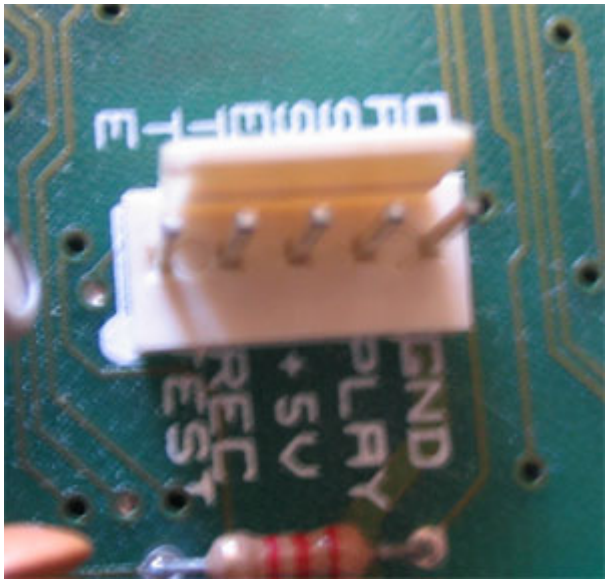
Il +2A non possiede tali porte di in/out avendo il registratore a cassette gia' incorporato sulla tastiera. Capita pero' che sia necessario collegare un registratore esterno (qualora per esempio si danneggiasse quello "on board"). Per fare cio' e' necessario disporre di:

- 2 cavi mini jack da montare sullo chassis
- filo conduttore
- stagno
- trapano (per perforare il case)

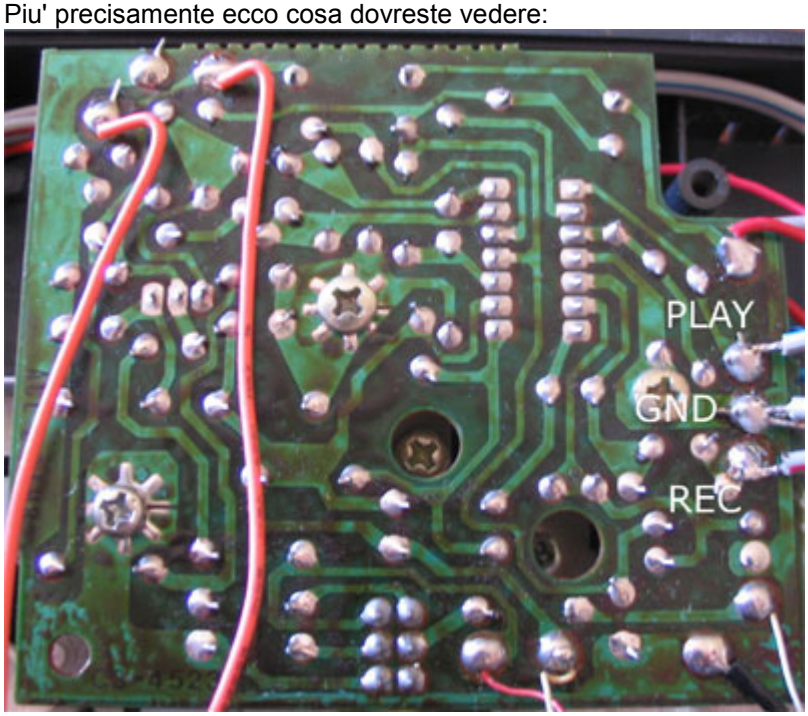
Apriamo con cura il case stando attenti a non danneggiare la membrana.
Va cercato il seguente filo che collega il registratore a cassette con la scheda madre.



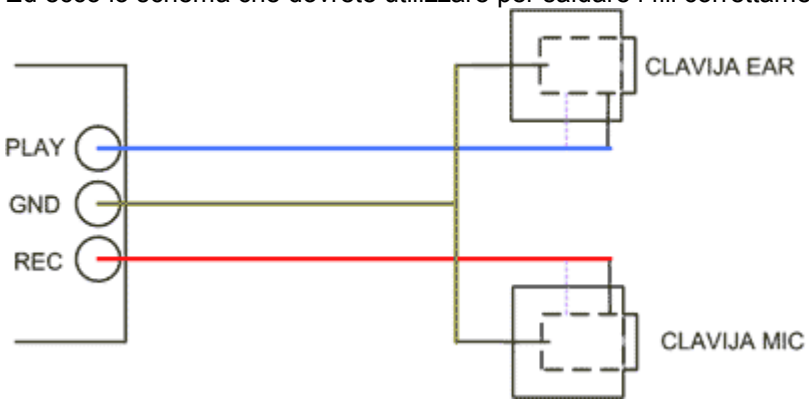
Questo e' il connettore



In ogni caso se vi foste dimenticati il verso dello spinotto ecco cosa rappresentano:
marrone = massa
azzurro = Play
rosso = +5V
bianco = Rec
Ovviamente il segnale bisognera' prelevarlo da tale connettore. Pertanto sara' necessario guardare il retro della scheda madre e collegare alle saldature del connettore i fili che andranno ai jack che aggiungeremo.
Piu' precisamente ecco cosa dovreste vedere:



Ed ecco lo schema che dovrete utilizzare per saldare i fili correttamente:



Riparare uno ZX81 od uno Spectrum

by Gianni "bbk" (http://www.bbk.org)

Appoggia la faccia sulla plastica dello ZX che avendo componenti passivi vecchio tipo tende a frusciare. Se lo lasci acceso qualche minuto sentirai la plasticona scaldarsi (sopra alla MB c'e' un bel lamellone dissipante). La tastiera fa dei piccoli "tic" attraverso il beeper quando preme.

Se non senti i tic, ma fruscia e scalda allora e' impiallata la macchina (comunemente si blocca con una schermata di quadratini visibile a video, ma non e' detto). Se e' cosi' passa la punto A :)

Se senti i Tic allora potrebbe essere il modulatore "stanco" o "scivolato". In questo caso passa al punto B

Se non senti Tic, non scalda e non fruscia allora e' il comunissimo problema dei connettori e cavi Sinclair. In questo caso passa al punto C

13.1.1. Punto A

Se la macchina si accende ma non "opera" (niente Tic) conviene intervenire cosi': aprite la macchina con ENORME attenzione ai flat della membrana. ATTENZIONE: di primo acchito viene l'impulso di rimettere dritte e levare le brutte pieghe che le flat della tastiera prendono nel tempo. NON FATELO! staccatele dalla piastra e mantenete il tutto allo stato in cui l'avete trovato.

Inutile che dica di soffiare e pulire bene la piastra, dopo di che premere (con dolcezza e decisione) gli integrati zoccolati.

Provare ad accendere, SENZA TASTIERA (meno la tokkate e meglio e'). Se funziona rimontate il tutto. Se non funziona sono "caxxi fernet". :))

Di norma il guasto piu' comune e' l'ULA (chippone ferranti), trovare il ricambio e' una rogna, ma se spargete la voce magari si rimedia. Impossibile trovarlo nuovo perche' e' fatto appositamente per i Sinclair e quindi la Ferranti non ci pensa proprio di produrli ancora (forse e' pure chiusa).

13.1.2. Punto B

Aprite la macchina con ENORME attenzione ai flat della membrana. ATTENZIONE: di primo acchito viene l'impulso di rimettere dritte e levare le brutte pieghe che le flat della tastiera prendono nel tempo. NON FATELO! staccatele dalla piastra e mantenete il tutto allo stato in cui l'avete trovato.

Il modulatore degli ZX non e' proprio il massimo della vita (e' quello scatolotto di metallo in alto a sinistra) quindi muore (alias si rompe) oppure "scivola" di frequenza. Abitualmente io accendo la macchina, metto il televisore sul canale 36 e poi, con un piccolo cacciavite (meglio se di plastica) giro il trimmer fino a sintonia trovata.

NB: e' abbastanza normale trovare un piccolo "filo" di gomma che esce dal trimmer, credo servisse proprio a mantenerlo fermo, se lo perdete o non l'avete non preoccupatevi, vorra' dire che dovrete regolarlo di nuovo tra qualche anno.:)

13.1.3. Punto C

Dato per scontato che l'alimentatore funzioni (quindi scalda) i problemi piu' comuni si concentrano sul cavo che dal trasformatore vanno fino al computer.

I due problemi piu' comuni in assoluto sono:

- Il connettore guasto o con falsi contatti Una grossa rogna perche' magari con il tester funziona, ma come lo riinfilate la posizione del cavo cambia e non funziona piu'. Infili il cavo e provi a far fare le curve piu' assurde al cavo (tenete presente che potrebbe anche essere interrotto a meta', ma non e' comunissimo)
- La "presa" in cui infilate il connettore ha "preso gioco" o e' ossidata. L'ossidazione e' facile da capire: una patina bianca e rugosa sulle parti metalliche interne. Antiossidante a secco (se vi va bene:)), sostituire la presa oppure collegare una presa "a filo" esterna.

  > Se il connettore ha preso gioco potrebbe essersi dissaldato o essersi rotto uno dei due poli e quindi : Aprite la macchina con ENORME attenzione ai flat della membrana. ATTENZIONE: di primo acchito viene l'impulso di rimettere dritte e levare le brutte pieghe che le flat della tastiera prendono nel tempo. NON FATELO! staccatele dalla piastra e mantenete il tutto allo stato in cui l'avete trovato. Estraendo la scheda dal guscio inferiore controllate le saldature all'altezza del connettore e guardate con attenzione tra il connettore e la piastra sollevando leggermente il connettore. Ho imparato a mie spese che saldare un connettore femmina su un cavo che esce dallo speccy evita un sacco di problemi d'estrazione involontaria della macchina dalle interfaccine. Magari con un bell'interruttore a meta':)

13.1.4. Consigli Utili

- La membrana, da NON fare:
  - NON aprire la macchina piu' del necessario
  - NON fare NULLA con la tastiera attaccata e la macchina aperta (tipo prove di caricamento o cose simili)
  - NON raddrizzare le pieghe delle flat
  - NON stirare con il ferro a vapore (vi giuro che l'ho visto!!)
  - NON appoggiare cose sulla tastiera per lunghi periodi (chi ha messo lo speccy in mezzo ad una catasta si e' giocato la membrana)
- Espedienti e riparazioni : La membrana si rompe d'abitudine all'altezza delle pieghe, quindi cercate li' se non funziona.
- Prevenire e' meglio che curare: la membrana ama l'umidita': il modo migliore per preservare la vostra membrana e' di non tenerla al caldo secco (il fresco umido della cantina e' perfetto. peccato che uccida l'elettronica:)). In caso recuperiate uno ZX con membrana funzionante la PRIMA cosa da fare e' inumidire i flat (da farsi SOLO se e' assolutamente necessario aprire la macchina, altrimenti aspettate quando lo sara'). NON FICCATELI SOTTO L'ACQUA:), il sistema piu' cauto e' metterli sopra a del vapore d'acqua (c'e' chi ha usato l'aerosol con un emoliente, ma magari e' una leggenda metropolitana:)). Il sistema piu' radicale e' il vaporetto ponti (quello piccolo!!!) con uno straccio di cotone davanti alla bocchetta. E' inutile (o quasi) vaporizzare dall'esterno verso la tastiera (specie negli ZX81:)) perche' la gomma dei tastini e' in un pezzo unico e quindi non passa vapore verso la membrana. Non l'ho mai provato, ma pare che ripassare le piste con un pennarello a base metallica (uniposta zinco o cromo) abbia ottime probabilita' di prevenire rotture.

- Riparare: c'e' chi ha provato i mezzi piu' assurdi (ho visto uno speccy con dei fili SALDATI sulla flat rotta), in linea di massima i piu' quotati sono : Pennarello al metallo per ricongiungere le piste crepate, sottili cavi di rame da wire wrap e sottilissime strisce di pellicola d'alluminio. Cercate in rete eventuali immagini.
- Conservare: se per qualche fortuito caso doveste trovarvi con delle membrane in piu' e' bene conservare in un posto umido e non vicino al camino:)). Se la membrana e' nuova (quindi piatta) un libro e' perfetto (ho dovuto rubare un'enorme divina commedia per la membrana QL che e' lunga un km). Se e' usata una scatola, il massimo sarebbe fermare la membrana con dei giornali mantenendo le pieghe in essere.
- Ultima nota: se doveste cambiare una membrana rotta vi auguro buona fortuna, sullo ZX Spectrum si deve rimuove la parte metallica superiore della tastiera che e' incollata alla plastica, di norma si piega e la vernice se ne va :((((.

Alimentatore stabilizzato 9V

by Davide (www.zxspectrum.it)

Questa modifica l'ho trovata sul numero 29 di Special Playgames. Amici e seguaci dello ZX Spectrum: vi siete mai accorti che il vostro computer fa talvolta degli strani ronzii e surriscalda? Questo non é un problema da trascurare perche' rischiate di trovarvi con un mucchietto di circuiti fusi al posto del computer. Eccovi, dunque, un consiglio utile, e semplice da applicare per evitare tutto cio'.
Controllare l'uscita dell'alimentatore con un tester: se avete piu' di 10V vi consiglierei di attuare una piccola modifica. Si tratta di aggiungere all'alimentatore un circuito integrato stabilizzatore di tensione del tipo 7808 o 7809 con condesatore ceramico.
Modificando cosi' l'alimentatore eliminerete i ronzii e i pericoli di surriscaldamento e vi assicurerete il corretto funzionamento del vostro computer (almeno dal lato alimentazione). Ovviamente questa modifica non vi creera'nessun problema con il collegamento delle varie periferiche. Passiamo subito all'atto pratico.
Procuratevi i seguenti compoinenti:
- circuito integrato stabilizzatore di tensione 7808
oppure 7809.
- condensatore ceramico da 100.000 pF (Picofarad).
- aletta dissipatrrice delle dimensioni della faccia.
posteriore dell'alimentatore.
- LED diodo elletroluminiscente verde o rosso.
- resistenza da 1 Kohm.
- interruttore 220 Volts da 2 Ampere.
Vi occorrerá anche un saldatore stilo da 30 Watt, dello stagno di buona qualita' per uso elettronico, un cacciavite, un trapano elettrico e un cavetto elettrico sottile. Quando vi sarete procurati tutto l'occorrente, iniziate ad aprire l'alimentatore svitando le tre viti poste nella parte inferiore. Vedrete al centro del trasformatore (composto da molte lamelle e avvolto di rame) con davanti due condensatori elettrici (due cilindri grigi con varie scritte). Estraete il fermacavo di gomma dalla sua sede per poter estendere il cavetto che porta l'alimentazione al
computer; poi, praticate un foro nella faccia piatta dell'alimentatore. Montate quindi il circuito integrato stabilizzatore passando una vite attraverso il foro del contenitore del computer.
A questo punto, se siete perfezionisti potrete aggiungere un LED che vi indichi lo stato accensione. Se volete farlo, praticate un foro nel fianco del contenitore vicino alla parte posteriore in modo che i terminali del LED si trovino nella piccola camera posteriore interna del contenitore stesso. Veniamo alla parte elettronica: saldate su tre terminali del circuito integrato degli spezzoni di cavetto lunghi circa 15 centimetri e di colore diverso e fateli passare nell'altro foro dell'aletta dissipatrice.
Se avete montato il LED, saldate sui terminale centrale dell'integrato un capo della resistenza da 1 Kohmn, mentre l'altro capo va saldato sul terminale piu' corto del LED. Saldato ora un pezzo di filo dell'altro terminale del LED al terminale destro del circuito integrato.
Procedete ora a dissaldare uno dei due cavetti che portano la corrente allo Spectrum: dissaldate quello che rappresenta la parte inferiore del circuito stampato.
Saldate qui il cavetto che arriva dal terminale sinistro del circuito integrato. Saldate poi il cavetto collegato al terminale centrale; quindi prendete il condensatore ceramico e saldate uno dei suoi terminali sullo stesso punto di prima, quello segnato con il -; l'altro terminale va saldato al terzo cavetto del circuito integrato e, in tale saldatura, va collegato anche il cavetto che avete saldato prima.

Fatto tutto cio' , siete arrivati al momento cruciale: quello della verifica del funzionamento. Il LED si deve accendere e all'uscita dell'alimentatore il tester deve segnare 8V continui. Se tutto corrisponde, l'impresa e' riuscita!
Ogni possibilitá di fusione é scongiurata e a questo punto non vi resta che accendere il vostro computer e continuare
a divertirvi con esso.
Personalmente ho realizzato questa modifica sull'alimentatore dello spectrum 128+ che é leggermente diverso.Adesso il grosso dissipatore laterale dello spectrum scalda molto meno (mi ricordo che quando ci giocavo d'estate non ci potevo appoggiare la mano, mentre in inverno faceva comodo per scaldarsi!

Immagine video disturbata - segnale debole

by Alberto Ernestini (http://x68000.2ya.com/)

Problemi di questo tipo sono dovuti generalmente al modulatore video.
Si puo' in questo caso usare l'ottima uscita composita che si trova a monte del modulatore stesso.
Bisogna munirsi di un cavo schermato (tipo audio quello delle cuffiette per esempio; lo si trova in vendita al metro), di un connettore scart maschio, di un saldatore e di un po' di pazienza

Lato Scart
I pin da saldare sono:
20 il centrale del cavo schermato
21 la "calza" del cavo schermato

Lato ZX
Aprire il case dello ZX Spectrum 48k ed in alto a sinistra si trova il modulatore. Vanno trovati 2 fili rigidi che partono dallo stampato.
Uno di questi entra nel modulatore ed e' isolato dal telaio di metallo da un'anello di plastica.
Questo e' il filo del segnale che va saldato sul centrale del cavo schermato. La calza si puo' saldare da qualsiasi parte sul case di metallo del modulatore.
Se il segnale risulta ancora disturbato va tagliato il filo che entra nel modulatore (facendo in modo che si possa un giorno risaldare).
Per quanto riguarda l'audio collegate due fili sul BEEPER dello spectrum, il segnale si capisce perchè sopra sul circuito componenti c'è un puntino arancione/giallo che sta indicare che è il segnale audio l'altro è la massa, collegate il segnale beeper con PIN 2 o PIN 2 e 6 insieme, per avere un segnale audio migliore, in fine la massa del beeper sul PIN 4.

SCHEMA ELETTRICO SCART:
PIN 6 AUDIO LEFT INPUT
PIN 2 AUDIO RIGHT INPUT
PIN 4 MASSA AUDIO
PIN 17 MASSA VIDEO
PIN 20 VIDEO INPUT (andra' come detto al centrale del cavo schermato)
PIN 21 CALZA CAVO SCHERMATO
PIN 8 +12 Volt
Il collegamento del Pin 8 a +12 V e' opzionale. Serve soltanto per posizionare il televisore su AV all'accensione.

Commenti di una persona che ha provato questa soluzione

Ho modificato come da istruzioni pubblicate l'uscita video, dalla mediocre RF all'ottimo video composito e senza fare fori aggiuntivi, ho escluso l'RF dissaldando il capo del componente attaccato al polo caldo del connettore (centrale) e l'ho usato come se fosse un'uscita video composito di serie.
Se un giorno si desiderasse ripristinare il collegamento RF sarà sufficiente ricollegare il componente che un tempo era collegato al polo caldo del connettore femmina del modulatore.

Sono riuscito a far passare un filo conduttore nel foro dove passa il filo rigido circondato dall'anello di plastica, allo scopo ho usato pochi centimetri di filo di un flat cable da 80 vie finissimo, quello che si usa per collegare i moderni hard disk o lettori cd o dvd rom. Visto l'esiguo spazio concessomi ho saldato un capo di questo al polo caldo del connettore RCA femmina del modulatore dello ZX, mentre l'altro, fatto passare nel foro con l'anello di plastica ed in seguito saldato al filo rigido stesso, all'esterno del modulatore.
Quindi ora con un semplice cavo schermato RCA>>RCA o se si vuole RCA>>SCART, se il televisore non dispone della presa video frontale, posso usufruire dell'ottima uscita video composito: l'ho provato su un Sony 29" ed il risultato è stato sorprendente. Ringrazio l'autore per la dritta.
Le foto del mio operato: una presa di fianco e una dall'alto rispetto al modulatore video.

Riparazione membrana

by Davide (www.zxspectrum.it)

Uno dei problemi principali dello Spectrum e' la tastiera.
Generalmente si tratta della membrana che a causa del tempo o dell'eccessivo utilizzo fa si che premendo i singoli tasti o una fila di tasti adiacenti non si ottenga il risultato desiderato.
Il procedimento per sostituire tale membrana sul 48K e' uguale a quello del 128K.
Cosa occorre: spezzone di Cavo flat, nastro adesivo, tanta pazienza!
Prima di tutto occorre trovare il punto di rottura: a volte si consumano nei pressi del connettore della scheda madre, in questo caso basta tagliare 3mm di membrana avendo prima staccato lo spessore di plastica che poi andra' naturalmente reinserito.
Altro punto dove si rompe facilmente e invece nei pressidella tastiera specialmente il flat con i 5 contatti sul 48K gommoso: Infatti il flat deve passare a pelo del modulatore TV e se quando si richiude non ci si assicura che il flat non rimanga sotto di esso, deve fare una piega di 180 gradi e quindi, specialmente dopo un po' di tempo si puo' danneggiare..
Prendete il cavo flat e spellate dei pezzettini di circa un cm e stagnateli; dopo di che preparate un pezzetto di nastro adesivo e sistemate perallelamente sul lato adesivo questi pezzettini equidistanti con il solito passo delle piste presenti sul flat della membrana.
Trovato il punto rotto occore aprire i due fogli con delicatezza ed applicare lo scotch sul lato del piste, richiudere i fogli e applicare due strisce di nastro lungo la costola del flat; cosi' facendo il flat non si apre piu' e diventa anche piu' robusto.
Ricordo nuovamente di stare attenti nel momento della chiusura della tastiera che il flat non vada sotto il modulatore.

Un altro metodo e' quello di usare la vernice conduttiva o in alternativa la carta stagnola (che si potrebbe usare anche in "integrazione" alla tecnica sopra citata a mo' di "saldatura fredda").
Il concetto di base e' lo stesso: le piste della membrana possono sembrare ossidate o consumate e spostate ma normalmente sono ancora a posto. L'interruzione del contatto e' dovuta praticamente sempre alla rottura del supporto in plastica.
Se il problema e' sulla membrana dei tasti, smontate la tastiera, aprendo lo ZX LENTAMENTE e sganciando la tastiera dalla motherboard con MOLTISSIMA ATTENZIONE E DELICATEZZA. Con calma, aprite la tastiera, e liberate la membrana dai tasti (parte gommosa sui vecchi ZX).
L'unico modo affidabile per scoprire dove i tasti sono interrotti e' provare, quindi potrebbe essere una buona idea richiudere lo ZX lasciando accessibile la membrana.
Una volta trovata l'ipotetica interruzione con MOLTISSIMA calma, partendo da un angolo, separate due dei tre fogli "grattando" tra i fogli con un cacciavitino; evitate di spaccare ulteriormente i fogli di plastica (ci vuole tempo).
E' probabile che la rottura sia nelle vicinanze di un tasto (comincerei da Q,A,O,P,M,CAPS SHIFT,0,5,6,7,8...), anche perche' in posizione dei tasti la membrana e' "bucata".
Qui ricostruite il contatto.Il modo piu' valido e' quello di usare vernice conduttiva, ma comunque facciate lavorate "in obliquo" per non sollecitare troppo la membrana.
Per quanto riguarda invece l'utilizzo della carta stagnola si tratta di collegare i punti interrotti con delle piste fatte con il cuki aluminio.
Prima di tutto occorre vede dove e' la rottura della pista. Se la rottura e' evidente tipo flat spaccato in due non disperate almeno non perdete tempo a cercare dove e' rotto.
Il problema puo essere quando ci sono dei falsi contatti in punti apparentemente sani.
Escludete per primo il flat sano, se il problema e' su un rigo o piu di tasti e' il flat a 8 piste, mentre se e' una colonna si tratta di quello di 5 piste. Di solito ho trovato rotto il flat da 5 piste, si puo' rompere all' attacco della

tastiera se rimane schiacciato fra il guscio superiore e il modulatore (occhio anche quando poi lo richiudete..) oppure alla attacco della tastiera dove finisce lo spessore che va insieme al flat dentro il connettore, questo si rompe perche' si crea una curva troppo secca dove finisce lo spessore, bisognerebbe "aiutare" il supporto facendolo curvare a raggio costante, e' anche utile se avete un saldatore fra le mani inclinare un po' il connettore in avanti.
Questa zona di rottura si verifica anche sul connettore a 8 piste. Per poter lavorare sulla membrana la dovete levare dal guscio superiore. Il guscio metalico puo' essere fissato con 4 dentini di rame/ottone (occhio a non romperli) prime versioni, oppure semplicemente incollato con lo scotch bioadesivo (e qui occhio a non piegare la lamiera).
Una volta che avete la membrana libera da tutto, prendete il cuki aluminio avvolgetelo alla membrana ed inseritelo nel forno a 150 gradi. Vi ricordo che con il cuki aluminio dovete fare una striscia larga quanto una pista del flat aiutandovi con un righello e trincetto, non e'semplice da fare ma basta avere voglia e pazienza.
Adesso ci vuole dello scotch da pacchi trasparente di quelli larghi 5 cm circa. In un pezzo dalla parte adesiva ci dovete appoggiare le striscie di cuki distanziate con il solito passo delle piste del flat facendovi aiutare con un paio di pinzetteda modellismo. Il pezzo con le piste lo dovete applicare dal lato colloso dalla parte interna del flat dove c'e' il materiale conduttore argentato. I lembi che escono dello scotch girateli intorno al flat.
Quando vi siete assicurati che il collegamento e' stato ripristinato (potete anche provare la membrana al volo senza rimontarla nel guscio) "rinforzate" il flat con lo scotch facendo chiudere le membrane fra di loro.
In precedenza avevo usato dello scotch isolante ma dopo circa 2 anni anche con il caldo dello Speccy mi si e' scollato...Meglio lo scotch da pacchi 3M.

COME COLLEGARE UNA TASTIERA DI UN PC E AVERE 2 PORTE JOYSTICK PROGRAMMABILI

by Giuliano M.

Ecco una scansione di "Sperimentare con l'elettronica e il computer" n.12 del Dicembre 1984 dove spiega come poter autocostruirsi tale interfaccia con tanto di schema circuitale

Tips & Tricks Sinclair QL

by Angelo F. (http://digilander.libero.it/angelmemories/)

Tra i tanti amici che ho conosciuto attraverso internet c'e' ne uno in particolare grazie al quale oggi possiedo due bellissimi Sinclair QL. Oltre a tali oggetti Gianfranco Robiglio, l'amico di cui sopra, mi ha mandato per e-mail tanti "Tips & Tricks" (come si diceva una volta ;) per quella meravigliosa e nello stesso tempo controversa macchina che e' il Quantum Leap.
Mi ha autorizzato a pubblicare tali trucchetti sul mio sito, ed io ho pensato di riproporveli cosi' come sono stati scritti (con qualche mio piccolo intervento per sapere di cosa si parla) poiche' a mio avviso possono risultare estremamente utili anche ad altri appassionati retrocomputeristi...
Buon "smanettamento" ;)
A>ngelo F.

Monitor e varie soluzioni sul Sinclair QL

by Angelo F. (http://digilander.libero.it/angelmemories/)

1.a soluzione: uscita uhf
occorre un normale televisore e se possibile un deviatore RF (tipo videogiochi). Risultati immediati - della serie presto fatto presto disfatto
2.a soluzione: video composito. (dal connettore monitor)

collegare un cavo coassiale ai pins del connettore monitor - mettere in serie un bel condensatore elettrolitico perche' l'uscita video composito porta la continua (3-4 volt). collegarsi alla scart del tvcolor. Risultati : un po' meglio che al punto precedente, qualche problema di sincronismo. Mantenere sempre collegate con un altro filo le masse del tv color e del QL (rischio di rotture)
3.a soluzione: uscita RGB con adattatore. -Soluzione ottimale-
Risultati discreti, paragonabili a quelli del monitor "originale" (a parte la perdita delle colonne 80-85) - occorre un piccolo adattatore con buffer d'uscita (HCT245 e qualche resistenza) verso il tvcolor - soluzione ideale anche per proteggere il chip video del QL che e' piuttosto fragile.. Occorre agire sul delay con qualche circuito, meglio se piu' furbo di quello che ho immaginato io (vedi riviste qitaly n.28...30). Per le colonne andate smarrite: settare opportunamente le finestre (il trucco non vale per i programmi altrui, a meno che non prevedano il settaggio) Perdi l'ingresso Scart del televisore (a meno di non fare commutazioni esterne) Nel circuito puo' tornare utile un interruttore (che disabilita il modo AV e ti permette di vedere la tv in modo normale).
4.a soluzione: Monitor "normale"
Utilizzare un monitor RGB Philips che da' buona risoluzione etc. (il Qfidelity non era il massimo, era quasi meglio il tvcolor mivar 14 pollici). Provare altri tipi di monitor - occhio al blanking che e' (un po') difficile da gestire. I vari monitor (RGB e simili) non sono tutti compatibili, ad esempio i miei amici hanno provato ad usare il monitor QL sull'Amiga ma non sono riusciti a vedere niente (tutta gente pratica e in gamba)
5.a soluzione
per soluzioni piu' avanzate secondo me e' meglio evitare (scheda aurora etc.etc.)

Memoria e programmi QDos per QL

by Angelo F. (http://digilander.libero.it/angelmemories/)

I files del ql sono preceduti da un header (invisibile) di 32 bit - passandoli in msdos l'header e relative informazioni si perdono e, ritornando in qdos andrebbe ricostruito. Puoi quindi "leggere" i file di testo in ms-dos (informazioni ed help) ma non "trasferire" gli eseguibili. Per i file di Quill (_doc) avresti qualche problema "minore": li passi in msdos, li filtri e li leggi come testo...
In Msdos esiste una versione dei 4 programmi psion, che funzionano benino, naturalmente non sono compatibili con i files creati sul ql (occorre esportare ed importare con perdite di informazioni)
Invece per il qdos era stata fatta dalla psion una versione migliorata ed integrata dei 4 programmi (exchange, credo di pubblico dominio) che pero' gira solo sul ql espanso.
Per quanto riguarda i programmi su eprom, (programmi residenti) il pacchetto di programmi deve essere preceduto (si vedono nei primi bytes dello slot) da un identificatore standard (che segnala alla cpu in fase di boot che si tratta proprio di estensioni del sistema operativo e/o programmi utente, etc.etc) se invece il dato dello slot (uno slot=16 kbytes) puo' essere modificato al boot, la cpu prende nota del fatto che si tratta di una ram. Durante il boot la cpu installa i programmi residenti identificati come tali, mette i nomi delle estensioni nella tabella utente e prende nota del valore di ram disponibile. Fa eccezione Tk2_ext che viene solo inizializzato dal boot e poi va "caricato" in un secondo tempo (per evitare eventuali problemi di compatibilita' con altri programmi o comandi).
Non esistono limitazioni di indirizzi come per le cpu intel, ma la ram deve essere continua: percio' avrai a partire dal basso : sistema operativo piu' o meno originale su ROM o Eprom (48k su uno o due ROM) - estensione ROM esterna (16k) - poi inizia la ram (video e memoria normale) estendibile a piacimento dai 128 iniziali, quando finisce la ram (o altrove piu' in alto) puoi mettere degli slot con programmi residenti, tutto fino ad un mega (del 68008) - il bus dati e' a soli otto bit (meno fili da cablare ... vediamola in modo positivo) - il limite di un mega di indirizzi e' vincolante e dipende solo dal limitato numero di piedini disponibili sul dual-in-line del 68008. Ti ricordo che la RAM deve essere continua, se la cpu vede uno slot con programmi residenti si ferma la scansione della RAM...
Per estendere il sistema operativo oltre al gia' citato tk2_ext (16Kbytes) pressoche' indispensabile e ben documentato puoi creare tuoi comandi ed applicativi, sia in basic (def proc) che in formato eseguibile. Il sistema che utilizzavo era quello di fare dei piccoli programmi basic, provarli e poi ... compilarli. Se durante la compilazione scegli questa opzione, l'eseguibile generato e' del tipo "residente" e potra' essere messo su eprom (meglio su e2prom)

Sistema operativo su Eprom per QL

by Angelo F. (http://digilander.libero.it/angelmemories/)

In origine il s.o. e' su due ROM (una da 32k ed una da 16k) - il sistema migliore (e piu' veloce) e' sostituire le due ROM interne e quella esterna (da 16K) con una unica eprom da 64Kbytes in cui metterai il nuovo sistema operativo (minerva) e il toolkit (che serve sempre). A questo punto (vado a memoria...) dovresti solo invertire un filo di comando (serve una porta invertente la ricavi da un hct00 o hct04). Il tutto (una eprom su zoccolo ed un dual in line 14 pins) va allocato su un piccolo pezzo di millefori che verra' fissato su uno dei due zoccoli della piastra madre. (n.b. senza invadere la zona dell'8049 su cui possono essere necessari altri interventi...

Espansione di memoria RAM e E2prom per QL

by Angelo F. (http://digilander.libero.it/angelmemories/)

Per espandere il QL conviene rinunciare per ovvie difficolta' alla soluzione interna e puntare su una piastrina esterna (lato sinistro del ql con prolunghe o diretto). Io mi sono collegato al connettore sinistro con adattatore 64-50 pins (e flat 50 pins) se trovi un flat e connettori 64 pins tanto meglio. Poiche' il clock e' basso (sic) puoi mettere qualche 10-15 cm fra cavi e flat... Ricordarsi che poi bisogna gestire anche i floppy disk (la espansione originale se l'era scordato....). Sulla piastrina troverranno quindi posto la RAM (statica ! , per quella dinamica non saprei dirti come fare il refresh) e i chips con i programmi residenti. Collegare dati e indirizzi - per i segnali di controllo occorre una manciata di integrati. Se sai usare una GAL tanto meglio, risparmi fili e fatica. Mi ricordo anche un piccolo transitor (dta_ck??, poi al momento controllo). Assegna bene le aree di memoria e i vari "slot" da 16k.... La memoria esterna e' anche piu' veloce e i programmi girano meglio (il doppio veloci) Naturalmente devi lasciare lo "spazio" in memoria per la estensione dei floppy (la metti in alto) Per "programmare" on board oltre a vari pgms piu' o meno fatti in casa metti in conto numerosi blocchi di sistema - se il nuovo programma (provato in ram) caricato in e2prom ti blocca il ql devi ripartire da floppy disk dopo aver disattivate le estensioni difettose su e2prom. (Sarebbe bene prevedere uno switch con cui commutare due banchi di e2prom...)

Tastiera e problemi annessi relativi al Sinclair QL

by Angelo F. (http://digilander.libero.it/angelmemories/)

Il controllo della tastiera avviene con una matrice di X x Y fili (non ricordo i numeri esatti) - quando si preme un tasto viene messo in contatto uno dei fili X con uno dei fili Y. Il collegamento viene fatto ai due connettori al centro della piastra madre, ma e' possibile anche collegarsi direttamente ai pins del coprocessore 8049. (ipotesi non del tutto remota) E' sufficiente quindi emulare dall'esterno le varie combinazioni di contatti per avere disponibili tutti i tasti del ql (e togliere qualche amenita' tipo DEL fatto con control frecce) La soluzione "finale" (ma protetta da copyright) consiste in una piastrina (montata a cavallo dell'8049) che utilizza un secondo microprocessore (credo un altro 8049) collegato ad una normale tastiera di PC. Il circuito non e' molto complesso. L'unico problema e' che non avendo il controllo del programma non si puo' configurare il layout di tastiera... (io ho riscritto i caratteri giusti col pennarello dopo aver grattato quelli sbagliati)

Lezioni di linguaggio macchina

IL LINGUAGGIO MACCHINA

Linguaggio macchina,assembler: questa espressione, per coloro che si dilettano a programmare in Basic, Fortran, Pascal ecc. e' spesso causa di disagio e si pensa a qualcosa al di la' della nostra mente, un divertimento riservato solo a pochi eletti. Questo nostro corso non vi dara' ovviamente tutte le nozioni necessarie alla creazione di un programma per giocare a scacchi ma vi dara' comunque una buona conoscenza sulla materia.

COSA E' IL LINGUAGGIO MACCHINA?

La risposta e': un linguaggio come un altro, anzi piu' semplice,perche' le istruzioni di cui si compone sono concettualmente semplicissime. La piu' grande differenza e' che non si scrivono le istruzioni come parole segni di interpunzione e nomi di variabili, ma sotto forma di numeri. Ogni computer ha la possibilita' di venir programmato in L/M anche perche' ogni istruzione di alto livello,per essere eseguita deve venir interpretata e tradotta in routine di L/M dalla ROM.
In L/M potete leggere e scrivere numeri nella memoria, potete fare addizioni,sottrazioni e salti nel programma, condizionati e non. Come esistono vari 'dialetti' Basic, esistono anche diversi codici macchina in dipendenza della CPU che il computer usa. Dato che nel nostro caso si tratta di uno Z80 i riferimenti saranno tutti sul set Z80 di comandi. Tutti gli esempi sono gia' nel vostro Spectrum e vi risparmiamo la fatica di caricarli. Per uniformarci alla maggioranza dei testi i numeri che formano i listati dei programmi in L/M verranno scritti in esadecimale, cioe' in un sistema di numerazione che anziche' avere come base 10 ha come base 16 La scelta della base 16 e dettata dal fatto che e' molto piu' vicina al modo di "pensare" della macchina di quanto non lo sia la base 10. Per adesso ci basta sapere che c9 in hex o 201 in decimale sono la stessa cosa.

COME ACCEDERE AL L/M

Il primo problema e' quello di scrivere qualcosa in l/m; il che significa mettere in qualche punto della memoria questi benedetti numeri. Tale scopo si raggiunge con l'istruzione BASIC "POKE"Se diamo il comando:POKE 25000,201 il risultato sara' che nella cella di memoria 25000 verra' scritto il valore 201. Per controllare sara' sufficiente fare PRINT PEEK 25000 che dara' come risposta 201.Il comando POKE non ha effetto se si tenta di scrivere nella zona di memoria riservata alla ROM (indirizzi da 0 a 16383) Gli amanti delle curiosita' potranno fare il seguente esperimento per vedere come effettivamente i POKE alterino il contenuto della memoria. Caricate un qualunque programmino BASIC e date dei POKE dove, al posto dell'indirizzo, scriverete dei numeri compresi tra 23755 e 24000 e,al posto del valore dei numeri casuali scelti tra 0 e 255. Le locazioni tra 23755 e 24000 sono approssimativamente quelle dove viene memorizzato il listato di un breve programma.Con le operazioni sopra descritte voi lo altererete e il comando LIST vi dara' delle sorprese. Un programma in linguaggio macchina potrebbe dunque essere scritto con tanti POKE,ma questo diventerebbe estremamente lungo e sarebbe facile causa di errori; per questo motivo esistono vari programmi chiamati assemblatori che potete trovare per esempio nella sezione "Load'n'Run" (riviste con cassette) caricando uno dei tanti corsi di linguaggio macchina. In ogni caso bisogna ricordare quanto segue prima di programmare in assembler:
- scegliere un intervallo di memoria dove scrivere il proprio programma che non vada a interferire in zone riservate per esempio al sistema operativo o alla rom
- utilizzare un programma assemblatore che sia in grado di interpretare le istruzioni "mnemoniche" oppure inserire una sfilza di POKE indirizzo,valore
- eseguire tale programma con l'istruzione USR indirizzo di memoria dove si e' scritta la prima istruzione Tale comando (USR) al momento del "RET" (ritorno al basic), restituira' un valore di un registro di memoria (verra' spiegato piu' avanti) in modo da avere una interazione tra: programma in linguaggio macchina e basic.

CONSIGLI

Visto quello che puo' succedere e' buona norma,prima di far girare un programma,salvarlo su nastro; cosi' ,in caso di crash il nostro lavoro sara' solo quello di trovare l'errore invece di dover riscrivere tutto da capo. Notate che il tasto BREAK e' inattivo durante l'esecuzionedi una routine macchina, quindi state attenti anche ai loop infiniti,altrimenti l'unico rimedio per interromperli e'quello di spegnere lo ZX con conseguente perdita di tutto.
C'e' da aggiungere che un programma non termina mai finche' non incontra l'istruzione RET che fa tornare dal codice macchina al Basic.
Un'altra operazione eseguibilee' NOP , abbreviazione di NO OPERATION, la quale,come dice il nome,non fa nulla ed e' paragonabile ad un REM vuoto in Basic. La sua utilita' e' quella di lasciare degli spazi vuoti tra

una istruzione e l'altra,per poter eventualmente fare delle inserzioni,o per fare delle cancellazioni.

COS' E' LA RAMTOP

Quando accendiamo lo Spectrum abbiamo a nostra disposizione 16k 48K o 128K di memoria dove poter scrivere i programmi Basic.
Col comando NEW tutta quest'area viene ripulita e ogni informazione cancellata. E' possibile tuttavia ridurre lo spazio a disposizione per il Basic e creare una sorta di "zona protetta" dove NEW non ha alcun effetto; per far cio' bisogna abbassare il tetto della memoria con le seguenti operazioni da farsi possibilmente prima di scrivere o caricare

CLEAR USR "a"-n

dove n e' il numero di celle che vogliamo siano protette; ora la zona di memoria che comincia all'indirizzo USR "a"-n , e' inutilizzabile per i listati Basic e la gestione delle variabili: ottima quindi per il linguaggio macchina. Un esempio pratico e' il seguente:

CLEAR 29999

A questo punto comincieremo a scrivere il nostro programma assembler dall'indirizzo 30000. Potrete poi dare NEW e cancellare tutto il Basic,ma il l/m sara' ancorali'. Mettere il codice sopra la RAMTOP e' dunque abbastanza conveniente. Per salvarlo si puo'usare:

SAVE "nome"CODE a,b

dove 'a'=indirizzo di partenza 'b'=lunghezza del codice.
Un altra zona dove scrivere i vostri programmi l/m e' in un REM all'inizio del programma Basic. Se sceglierete questa strada l'indirizzo da cui partire sara' 23760,ma prima avrete provveduto a scrivere alla riga 1,un REM con tanti caratteri quanti sono i bytes di lunghezzadel l/m.
Ad esempio se vorrete scrivere un programmino lungo 16 bytes, dovrete prima inserire la seguente riga:

1 REM 1234567890123456

Quando avrete finito di inserire il programma, se darete LIST noterete che alla riga 1 sono successe 'cose strane'. In realta' quello che vedrete non e' altro che il listato-macchina espresso dai caratteri corrispondenti ai codici: controllare per credere.
Col metodo del REM il salvataggio del l/m avviene come per un programma in Basic.
In caso di programmi di una certa lunghezza(>200 bytes),il metodo REM e' sconsigliabile, vista la necessita' di scrivere inizialmente REM molto lunghi.

PROBLEMI DI CRASHING

Quando un programma Basic incontra degli errori (ad esempio variabili non specificate) si e mostra il tipo di errore e il numero di riga a cui esso si verifica. Purtroppo se ci sonodegli errori in un listato/macchina,il piu' delle volte si assiste a un 'crash', cioe' puo' succedere di tutto, inclusa la cancellazione del programma.

I REGISTRI

Ora finalmente possiamo entrare nel vivo della programmazione in l/m: il set di istruzioni dello Z80.
Lo Z80 ha in se' una piccolissima area di memoria dove possono venire memorizzate delle variabili dette registriche si chiamano: A,B,C,D,E,H,L
Tali registri possono contenere soltanto numeri da 0 a 255. Per memorizzare valori piu' grandi,i registri possono essere usati in coppia. Ad esempio se H=64 e L=130, il registro HL contiene 64*256+130=16514.
Gli altri registri combinabili sono BC e DE
Non esiste la possibilita' di operare ad esempio,col registro CD.
Nello Z80 ci sono anche altri registri ma il loro uso e' molto particolare e ne parleremo quando ci capitera' di incontrarli; per adesso mi limito a citare uno dei piu' importanti cioe' il registro PC (Program Counter), che e' il registro dove e' memorizzato l'indirizzo dell'istruzione da eseguire.
Possiamo attribuire dei valori ai registri con l'istruzione LD (abbreviazione di 'LOAD') che e'equivalente al notissimo LET in Basic.
Cosi' LD A,53 significa LET A=53 e LD A,B significa LET A=B
Per esempio,considerate il seguente programma:

```
LD B,0
LD C,53d
NOP
RET
```

Se questo programma viene caricato dall'indirizzo 30000, PRINT USR 30000 da' 53. Questo dopo aver dato ovviamente CLEAR 29999.
LD C,53 occupa solo 2 bytes di memoria contro i 17 di LET C=53,se si tiene conto anche del numero di riga,e del carattere di fine riga.
Provate ora a cercare di capire cosa succederebbe se le prime due istruzioni del programma appena visto venissero sostituite da:

```
LD B,53
LD C, 7
```

quale risultato dara' questa volta PRINT USR 30000 ?
Ebbene il risultato sara' 13575 cioe' (B*256)+C=53*256+7
Questo e' cio' che riguarda i registri singoli o i registri doppi trattando separatamente le due parti che li costituiscono.
Per considerare i registri doppi come un unica entita', si usano 3 bytes: uno e' il codice dell'istruzione e gli altri 2 servono per scrivere il numero.
C'e' pero' una cosa di cui bisogna tener conto e cioe' che il numero va scritto "capovolto"
Per meglio capire il significatodi capovolto sara' utile fare degli esempi.
1) Se vogliamo fornire allo Z80 il numero 1975h (6517d) noi dovremo scrivere 7519
2) LET DE=6517 in l/m si scrive LD DE=1975
3) Se vogliamo caricare in HL il valore 202 (00ca in hex) il codice e' 21ca00 e non 2100ca. (21=codice esadecimale dell'istruzione)
L'operazione di LOAD per il registro doppio puo' anche essere fatta caricando separatamente i due registri corrispondenti,ma cio' e' piu' lungo e comporta uno spreco di memoria.
Per assegnare a BC il valore 16396(400c in hex) si puo' fare:

```
LD B,40
LD c,0c
```

oppure

```
LD BC,400c
```

Come in Basic e' possibile dare a una variabile il valore di un'altra (ad esempio LET x=y),cosi'avviene anche in l/m.
Il risultato, toccando certi registri, potrebbe avere degli effetti imprevisti se non si sa di preciso cosa si sta facendo. Il registro I e' uno di questi e serve per controllare una routine detta di interrupt ma per adesso e' meglio non addentrarci in questo argomento
Per adesso limitiamoci a scrivere un programmino che alteri il valore di 'I' e a vedere cosa succede, anche se per ora non siete ancora in grado di capire il perche'. Ecco il listato:

```
LD A,100d
LD I,A
RET
```

Lo schermo e' confuso da interferenze. Per rimettere al registro I il suo valore originale (9):

```
LD A,9
LD I,A
RET
```

ISTRUZIONI DI LOAD

Sommario
LD r,r' dove r e r' sono uno dei seguenti: A,B,C,D,E,H e L.
LD r,n dove n e' un numero tra 0-255

LD A,(nn) dove nn e' tra 0 e 65535
LD (nn),A

Le istruzioni di caricamento copiano il contenuto di un registro in un altro registro o ad una locazione di memoria, eviceversa. Naturalmente come in tutte le copie l' originale non cambia. L' istruzione di Load e' abbreviata a LD. La forma piu' semplice copia il contenuto di un registro in un altro, es. LD A,B. In queste abbreviazioni (mnemonici) e'convenzione porre prima il registro destinazione poi quello sorgente. Quindi LD A,B copia il contenuto del registro B nell'accomulatore. E' possibile caricare in un registro anche numeri; es. LD C,123 carica 123 nel registro C. Da dove provenga 123 lo discuteremo piu' avanti, per ora basti sapere che le istruzioni vengono memorizzate come numeri. Ad esempio LD A,B e'memorizzata come 78 (guarda l' appendice A del manuale dello Spectrum). Alcune istruzioni dello Z80 richiedono fino a 4 numeri. I due tipi di istruzioni discusse(LD r,r' e LD r,n) possono trattare qualsiasi registro, mentre unicamente il registro A puo' essere caricato con il contenuto di una locazione di memoria eviceversa. La forma dell' istruzione e' LD A,(nn) che copia il contenuto della locazione di indirizzo 'nn' (un numero tra 0 e 65535) nell' accomulatore.Tutti gli altri registri non possono essere caricati direttamente da una locazione di memoria. Di solito vengono usate due istruzioni che fanno uso del registro A; es. LD A,(30000) e LD E,A caricano il contenuto della locazione 30000 nelregistro E.

COPPIE DI REGISTRI

Sommario
LD dd,nn dove dd e' una qualsiasi coppia BC, DE e HL; nn e' un numero da 0 a 65535
LD dd,(nn)
LD (nn),dd EX DE,HL scambia il contenuto di HL e DE

Il fatto che un registro puo' contenere solo numeri fino a 255 e la memoria disponibile e' fino a 65535, e' una limitazione. Per questa ragione esiste un insieme diistruzioni che consentono di accoppiare i registri. Le coppie sono BC, DE, e HL. I due registri contengono parti diverse di un numero. Considera il numero decimale 27, questo possiamo pensarlo come diviso in due parti; una piu' significativa (il 2 poiche' esso rappresenta 2*10) e una meno (il 7 cioe' 7*1). Il totale delle due parti e' 2*10+7*1=27. Ciascuna cifra puo' essere solo da 0-9 cioe' 10 numeri differenti. Per noi questa cifra sara' il BYTE.
Un singolo registro puo' contenere da 0-255 numeri differenti, cosi' in una coppia come HL, il registro H contiene la parte piu' significativa e L quella meno. Allora nella coppia di registri noi possiamo memorizzare il numero H*256+L*1 cosi' come 27=2*10+7*1. Il massimo, quindi, che una coppia di registri puo' contenere e' 65535 (255 * 256 + 255). Tali coppie possono essere direttamente caricate con istruzioni del tipo LD HL,nn. Cosi' come per i registri, e' possibile accoppiare due locazioni di memoria adiacenti. Per convenzione la parte meno significativa e' memorizzata nella locazionedi memoria ad indirizzo piu' basso.
Possiamo percio' caricare coppie di registri con i contenuti di locazioni di memoria adiacenti e viceversa, es. LD DE,(nn) carica la coppia DE con il contenuto delle locazioni di memoria (nn+1) e (nn). Viceversa possiamo caricare due locazioni di memoria con il contenuto di una coppia di registri, es. LD (nn),BC che copia ilcontenuto di B nella locazione (nn+1) e il contenuto del registro C nella locazione di memoria (nn). Non esistono pero' delle istruzioni per caricare direttamente una coppia di registri tra loro. Di solito vengono utilizzate due istruzioni della forma LD r,r' .
L' istruzione EX DE,HL scambia il contenuto della coppia DE con la coppia HL. N.b. La precedente istruzione e' disponibile in un' unica forma, cioe' tra le coppie DE e HL.

INDIRIZZI

Sarebbe veramente difficile scrivere programmi in linguaggio macchina utilizzando come variabili solo i pochi registri a nostra disposizione. Si tratta di trovare il modo di scrivere dati che ci interessano non solo nell' esiguo spazio all'interno dello Z80 ma anche in quello che si chiama RAM, cioe' la memoria messaci a disposizione dallo ZX.
Per far questo si usa ancora l' istruzione LD ma con una scrittura leggermente diversa dal solito. Se vogliamo memorizzare o per meglio dire copiare il contenuto del registro A nell' indirizzo 'x' scriveremo: LD (x),A equivalente a POKE x,A
Al contrario se vogliamo caricare il registro A col contenuto della cella 'x' l'istruzione sara': LD A,(x) che significa LET A= PEEK x
Nel caso dei registri doppi il numero occupa 2 celle di memoria dato che anch'esse, come i registri possono contenere solo numeri interi da 0 a 255. Quindi LD (16320),HL corrisponde alle 2 istruzioni basic POKE 17000,HL-256*PEEK(HL/256) POKE 17001,PEEK(HL/256) o se preferite l'equivalente POKE 17000,L POKE 17001,H
Ancora una volta notate l'inversione del numero: la parte alta va dopo e la parte bassa va prima; cosi' se HL

contiene 32000 (7d00h), LD (19000),HL avra' l'effetto di mettere in 19000 il valore 0 e in 19001 il valore di 125 (7dh).
Per le operazioni di caricamento di un accumulatore doppio col valore di 2 celle di memoria valga quest'esempio: se nelle celle 31123/24 c'e' il valore 25435, LD HL,(32123) significa LET HL=PEEK 32123+256*PEEK 32124o l'equivalente LET H=PEEK 32124 LET L=PEEK 32123
Dopodiche' HL varra 25435.L' operazione: LD (indirizzo),accumulatore non altera il contenuto dell' accumulatore.I codici per queste operazioni sono:
LD A,(pq)
LD BC,(pq)
LD DE,(pq)
LD HL,(pq)
LD (pq),A
LD (pq),BC
LD (pq),DE
LD (pq),HL
Ci sono anche altre istruzioni PEEK e POKE:
LD A,(BC)(LET A=PEEK BC)
LD A,(DE)
LD A,(HL)
LD B,(HL)
LD C,(HL)
LD D,(HL)
LD E,(HL)
LD H,(HL)
LD L,(HL)
LD (BC),A (POKE BC,A)
LD (DE),A
LD (HL),A
LD (HL),B
LD (HL),C
LD (HL),D
LD (HL),E
LD (HL),H

I PUNTATORI

Usciamo un attimo dall'ambito specifico del l/m per parlare di un argomento molto importante in tutti i linguaggi: i pointers. Un puntatore come dice il nome e' una cosa che punta,cioe' indica determinati oggetti all'interno del computer. E'in pratica un numero che rappresenta l'indirizzo a cui noi possiamo trovare gli oggetti sopracitati quali ad esempio l'area delle variabili,la RAMTOP o tabelle che noi stessi abbiamo creato. I puntatori sono un po' come delle scatole nel cui interno troviamo dei bigliettini che ci dicono dove andare per trovare certe cose. E' molto importante capire bene ed essere padroni della differenza che c'e' tra puntatore e oggetto puntato: il puntatore e' la scatola con il suo bigliettino che si trovano sempre allo stesso posto fisso;l'oggetto puntato,invece,puo' essere qualsiasi cosa e il suo indirizzo non e' fisso. Questo tipo di struttura dinamica e' molto usato in Pascal e si rivela molto utile anche in l/m dato che in tal modo noi possiamo costruire delle tabelle di puntatori che ci dicono dove abbiamo messo cio' che ci interessa. La piu' importante di queste tabelle e' quella che viene costruita automaticamente dal computer appena lo si accende ed e' quella dedicata alle 'variabili di sistema'che nello ZX parte dall'indirizzo 23552 e si estende fino a 23733 ; da questa noi possiamo ricavare molte informazioni.
23730/31 troviamo il puntatore al primo byte non usabile per il BASIC: in pratica questo e' l'indirizzo della RAMTOP.
23627/28 qui c'e' il puntatore all'area delle variabili. Se in un programma BASIC noi scriviamo LET COSTO=1000, il numero 1000, associato alla parola costo vengono memorizzati nella zona di memoria indicata dagli indirizzi 23628/29
23641/42 questo e' il puntatore che ci indica l'ultimo byte dell'area delle variabili. Dato che questa viene dopo il listato dei programmi,il PEEK di queste locazioni ci indica quanta memoria abbiamo abbiamo occupato tra programma e variabili relative. Se poi facciamo la differenza col valore di RAMTOP, abbiamo il numero dei bytes di memoria libera, con le seguenti operazioni PRINT (PEEK 23730+256*PEEK 23731)-(PEEK 23641+256*PEEK 23642)
23618/19 questo e' il puntatore alla prossima riga da eseguire
23620 questo e' il puntatore al numero di statement all'interno della riga alla quale si deve saltare. Ad esempio se noi volessimo con delle istruzioni in l/m simulare un GOTO 2560 sara' sufficiente fare:

LD HL,2560d
LD A,1
LD (23618),HL
LD (23620),A
Questi sono dunque gli indirizzi ai quali troverete alcuni dei puntatori piu' utili,la cui conoscenza da' grande potere sulla macchina. Un elenco completo potete comunque trovarlo sul manuale dello ZX. Citiamo ancora 23659che ci da' il numero di righe che vengono usate per la parte inferiore dello schermo; se pokato a 0 permette di al posto delle 22 righe di schermo normalmente disponibili, di utilizzarne 24

INDIRIZZAMENTO INDIRETTO

Sommario
LD r,(HL) dove r' e' uno qualsiasi dei registri A,B,C,D,E,H,L.
LD (HL),r
LD A,(dd)
LD (dd),A dd=BC o DE.

Finora abbiamo utilizzato istruzioni nelle quali le locazioni di memoria erano specificate nell'istruzione stessa. Un modo diverso, molto utile, e' quello di utilizzare il contenuto di una coppia di registri, conosciuto come indirizzamento indiretto.
Per esempio l'istruzione LD B,(HL) carica il registro B con il contenuto della locazione il cui indirizzo e' nella coppia di registri HL.
Tutti i singoli registri possono essere caricati tramite HL come puntatore. Viceversa una locazione di memoria puo' essere caricata con il contenuto di un registro tramite HL, es. LD (HL),C.L' uso delle coppie BC e DE come puntatori e' limitato al solo registro A, cioe':
LD A,(DE) , LD A,(BC) ecc.

ADDIZIONI E SOTTRAZIONI

Sui registri si possono anche fare delle operazioni molto semplici:

ADD,ADC,INC,SUB,SBC,DEC

ADD e ADC riguardano l'addizione. A un registro si puo' sommare una costante oppure se ne puo' sommare un altro. La differenza tra ADD e ADC e' che ADD non tiene conto del riporto se in un conto c'e un overflow;ADC invece si'. Se usiamo ADD per sommare a un registro contenente un altro contenente 173, il risultato non sara' 403 (ricordate che i registri possono contenere solo numeri tra 0 e 255) ma 147,cioe' 403-256 o, se preferite, 403 MOD(256). Lo stesso tipo di ragionamento vale per i reistri doppi quando si supera 65536 cioe' 256^2: 65000+10000 non da' 75000 ma 9464 (=75000- 65536).
Tutte le volte che c'e' un overflow un particolare bit (appunto il bit di overflow) viene messo a 1; in pratica si tratta del riporto. L' istruzione ADC al risultato della somma aggiunge 1 se nella somma eseguita immediatamente prima si era verificato un overflow. Il bit del riporto si chiama CARRY
Cosi':

ADD A,E significa LET A=A+E e LET CARRY=INT((A+E)/256)
ADD HL,BC significa LET HL=HL+BC e LET CARRY=INT((HL+BC)/65536)
ADC A,E significa LET A=A+E+CARRY e LET CARRY=º((A+E+CARRY)/256)

Le operazioni di somma (e anche differenza) si possono fare solo tra 2 operandi di cui uno e' il registro dove andra' messo il risultato. Quindi se volessimo fare:

HL=DE+1234

dovremmo risolvere il problema in questo modo:

LD HL,0 (LET HL=0)
LD BC,1234 (LET BC=1234)
ADD DE,BC (LET DE=DE+BC)
ADD HL,DE (LET HL=HL+DE)

Notate l'importanza dell'istruzione LD HL,0 ; se non ci fosse il risultato delle operazioni sarebbe LET HL=HL+DE+1234

Tuttavia e' possibile sommare delle costanti a un registro direttamente,ma cio' e' possibile solo col registro A:

ADD A,51 (LET A=A+51)
ADC A,73 (LET A=A+73+CARRY)
Un altro modo di sommare cosi' tanti, se si tratta di valori abbastanza piccoli e' quello di usare INC che incrementa il valore del registro di 1 :
INC E significa LET E=E+1
Quindi LET HL=HL+4 puo' essere risolto cosi':

INC HL
INC HL
INC HL
INC HL

INC lascia il Carry inalterato anche in caso di overflow; se A=254, INC A avra' come effetto A=255 e ancora INC A dara' A=0.

Esistono anche operazioni che permettono di alterare direttamente il CARRY che sono:
SCF (LET CARRY=1)
ADD A,0 (LET CARRY=0)
CCF (LET CARRY=1-CARRY)

Simile a INC e' DEC che invece di incrementare decrementa:cioe' DEC BC significa LET BC=BC-1
In questo caso se BC=0 , DEC BC dara' BC=65535.
Sempre in tema di sottrazioni SUB si comporta similmente a ADD e SBC come ADC ,cioe' al risultato della sottrazione sottrae anche il contenuto di CARRY.
Il CARRY viene messo a 1 ogni volta che il risultato della sottrazione e' un numero negativo. In questo caso il risultato sara' il numero negativo+256,nel caso di registri singoli o +65536 nel caso di registri doppi.
A volte,al posto di SUB A,B (LET A=A-B) capita di veder scritto solo SUB B e questo succede perche' e' sottointeso il registro A,dato che solo da A si puo' sottrarre qualcosa.
Per i registri doppi la sottrazione puo' avvenire solo da HL.

ADDIZIONE E FLAG DI CARRY

Sommario
ADD A,n dove n e' un numero tra 0-255
ADD A,r dove r e' un singolo registro
ADD A,(HL)
ADD HL,BC
ADD HL,DE
ADC A,n
ADC A,r
ADC A,(HL)
ADC HL,BC
ADC HL,DE

Lo Z80 consente di eseguire somme sia con singoli registri sia a coppie. Tutte le somme con singoli registri vengono fatte attraverso l'accomulatore. Al registro A puo' essere sommato un numero (ADD A,6),un registro (ADD A,B) o il contenuto della locazione puntata da HL (ADD A,(HL)). Il risultato rimane nell' accomulatore mentre il secondo operando non viene modificato. L'addizione di coppie di registri viene eseguita utilizzando la coppia HL e puo' solo trattare come secondo operando i registri BC e DE (ADD HL,BC e ADD HL,DE). Di nuovo il risultato rimane in HL mentre l'altra coppia non subisce modifiche. La somma in entrambi i casi risultera 'corretta' se il risultato finale non supera il massimo numero esprimibile con il singolo registro o con la coppia. Se tale evenienza accade il flag di carry viene posto a '1', altrimenti a '0'. In questo modo successive istruzioni possono esaminare lo stato del flag (segnale) stesso e prendere le decisioni opportune del caso. Una seconda forma di somma e' disponibile sullo Z80, conosciuta come addizione con riporto (carry) abbreviata ad ADC. Questa e' simile alla precedente solo che, se il flag di carry era posto ad 1, il risultato viene incrementato di 1. Tutte le forme viste per l'istruzione ADD sono valide per ADC. L'istruzione ADC puo' essere concatenata per eseguire L'addizione di due numeri diqualsiasi lunghezza.

SOTTRAZIONE E FLAG DI CARRY

Sommario
SUB n sottrazione senza carry
SUB r
SUB (HL)
SBC A,n sottrazione con carry
SBC A,r
SBC A,(HL)
SBC HL,DE sottrai da HL, DE con carry
SBC HL,BC sottrai da HL, BC con carry
SCF setta a '1' il flag di carry
CCF complementa il flag di carry

La sottrazione di singoli registri, cosi' come per la somma, avviene tramite l'accomulatore. Tutte le forme valide per ADD possono essere utilizzate nella sottrazione. L' abbreviazione o mnemonico e' SUB, il quale e' sempre scritto senza la A che e' implicita. Di nuovo il risultato e' tenuto nell'accomulatore. Il flag di carry e' posto a '1' se il risultato esce dall'intervallo 0-255.
Non esiste l'istruzione SUB per le coppie di registri.
Tutte le forme di ADC possono essere usate con SBC (Sottrai con Carry). L'operazione e' simile alla SUB, trann nel fatto che il risultato e' decrementato di uno se il flag di carry era posto a '1'. Come per l'istruzione ADC, SBC puo' essere concatenata per eseguire la differenza di numeri di qualsiasi dimensione.
Poiche' la sottrazione con coppie puo' solo essere eseguita con Carry, lo stato di quest'ultimo prima di SBC HL,BC e SBC HL,DE deve essere '0'. Il flag di carry puo' essere posto a '1' (settato) con l'istruzione SCF, mentre l'istruzione CCF complementa lo stato del carry, cioe' se era '1' lo pone a '0' eviceversa. Vedremo piu' tardi che tutte le istruzioni logiche pongono a 0 'reset' lo stato del carry.

ISTRUZIONI DI INCREMENTO E DECREMENTO

Sommario
INC r
INC (HL)
INC dd
DEC r
DEC (HL)
DEC dd

Le ultime istruzioni aritmetiche da trattare possono essere eseguite sia su registri singoli che doppi. Queste sono INC e DEC. INC incrementa, di uno, il contenuto del registro o della locazione di memoria indirizzata indirettamente da HL.
DEC decrementa, di uno, il contenuto di un registro o della locazione di memoria puntata dalla coppia di registri HL. Il flag di Carry non viene modificato da queste istruzioni. INC e DEC vengono di norma utilizzate quando sono richiesti contatori, o per indirizzare indirettamente una sequenza di locazioni di memoria (utilizzando ad es.il registro HL come puntatore in memoria).

IL FLAG DI ZERO

Un altro flag molto importante e' il flag di zero: questo viene posto a '1' se il risultato di una operazione aritmetica e' zero altrimenti e' posto a '0' (reset). Questo flag non viene modificato dalle istruzioni aritmetiche che trattano coppie di registri, tranne per quelle con il carry, cioe' ADC HL,dd o SBC HL,dd.
Il flag di zero come quello di carry non viene modificato da istruzioni come LD o EX.

ESEMPI DI SOMMA

Eccovi un semplice programma che consente di fare la somma di 2 numeri minori di 256 , presa in modulo 256il che significa che se il risultato della somma dovesse essere maggiore di 256, al risultato verra' sottratto 256.
Ad esempio 200+100=44 cioe' 300-256; mentre 100+100 fara' 200 senza problemi.
Per eseguire il programma dovete dare:
POKE (I+1),primoaddendo, POKE (I+3),secondoaddendo quindi PRINT USR Idove 'I' sta per l'indirizzo al quale avete caricato il programma. Nel nostrocaso I vale 32000, come spiegato all'inizio.
Ecco dunque il listato:

LD A,primoaddendo

```
ADD A, secondoaddendo
LD C,A
LD B,0
RET
```

I due POKE sono necessari per scrivere nella memoria gli operandi delle due istruzioni LD e ADD. In seguito il risultato e' caricato nel registro BC che e' quello che viene stampato dall' istruzione USR

Un piccolo trabocchetto nel quale e' facile cadere: confrontate questi due programmi:

```
ADD A,0
ADD A,0
SUB A,2
DEC A
SBC HL,DE
DEC A
SBC HL,DE
```

Apparentemente essi sono uguali e il loro scopo e' quello di sottrarre 2 da A e quindi sottrarre DE da HL. Il primo programma pero' non e' corretto: l' unica istruzione disponibile per sottrarre DE da HL e' SBC HL,DE, ma questa sottrae pure il CARRY che quindi deve essere messo a 0 prima di tale operazione: questo e'cio' a cui serve ADD A,0. Nel primo esempio pero', con l'istruzione SUB si corre il rischio di alterarlo se per caso il precedente valore di A era 1 o 0, cosicche' il risultato di SBC HL,DE sarebbe stato HL-DE-1 e non HL-DE. Al contrario, nel secondo esempio,l'istruzione DEC non altera minimamente il CARRY

I SALTI

Ci sono due tipi fondamentali di salto in l/m: il "GO TO" assoluto e il "GO TO" relativo.
Il primo non presenta problemi: si chiama JP abbreviazione di JUMP. Il suo effetto e' quello di saltare all'indirizzo 'xxyy' come un GO TO xxyy. Il salto relativo (JR) invece ci dice di quanti bytes avanti o indietro saltare, rispetto all'indirizzo attuale, per un massimo di 127 bytes.
Quest'ultima istruzione rispetto a JP ha 2 vantaggi: occupa solo 2 bytes contro i tre di JP e, cosa importantissima, una routine scritta con i JR puo' essere interamente rilocata senza alterare i valori del salto.
"JR 1" significa che la successiva istruzione, se essa e' lunga un byte, verra' saltata. Ad esempio :

```
JR +3
LD BC,1234h
LD HL,0000
ecc...
```

In questo esempio l'istruzione LD BC non sara' mai eseguita perche' verra' saltata.
Per i salti negativi la cosa e' un po' piu' complessa e bisogna cominciare a contare all' indietro a partire dal secondo byte dell'istruzione JR. Per i salti in avanti si usano i valori esadecimali da 0a 7f e per quelli all'indietro quelli da 80 a ff. JR fe significa JR -2, cioe' un loop infinito, non interrompibile. Il suo effetto e' simile a quello della riga Basic: 10 GO TO 10
La vera utilita' del GO TO nel Basic sta nel fatto che essi possono essere associati a delle decisioni del tipo IF...THEN.
Questo,anche se in modo piuttosto rudimentale e' possibile pure in l/m.
Esistono infatti le istruzioni:JP Z,xxyy, JP NZ,xxyy, JR Z,nn, JR NZ,nn
Il significato di JP Z e di JR Z e' : salta all'indirizzo indicato se l'ultima operazione effettuata (ADD,ADC,SBC,SUB,INC,SUB e altre) ha dato come risultato zero. Non tutte le operazioni indicate sopra funzionano in relazione ai salti condizionati sia con i registri semplici che con quelli doppi ma studieremo meglio queste particolarita' in seguito quando amplieremo il discorso sulle condizioni di salto imparando anche a utilizzare i '>' e '<' e a evitare di dover per forza alterare il valore di un registro con una delle operazioni sopracitate.
Similmente funzionano JP NZ e JR NZ e servono per effettuare il salto se il risultato e' <>0. Tutto questo ci serve per poter scrivere alcuni programmi veramente interessanti.
Il primo che vi presentiamo. Consente di cambiare gli attributes dello schermo. Infatti il valore assegnato al registro 'A' viene scritto nella zona di memoria da 22528 a 23296 che e' appunto quella dove sono memorizzati i colori dello schermo. I criteri per il caricamento di 'A' sono gli stessi con cui viene dato il risultato della funzione Basic ATTR:
COLORE DELL' INCHIOSTRO + COLORE DELLA CARTA *8 + 64 SE BRIGTH + 128 SE FLASH

Il programma di cui vi presentiamo il listato potete scriverlo a qualsiasi indirizzo, dato che si usano dei JR.

```
LD A,23d
LD HL,22528
LD B,24d
LD C,32d
{LOOP1}
LD (HL),A
{LOOP2
INC HL
DEC C
JR NZ,LOOP2
DEC B
JR NZ,LOOP1
RET
```

Il valore 23 nella prima istruzione significa inchiostro bianco(7) e carta rossa(2):7+2*8=23. Per cambiare colore della carta date POKE ind.+1,valore e RANDOMIZE USR ind.
Il secondo programma che vi presentiamo inverte lo schermo cioe' ne da' l'immagine in negativo lasciando pero' inalterati i colori. Per far questo ogni singolo byte dello schermo (da 16384 a 22528) viene complementato. Il complemento di un numero e' 255 meno quel numero; il che, a livello binario ha l'effetto di trasformare ogni 0 in 1 e ogni 1 in 0.
Questo programma e' stato memorizzato all'indirizzo 32000:

```
LD HL,16384
LD B,24d
LD C,0
{LOOP1}
LD D,(HL)
{LOOP2}
LD A,255
SUB D
LD (HL),A
INC HL
DEC C
JR NZ,LOOP2
DEC B
JR NZ,LOOP1
RET
```

Sempre per restare in tema di salti, per simulare l'azione di un FOR TO NEXT, se le ripetizioni del ciclo sono < 255, e' molto comoda la seguente istruzione: DJNZ,xx che esegue le seguenti operazioni:

```
DEC B
JR NZ,xx
```

Cosi' se dobbiamo ripetere per 7 volte una particolare routine, il programma sara':

```
LD B,7
{LOOP}
istruzioni da ripetere
DJNZ, LOOP
```

E ora veniamo a una delle istruzioni piu' usate in l/m: LDIR. Questa istruzione e' particolarmente indicata per spostare blocchi di memoria in quanto :
1) Trasferisce il contenuto della cella il cui indirizzo e' HL nella cella il cui indirizzo e' DE
2) Incrementa HL
3) Incrementa DE
4) Decrementa BC
5) Se BC <>0 ripete dal punto 1)
Quindi se dovete trasferire 'n' bytes dall' indirizzo 'x' all'indirizzo 'y' potete utilizzare la seguente routine:

```
LD BC,n
LD HL,x
```

```
LD DE,y
LDIR
(RET)
```

Esiste anche LDDR che e' similea LDIR solo che al posto di incrementere DE e HL li decrementa.
Come esempio d' uso di LDIR vedrete una routine che operera' uno scroll laterale di un byte dello schermo e un'altra che invece fara' lo scroll verso l'alto degli attributes.
La prima e' molto semplice in quanto si tratta di spostare indietro di un bytes tutta l'area dello schermo. L'operazione che sta alla base di questo e' POKE x,PEEK (x+1) con x che va da 16384 a 22527: deve quindi essere ripetuta 6143 volte. Il listato e' questo:

```
LD DE,16384
LD HL,16385
LD BC,6193
LDIR
RET
```

Per quanto riguarda lo scroll verticale dei colori: lo schermo degli attributes e' lungo 32*24=768 bytes e parte dall'indirizzo 22528. Ogni riga e' lunga 32 caratteri, quindi se vogliamo scrivere un carattere una riga piu' in alto, dobbiamo copiarlo a un indirizzo 32 bytes piu' basso.

```
LD DE,22528
LD HL,22560
LD BC,736
LDIR
RET
```

COS'E' LO STACK

La traduzione letterale di stack e' mucchio, pila, e proprio di una pila di numeri si tratta. Quando il computer deve mettere dei numeri da qualche parte, li mette in pila sullo stack. Ad esempio quando si usa in un programma Basic un GOSUB, lo ZX deve mettere da qualche parte il numero della riga a cui il GOSUB si trova, in modo che al termine della subroutine, il RETURN sappia dove tornare. Tali numeri vengono messi sullo stack. Per mettere il contenuto di un registro nello stack si usa l'operazione PUSH che non altera il contenuto del registro ma si limita a copiarlo. Invece per recuperare tale numero si usa POP.
Considerate il seguente esempio: supponiamo di avere tutti i registri contenenti informazioni che ci servono e di dover alterare HL per determinate operazioni. Il problema puo' essere risolto dando PUSH HL seguito dalle operazioni che dovevamo fare, quindi quando avremo bisogno del precedente valore di HL, POP HL.
I numeri sono dunque messi su una pila e questo significa che come in una pila di scatole, noi possiamo prendere solo quella che sta in cima alla pila. Cosi' se diamo nell'ordine PUSH DE, PUSH BC, PUSH HL, avremo una situazione di questo tipo:
HL
BC
DE
Quando vorremo rimettere i valori nei rispettivi registri dovremo dare nell' ordine POP HL, POP BC, POP DE. Se al contrario noi li riprendessimo cosi': POP BC, POP DE, POP HL, in BC verrebbe messo il vecchio valore di HL, in DE quello di BC e in HL quello di DE. Cercate di immaginare lo stack come una pila appesa al soffitto capovolta: i numeri vengono messi nella parte piu' alta della memoria disponibile e singolarmente. Per esempio il registro HL "pushato" su un DL sara' cosi' rappresentato:

```
_____
  |
|H||L|
  |
|D||E|
  |
 ...
```

L' indirizzo a cui si trova l'elemento piu' basso dello stack (nell'esempio precedente HL) e' contenuto nel registro speciale "SP" (=Stack Pointer) che ci fa sapere quanto e' alta la pila e a che punto siamo arrivati. Non si possono usare PUSH o POP con i registri singoli: quando si memorizza con tale metodo un registro doppio, lo spazio necessario alla memorizzazione e' di 2 byte; di conseguenza ogni PUSH decrementa SP di 2. Per esempio: l'equivalente di PUSH HL in BASIC sara':
POKE SP-1,H: POKE SP-2,L: LET SP=SP-2

Mentre l'equivalente di POP HL sar':
LET L=PEEK SP: LET H=PEEK (SP+1): LET SP=SP+2
Al posto di A si usa AF: questo registro doppio e' formato dai due registri A e F allo stesso modo in cui DE e' formatoda D e da E. 'A' e' il solito registro cho abbiamo incontrato altre volte; F invece e' una specie di registro che contiene 8 FLAGS. FLAG e' un 'registrino' grande 1/8 di registro normale, che puo' contenere solo uno 0 o un 1. Una parte del registro F e' formata dal CARRY che conosciamo gia'; delle altre 7 ne parleremo in seguito. Per adesso sappiate solo che invece di PUSH A dovete ricorrere a PUSH AF, ma per ogni uso pratico non vi e' alcuna differenza.
Un semplice esempio d'uso dello stack e' il seguente: supponiamo di voler sommare 64d a E senza alterare il valore degli altri registri:

```
PUSH AF
LD A,E
ADD A,64d
LD E,A
POP AF
```

Sul registro SP si possono fare operazioni direttamente come se si trattasse di un qualunque altro registro.
Quando scriviamo una subroutinein BASIC, al termine si mette una istruzione di RETURN. Anche in l/m si segue questa regola e si mette RET. Questa istruzione vi e' famigliare: infatti e' la stessa che mettiamo sempre alla finedi ogni programma in l/m per tornare al basic. Infatti un programma in l/m e' una subroutine che viene chiamata dal Basic. Le azioni svolte da RET sono
1)prende l'elemento sullo stack con un POP e
2)carica il program counter con quel valore.
CALL e RET possono anche essere condizionali, seguendo le stesse regole dei JP.
Esiste anche un altro tipo di CALL, piu' limitato perche' non e' condizionale e solo certi indirizzi della ROM possono essere chiamati. E' pero' piu' comoda perche' e' lunga un solo byte.
Tale istruzione e' RST
Ripendiamo il discorso sui salti condizionali. Quanto detto vale anche per CALL e RET.
Dalle prime lezioni sapete dell'esistenza di un flag detto di CARRY. Esistono anche altri flag e, cosi' come il carry ci dice se nella operazione precedente c'e' stato un overflow, cosi' questi altri memorizzano varie caratteristiche dell'operazione appena effettuata.
Il flag 'SIGN' se uguale a 0 indica che il risultato e' positivo; se uguale a 1 il risultato appena calcolato era negativo. Questo flag viene controllato dalle istruzioni:

```
JP P
JP M
CALL P
CALL M
RET P
RET M
```

La 'P' significa che l'istruzione viene eseguita se l'ultimo risultato calcolato era positivo. La 'M' si riferisce invece a risultati negativi(Meno).
Il flag ZERO ci segnala se l'ultimo risultato calcolato era uguale a 0. E' il flag che viene controllato dalle istruzioni condizionali che conosciamo gia' che specificano la condizione aggiungendo Z o NZ (esempio: JP Z xxxx).
Il flag CARRY e' quello che conoscete e il suo stato puo' essere controllato dalle istruzioni:

```
JR C
JR NC
JP C
JP NC
CALL C
CALL NC
RET C
RET NC
```

Esistono anche altri flag come l'HALF CARRY, il PARITY/OVERFLOW e il SUBTRACT ma questi sono raramente usati e non vale la pena di soffermarcisi.
Per avere in l/m la stessa facilita' d'uso delle istruzioni Basic IF/THEN, e' necessario introdurre un'ultima istruzione: CP che e' l'abbreviazione di ComPara CP
compara il contenuto del registro A con un altro registro o con una costante. Il vantaggio di questa istruzione e' che altera i vari flag ma senza modificare il valore di alcun registro.
CP costante = FE costante
Eccovi ora alcuni esempi per chiarire quanto presentato.

IF A=B puo' essere scritto in l/m con:
CP B
JP Z,100
IF A<50 si puo' risolvere con
CP 33h
CALL M,100
Infatti CP opera in questo modo: esegue una sottrazione fittizia A - l'operando. Quindi se A<50, A-50 da' un risultato negativo che viene comunicato al sign flag.
Avrete notato che non esiste un JR P o un JR M, tuttavia il salto relativo condizionato da un < o un > puo' essere realizzato nel seguente modo:
IF A>C THEN GOTO (-10) si realizza con
CP C
JR NC,-10
Infatti il risultato della sottrazione A-C (ricordate comunque che nessun registro viene pero' alterato) non produce un CARRY se A>C. Viceversa se C>A si avrebbe un overflow che sarebbe messo nel CARRY. Il JR NC viene eseguito solo se **N**on c'e' **C**arry,quindi se C<=A.

SCAMBIO DEGLI ATTRIBUTI VIDEO

Vi proponiamo una routine che prende l' 'INK' e il 'PAPER' di ciascun carattere dello schermo e li scambia tra loro.
La memoria degli attributi nello Spectrum inizia alla locazione 5800H, e si estende fino al buffer della stampante cioe' 5B00H.
E' inoltre importante sapere come gli attributi di un singolo carattere vengono memorizzati all' interno di un byte.
I bit 0-2 contengono l' INK (8 possibili combinazioni), mentre i bit 3-5 codificano il PAPER.
Gli ultimi due bit 6-7 memorizzano lo stato degli attributi FLASH e BRIGHT.Il codice macchina e' memorizzato alla locazione 55400 ed e' lungo 29 bytes.
Guarda il listato

SCROLL VERTICALE GRAFICO

Vi proponiamo una routine che esegue lo scorrimento (scroll) verticale (verso l'alto) di una porzione rettangolare dello schermo.
Per il suo utilizzo e' necessario specificare la finestra che si vuole scrollare dando le coordinate dell'angolo superiore a sinistra (X1,Y1) e le coordinate dell'angolo inferiore a destra (X2,Y2).E' necessario sapere che in questa routine e' stato utilizzato un diverso sistema di coordinate; con l'origine (0,0) posta nell'angolo superiore a sinistra. Inoltre le X variano tra 0-31 (colonne in modalita' a caratteri)mentre le Y variano tra 0-191 (righe in modalita' grafica).
Vi e' infine da specificare la modalita' con cui si vuole lo scroll. Queste sono:
0 - Lascia l'ultima riga cosi' com'e'.
1 - Copia la riga fuoriuscita nell'ultima.
2 - Copia nell'ultima riga un byte di fill.
La seconda opzione fa' in modo che la finestra ruoti nel rettangolo di video riservato. Mentre l'ultima opzione fa' si che,ad esempio, il valore 15 (00001111 binario) lasci una griglia verticale nella finestra di scroll.Come puoi vedere, listando il programma (Basic) dalla linea 4000, le coordinate vengono pokate direttamente nel programma, mentre la modalita' e l'eventuale byte di fill sono memorizzati in locazioni libere della zona variabili di sistema e rispettivamente alla 5B41H e 5B42H.
Il codice macchina e' memorizzato alla 55450 e si estende per 115 byte.
E' ora opportuno, per la comprensione della routine, che tu sappia come e' mappato il video in memoria.Ogni pixel (puntino) richiede un bit per segnalare se questo e' acceso o spento. Ogni riga dello Spectrum ha 256 pixel, quindi sono necessari 32 byte per ogni riga grafica. Ne esistono 192 di queste righe, divise in tre zone: superiore, di mezzo e inferiore, ciascuna composta da 64 righe.I 32 byte di ogni riga sono memorizzati sequenzialmente. La prima riga inizia alla locazione 4000H, quindi le successive sette (le quali insieme compongono la prima riga a caratteri) sono memorizzate 256 byte dopo l'inizio della precedente.Quindi la prima riga (grafica) della seconda riga a caratteri e memorizzata subito dopo la prima riga (grafica) della prima riga a caratteri. Il tutto si ripete fino all' ottava riga (grafica) dell' ottava riga a caratteri della zona video relativa.
Il sistema e' immediatamente comprensibile se si osserva un qualunque programma che carica uno screen da nastro e si segua la successione con cui viene riempito il video, tenendo presente che il computer sta caricando la memoria video sequenzialmente, cioe' senza salti.
Guarda il listato

Per quanto riguarda lo scroll esteso pero' all'intero schermo in alta risoluzione si puo' anche usare il seguente programma

```
LD HL,#401F
LD D,H
LD E,L
DEC H
LD B,#03
PUSH BC
LD B,#08
PUSH BC
LD B,#08
PUSH BC
INC H
LD BC,#0020
LDDR
LD DE,#0020
ADD HL,DE
LD D,H
LD E,L
POP BC
DJNZ L7E9E
LD BC,#F820
ADD HL,BC
POP BC
DJNZ L7E9B
LD A,#07
ADD A,H
LD H,A
POP BC
DJNZ L7E98
LD HL,#57DF
LD B,#20
INC HL
LD (HL),#00
DJNZ L7EC1
RET
```

l'istruzione chiave e' LDDR che nel nostro caso e' stato messo per esempio all'indirizzo 7ea3. BC ha il valore 32 (istr. 7ea0) per generare un loop che viene ripetuto 32 volte, tante quanti sono i Byte in una riga di schermo: ricordate che 1 pixel e' uguale a un bit, quindi 256 pixels=256/8=32 bytes. Lo schermo, come sapete, e' disposto in maniera piuttosto strana. Prima viene la prima linea in alto poi nell'ordine la 9,17,25,33,41,49,56, poi la 2,10, ecc... sempre in passi di 8 e questo per 8 volte per un totale di 32*8*8=2048 bytes. Questo tipo di struttura si ripete identicamente 3 volte nello schermo; la 2° volta dalla linea 65 e la 3° dalla 129. Per questo motivo troviamo 3 loop principali: il primo si ripete 3 volte (7e96) da 7e98 a 7eba, il 2° 8 volte da 7e9b a 7eb3 e il 3° 8 volte da 7e9e a 7eac. L'uso delle istruzioni di PUSH e POP permette di utilizzare il registro BC piu' volte per il controllo di loop diversi. Tornando al problema dello scroll, se lo schema seguisse una struttura logica durante uno scroll verso l'alto la 2° riga passerebbe alla 1°, la 50° alla 49° e cosi' via, con una differenza di 32 tra la locazione di partenza e quella di arrivo.
Dato che gli indirizzi fisici sono ordinati in modo diverso, se vogliamo che una riga dello schermo occupi il posto di quella immediatamente superiore, dobbiamo spostare di 256 locazioni di memoria indietro, il contenuto di ogni cella che si trova all'interno di un gruppo di 8 righe. A questo provvede la istruzione INC HL all' indirizzo 7e9f che fa in modo che HL=256+DE.
Ogni volta che ci troviamo a dover spostare una riga dal 1° posto di un gruppo di 8,all' ultimo del gruppo precedente, la differenza in locazioni di memoria e' 1760. Per fare questo le istruzioni 7eae_7eb1 sottraggono da HL 2016 (in realta' sommano 63520 ma in MOD(2^16) e' la stessa cosa) che sommando il 256 dell' istruzione 7e9f da' 1760 come e' richiesto.
La cosa e' un po' macchinosa e se avete ancora delle perplessita' provate a tradurre il l/m inBasic, cioe' in qualcosa del genere: FOR F=16384 TO 18432: POKE F,PEEK (F+256): NEXT F"
Chiaramente questo esempio funziona male perche' la differenza non e' sempre 256 ma in alcuni casi 1760, e poi il valore 18432 limita la routine solo al terzo superiore dello schermo. Come visto da quest'esempio, puo' essere piuttosto difficile scrivere delle routines che operino sullo schermo. Fortunatamente il compito ci puo' essere notevolmente semplificato dato che queste routines esistono gia' e sono scritte nella ROM, per

cui a noi basta richiamarle con un'apposita istruzione paragonabile al GOSUB: CALL con codice CD
Ad esempio all'indirizzo 10h c'e' la routine di stampa di un carattere, e precisamente quello contenuto nel registro A; se noi vogliamo stampare il set di caratteri, il programma e':

```
LD A,32d
LOOP
PUSH AF
CALL 16d
POP AF
INC A
CP 128
JR NZ,LOOP
RET
```

Probabilmente vi state chiedendo il perche' delle due istruzioni di PUSH e POP presenti; la spiegazione e' molto semplice: la procedura per la stampa del carattere altera i valori dei registri per cui se noi non vogliamo perdere A che controlla il LOOP, dobbiamo copiarlo sullo stack. Ricordate dunque prima di richiamare delle routines della ROM, di salvare con dei PUSH i registri che il vostro programma utilizza e di recuperarli dopo la CALL con altrettanti POP.
Le CALL possono anche essere condizionali come i JP e possono anche far riferimento a subroutines scritte da noi
Per alcuni indirizzi particolari, la CALL puo' essere sostituita dalla piu' breve e rapida RST. Per ora citiamo solamente RST 10 che sostituisce in un solo byte i tre di CALL 10h.
Un programma che contiene dei RST 10 o altre chiamate a subroutines che operano sullo schermo, deve cominciare con

```
LD A,2
CALL 1601h
```

che si puo' considerare una routine di inizializzazione dello schermo.
La seconda cosa e' che RST 10 ha la stessa flessibilita' di PRINT CHR$. Quindi si possono anche sfruttare i caratteri di INK (16d), PAPER(17d), TAB(23d) ecc...
Ad esempio

```
LD A,22
RST 10
LD A, 5
RST 10
LD A,7
RST 10
LD A,16
RST 10
LD A,2
RST 10
LD A,66
RST 10
```

e' equivalente a PRINT CHR$ 22;CHR$ 5;CHR$ 7;CHR$ 16;CHR$ 2;CHR$ 66 e avra' l'effetto di stampare una B maiuscola in blu alla posizione 5,7.

CLEAR EVANESCENTE

Un'altra routine per poter effettuare semplice clear dello schermo eseguito resettando tutti i bit del display file (memoria video) in una particolare sequenza.
Una maschera a 8 bit contiene tutti i bit a 1 tranne uno che e' posto a zero. Questo viene fatto ruotare in tutte le posizioni, ed a ogni passo viene fatto l' AND tra questa maschera ed ogni byte del display file.
Il risultato e' tutti i bit di ogni byte sono posti a zero, eseguendo cosi' il clear dello schermo. Alla fine del clear il programma provvede a ripristinare il colore del PAPER corrente (memorizzato nella variabile di sistema ATTR-P alla locazione 5C8DH) sulle prime 22 linee, e quello del BORDER nelle ultime due.Si noti che dopo il clear non e' necessario posizionare il registro HL per puntare alla zona attributi; questo perche' le due zone (il display file e la memoria attributi) sono consecutive.
Il codice macchina e' localizzato a partire dalla locazione 55545 e si estende per 47 bytes. Nessuna Poke e' necessaria per il suo funzionamento.
Guarda il listato

LETTURA GEOGRAFICA DEI TASTI

Questa routine esegue uno scan (interrogazione) multiplo della tastiera ritornando un codice per ogni combinazione di tasti premuti contemporaneamente. Lo scopo e' quello di avere una INKEY$ multipla, cioe' che ritorna lo stato di uno o piu' tasti.
L'utilizzo della routine e' intuitivo, cioe' come routine di input per un tipico Arcade Game (gioco d'azione) dove le possibili direzioni di movimento sono le quattro usuali piu' quelle combinate (diagonali) e l'immancabile tasto di FUOCO.
Per far cio' la routine indaga la porta d'ingresso 255 (FEH) che corrisponde alla tastiera e verifica se uno o piu' tasti (di quelli definiti) sono premuti. In caso affermativo somma ad un contatore il codice relativo al tasto premuto.Nel nostro caso i tasti sono: Q alto, A basso, O sinistra, P destra mentre l'ultima riga della tastiera corrisponde al fuoco. A questi tasti sono associati rispettivamente i valori 8, 4, 2, 1 e 16.Questi valori corrispondono ai valori assunti dai primi cinque bit di un byte, quindi, di fatto, alla pressione di un tasto il suo codice non viene effettivamente sommato con un istruzione di ADD, bensi viene settato (SET) il bit relativo nel byte di codice.
La routine risolve situazioni in cui l'input e' in conflitto (O e P premuti contemporaneamente) escludendo un tasto.La routine e' localizzata nella memoria a partire dalla locazione 55470 e si estende per 49 byte. Il suo funzionamento non richiede alcun parametro. Il codice di ritorno puo' essere preso o con l'istruzione Basic "LET A=USR 55470" oppure, se utilizzata da routine in linguaggio macchina, prendendo il contenuto del registro C.
Guarda il listato

DELETE UTILITY

Lo scopo di questa routine e' quello di fornire un' aiuto al programmatore nella stesura di programmi Basic; consentendo di cancellare interi blocchi di linee in un solo colpoLa routine richiede due valori in ingresso: la linea d'inizio blocco e la linea di fine blocco. Questi valori devono essere posti rispettivamente nelle coppie di locazioni di memoria di indirizzo 23357 e 23359.Le istruzioni Basic per introdurre i valori sono (posto che I e F contengano la linea iniziale e finale):

POKE 23357, I-256*INT(I/256) : POKE 23358, INT(I/256) : POKE 23359, F-256*INT(F/256) : POKE 23360, INT(F/256)

La routine esegue una scansione attraverso il programma Basic ricercando prima la linea corrispondente a quella iniziale (fermandosi se il numero di linea corrente e' maggiore di quello cercato) quindi prosegue a cercare il secondo numero di linea (fermandosi alla linea il cui numero e' immediatamente inferiore a quello cercato se quest'ultimo non viene trovato).Alla fine la routine ha ottenuto il pointer al primo byte della prima linea da cancellare e il pointer al byte successivo all' ultimo dell' ultima linea da cancellare. Questi valori contenuti rispettivamente nei registri DE e HL vengono utilizzati dalla routine in ROM alla locazione 19E5H. La routine in ROM sposta il blocco di memoria a partire dall'indirizzo in HL nella zona di memoria a partire dall' indirizzo DE, cancellando in questo modo le linee indicate.Per capire meglio la routine e' utile conoscere come lo Spectrum memorizza le linee Basic. Ogni linea inizia con due byte che contengono il numero di linea corrispondente (memorizzato come byte-low, byte-high), i successivi due byte contengono la lunghezza della linea stessa (memorizzato come byte-high, byte-low). Quindi c'e' la linea Basic vera e propria terminata dal carattere speciale 0DH (CR o Carriage Return).
Un utile Sideline (utilizzo secondario) e' quello di poter salvare le variabili da sole in un colpo solo. La cosa e' ottenibile facendo eseguire il programma Basic quindi cancellare l'intero programma e salvare le variabili con il comando SAVE ""filename"". Infatti poiche' la routine non tocca la zona variabili queste rimangono inalterate. Per ricaricarle basta eseguire un CLEAR, quindi utilizzare il MERGE sul file di variabili salvato.La routine e' memorizzata a partire dalla locazione 55300 0 byte.
Guarda il listato

FIND & REPLACE UTILITY

Questa routine e' un utile tool per il rimpiazzamento di caratteri in un programma Basic. Sostanzialmente la routine attraversa il Basic cercando il carattere da rimpiazzare e, quindi, effettua la sostituzione con il carattere di replace.I caratteri vengono passati alla routine mettendoli con delle POKE alle locazioni 23352 e 23353; rispettivamente per il carattere da rimpiazzare e per il carattere di rimpiazzo. Le istruzioni Basic necessarie sono:

POKE 23352,CODE "f" : POKE 23353,CODE "r"

La routine viene quindi invocata utilizzando l'istruzione

RANDOMIZE USR 55380 o LET A=USR 55380 (La dimensione e' di 81 byte.)

La routine ignora tutto quello che c'e' dopo una istruzione REM per evitare di toccare un eventuale codice macchina codificato in essa. Poiche' le istruzioni Basic dello Spectrum vengono memorizzate con un codice di un carattere, la nostra routine consente di cambiare intere istruzioni, come, ad esempio, passare da una PRINT a una LPRINT, oppure, togliere momentaneamente tutte le PRINT (per velocizzare un programma) trasformandole in REM.Si tenga comunque presente che la routine non esegue alcun controllo sintattico sulle sostituzioni effettuate e che, quindi, un uso inappropriato potrebbe portare ad una segnalazione del tipo: NONSENSE IN BASIC.
L'uso della routine e' per ora limitato a cambi che coinvolgono l'intero programma Basic. Per realizzare una utility che esegua un Find&Replace su porzioni definite del programma Basic occorre limitare la ricerca nel range (intervallo) desiderato.Per far questo possiamo proporre, all' allievo piu' intraprendente, di eseguire una specie di merge (unione) delle due utility qui presentate
In effetti la routine di Delete esegue sostanzialmente un aggiustamento dei puntatori al blocco di linee da cancellare. Quindi se togliamo l'ultima linea (che cancella effettivamente il blocco) abbiamo rispettivamente in DE e HL i puntatori all'inizio e alla fine del blocco.A questo punto la cosa e' immediata; basta utilizzare questi puntatori al posto di quelli che indicano l'inizio e la fine del programma Basic e il gioco e' fatto!Di fatto quello che devi fare e' togliere la prima linea della routine di Find&Replace, la quale carica in HL il puntatore all'inizio del programma e sostituirla con LD H,D e LD L,E. Quindi invece che caricare DE con il contenuto della variabiledi sistema VARS (5C4BH), che rappresenta la fine del programma Basic, utilizzare il puntatore (in HL) ottenuto dalla routine di Delete e salvato, ad esempio, nelle locazioni 5CB0H e 5CB1H, che sono inutilizzate dallo Spectrum.Ricordate pero' che d'ora in poi, oltre a dare i caratteri di find&replace, dovrete specificare (nella maniera gia' vista) i numeri di linea iniziale e finale del blocco da trattare.
Guarda il listato

ALCUNI CONSIGLI SUI LOADER

L'istruzione MERGE, permette di mischiare il programma gia' in memoria con quello che viene caricato. Questa istruzione inoltre impedisce l'autostart del programma, costituendo un problema per chi sperava di evitare l'accesso di mani estranee al proprio programma semplicemente salvandolo con l'autostart. Esiste pero' una maniera per impedire l'accesso a un programma caricato con MERGE : e' sufficiente mettere del linguaggio macchina nell'ultima linea di programma: il sistema saltera'.
Spesso vengono usate tecniche di questo tipo dai programmatori per evitare che i programmi vengano copiati o 'letti', ma e' sufficiente venire a conoscenza dell'indirizzo da cui viene caricato il file, cosa che si puo' fare disassemblando il programmino caricatore con un monitor e utilizzare la semplice routine in L/M:

```
ISTRUZIONI    SIGNIFICATI
LD IX,start   Carica il reg.IX con l'ind. di partenza
LD DE,length  Carica il reg. DE con la lunghezza
LD A,255      Carica il reg. A con 255S
CF            Setta il flag di carry.
CALL 1366     Chiama la routine di LOAD in ROM
RET           Torna al basic
```

Se invece vogliamo caricare l'HEADER di unprogramma per conoscerne il nome, il tipo,l'indirizzo di partenza,ecc. dobbiamo sostituire LD A,255 con LD A,0
Quindi per poter analizzare il vostro HEADER dovete caricarlo in memoria (p.es. a 30208) e analizzarlo con opportune PEEK come segue:

```
BYTES          SIGNIFICATI
1              Tipo di file:
0              = programma basic
1              = file numerico
2              = file alfanum.
3              = L/M10  Nome del file in ASCII
2               Lunghezza del file
2               Indirizzo della locazione di inizio
2               Linea di autostart se in basic
N.B.: se non ha autostart il suo contenuto e' >=8000h (>=32768)
```

Nel programmino precedente in L/M se volete verificare un file salvato senza header dovete far precedere CCF (complementa carry flag) a CALL 1366.

Introduzione all'hardware dello Spectrum

Anche se di ridotte dimensioni lo Spectrum contiene un sacco di circuiti, che tradotti in transistor basterebbero a saturare la stanza in cui vi trovate.L'ingombro ridotto e' il frutto della tecnologia a LSI (LargeScale Integration) e con meno di 25 integrati lo ZX funziona a meraviglia.Ogni integrato ha una funzione ben precisa e specializzata: il'cervello' che coordina tutte le operazioni e' lo Z80 (CPU); il braccio e' la ULA della Ferranti; l'artista che provvede ai colori e' il codificatore PAL ed infine l'archivio delle informazioni sono la ROM e le RAM." Anche se ogni integrato sa bene cosa fare e' necessario coordinare tutte le operazioni: per questo esiste la ROM che contiene il Sistema Operativo e il Basic. Quando accendete il computer la CPU incomincia a leggere nella ROM cosa deve fare e tutti i circuiti eseguono quello che la CPU ordina. Anche se ai vostri occhi il computer non sta facendo nulla, sta eseguendo parecchie decine di migliaia di operazioni al secondo. Nel frattempo anche il Ferranti lavora come un matto provvedendo alla visualizzazione dei dati in memoria corrispondenti al video. Dal punto di vista tecnico il vostro ZX e' stato progettato in modo poco usuale: molte delle funzioni solitamente svolte da normali integrati TTL del tipo 74LS sono risolte dal Custom di produzione Ferranti e addirittura tutta la procedura di visualizzazione su TV e' espletata da questo 'superintegrato'.Oltre che per motivi di costi di produzione, la Sinclair ha utilizzato questa soluzione per complicare al massimo la vita a tutti i 'copioni'. Per usare un solo circuitone 'fac totum' sono pero' stati necessari compromessi sulla velocita' del computer.Tanto per darvi un'idea di cio' che accade, vi diciamo che il perfido Ferranti sospende il segnale di clock alla CPU quando non vuole essere disturbato durante la visualizzazione della pagina video, e questo capita abbastanza frequentemente, con una riduzione del 20% della velocita' di esecuzione.Un altro impiccio all'esecuzione 'speedy' dei programmi sono gli Interrupts, ovvero dei salti forzati ad un breve programma in l/m ogni tot millisecondi. Allo ZX gli Interrupts servono per controllare la pressione di un tasto e per alterare alcune variabili del sistema; ovviamente non si tratta di tempo sprecato, ma utilizzando delle soluzioni circuitali tradizionali questo non sarebbe stato necessario.A parte il Custom, tutto il resto e' regolare, molto piu' che nello ZX81 per cui era necessario molto lavoro per collegare l'interfaccia piu' semplice e banale.

Alcuni interessanti links sull'hardware dello Spectrum

Interfaccia USB
Smart Card, interfaccia ide (hardisk e floppy), UVY-RGB converter, ROM switch, programmatore Eprom
Compact flash, porta seriale, interfaccia audio AY per 48K (come sul 128K), Ethernet interface
Schema interfaccia floppy Disciple
Interfaccia floppy Plus D
Interfaccia Kempston
Schemi di altre interfacce, schede madri ecc.